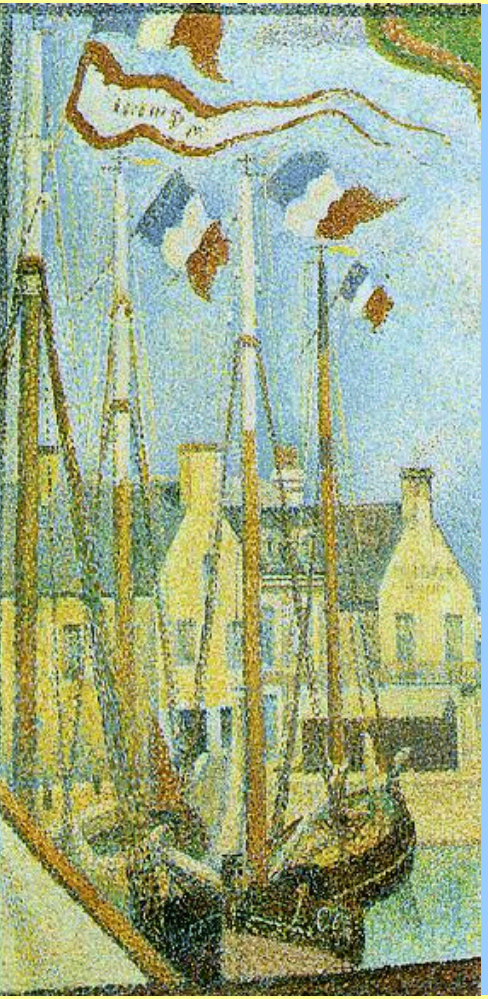
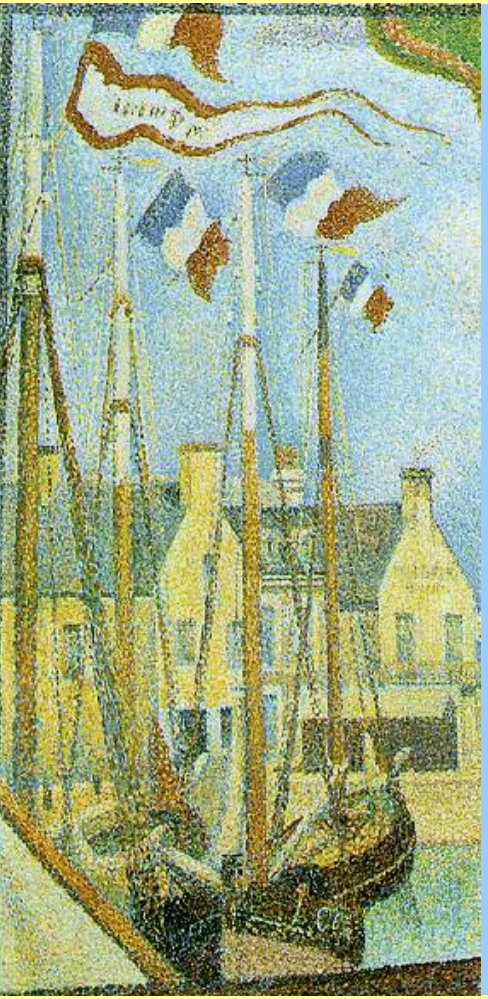


Software Engineering Using Rationale

Janet E. Burge, Miami University

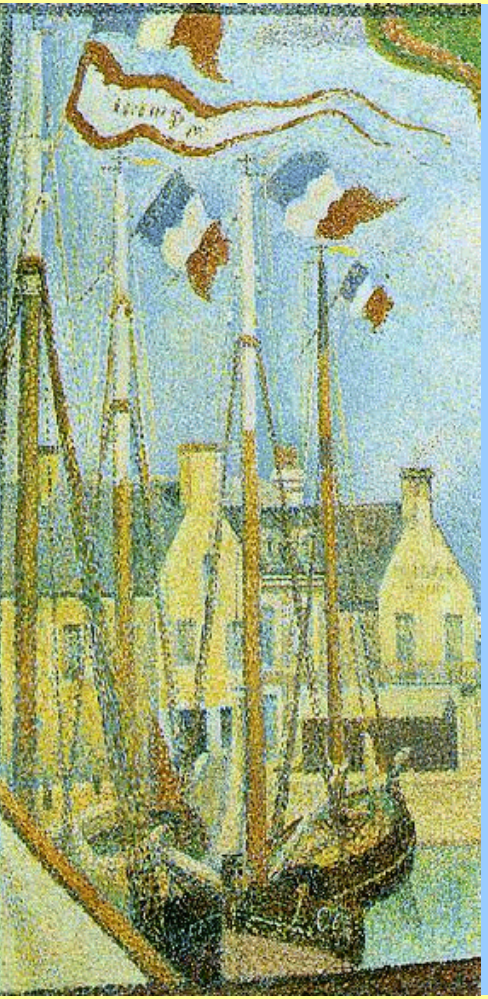
http://www.users.muohio.edu/burgeje/SEURAT/

