**Calculations/Logic for Learning Profile**

* **Capturing Missing concepts and misconceptions**

Learning Mode :

* For selected answer take the hasBug property.Put it in capturedBugs of

User class and update the openBugCount by 1 in Bug class.

* For every correct answer, get all bugs for all hasOption of the question uri.For each Bug if the bug is captured update the Resolved Bug count by 1.

Solving Mode :

* For anywrong input in Given,all bugs associated with given expression is captured in User class and open Bug count is updated.
* When all Given expressions are enetered the resolved bug count for all the captured bugs associated with the given expression is incremented by 1.
* Same logic is followed for ToFind expression.
* For every wrong statement(excluding missing unit,missing rhs) in the solution part the bug for that task is captured and open bug count is updated.
* When the problem is completed two things are done

1) Get all the bugs associated with all the tasks of the problem for the given strategy and check whether the bug is captured in the User class. If captured the resolved bug count is updated by 1.

2) Get all incomplete tasks from the strategy by subtracting all tasks from completed tasks for the given strategy. Capture all bugs associated with incomplete tasks and update the open bug count by 1 for all associated bugs.

**Calculation in learning profile:**

For each bug captured in User class

( Resolved Bug count/(Resolved Bug count + Open Bug count))\*100, rounded off to the nearest integer.

* **Learned Concepts**

Load all completed tasks for the given problem type from the User model,and get the supreclass of the task as taskname. From the tasks obtained get the type of **inverse\_of\_learnedConcepts** property. Then plot the bar graph for this combination i.e taskName ( inverse\_of\_learnedConcept name). Count is done for the combination and not the individual task name.

For ex : FindingInnerRadiusOfRCCyl (Has\_Dimension\_Of\_the\_Object\_Whose\_CrossSection\_Is\_Given)

* **Learning Profile of Mensuration domain**

For all the completed problems of User class get the Mensuration type using ofMensurationType property. Split the type at “\_” where the left hand side is the problem type and right hand side is the mesuration type (cube,cuboid etc).All combination type shapes are merged into CombinationOfDifferentShapes type.The mensuration types are fixed at 7 types

Cube,Hemisphere,RightCircularCone,CombinationOfDifferentShapes,Sphere,Cuboid,RightCircularCylinder.

*So any other mensuration type cannot be dynamically added.*

* **Problem solving progress**

Get all completed problem and it types and difficulty level.For each of this problems for each difficulty level get the percentage of completed tasks to incomplete tasks from the user model.

Calculations

For each problem type

easy\_correct = (sum(completed\_tasks for easy problems)/(sum( completed\_tasks for easy problems) + sum(incomplete\_tasks for easy problem)))\*100

medium\_correct = (sum(completed\_tasks for medium problems)/(sum( completed\_tasks for medium problems) + sum(incomplete\_tasks for medium problem)))\*100

hard\_correct = (sum(completed\_tasks for hard problems)/(sum( completed\_tasks for hard problems) + sum(incomplete\_tasks for hard problem)))\*100

Overall = (easy\_correct + medium\_correct + hard\_correct)/3

All percentages are rounded off to nearest decimel places.

* **Activity**

On first opening the learning mode or for a problem the learningStartTime or solvingStartTime timestamp is entered in the ontology for the given problem.Similarly on learning and solving completion of the problem the timestamp learningEndTime and solvingEndTime timestamp is updated.The graph is plotted for the time taken from startTime to endTime for Learning and solving mode in seconds.