

Capturing the Content of Physics Systems, Observables, and Experiments

MICHAEL KOHLHASE

joint work with Eberhard Hilf (Oldenburg) & Heinrich Stamerjohanns (IUB)

School of Engineering & Science

International University Bremen, Germany

<http://kwarc.eecs.iu-bremen.de/kohlhase>

Goals and Results of this work

- **From a MKM prespective** (my angle)
 - **Goal:** Validate MKM technology by extending its scope
 - **Goal:** fulfil the “Math as a test-tube” promise
 - **Result:** content/context markup scales for cross-domain interoperability!
- **From a Physicists perspective:** Participate in MKM technology
 - **Goal:** deep structure, searchability, interoperabiliy, long-term archiving
 - **Goal:** Active Documents: click-through access to experimental data in published documents
 - **Result:** Markup Language for Physics

Desiderata for a Physics Markup Language

- **Idea:** Design a markup language to map research/teaching/communication practice of the field. (center design around principal objects)
 - **Mathematics:** Mathematical Formulae and statements (MATHML, OPENMATH, OMDoc)
 - **Chemistry:** Molecules and chemical reactions (CML)
 - **Geo/Astro:** gathered Data (MARINEML...)
- **Question:** What are the principal objects of Physics?

Physics as a Science of Measurements

The Operational Procedure to gain Physics Insight

DP1 Prepare, read the literature, . . .

DP2 Define an **observable** (This rule separates physics from other fields)

- construct physical device, called an **apparatus**
- with which measurements can be made giving real valued numbers depending on the specific experimental setup.

DP3 Set an **iterative operational construction rule** to refine the apparatus

DP4 Take this construction to its limit

- define the virtual outcome of this ‘ideal device’ as the **physical observable**,
- observables map physical phenomena to mathematical descriptions
- theoretical physics can reproduce previous experiments and predict future ones within the construction-dependent uncertainties of apparata.

(Physics results are “relations between observables”)

DP5 Do a set of experiments, map to theory, check with the predictions

- if all are borne out we have a new **natural law**.
- otherwise the set of assumptions and results are called '**model**'.

DP6 Distribute the results so that the experiments and calculations can be repeated by others.

- Physics results are independent of space and time (and math format)
- They should be repeatable by other physicists at other laboratories
- **Spreading the information** on the findings **is part of the procedure**

Example: Measuring the Position in Space of an Object

- design an 'object detector' covering a finite area $(x_i, \Delta x_i)$
- tile (non-overlapping, touching detectors) the physical space (check all)
- we learn in which detector $(x_i, \Delta x_i)$ the object is to the precision Δx_i .
- Refine experiment: detector set with $\Delta x_{i+1} = \Delta x_i/2$ (improved precision)
- Infinite refinement would give us the ideal physics result.
(but we cannot pay/physically construct this)
- Virtual limit process for the *physical observable* (i.e. map to math)
space position observable \rightsquigarrow vector in Euclidian space.
- Test all mathematical properties by suitable experiments. (adequate?)
test commutative algebra properties, otherwise revise measurement process

Principal Objects for a Physics Markup Language

Observables defined by the operational description of the defining apparatus, and iterative refinement rule.

Apparata An apparatus \mathcal{A} is defined by a detailed description on how to build it, so that others may redo experiments.

- \mathcal{A} can be fully described by all its simultaneously measurable observables a_i
- A full set of values for these defines a **State** $|a_i\rangle$, of \mathcal{A} .
- e.g. \mathcal{A} = **gas-filled bottle**, with observables **gas-density**, **size**, **color**, and **material**, and **temperature**.