What is Rationale?



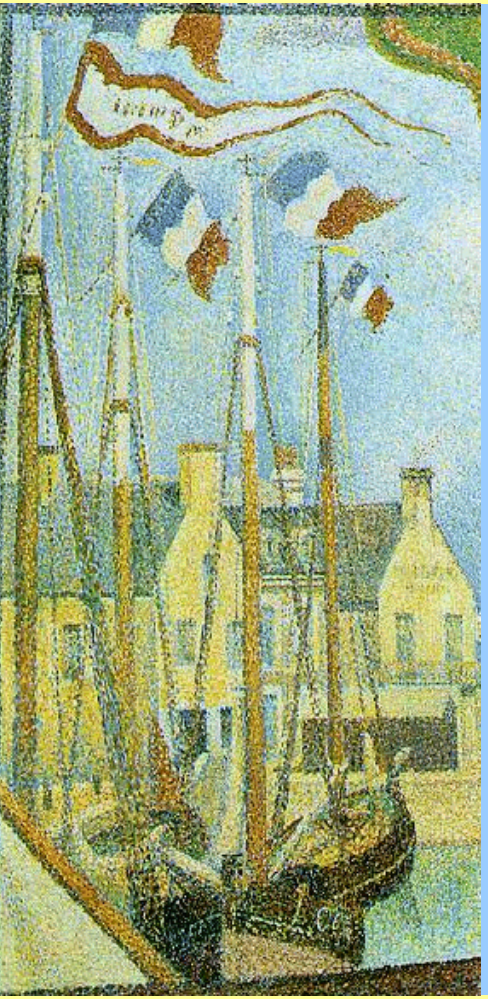
- The rationale for a system describes the reasons behind decisions made during its development
- Not just a snapshot of the final decision – the rationale contains all alternatives considered, arguments for and against them, and tradeoffs considered
  - The argumentation behind the decision

Example (from a Conference Room Scheduling System)



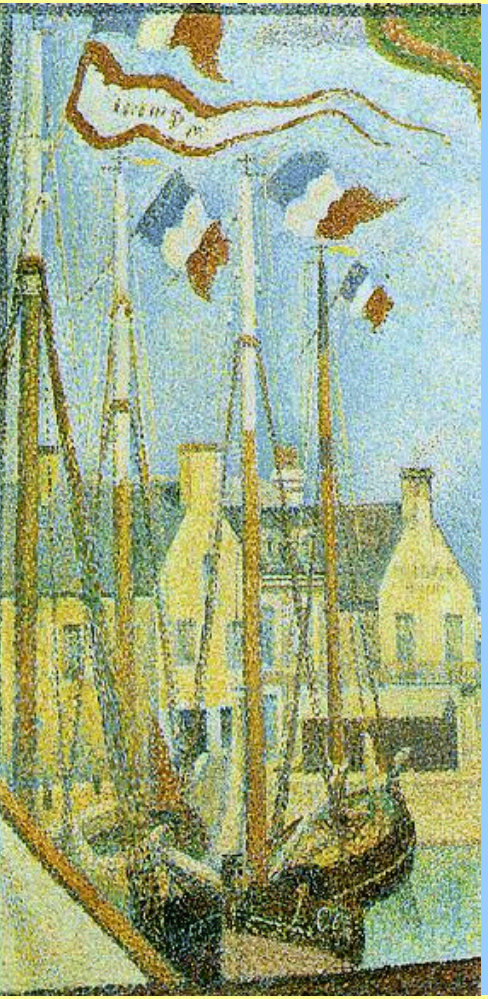
- *Decision*: How do we represent a conference room in the system?
  - *Alternative*: store the name (location) as a string
    - *Argument for*: simple to code
    - *Argument against*: difficult to extend
  - *Alternative*: create a conference room class
    - *Argument for*: can contain information other than location

Why is Rationale valuable?



