LASAD Grammar Modifications

Developer Guide

Author: Simon Sun (simonthesun@yahoo.com)
Updated August 27, 2016

NOTE: This grammar app is based on the LASAD codebase; please also refer to the LASAD documentation, as well as the existing desktop-only grammar app. Also, this does not include some of Judy's changes….

Contents:

List of Modified Files

Located in LASAD-Grammar/Develop

LASAD-Client/src/lasad/gwt:

client.communication.LASADActionReceiver
client.communication.helper.ActionFactory
client.model.GraphMapInfo
client.model.organization.ArgumentModel
client.model.organization.GrammarNode
client.model.organization.NodeCreator
client.ui.box.AbstractBox
client.ui.common.elements.AbstractExtendedAwarenessElement
client.ui.common.pattern.elements.ExtendedAwarenessElementPattern
client.ui.workspace.argumentmap.ArgumentMapMenuBar
client.ui.workspace.argumentmap.ArgumentMapMVCViewSession
client.ui.workspace.argumentmap.CreateNewMapDialog
client.ui.workspace.graphmap.AbstractGraphMap
client.ui.workspace.tabs.authoring.steps.CreateModifyAndDeleteOntology

LASAD-Server/src/lasad:

controller.ManagementController
controller.MapController
entity.Element
helper.ActionPackageFactory
logging.commonformat.Action2CFTranslator
logging.commonformat.util.ActionTranslatorHelper
processors.ActionProcessor
processors.specific.FeedbackActionProcessor
processors.specific.LoginActionProcessor
processors.specific.ManagementActionProcessor
processors.specific.MapActionProcessor

LASAD-Shared/src/lasad:

shared.communication.objects.commands.Commands

Important Changes From LASAD

(not comprehensive)

LASADActionReceiver

Added code to `processMapAction()` to add the form or the function to the ArgumentModel nodes if/when a certain type of box is being created.

ActionFactory

Created the new method `removeAllElements()` to clear the canvas, used when creating a new sentence in CreateNewMapDialog.

Created a separate `createBoxElementsAction()` method with a text argument to be used when creating a box with predefined text content (notably when creating a new sentence). Kept the original `createBoxElementsAction()` without the text argument so as to not break other LASAD code that relied on it. Similarly, created a separate `createBoxWithElements()` method with a text argument.

Created the new method `createBoxesWithElements()` to create multiple boxes with text at once in a defined order, used in creating a new sentence in CreateNewMapDialog. Necessary to prevent words from appearing out of order when sending each word individually due to server latency.

Created the new methods `createLinkSet()` and `createBoxAndLinks()` to handle the creation of multiple links simultaneously to bypass the server limitations. Used in the creation of nodes in NodeCreator and uses the new CreateElementSet command.

Created the new method `createFormInsert()` to handle the specific case of creating a node under a supernode, again to bypass server-client communication limitations.

ArgumentModel

Added functionality for the GrammarNode object to be stored in the model. Notable additions include `addNode()`, `removeNode()`, and `getNodes()`, as well a vector instance variable `nodes` to store the GrammarNodes in each model.

GrammarNode

New file. Object used to store data about each node. Uses a tree-like structure.

Each instance of GrammarNode has two LinkedBoxes, `form` and `function`, two Vectors, `words` used to store the LinkedBoxes of the sentence words that are directly under the node and `subNodes` used to store other instances of GrammarNode that fall under the particular node (tree-like structure), and a boolean `selected` that is used to detect which nodes have been, well, selected.

NodeCreator

New file. Used to handle node creation/deletion. See the “Node Creation” and “Node Deletion” sections in “Implemented Functionalities.”

AbstractBox

Various visual changes.

Modified `setComponentHandling()` to remove the `onDoubleClick()`, `onMouseOver()`, and `onMouseOut()` behaviors, as well as to implement selection functionality to `onClick()`.

ArgumentMapMenuBar

Removed the “LASAD” and “Help” menu options, as well as various MenuItems.

Created a new MenuItem `newItem`, handled in the new method `createNewItem()`. Used to create a new sentence and triggers an instance of CreateNewMapDialog.

Created a new MenuItem `createNodeItem`, handled in the new method `createCreateNodeItem()`. Used to create a node upon making selections, handled in an instance of NodeCreator.

