Interactive Tutoring Module for High-school Geometry

Dual Degree Project

Jayanth Tadinada
06D05016

Motivation

- Advantages of learning from a computer
  - Learn at his own pace and convenience
  - Focus on the specific topics after school hours
  - Interactive and interesting
  - Automatic evaluation and instant feedback

Motivation

- Computers as genuine teaching tools rather than mere learning aids.

- Students learn 3 times faster in a one to one setting

- Existing Systems
  - Objective type questions
  - Not suitable for all topics (e.g. Proof type problems)

Mindspark

- Adaptive self-learning program for school students
- Learn by answering progressively difficult questions
- Interactive, live feedback and adaptive logic
- Addresses misconceptions through visual or animated explanations

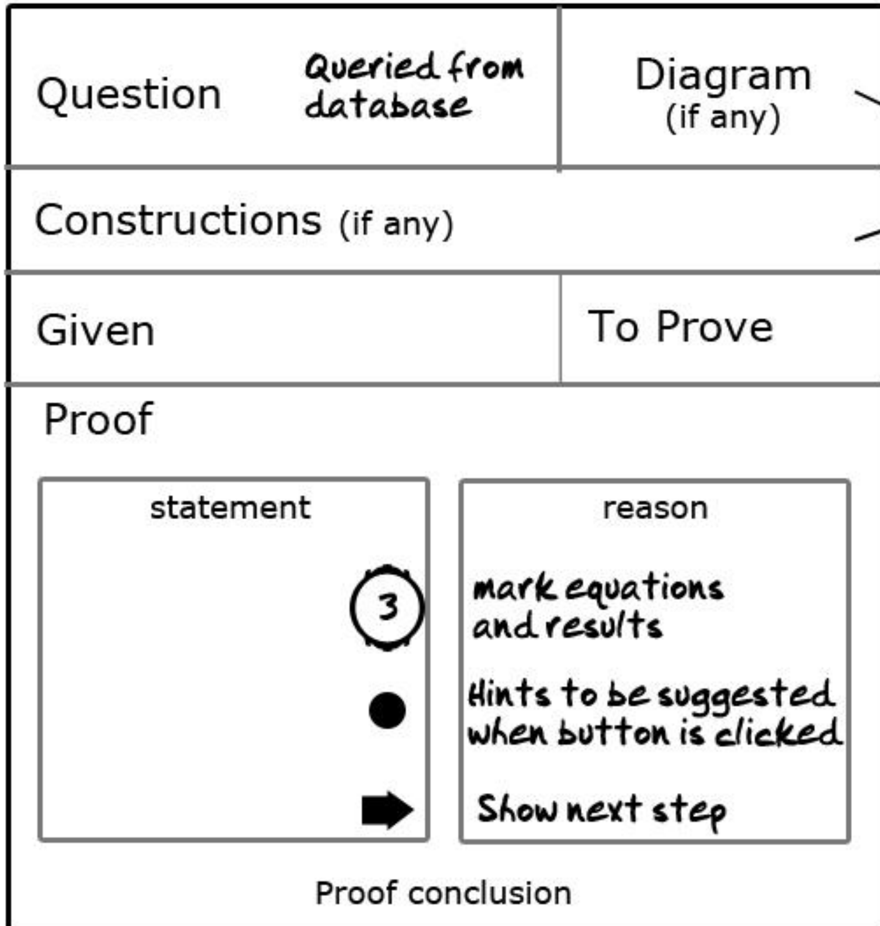
Problem Statement

- Design and build an interactive proof module

Scope

- Restricted to high school geometry
  - Properties of Triangles – congruency, similarity etc.

Functional Requirements



Other Requirements:

* Child friendly UI for entering mathematical symbols, operations etc

* Auto-completion of reasons, statements etc.

* Easy-to-use back end interface for entering questions, models solutions

* seamless integration into their existing system

Existing Systems

- Mindspark's existing geometry proof module
- Carnegie Learning's Cognitive Tutor
- Other Commercial Software Packages

MINDSPARK'S PROOF MODULE

34. Complete the proof of the converse of the Pythagoras theorem shown below.

This converse of the Pythagoras theorem states that if the square on one side of a triangle equals the sum of the squares on the other two sides, then the angle opposite to the third side is a right angle.

We prove this by contradiction.

Assumption: Let there be a triangle $\triangle PQR$ such that $PQ^2 + QR^2$ ____ PR^2 but ____.

Now construct another triangle $\triangle XYZ$ such that ____, ____ and ____ ----- ①

By applying Pythagoras theorem to $\triangle XYZ$, we get ____ --------- ②

From ① and ②, ____ follows.

By ____ congruence, $\triangle PQR \cong \triangle XYZ$ ⇨ ____

This contradicts the assumptions made. Hence Proved.

$\angle Q = 90^0$	$\angle Y = 90^0$	$\angle Q \neq 90^0$	$\angle Y \neq 90^0$	$XZ = PR$	$XY = PQ$	$YZ = QR$	
$XZ^2 = XY^2 + YZ^2$	$XZ^2 \neq XY^2 + YZ^2$	$=$	$\neq$	SAS	SSS	RHS	ASA

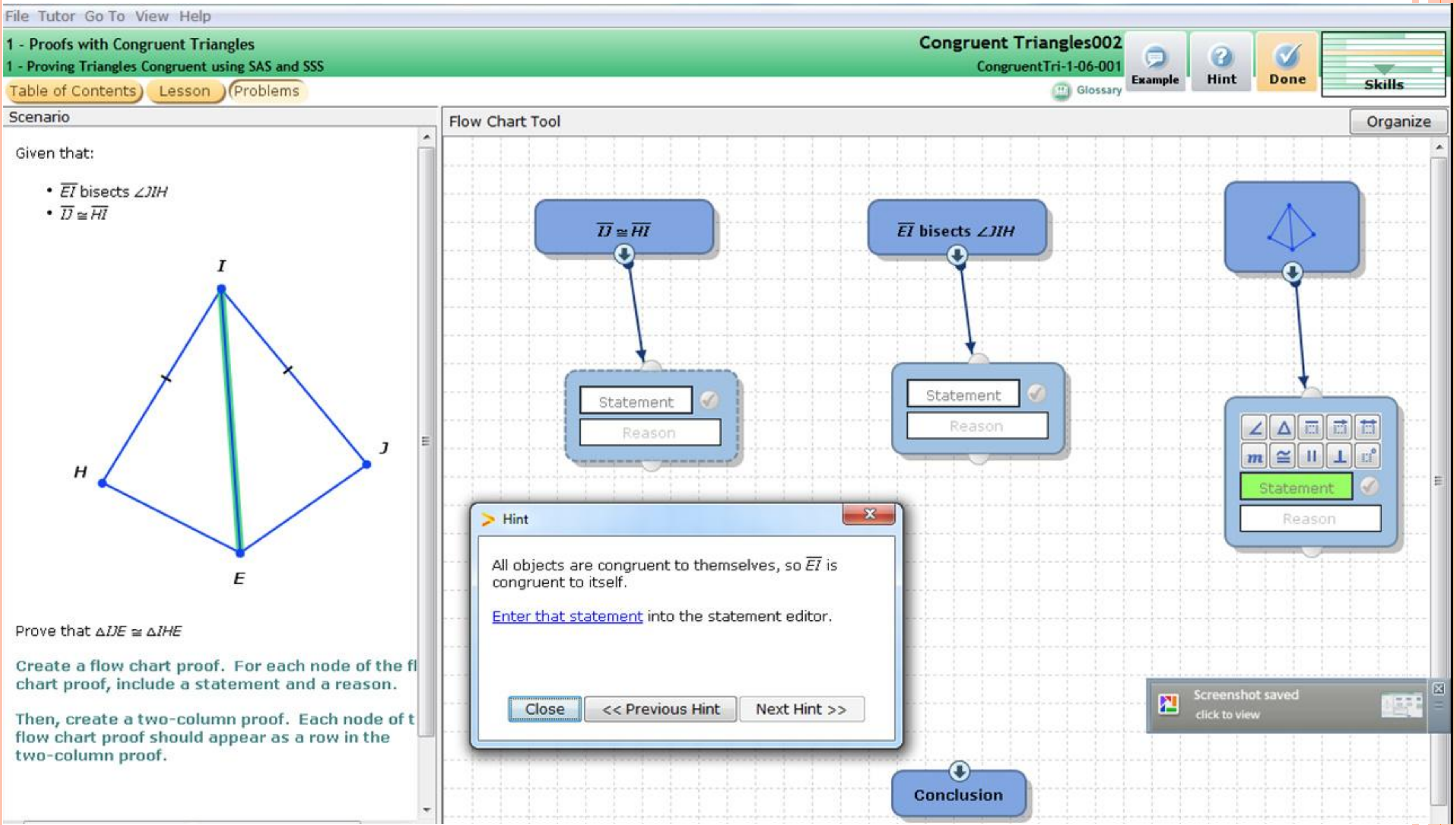
Cognitive Tutor

- Based on J. Anderson's ACT* Theory of Learning

- According to ACT*, learning happens through
  - Generalization
  - Discrimination
  - Strengthening

- Found to be very effective in controlled studies

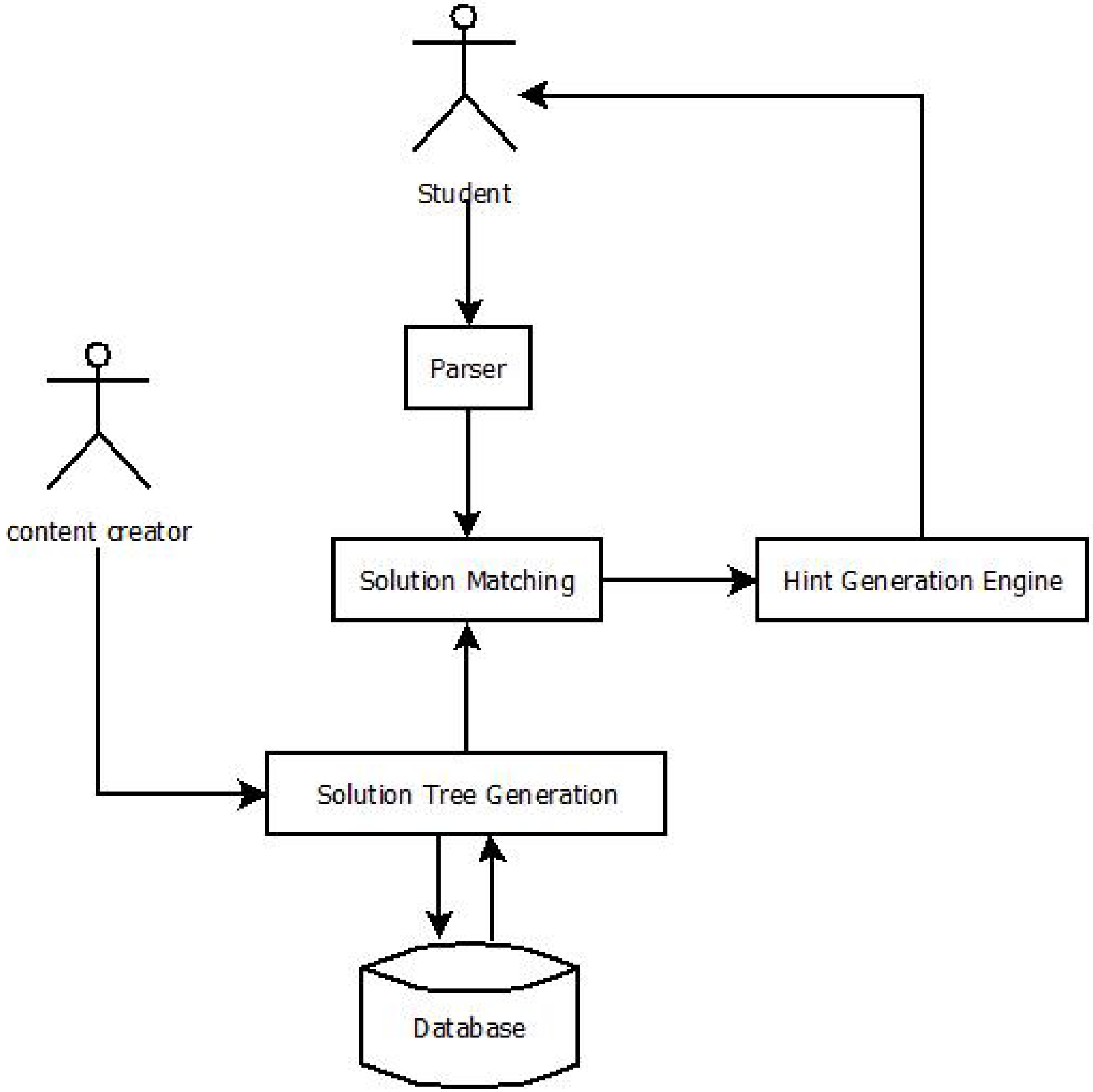
Cognitive Tutor



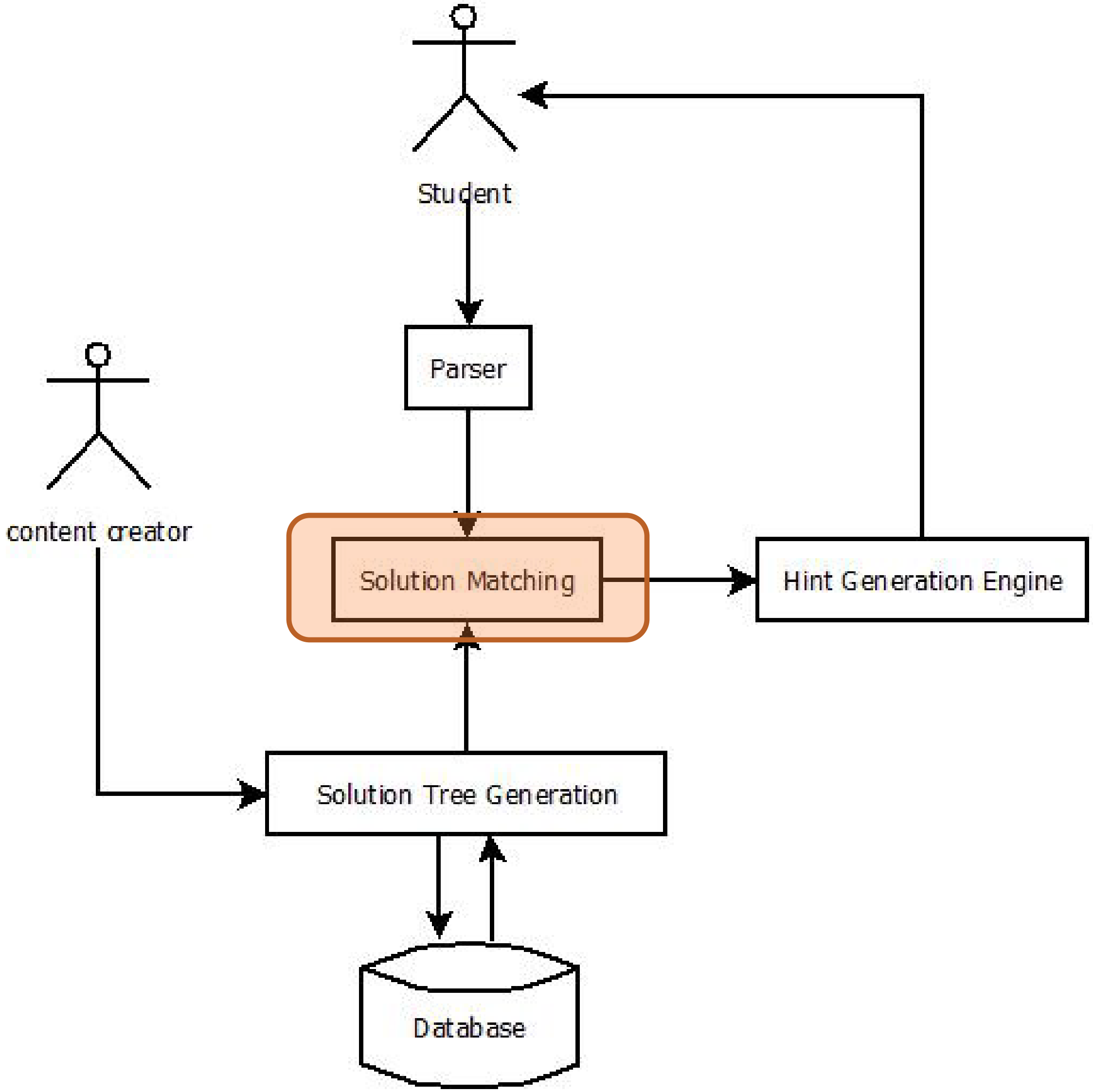
Cognitive Tutor

- Implemented as part of curriculum in a few counties in the US
- Very useful for schools in poor neighborhoods and various special schools
- Not much improvement in student's performance in standard tests

