

An Experiment Data Analysis Framework: Evaluating Interactive Information Behavior with R

Ralf Bierig, Jacek Gwizdka,
Michael Cole, Nicholas J. Belkin



 ralf@bierig.net

 www.bierig.net

Overview

1. Introduction / Background / Rutgers
2. System framework
 - Experiment System
 - Analysis System
3. Context evaluation with the framework
4. Screenshots
5. Current State (Demo) and future plans

Introduction / Background

- Information Science / Information Retrieval
 - Traditionally: Matching search request to documents
 - Query as search unit
 - Context-free search
 - Performance measure: Relevance
 - Integrating Interaction as part of the search process
 - Task as search unit
 - User's information interaction (central) part of search process
 - Developing models that explain usefulness of content
- Need for more effort for context model evaluation
 - Initial activity in IRIX workshops and IiX conferences (3rd time this year in New Brunswick)
 - **PooDLE project @ Rutgers**

The PooDLE Project

<http://comminfo.rutgers.edu/imls/poodle>



- **Goal:** a personalization assistant to support contextualized (i.e. personalized) information retrieval
- **Questions** we address:
 - Determination of significant contextual factors
 - Implicit identification of values of factors
 - Determination of interaction effects amongst factors
 - Construction of a computational framework for taking account of context factors and their interactions
 - Creation and Evaluation of a personalization prototype
- **What we do:**
 - Experiments (information seeking for different tasks)
 - Observing users' interactive search behavior
 - Collect feedback on usefulness from their saved pages
 - Analyze data to find behavioral correlates of contextual features and predict usefulness based on behaviors

Technical challenges / requirements

- Large data sets
 - \sim 1+ GB of textual log data / experiment
 - \sim 1x GB of video log data / experiment
- Processing Speed of models / test / visualizations
- Flexibility (many possible models/combinations)
- Support

Technical challenges / requirements

- Large data sets
 - \sim 1+ GB of textual log data / experiment
 - \sim 1x GB of video log data / experiment
- Processing Speed of models / test / visualizations
- Flexibility (many possible models/combinations)
- Support



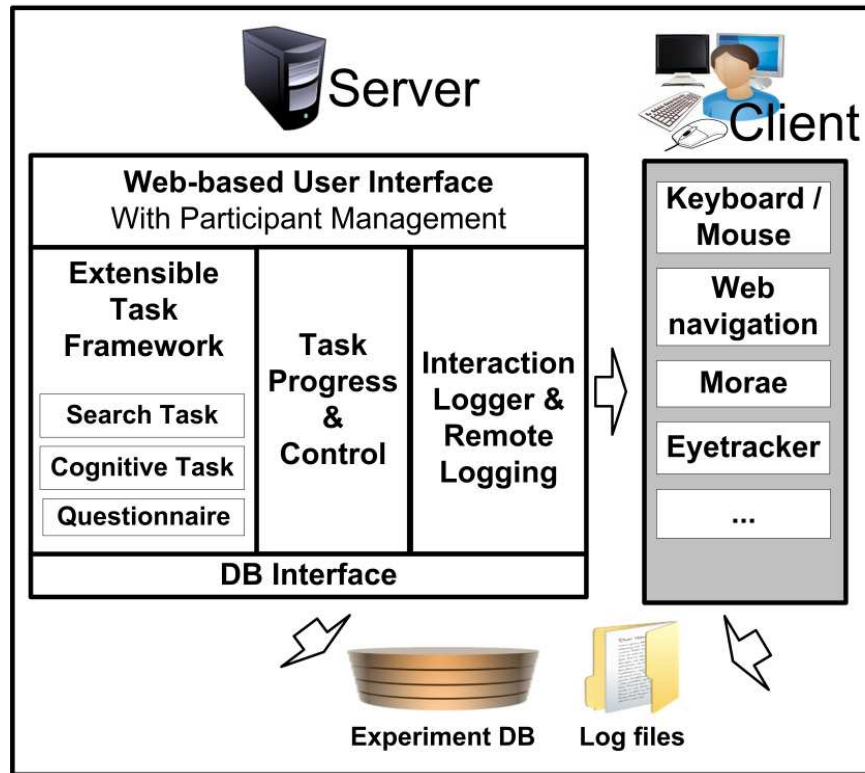
Technical challenges / requirements

- Large data sets
 - \sim 1+ GB of textual log data / experiment
 - \sim 1x GB of video log data / experiment
- Processing Speed of models / test / visualizations
- Flexibility (many possible models/combinations)
- Support
 - Integration with Java (JRI)
 - Data access (RJDBC)
 - Data processing
 - Data manipulation
 - Numerical processing
 - Libraries

