OpenDreamKit Deliverable D4.2
Active/Structured Documents Requirements and existing Solutions

Michael Kohlhase

http://kwarc.info/kohlhase
Computer Science
Jacobs University Bremen, Germany

OpenDreamKit Workshop, Bremen, 28. June 2016

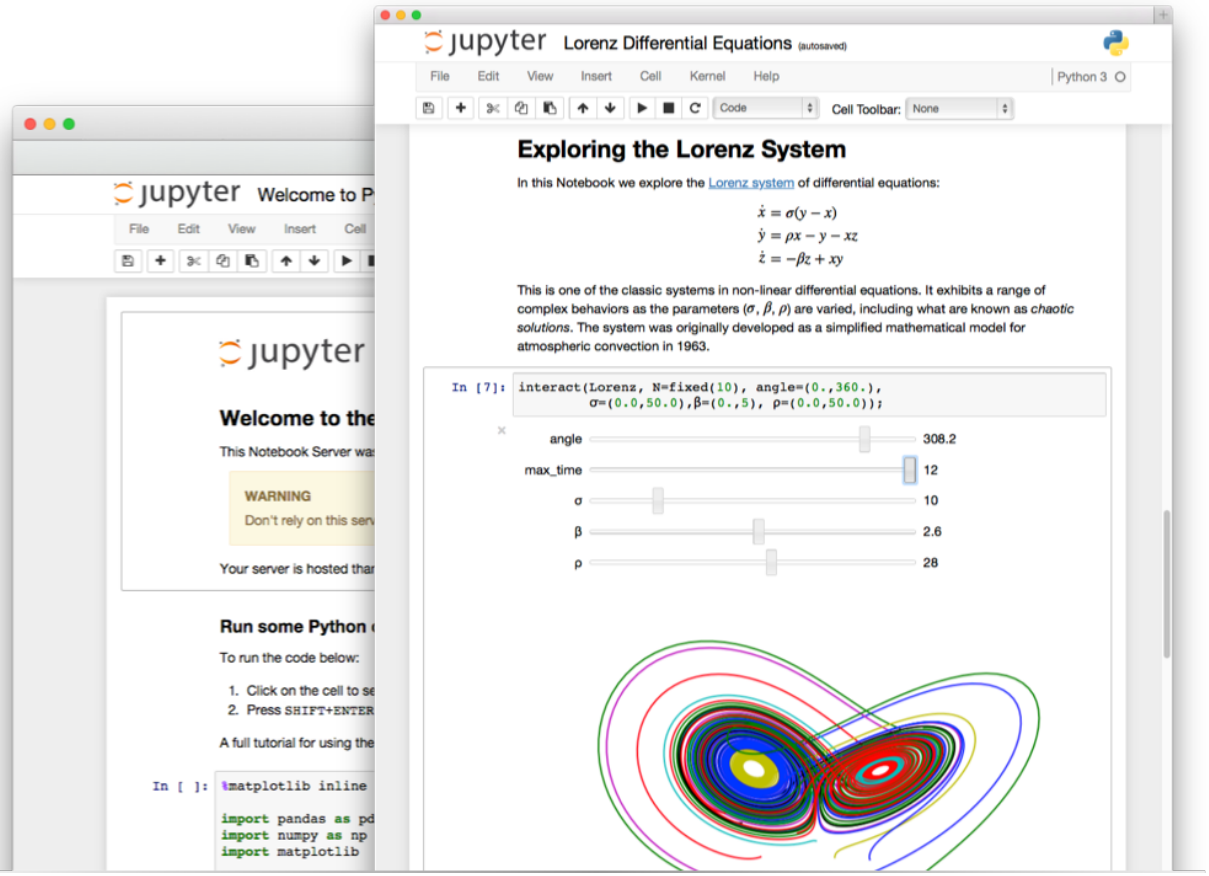
Background and Motivation

- Idea: OpenDreamKit aims to build a math VRE on top of an ecosystem of specialist symbolic software systems.
- Problem: The user wants to have a seamless experience when interacting with math
- Towards a Solution: supply a joint User Interface that integrates the system
- Existing Solutions: in OpenDreamKit
  - Project Jupyter: joint web-based notebook UI for command-line systems (macro-scale)
  - Active Documents: embedding semantic services into documents (micro-scale)

