CURE-ICULUM: DEMYSTIFYING THE COURSE DESIGN PROCESS

Daniel Schoepflin

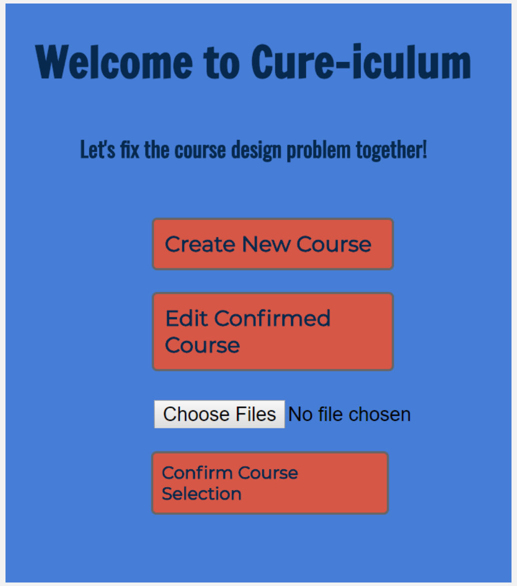
June 6, 2018

CURE-ICULUM DESCRIPTION

- High Level Description: Cure-iculum uses a node-link representation of graphs for the development and "sharing" of courses. By connecting highest level nodes (the main course topics) with each other, users can create dependencies which Cure-iculum uses to create a valid schedule. Subgraphs "within" the course topics can correspond to individual learning goals which can be associated if they are related. Assignments can then be developed and tagged with these goals so that users can track coverage of these goals and proficiency when the results are entered into the software.
- Intended Users: Faculty members at any level of institution
- Goals: Replace the ad hoc methods of developing courses and tracking student performance across classes and within a class.

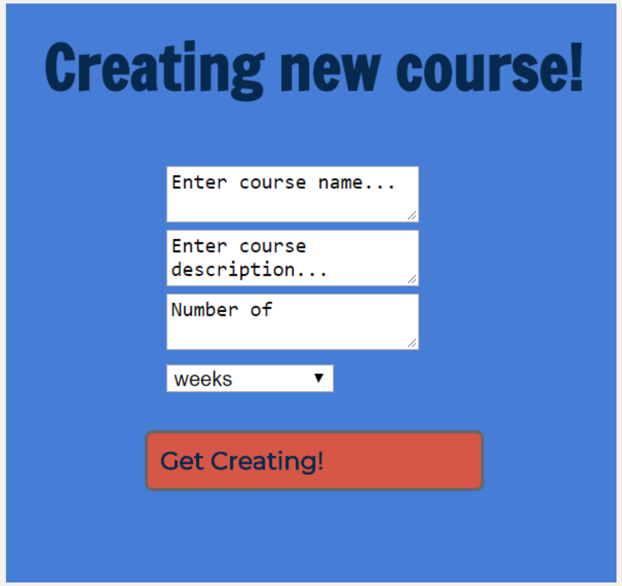
IMPLEMENTATION – CREATION FLOW

- The course is maintained as a Javascript object and is passed from page to page as a JSON string in the sessionStorage of the page
- Maintaining the object as a JSON string allows for easy import and export so that courses can be edited in future sessions without needing to keep a server available for all users. It also allows sharing from educator to educator
- The flow on creation of a course and ultimate exporting is shown below

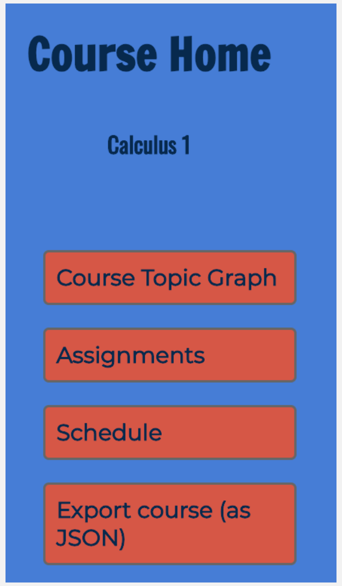


Welcome to Cure-iculum

Let's fix the course design problem together!