Approach


Student
Parser
content creator
Solution Matching
Hint Generation Engine
Solution Tree Generation
Database

Approach


Student
Parser
content creator
Solution Matching
Hint Generation Engine
Solution Tree Generation
Database

Approach

- To model the solution tree, two models were tried
  - Tree Model
  - Box Model

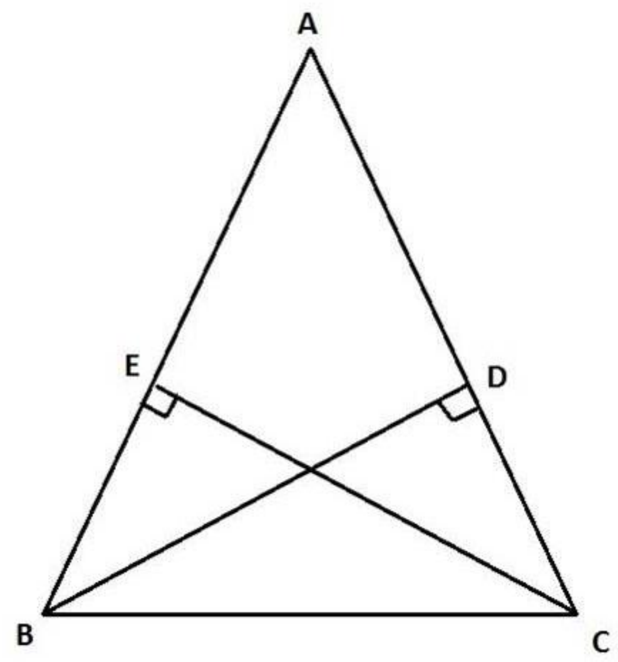
The Tree Model

- Let us explain through an example problem

Example 1:

Given BD and CE are perpendiculars
on AC and AB respectively
and BD = CE.

Prove that ABC is an
Isosceles triangle



The Tree Model

- There are a lot of ways to solve this problem using properties of triangles
- Four different solutions are considered

Example 1

SOL 1

Given:

1. $\angle AEC = 90$
2. $\angle BDA = 90$

3. BD = CE

To prove

AB = AC

Proof:

In ΔABD & ΔACE

BD = CE (given)

$\angle AEC = 90 = \angle BDA$ (given)

4. $\angle A = \angle A$ (common angle)

5. Therefore, $\Delta ABD \cong \Delta ACE$ (AAS)

6. AB = AC (c.p.c.t)

SOL 2

Given:

1. $\angle AEC = 90$
2. $\angle BDA = 90$

3. BD = CE

To prove

AB = AC

Proof:

In ΔABD & ΔACE

BD = CE (given)

7.$\angle ABD = 90 - \angle A$

8. $\angle ACE = 90 - \angle A$

9. $\angle ABD = \angle ACE$

10. Therefore, $\Delta ABD \cong \Delta ACE$ (ASA)

6. AB = AC (c.p.c.t)

Example 1

SOL 3

Given:

11. $\angle BEC = 90$

12. $\angle BDC = 90$

3. BD = CE

To prove

13. $\angle$ ABC = $\angle$ACB

Proof:

In ΔBDC & ΔBEC

BD = CE (given)

$\angle BEC = 90 = \angle BDC$ (given)

14. BC = BC (common side)

15. Therefore, $\Delta BDC \cong \Delta BEC$ (RHS)

16. $\angle EBC = \angle DCB$ (c.p.c.t)

13. $\angle$ ABC = $\angle$ACB (same angle as above)

SOL 4

Given:

1. $\angle AEC = 90$

2. $\angle BDA = 90$

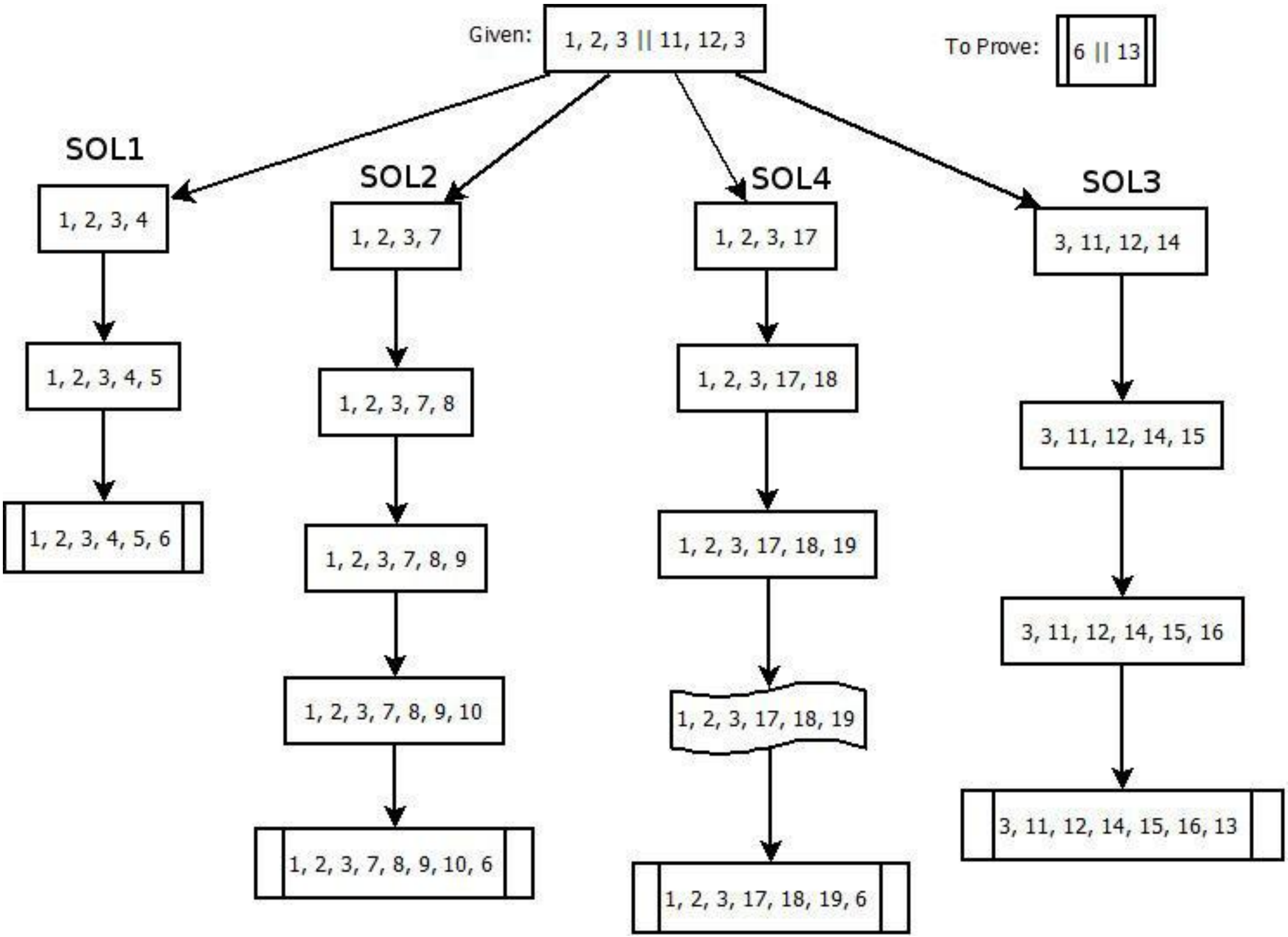
3. BD = CE

17. Area of ΔABC = ½ (BD)(AC)

18. Area of ΔABC = ½ (CE)(AB)

19. ½ (BD)(AC) = ½ (CE)(AB)

6. AB = AC (because BD = CE)


Given:
1, 2, 3 || 11, 12, 3
To Prove:
6 || 13
SOL1
1, 2, 3, 4
1, 2, 3, 4, 5
1, 2, 3, 4, 5, 6
SOL2
1, 2, 3, 7
1, 2, 3, 7, 8
1, 2, 3, 7, 8, 9
1, 2, 3, 7, 8, 9, 10
1, 2, 3, 7, 8, 9, 10, 6
SOL4
1, 2, 3, 17
1, 2, 3, 17, 18
1, 2, 3, 17, 18, 19
1, 2, 3, 17, 18, 19
1, 2, 3, 17, 18, 19, 6
SOL3
3, 11, 12, 14
3, 11, 12, 14, 15
3, 11, 12, 14, 15, 16
3, 11, 12, 14, 15, 16, 13

The Tree Model

- Advantages
  - State based
  - Handles multiple solutions for a given problem

- Disadvantages
  - Slight modification in proof will require a whole new branch
  - Change in order of steps will spawn a new branch
  - Difficult to model steps with algebraic manipulations
  - Depending on how the hypothesis is interpreted, two disjoint trees may be formed
  - Very inefficient in space

The Box Model

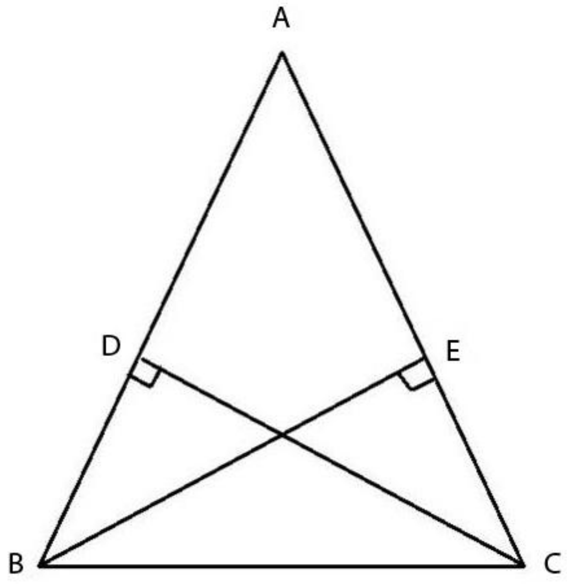
- Let us explain the box model using a modification of Example 1

Example 2:

Given ABC is an Isosceles triangle.

BD and CE are perpendiculars on AC and AB respectively.

Prove that BD = CE.



Example 2

Proof 1 (P1):

Given:

1. $AB = AC$
2. $\angle BDC = 90$
3. $\angle BEC = 90$

To prove:

$BE = CD$

Proof:

In ΔABE & ΔACD

4. $\angle A = \angle A$ (common angle)
5. $\angle ABE = 90 - \angle A$
6. $\angle ACD = 90 - \angle A$
7. $\angle ABE = \angle ACD$
8. $\Delta ABE \cong \Delta ACD$ (A.S.A property)
9. $BE = CD$ (c.p.c.t)

Proof 2 (P2):

Given:

10. $\angle ABC = \angle ACB$
2. $\angle BDC = 90$
3. $\angle BEC = 90$

To prove:

$BE = CD$

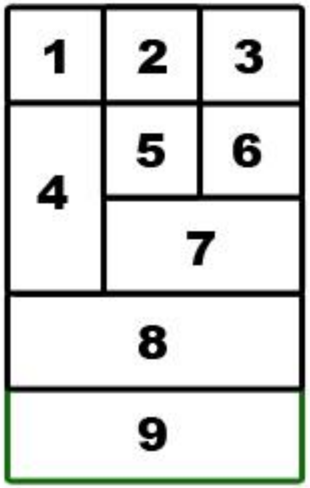
Proof:

In ΔBDC & ΔCEB

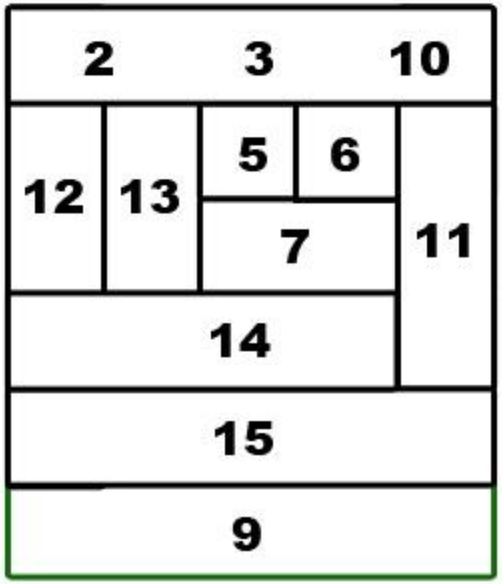
11. $BC = BC$ (common side)
5. $\angle ABE = 90 - \angle A$
6. $\angle ACD = 90 - \angle A$
7. $\angle ABE = \angle ACD$
12. $\angle EBC = \angle ABC - \angle ABE$
13. $\angle DCB = \angle ACB - \angle ACD$
14. $\angle EBC = \angle DCB$ (from 7, 10, 12, 13)
15. $\Delta BDC \cong \Delta CEB$ (A.S.A property)
9. $BE = CD$ (c.p.c.t)