1 Existing Solutions

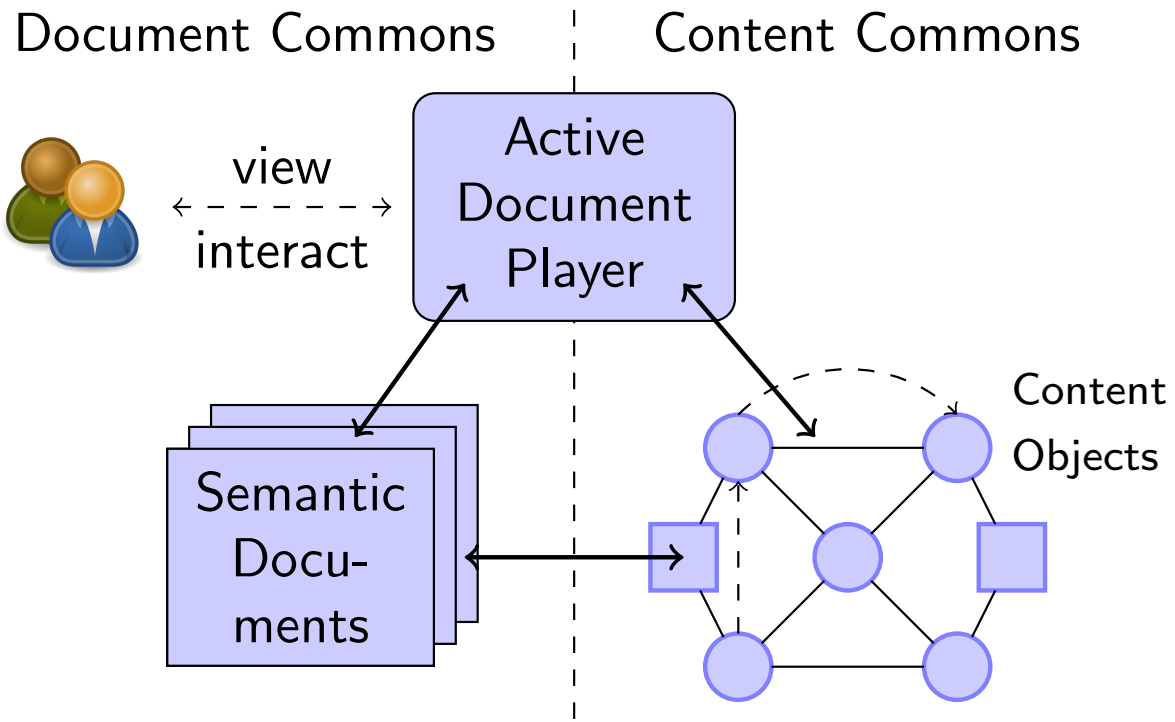
Project Jupyter in a Nutshell

- Idea: Notebooks: text + interaction cells as a UI metaphor
- Infrastructure: rational reconstruction if iPython (sound software engineering)
  - Communication layer: start/maintain various computational kernels
  - document format: web-based notebooks with interaction cells
  - widgets: make notebooks interactive in the small