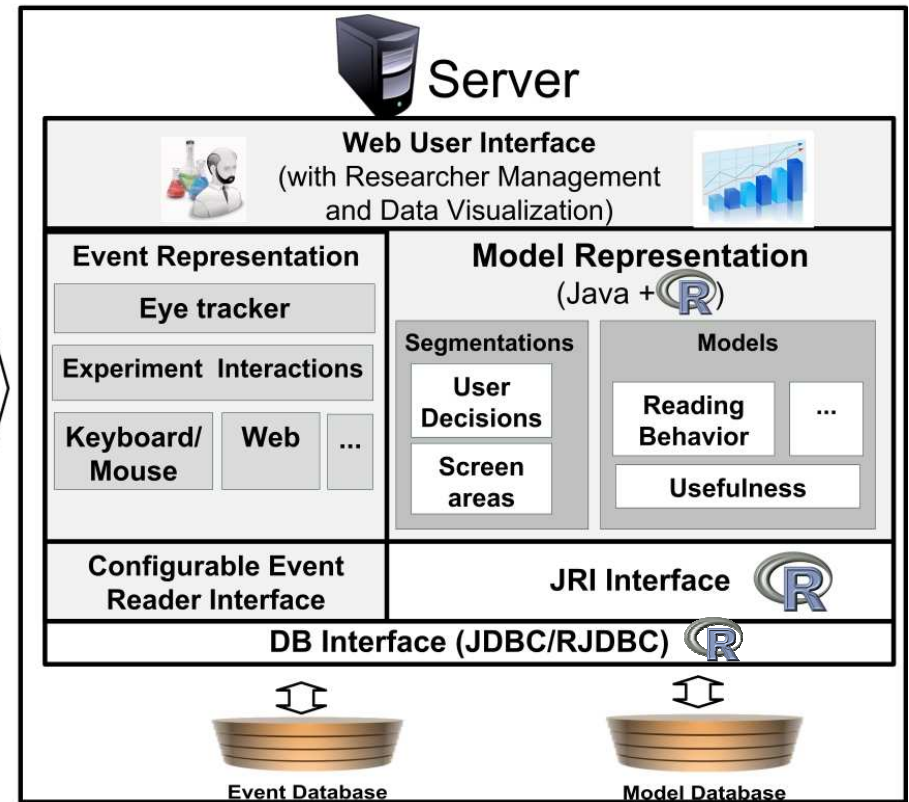


Overview

1) Experiment System



2) Analysis System



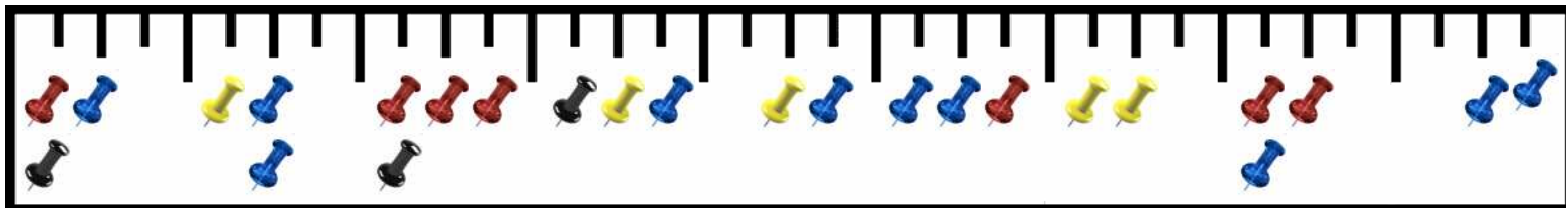
- Experiment system for designing and conducting interactive experiments in realistic application environments
- Analysis system for integrating and analyzing results from experiments