The Box Model

proof 1



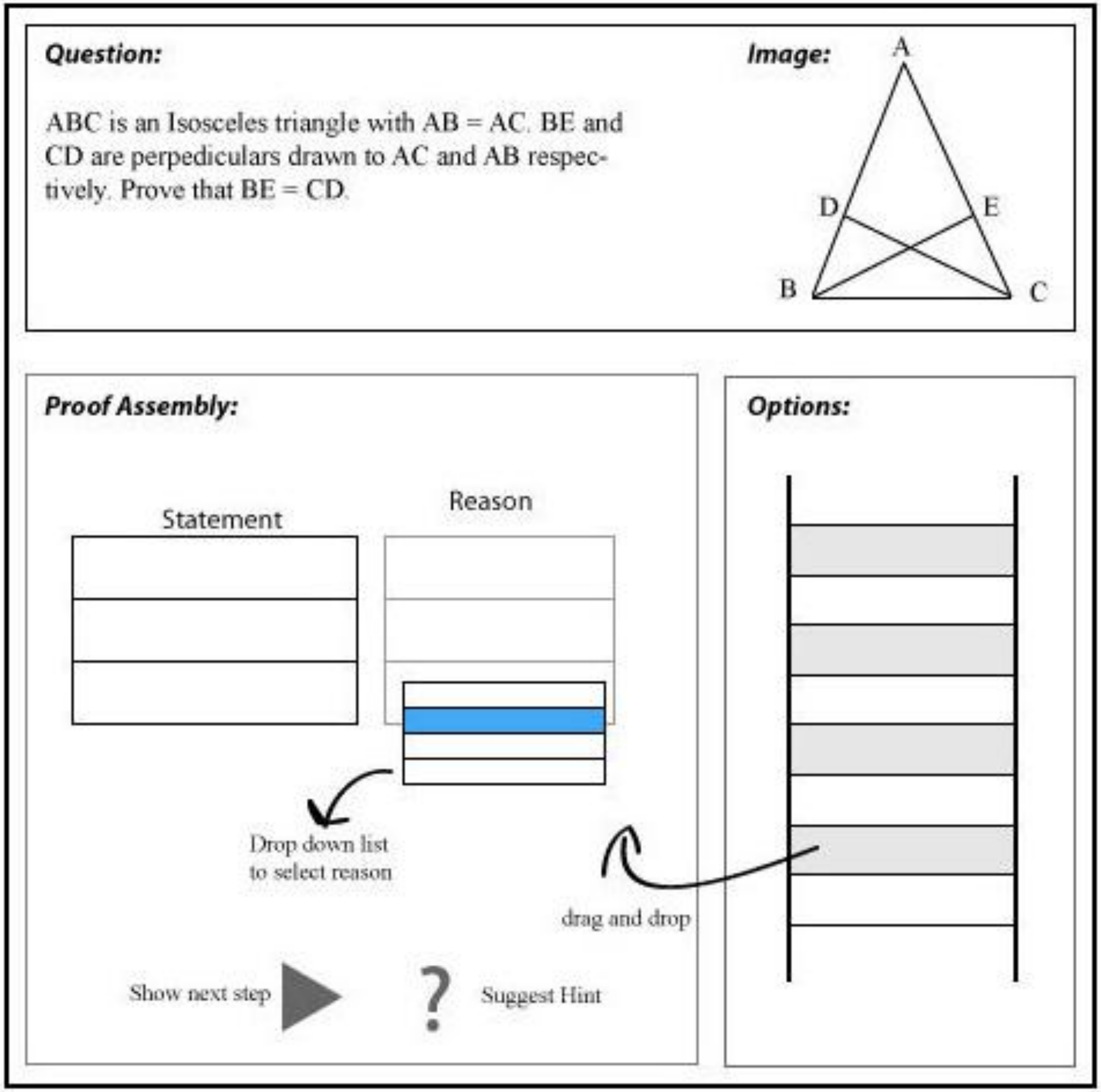
proof 2



The Box Model

- Advantages
  - Handles variable order of steps using no extra space

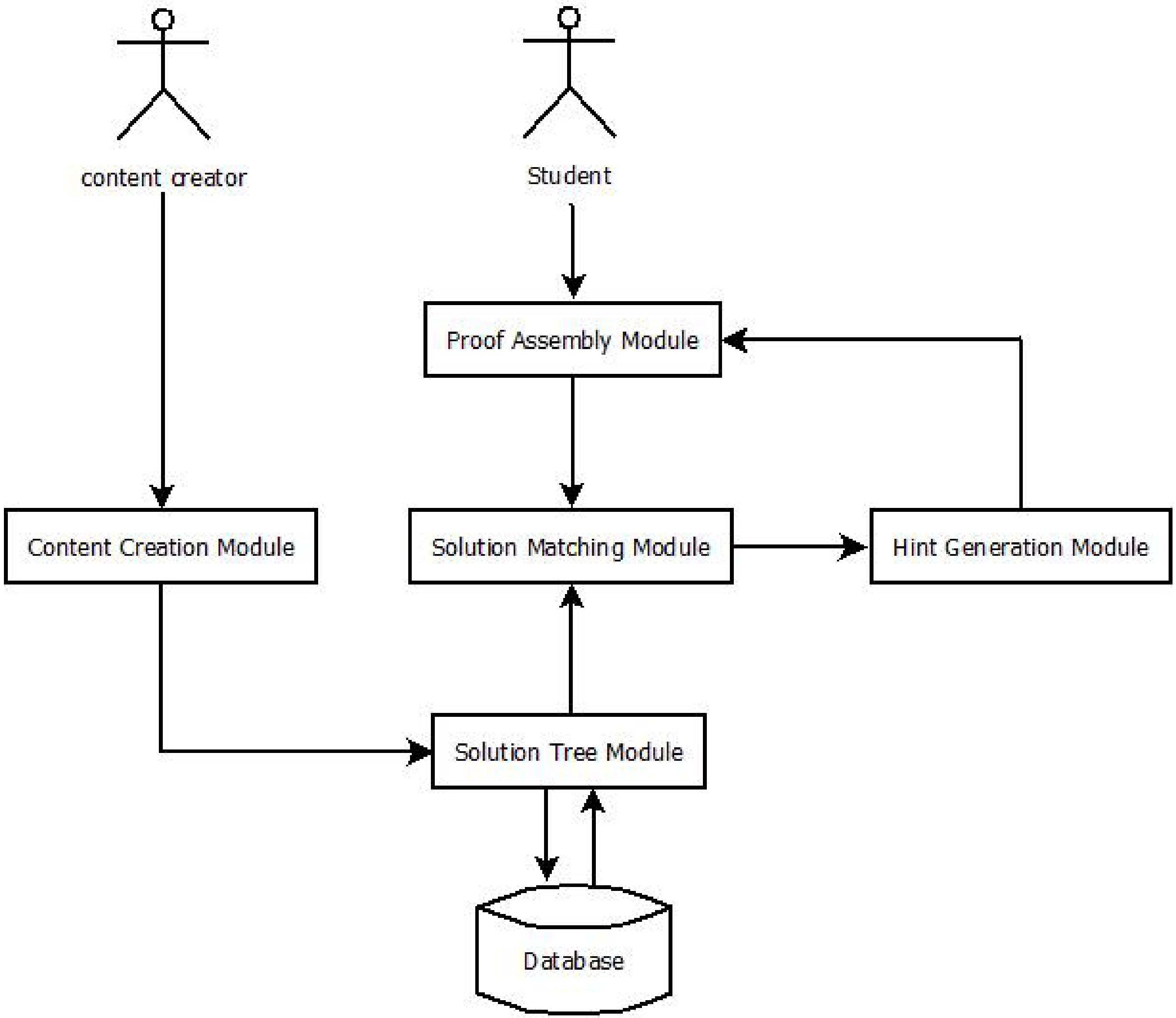
- Disadvantages
  - Generation of box models is not trivial
  - Does not handle algebraic manipulations efficiently
  - Very tedious to implement and use



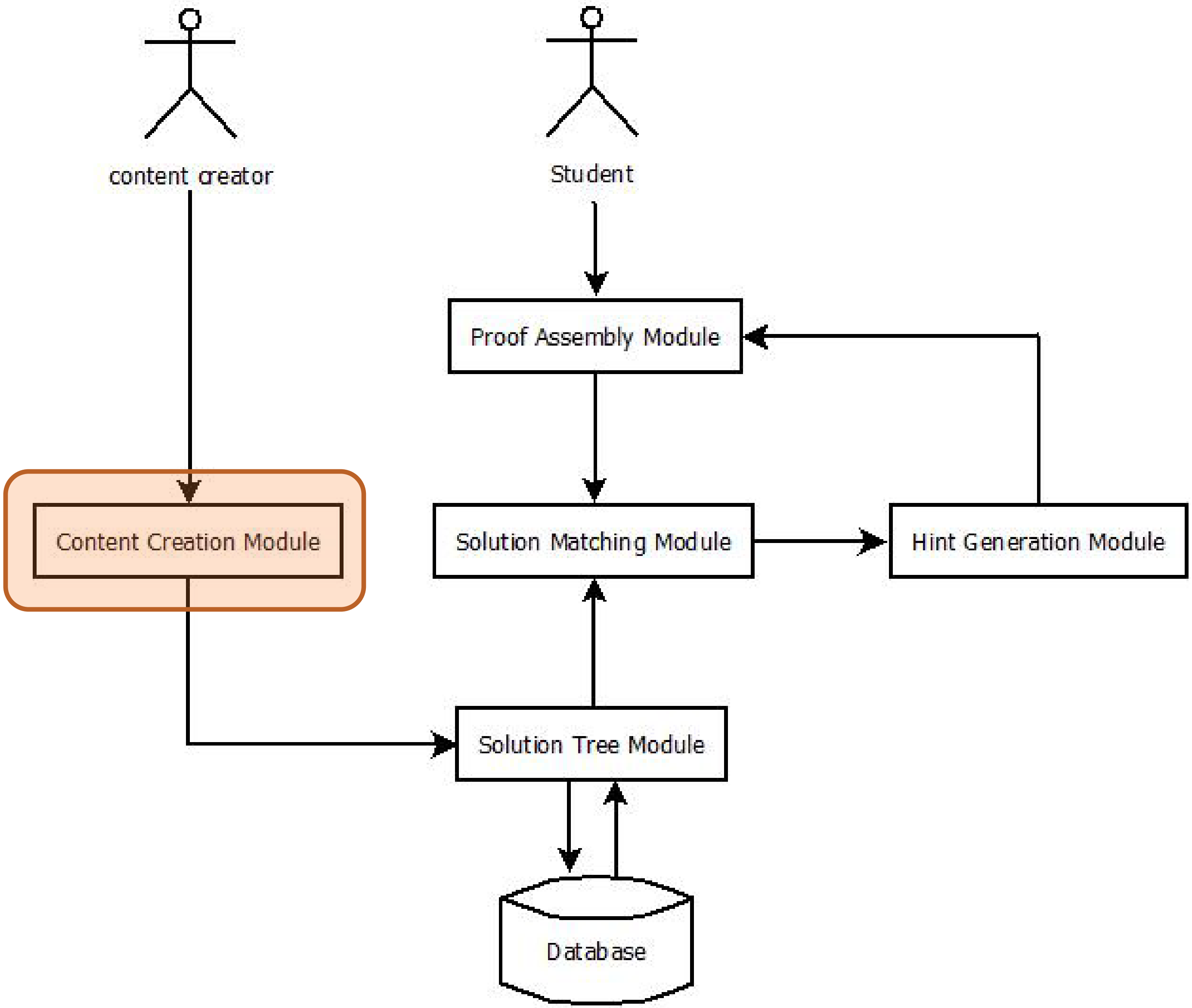
Problem Statement Revised

- The proof is assembled using an MIT Scratch-like Interface
- The rest of the functional requirements remain more or less the same

Design

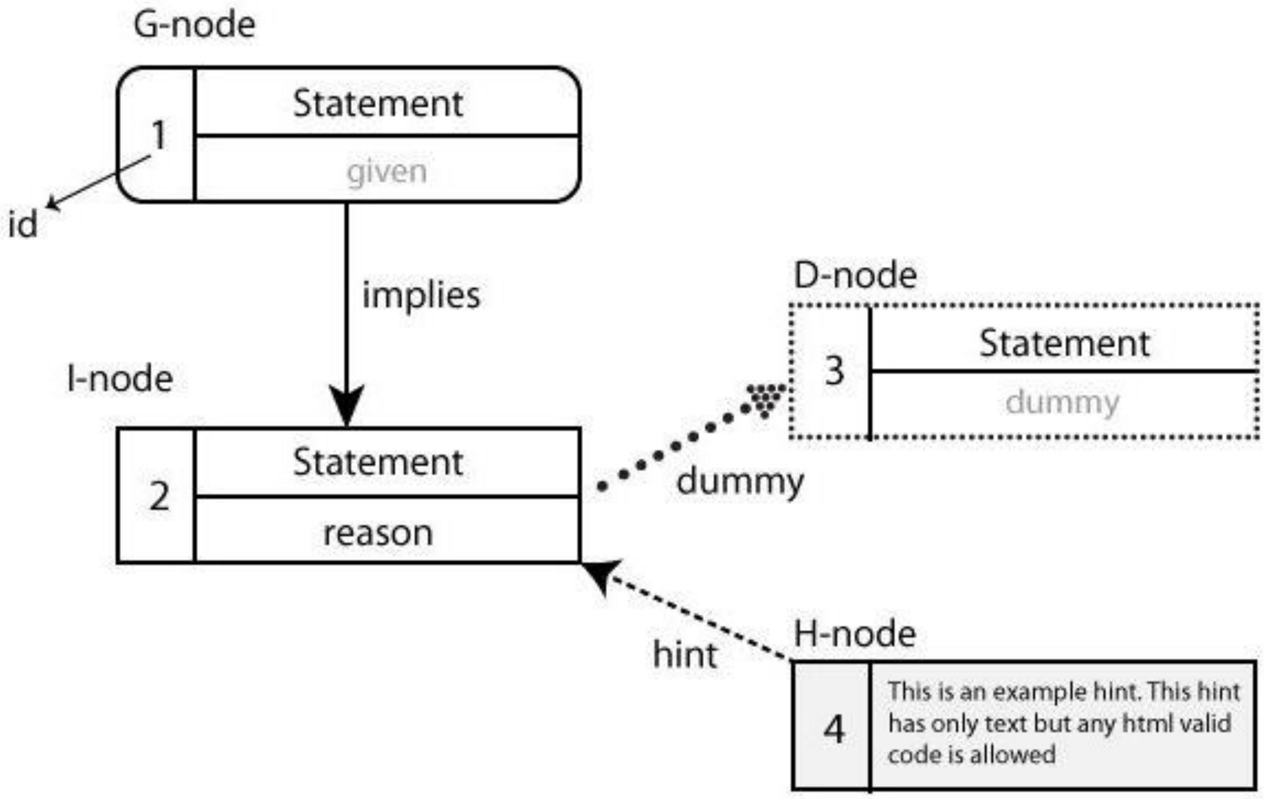

content creator
Student
Proof Assembly Module
Content Creation Module
Solution Matching Module
Hint Generation Module
Solution Tree Module
Database

