Forum Feature in Dragoon

Table of Content

Introduction: 3

Requirements: 3

Dependencies: 3

Implementation: 4

Test Plan: 5

Timeline: 6

## Introduction:

Forums are introduced in order to allow students and author to interact with each other, encourage more engagement in modeling activity, which will help in creating a better learning environment. While working on already authored model, students will be able to write comments about the way author has created this model. Student’s comment may be agreement or disagreement about the model or may be about a better way to model the same Systems Dynamics.

Forums will also allow authors to receive feedback about their model and later incorporate changes to their authored models. Authors will also be able to post response comments which will be seen my all the student’s working on that particular model.

~~Based on the way user management is setup in LAITS, authors and students may also receive email notifications when someone posts a comment on the model (the author of the model and the students who have posted a comment on the model).~~ [Move to Roadmap section]

Target date for deployment is August 30, for Dan Childer's class.

[Not sure where this goes in the document, but need a concept of a “section” or “group” of students. The messages generated by members of a section would only be visible to other members of that section.]

## Requirements:

1. User (Student or Author) should be able to post comments at the granularity of a specific Node or about the complete mode. Node level comments will be shown in a separate tab of Node Editor (called Discussion Tab). Model level comments will be shown on a separate dialog box.
2. User should be able to view earlier comments posted by other users in a scrollable window. ~~Initially, only a default number of recent comments will be loaded, and a button will be provided to load older comments as well.~~ [Move to Roadmap section]
3. User should receive notifications when other users post comments on the models, via some non-blocking flag on the user interface. Notification outside of the application itself is listed in the Roadmap. (For students, precondition is that student has commented on this model. Author will always receive notification if another use has posted a comment on their authored model).
4. ~~Forum component will be configurable, admin will be able to enable or disable forum for a particular section or model.~~ [This is too vague. Also, I don’t think that it is useful to refer to ‘admin’ in this document since that is not a defined concept in our software. Rather, talk about section-wide switches stored in server database. Moved section-wide disable to Roadmap section (may want to revisit this?).]
5. ~~All the models that are provided by default in LAITS, admin will be considered as author of those models. Admin will receive notifications when Users post comments on these models.~~ [Again, there is no such thing as an admin. See item on Roadmap.]
6. ~~User should be able to delete the comments posted by them earlier.~~ [This is hard to do and is not a feature of any chat system I know of.]
7. The problem xml may define initial comments to populate the chat windows.
8. The chat windows should be disabled in TEST mode.

## Dependencies:

Discussion forum feature is dependent on two major UI components:

1. **LAITS Main Window:** Forum that is pertinent to complete model will be opened using either “Show Forum” button on the toolbar or “Show Forum” submenu under model menu. Clicking on any of these buttons will open a dialog box with Discussion Forum on it. This dialog box will be a model window (with parent as main GUI of LAITS), so user will need to close this before starting any other action.
2. **Node Editor:** Forum that is specific to a node will be shown as another tab in the Node Editor window. Node Editor needs to be updated in order to add this functionality. The forum tab in the node editor will be disabled until the user defines a valid quantity in the first tab.

## Implementation:

The main implementation difficulty is that we need something that is between a “chat window” (synchronous communication) and a “forum” (asynchronous communication). Chats have the concept of a “session” which individual users can join or leave, while forums have the concept of a “thread” but don’t really have a concept of joining or leaving, or of closed conversations.

The forum communication may be implemented via COMET. In that casae, we will have to identify a suitable COMET server (such as jetty), and install it on dragoon.asu.edu, and include installation instructions in our documentation. There are java libraries that provide interfaces to COMET.

1. **Forum data storage:** In order to store the comments posted by authors and students, we need permanent storage. We have two choices here:
   1. ~~Use a third party discussion forum: This will require connecting to their system using an API and will certainly require some registration (username, email, passwords etc.) for each user. This will have inherent complexities from the provider as well. Apart from this, it will require Internet connection if forum is not installed and managed by us.~~
   2. ~~Since, we only need to store comments by users, we can create a table in our own database. We will identify comments based on usernames used while launching LAITS.~~

[Any COMET server or similar technology will have some sort of logging mechanism]

1. **GUI:** I have developed a mock interface for discussion forum. Following are the images:

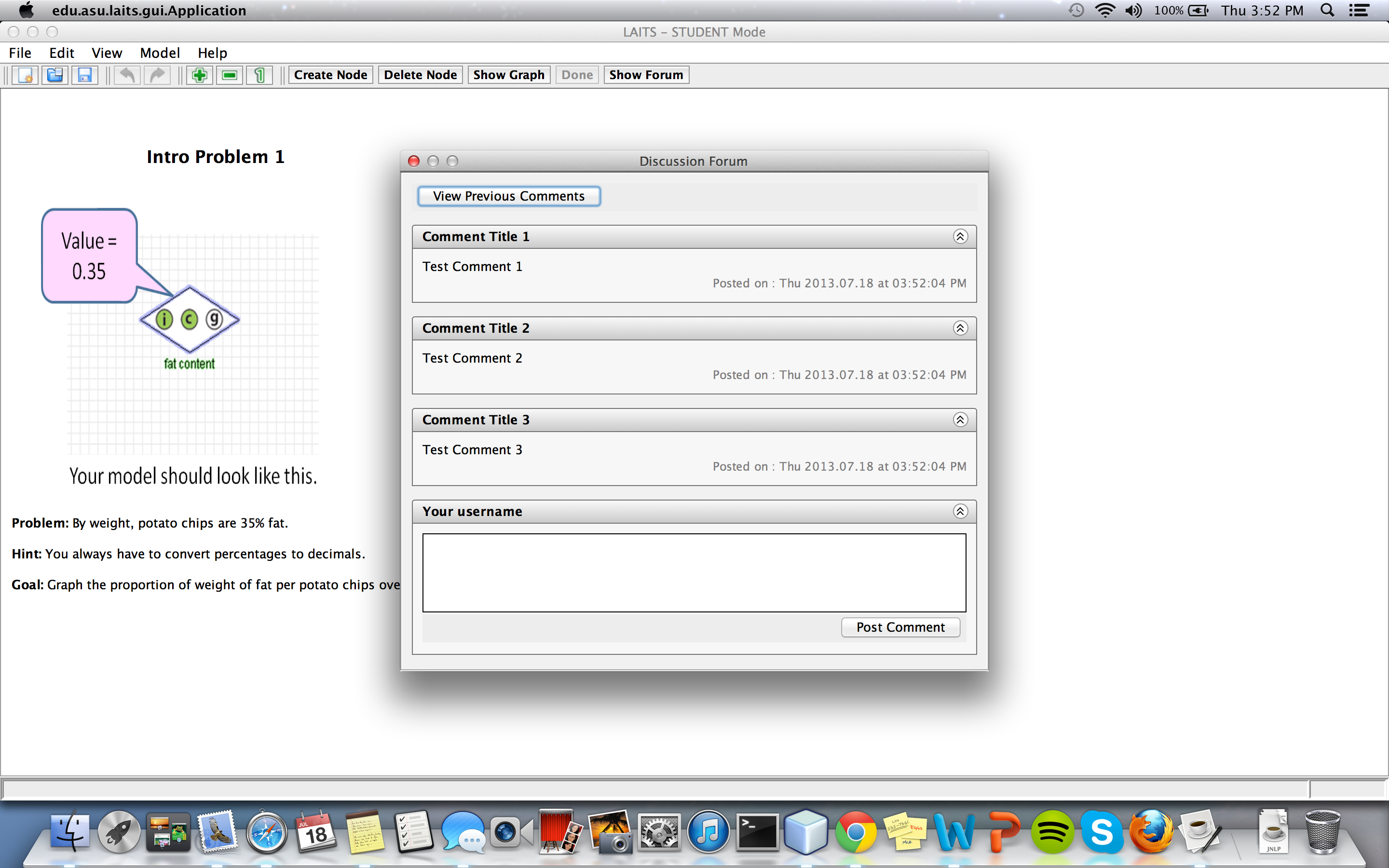


Figure : Discussion Forum with LAITS

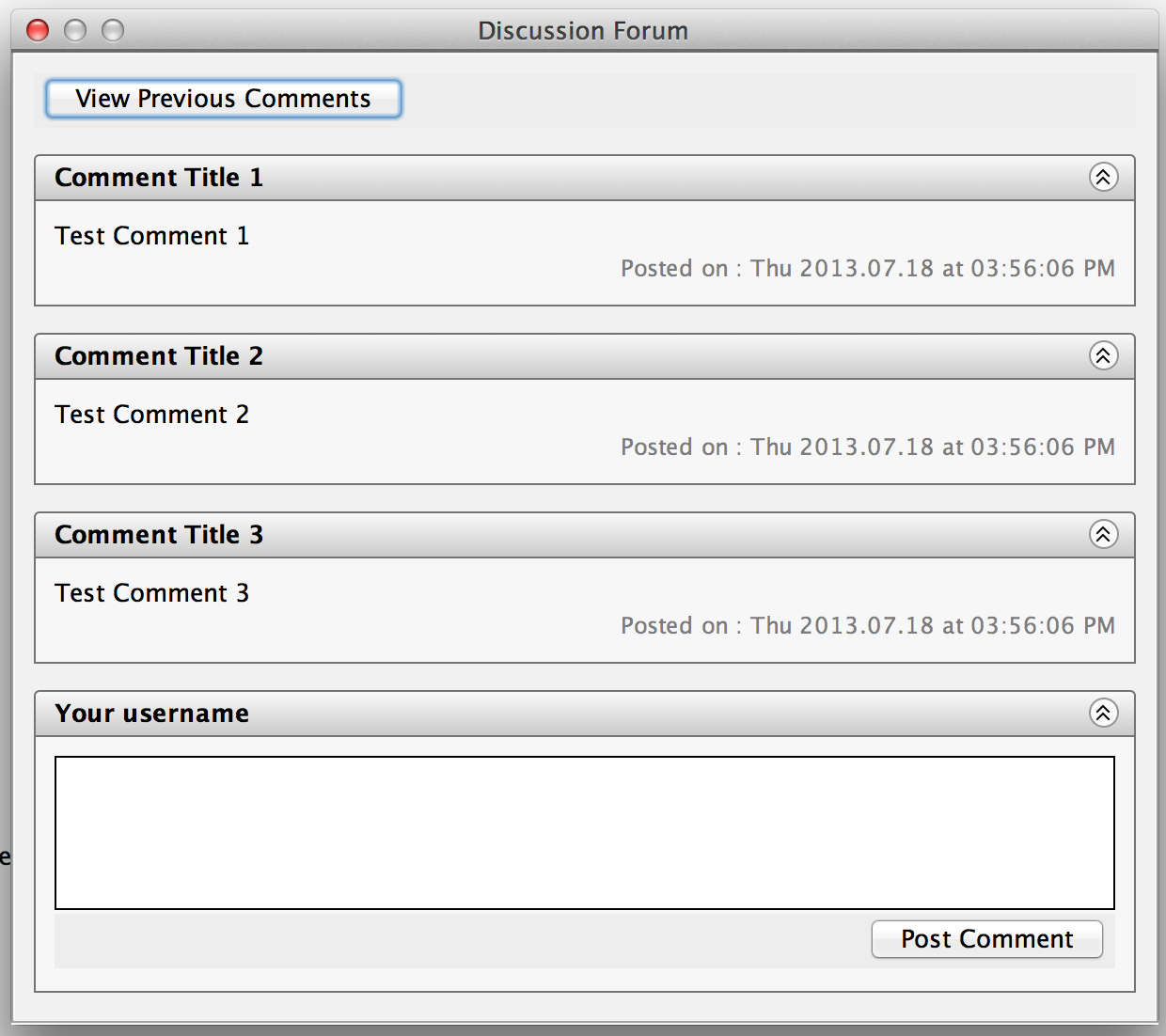


Figure : Discussion Form Panel

[The Skype chat window looks pretty close to what we want. Previous comments should be scrollable. Remove “View Previous Comments” since that is a roadmap item.

## Test Plan:

Testing of this feature will be manual. Separate test cases will be added to the online test sheet. Main focus will be:

1. User should be able to post comment and see it reflected in the form window immediately.
2. Posted comments should be shown, when any user opens the comment window after the comments are posted. [Only for users in the same section.]
3. Posting a comment should send notification to author and other users who have posted comment in this node/model. [Only for users in the same section.]
4. ~~Initially, only a default number of latest comments are loaded in the discussion forum window.~~
5. ~~View Previous Comments button should fetch a default number of older comments (ordered by date/time).~~
6. Forum window is scrollable.
7. ~~Individual comment windows can be collapsed.~~ [Move to roadmap]
8. ~~Users should be able to delete their comments. There should be a delete comment button for each comment posted by current user.~~ [This would be hard to do (for instance, do I delete from database, too?) and does not correspond to any existing chat I know of.]
9. If a student closes a problem and re-opens it, the chat windows contain conversations from the previous session, plus any intervening messages that the student should have access to.
10. Chat history is preserved under server restarts.

## Timeline:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.** | **Task** | **Deadline** | **Comments** |
| 1. | Finalize requirements | Monday 22nd | Implementation of essential features will be carried out in parallel |
| 2. | ~~Design data model and create tables in the database~~ Deploy comet server on devel server, and connect to database |  |  |
| 3. | Develop and Integrate UI in Main Window and Node Editor |  |  |
| 4. | Implement actions for all the events |  |  |
| 5. | Unit Testing and Integration |  |  |
| 6. | Integration and System Testing |  |  |
| 7. | Deployment to production server |  |  |

[Add section “Roadmap” that included features that we want but may be completed after the target date. This includes:

1. E-mail notifications to authors and section members about chat updates
2. Do not show very old messages from other section members. In that case, include “show more” button or link at the beginning of the scrolling area.
3. Have a section-wide switch to turn off access to chats generated by other members of the section, but not the author. Value for switch stored on in database.
4. There should be an interface for developers/researchers to view all comments, with associated filters for problems, sections, and dates.
5. Individual comments can be collapsed (I think this is low priority).
6. Instructors have method for removing individual comments.

]