CreateNewMapDialog

New file. Handles the creation of new sentences. See the “Sentence Creation” section in “Implemented Functionalities.”

AbstractGraphMap

Added an instance of NodeCreator as well as a `getNodeCreator()` method.

MapActionProcessor

Added processing method `processCreateElementSet()` to handle the CreateElementSet command, as well as added a case for the command in `processAction()`. Triggered server-side through ActionProcessor when the new CreateElementSet command is sent from the client.

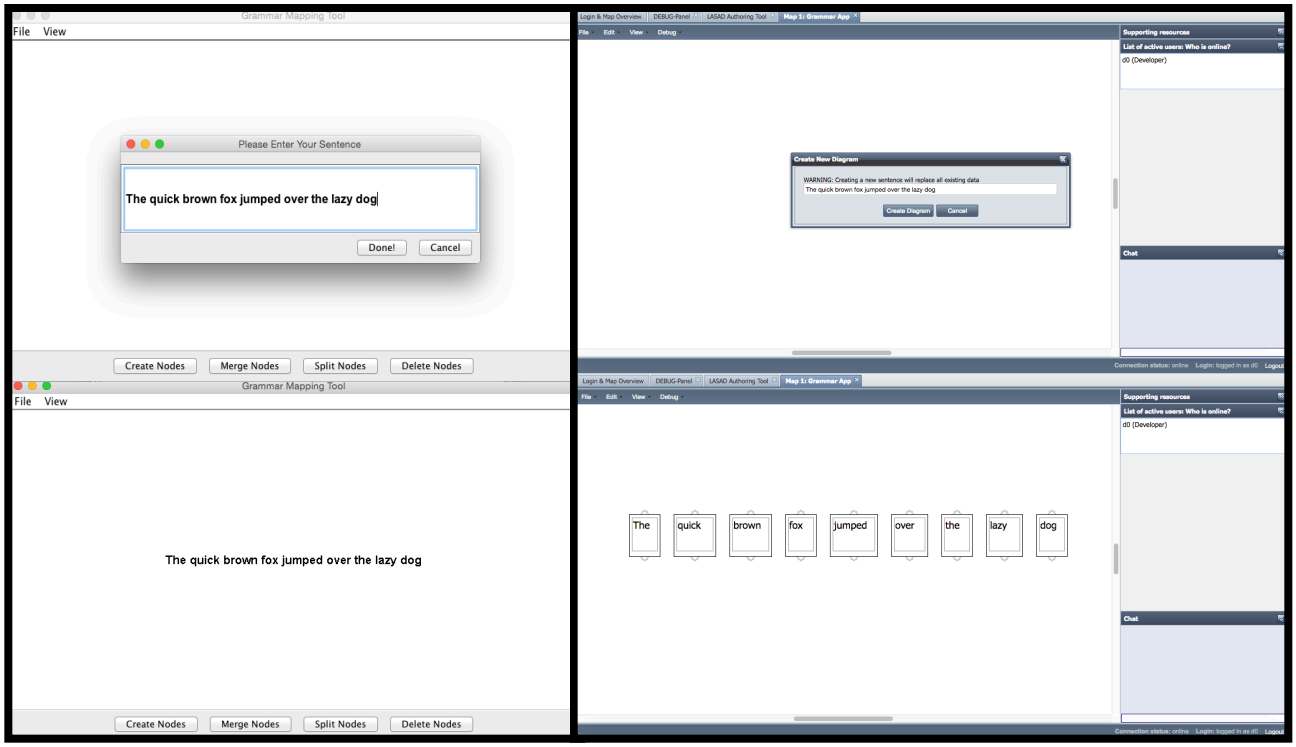
Added handling of the “LAST-ID” placeholder to `processUpdateElement()`. Used in `createFormInsert()` in ActionFactory.

Commands

Added the CreateElementSet command.