Content Creation Module

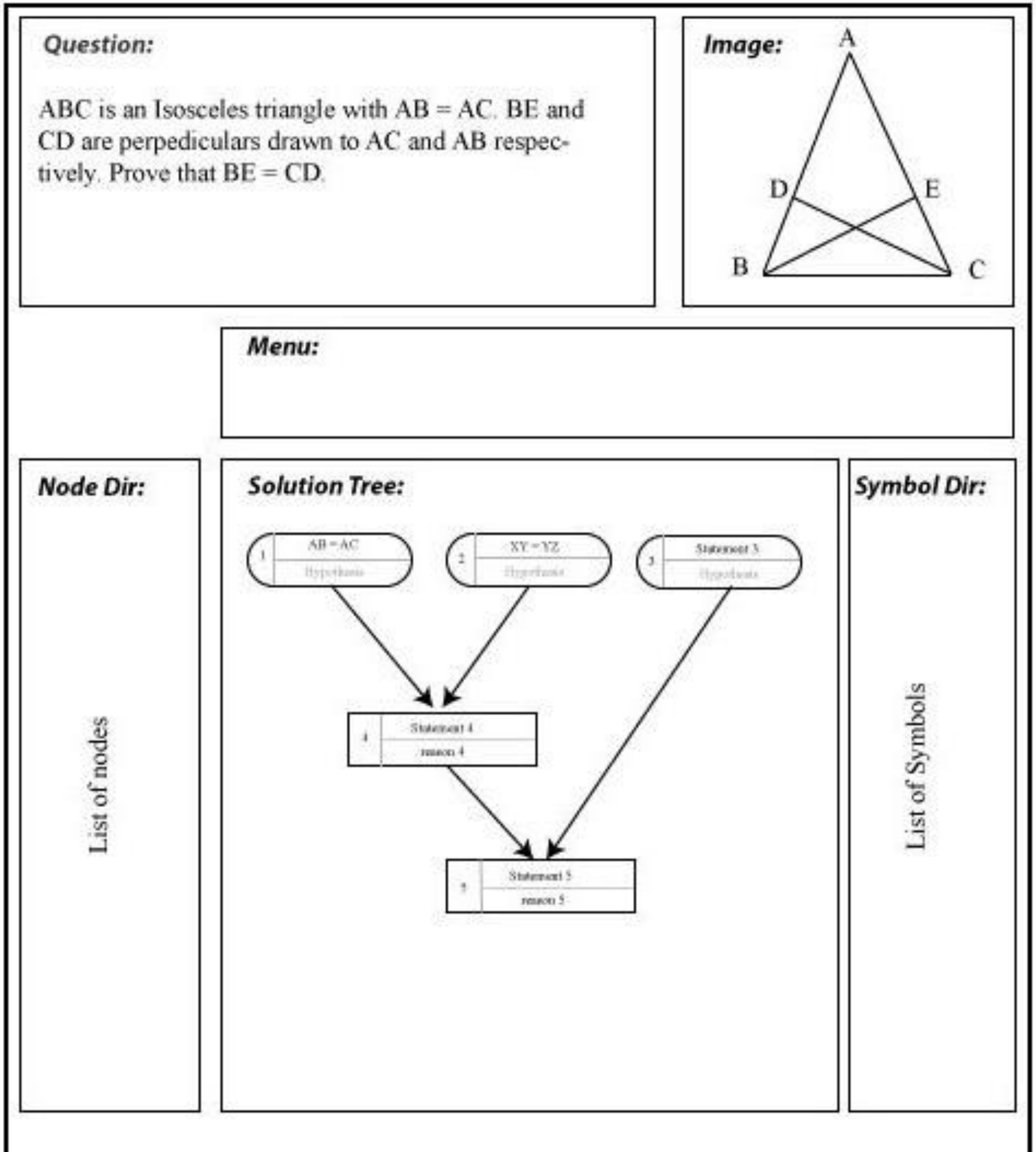


Content Creation Module

- Solution Tree:
  - Nodes and Links

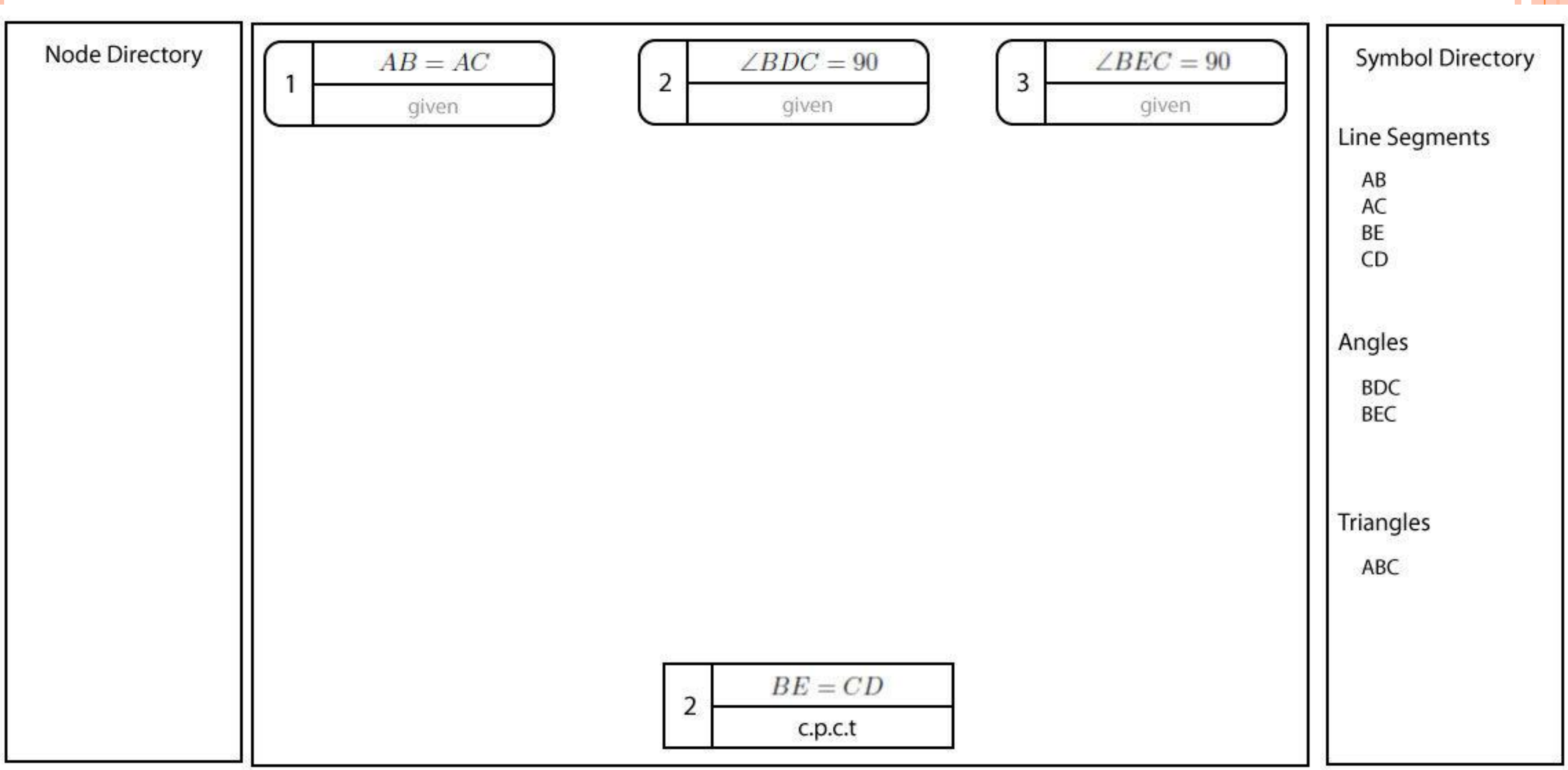


Content Creator's Interface

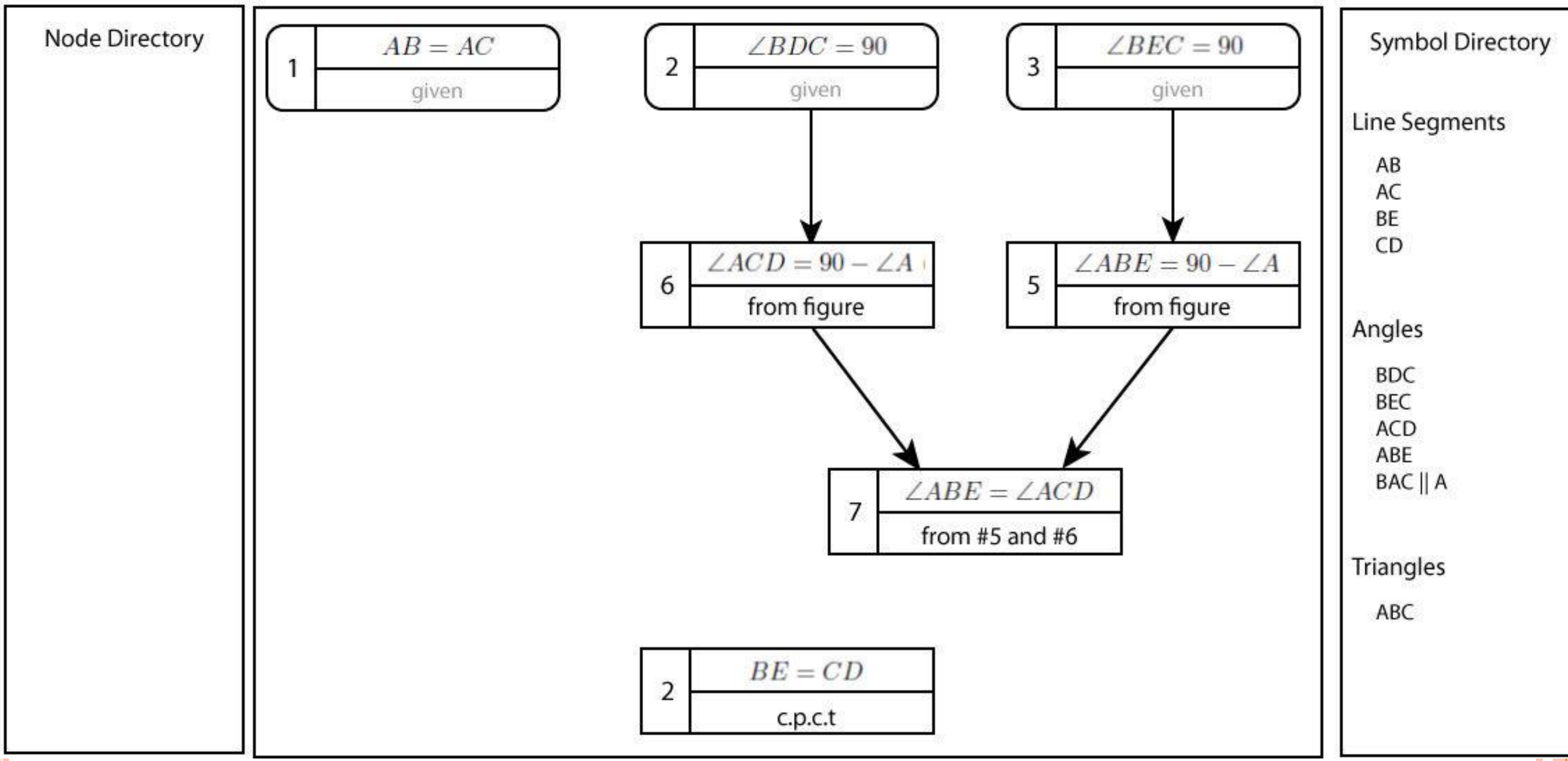


- The content creator builds the solution tree using the tools that are provided in the menu

Building the Solution Tree

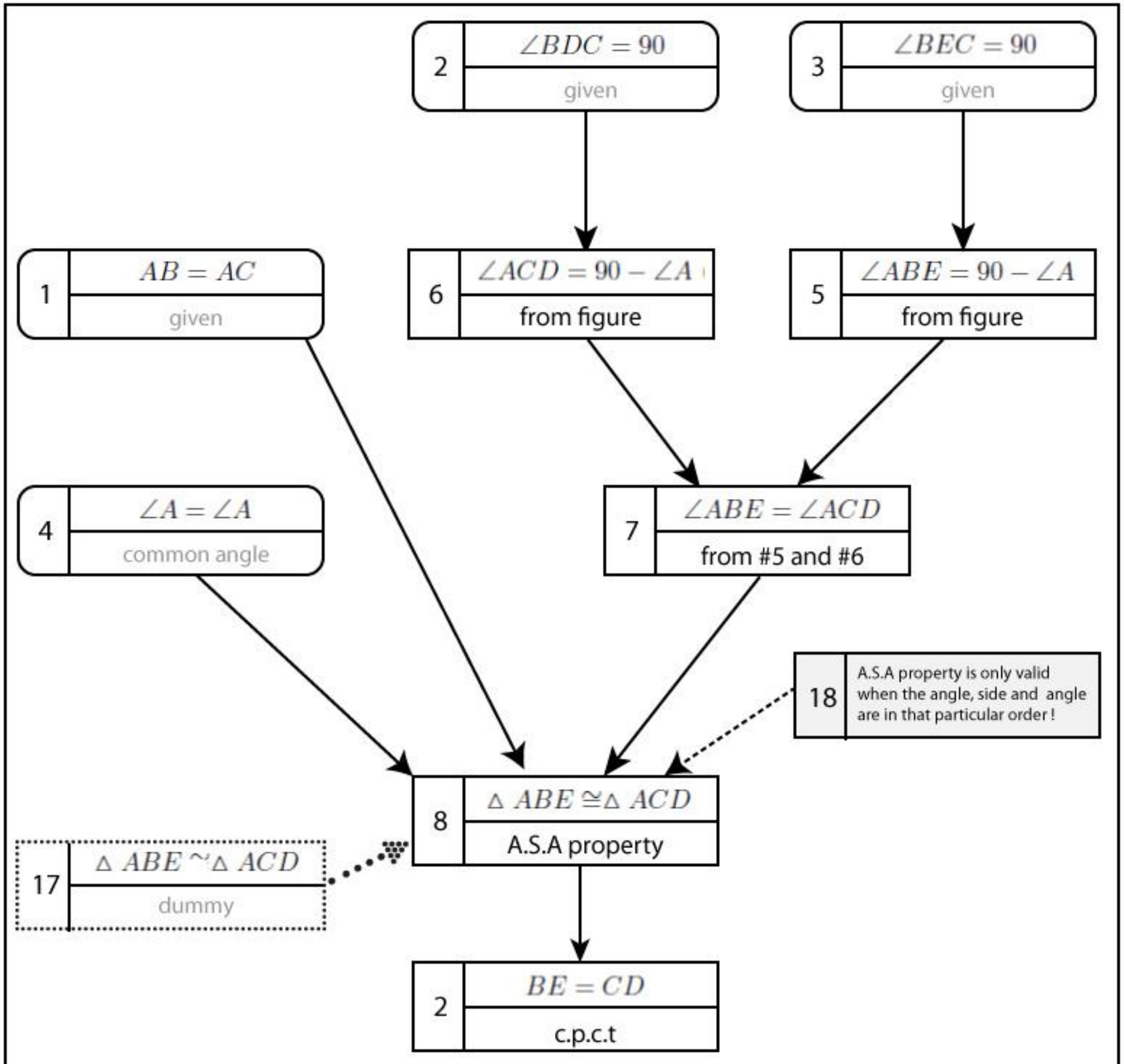


Building the Solution Tree



Building the Solution Tree

Node Directory



Symbol Directory

Line Segments

AB
AC
BE
CD

Angles

BDC
BEC
ACD
ABE
BAC || A

Triangles

ABC
ABE
ACD

Solution Tree

- Representing the Solution Tree
  - We represent the solution tree in the system in XML
  - Flexible, scalable and cross platform compatibility
  - The schema is defined as follows

XML Schema

- Node

```
<node id="2" type="g-node">
  <text>BDC = 90</text>
  <statement>
    <eq>
      <ang>BDC</ang>
      <num>90</num>
    </eq>
  </statement>
 <reason>given</reason>
</node>
```

- Link

```
<link type="implication" source="1" target="3" />
<link type="implication" source="2" target="5" />
```

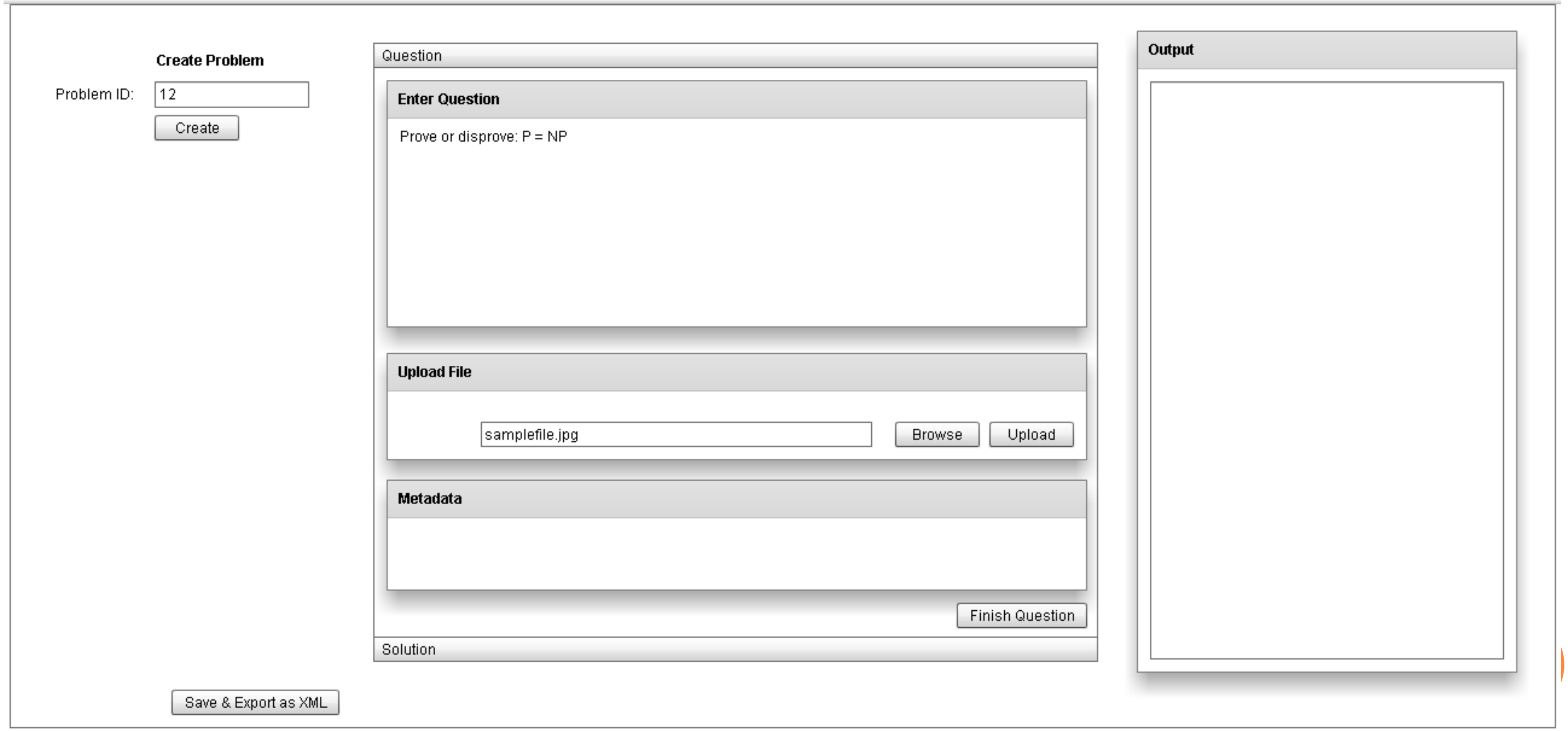
XML Schema

- Problem

```
<problem id="2">
  <question> lorem ipsum… </question>
  <image src="path/to/image" />
  <solution id="1">
    <node id="1">
      …
    </node>
    <link type="implication" source="1" target="3" />
      …
  </solution>
  …
  …
  …
</problem>
```

