

Towards a **collaborative work environment** with **services** for **scientific knowledge management**, both for scholars and for researchers by extending a given system by **domain-specific ontologies** and designing **services** on top of an **ontology abstraction layer**

Problem Statement

Many **collaborative work environments (CWEs)** for **scientific knowledge** out there (Wikipedia, PlanetMath, Connexions), but they lack **semantic services** and thus cannot offer **sophisticated knowledge management**.

'''Pythagoras' Theorem''' is a relation among the three sides of a rectangled [[triangle]].

== Equation ==
$a^2 + b^2 = c^2$

== Proof ==

...

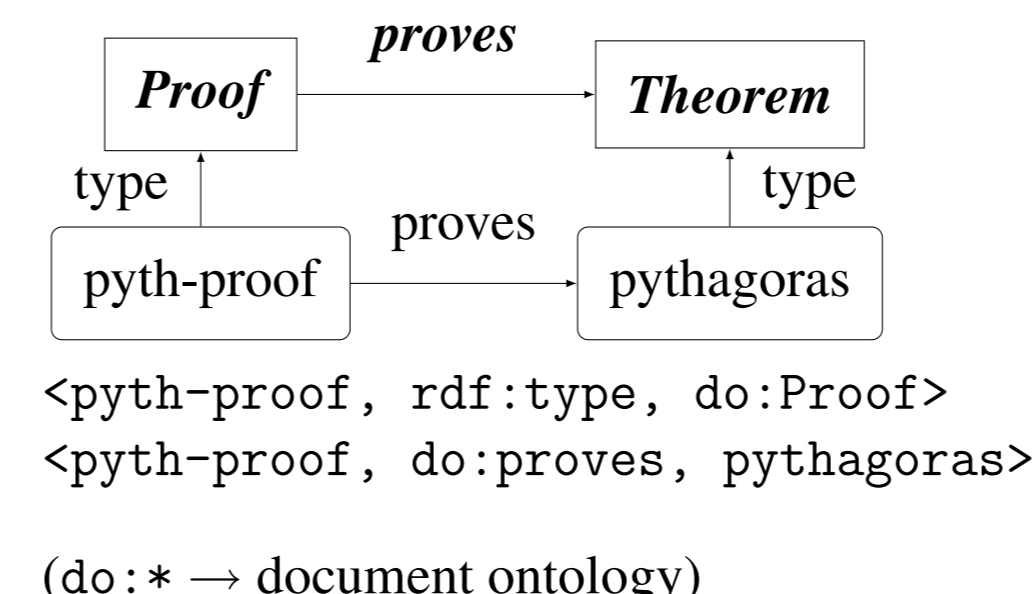
Try to query **Wikipedia** for

- $z = \sqrt{x^2 + y^2}$

- unproven theorems about triangles

- semantic web**: capture document structure (\approx structure of **communicated knowledge!**) in one **ontology** per scientific domain and general "upper ontology"
- Extract **RDF instances** of these ontologies from the markup \Rightarrow **scientific knowledge** that **practical applications** can reason on

```
<omdoc>
  <proof
    id="pyth-proof"
    for="pythagoras">
      ...
    </proof>
</omdoc>
```



(do:* \rightarrow document ontology)

State of the Art

Collaborative systems **Knowledge mgmt.**

- social interaction
- subscribing
- agility
- stepwise refinement
- structural semantic markup (in **science**)
- ontologies
- reasoning

Proposed synthesis: **semantic social** software (e. g. **wikis**). But how to utilise knowledge (mostly RDF) for **more** than just

- navigation links
- editing assistance
- semantic search?