Implemented Functionalities

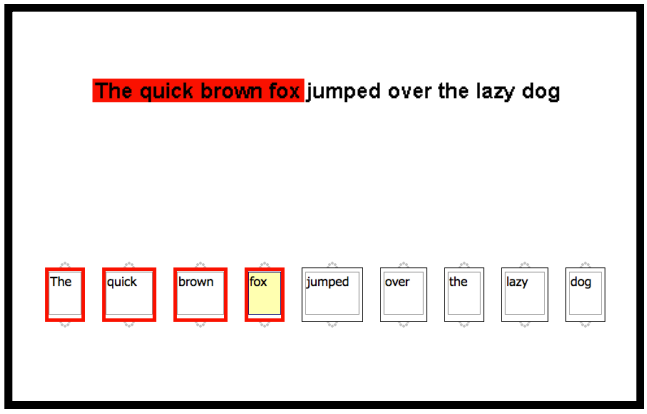
Sentence Creation



Sentence creation in the original grammar app (left) and the LASAD-based app (right)

Modifications to the menu bar through ArgumentMapMenuBar create a new “New Sentence” option in the “File” dropdown. Method `createNewItem()` has a selection listener that creates an instance of CreateNewMapDialog, which contains the bulk of the code to handle sentence creation in `createForm()` within the selection listener of the button `btnDone`.

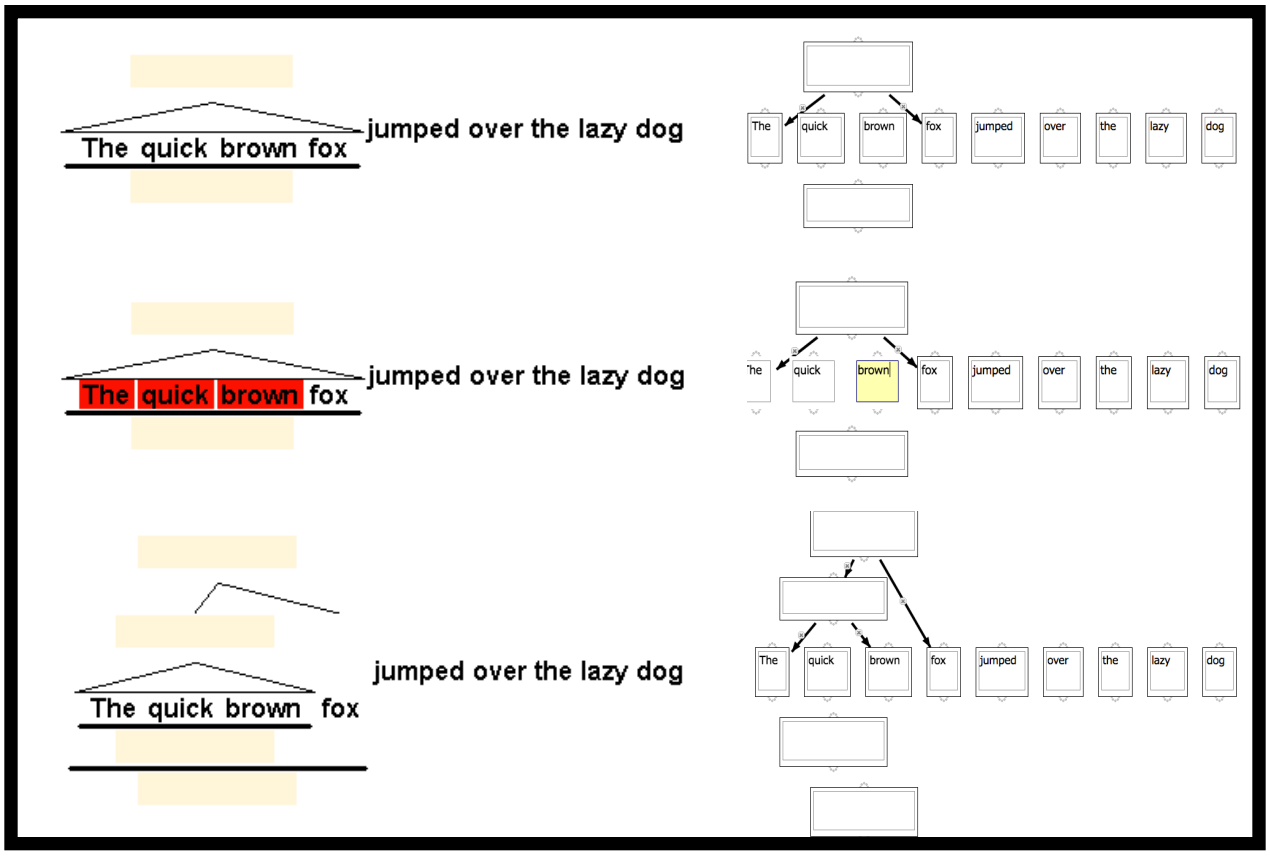
Element Selection



Word selection in the original grammar app (top) and in the LASAD-based app (bottom)