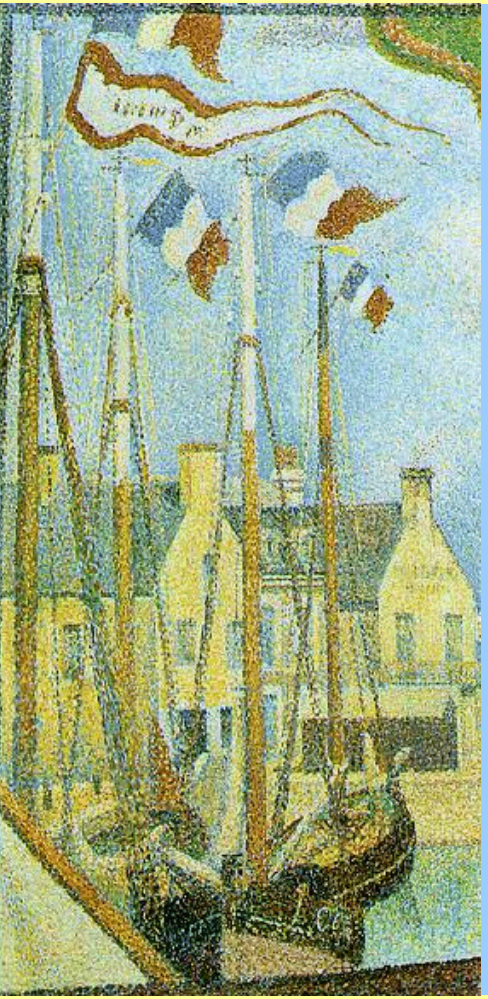
- Captures the developer’s intent
- Avoids duplicating past effort by providing alternatives already considered
- Avoids repeating past mistakes by documenting when something was tried and failed

SEURAT: Software Engineering Using RATionale



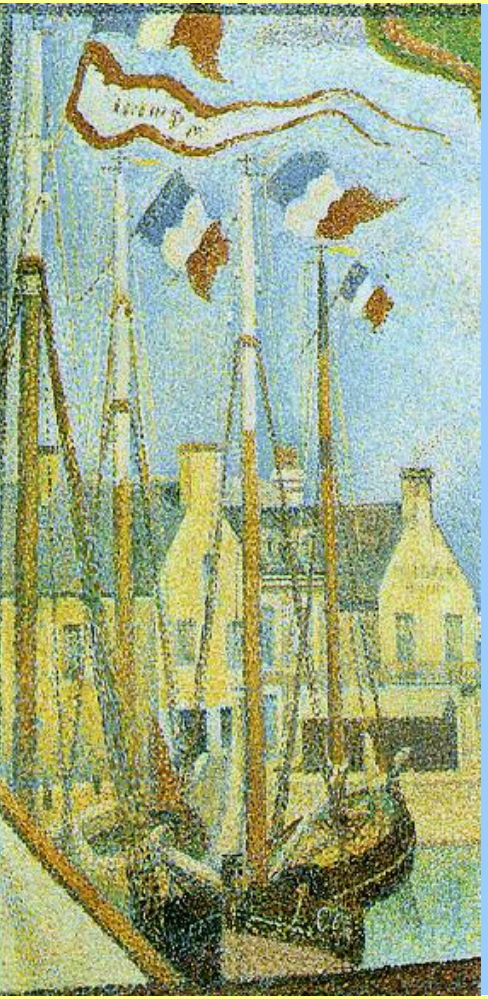
- Using rationale to assist in software development and maintenance:
  - verifying consistency and completeness of the rationale
  - evaluating the support for design alternatives
  - ensuring that rejected alternatives are not repeated
  - presenting applicable rationale to the maintainer to assist in modification
  - maintaining rationale consistency by propagating results of rationale modifications

