Using Open Mathematical Documents to interface Computer Algebra and Proof Assistant systems

Jónathan Heras Vico Pascual Julio Rubio

Departamento de Matemáticas y Computación
Universidad de La Rioja
Spain

8th International Conference on Mathematical Knowledge Management

July 11, 2009

Table of Contents

1. Introduction
2. Specifying with OMDoc Documents
3. Conclusions and Further Work

Table of Contents

1 Introduction

2 Specifying with OMDoc Documents

3 Conclusions and Further Work

Introduction

- Kenzo:
  - Symbolic Computation System devoted to Algebraic Topology
  - Homology groups unreachable by any other means

Introduction

- Kenzo:
  - Symbolic Computation System devoted to Algebraic Topology
  - Homology groups unreachable by any other means
- Goal:
  - Increase the reliability of Kenzo
    - Integration of Kenzo with ACL2

Introduction

- Kenzo:
  - Symbolic Computation System devoted to Algebraic Topology
  - Homology groups unreachable by any other means
- Goal:
  - Increase the reliability of Kenzo
    - Integration of Kenzo with ACL2
- Necessary:
  - Computation
  - Representation of the Mathematical Knowledge
  - Deduction

Computation

- Kenzo + mediated access

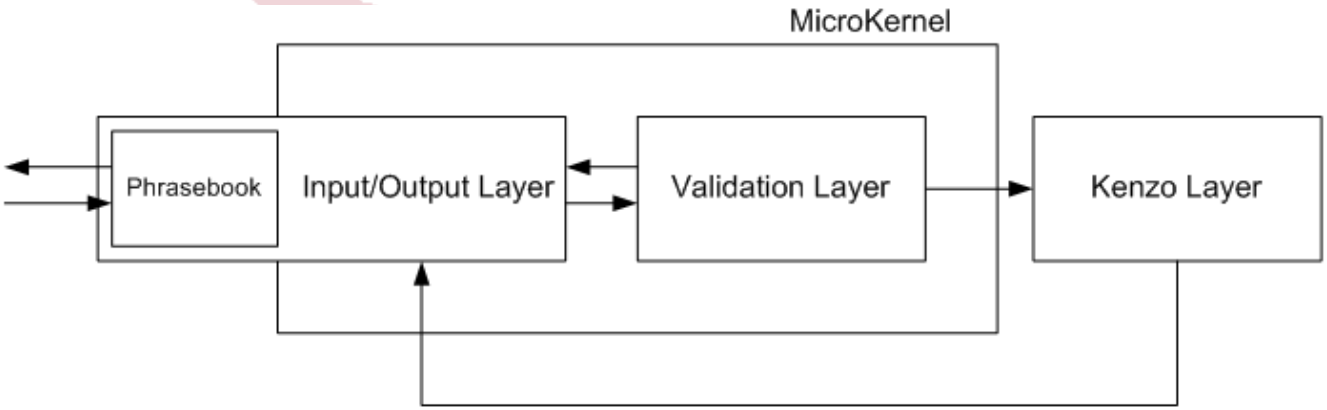


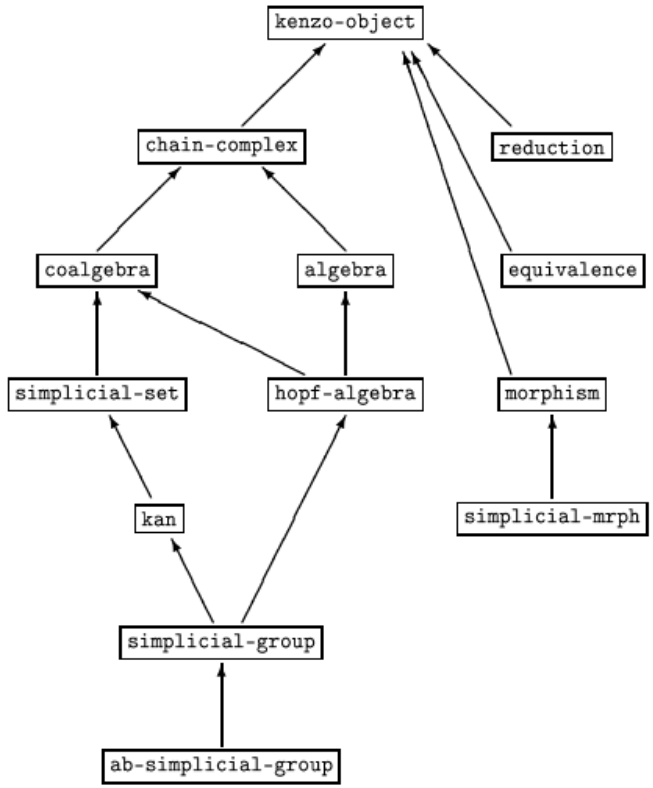
Figure: Simplified diagram of the architecture

Representation

- OpenMath: XML standard

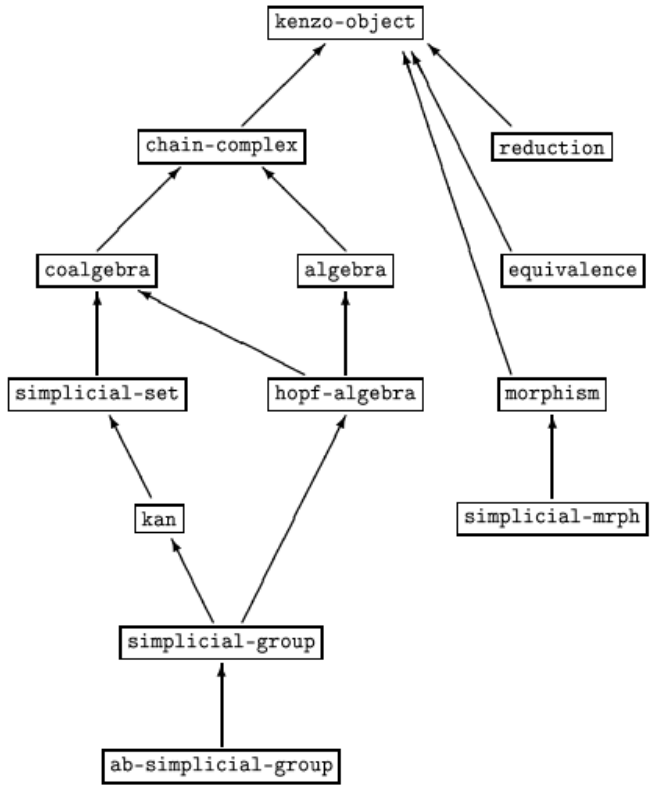
Representation

- OpenMath: XML standard
- Kenzo works with the main mathematical structures used in Simplicial Algebraic Topology