Approach

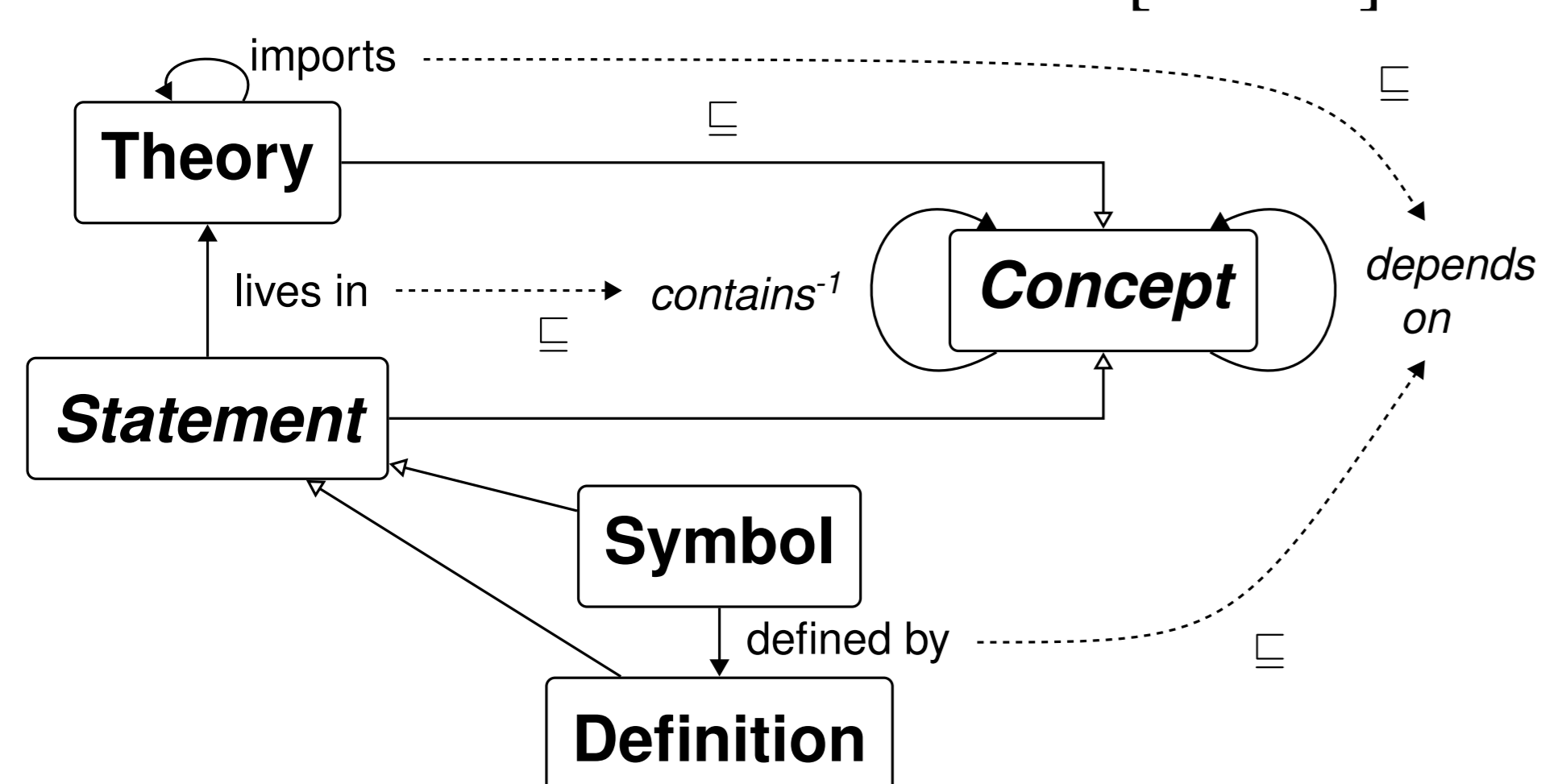
Integrate cross-domain **scientific knowledge** as **structural semantic markup (SSM)** into a **CWE**, starting with **mathematics** (covered and understood best so far; foundation of science).

1. Make users generate knowledge through **user-centered services**
2. Leverage this knowledge by an **ontology abstraction layer**
3. Build or integrate **productive/innovative/cool services** on top of that (GOTO 1)

Knowledge Representation

- Three-layered model of **scientific knowledge**:
 1. **objects** (symbols, equations, molecules)
 2. **statements** (axioms, hypotheses, experiments)
 3. **theories** (collections of related statements, give symbols context)

Realized for mathematics: OMDoc [Koh06]