SEURAT Capabilities



- Tight integration with development/maintenance environment
- Allows “what if analysis” of
  - changing design priorities
  - disabling assumptions
  - disabling requirements
- Supports traceability of requirements to decisions (and then to code)

Integration



- SEURAT is implemented as an Eclipse plug-in
  - Rationale is more likely to be used if the developer does not need to switch tools
  - Rationale can be directly associated with code and its presence indicated in the editor used by the developer to write and maintain code
- Rationale is stored in a MySQL database
  - Scalable to large amounts of rationale
  - SQL queries support inference and presentation

Java - CRSEnhancive.java - Eclipse Platform

File Edit Source Refactor Navigate Search Project Run Window Help

RationaleExplorer

- Rationale
  - Requirements
  - Decisions
    - How do we compare dates?
    - how do we know what the conference rooms are?
    - How is the user associated with the meeting
    - how to display error messages
      - error line on main display
        - easy to code
          - easier to code
            - IS Reduces Development Time
        - no extra user response needed
      - pop-up error box
    - how to display which user "owns" which meeting
    - how to represent a conference room
    - how to represent date?
    - how to send meetings to display
    - how to specify conference room location
    - how to specify meeting duration
    - how to specify/select room

RationaleExplorer | Hierarchy | Navigator

Package Explorer

- CRSEnhancive
  - meeting
  - schedule
  - schedulingSystem
  - ConferenceRoom.java
  - CRSEnhancive.java
  - EditUser.java
  - LoginUser.java
  - MessageBox.java
  - ScheduleMeeting.java
  - UserInfo.java

CRSEnhancive.java | ConferenceRoom.java | ScheduleMeeting.java

```
Label           weekLabel;
Button          nextWeek;
Button          nextMonth;
Label           status;

// Global Time
private int numHalfHours = 20; //was 28
private int numDays = 5;
private int interval = 30;
private int START_DAY = Calendar.MONDAY;
private int END_DAY = Calendar.FRIDAY;
private int START_TIME = 8; // 8 am

private int MONTH_SPAN = 28;
private int WEEK_SPAN = 7;

private static String USER_FILE = "users.txt";

private Frame frame;
```

Rationale Task List

Description	Rationale
Alt 'ascii file giving a list of room names' selected but not the b...	how do we know what the conferenc
Alt 'check overlap first' selected but not the best	meeting info before overlap ch
alt 'create new date class (meetingDate)' violates tradeoff 'Inc...	how to represent date?
Alt 'display meetings for current week' selected but not the best	which week to display when r
Alt 'display meetings for current week selected for which wee...	which week to display when r
Alt 'display name with meeting topic' selected but not the best	how to display which user "ow
Alt 'error line on main display' selected but not the best	how to display error messages
Alt 'just the meeting name' selected but not the best	what information to enter with meeting
Alt 'only owner can cancel meetings' selected but not the best	who is allowed to cancel meetings

