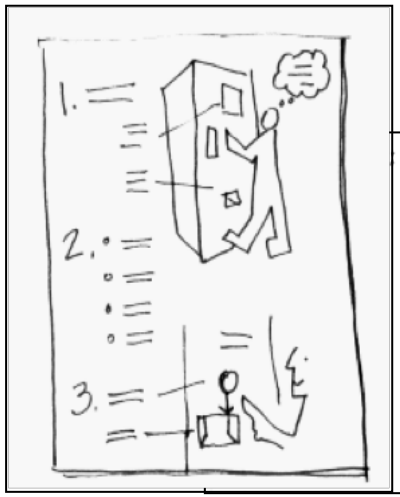
NTNU

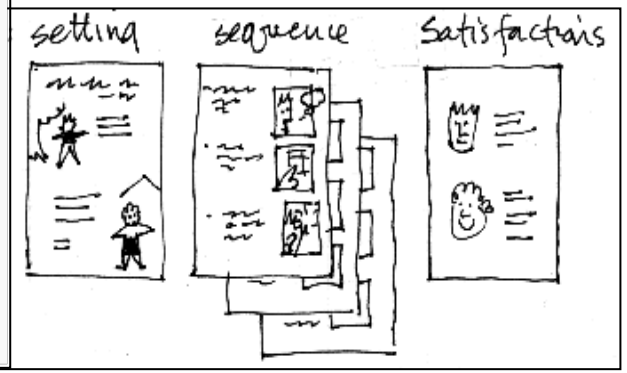
Norwegian University of
Science and Technology

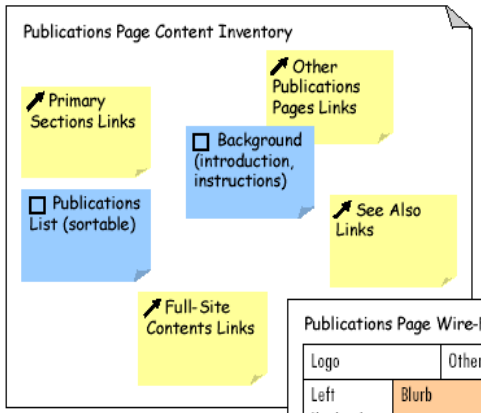
Modeling user interfaces with Cal and Ptolemy

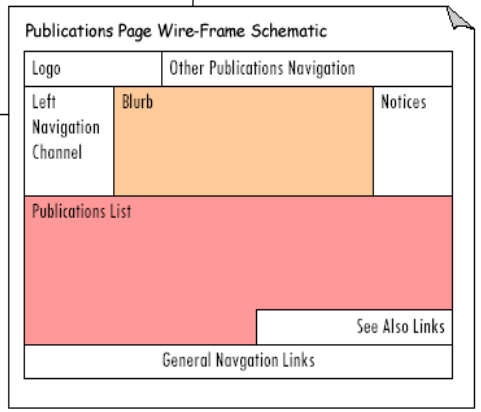
Hallvard Trætteberg, Associate Professor
Dept. of Computer and Information Sciences
Norwegian Univ. of Science and Technology

Designers use informal representations...