The on-click behavior of the boxes is handled in AbstractBox within `onClick()`. There is currently a bug where the first time a box is clicked, it demonstrates intended behavior and highlights the borders, but any subsequent selections will unfortunately make the border disappear. Additionally, the visuals for entire nodes has not yet been implemented (currently just selects the form and function of the node).

Node Creation



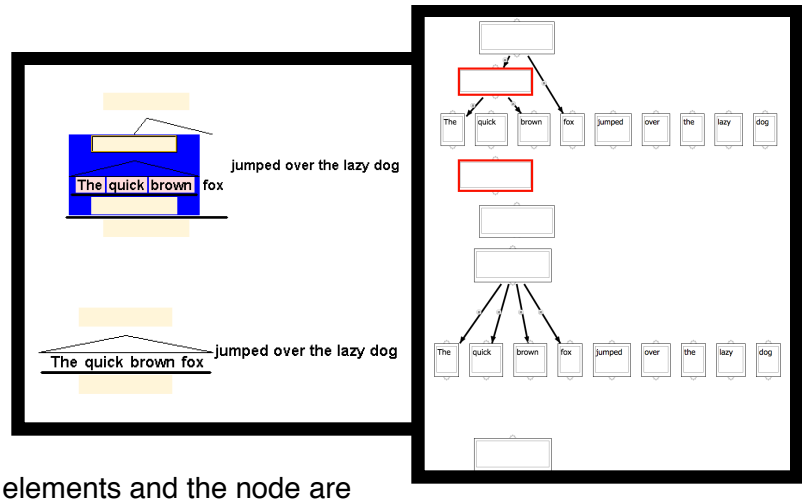
Node creation in the original grammar app (left) and in the LASAD-based app (right)

Most of node creation is handled in NodeCreator, aside from the MenuItem (handled in ArgumentMapMenuBar). The algorithm is a bit involved, but the basic premise is

1) Test if the selected nodes and/or words are adjacent. If not, recursively run this algorithm on each adjacent "group."
2) Test if the selected nodes and/or words are *directly* under the same node. If not, recursively run this algorithm on each "grouping" under the same node.
3) If the selected words/nodes are not under a node, create the node and the visual elements (boxes and links).
4) If the selected words/nodes are under a node, move all nodes above the selection outward, delete all links to the selected words/nodes, create the node and visual elements, then re-link the previous supernode to the newly created node.

The code might get a bit messy here (sorry). Also, note that several other changes had to be made to accommodate this feature. Specifically, the CreateElementSet command had to be made, as well as corresponding server-side handlers for it and ActionFactory calls to it, to bypass the limitation that sometimes box/element data needed to be retrieved from elements that were not yet created server-side at the time of the call (due to server latency).

Node Deletion



Node deletion in the original grammar app (left) and in the LASAD-based app (right)

Node deletion is also handled in NodeCreator and is much less involved than node creation. Because each call of `deleteNodes()` iterates the algorithm on each selected node, it can be assumed that the algorithm only will deal with one node. If the selected node has no supernode, then the visual elements and the node are simply deleted. If there is a supernode, then the elements and node are deleted, the supernode is linked to the deleted node's subelements, and the other node elements may be moved inward.

To-Do List

Things that need to be done to fully re-implement the grammar app in LASAD; may not be comprehensive

---

Features to Implement

Reduce box sizes, esp. vertically

Remove box outlines

Highlight entire node when selected

Properly implement the “divided” node algorithm

Fix/implement box movements upon node creation/deletion

Administrative features such as the creation of login credentials

Change appearance of links

Bugs to Squash

Highlighting bug where the border no longer gets highlighted after one selection, instead just disappearing altogether when selected

Links not disappearing visually sometimes when creating an “inserted” node

Boxes not appearing sometimes visually when creating an “inserted” node

“Divided” node algorithm functioning oddly

Can only have one map ever due to a static map name in the automatic map creation

Any further bugs that come up (they never end…)