Error Log | Console | Bookmarks | Rationale Task List | Tasks | Search

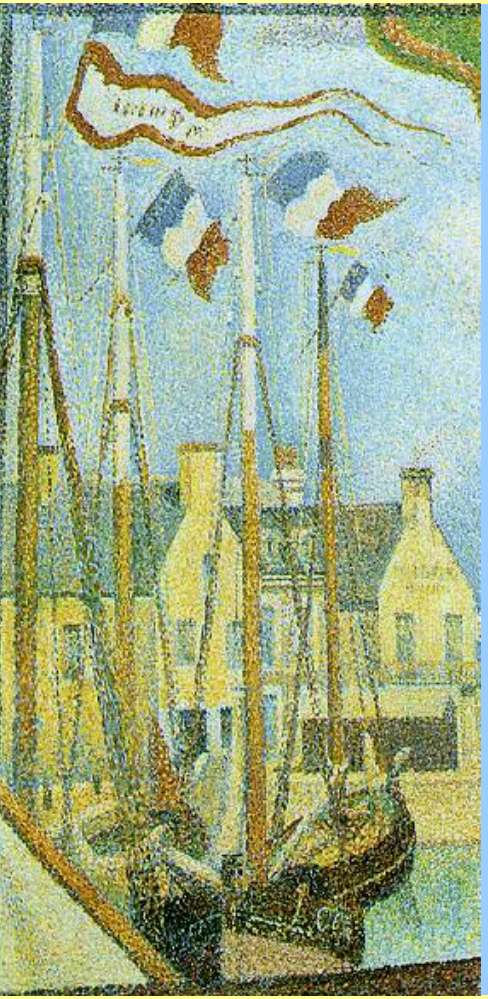
Rationale Explorer

Package Explorer

Source Code Editor

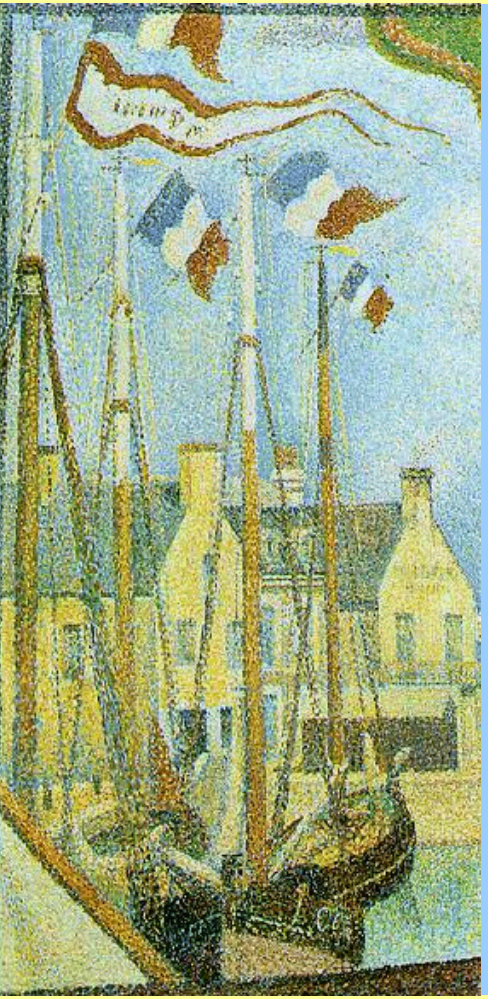
Rationale Task List

Evaluation Using Rationale



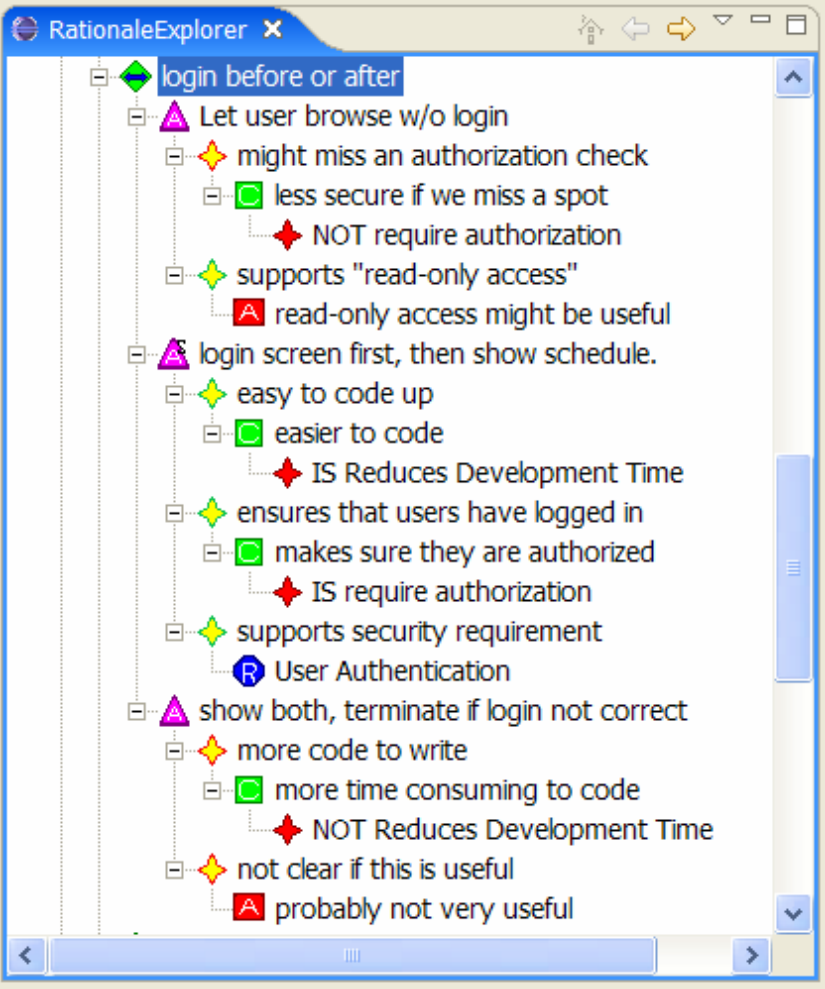
- Look for decisions with no selected alternative
- Look for selected alternatives with no supporting arguments
- Check for unanswered questions
- Evaluate alternatives and alert if weaker alternatives are selected
- Re-evaluate decisions after an assumption is disabled
- Re-evaluate decisions if argument priorities change
- Check for tradeoff violations
- Check for dependencies between alternatives
- Check for requirement violations