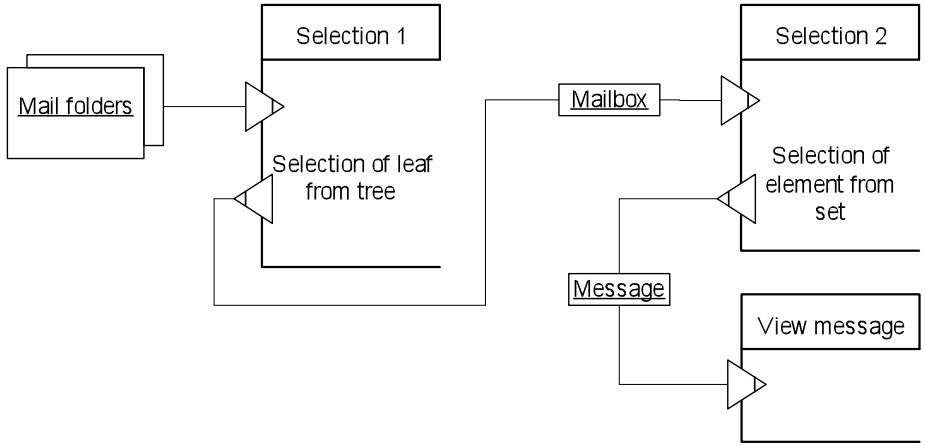




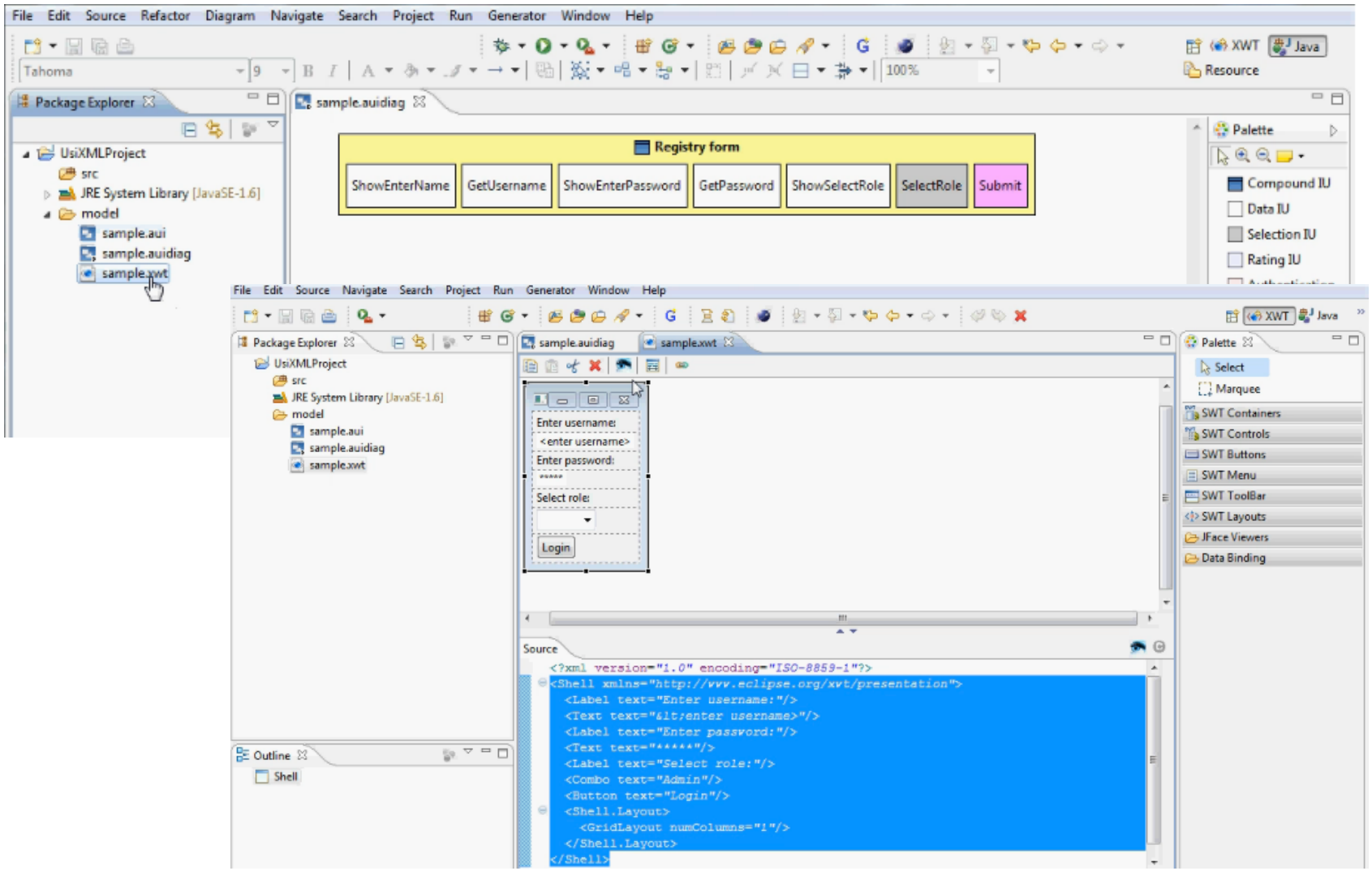




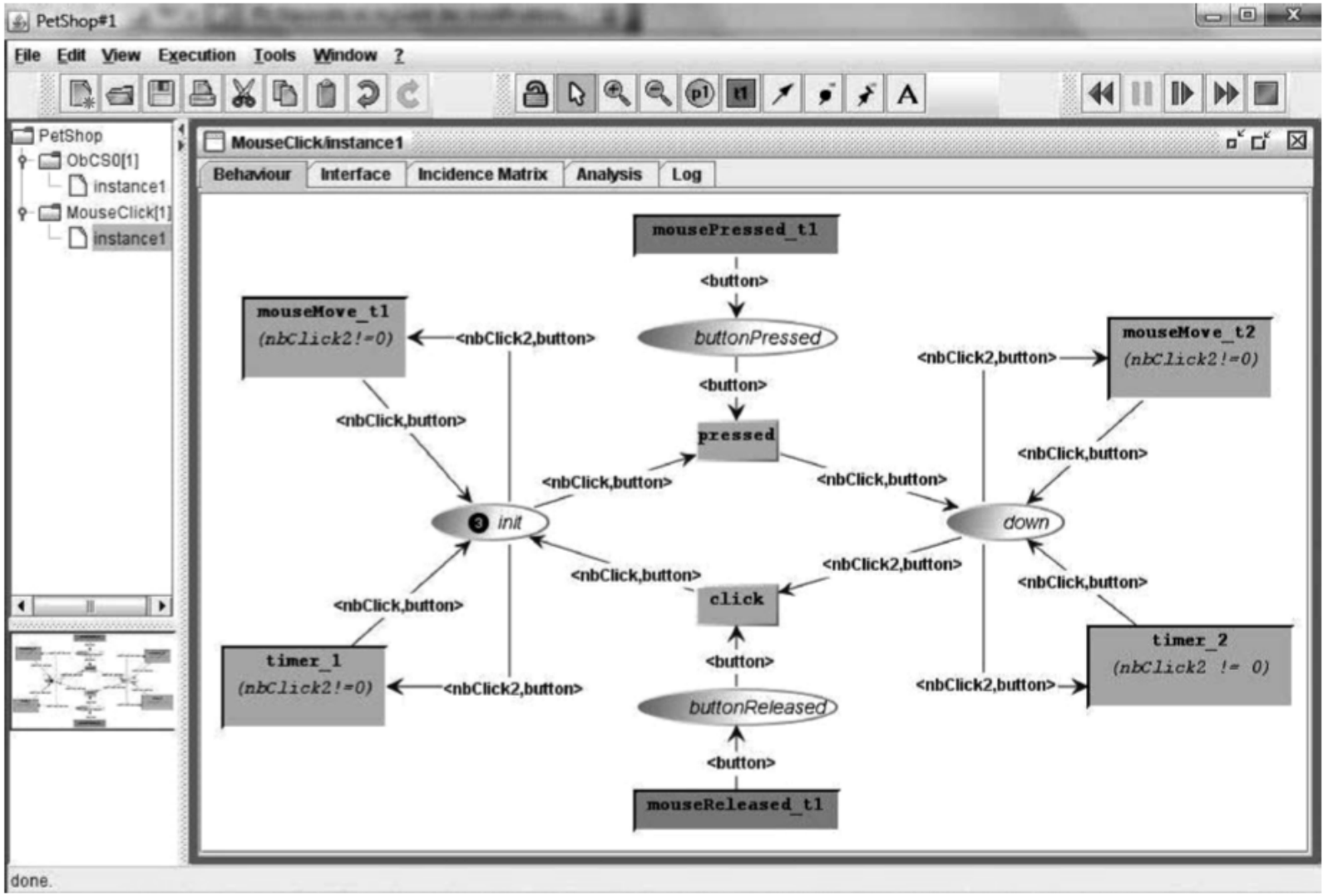
Engineers use models



UsiXML [Vanderdonckt] – A family of XML-based notations for UI elements



Pet shop [Palanque] – Modeling safety critical UIs with ICO PetriNets

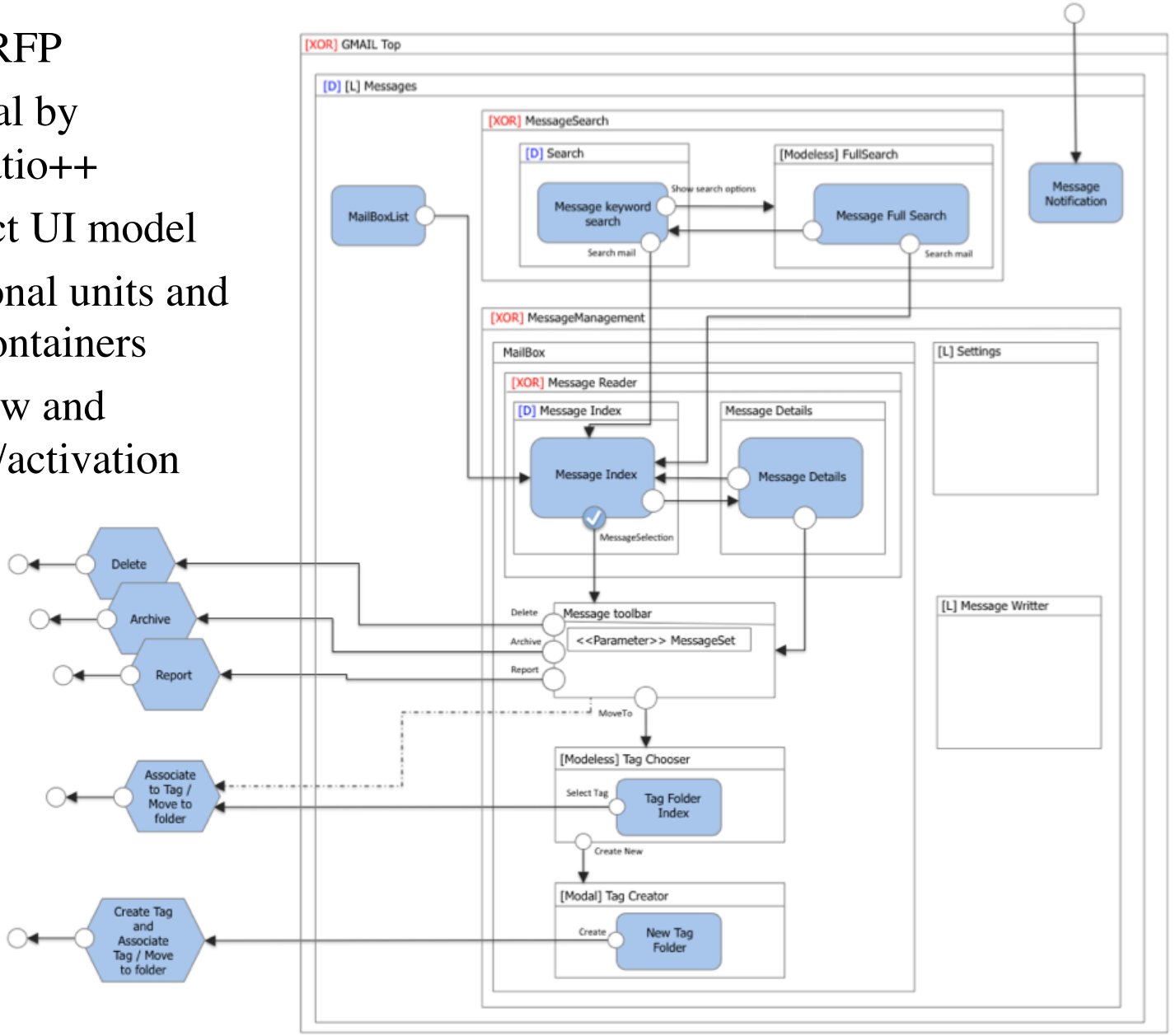


Lots of pragmatic approaches (read: non-academic and useful)

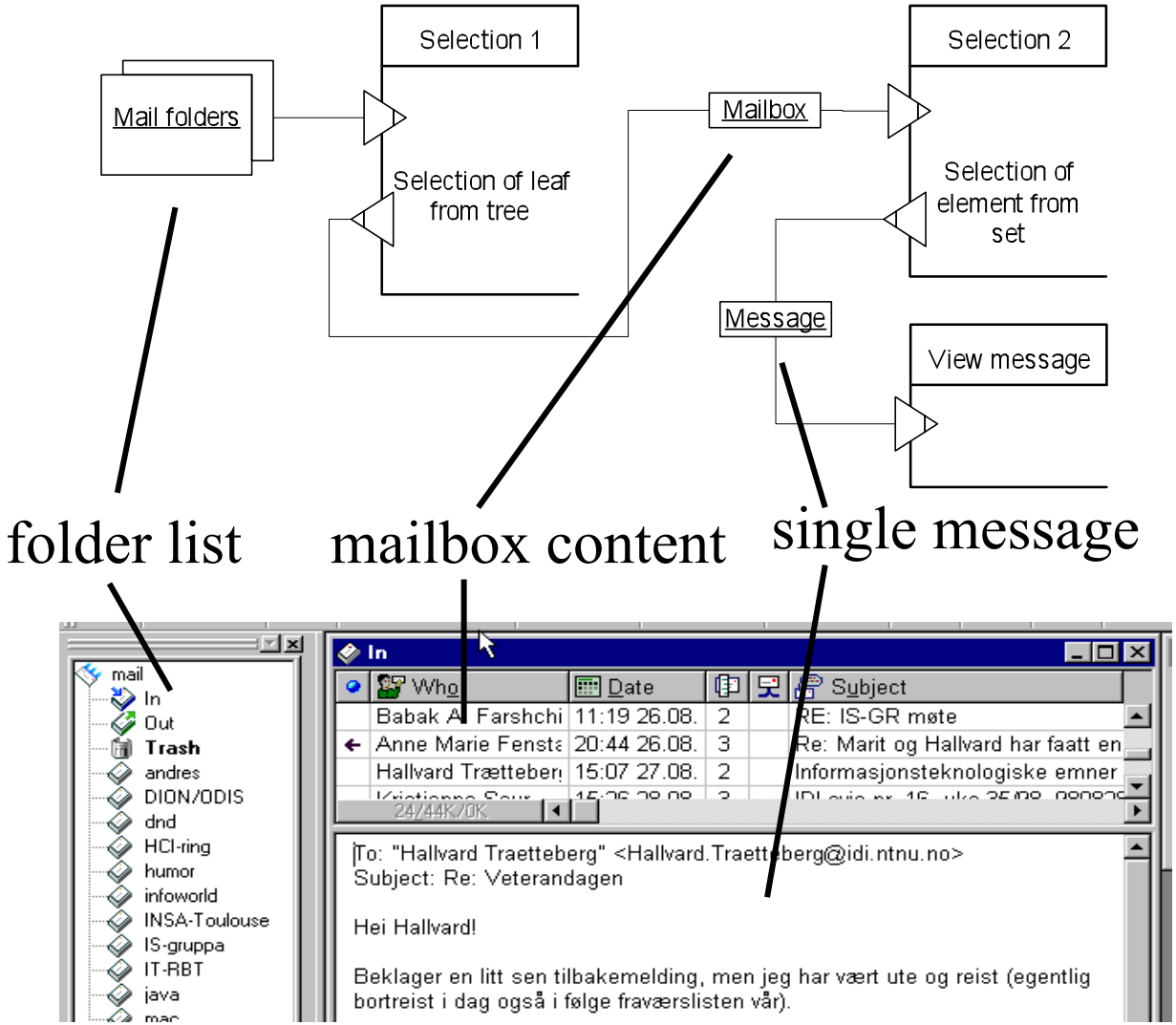
- XML-based formats for describing user interface layout and style
  - XHTML (W3C) , XAML (Microsoft), JavaFX (Oracle), XUL (Mozilla)
  - template languages for web pages
- DSLs
  - Ecore-based: Eclipse 4's workbench model, Wazaabi
  - Xtext-based: APPlause, MOBL, Agentry
- Application modeling
  - Esito's Genova – business applications for the desktop and web
  - WebRatio - business applications for the web
- Standardization
  - WebML
  - IFML (in progress)
  - Model-Based User Interfaces (MBUI) Working Group