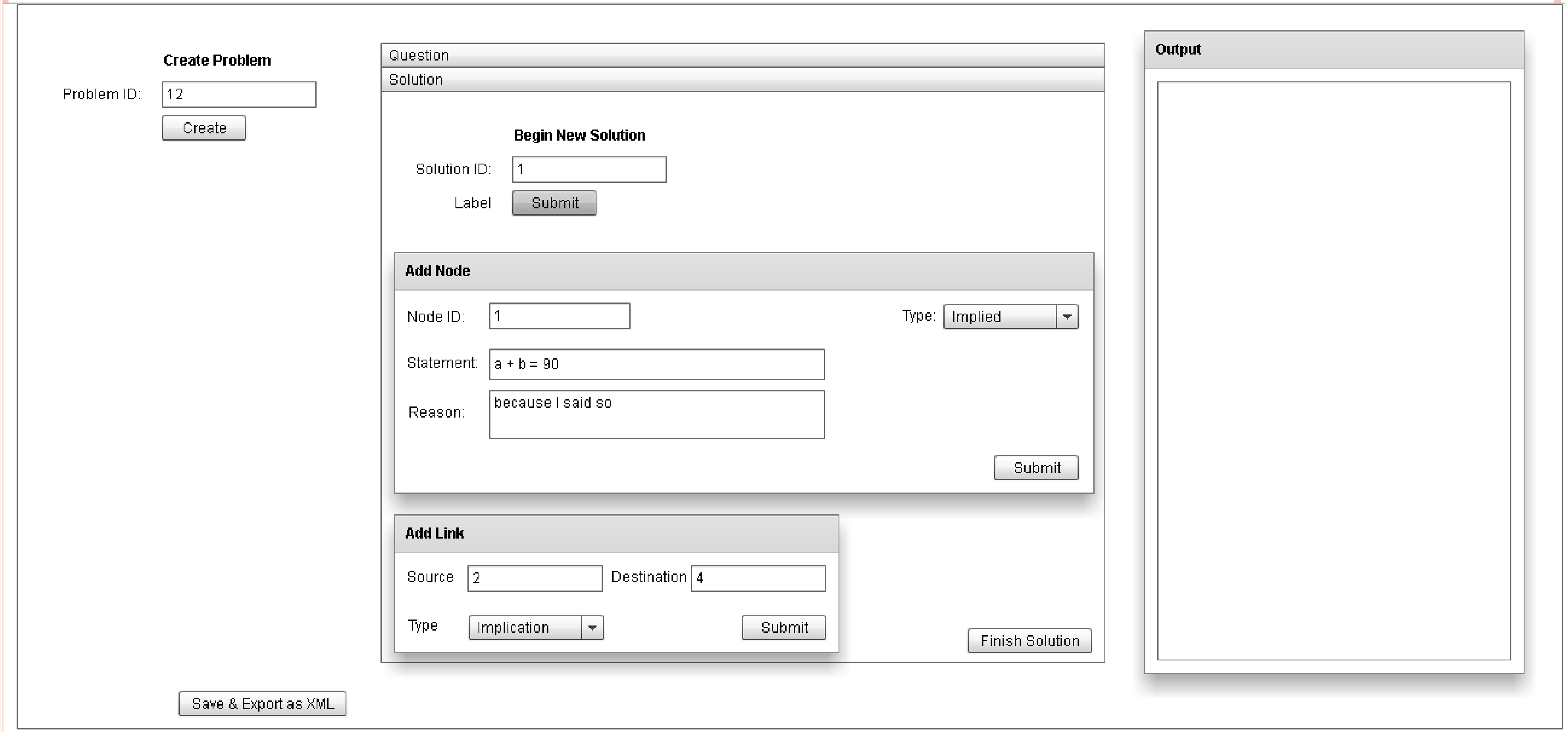
Content Creator Interface

- CC interface in Question mode:

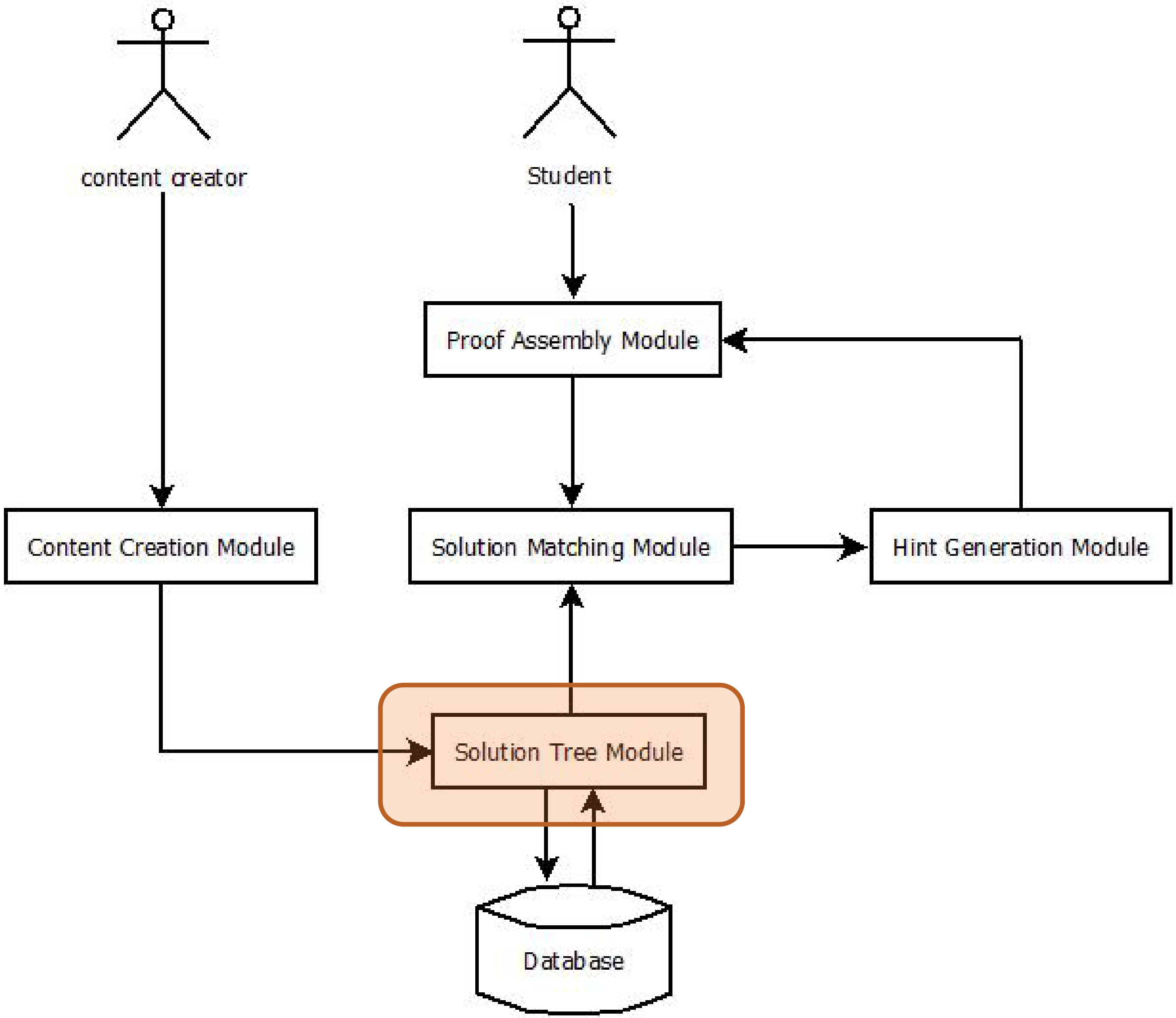


Content Creator Interface

- CC interface in Solution mode:

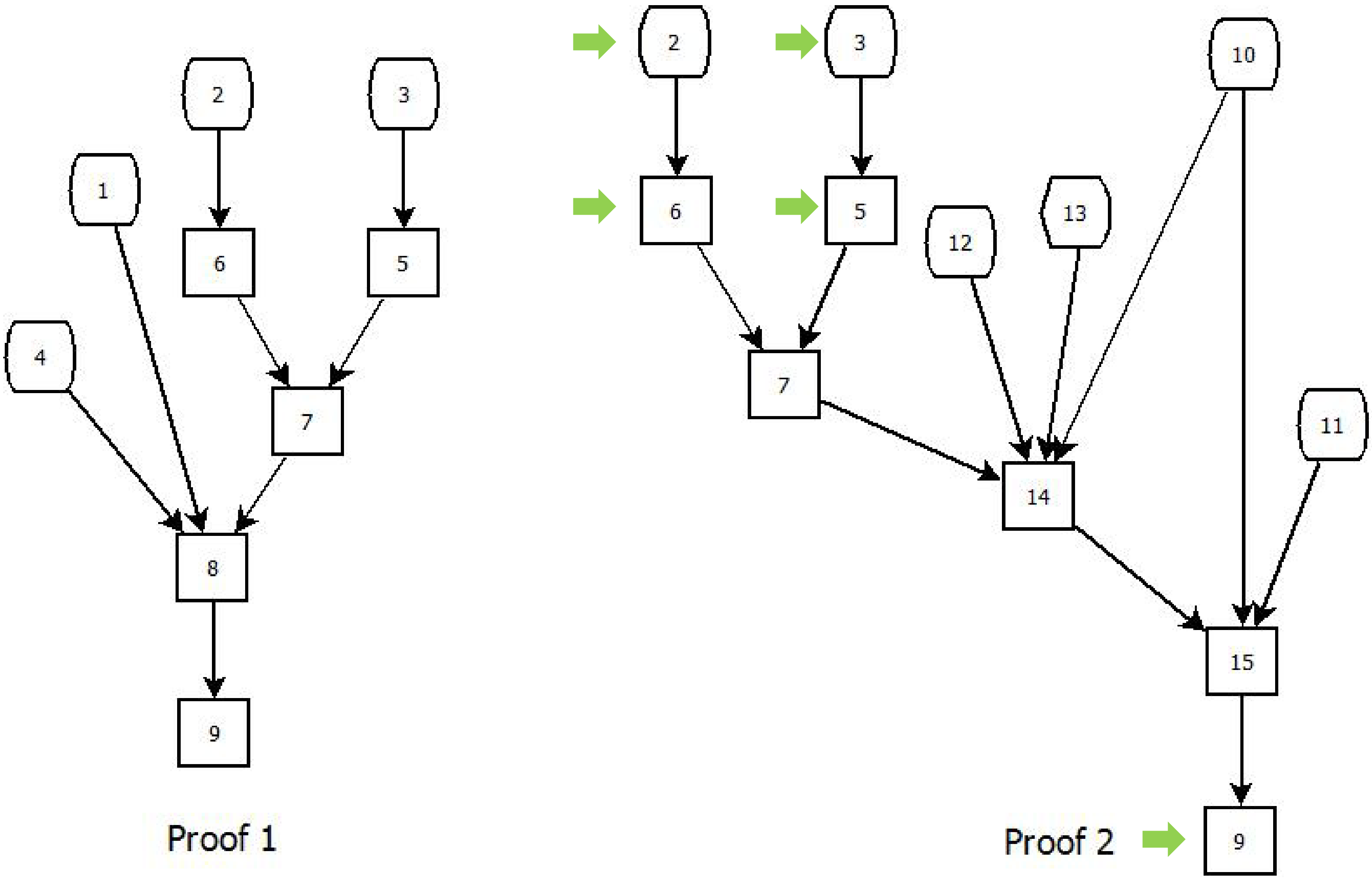


Solution Tree Module



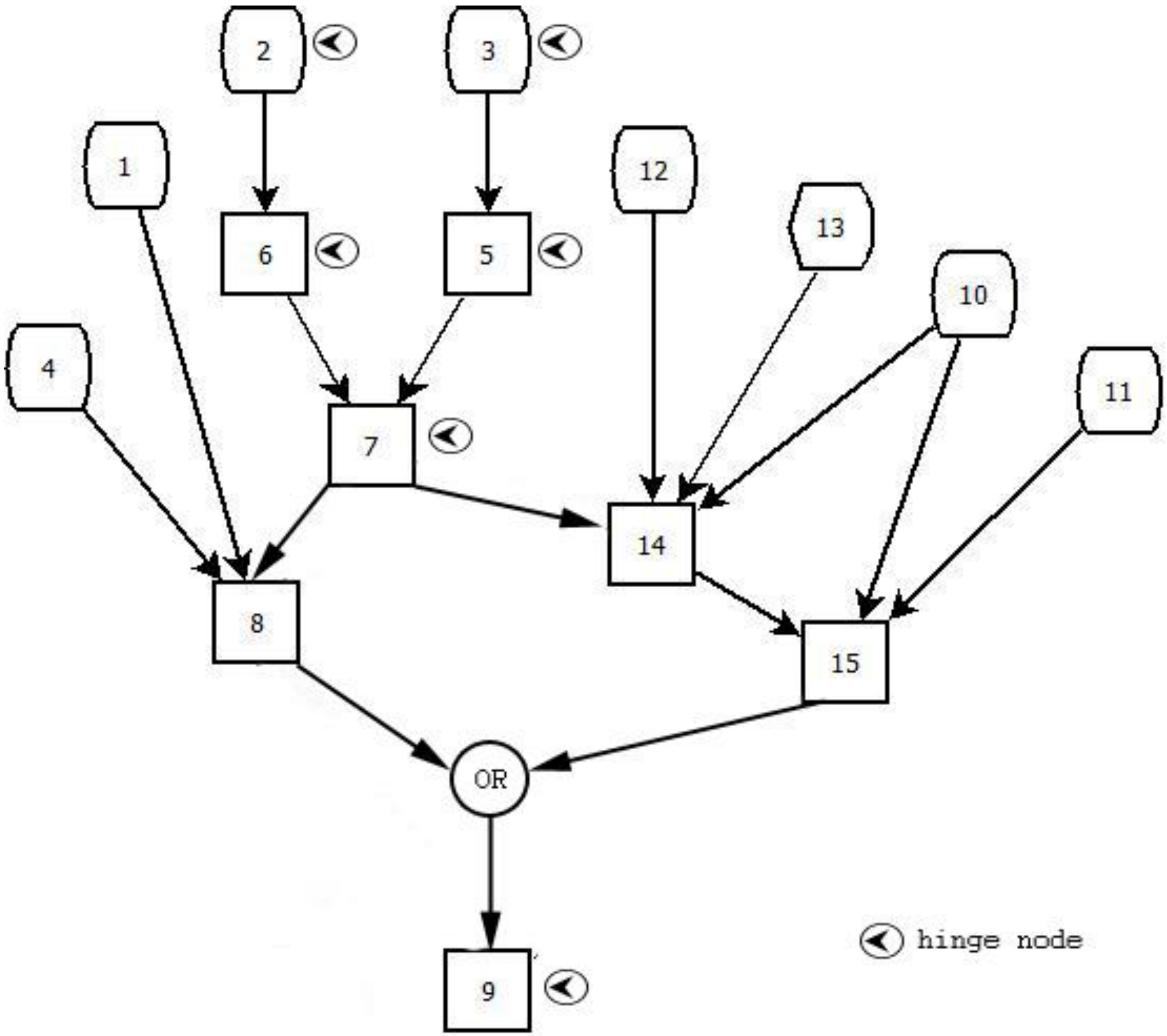
Merge Solutions

- Two solutions of Example 2



Merge Solutions

- Solutions merged along common nodes



Solution Tree Module

- Equation Node
  - Fundamental element of the GST
  - Acts as hinge node whenever required