The Active Documents Paradigm

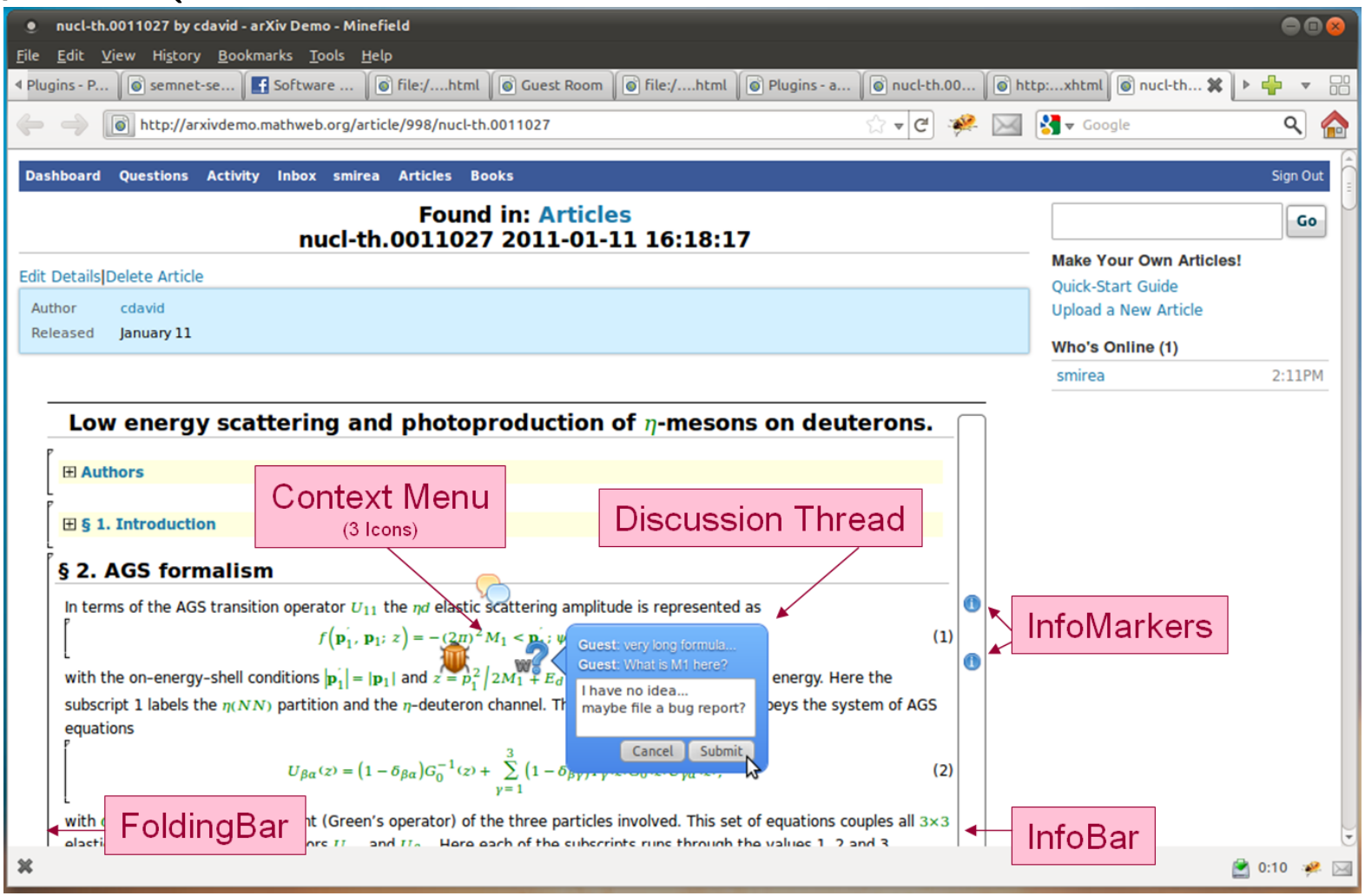
- **Definition 0.1** The active documents paradigm (ADP) consists of
  - *semantically annotated documents* together with
  - background ontologies (which we call the content commons),
  - *semantic services* that use this information
  - a document player application that embeds services to make documents executable.