IFML – Interaction Flow Modeling Language

- OMG RFP
- Proposal by WebRatio++
- Abstract UI model
- Functional units and view containers
- Dataflow and control/activation signals



Diamodl

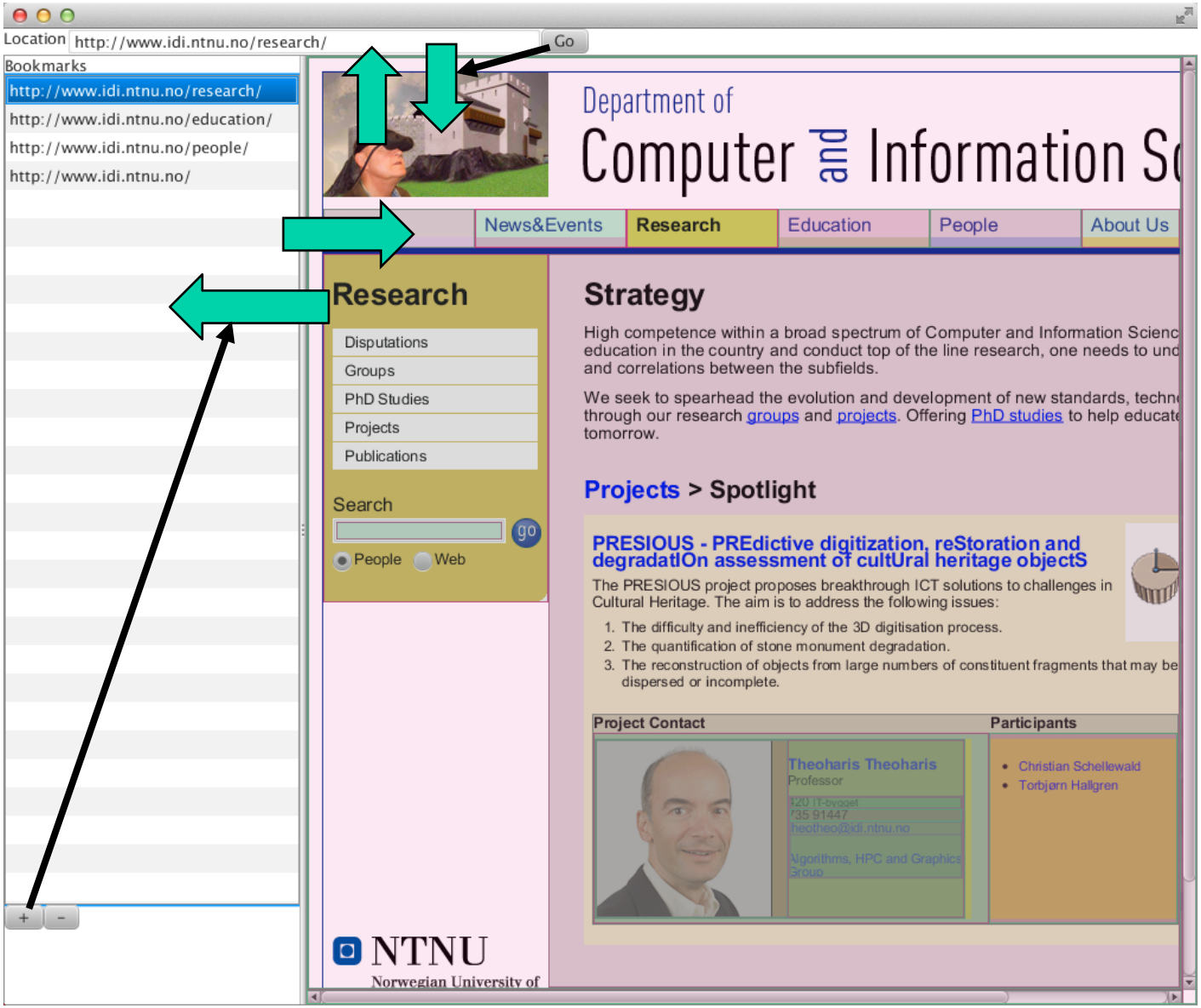

Selection 1
Selection 2
Mail folders
Mailbox
Selection of leaf from tree
Selection of element from set
Message
View message
folder list
mailbox content
single message
mail
In
Out
Trash
andres
DION/ODIS
dnd
HCI-ring
humor
infoworld
INSA-Toulouse
IS-gruppa
IT-RBT
java
In
Who
Date
Subject
Babak A Farshchi 11:19 26.08. 2
RE: IS-GR møte
Anne Marie Fenst 20:44 26.08. 3
Re: Marit og Hallvard har faatt en
Hallvard Trætteber 15:07 27.08. 2
Informasjonsteknologiske emner
24/44K/0K
To: "Hallvard Traetteberg" <Hallvard.Traetteberg@idi.ntnu.no>
Subject: Re: Veterandagen
Hei Hallvard!
Beklager en litt sen tilbakemelding, men jeg har vært ute og reist (egentlig bortreist i dag også i følge fraværslisten vår).

