Visualization for Process Guidance

Mario Barrenechea, LASER
mbarrene@cs.umass.edu

Background

Background

- Human-intensive processes may be very involved and complex (i.e. diverse agents with diverse tasks, lots of resources, concurrent and exceptional flow, etc)
  - Interruptions do happen, and synchronization among agents needs to occur.

- Exploring whether guiding the process performer can help reduce errors and improve efficiency
  - Show process progress: past, current, and future tasks
  - Address these problems associated with human-intensive processes.

Process Guidance

- Provides support as the user performs each task in the process:
  - Visually represent past, impending, and short-term tasks through a user-interface
  - Should help the user stay on track if interruptions do occur
- Clearly represent the different kinds of process control flow:
  - Parallel (process steps can be performed simultaneously)
  - Choice (exactly one alternative must be chosen)
  - Sequential (steps are performed one after another in order)
  - Try (alternatives tried in order until one succeeds)

Design Goals for Process Guidance Visualization

- Goals for the design:
  - Support end-user provisioning of process steps
  - Maintain changeability of UI events on the screen (stability)
  - Visualize any Little-JIL process definition
- Goals for medical safety:
  - Elicit feedback from domain experts (nurses and physicians) about the user-interface
  - Build UI prototypes to try out in future experimental studies

Design Approach

- Process steps shown as rectangular bubbles
- Color-coding scheme indicates step state
- Layout of the steps corresponds to the order in which the steps should be performed.
- Indentation lines represent parent-child relationships from Little-JIL models.

Demonstration of UI Design Mockups

- Perform Pre-Release Checks (Sequential, Parallel, Exceptional Flow)
  - https://gomockingbird.com/mockingbird/#7dtfm04/OF72gc
- Perform Pre-Infusion Work + Confirm Product Label Matches Product Tag (Multiple-Patient Example)
  - https://gomockingbird.com/mockingbird/#7dtfm04/TUSJJu

Discussion of Issues

- One or more views for the process performer
  - Show as one entire view with process steps
    - But view can have too many steps! → too much indentation.
    - Make view scrollable and/or zooming in/out.
  - Break up process into multiple views
    - Allow for "window management"
    - Allow process performer to manage view windows when performing a step with many sub steps

More Issues

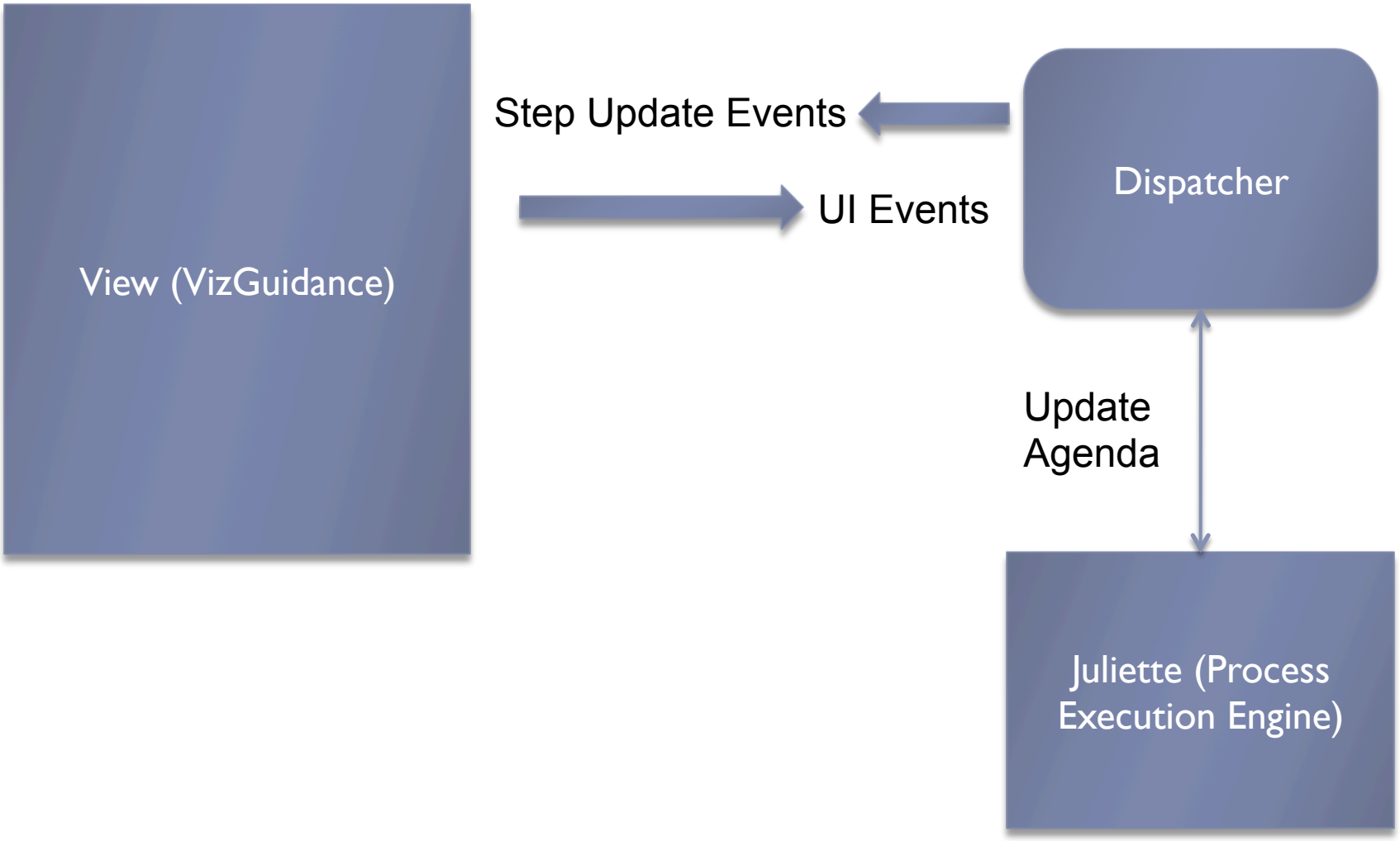
- Distinguishing **Exceptional** flow from **Normal** flow
  - Use a different color for exceptional steps?
  - Use a special symbol for exceptional steps?
  - Have an extra view dedicated to handling the exception?
- Color Coding for Process Step States
  - Current colors seem somewhat intuitive (but probably not to everyone)
  - Color selection will be customizable
- Orientation of process steps based on step kind
  - Can the process performer identify when to work on multiple steps at once? (parallel process steps)
  - Try steps? Sequential steps? Choice steps?