- **Example 0.2** Services can be program (fragment) execution, computation, visualization, navigation, information aggregation and information retrieval

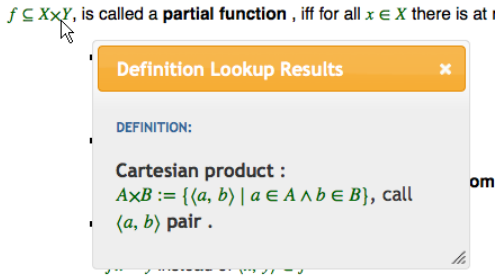
Planetary at the Presentation/Structural Level

- Planetary can make use objects and relations at various levels,
- **Example 0.3 (arXivdemo: Document Structure and Presentational Math)**

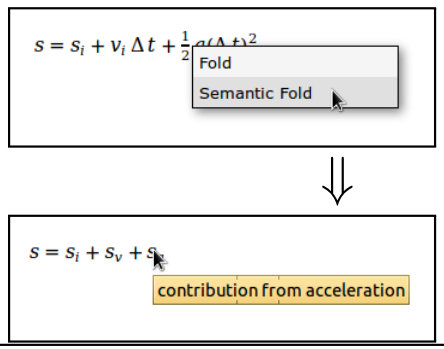


User Services at the Semantic Level in Planetary

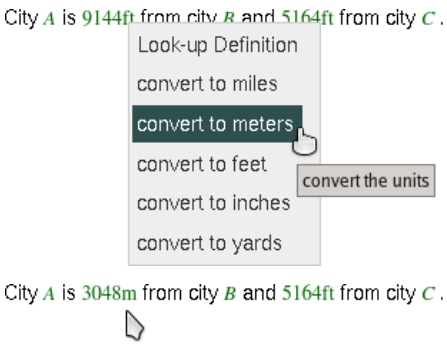
Definition Lookup



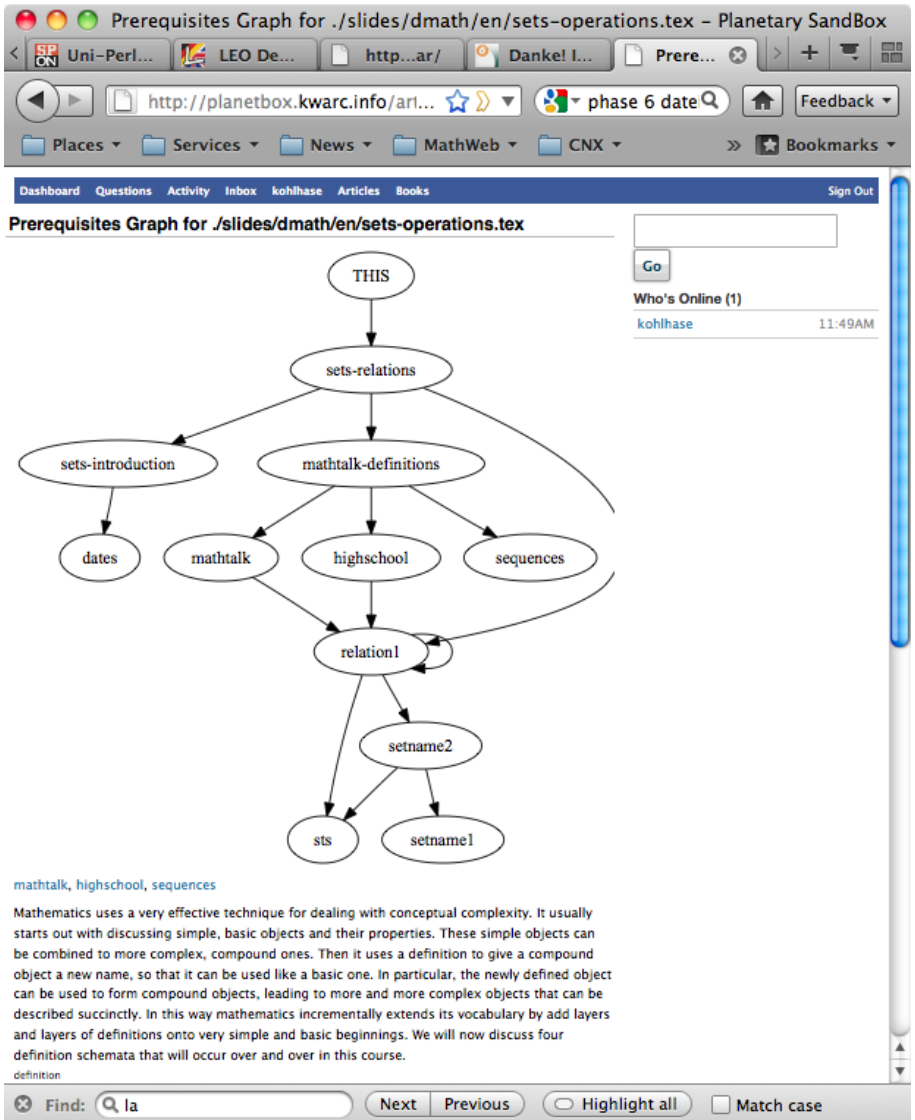
Semantic Folding



Unit Conversion



Prerequisites Navigation



User Services at the Formal Level in Planetary

- Formal Level: e.g. specification and verification in the LATIN Logic Atlas