NTNU

But what is

- the *semantics* of the model (runtime behavior)?
- the *role* of the model (scope/interoperability)?

Example – web browser

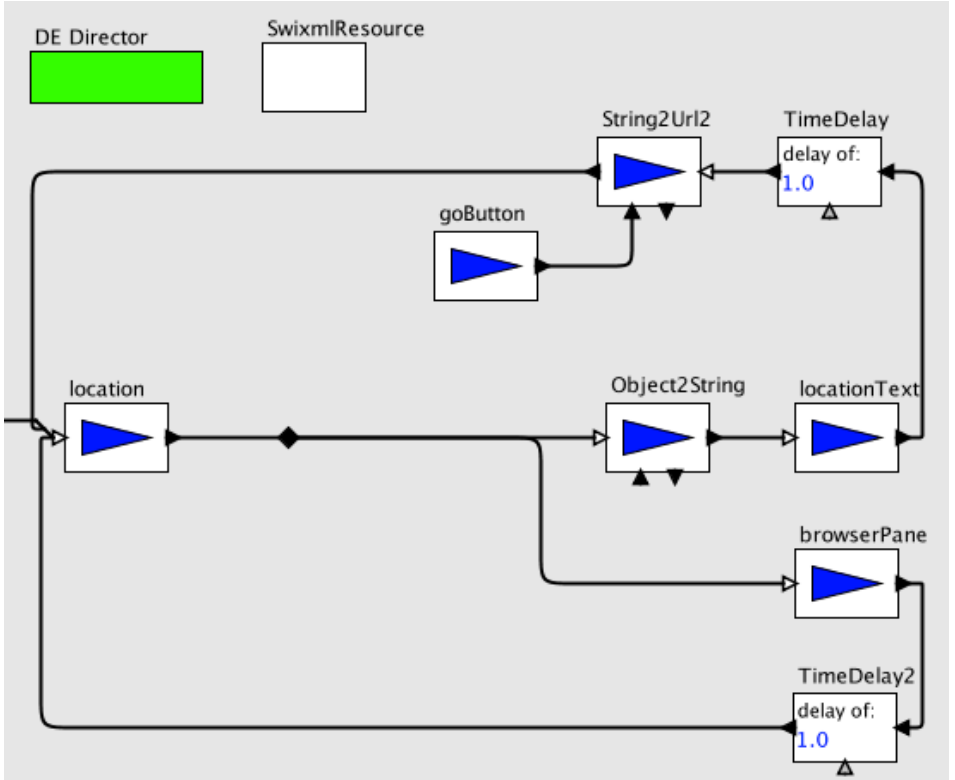


Three iterations

- Take 1 – hand-code Ptolemy actors and Java Swing toolkit
  - showed the feasibility of using Ptolemy
  - lot of work writing generic and configurable actors
  - a specialized actor language would be nice, e.g. Cal
- Take 2 – Cal implementation for Ptolemy runtime library
  - thin layer on top of atomic actors to support Cal implementation
  - extra Cal constructs for event handling, as an actor can be triggered by data and widget events, in addition to input on ports
- Take 3 – moved to Javafx toolkit
  - more Cal constructs for UI state update
  - improved thread handling

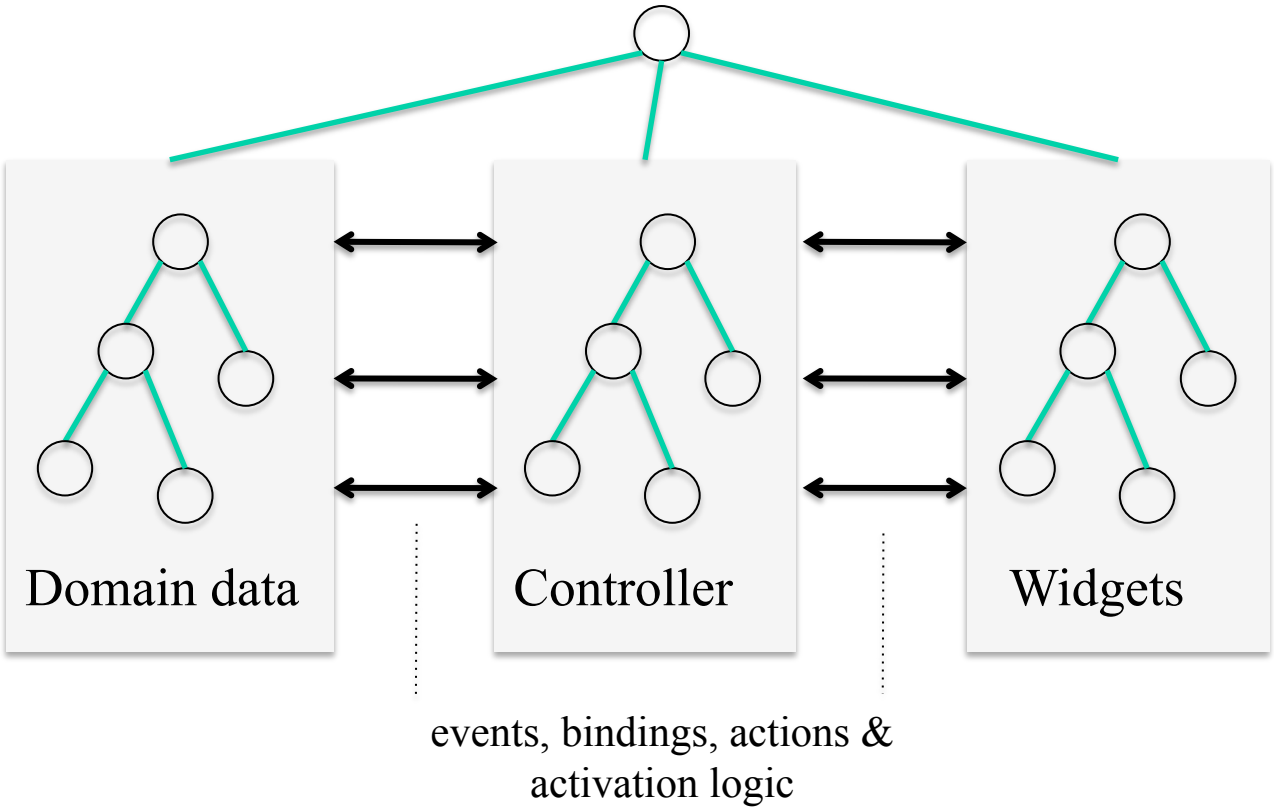
Take 1

- Event-driven, use DE Director
- Load UI with SwixmlResource
- Break cycles with TimeDelay



