Software Navigation

Thomas Zimmermann

Oral exams

- ☐ All slides
- ☐ List of ten papers
- ☐ Guest lectures
- ☐ Your course notes

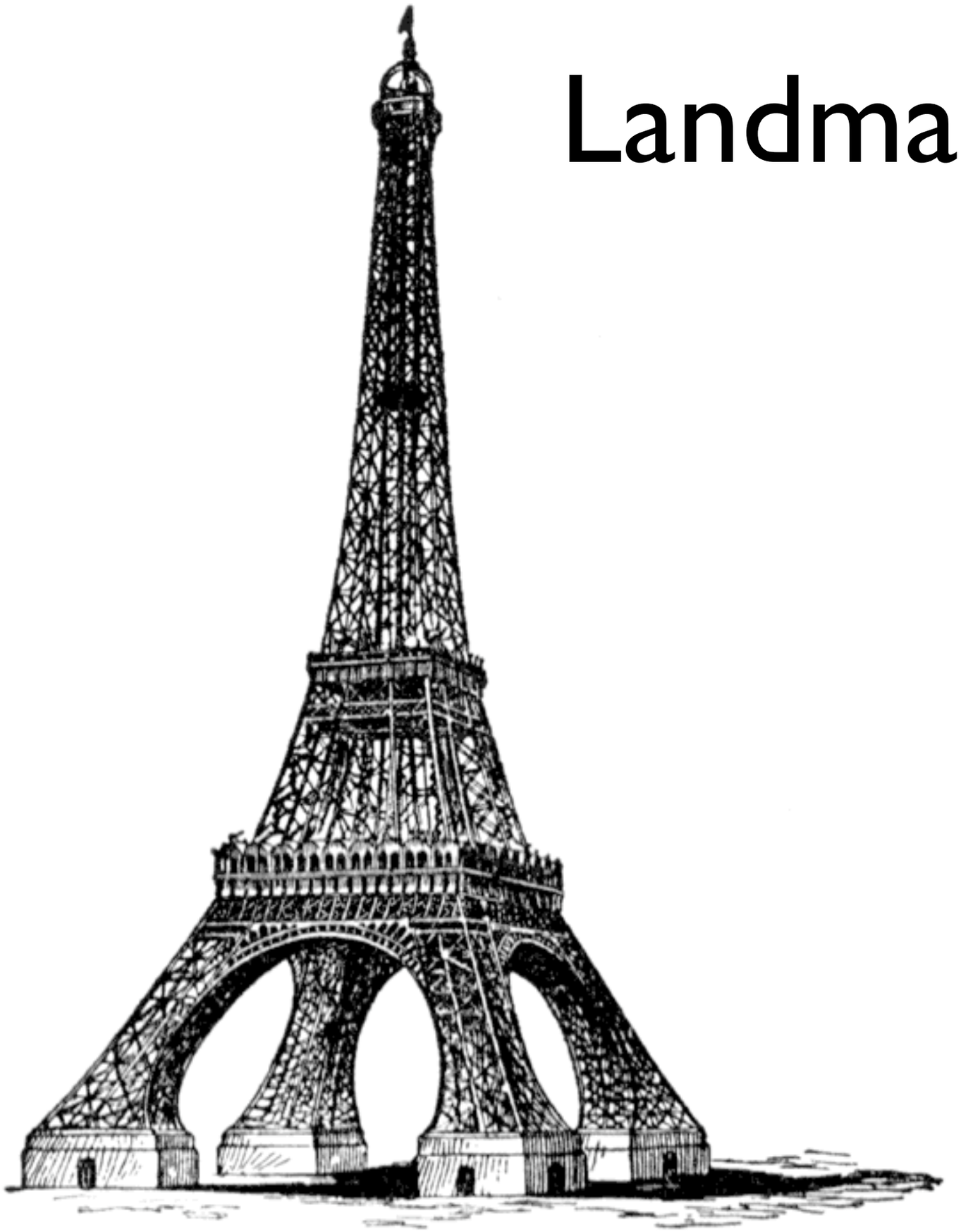
Concept assignment problem

Identify how high-level concepts are implemented in source code.

