Capturing the Content of Physics:
Systems, Observables, and Experiments

Eberhard Hilf[1], Michael Kohlhase[2], and Heinrich Stamerjohanns[2]

[1] Institute for Science Networking, Oldenburg
`hilf@isn-oldenburg.de`
[2] Computer Science, International University Bremen
`m.kohlhase@iu-bremen.de`
`h.stamerjohanns@iu-bremen.de`

Abstract. We present a content markup language for physics realized by extending the OMDOC format by an infrastructure for the principal concepts of physics: *observables*, physical *systems*, and *experiments*. The formalization of the description of physics observables follows the structural essence of the operational theory of physics measurements. The representational infrastructure for systems and experiments allow to capture the distinctive practice of physics: natural laws are supported by evidence from experiments which are described, disseminated and reproduced by others.

1 Introduction

The distributivity of information and services over the Internet has changed all aspects of life, and science is not an exception. We anticipate that the systems currently investigated in the community will eventually change scientific practice and that they will have a strong societal impact, provided that they can interoperate to cover the whole work-flow of scientific research, education and application.

To further this vision we need to develop, implement, and provide semantic-based and context-aware techniques for acquiring, organizing, processing, sharing and using knowledge in science.

Our starting point is the view of the *scientific method* as a spiral (see Fig. 1), where we have our focus on physics here. In this view, scientific research in physics moves in a spiral trajectory from original ideas to results and even applications. Ideas pass through the processes of observation of natural pro-

Fig. 1. The Scientific Method

cesses, then of concept formulation to describe these. These allow scientists to
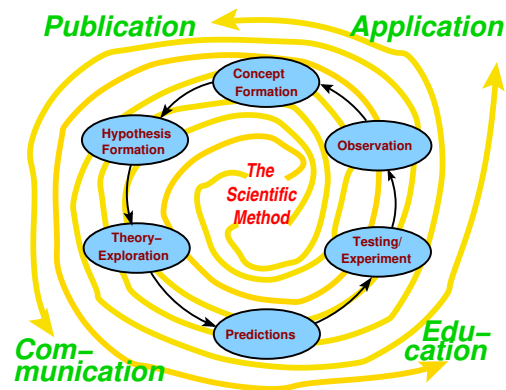
express initial theories about (quantitative laws of nature governing) them, which are then explored (what are the consequences of the model assumptions) leading to predictions about processes that can be verified or falsified (to a certain degree) experimentally. These experiments usually lead to new observations, starting the next round in the spiral until a quantitative (mathematically formulated) *theory* predicting exclusively correct results from experiments is formulated. Observables in physics have to be suitably found such that they can be physically measured, their algebraic counterparts being then candidates for building stones of a theory. The semantics of mathematics as such is more confined, searching for logically correct sets of rules.

At the moment, most of the steps in Fig. 1 are separately supported by software systems, e.g. literature searches in GOOGLE SCHOLAR or WIKIPEDIA, theory exploration in computer algebra systems like MATHEMATICA, and experiments in simulation systems. But the systems are, by and large, not able to inter-operate since they use differing data formats, make differing model assumptions, and are bound to an implicitly given context that is only documented in publications about the systems. For instance, copy-and-paste from GOOGLE SCHOLAR or WIKIPEDIA to MATHEMATICA or a simulation system is impossible because of this format problem. Moreover, where possible, copy and paste can be very dangerous, since computer algebra systems make differing assumptions on the Computercode-libraries, the simulation systems are based on[3].

We are set here to arrive at a content markup format for physics. Early concept discussions and visions [Hil05a,ERH05,HMS03,Hil05b] have not led to a realization in terms of an encoding, since the problem was attacked from the ground up. In this paper we will build the bridge from vision to a usable markup language by extending the OMDOC (Open Mathematical Documents) format [Koh06b] by an infrastructure for (physical) systems, observables and experiments and call this new module and the extended system PHYSML (Physics Markup Language). Since we can now share all the infrastructure — in particular the theory and statement levels — with mathematics, the language design for PHYSML becomes feasible.