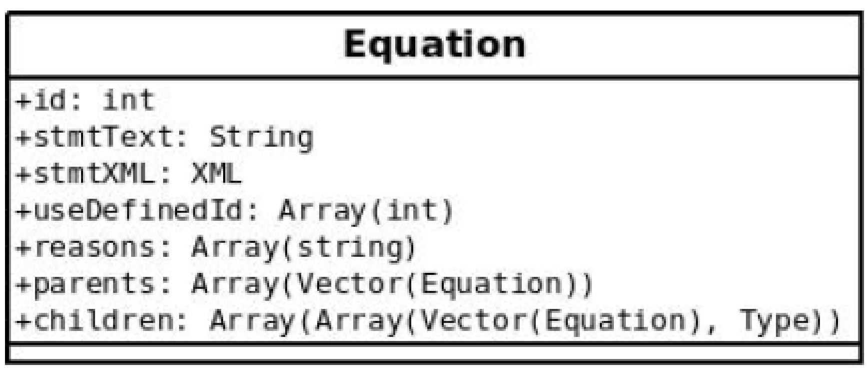


Figure 5.2: Equation Class and it's Attributes

Solution Tree Module

- Tree Merge Algorithm

```
foreach node in Solution Tree do
  if Equation with same Statement as node exists then
    Merge contents of node with Equation
  else
    Create new Equation with contents of node
  end
end
foreach link in Solution Tree do
  update children and parents arrays of corresponding Equations
end
```

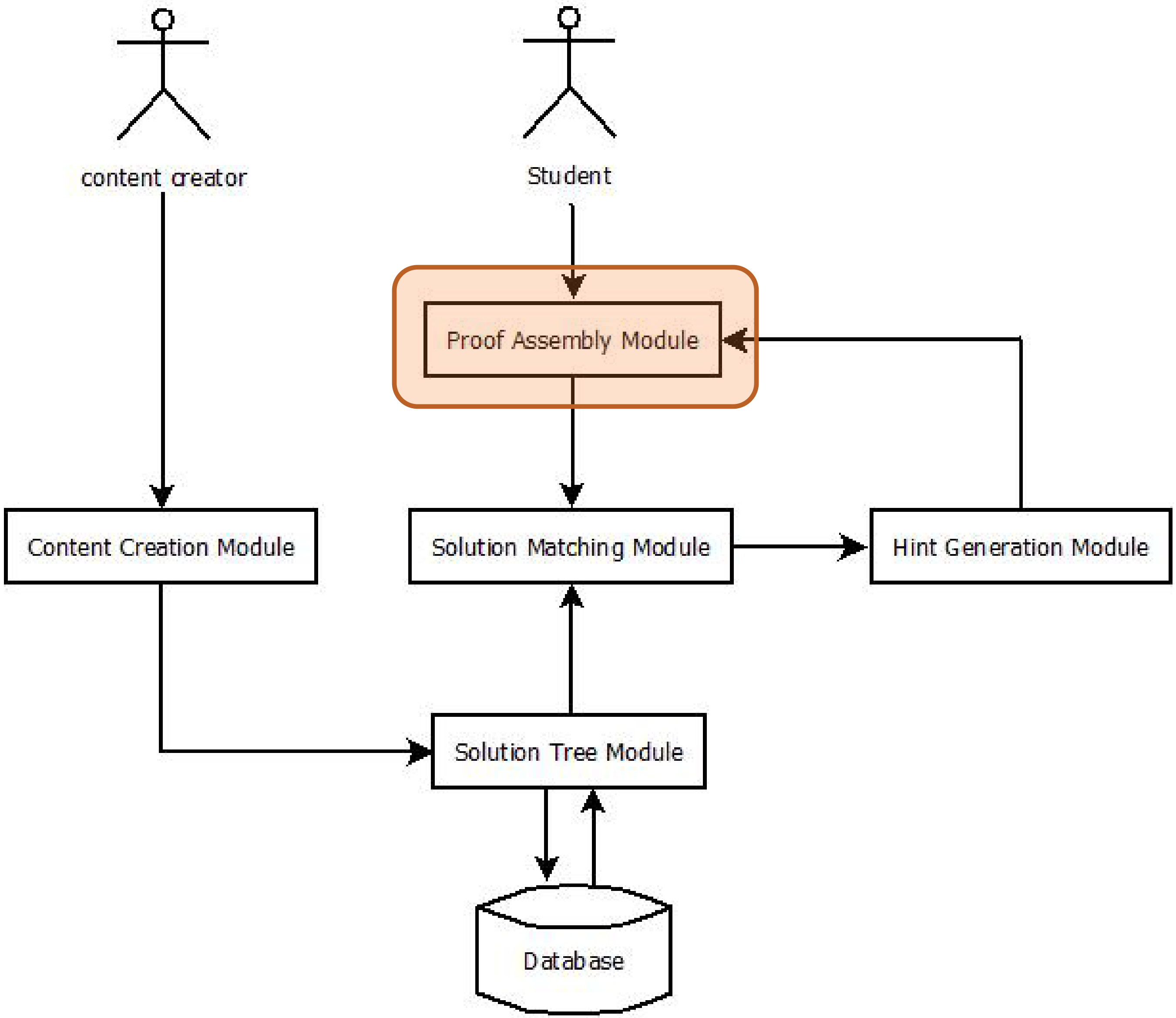
The General Solution Tree

- GST
  - Contains all the solutions in one tree
  - Includes generated dummy nodes, extra images and hints etc.
  - Saved as XML

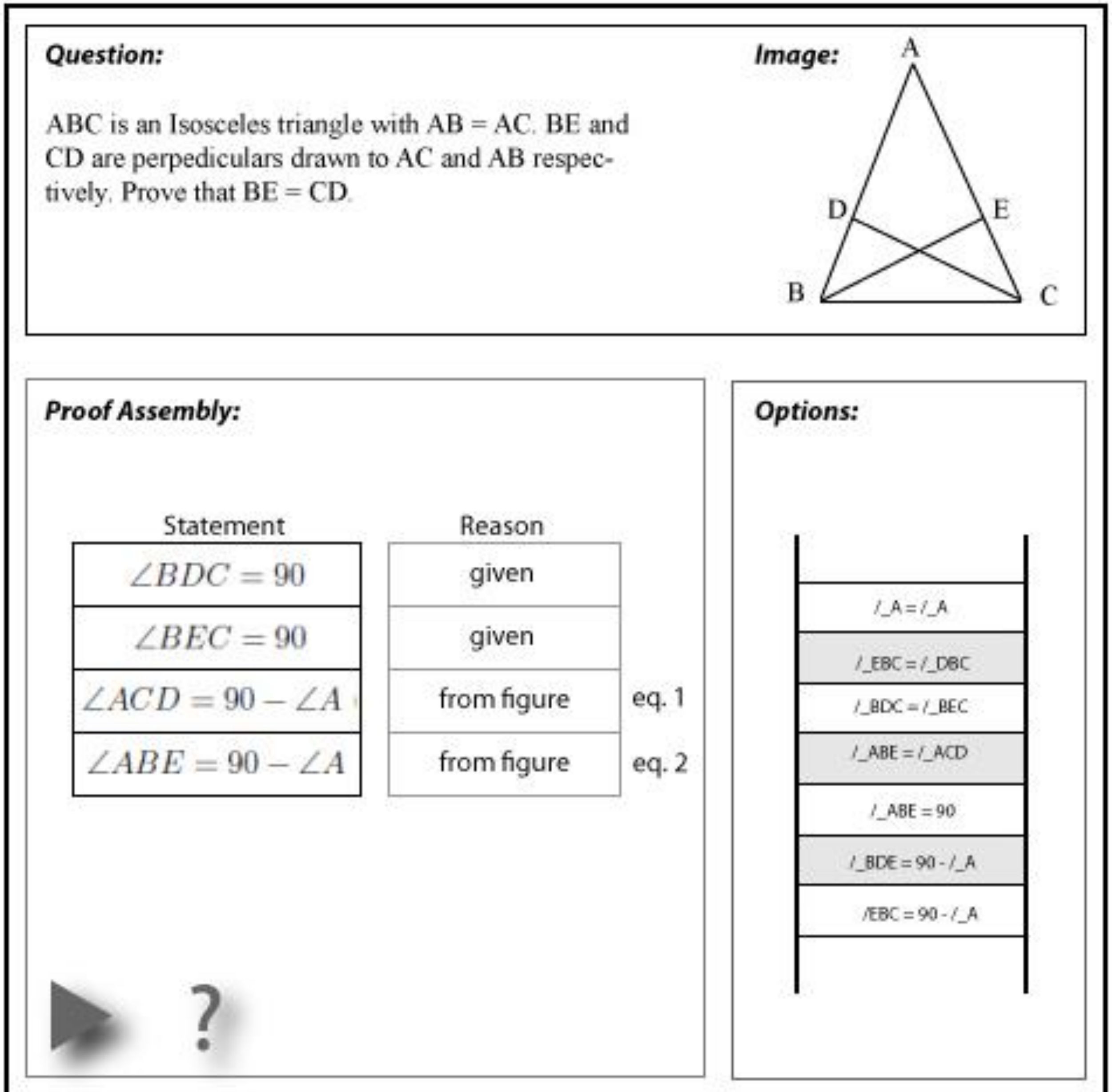
General Solution Tree

```
<problem id="1">
   <question>lorem ipsum… </question>
   <image src="path/to/image" />
   <equations>
      <equation id="$eqn_id">
         …
      </equation>
      …
   </equations>
   <solution id="1">
      <link src="4" target="7" type="implication" />
      …
      …
      <reason id="$eqn_id">Given</reason>
      …
      …
   </solution>
   …
   …
</problem>
```