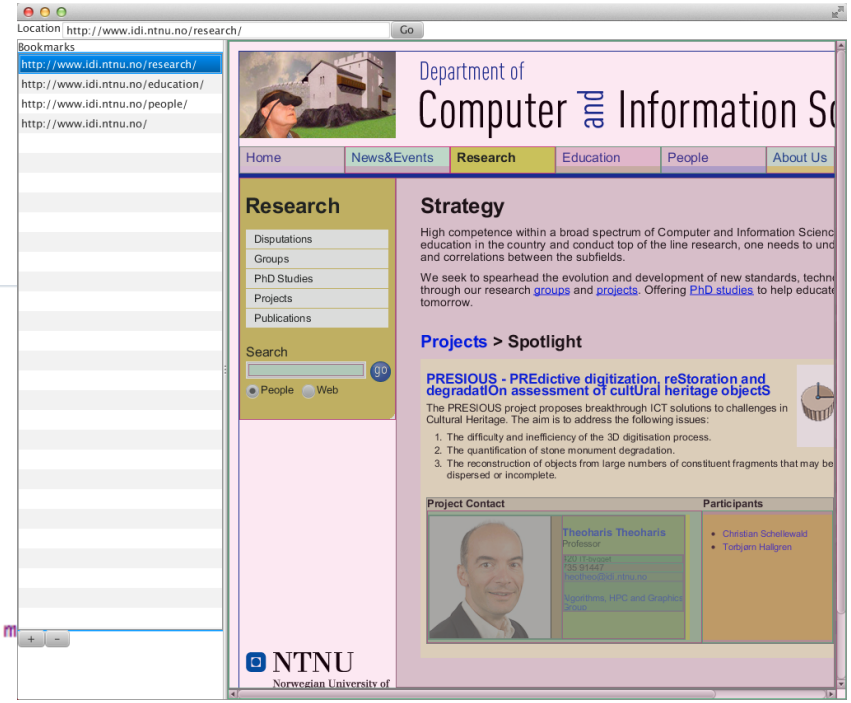
- Issues
  - hand-coding actors is difficult and tedious
  - Swing is being replaced by Javafx

Application architecture



- The whole runtime state is captured as coordinated graphs of data
- The widget hierarchy is continuously rendered on a device