Rationale Representation



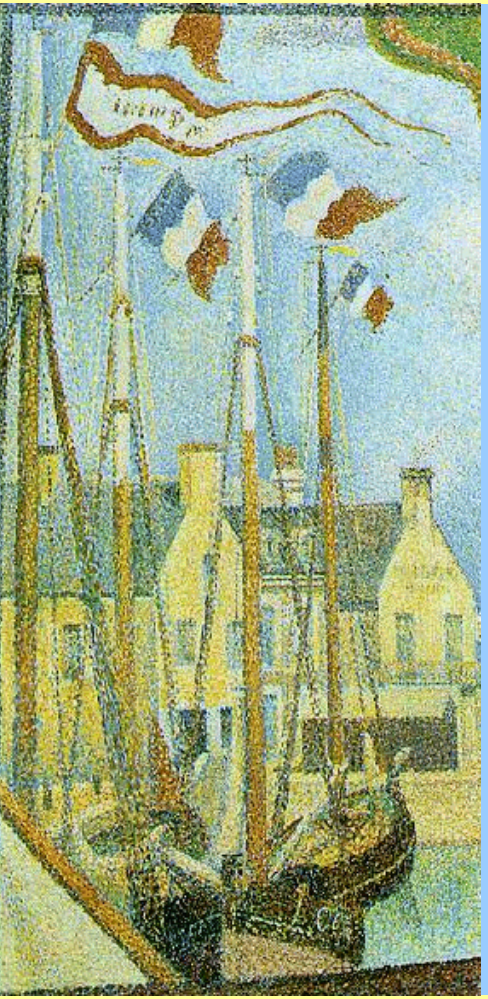
- Argumentation Representation
  - Semi-structured representation that is readable by machines and people
  - Captures the arguments for and against each alternative
  - Supports arguments about requirements, assumptions, claims (non-functional requirements), and other alternatives (dependencies)

Comparing Decision Alternatives



- Rationale Explorer shows alternatives for each decision
- Selected alternatives are denoted by an “S”
- Arguments are color coded: green is for, red is against, no color indicates rationale dependencies
- Arguments can refer to requirements, claims (non-functional requirements), assumptions, and relationships to other alternatives

Example: Analyzing Impact of Development Priorities



- Non-functional requirements (NFRs) may have a significant impact on the final software product
- What if these change? How would that affect the software?
- The rationale captures the impact of NFRs on the software and can be used to determine what should be changed if the importance of an NFR, or NFRs, changes

Analyzing Impact of Development Priorities

Common Argument Display

Ontology Entry	Total	For	Against
Reduces Development Time	11	8	3
minimizes keystrokes	4	4	0
intuitiveness	3	3	0
reduces coupling	2	2	0
Increases Scalability	1	1	0
minimizes connections to be set up	1	0	1
{is a | uses a} efficient algorithm	1	1	0
provides user guidance	1	1	0
{provides | supports} effective use of screen real-estate	1	1	0
{allows | supports} additional users	1	1	0
{provides | supports} data entry consistency	1	1	0
increases cohesion	1	0	1
Availability	1	1	0
require authorization	1	1	0
Increases Additivity	1	1	0
increases aesthetic value	1	1	0
Increases Flexibility	1	1	0
{provides | supports} code readability	1	1	0
Reduces Latency	1	0	1