Experiments the essence of Physics (repeatable, not data recorded over time)

- conduct experiment by bringing \mathcal{A} into contact with another apparatus \mathcal{B} .
- e.g. thermal contact with **thermometer** \mathcal{B} (neglect change in state $|a_i\rangle$)
- use \mathcal{B} to get information about the state of \mathcal{A} , while varying e.g. **size**.
- call \mathcal{A} **system** and \mathcal{B} **measurement device** (both are apparata)

PHYSML: Observables, Systems, and Experiments

- **Idea:** Extend OMDoc minimally to cover principal concepts of physics
(Abstract Measurement theory (developed for quantum mechanics))
 - **Dimensions** and **Units** (model in OMDoc as type system)
 - **Observables:** defined by sequence of measurements
(new statement-level construct)
 - **Systems:** defined by their observables and states
(new statement-level construct)
 - **Natural Laws** (model in OMDoc as axioms/assertions)
- **Status:** First concrete proposal/case study (EPS working group)

Representing Dimensions and Units in a Type System

- **Idea:** use dimensions as types and units as constructors
- New symbols for the principal and defined dimensions (from the SI system)

```
<theory xml:id="dimensions">
  <symbol name="mass"/><symbol name="length"/><symbol name="time"/> ...
  <symbol name="volume"/>
  <definition for="volume" type="simple"> $length^3$ </definition>
</theory>
```

- Use them as types in expressions for units and quantities.

(dimension-checking)

```
<theory xml:id="units">
  <symbol name="Kelvin"><type system="dimensions"> $temperature$ </type></symbol>
  <presentation for="Kelvin"> $^{\circ}K$ </presentation>
  <symbol name="Celsius"><type system="dimensions"> $temperature$ </type></symbol>
  <presentation for="Celsius"> $^{\circ}C$ </presentation>
  <definition for="Celsius" type="implicit"> $\forall x > 0. x^{\circ}K = (x - 273.15)^{\circ}C$ </definition>
</theory>
```

- More refined type systems? (ranges... OMDoc tolerates multiple types)

Representing Observables in PHYSML

- The objects of Physics are “**Observables**” (**foundational measurement theory**)
 - e.g. mass of a proton, size of our galaxy, orbital escape velocity,...
 - “defined” by sequence of evermore fine-grained measurement regime
 - have a counterpart that is a mathematical object (**algebraic properties**)
 - have a dimension (and sometimes a domain of definition)
- New “definitional schema” (**gives us new symbol**)

```
<observable name="temperature" xml:id="temperature.obs">
  <opdef><CMP>Measure with a thermometer.</CMP></opdef>
  <refinement>
    <CMP>Make the thermometer stepwise smaller.</CMP>
  </refinement>
  <type system="dimensions">temperature</type>
</observable>
<presentation for="temperature"><use format="default">T</use></pres
```

Representing Natural Laws

- **Remember:** Natural laws are relations between observables.
- **Idea:** Observables introduce symbols; use those for representation of natural laws

- **Example:**

```
<axiom xml:id="force_mass_acceleration" type="natural_law">  
  <CMP>Force is mass times acceleration.</CMP>  
  <FMP> $\mathbf{F} = \mathbf{m} \cdot \mathbf{a}$ </FMP>  
</axiom>
```

- **Note:** Natural Laws are mathematically “axioms”, since they cannot be proven

Systems and their States

- **Idea:** Physical Systems are characterized by a (finite) set of observables.

```
<system xml:id="thermometer">
  <realization><CMP>A thin glass tube with mercury in it.</CMP></realization>
  <observable xref="#temperature"/>
</system>
```

- A system prepared for an experiment (fix some values for observables!)

```
<system xml:id="gas_cylinder">
  <realization><CMP>A gas-tight wooden cylinder</CMP></realization>
  <observable xref="#pressure.obs"/><observable xref="#density.obs"/>
  <observable xref="#temperature"/><observable xref="#color.obs"/>
  <preparation><CMP>We make the cylinder red!</CMP></preparation>
  <state><value for="color">red</value></state>
</system>
```

- **Idea:** Measure by bringing these two into (thermal) interaction

Experiments and Measurements

- **Idea:** Experiments bring system and apparatus in to contact

```
<experiment xml:id="ex_pressureXdensity_vs_temp">
  <CMP>Measuring the pressure vs. temperature of a compressed gas cylinder<
  <system xref="#gas_cylinder"/>
  <system xref="#thermometer"/>
  <measurement xml:id="m_213">
    <state of="#gas_cylinder">
      <value for="pressure">332.49586 psi</value>
      <value for="density">19g/l</value>
    </state>
    <state of="#thermometer"><value for="temperature">17.52K</value></state>
  </measurement>
</experiment>
```

Measurements specify input and output states

- **Invariant:** Input obs. \cup prepared obs. \cup contact obs. = system1 states

How can we create PHYSML Documents?