  - flexible elision of brackets
  - argument reconstruction via external TWELF system
  - verification by the HETS system
- **Example 0.4** Adapting formal representations user preferences

document derived.omdoc

remote module FalsityExt

remote module NEGExt

theory IMPExt **meta** lf

include IMP

imp2I : $((\text{ded}\ A \to \text{ded}\ B \to \text{ded}\ C) \to \text{ded}\ A\ \text{imp}\ (B\ \text{imp}\ C))$

= $[f{:}\text{ded}\ A \to \text{ded}\ B \to \text{ded}\ C]\text{impI}\ ([p{:}\text{ded}\ A]\text{impI}\ ([q{:}\text{ded}\ B]f\ p\ q))$

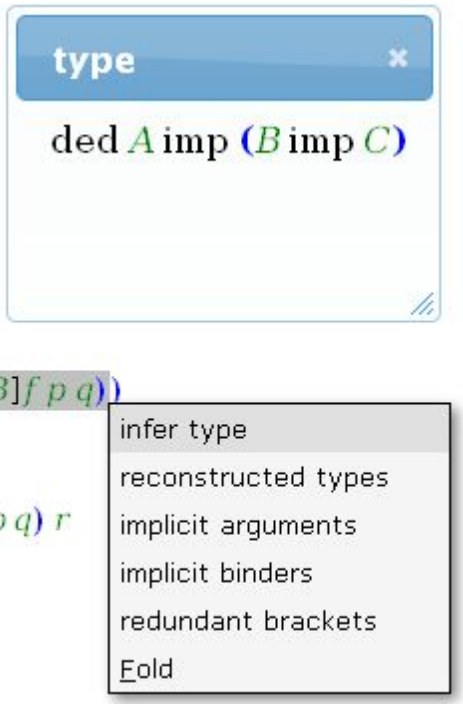
imp2E : $(\text{ded}\ A\ \text{imp}\ (B\ \text{imp}\ C) \to \text{ded}\ A \to \text{ded}\ B \to \text{ded}\ C)$