Edit Close

- Can analyze rationale to determine which NFRs were most frequently used in arguments
- Compare those criteria to user and developer goals – do they match?
- In this example, the most commonly used NFR was “Reduces Development Time”

SEURAT - MeetingEventObject.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Window Help

RationaleExplorer

- Rationale - testrat
  - Requirements
  - Decisions
  - Tradeoffs
  - Co-occurrences
  - Argument-Ontology
    - Affordability Criteria
      - Development Cost
        - Uses Standard Tools and Environments
        - Uses Familiar Tools and Environments
        - Reduces Development Time
          - is component based
          - uses COTS/GOTS software
          - reduces customization
          - utilizes existing code developed i
          - uses automatically generated co
        - Reduces Project Sucess Risk
        - Reduces Prototyping Cost
        - Reduces Risk Analysis Cost
        - Reduces Component Integration C
        - Reduces Domain Analysis Cost
        - Reduces Inspection Cost
      - Deployment Cost
      - Operating Cost
      - Maintenance Cost

MeetingEventObjec... | MeetingBeanBeanIn... | MeetingObj.java | TableRegion.java

```
// Class        :  MeetingEventObject

package meeting;

import java.util.EventObject;


/**
 * This class represents a meeting event which either adds or removes a
 * from the schedule. It is used to tell the schedule to add or remove a
 * @author jburge
 *

ntObject extends EventObject {


int ADD = 1;
int CANCEL = 2;
```

OntEntry Information

Name: Reduces Development Time

Description: Reduces the amount of time and effort to develop the software

Importance: Not

Save | Cancel

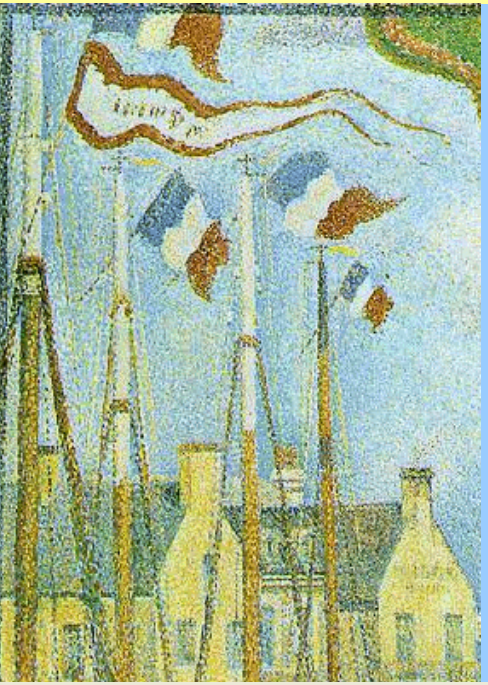
Package Explorer

- CRSCorrective
  - meeting
  - schedule
  - schedulingSystem
    - ConferenceRoom.java
    - CRSCorrective.java
    - EditUser.java
    - LoginUser.java
    - MessageBox.java
    - ScheduleMeeting.java
    - UserInfo.java

Problems | Javadoc | Declaration | Bookmarks | Rationale Task List

Description	Rationale	Type
Alt 'ascii file giving a list of room names' selected but not the b...	how do we know what the c...	Decision

Analyzing Impact of Development Priorities

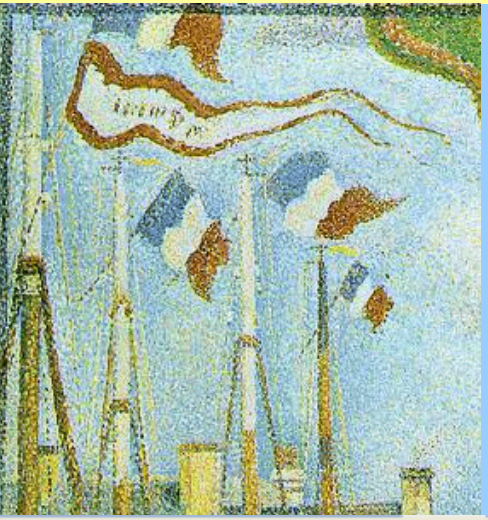


- Initially, there was only one warning that the best alternative was not selected
- Two more warnings appear if “Reduces Development Time” becomes unimportant