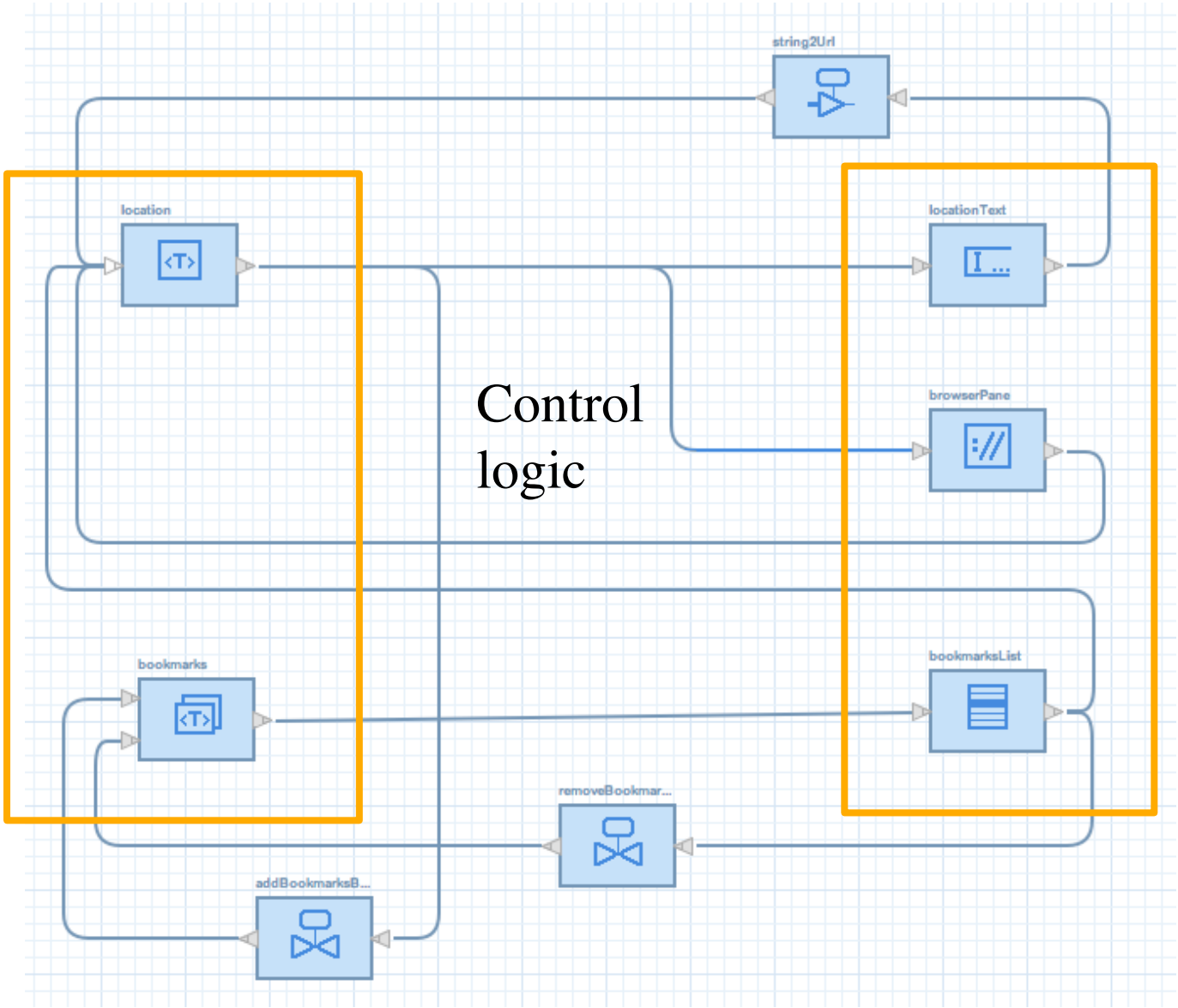
Javafx widgets with fxml



browser.fxml | PTextField.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.paint.*?>
<?import javafx.scene.web.*?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" m
  <top>
    <HBox>
       <Label text="Location" />
       <TextField id="browser_location_text" prefWidth="400.0" text="http://www.idi.ntn
       <Button id="browser_location_goButton" mnemonicParsing="false" text="Go" />
    </HBox>
  </top>
  <center>
    <SplitPane dividerPositions="0.25" focusTraversable="true">
      <items>
         <VBox>
            <Label text="Bookmarks" />
            <ListView id="browser_bookmarks_list" prefHeight="700.0"  prefWidth="200.0"
            <HBox>
                <Button id="browser_bookmarks_addButton" text="+" />
                <Button id="browser_bookmarks_removeButton" text="-" />
            </HBox>
         </VBox>
         <WebView id="browser_viewer" prefHeight="800.0" prefWidth="800.0" />
      </items>
    </SplitPane>
  </center>
</BorderPane>
```

Model of the controller



Data

Control logic

Widgets

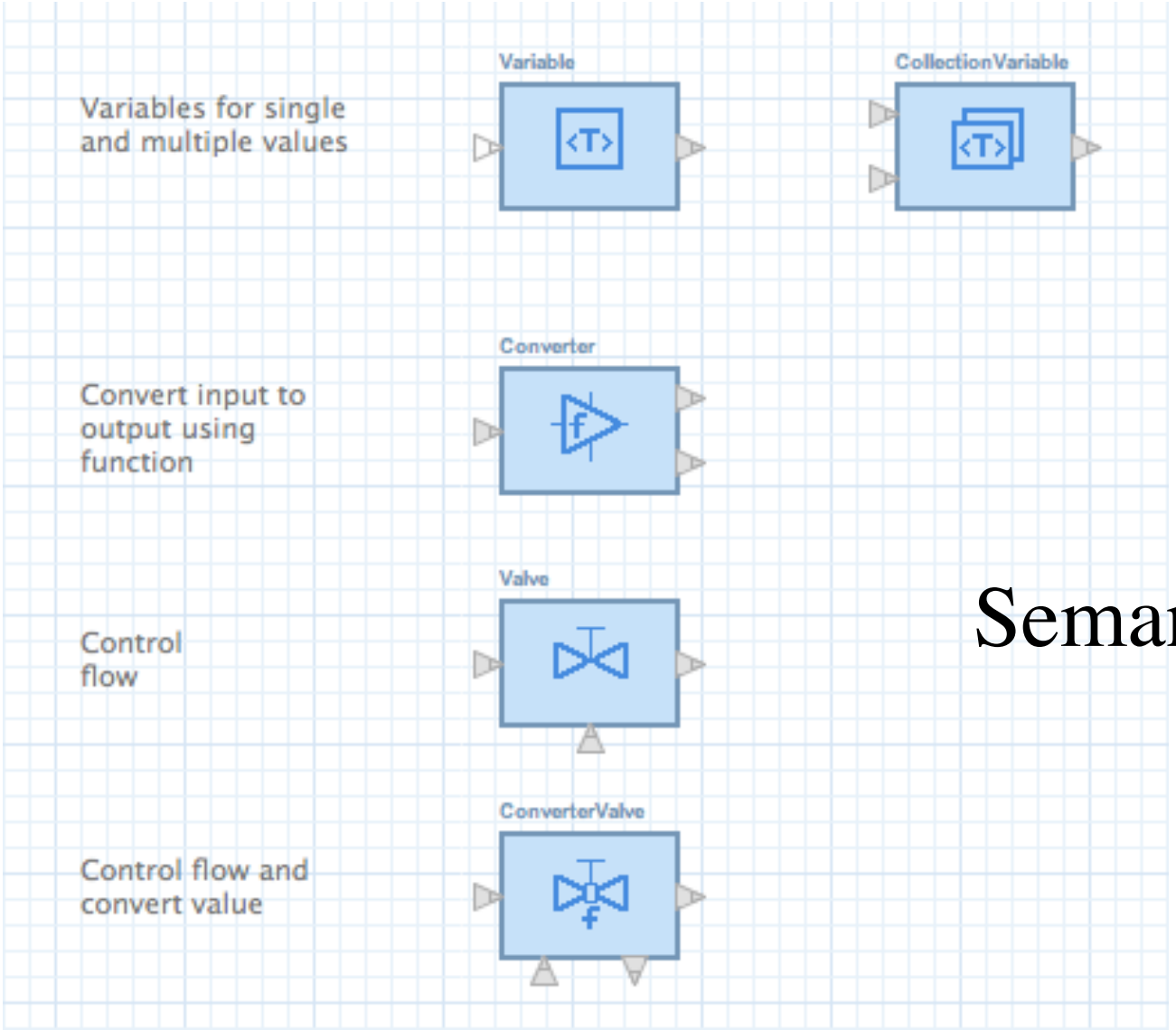
Model of the controller

browser.xactor | ui.xactor | swing.xactor | browser.swixml. | AbstractCaltrop | javafx.xa