= $[p{:}\text{ded}\ A\ \text{imp}\ (B\ \text{imp}\ C)][q{:}\text{ded}\ A][r{:}\text{ded}\ B]\text{impE}\ (\text{impE}\ p\ q)\ r$

remote module CONJExt

remote module DISJExt

remote module Equiv

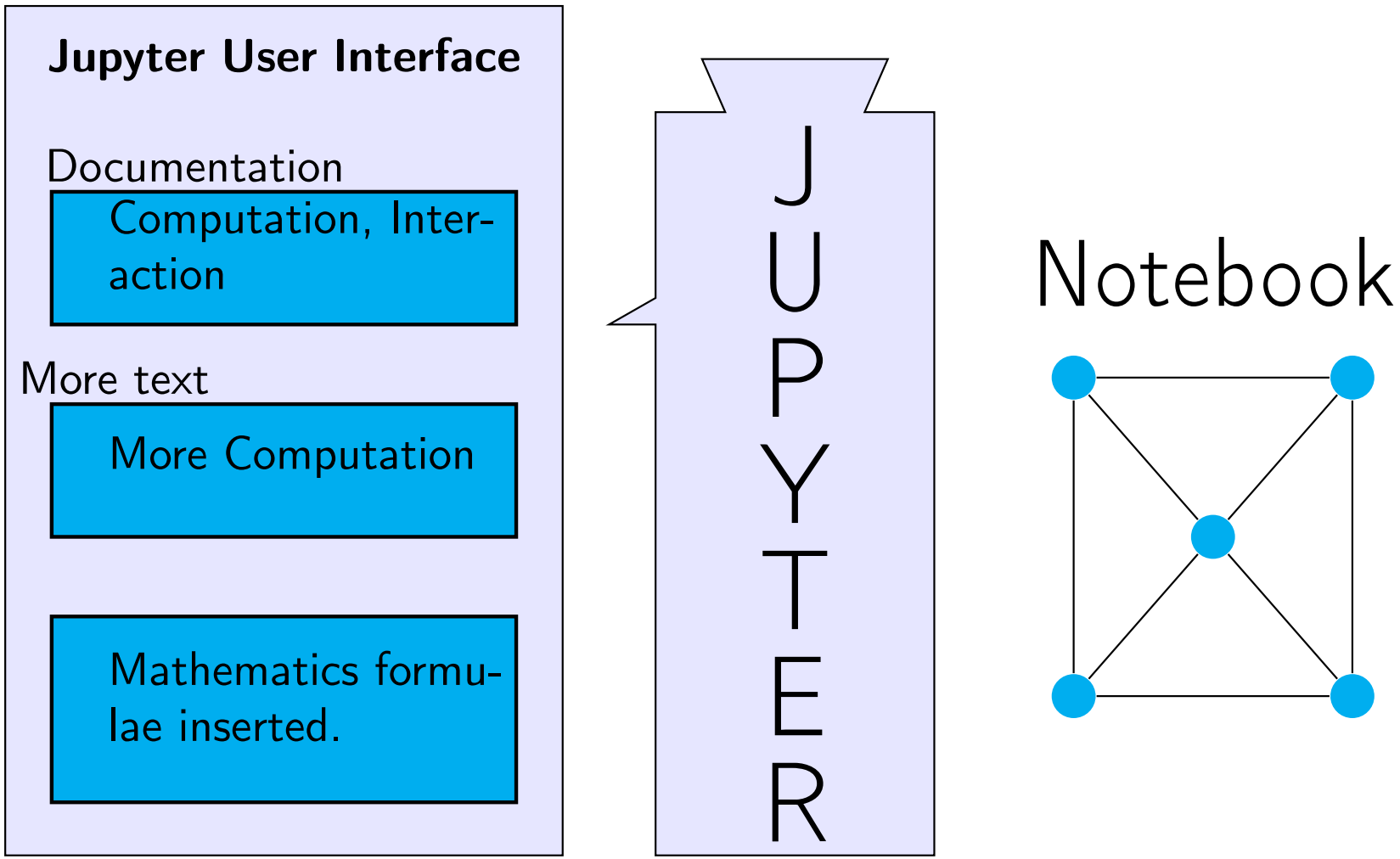


Active Documents in Action

- Informal Active Documents: Weierstrass Equation in ODK Glossary
- Informal Active Documents: Weierstrass Equation in ODK Glossary
- Formal Active Documents: propositional logic in MMT