VizGuidance - Implementation

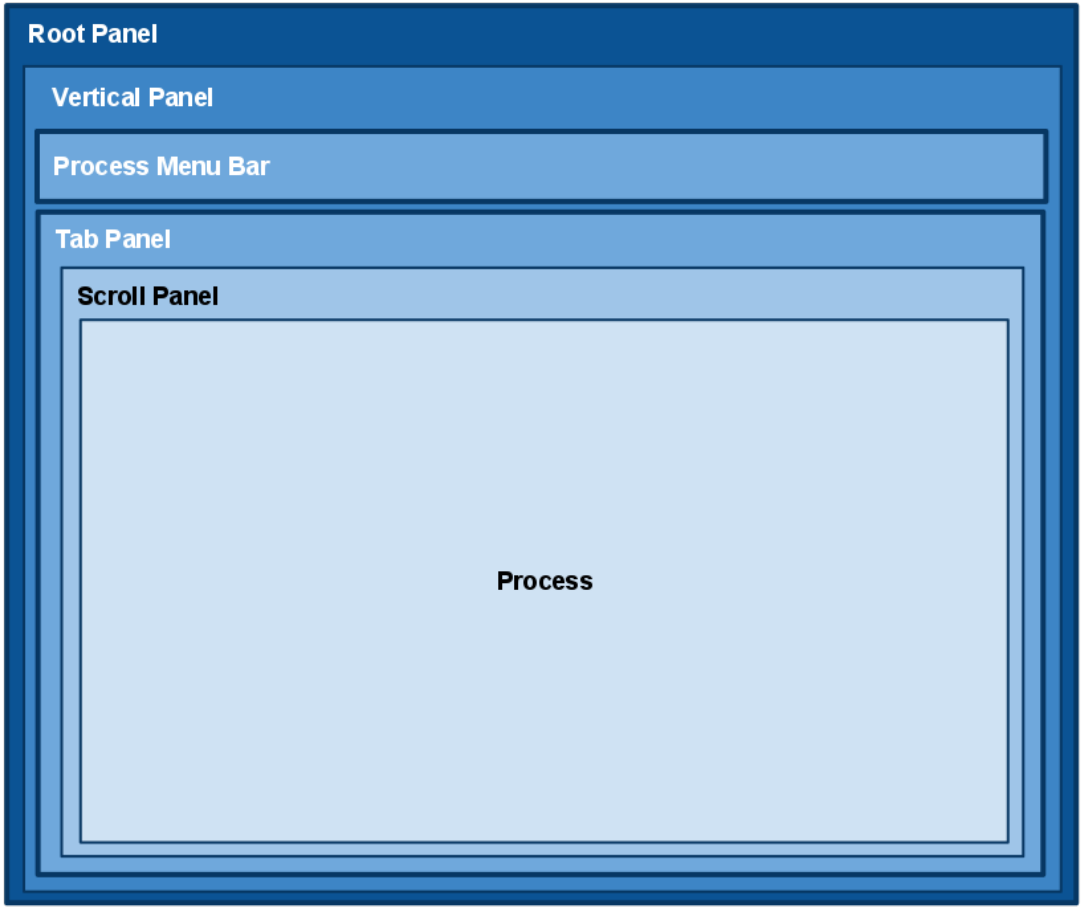
- Built on top of the Google Web Toolkit (GWT) Framework:
  - Features Java UI development and Drag-Drop UI building
  - UI Code gets transformed into AJAX UI code
  - HTML5 Support and cross-platform compliance

- Seeks to replicate the appearance and functionality of the Mockingbird examples
  - Appearance - **Panels** help structure the process view while **widgets** are the interactive pieces.
  - Functionality - Event-handling and view-dispatcher communication code is modularized within the project as well

View-Dispatcher



Structure of Process View



Demonstration of Implementation

Summary and Future Work

- Summary
  - Developed and crafted a user-interface through UI mockup examples that provides stable process workflow
  - VizGuidance supports dynamic process execution workflow for process performers
- Future Work
  - Experimental evaluation of process guidance using UI mockups and/or GUI prototypes
  - Dispatcher communication
  - Synchronize workflow among agents
  - Further Implementation of the GUI

Feedback

- Questions? Comments?
- Continuously looking for feedback in order to improve this UI frontend work.
- Please email me at mbarrene@cs.umass.edu if you have feedback.