- \LaTeX : semantic \LaTeX (\LaTeX as a content/context markup format)
- \LaTeXML work-flow: (used in our case study)
 - $\text{\LaTeXML} \hat{=} \text{\TeX}$ parser + XML emitter + post-processing pipeline.
- Example: (Just in \LaTeX as usual)

```
\begin{module}[id=units,uses=dimensions]
  \symdef{Kelvin}{\Kelvin}{^\circ\kern-.3em{K}}[type=tempera
  \symdef{Celsius}{^\circ\kern-.15em{C}}[type=temperature]
  \begin{definition}[for=Celsius]
    $\forall x>0.x\text{\Kelvin}=(x-273.15)\text{\Celsius}$
  \end{definition}
\end{module}
```

How can we create PHYSML Documents?

- \LaTeX : semantic \LaTeX (\LaTeX as a content/context markup format)
- L \TeX XML work-flow: (used in our case study)
 - L \TeX XML $\hat{=}$ T \TeX parser + XML emitter + post-processing pipeline.
- Example: (Write in \LaTeX , get this!)

```
<theory xml:id="units"><imports from="dimensions"/>
  <symbol name="Kelvin"><type system="dimensions">temperature</type></symbol>
  <presentation for="Kelvin">Kelvin</presentation>
  <symbol name="Celsius"><type system="dimensions">temperature</type></symbol>
  <presentation for="Celsius"> $^{\circ}\text{C}$ </presentation>
  <definition for="Celsius" type="implicit"> $\forall x > 0. x^{\circ}\text{K} = (x - 273.15)^{\circ}\text{C}$ </definition>
</theory>
```

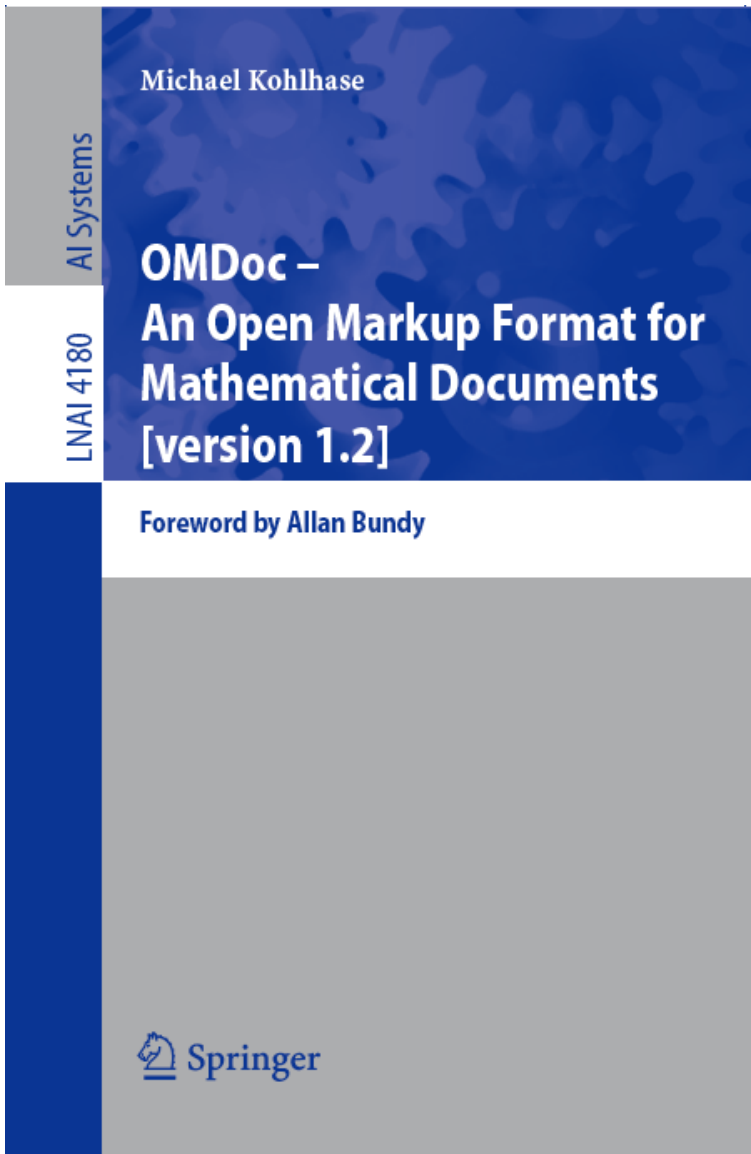
What is E-Science?