Representation

- OpenMath: XML standard
- Kenzo works with the main mathematical structures used in Simplicial Algebraic Topology



- A CD without axioms for each Mathematical Structure was developed

Deduction

- ACL2 (A Computational Logic for an Applicative Common Lisp)

Deduction

- ACL2 (A Computational Logic for an Applicative Common Lisp)
- ACL2
  - Programming Language
  - First-Order Logic
  - Theorem Prover

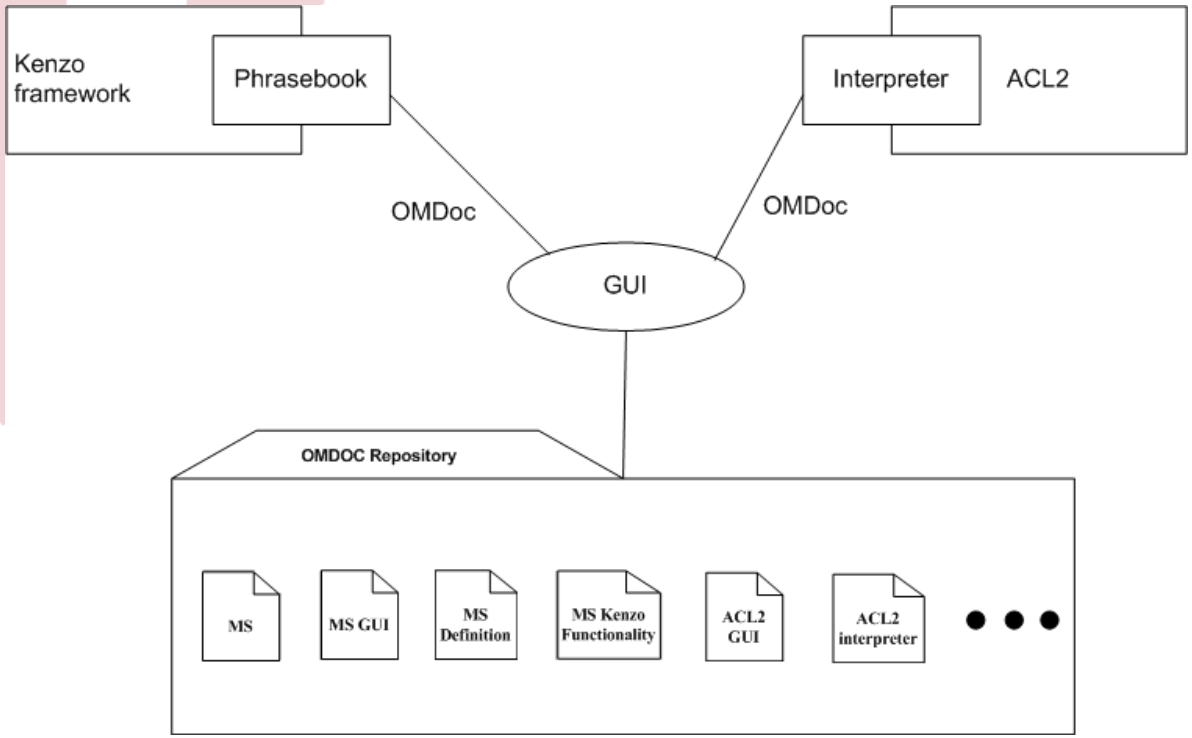
Deduction

- ACL2 (A Computational Logic for an Applicative Common Lisp)
- ACL2
  - Programming Language
  - First-Order Logic
  - Theorem Prover
- Proof techniques:
  - Simplification
  - Induction
  - *"The Method"*

Deduction

- ACL2 (A Computational Logic for an Applicative Common Lisp)
- ACL2
  - Programming Language
  - First-Order Logic
  - Theorem Prover
- Proof techniques:
  - Simplification
  - Induction
  - *"The Method"*
- Encapsulate: to the constrained introduction of new functions
  - Signatures
  - Properties
  - Witness

Gathering all the pieces



- A Graphical User Interface to gather all the pieces was developed
  - Customizable by means of an OMDoc Document Repository
  - OpenMath $\longrightarrow$ OMDoc due to OMDoc tools

Table of Contents

1 Introduction

2 Specifying with OMDoc Documents

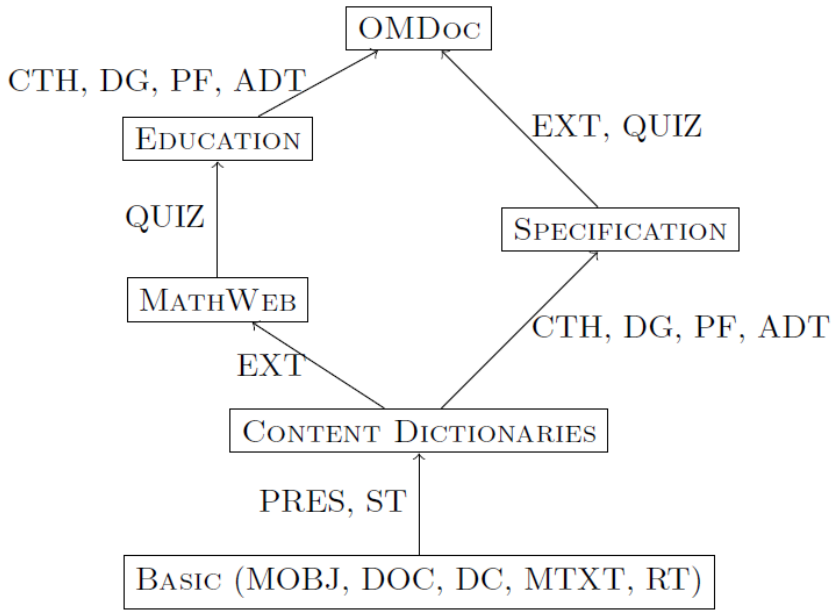
3 Conclusions and Further Work

OMDoc Documents

- OMDoc format:
  - mathematical documents + knowledge encapsulate in them
  - three levels of information:
    - formulæ
    - mathematical statements
    - mathematical theories