Proof Assembly Module

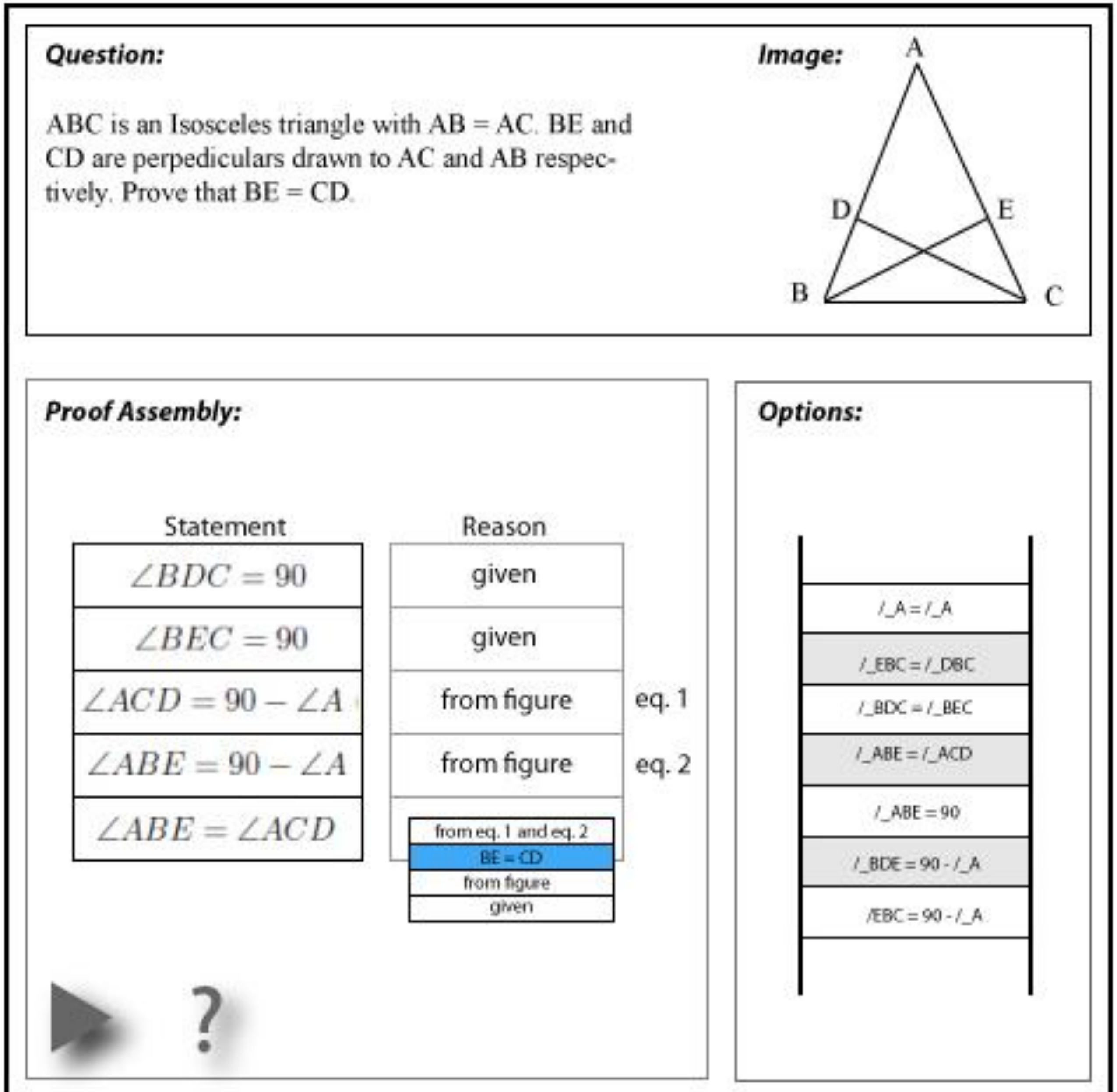


Proof Assembly



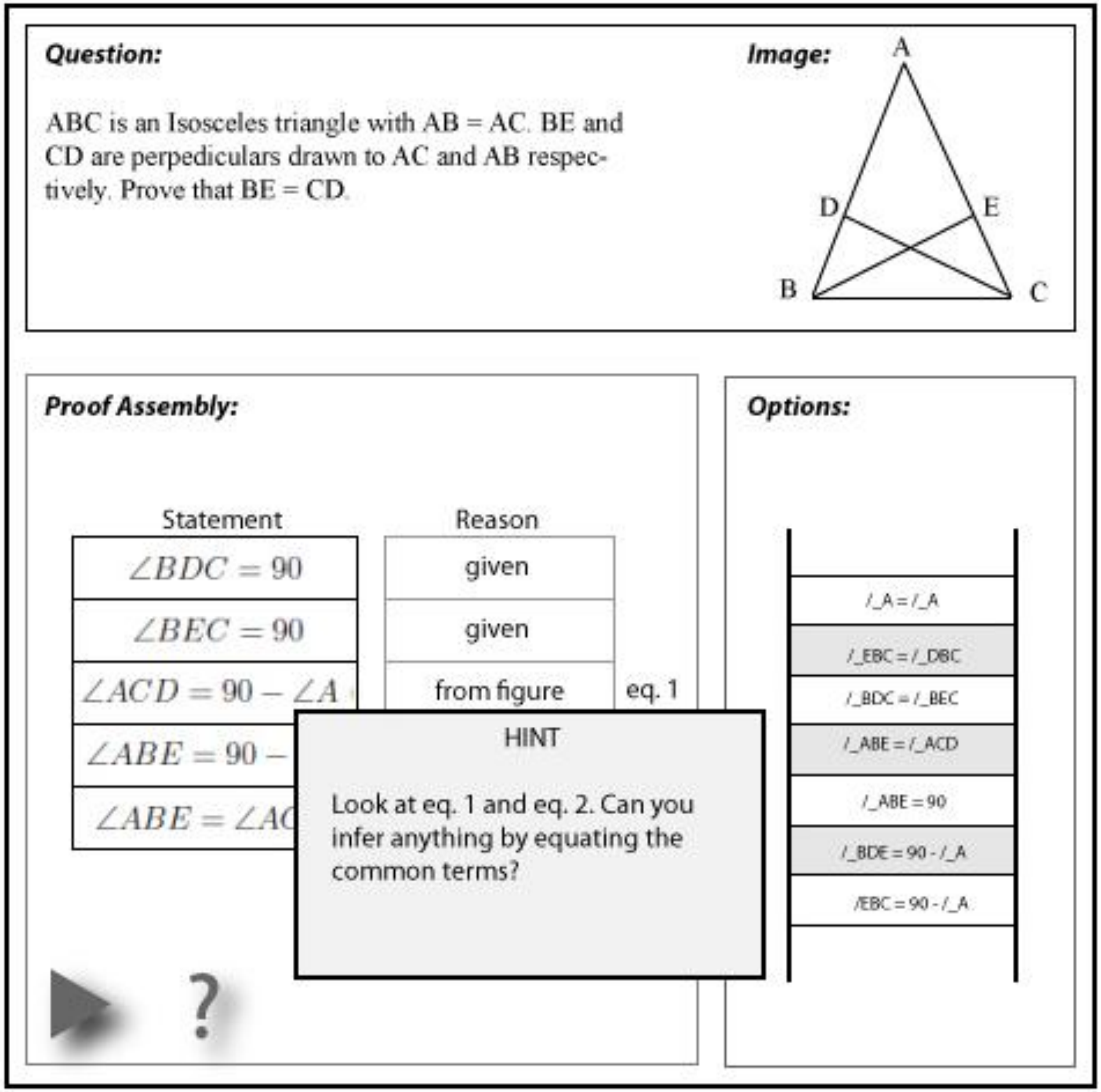
The student chooses an option from the options stack and drags it to the proof assembly area

Proof Assembly



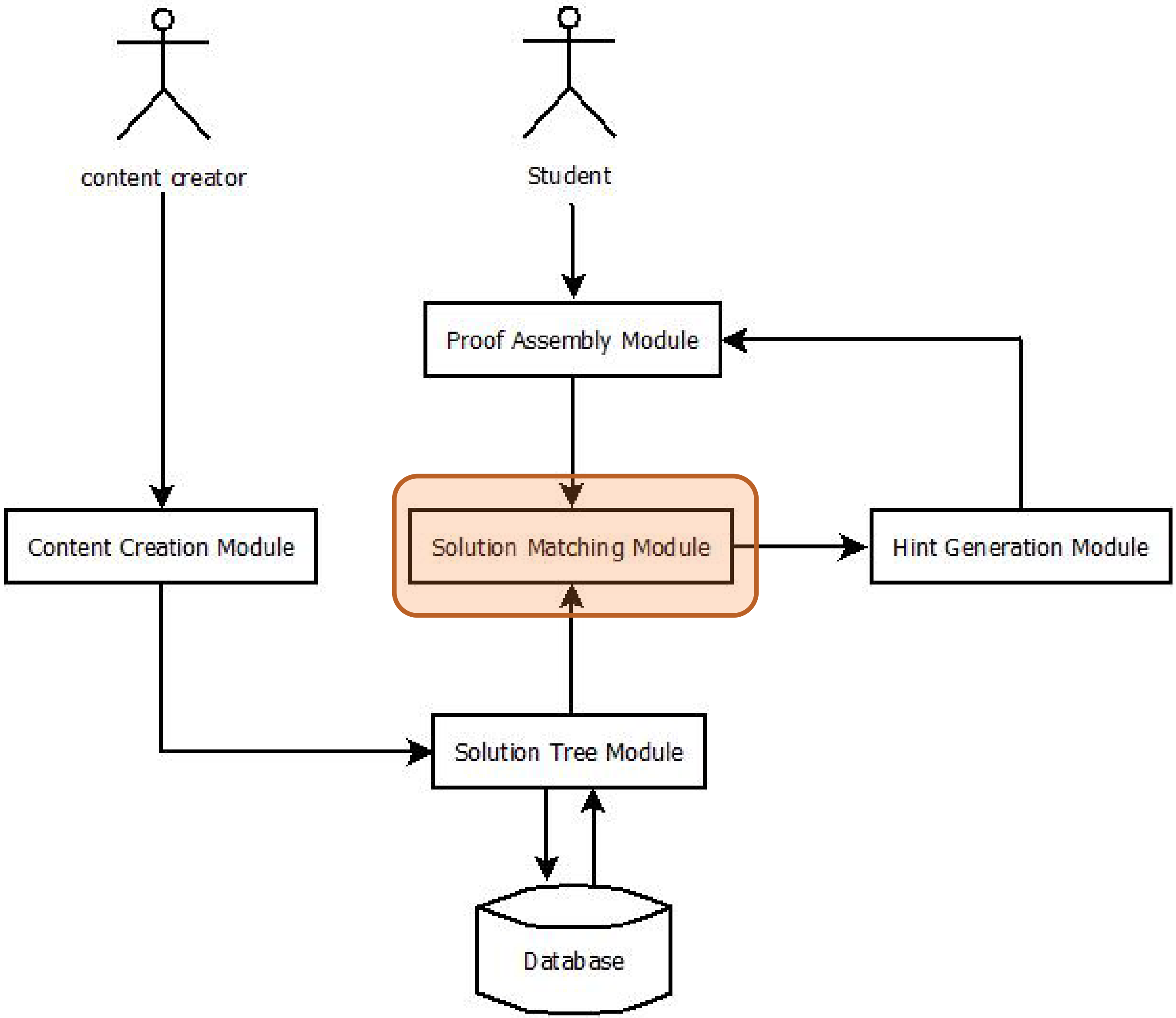
As soon as he drags and drops an assertion, a drop down menu appears from which the student has to choose a reason for the assertion

Proof Assembly



If he makes a mistake or if he presses the "next step" or "hint" button, the hint generation module is called which will give a hint

Solution Matching Module



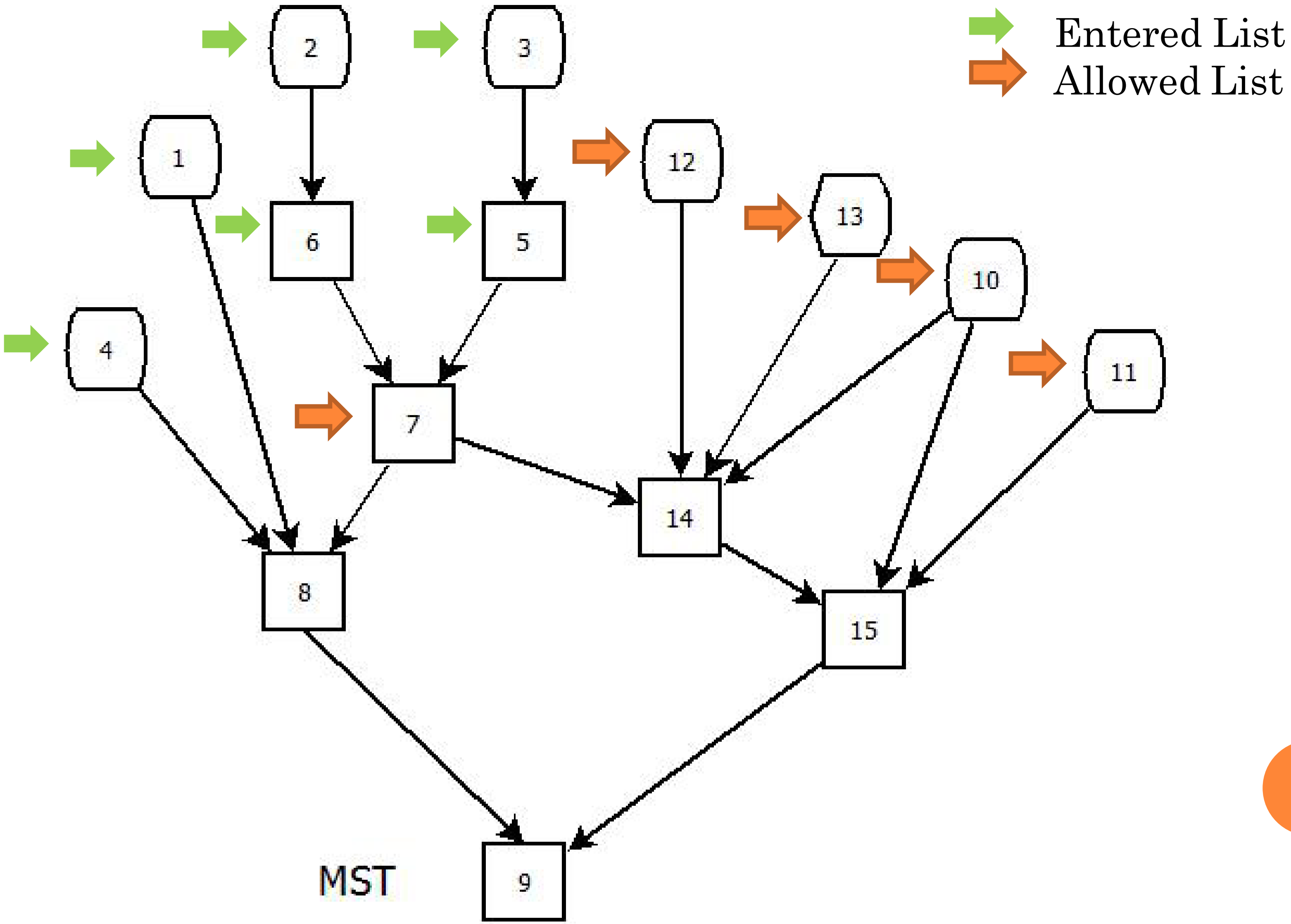
Solution Matching Module

- Reacts to what the student is doing
- Traverses through GST and determines the next course of action
- Invokes Hint generation module when required

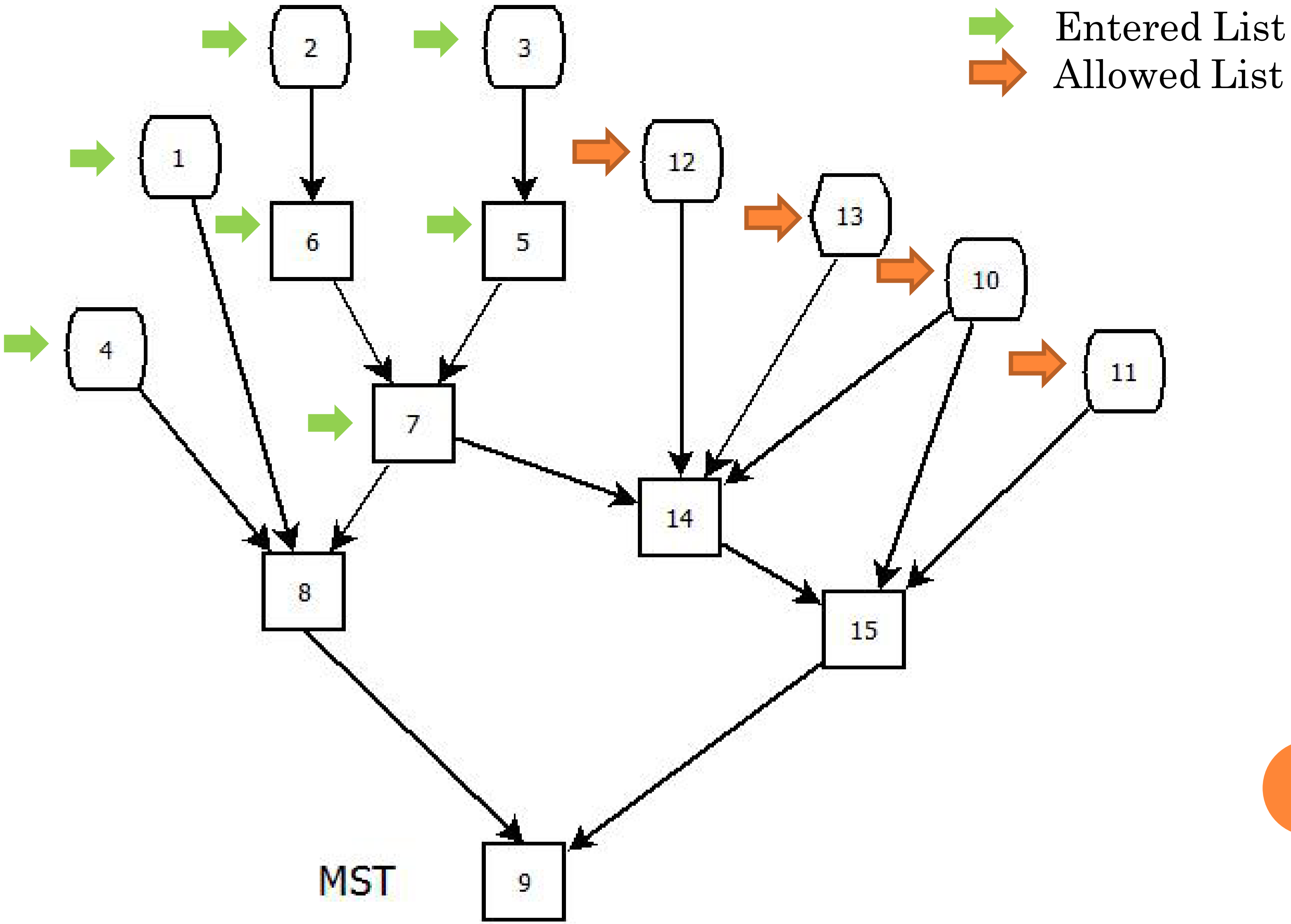
Solution Matching Module

- Solution Matching Algorithm
  - ***this.children()*** – returns an array of all children of a node in GST
  - ***this.parents()*** – returns an array of all parents of a node n in GST
  - ***Entered_list*** – list of all nodes that have been entered as solution steps
  - ***Allowed_list*** – List of all nodes that are valid as a next step
  - ***refreshAssertionStack( )*** – refresh options in assertion stack
  - ***refreshAllowedList*** ( ) – refreshes the allowed list every step