```
network browser :
    entities
        location = Variable<URL>(allowNull=false)

        locationText = PTextField(id="browser_location_text")
        string2Url = PButtonEagerConverter<String, URL>(id="browser_location_goButton",
            fun=[String s|try { new URL(s) } catch (Exception e) { throw new RuntimeException(e)}]
        )
        browserPane = PBrowser(id="browser_viewer")

        bookmarks = CollectionVariable<URL>
        bookmarksList = PCollection<URL>(id="browser_bookmarks_list")
        addBookmarksButtonValve = PButtonValve<URL>(id="browser_bookmarks_addButton")
        removeBookmarksButtonValve = PButtonValve<URL>(id="browser_bookmarks_removeButton")

    structure
        location.value -- toString() when it != null --> locationText.systemOutput
        location.value          --> browserPane.systemOutput
        string2Url.output       --> location.setValue
        locationText.userInput  --> string2Url.input
        browserPane.userInput   --> location.setValue

        bookmarks.value --> bookmarksList.systemOutput
        bookmarksList.userInput --> location.setValue

        location.value --> addBookmarksButtonValve.input
        addBookmarksButtonValve.output --> bookmarks.add

        bookmarksList.userInput --> removeBookmarksButtonValve.input
        removeBookmarksButtonValve.output --> bookmarks.remove
end
```

Network of (instances of) reusable actors

Generic actors, based on Diamodl



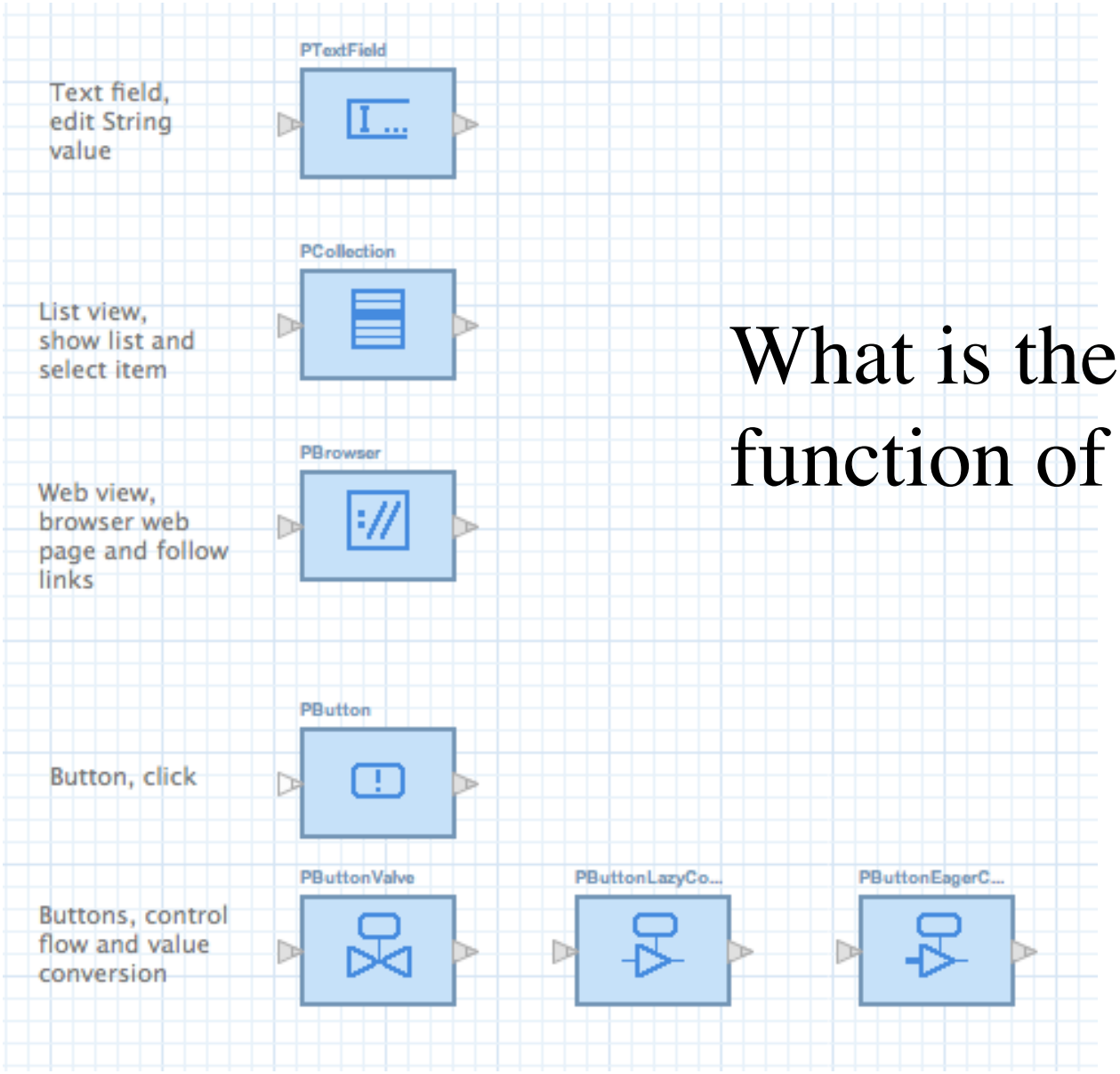
Semantics!

Cal – generic actors

browser.xactor | ui.xactor | swing.xactor | browser.swixml. | AbstractCaltrop

```
namespace org.ptolemy.xtext.examples.ui :

import java.util.Collection
import java.util.ArrayList

actor Variable<T>(T initialValue=null, boolean allowNull=true, (T) => boolean validator)
T* setValue ==> T value :
    var T current = initialValue;
    function diff(T oldValue, T newValue) --> boolean :
        (oldValue != newValue && (oldValue == null || (! oldValue.equals(newValue))))
    end
    function isValid(T value) --> boolean :
        (allowNull || value != null) && (validator == null || validator.apply(value))
    end
    action [newValue] any ==> [current] when current != old_current do
        val T value = newValue.values.head
        if (diff(current, value) && isValid(value)) {
            current = value
        }
    end
end
```

Widget actors, wrappers/abstractions



What is the essential function of widget?

Cal – widget actors

javafx.xactor | browser.xactor | ui.xactor | swing.xactor | browser.swi

```
actor PTextField(String id)
String systemOutput ==> String userInput :
    val @javafx TextField textField = get(id, TextField);
    var @javafx String text := textField.text update textField.text = text;
    initialize ==> [text] end
    action [string] ==> do
        text = string
    end
    event textField.text [propertyEvent] ==> [text] end
    event textField!^action [actionEvent] ==> [text] end
end
```

Widgets

Cal implementation

- Xtext and Xbase provide tight integration with Eclipse platform
  - editor with syntax highlighting, code completion, navigation, ...
  - can refer to and use Java APIs (standard, third-party, custom)

- Implementation liberties
  - expressions – Java-like, closures, syntactic sugar, due to Xbase
  - atomic actors – event specifications
  - network – inline atomic actors, data transforming relations

- Runtime state
  - referring to contextual data
  - updating contextual data
  - threads

Summary

- Ptolemy as a platform for exploring and experimenting with semantics for domain-specific languages

- Utilize Ptolemy and Cal for developing apps
  - language, architecture and tooling issues