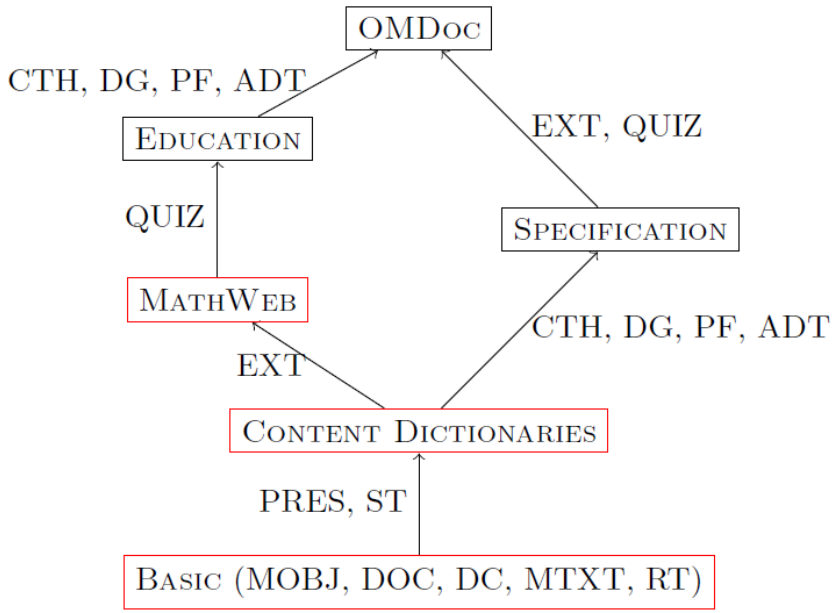
OMDoc Documents

- OMDoc format:
  - mathematical documents + knowledge encapsulate in them
  - three levels of information:
    - formulæ
    - mathematical statements
    - mathematical theories
- Sub-languages:



OMDoc Documents

- OMDoc format:
  - mathematical documents + knowledge encapsulate in them
  - three levels of information:
    - formulæ
    - mathematical statements
    - mathematical theories
- Sub-languages:



OMDoc Documents

- OMDoc format:
  - mathematical documents + knowledge encapsulate in them
  - three levels of information:
    - formulæ
    - mathematical statements
    - mathematical theories
- Sub-languages:
- 5 kinds of OMDoc documents:
  - Definition of Mathematical Structures
  - Logic to interact with Kenzo
  - Presentation for the GUI
  - ACL2 interpreter
  - ACL2 presentation for the GUI

Definition of Mathematical Structures

- Goal:
  - Define the mathematical structures of Kenzo

Definition of Mathematical Structures

- Goal:
  - Define the mathematical structures of Kenzo
- Sub-language:
  - OMDoc Content Dictionaries Sub-language

Definition of Mathematical Structures

- Goal:
  - Define the mathematical structures of Kenzo
- Sub-language:
  - OMDoc Content Dictionaries Sub-language
- OMDoc Content Dictionaries for each mathematical structure:
  - Signature
  - Properties
  - Example

Definition of Mathematical Structures

- Goal:
  - Define the mathematical structures of Kenzo
- Sub-language:
  - OMDoc Content Dictionaries Sub-language
- OMDoc Content Dictionaries for each mathematical structure:
  - Signature
  - Properties
  - Example
- OpenMath CDs $\subset$ OMDoc CDs

Logic to interact with Kenzo

- Kenzo keeps on growing
- Goal:
  - Include new functionality
  - Extensibility of the system

Logic to interact with Kenzo

- Kenzo keeps on growing
- Goal:
  - Include new functionality
  - Extensibility of the system
- Sub-language:
  - MathWeb sub-language (EXT module)
    - <code> tag
    - Common Lisp code
    - code for different applications

Logic to interact with Kenzo

- Kenzo keeps on growing
- Goal:
  - Include new functionality
  - Extensibility of the system
- Sub-language:
  - MathWeb sub-language (EXT module)
    - <code> tag
    - Common Lisp code
    - code for different applications
- Functionality for each mathematical structure defined

Presentation for the GUI

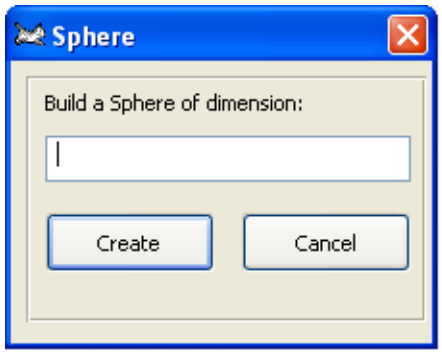
- Goal:
  - Define the structure of our GUI

Presentation for the GUI

- Goal:
  - Define the structure of our GUI
- Sub-language:
  - MathWeb sub-language
    - `<OMForeign>` tag
    - XUL

Presentation for the GUI

- Goal:
  - Define the structure of our GUI
- Sub-language:
  - MathWeb sub-language
    - `<OMForeign>` tag
    - XUL
- XUL:
  - Mozilla's XML-based user interface language
  - To build feature rich cross-platforms




```
<OMForeign>
  <window name="Sphere">
    <groupbox>
      <label value="Build a Sphere of dimension:"/>
      <textbox id="n" type="number" min="1" max="14"/>
      <hbox>
        <button label="Create" onclick="create-sphere-on-click"/>
        <button label="Cancel" onclick="cancel-sphere-on-click"/>
      </hbox>
    </groupbox>
  </window>
</OMForeign>
```

ACL2 Interpreter

- Goal:
  - Integrate Kenzo with ACL2
  - By means of OMDoc CDs

ACL2 Interpreter

- Goal:
  - Integrate Kenzo with ACL2
  - By means of OMDoc CDs
- Sub-language:
  - MathWeb sub-language (EXT module)
    - <code> tag
    - Interpreter from OMDoc CDs to ACL2 encapsulates