From Users to Knowledge Workers Through Services

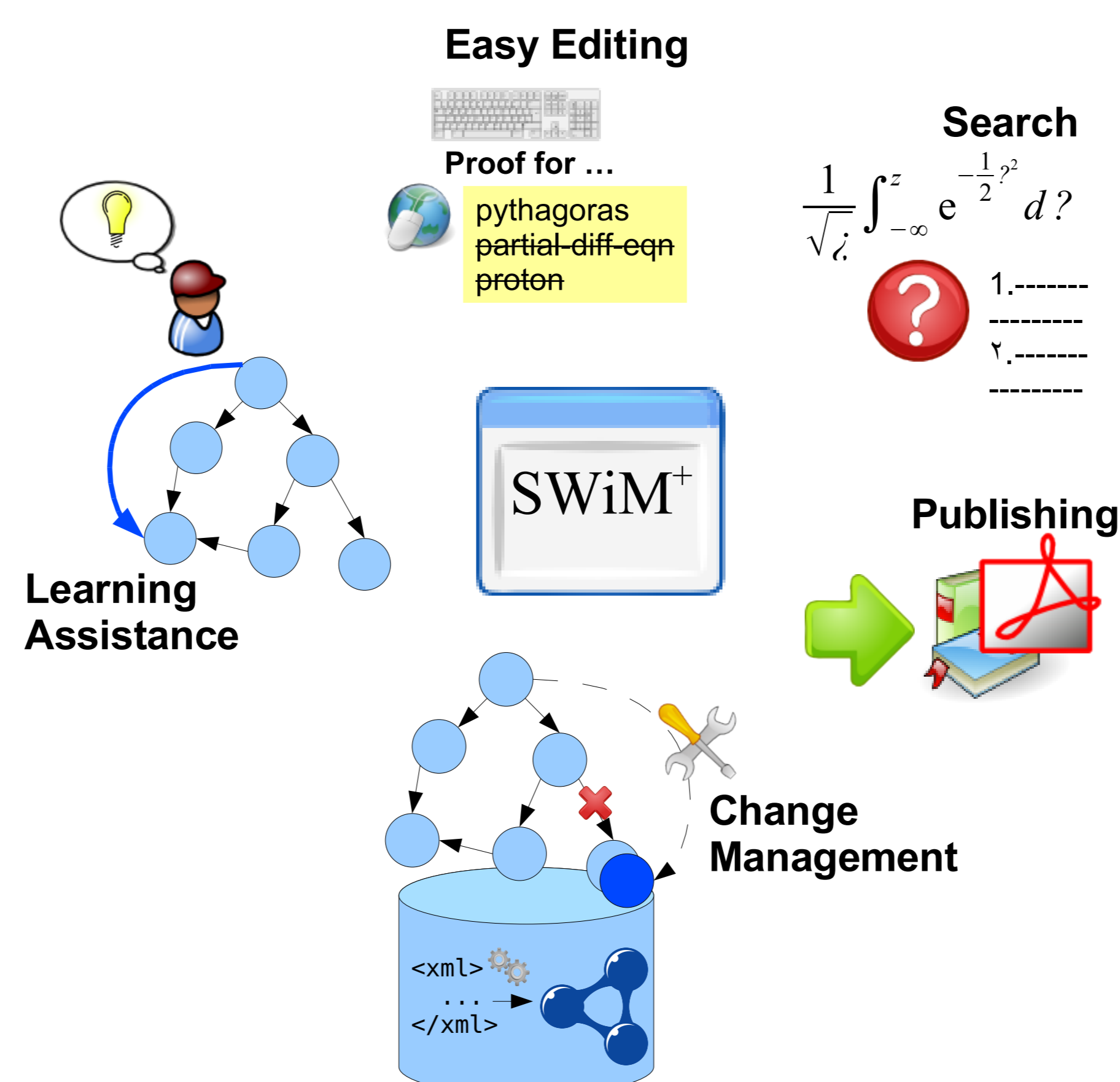
SSM inherently hard to edit \Rightarrow design services in an user-centered way to motivate users into action. Share benefits of services using semantic markup with the user who contributes [KohMül07]

Dimensions of services:

- generality**: domain-specific or not?
- abstractness** (from syntax)
- complexity**: usage of inferred knowledge?
- workflow**: single-/multi-step?
- implementation**: in the environment vs. under the hood (\updownarrow) vs. connection to external service (\leftrightarrow)

Offer an "operating system" for services:

- Access to RDF extract or full documents
- Ontologies for document structure and for the collaborative work environment
- Combined use case:
 1. Many **user questions** to a **mathematical proof**
 2. "Must be hard to understand!"
 3. **Auto-create examples** for special cases
- RESTful interface for **web services**



Easier editing:

- Auto-completion of link targets
- Simplified syntax or WYSIWYG
- Section-wise editing [Cnx06]

Learning assistant (explore transitive dependencies) \leftrightarrow Community-of-Practice analyser [Mül07]