2 Towards a math VRE via a joint UI

Jupyther: Information Processing Architecture



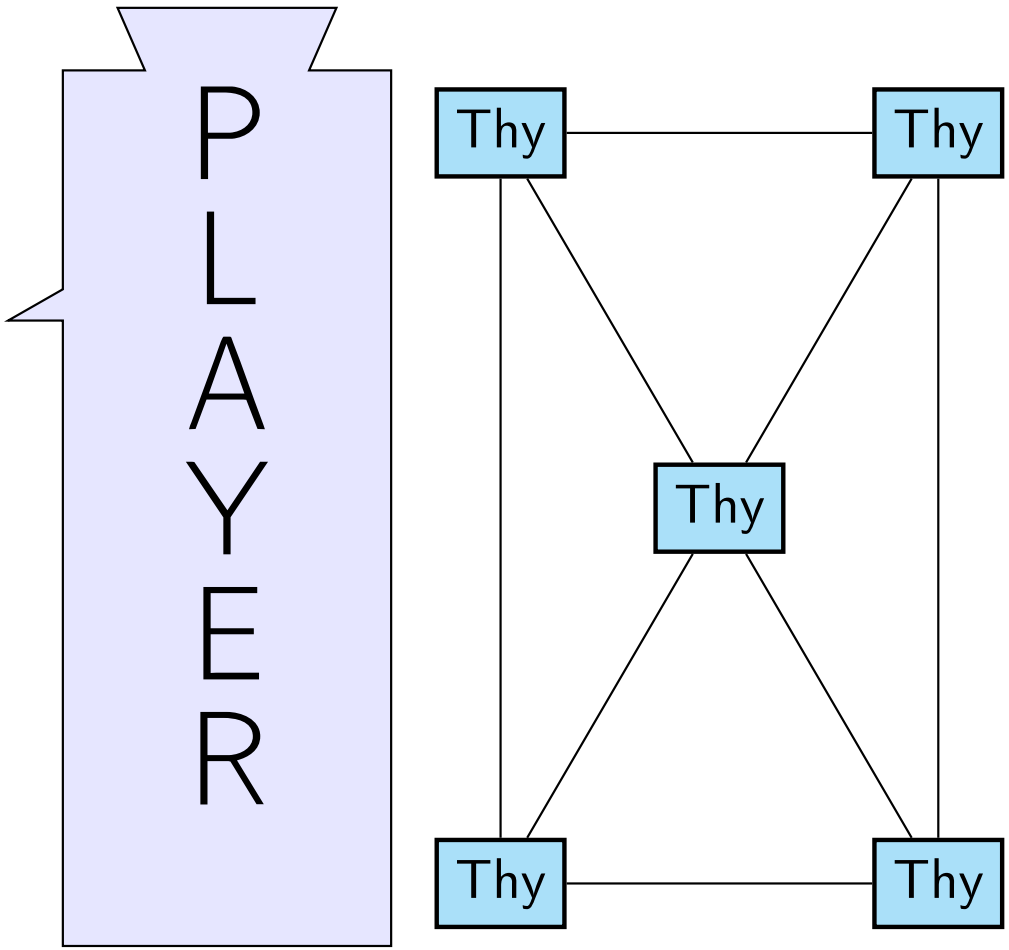
Active Documents: Information Processing Architecture

My Math Document

This is a sentence used to represent the interactivity between the text and the active document.

Among the text of an active document there are fields which can interact with the user such as:

Field 1 Field 2



Synthesis: A Document/Interaction-based UI for ODK

My Math Document

This is a sentence used to represent the interactivity between the text and the active document.

Interactivity.

Task 1
Task 2
Task 3
Task 4
Task 5

This is also interactive.

OpenDreamKit Notebook