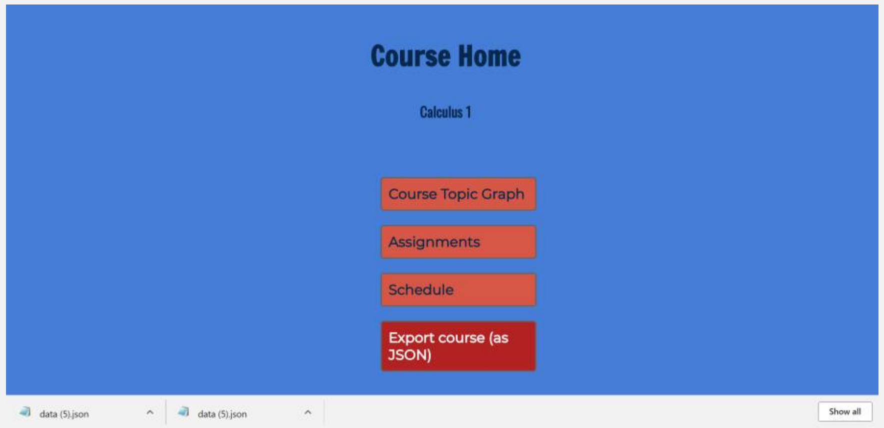


Creating new course!



Course Home

Calculus 1

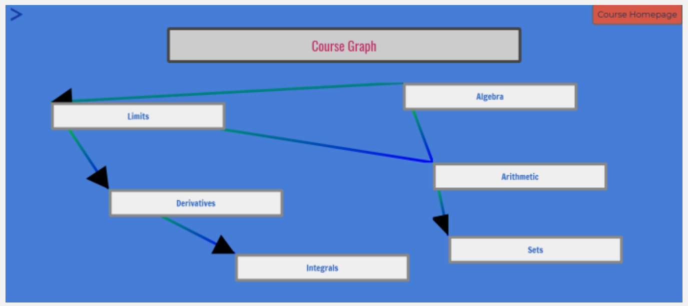


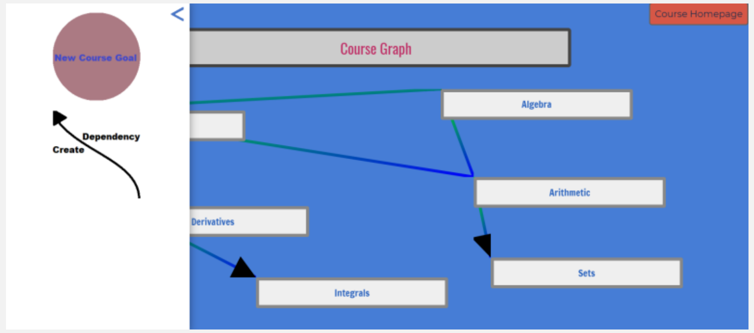
Course Home

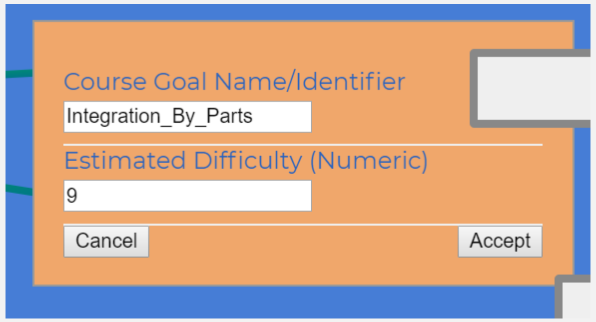
Calculus 1

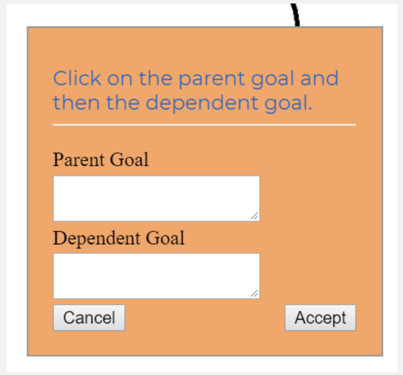
IMPLEMENTATION – THE COURSE GRAPH

- The application centers around the course graph, where users define learning goals for a class and create dependencies between them. The <div> elements representing the courses are generated on demand and the arrows connecting them are <svg> elements with gradients and marker heads indicating direction
- A course graph being built can be seen on left and the open sidebar allowing for creation of courses and dependencies (whose dialogs are also in images below) can be seen on middle-left.
- Ultimately, all of these courses are stored as vertices in a graph and the dependencies as edges connecting them. This allows the creation of a "suggested schedule" based on the number of available class sessions, estimated difficulty of each course, and dependencies
- The topic graph for a course goal can be entered by double clicking on any course goal. This associates the topics with a particular goal.