- **Observation:** E-Science seems to be a hot topic!
- The term is used for
 - simple multimedia presentation of scientific data,
 - for middleware like the science grid,
 - even for e-learning platforms for the natural sciences.
- In this talk, I want to look at E-Science from a different angle:
 - **Goal:** Computer-Support all of Science (like office-suite)
 - **Method:** Interoperability of software systems by safe communication
 - **Prerequisite:** Semantic transparency of objects/knowledge
 - **Technology:** XML-based Content/Context Markup Formats
 - **Approach:** Start with Math, extend to Science & Engineering

OMDoc in a Nutshell (three levels of modeling)

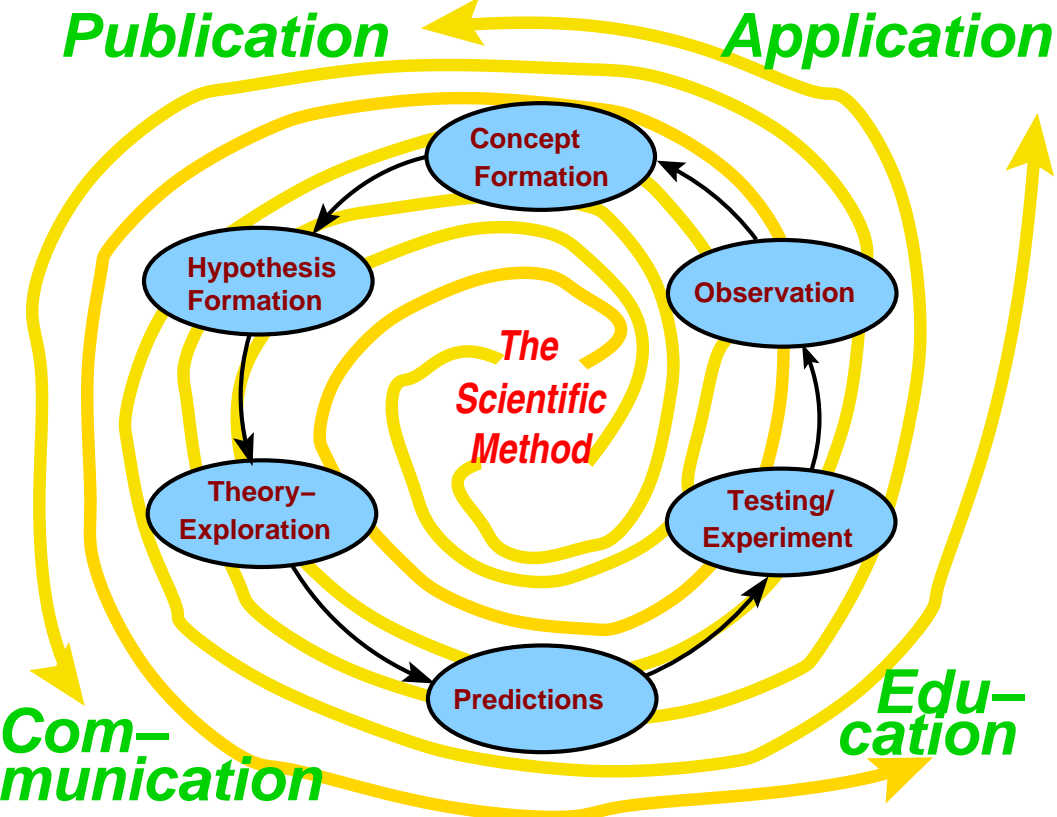
<p>Formula level: OPENMATH/C-MATHML</p> <ul style="list-style-type: none"> • Objects as logical formulae • semantics by ref. to theory level 	<pre><OMA> <OMS cd="arith1" name="plus" /> <OMS cd="nat" name="zero" /> <OMV name="N" /> </OMA></pre>
<p>Statement level:</p> <ul style="list-style-type: none"> • Definition, Theorem, Proof, Example • semantics explicit forms and refs. 	<pre><defn for="plus" type="rec"> <CMP>rec. eq. for plus</CMP> <FMP>X+0 = X</FMP> <FMP>X+s(Y) = s(X+Y)</FMP> </defn></pre>
<p>Theory level: Development Graph</p> <ul style="list-style-type: none"> • inheritance via symbol-mapping • theory-inclusion by proof-obligations • local (one-step) vs. global links 	<p>The diagram illustrates a Development Graph with four theory nodes: Nat-List, List, Nat, and Param. Each node contains a list of symbols: Nat-List (cons, nil, 0, s, Nat, <), List (cons, nil, Elem, <), Nat (0, s, Nat, <), and Param (Elem, <). Relationships are shown as follows: <ul style="list-style-type: none"> Imports: Blue arrows labeled 'imports' point from Nat to Nat-List and from Param to List. Actualization: A green arrow labeled 'Actualization' points from List to Nat-List, with a green dot on the arrow. Proof Obligations: A yellow box labeled 'Proof Obligations' is connected by dashed lines to the 'Actualization' arrow and a green arrow labeled 'theory-inclusion' pointing from Param to Nat. Imports: A dashed blue arrow labeled 'imports' points from List to Nat. </p>