2   Desiderata for a Physics Markup Language

The design of a semantic markup language for a learned field is more sophisticated than it might seem. The reason is that, in order to be useful it has to map the way research is organized. This leads to language designs centered around the *principal objects* of the respective research field. In chemistry, the Chemical Markup Language CML [CML05] was designed with the *name of a molecule* as

---

[3] A simple example, where the lack of explicit context led to a very expensive failure was the September 1999 loss of a $125 million Mars orbiter, which crashed on Mars. The cause was that NASA used for its specifications metric units, but the Lockheed Martin engineers misinterpreted the data assuming they were given using Imperial units of measurement.

principal object, to which properties and its chemical reactions are attached as properties[4].

In mathematics, the earliest discipline to have a dedicated markup language, we have MathML and OpenMath as markup formalisms that take mathematical objects as principal objects. The OMDoc format [Koh06b] extends them with content markup for *statements* (like definitions, axioms, theorems, and proofs) and *theories* (conceived as principal objects on a higher level).

In physics, since the times of Galilei (‘*Experimental results are the highest authority*’), the young Einstein (‘*A theory is to be accepted if it describes and predicts all possible experiments,— independent of the feelings and ‘intuition’ of the scientist*’ [EB72]), and most important of P. Bridgeman [Bri27], who elucidated first the logical steps of physics learning, by analyzing the *operational* steps in doing research, it is now consensus [Mit70,Fal70b,Fal70a,Sak93] that research is accepted as physics if it does physical experiments with apparatuses which represent physical observables. This simple sounding requirement will be the entry point for us to design a specific Physics Markup Language, a construction which tries to mirror the way physicists operationally think.

2.1 Physics as a Science of Measurements

We start the same way as a physicist would enter a new field: by *operationally* following the consecutive steps:

DP1 Decide to work on a specific field, and gather *’pre-scientific’ available knowledge*[5]

DP2 Define an *observable*; In physics this has to be done by *constructing a physical device*[6], called an apparatus, with which measurements can be made giving real valued numbers[7] depending on the specific experimental setup.[8]

DP3 *Set an iterative operational construction rule* to refine stepwise the design of the apparatus such that (just as in ‘proof by induction’) by applying the rule iteratively, it will be accepted that successively more precise apparatuses can be built in principle[9]

---

[4] We think that an alternative approach would have had more merits, to designate *chemical reactions* as the principal objects — the principal action, a chemist does.

[5] By this we concentrate on fields of interest, where we at least *assume* that by preparing physical experiments we may gain new knowledge.

[6] Historically, in the 1960s, there has been a long debate, whether in classical Mechanics, in contrast to all other fields, instead of *building* a physical device, already the *description* of how to build that device is sufficient [Mit70]

[7] In the modern *Theory of Measurements* observed numbers are to be mapped to *Eigenstates* of a *Hermitian Operator*, which is the mathematical image of the physical apparatus.

[8] This rule separates physics from other fields, such as mathematics.

[9] This absolutely essential rule assures that we stay with doable physics experiments. The condition that the rule has not to depend on the status of actual refinement assures that the limit (see next rule) to a virtual ideal mathematical counterpart of the observable will be secure and correct.

DP4 *Take this construction to its limit*, and define the virtual outcome of such an 'ideal device' as the *physical observable*, which then can directly be related (mapped) to the respective mathematics.[10] By this mapping to mathematical objects and their algebra, theoretical physics can be done with the aim to reproduce all previously conducted experiments and correctly predict any doable experiment in the field within the construction-dependent uncertainties of the apparatuses.

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 in a way so that the experiments and calculations can be repeated by others in the world*. Physics results (relations between observables) are independent of representations chosen for the mathematical objects needed, and independent of where and when (space and time chosen). They should be repeatable by other physicists at other laboratories in the world. Therefore the actual spreading of the information on the findings to other laboratories in the world *is* part of the operational procedure to gain physics insight.