Main features

- Integrating behavioral experiment logs and user feedback
 - Unified representation (event data)
 - Single timeline
 - Synchronization of events in 
 - Error recovery
- Modeling on events in 
 - Segmentation (semantic categorization) of events
 - Hypothesis testing and learning

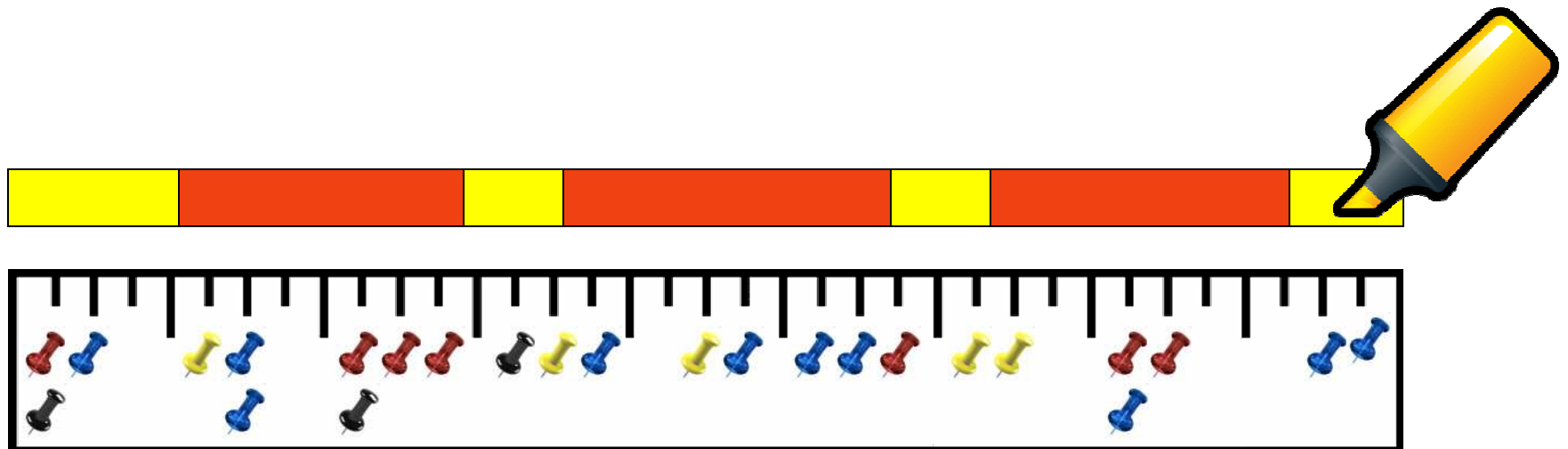
Logical elements:

- **Events** 






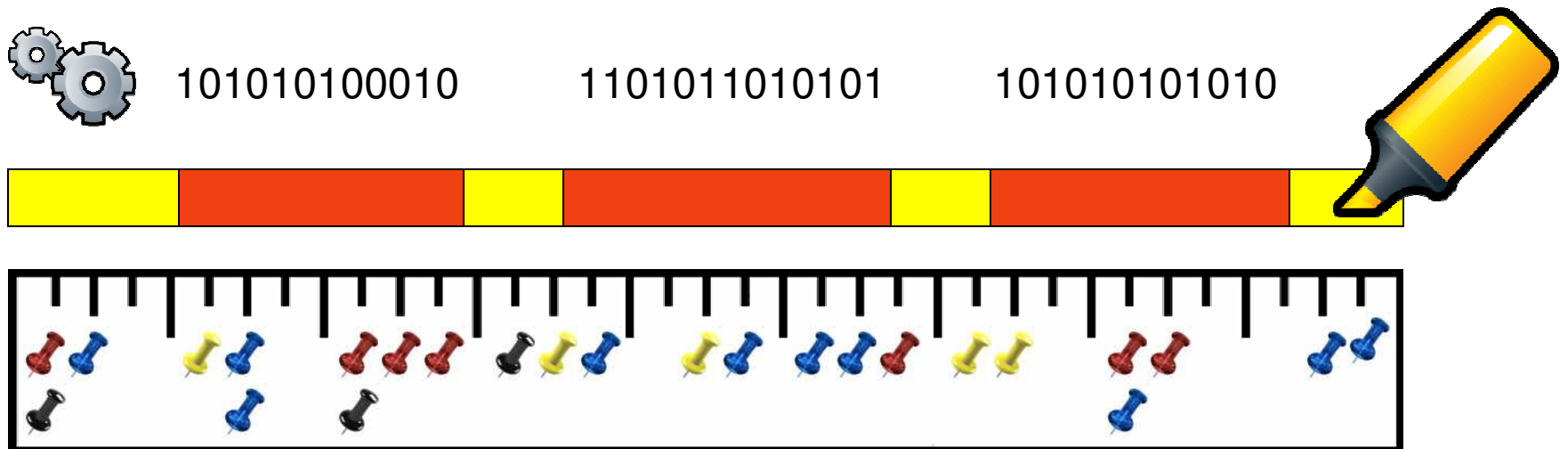
Logical elements:

- **Events** 
- **Segmentations** 



Logical elements:

- **Events** 
- **Segmentations** 
- **Models** 



How does the framework support context evaluation?

- Modularity
 - Capture the context you need
 - Analyse selective contextual effects and their interactions
- Extensibility
 - Add own events, read them from own log files
 - Add own segmentations and models (written in R)
 - Adapt framework-based models to your needs
- Data is separated from segmentations / models
- Data and models can be accessed as service and shared among researchers

Screenshots (1) - Importing Data

The screenshot shows a web application interface for importing data. The interface is divided into several sections:

- Left Sidebar:**
 - Data Imports:** A list of existing data imports: "ecir2010", "FABIO", and "ecir2010_2". Each entry has a small 'x' icon to its right. Above the list is a text input field and a "Create" button.
 - Models:** A section with a gear icon and the label "Models".
- Top Navigation Bar:** A series of icons and labels: "Connection & Readers" (selected), "User / Task", "Interaction", "RUIConsole", "Tobii", and "UsaProxy".
- Main Content Area:**
 - Enter Experiment Data Source:** A section with the following form fields:
 - Data Import: ecir2010
 - Server: localhost
 - Port: 3306
 - Experiment Database: kpe1
 - User: kpe1
 - Password: [masked with dots]
 Below these fields is a "Save and Connect" button with a floppy disk icon.
 - Enable Readers:** A section with a list of readers and their status:

Reader	Status	Checkbox
User / Task	On	<input checked="" type="checkbox"/>
Interaction	On	<input checked="" type="checkbox"/>
RUIConsole	On	<input checked="" type="checkbox"/>
Tobii	On	<input checked="" type="checkbox"/>
UsaProxy	On	<input checked="" type="checkbox"/>
Morae	Off	<input checked="" type="checkbox"/>
URLTracker	Off	<input checked="" type="checkbox"/>

 Below this table are two buttons: "Save" (with a floppy disk icon) and "Run Readers" (with a gear icon).

Screenshots (2) – Model Selection

Data Imports

Models

Create

test x

ecir2010model x

Definition & Selection | Information | Features | Results

Name and Define Model

Name: test

Data Import: ecir2010

Model: ScreenRegionSegmenter

Save

Select Users, Tasks and Eventtypes

Users:

kpe1_s001 kpe1_s002 kpe1_s003 kpe1_s004 kpe1_s005

Tasks:

KPE1 Task 11 KPE1 Task 12 KPE1 Task 13 KPE1 Task 14 KPE1 Task 21 KPE1 Task 22 KPE1 Task 23 KPE1 Task 24

Experiments:

KPE1

Event types:

User / Task Interaction RUIConsole Tobii

UsaProxy

Save Run Model

Screenshots (3) – Model Info & Features

The screenshot displays a software interface with a sidebar on the left and a main content area on the right. The sidebar contains 'Data Imports' and 'Models' sections. The 'Models' section has a 'Create' button and two model entries: 'test' and 'ecir2010model', each with a close button. The main content area has a navigation bar with 'Definition & Selection', 'Information' (selected), 'Features', and 'Results' tabs. Below the navigation bar, the 'Information' tab is active, showing 'General Info' and 'Model Input Features' sections.

General Info

Name: ScreenRegionSegmenter
 Type: edu.rutgers.poodle.models.ScreenRegionSegmenter
 Description: Segments Tobii eye fixations into screen regions defined by the researcher.