ACL2 Interpreter

- Goal:
  - Integrate Kenzo with ACL2
  - By means of OMDoc CDs
- Sub-language:
  - MathWeb sub-language (EXT module)
    - <code> tag
    - Interpreter from OMDoc CDs to ACL2 encapsulates

OMDoc CDs		ACL2 Encapsulates
Signatures	$\longrightarrow$	*Signatures*
Properties	$\longrightarrow$	*Properties*
Examples	$\longrightarrow$	*Witness*

ACL2 presentation for the GUI

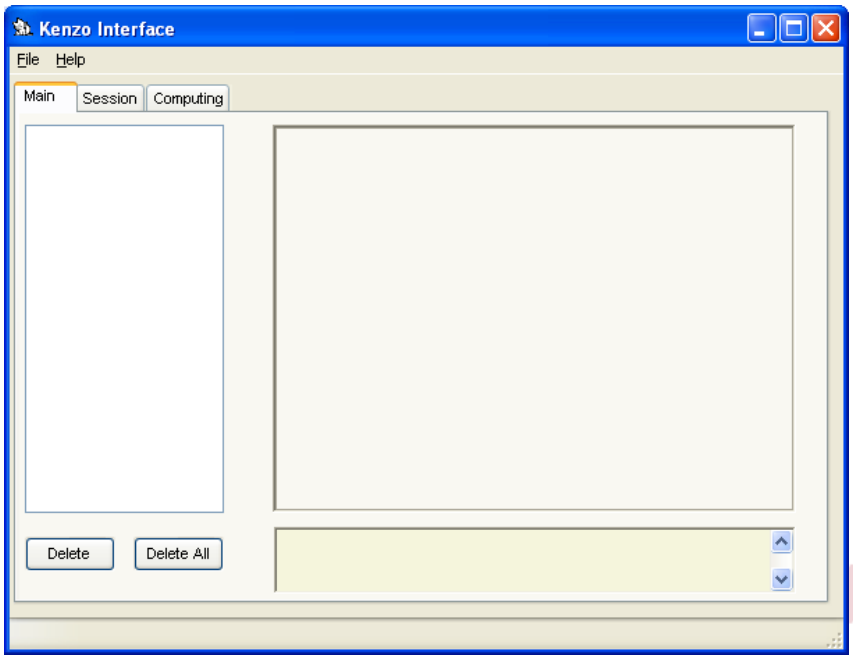
- Goal:
  - Integrate the GUI with ACL2

ACL2 presentation for the GUI

- Goal:
  - Integrate the GUI with ACL2
- Sub-language:
  - MathWeb sub-language
    - `<OMForeign>` tag
    - XUL

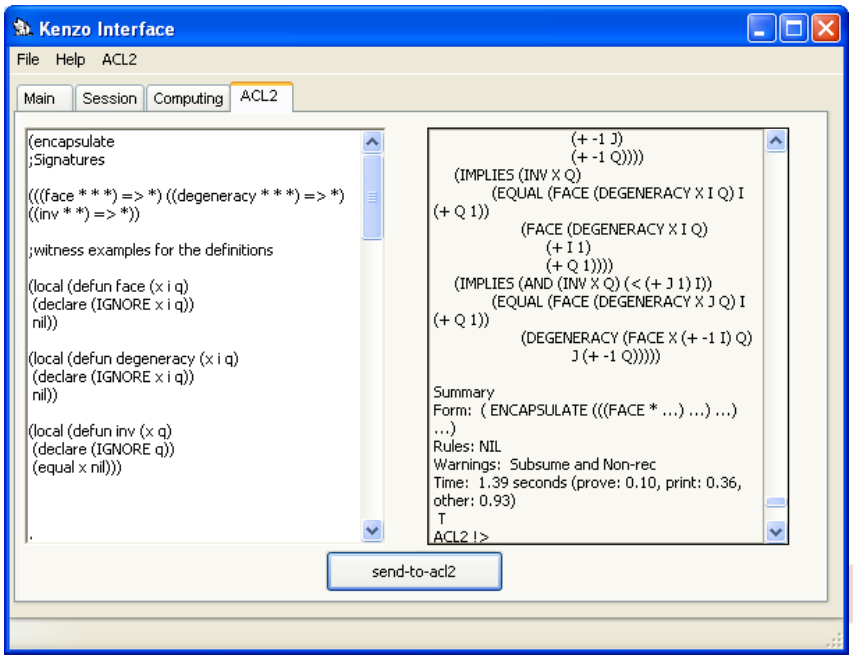
ACL2 presentation for the GUI

- Goal:
  - Integrate the GUI with ACL2
- Sub-language:
  - MathWeb sub-language
    - `<OMForeign>` tag
    - XUL



ACL2 presentation for the GUI

- Goal:
  - Integrate the GUI with ACL2
- Sub-language:
  - MathWeb sub-language
    - `<OMForeign>` tag
    - XUL



Gathering all the pieces

- Goal:
  - Glue all the parts of a mathematical structure

Gathering all the pieces

- Goal:
  - Glue all the parts of a mathematical structure
- Sub-language:
  - Basic OMDoc
    - `<omgroup>` tag

Gathering all the pieces

- Goal:
  - Glue all the parts of a mathematical structure
- Sub-language:
  - Basic OMDoc
    - `<omgroup>` tag

```
<omdoc id="simplicial-sets.omdoc">
    ...
    <omgroup type="sequence">
        <ref xref="simplicial-sets-logic"/>
        <ref xref="simplicial-sets-presentation"/>
        <ref xref="simplicial-sets-conceptual-model"/>
    </omgroup>
    ...
</omdoc>
```

Workflow

Loading Simplicial Sets and ACL2 in our GUI:

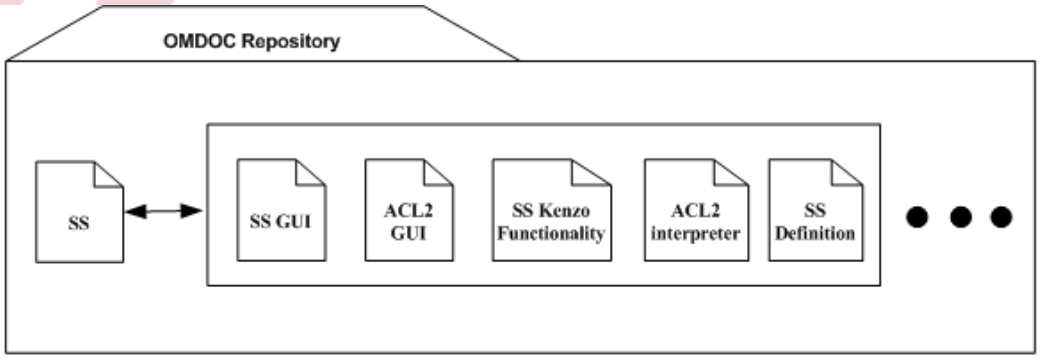


Figure: Workflow diagram

Workflow

Loading Simplicial Sets and ACL2 in our GUI:

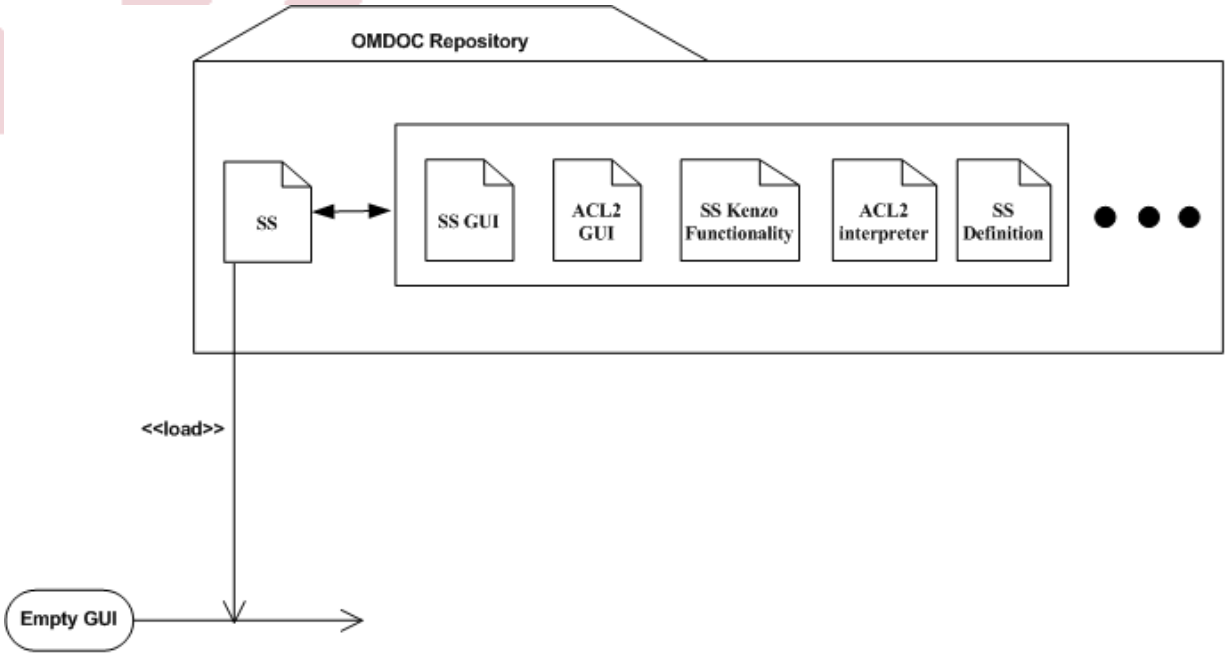


Figure: Workflow diagram

Workflow

Loading Simplicial Sets and ACL2 in our GUI:

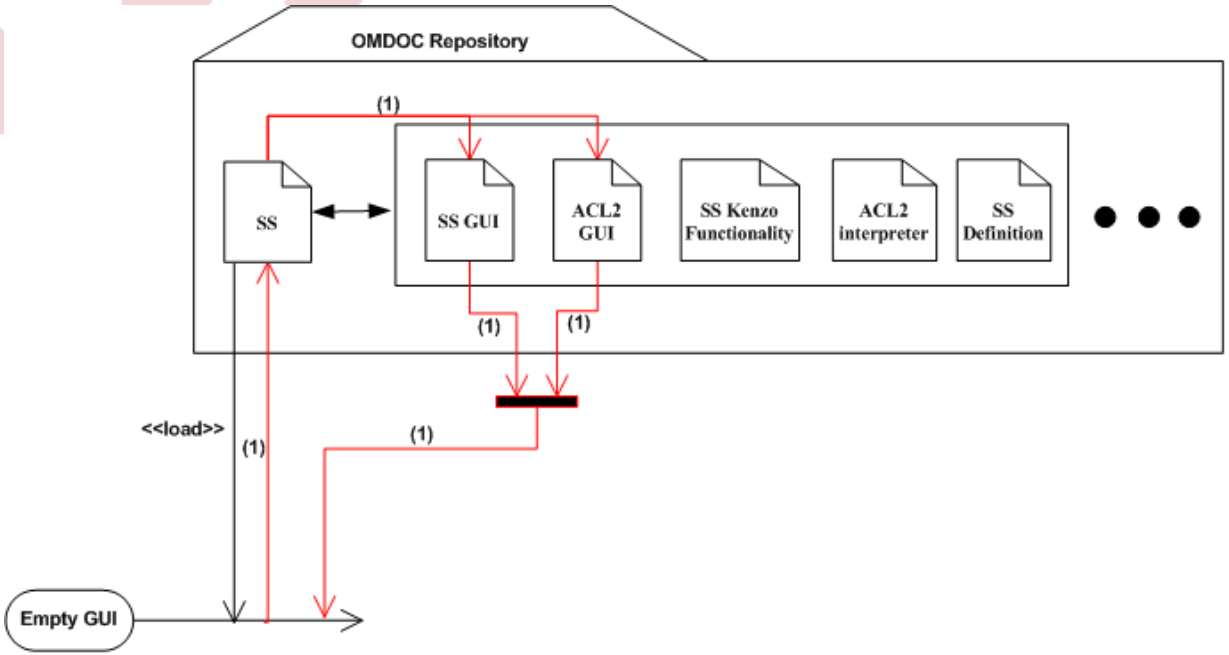


Figure: Workflow diagram

Workflow

Loading Simplicial Sets and ACL2 in our GUI:

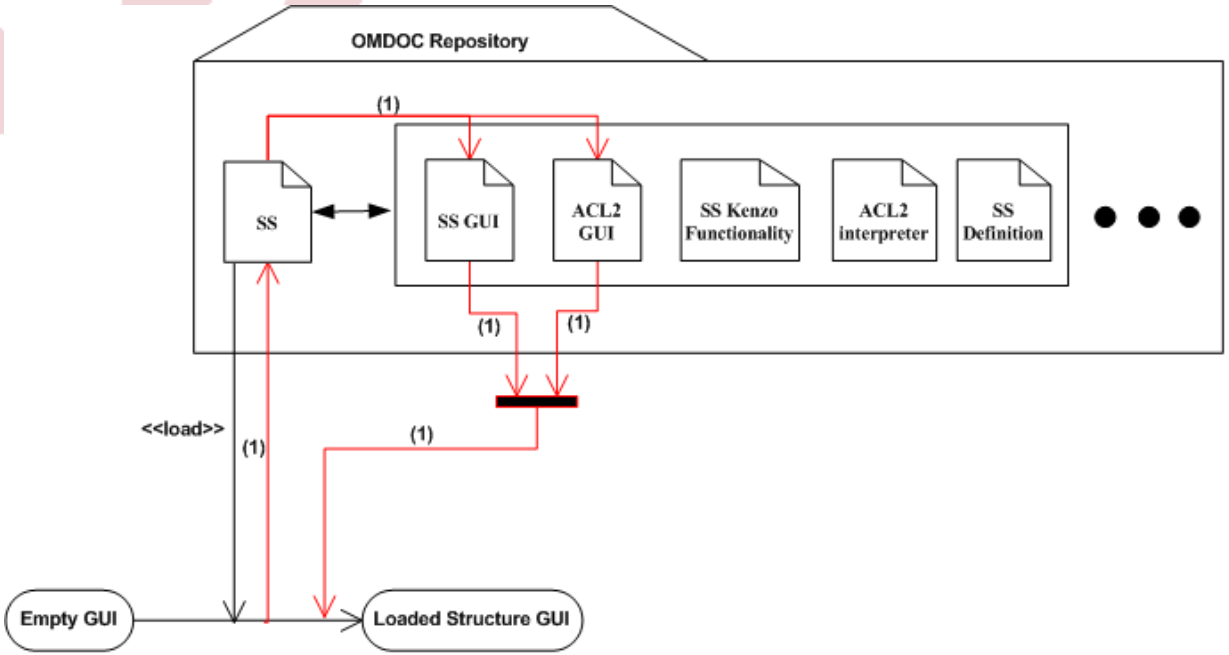


Figure: Workflow diagram

Workflow

Loading Simplicial Sets and ACL2 in our GUI:

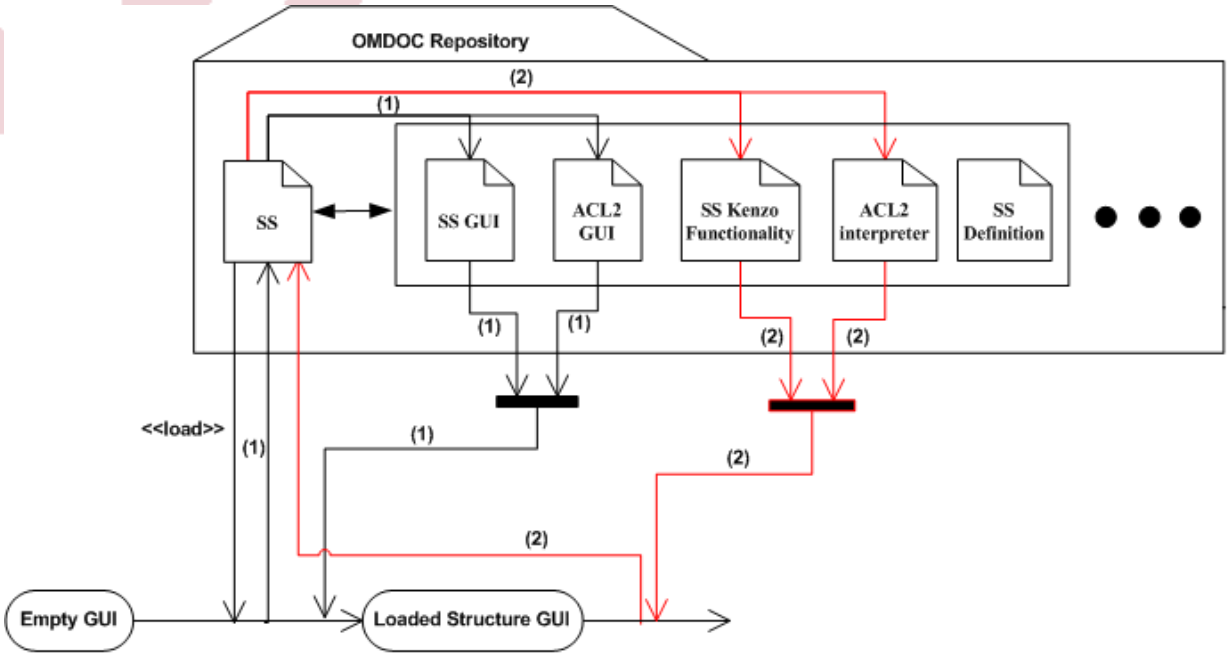


Figure: Workflow diagram

Workflow

Loading Simplicial Sets and ACL2 in our GUI:

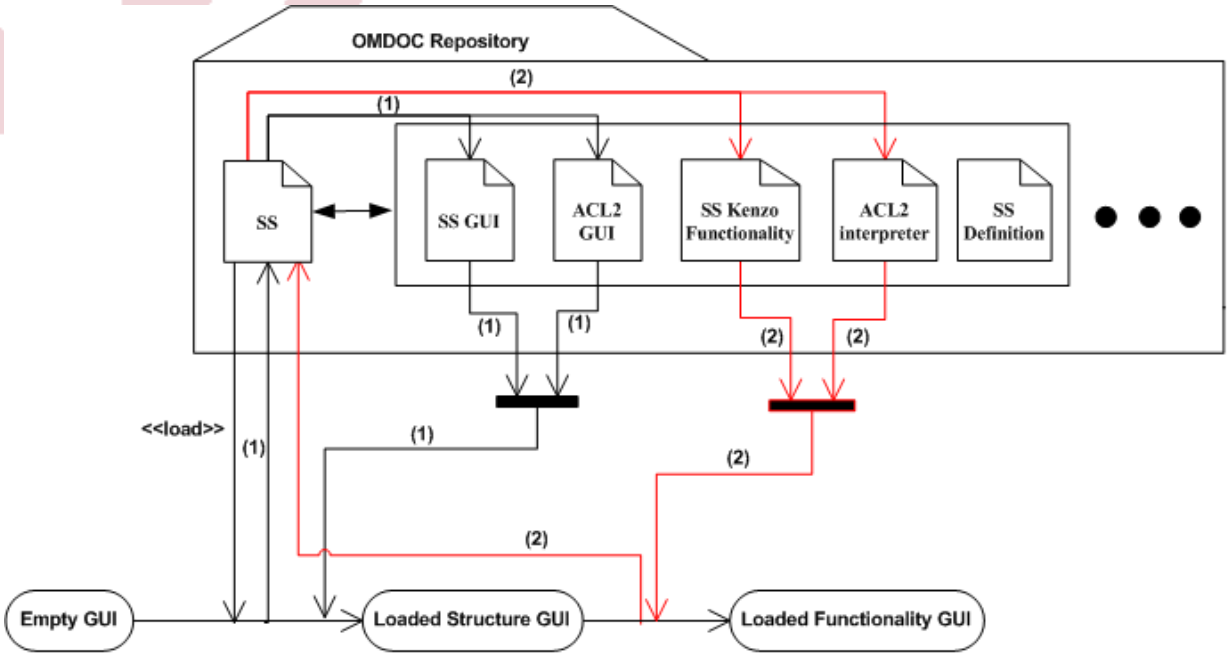


Figure: Workflow diagram

Workflow

Loading Simplicial Sets and ACL2 in our GUI:

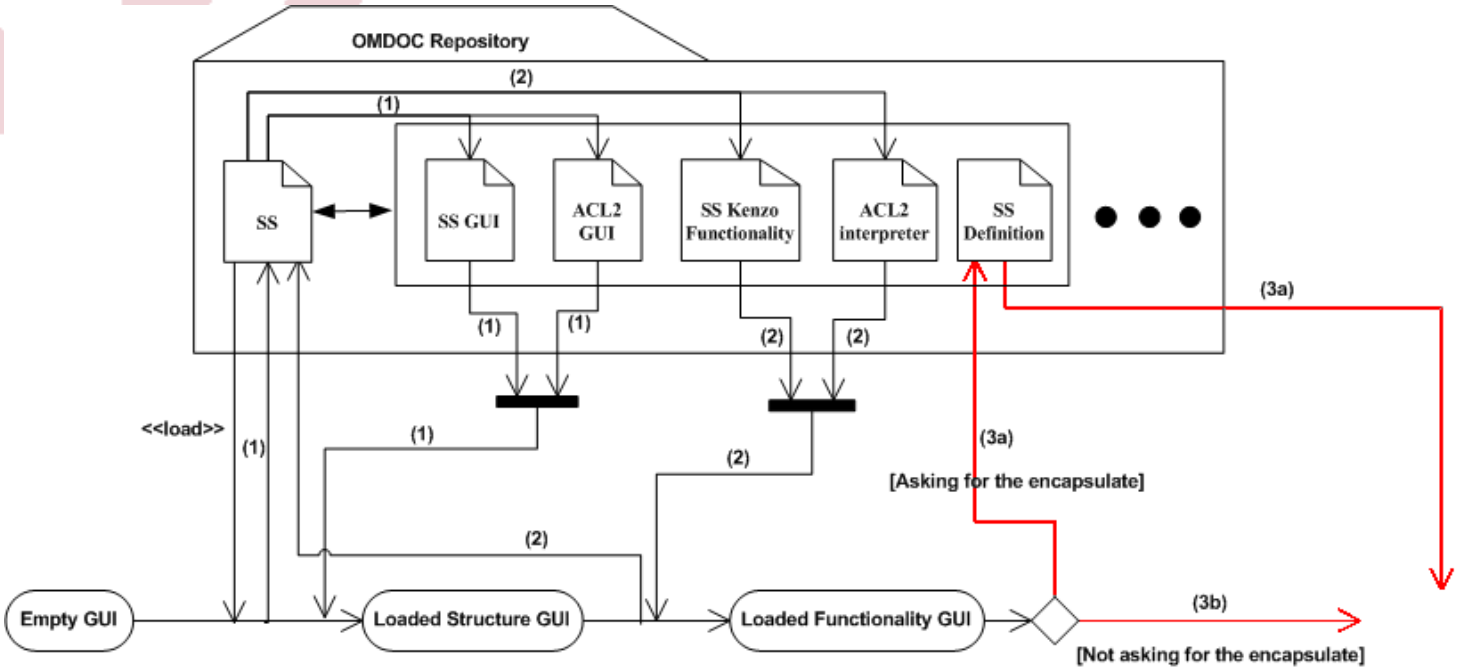


Figure: Workflow diagram

Workflow

Loading Simplicial Sets and ACL2 in our GUI:

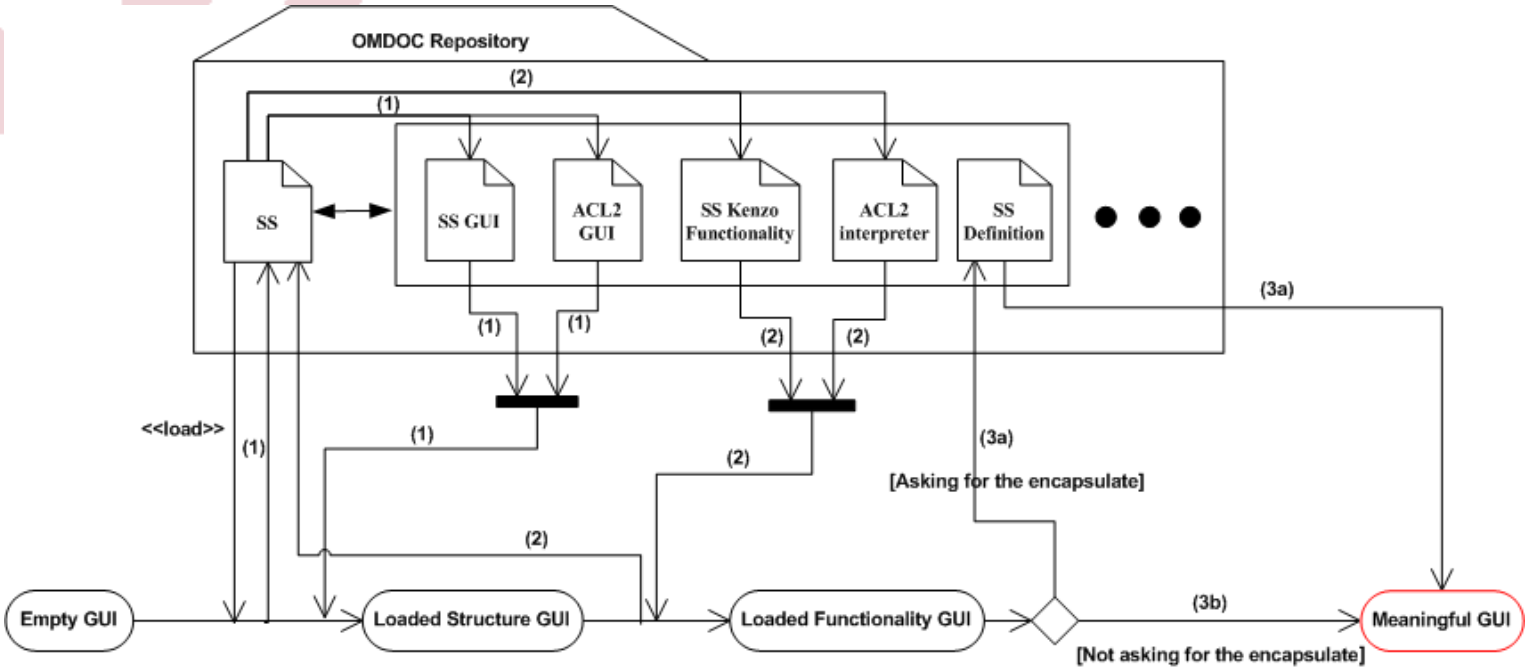


Figure: Workflow diagram

Table of Contents

1 Introduction

2 Specifying with OMDoc Documents

3 Conclusions and Further Work

Conclusions

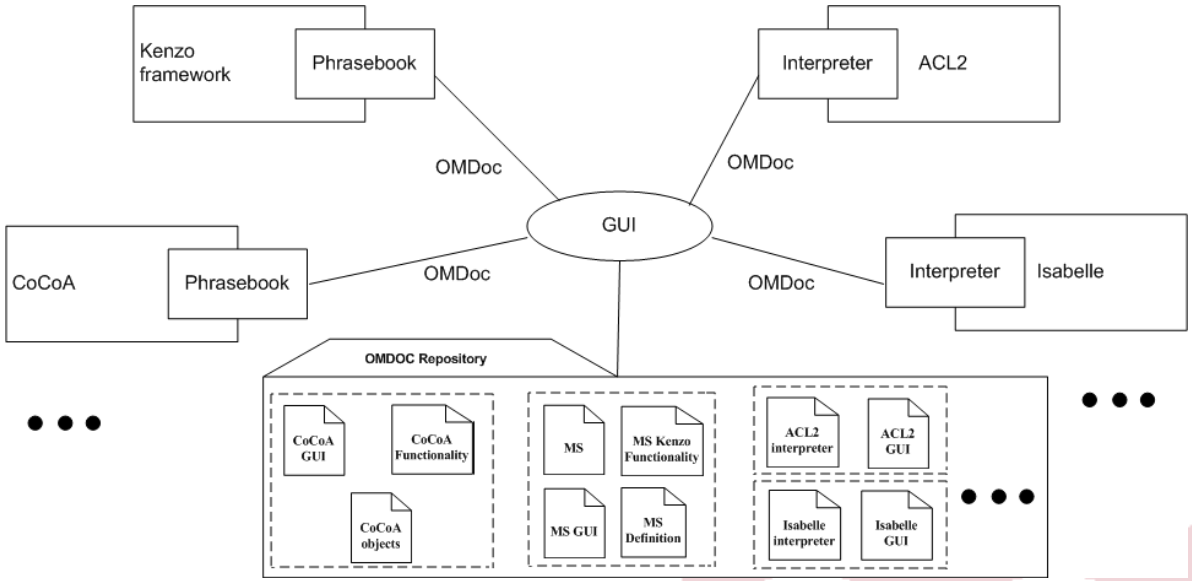
- Conclusions:
  - An OMDoc Documents Repository has been developed
  - Same OMDoc sub-language to reach different goals
  - Integration of representation, computation and deduction in the same system

Further Work

- Conclusions:
  - An OMDoc Documents Repository has been developed
  - Same OMDoc sub-language to reach different goals
  - Integration of representation, computation and deduction in the same system
- Further Work:
  - Implement interesting interactions between ACL2 and Kenzo
  - Integration with other Symbolic Computation Systems

Further Work

- Conclusions:
  - An OMDoc Documents Repository has been developed
  - Same OMDoc sub-language to reach different goals
  - Integration of representation, computation and deduction in the same system
- Further Work:
  - Implement interesting interactions between ACL2 and Kenzo
  - Integration with other Symbolic Computation Systems



Using Open Mathematical Documents to interface Computer Algebra and Proof Assistant systems

Jónathan Heras Vico Pascual Julio Rubio

Departamento de Matemáticas y Computación
Universidad de La Rioja
Spain

8th International Conference on Mathematical Knowledge Management

July 11, 2009