To strengthen our intuition about the crucial step **DP3**, let us consider an example: Assume we want to measure the position in space of a physical object in classical mechanics. First we design a physically constructible 'detector' covering a finite space area $(x_i, \Delta x_i)$ which can distinguish whether the object is inside the detector area or not. Then we buy very many of these detectors and plaster (non-overlapping, touching detectors) the physical space. By checking all of them we learn in which detector $(x_i, \Delta x_i)$ the object is to the precision Δx_i. Repeating the experiment but with (may be a more expensive) detector set with *finitely* smaller detector space, say $\Delta x_{i+1} = \Delta x_i/2$ will give a better precision of the experiment. Repeating the application of the rule, which is obviously independent of the absolute value of precision gained in a certain step, would give us the ideal physics result. However we cannot experimentally do or pay for many refinement steps, and have to fear that the correctness of the experiment will break down if we physically go too far. That is why the limit process for the *physical observable* is done by virtually, not physically, going to the limit and mapping the result to a mathematical object as the mathematical representative of the observable. Each of the assumed algebraic properties has then to be tested by respective physical experiments. Thus only after experimental testing e. g. all commutative algebra

---

[10] A Hilbert operator for the ideal apparatus, a Hilbert State for its actual physical momentary realization, and *eigenvalues* for its measurement results. We confess that in practice, most scientists use real continuous variables for convenience, say for the position in space of a classical mechanical object, but with the strategy given here, we assure that we arrive at the correct quantum mechanics first and gain the classical mechanics statement by averaging over space from there using the standard Ehrenfest principle. The price for the convenience is high: we have to use Banach spaces instead of Hilbert space, any proof has to be done by iterating back to Hilbert space, use of distribution theory instead of functions, etc.

properties of the mathematical representative of the space position observable we can identify it with a vector in Euclidian space.

In short, we need the process given in the **DP** steps to ensure the choice of the related mathematical object and to get the best strategy for a semantic encoding of physics in a markup language. But how does this formal *operational definition* fit to the actual practical fixing of physics observables by international committees, e.g. the CGPM (Conference Generale des Poids and Measures), CODATA, IUPAP (International Union of Pure and Applied Physics), and SUNAMCO (Standards, Units and Nomenclature, Atomic Masses and Fundamental Constants)? This question is the domain of *Metrology*, an active research field of its own (see [Pen06] for a recent summary). The international metrologic commissions dwell on the next step of fixing observables once the operational definition has been set, focusing on

precise measurement procedures extending the practical measurement of observables to very large and very small scales is achieved (for the length scale from cosmological to subatomic).

determining physical constants by finding quantitative natural laws which connect real observations and thus can be reformulated to define a *physical constant* which is given by a physical process (such as the gravitational constant, the speed of light, etc.). Examples are: the scale for the time is set to be the *second* fixed as 9.192.631.770 periods of the hyperfine split light radiation of the atom *Cesium*. The *metre* is the length of the path traveled by light in vacuum during a time interval of 1/299.792.458 of a second, thus replacing the Ur-metre at Paris measured by a length ruler.

hunting for higher precision which is especially necessary when long time unique series of measurements of given observables have to be trusted such as in geophysics, astrophysics.

2.2 Principal Objects for a Physics Markup Language

Given the above, we have to model the following principal concepts in a content/context markup language for physics.

Observables As described, an observable is defined by the operational description of the defining apparatus, an iterative refinement rule, and properties such as dimension, scale, and attached algebraic object. The relations in which this observable occurs, etc. can be represented in OMDoc.

Experiments Physics is distinct from other sciences by strictly sticking to reproducible experiments' outcomes as the source of knowledge. Reproducible means: to be able to tell others about the experiment so that they can reproduce it. This is in contrast to other sciences such as meteorology, history, or biology, which have records (data recorded over time) as principal source.