Model Input Features

#	Feature name	parameter1	parameter2	parameter3	parameter4	parameter5
1	Screen Region	name	x1	y1	x2	y2

Screenshots (3) – Model Info & Features

This screenshot shows the 'Model Info' tab of a software interface. The left sidebar contains 'Data Imports' and 'Models' sections. The 'Models' section lists 'test' and 'ecir2010model'. The main content area is divided into 'General Info' and 'Model Input Features'.

General Info

- Name: ScreenRegionSegmenter
- Type: edu.rutgers.poodle.models.ScreenRegionSegmenter
- Description: Segments Tobii eye fixations into screen regions defined by the researcher.

Model Input Features

#	Feature name	parameter1	parameter2	parameter3	parameter4	parameter5
1	Screen Region	name	x1	y1	x2	y2

This screenshot shows the 'Features' tab of the same software interface. The left sidebar is identical to the previous screenshot. The main content area is titled 'Add Model Features' and contains a 'Create Feature' section with a dropdown menu set to 'Screen Region' and a 'Create' button. Below this, there are two rows of feature configuration, each with a 'Save' button and input fields for name, x1, y1, x2, and y2.

Add Model Features

Create Feature: Screen Region [Create]

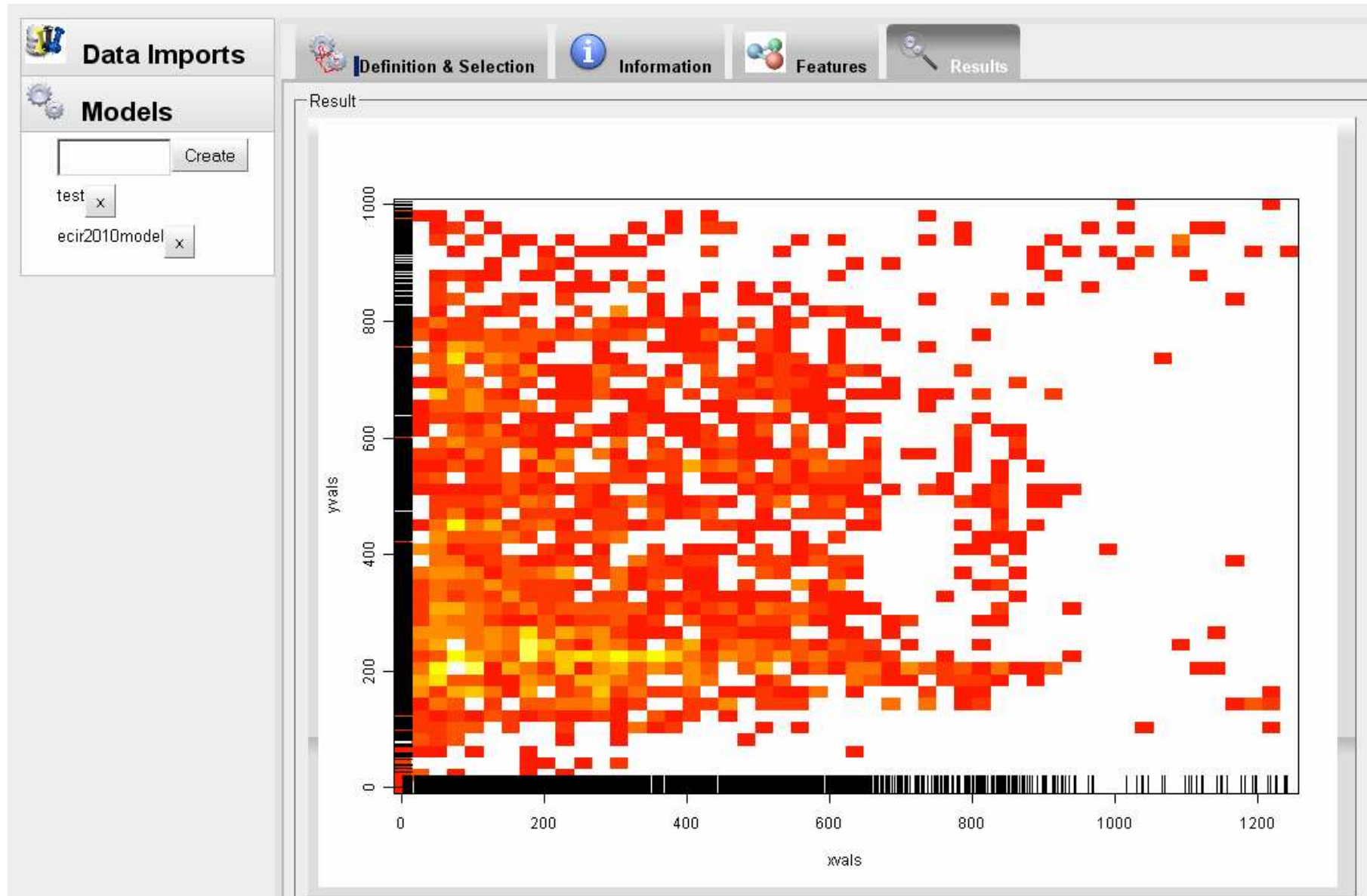
Feature 1: Save name: querybox x1: 100 y1: 100 x2: 500 y2: 200