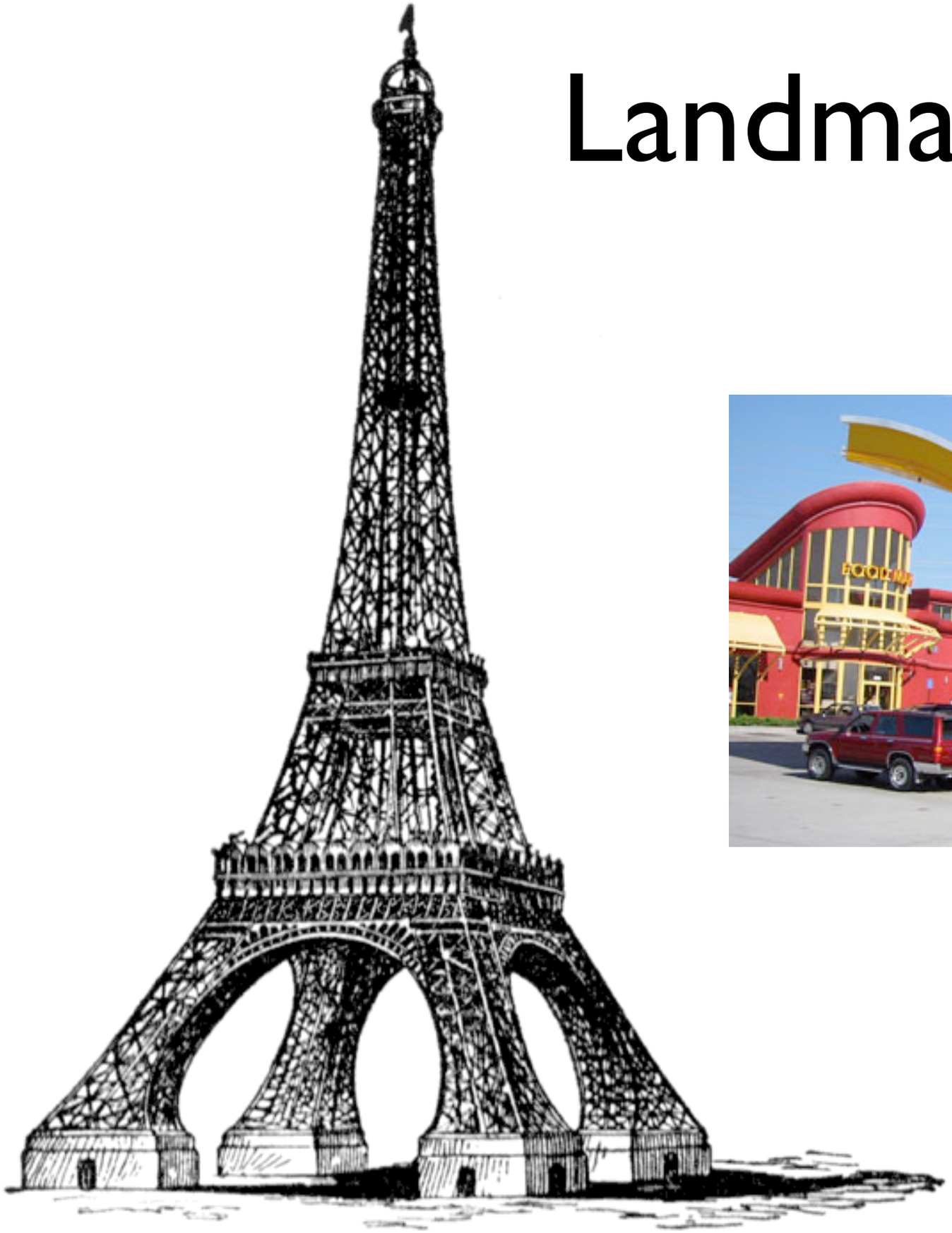
For software modification: understand the concepts that are associated to the task.