Apparatuses Experimental measurements are done using apparatuses. An apparatus $\mathcal{A}$ is defined by a detailed description on how to build it, so that others may redo the experiment. Alternatively an apparatus can be fully

described by *all* its simultaneously measurable observables[11]. A set of given values for all its properties defines a **State** $|a_i\rangle$, $i = 1, 2, 3, \ldots$ of $\mathcal{A}$. An experiment is conducted by bringing $\mathcal{A}$ into contact with another apparatus $\mathcal{B}$. The logical asymmetry of a typical experiment comes only with the mind of the observer, the experimentalist. She uses $\mathcal{B}$ to get information about the state of $\mathcal{A}$.

Again, an example is in order: Assume we have an apparatus *gas-filled bottle*, with a set of observables such as density of *gas*, *size*, *color*, and *material*, and one thermostatic observable, the *temperature*. We choose $\mathcal{B}$ to be a device which mostly has the same observables, such that if brought into thermal contact with $\mathcal{A}$ it does not affect the properties of $\mathcal{A}$ significantly but adapts its temperature to that of $\mathcal{A}$. We call $\mathcal{B}$ a *thermometer* and $\mathcal{A}$ a *system* in this case. Our interest here is on the value shown by the thermometer as a result of the value of the temperature of the apparatus. We neglect the (inevitable) changes of other observable's values, both of the system and the measurement device: such idealizations of experiments are common in physics.

3 Extending OMDoc for Physics

In this section we will extend the OMDoc format[12] by an infrastructure for (physical) systems, observables and experiments.

With the existing representational infrastructure in OMDoc we can already represent structured collections of interrelated concepts and statements about them via OMDoc `theory`[13] contexts. One of the central concepts in physics, the theory of measurable quantities can be set up in this way using OMDoc `symbol`s.

We start with a simple example, the dimensions of the SI units.

Listing 1.3. Introducing Basic Concepts in a OMDoc Theory

<theory xml:id="dimensions">

---

[11] We note that in physics the list of properties of an apparatus is either finite or countably infinite (in contrast to e.g. biological systems). This assures a Hilbert space of states and real numbered values for the observables as the eigenvalues of the Hermitean Operator representing the Observable. This restriction to at most countable infinite property list is absolutely essential for physics. Only by that we get, after mapping to the formal mathematical context the correct observation that in all physics experiments measured numbers are real, as assured by the Hilbert state space and the Hermitean Operators therein.

[12] Due to space restrictions we cannot introduce the format here; we refer the reader to [Koh06b] for the language definition and examples.

[13] The nomenclature in mathematics, which gave rise to the element names in OMDoc and the naming conventions in physics clash here. In physics a set of assumptions about the physical world are called a "model" until they are generally accepted, only then are they called a "theory" (e.g. the Nuclear shell-model; however: quantum theory, general relativity theory).

Element	Attributes		Content
	Req.	Optional	
`observable`	`name`	`algebra, xref`	`metadata?, opdef, refinement, type?`
`refinement`		`xml:id, xref`	`metadata?, CMP*, FMP*`
`opdef`		`xml:id, xerf`	`metadata?, CMP*, FMP*`
`system`		`xml:id, xref`	`metadata?, realization?, observable*,` `preparation?, state?`
`realization`		`xml:id`	`metadata?, CMP*, FMP*`
`preparation`		`xml:id`	`metadata?, CMP*, FMP*`
`state`	`of`	`xml:id, xref`	`metadata?, value*`
`value`	`for`	`xml:id, xref`	$\langle\langle mobj\rangle\rangle$
`experiment`		`xml:id, xref`	`metadata?, CMP*, FMP*, measurement*`
`measurement`		`xml:id, xref`	`metadata?, state, state`
`evidence`	`for, type`	`xml:id,`	`metadata? CMP*, FMP*,interpretation`
`interpretation`		`xml:id`	`metadata?, CMP*, FMP*`
where `metadata, CMP, FMP and type are` OMDoc `elements described in [Koh06b] and where` $\langle\langle mobj\rangle\rangle$ `is (OMOBJ |m:math |legacy)`			

Fig. 2. The Structure of PhysML Elements

<symbol name="mass"/><symbol name="length"/><symbol name="time"/>
<symbol name="charge"/><symbol name="temperature"/>
<symbol name="volume"/>