Read all about OMDoc in the new book



- Springer LNAI 4180 (August 2006)
- Primer (quick-start/best practices)
- Specification [version 1.2]
- resources (style sheets, Schema, etc.)
- Applications/Projects (contributed articles)

The “Scientific Method” (Spiral View)



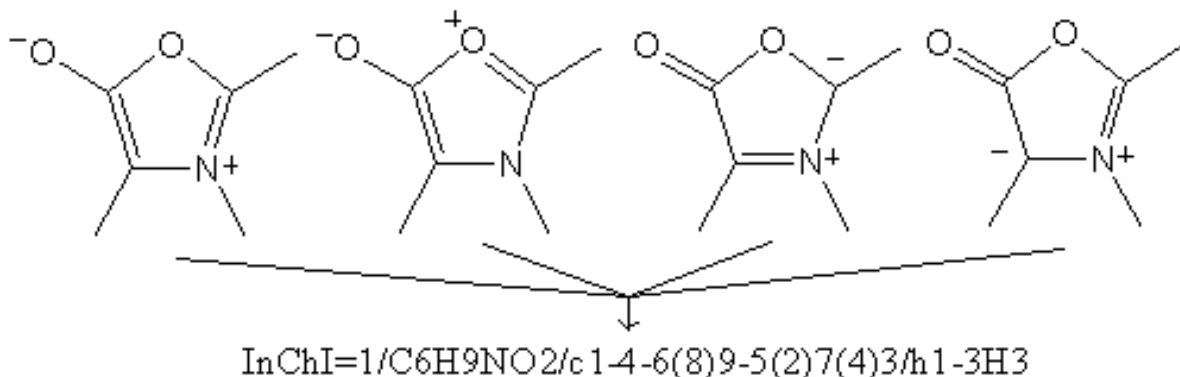
The Scientific Method
in the Creativity Spiral
after [Buchberger 95]