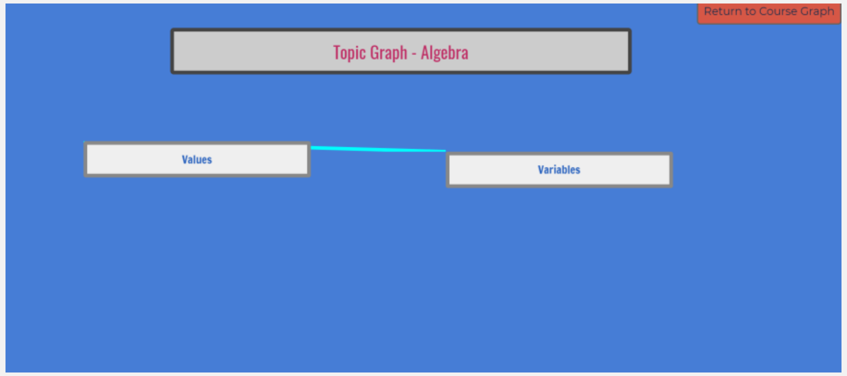


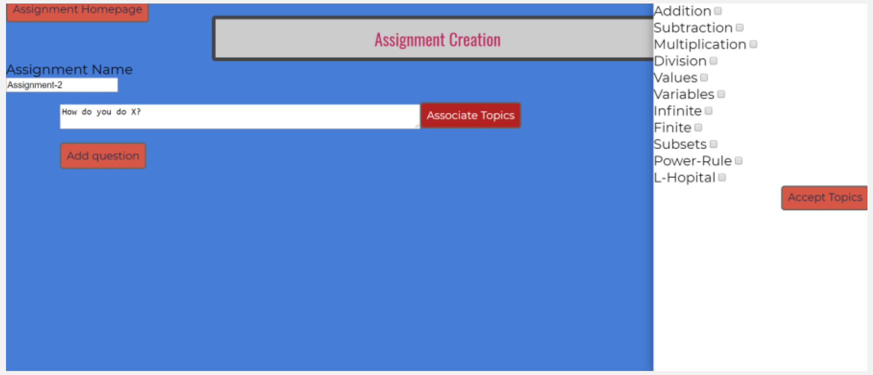

Course Goal Name/Identifier
Integration_By_Parts
Estimated Difficulty (Numeric)
9

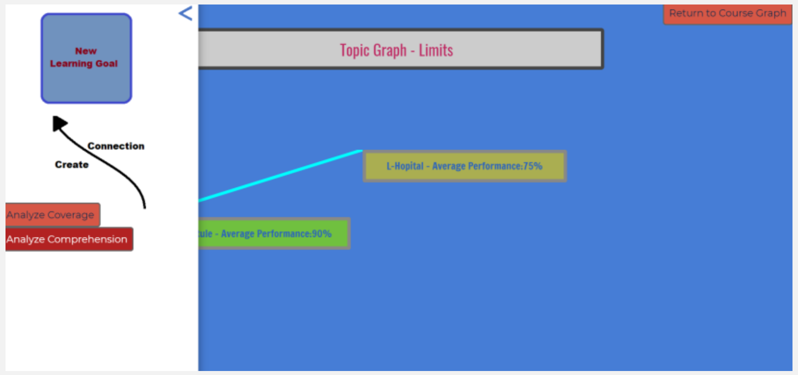

Click on the parent goal and then the dependent goal.
Parent Goal
Dependent Goal

IMPLEMENTATION – ADDITIONAL FEATURES AND USE

- Users can create assignments from the assignments homepage and while making them, can associate topics in the topic graphs with individual questions. This can be seen in the three leftmost figures below.
- Using these associations, after an assignment has been saved, a user can check coverage of different topics from within topic graphs with the topics "changing color" and displaying numerical data in order to cue users to how many questions have been asked related to that topic.
- Further, a user can return and enter the results from an assignment and by returning to the topic page, a user can track roughly how well the comprehension of the students is for a particular topic in real time. The average score for each topic updating with every graded question. This feature can be seen in the image on the bottom right.
- The hope is that this tool can complement the standard methods for in class maintenance (i.e. Blackboard, Moodle) by allowing for development of new courses more easily


Return to Course Graph
Topic Graph - Algebra
Values
Variables


Assignment Homepage
Assignment Creation
Assignment Name
Assignment-2
How do you do X?
Associate Topics
Add question
Addition
Subtraction
Multiplication
Division
Values
Variables
Infinite
Finite
Subsets
Power-Rule
L-Hopital
Accept Topics


New Learning Goal
Connection
Create
Analyze Coverage
Analyze Comprehension
Topic Graph - Limits
Return to Course Graph
L-Hopital - Average Performance:75%
ule - Average Performance:90%