<definition for="volume" type="simple"> $\boxed{length^3}$ </definition>

</theory>

We can introduce derived dimensions like the dimension for volume as defined concepts. Note that all of the `symbol` declarations make the concepts available for the use in OpenMath-encoded formulae via `OMS` elements and for the markup of technical terms via the OMDoc `term` element. Both identify a concept by its `name` and home theory (called a *content dictionary*; hence the attribute `cd`). Here as in the following, we use mathematical notation in boxes to abbreviate the OpenMath objects in the listings to save space.

We will use these dimensions as a type system for quantities, and introduce the units as constructors for the dimensions (note that we introduce the symbols with a `type`[14]).

Listing 1.4. A Theory of SI Units

<theory xml:id="units">
  <symbol name="gram"><type system="dimensions"> $\boxed{mass}$ </type></symbol>
  <symbol name="Kelvin"><type system="dimensions"> $\boxed{temperature}$ </type></symbol>
  <presentation for="Kelvin">K</presentation>
  <symbol name="Celsius"><type system="dimensions"> $\boxed{temperature}$ </type></symbol>
  <presentation for="Celsius">$^\circ C$</presentation>

---

[14] In the example, we have not executed this, but it is possible to extend the type system to model ranges of numerical values in quantities in this type system: Instead of simply specifying that the unit K is of type `\temperature` we give K the complex type $\langle temperature, \mathbb{R}^*\rangle$ and adjust the dimension-types of the arithmetic operators, so that they check for range admissibility. This puts a considerably higher load on the type checking algorithm, but gives more control and quality assurance. As OMDoc encoding tolerates multiple type systems, we need not even choose one, but can accumulate the knowledge in the representations and use the one appropriate to the task at hand.

```
<definition for="Celsius" type="implicit"> $\forall x \neq 0. K = (x - 273.15)^{\circ}C$ </definition>
</theory>
```

As usual, we can define the intended notation of a concept via `presentation` elements (see section 4) and we can introduce derived units via definitions. With this machinery, we can also state natural laws:

Listing 1.5. A Natural Law Expressed as an OMDOC `axiom`.

```
<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 that in OMDOC terminology we are dealing with an axiom, i.e. with an assertion that cannot be mathematically proven[15] but has to be assumed about the world. In physics a relation between observables has to be supported by sets of experiments, with no counter-evidence within the range of the variables of the involved observables.

3.1 Observables

Above we have determined the notion of an *observable* as a primary object of physics. As any observable — e.g. the temperature, or velocity — of a given physical system can be used in formulae describing the system, we need to extend the OMDOC format by a new statement-level language element that is definition-like. The `observable` element introduced by the PHYSML module in OMDOC (see Figure 2 for an overview) has three relevant children[16] `opdef`, `refinement`, and `type`, to model the properties of observables we have identified in Section 2. The `opdef` and `refinement` elements contain *mathematical vernacular*, i.e. structured text interspersed with mathematical formulae. Mathematical vernacular is represented in OMDOC by a multilingual group of `CMP` (commented mathematical property) elements with mathematical text, and (possibly) a multi-system group of `FMP` elements with formalizations of the properties expressed in the `CMP`s. The `opdef` element is used for describing the *operational definition* of the observable, i.e. the defining process of measurement, whereas the `refinement` element is used to specify the rule of iterative refinement that takes the measurement process to its (idealized) limit.

The *dimension* of the observable is specified as a `type` element. Here we can directly use the type system for dimensions we have introduced in the last section. In our example in Listing 1.6 this is just the temperature.

---

[15] There may be physical evidence that supports it though.

[16] Here and in the following, we will not explicitly describe the `metadata` element, which is used in OMDOC to accommodate bibliographic and administrative metadata, specifying titles, digital rights, licensing, authorship, timestamping, etc. or the `xml:id` attribute which is used for identification. Details can be found in the OMDOC specification [Koh06b].

The `observable` element carries a `name` attribute, which is used by OM-Doc to introduce a symbol that can be referenced by an `OMS` element just like the `symbol` element. Furthermore, it carries an optional `algebra` attribute that contains a pointer to an OMDoc representation to the mathematical object introduced by this observable.

All of these elements also carry an optional `xref` attribute that allows to refer to an already existing representation of the same element via an URI reference; the effect is that the referred object is virtually copied in to the place of the referring one.

Listing 1.6. An Observable for the Temperature