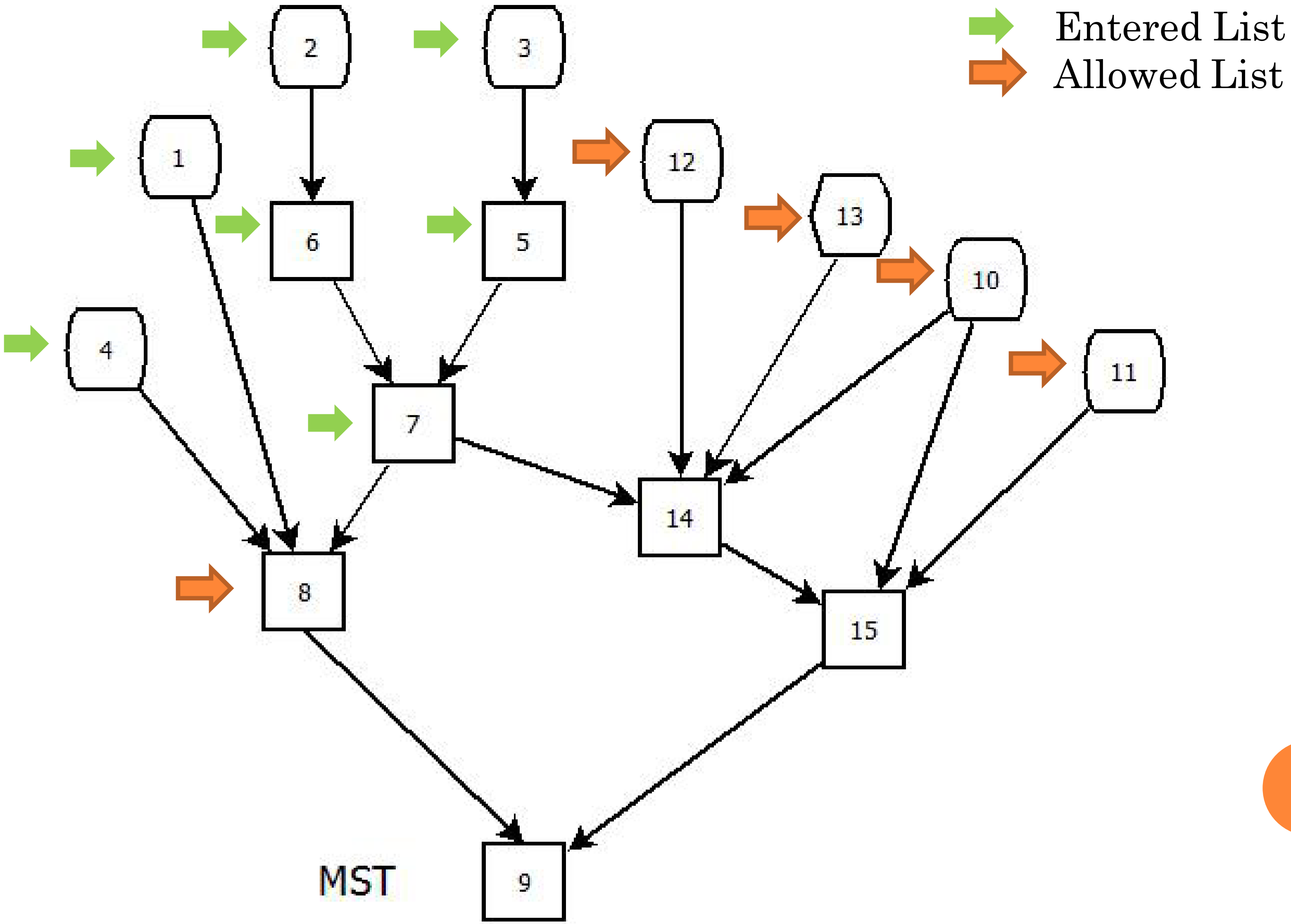
Solution Matching Module



Solution Matching Module



Solution Matching Module



Solution Matching Module

- Algorithm for `refreshAllowedList()`

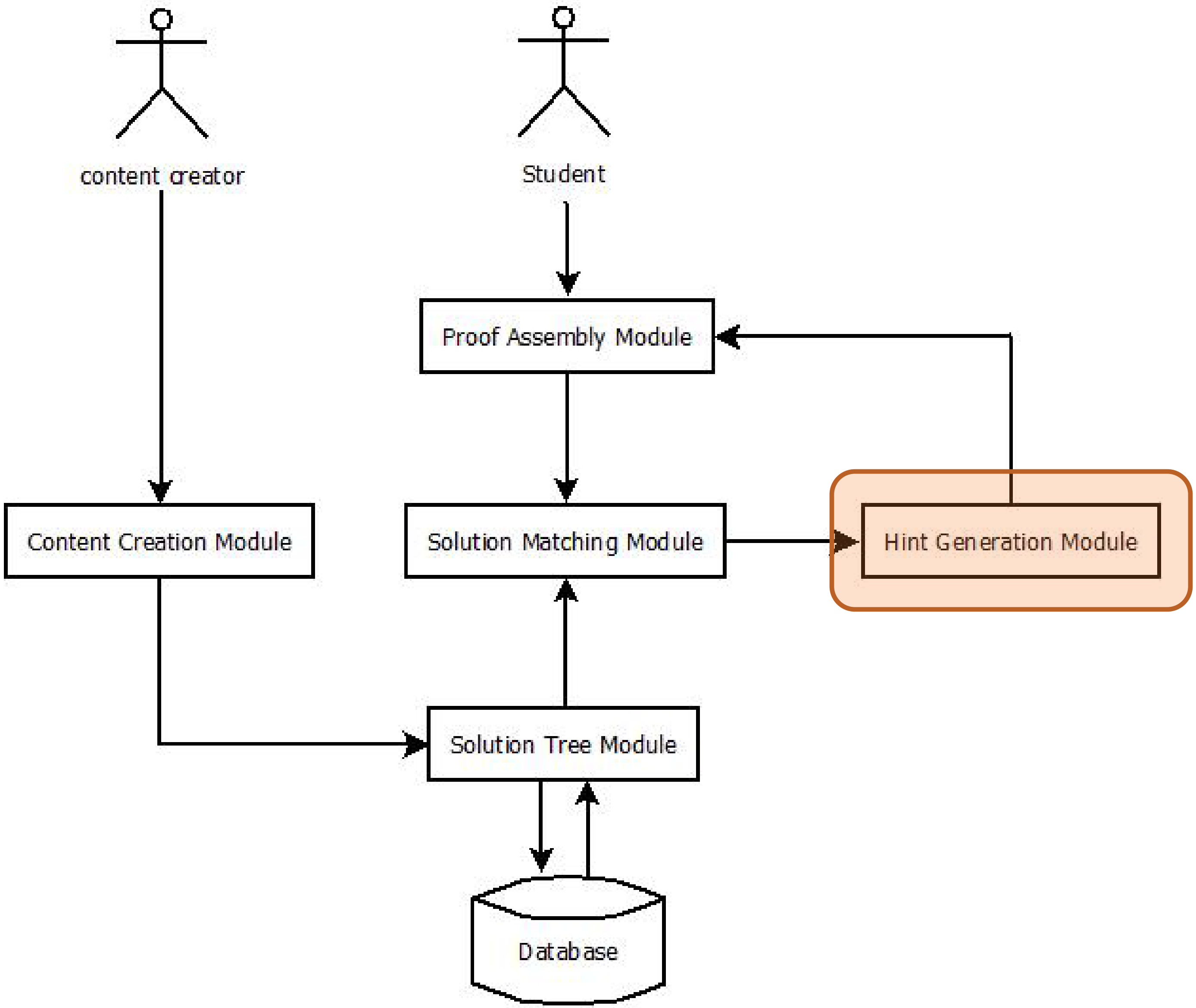
```
refreshAllowedList(n)
foreach child m of n;
do
    if all parents of m are in entered_list then
        add m to allowed_list;
    end
end
```

Solution Matching Module

- Algorithm for `SolutionMatching`

```
if assertion AND reason are correct then
    add node n to entered_list;
    refreshAllowed(n);
    refreshAssertionStack();
else
    callHintGenerationModule();
end
```

Hint Generation Module



Hint Generation Module

- Some ideation:
  - If Assertion is wrong
    - If there is a problem specific hint in the GST, then give that hint
  - If Assertion is correct and reason is wrong
    - Hint could be definition of the wrong selected option
  - If the student presses the hint button
    - Look at the allowed list and point to one of the nodes
  - If the student presses the next step button
    - pick one of the nodes in allowed list that is not a G-node

Hint Generation Module

- ET 801 Course Project
  - Took three standard textbook geometry problems
  - First Iteration
    - Wrote down all possible solutions using domain knowledge
    - Wrong options created for each solution step
  - Second Iteration
    - Extracted patterns in hints and tried to generalize them
  - Third Iteration
    - Wrote down rules for hint generation based on known misconceptions

Hint Generation Module

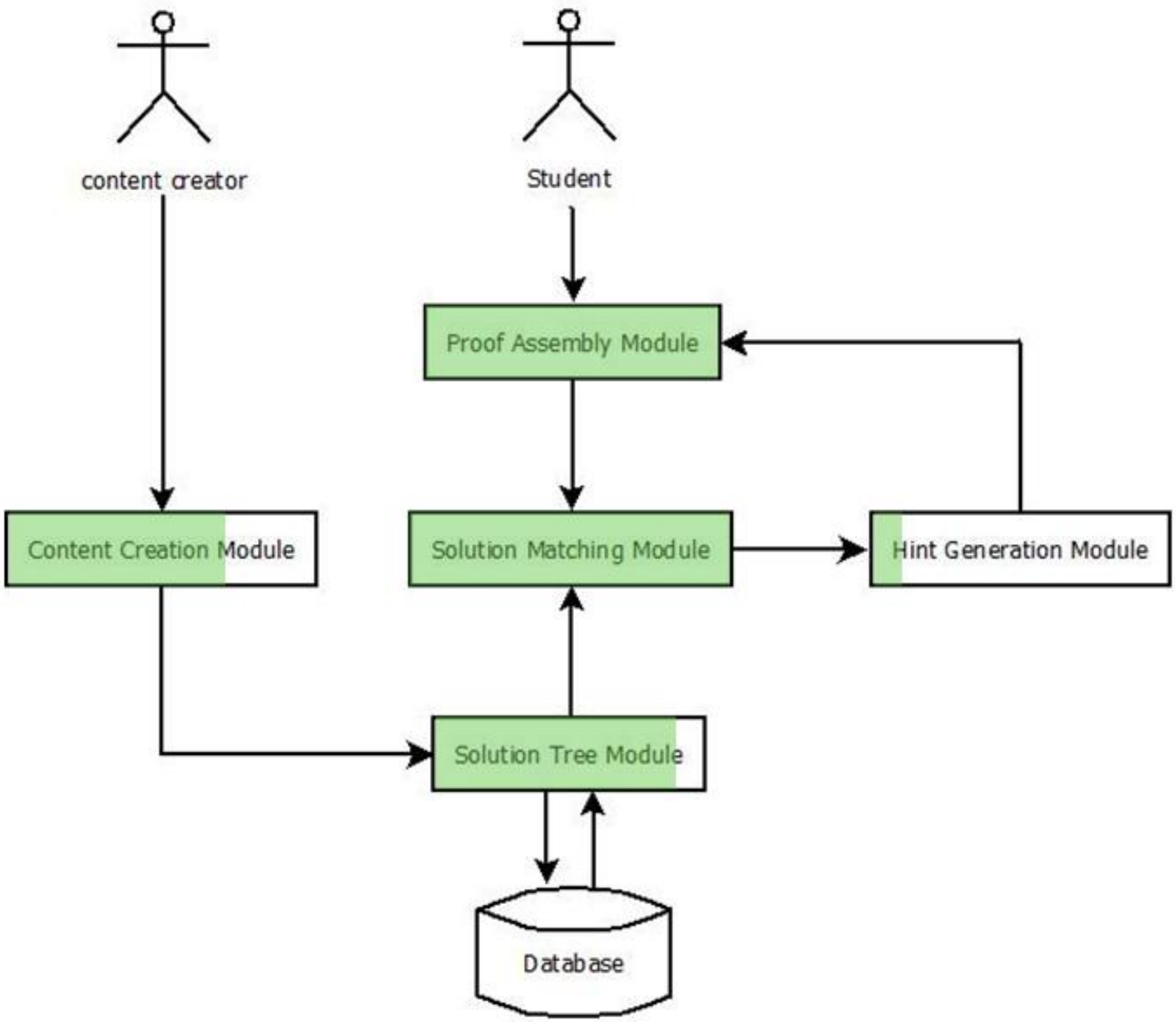
- **Classification of System Responses**
  - ***r0***: Assertion is correct and reason is the correct explanation
  - ***r1***: correct response - proceed to next step
  - ***r2***: you cannot deduce statement with the information you have
  - ***r3***: The opposite sides of a parallelogram are always equal but adjacent sides need not be equal.
  - ***r4***: Reason is not the correct justification for #statement
  - ***r5***: You cannot deduce it from figure alone
  - ***r6***: The statement is correct but it may not be useful in the solution

Hint Generation Module

- **Generated Hints based on Responses**
  - ***h1***: Identify what is given in the question first
  - ***h2***: Observe the pair of triangles and see if you can deduce anything
  - ***h3***: definition of wrong answer
  - ***h4***: You want to prove statement. See if you can deduce it from the information you have.
  - ***h5***: You have already proved statement. Try to use that in the next step.
  - ***h6***: Are you sure you have all the information to make this assertion?
  - ***h7***: study the figure carefully and see if you can choose an assertion
  - ***h8***: Are you sure all the dependencies are accounted for?
  - ***h9***: You have already deduced statement, try to use that result

Future Work

- Interfaces and Hint Generation



Future Work

- Interfaces and Hint Generation

- Evaluation
  - User Experience
  - Learning Objectives

- Integration
  - Student activity logging
  - Mining for new misconceptions

- Expanding the Scope of the System
  - Other areas of mathematics
  - Towards a typing based input

References

1. William Curtis. How to Improve Your Math Grades. Occam Press California, 2538 Milvia St. Berkeley, CA 94704-2611, 2008. chap. 1.
2. David Foster. Assessing Mathematical Proficiency, volume 53. MSRI Publications, 2007.
   Chap. 12.
3. Brian Grossman. Intelligent algebraic tutoring based on student misconceptions. Master's thesis, Massachusetts Institute of Technology, 1996.
4. Adobe Inc. Adobe ex documentation, June 2011.
5. http://www.adobe.com/devnet/ex/documentation.html
6. Berinderjeet Kaur. Some common misconceptions in algebra. Teaching and Learning, 11(2),33-39, 1990.
7. Lindsay M. Keazer. Students' misconceptions in middle school mathematics Master's thesis, Ball State University Muncie, Indiana, 2003.
8. K. R. Koedinger and A. T Corbett. Cognitive tutors: Technology bringing learning science to the classroom. The Cambridge Handbook of the Learning Sciences, 2006.
9. Roblyer M. and Doering A. Integrating Educational Technology into Teaching. Pearson Education, 5th edition, 2009.
10. M. Matz. Towards a process model for high school algebra errors. Intelligent tutoring systems (pp. 25-50), 1982.
11. Robert McCormick. Conceptual and procedural knowledge. International Journal of Technology and Design Education, 7:141{159, 1997.

References

12. http://www.mindspark.in Mindspark
13. Alwyn Olivier. Handling pupils' misconceptions. Mathematics Education for Pre-Service and In-Service, 1992. page 193-209
14. Learn Quebec. Algebra: Some common misconceptions, August 2010. Algebra misconceptions with visuals.
15. J. Ryan and J. Williams. Mathematics 4-15: learning from errors and misconceptions. Open University Press Maidenhead, 2007.
16. World Wide Web Consortium (W3C). www.w3.org/xml, June 2011.
17. www.counton.org. Misconceptions in mathematics, August 2010.
18. http://www.counton.org/resources/misconceptions/.
19. www.toptenreviews.com. Algebra software review.
20. Zhicheng Zhang. Carnegie learning cognitive tutor algebra 1 and geometry followup report. 2007.