Feature 2: Save name: content x1: 100 y1: 201 x2: 1000 y2: 1000

Screenshots (4) – Model Results



Screenshots (4) – Model Results



Current State and Future Work

- Context models for the prediction of usefulness
- Software
 - Experiment System prototype completed and will soon be available as open source from <http://sourceforge.net/projects/piirexs>
 - Analysis System will be released as open source in Fall 2010.
- Demo (on request)
- Feedback?

Acknowledgments



This research is supported by IMLS grant
LG-06-07-0105-07.



The UseR! 2010 conference visit
was enabled by
NIST.





Ralf Bierig

Post-Doc (looking for the next job!)

School of Communication and Information

Rutgers University

New Brunswick, NJ



ralf@bierig.net



<http://www.bierig.net>



<http://comminfo.rutgers.edu/imls/poodle>

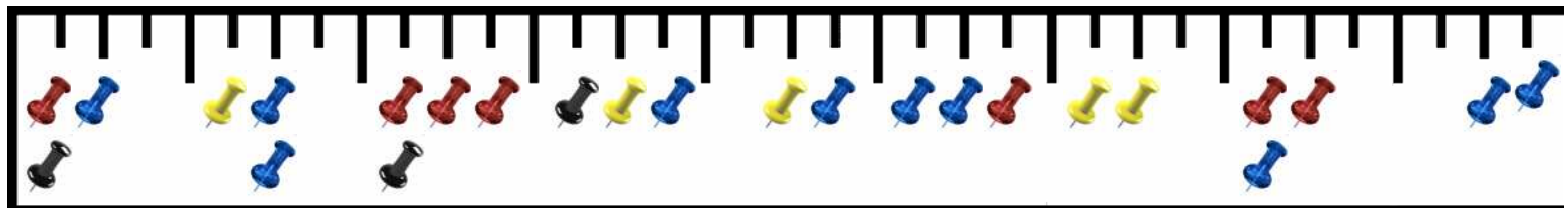
Additional Slides

Definitions

- **What is an event'**

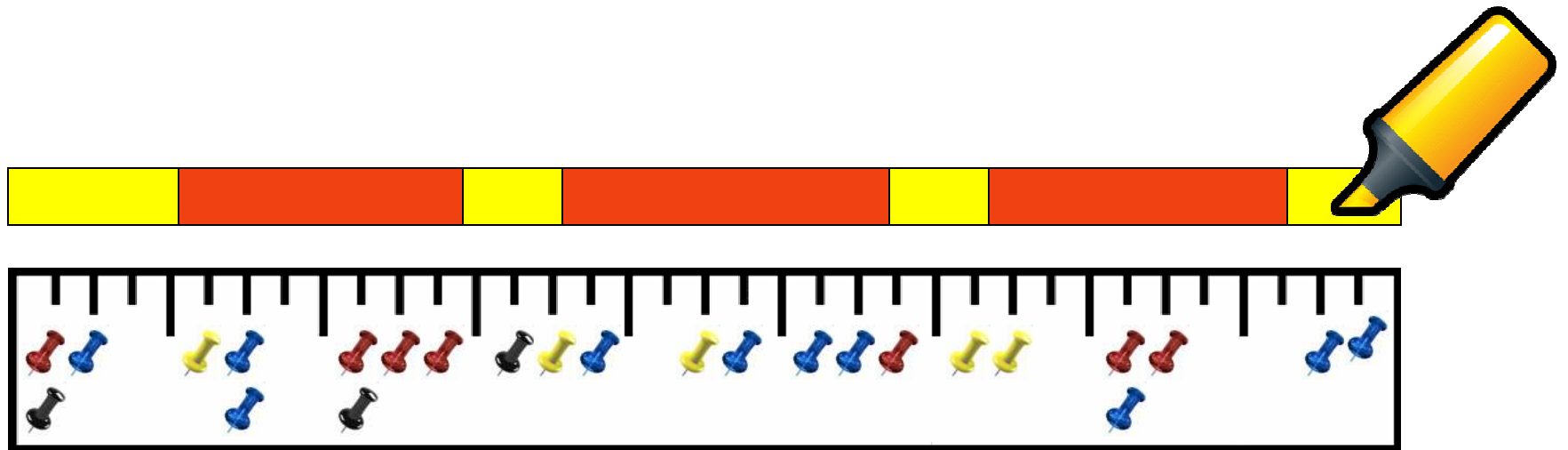