```
<observable name="temperature">
  <opdef><CMP>Measure with a thermometer.</CMP></opdef>
  <refinement><CMP>Make the thermometer stepwise smaller.</CMP></refinement>
  <type system="dimensions"> temperature </type>
</observable>
```

3.2 Physical Systems and their States

One of the basic building block of PHYSML is the `system` element that is used to represent a physical system. The system is described via the mathematical vernacular in a `realization` element which is the first relevant child. As we have seen above, a physical system can be characterized by a (in practice very finite) set of observables, i.e. physical variables that can be measured independently. These are represented by a non-empty set[17] of `observable` children. Listing 1.7 shows a very simple system, which we will use as a concrete measuring apparatus later.

Listing 1.7. A Simple Physical System

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

In this setup, we have represented only the observable we are interested in: all other physical traits of the apparatus are irrelevant for our current purposes. If other physical properties also matter, then we can add other observables. However, we have to make sure that we fix the states of all of the observables that

---

[17] Enjoy the special cases: By use of an apparatus, which cannot measure anything (that is: has no observable) one cannot learn anything. The respective mathematical operator would be the identity. Less trivial is the case, where we prepare a system in state $|a\rangle$, then try a measurement 'is the system in state $|a\rangle$'? If it is already in that state, one does not learn anything new, and that is: no-one can decide whether the experiment took place or not. Example: heat a system and a thermal measuring device to 40 deg. Then measure the temperature of the system by the device: Your result 40 deg can by no means be distinguished from the suspicion you did not do the experiment.

we do not want to measure. This can be done informally in mathematical vernacular in the optional `preparation` element, which may follow the `observable` elements, and more formally in a `state` element. A `state` element specifies a set of values for observables in the system it refers to (either its parent `system` or the system specified to in the optional `of` attribute) via a set of `value` children. A `value` element specifies the observable it refers to by referring to it's name in the required `for` attribute. Its content is a representation of a physical quantity as an OPENMATH, content MATHML, or OMDOC `legacy` element. In the example below, we have (somewhat arbitrarily) prepared a gas cylinder for an experiment by making it red.

Listing 1.8. A Physical System Prepared for an Experiment

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

3.3 Experiments

Physical experiments are represented by the `experiment` element in PHYSML. The body of this element consists of two `system` elements followed by a set[18] of `measurement` elements. The first child represents the system which is measured, the second the measuring device. The `measurement` elements contain two `state` elements as described above which correlate the state of the system on which the measurement is performed with the state of the system of the measuring device. In the following example, we represent the result of measuring the temperature of a gas cylinder with varying density and pressure.

Listing 1.9. Experiment: measuring the temp. of a gas cylinder

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

---

[18] We explicitly allow an empty set of measurements here in order to describe future, planned or failed experiments that have not yielded measurements (yet).

Note that this only represents the raw data from an experiment. We can link experiments and natural laws, such as the one stated in Listing 1.5 via the `evidence` element. The main insight here is that as we cannot "prove" natural laws, but only observe them. We can only keep on experimenting in physics and collect evidence or counter-evidence for any relations between observables. The `evidence` element contains a non-empty set of `experiments` followed by an `interpretation` element that allows to detail any interpretative steps, e.g. an account how the data was fitted to a curve, etc. Its `for` attribute specifies the relation it concerns, and the `type` attribute specifies whether the evidence supports it (value `for`) or falsifies it (value `against`).

In reality one is left with a residual ambiguity because physical experiments are conducted with real apparata, while the physics law gives a mathematical relation between the idealized quantities of the physical observables and apparata obtained as the (virtual) limit of the stepwise refinement iteration rule.

4 Reading, Writing and Arithmetic with PHYSML Documents

Of course, the XML-based PHYSML format presented here is not directly suited for humans to read and write. And indeed it is not intended to be; humans should use adaptive presentations for reading and invasive editors [KK04] for manipulating PHYSML documents.

The OMDOC style sheets have been extended appropriately for the PHYSML-specific elements. With these, PHYSML documents can be converted to XHTML documents with MATHML formulae that can be displayed in a browser or to PDF documents for printing via the LATEX formatter.

PHYSML inherits a well-established notation declaration language and presentation system from the OMDOC format: for new concepts that are introduced via `symbol` elements notation information can be specified via OMDOC `presentation` elements: In the presence of the following declaration,