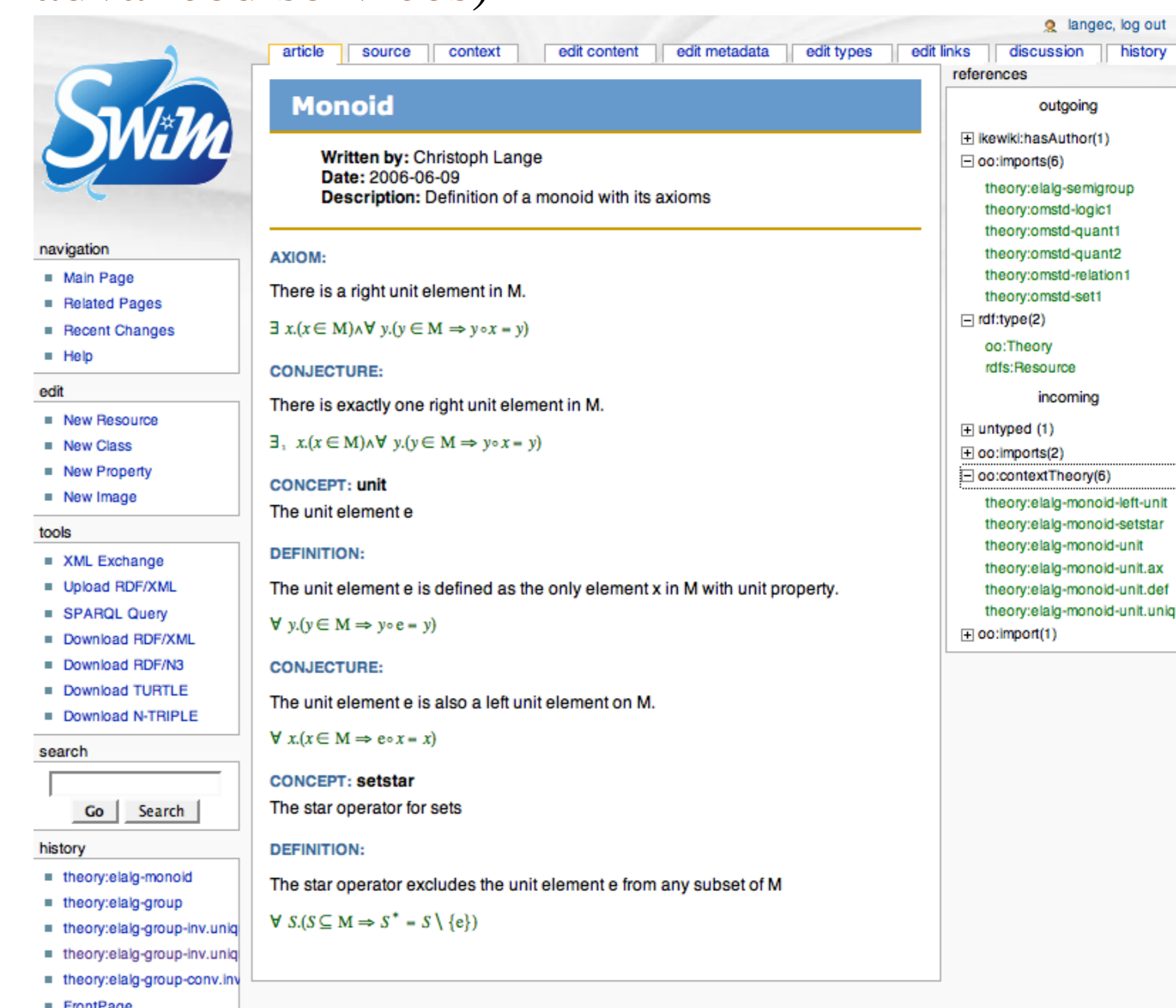
Maintaining dependencies \updownarrow backend extended by change management [Mül06]

Text/formula/molecule search **searchable object** \leftrightarrow **search engine** [KohSüc06]

Publishing: Compile knowledge collection, \leftrightarrow export to web or printer

SWiM, a Semantic Wiki

Technical base (from my diploma thesis): Semantic wiki for OMDoc (**mathematics only, no advanced services**)



- based on IkeWiki [Sch⁺06]; extended by:
 - OMDoc as alternative data format
 - rendering as XHTML+MathML
 - typed navigation links \cong RDF triples
 - OMDoc/XML to RDF extraction
- Try the online demo or install the prototype! Comments highly appreciated ☺

Work Plan

- Extend prototype
 - towards **other domains** (\Rightarrow formalise these domains in **ontologies!**)
 - towards a **service platform** (abstraction layer!)
- Specify, design, prototype, evaluate **services**
- Case studies:
 - Education**: introductory CS course (fall 2007)
 - Research**: scientists from various domains
- Feedback anticipated for
 - **service and user interface design** (also: **connecting users!**)
 - **ontology design**

If the **ontology abstraction layer** approach does facilitate the design and integration of **semantic services** that increase benefit and reduce users' investment, improving **other CWEs**, even in **non-scientific domains**, in a similar way will become possible.