Problems | Javadoc | Declaration | Bookmarks | Rationale Task List

Description	Rationale	Type
Alt 'ascii file giving a list of room names' selected but not the best	how do we know what the conference rooms ...	Decision
Alt 'send individually' selected but not the best	how to send meetings to display	Decision
Alt 'start date' selected but not the best	what should be key for schedule table	Decision

Finding the Implementation



- Bookmark List
  - Lists the file, folder, and line number
  - Double-clicking brings up the code in the editor

Problems | Javadoc | Declaration | Bookmarks | Rationale Task List

7 items

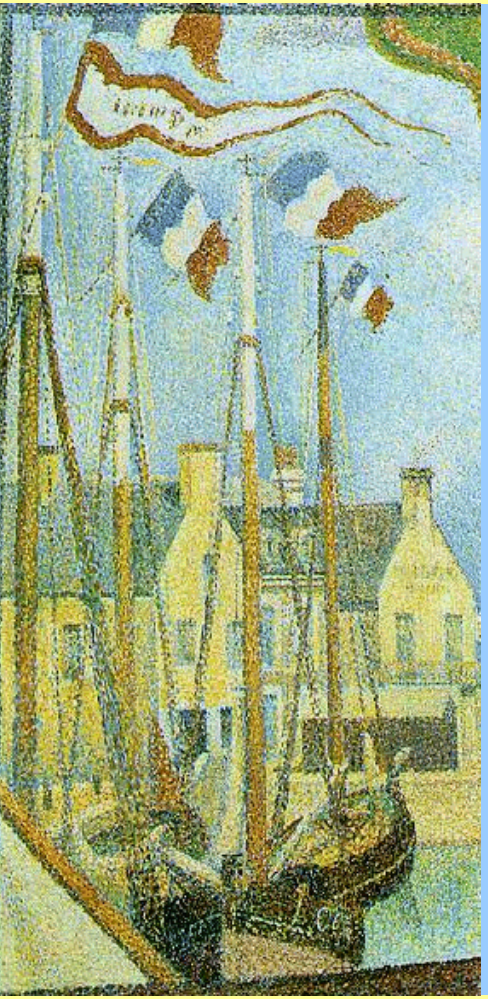
Description	Resource	In Folder	Location
Alt: 'user name stored as part...	MeetingObj.java	CRSCorrective/meeting	line 21
Alt: 'start date'	MeetingBean.java	CRSCorrective/meeting	line 132
Alt: 'start date'	MeetingBean.java	CRSCorrective/meeting	line 157
Alt: 'send individually'	MeetingBean.java	CRSCorrective/meeting	line 67
Alt: 'error line on main display'	CRSCorrective.java	CRSCorrective/schedulingSystem	line 684
Alt: 'Custom equals method'	MeetingDate.java	CRSCorrective/meeting	line 227
Alt: 'ascii file giving a list of roo...	CRSCorrective.java	CRSCorrective/schedulingSystem	line 430

Finding the Implementation

MeetingDate.java | CRSCorrective.java | ScheduleMeeting.java | MeetingBean.java

```
/**
 * Loads all meetings occuring between the start and end MeetingDate. T
 * generating a meeting event for each retrieved meeting.
 *
 * @param start
 * @param end
 */
    public void loadMeetings(MeetingDate start, MeetingDate end) {

    Enumeration meetingEnum;
    MeetingObj newMeeting;

    meetingEnum = retrieveMeetings(start, end);

    while (meetingEnum.hasMoreElements())
    {
        newMeeting = (MeetingObj) meetingEnum.nextElement();
        // create and send meeting event
        MeetingEventObject meo = new MeetingEventObject (this,
           newMeeting.getStart(), newMeeting.getEnd(),
           newMeeting.getInfo(), newMeeting.getOwner(), MeetingEventObj
        fireAction(meo);
    }

    }
```

SEURAT Summary



- Demonstrated that with appropriate tool support, rationale can provide useful support to the software developer:
  - Demonstrated uses of rationale that go beyond browsing and presentation
  - Integrated rationale support with a standard software development environment