```
<presentation for="#Celsius">
  <use format="html|pmml">&#x000B0;C</use>
  <use format="TeX">{}^{\circ}C</use>
</presentation>
```

The OPENMATH object representing the temperature in of the thermometer in Listing 1.9 will indeed look like the visualization in the box.

To write PHYSML documents, we have concentrated on the LATEX workflow that is well-established in physics. Concretely, we have extended the semantic TEX system sTEX [Koh06a] by PHYSML functionality.

Listing 1.10. Writing the PHYSML for Listing 1.4 in sTEX

```
\begin{module}[id=units,uses=dimensions]
  \symdef[type=$\mass$]{gram}{g}
  \symdef[type=$\temperature$]{Kelvin}{K}
  \symdef[type=$\temperature$]{Celsius}{{}^{\circ}C}
  \begin{definition}[for=Celsius]
    $\allcdot{x>0}{x\Kelvin=(x-273.15)\Celsius}$
```

```
\end{definition}
\end{module}
```

For more choice in invasive editors, we will extend the OMDoc wiki system [LK06] and the PowerPoint plugin for OMDoc [KK04] to PhysML.

The explicit, and standardized content representations for physical documents in PhysML will allow us to offer added-value services that cannot be offered on conventional representations. Examples are the dimension check comparing the physical dimensions, and the units used in an equation presented in a paper. If the dimensions on both sides of an equation do not match (say *kg* on one side, and *meter* on the other, the equation is physically openly wrong, if different units for the same dimensions were used on both sides this is called 'unlawful sloppiness' (say K on one side, $°C$ on the other). Other checks will include the algebraic matching of both sides of an equation (say if *vector* on one side and *coaxial vector* on the other, this equation is bluntly incorrect). But more intelligent codes could also read the semantics delivered and offer mapping of algebraic results in different representation (say: integral instead of differential formulation, vector vs. vector-component or exterior form, etc.) thus directly assisting the reader to not having to read clumsy formulations of theoretical results from old times, but get it in the present used representations and notations.

5 Conclusion and Further Work

We have demonstrated that a Markup Language for the *content* of physics can be designed by extending the content and context markup format OMDoc with a representational infrastructure for the principal objects of physics: observables, systems, and experiments. The resulting language PhysML is able to catch the logical and operational structure specific to physics, differentiating this field from others. The extension presented in this paper is part of the ongoing enterprise to extend the OMDoc format to the **STEM** fields (Sciences, Technology, Engineering and Mathematics).

The next step is now to evaluate the language by marking up a larger body of knowledge in physics in PhysML. We have started work on the technically ubiquitous and basic field of thermostatics. This should give us a clear indication whether PhysML is adequate for all of physics, or pinpoint the necessary changes to the language design. An international collaboration on the further development of PhysML is looked for, including experts from theoretical and applied physics and related fields, in particular mathematics and chemistry.

New and powerful services can be implemented once the scientific content can be semantically encoded, retrieved, and reused digitally. In physics, these include the search for other experiments on the same observables, dimension and algebraic checking of mathematical equations, mapping to other mathematical representations of the same theoretical physical expression, etc.