- **Goal** Supported every step by software systems
(Towards an infrastructure for E-Science!)
- **Idea**: concentrate on communication of the objects and knowledge involved
(leave the applications to specialists)

CML: Chemical Markup Language [P. Murray Rust]

- **Beilstein model:** Molecules, Properties and Citations (for observations)
- **CML:** Mark up molecules (name is essentially enough)

e.g. `<molecule
id="benzene"
formula="C6H6" />`



- **Idea:** Define OMDoc theories/symbols for molecules (~10 000 000 known)
 - **Properties** (model as OPENMATH symbols)
 - **Units and Observables** (as in PHYSML)
 - **Reaction Equations** (As OPENMATH equations)
 - **Natural Laws** (model in OMDoc as axioms/assertions)
- **Status:** First surveys (Collaboration with Peter Murray Rust)

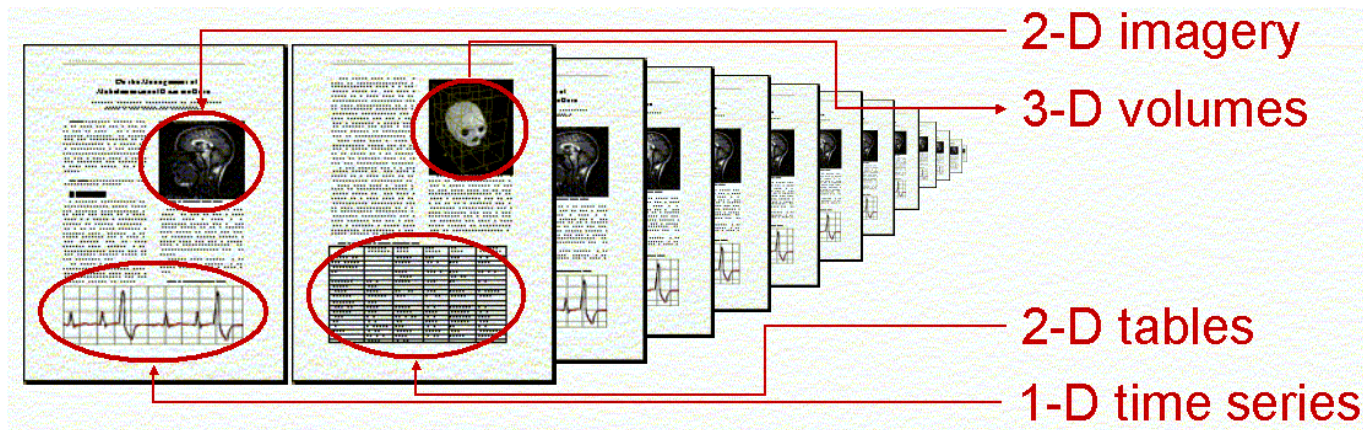
Extension for Raster Data (with IUB [Baumann, Unnithan])

- The objects are multi-dimensional arrays of scalars together with
 - description of the coordinate system (tied to the dimensions)
 - definition of the observed quantities (tied to the cells)
 - description of the measurement process (tied to the object)
- **Extension to OMDoc**: generalize from OpenGIS emerging standards
- **Application**: Scientific documents with full data access
(wouldn't it be nice, if we could click through to the real data)

Active Formal Documents

- Tables, Images, and Diagrams contain important information
- semantic modeling of the underlying data and processes
- Integration of added-value services into the document.

(visualization, management of change, reinterpretation)



- **Example:** Spreadsheet program (value vs. formula, pie-charts, ...)
- **Future:** Active technical documents, formal digital library

Conclusions

- Towards an E-Science **infrastructure** based on **communication of knowledge**
- **Approach**: Start with **content/context markup** for **math** and **physics**.
- **OMDoc** standardized communication language (based on XML)
- gives us the beginning of a **E-Science assistant**
 - integration of external software systems (make use of the Internet)
 - a common ontology for integration (is zero a natural number?)
 - a universal math/-Science repository (data mining, semantic search)
 - bookkeeping in experiments and theories (management of change)
- **Slogan**: Light-weight formal methods for added-value applications