- Unique, single, and smallest unit of activity
- Marking one point in time (no time span)
- Synchronized in one time system
- Simple and no hypothesis
- Typed by origination (logging source) and categorized in simple sub-types (e.g. mouse move, keystroke)



Definitions

- **What is a segmentation?**
 - Categorizes events in semantic groups
 - Pre-conditions event data for modeling
 - Segmentation is a categorization model; many can co-exist and operate on one event data set



Definitions

- **What is a model?**

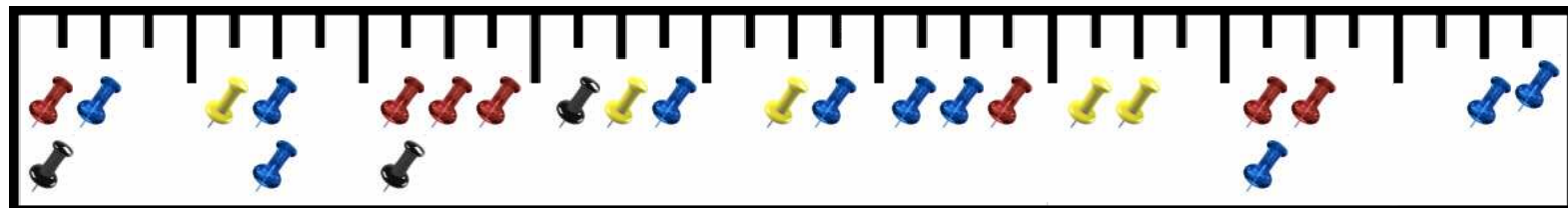
- Defines one or more hypothesis
- Generates secondary data from events
- Independent from events
- Can be based on segmentation



101010100010

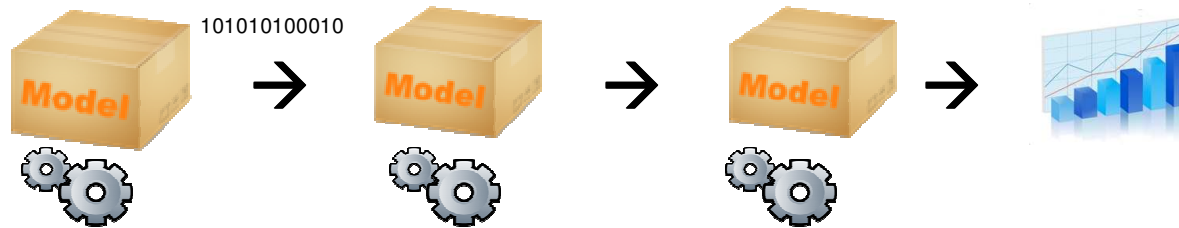
1101011010101

101010101010

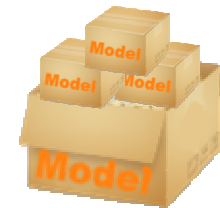


What to do with Models?

- Models can be chained



- Models can re-use other models
(Templates/Pattern)



- Models can be adapted or extended