Using the approach of analyzing the operational and logical practices of a scientific discipline field, and map this to field-specific modules extending the se-

mantic markup language OMDOC will allow to spread semantic content markup to other scientific fields.

With authors to increasingly make use of markup languages, and retrieval engines following suit to offer intelligent search algorithms making use of the known markup languages, users will gain effective tools to increase the reachout of their scientific work, having the *content*, not just the text, of the work of others at their fingertips.

References

[Bri27]   Percy Bridgeman. The logic of modern physics. *Comment: the book could not be traced but is well cited*, 1927. and all publications following this on operational foundation of the theory of measurement.

[CML05] Chemical Markup Language CML. Web page at `http://www.ch.cam.ac.uk/CUCL/staff/pm.html`, seen July 2005.

[EB72]   Albert Einstein and Max Born. *Briefwechsel 1916 - 1955*. Rowohlt Verlag, Reinbek, 1972.

[ERH05] Julika Mimkes Eberhard R. Hilf. Zu einem verlustfreien Publizieren und Archivieren. Web page at `http://www.isn-oldenburg.de/~hilf/vortraege/mathdiss02/`, seen July 2005.

[Fal70a] Gottfried Falk. *Theoretische Physik I und Ia*. Heidelberger Taschenbücher. Springer Verlag, 1970.

[Fal70b] Gottfried Falk. *Theoretische Physik II Thermodynamik*. Heidelberger Taschenbücher. Springer Verlag, 1970.

[Hil05a] Eberhard R. Hilf. Kann man TEX beibringen, Physik zu verstehen? In *Der Aufbau eines deutschen TeXDoc-Centers an der SUB Göttingen, — Chancen von TeX-Dokumenten für eine dauerhafte Verwertung und Publikation.* Workshop des DFG-TeXDocC-Projektes, Die Deutsche Bibliothek Frankfurt, seen 2005-07-24 2005.

[Hil05b] Eberhard R. Hilf. Physml. In *Sesame 2005 Workshop Bremen*. MKM Mathematical Knowldege Management, seen July 2005. Web page at `http://www.mkm-ig.org/meetings/sesame05/program.html`, talk at Web page at `http://www.isn-oldenburg.de/~hilf/vortraege/sesame05/index.html`.

[HMS03] Eberhard R Hilf, Julika Mimkes, and Helmut Schottmüller. Die Zukunft des wisenschaftlichen Publizierens — Vom Publizieren zum Austausch von Informationen in der Wissenschaft. Web page at `http://www.isn-oldenburg.de/~hilf/vortraege/guestrow03/`, 2003.

[KK04]   Andrea Kohlhase and Michael Kohlhase. CPoint: Dissolving the author's dilemma. In Andrea Asperti, Grzegorz Bancerek, and Andrej Trybulec, editors, *Mathematical Knowledge Management, MKM'04*, number 3119 in LNCS, pages 175–189. Springer Verlag, 2004.

[Koh06a] Michael Kohlhase. A LATEX-based workflow for omdoc. [Koh06b]. to appear, manuscript at `http://www.mathweb.org/omdoc/pubs/omdoc1.2.pdf`.

[Koh06b] Michael Kohlhase. OMDOC *An open markup format for mathematical documents (Version 1.2)*. LNAI. Springer Verlag, 2006. to appear, manuscript at `http://www.mathweb.org/omdoc/pubs/omdoc1.2.pdf`.

[LK06]   Christoph Lange and Michael Kohlhase. A semantic wiki for mathematical knowledge management. In *Proceedings of the 1st Workshop: "SemWiki2006 - From Wiki to Semantics"*. 2006.

[Mit70] Peter Mittelstaedt. *Klassische Mechanik*, volume 500/500a of *Hochschul-taschenbuecher*. Bibliographisches Institut, 1970.

[Pen06] L. R. Pendrill. Metrology: time for a new look at the physics of traceable measurement? *Europhysics News*, 37(1):25, 2006.

[Sak93] J. J. Sakurai. *Modern Quantum Mechanics*. Prentice Hall, 1993. ISBN 0201539.