Software navigation

Landmarks

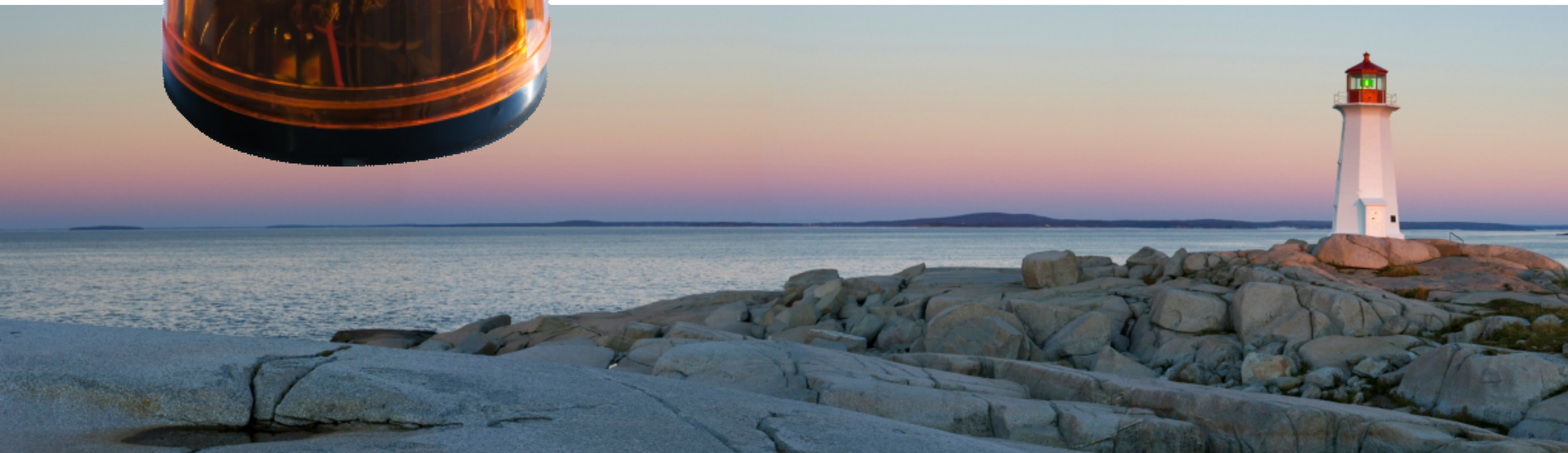


Landmarks



Beacons

Beacons

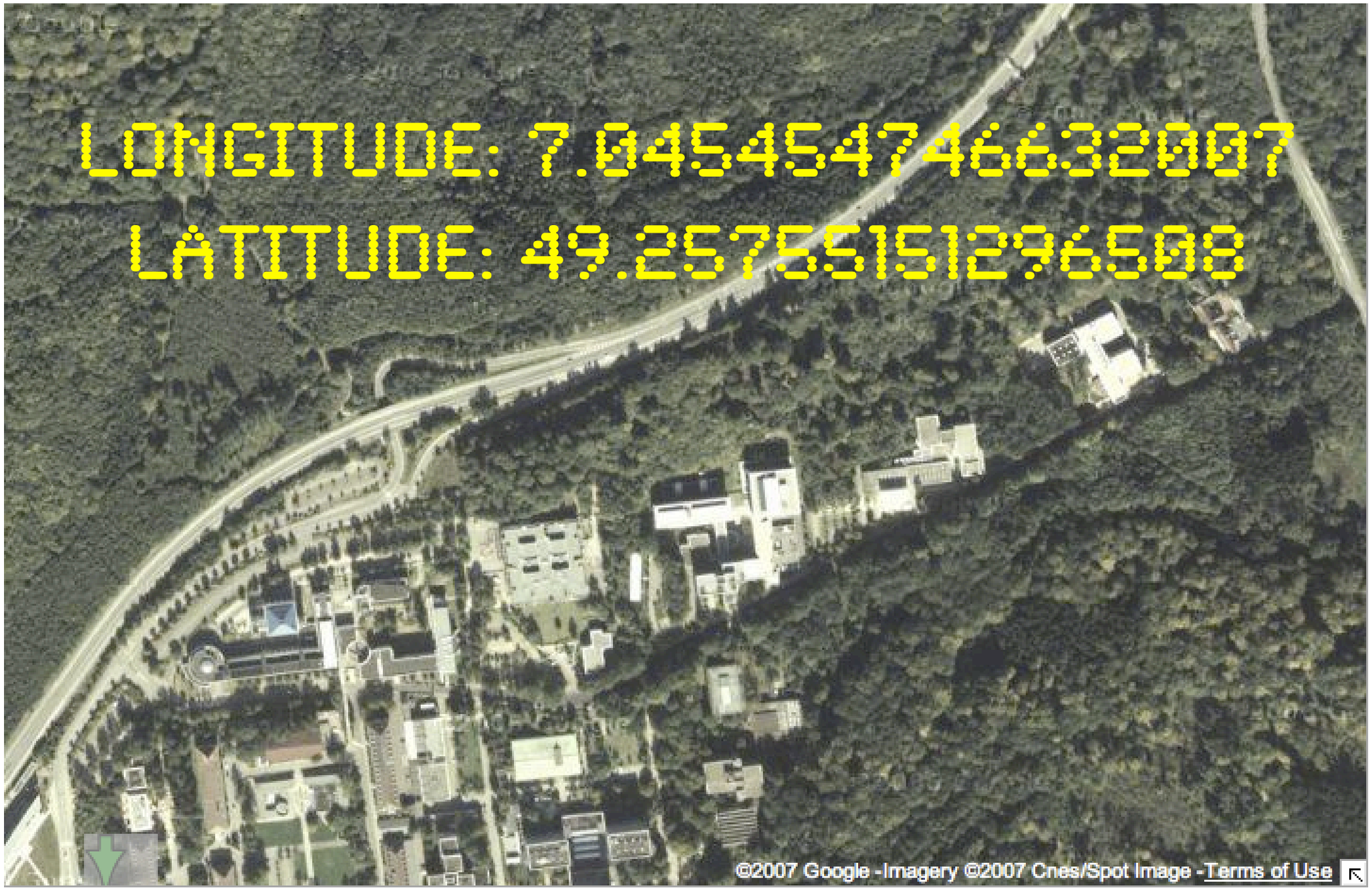


Landsmarks and beacons

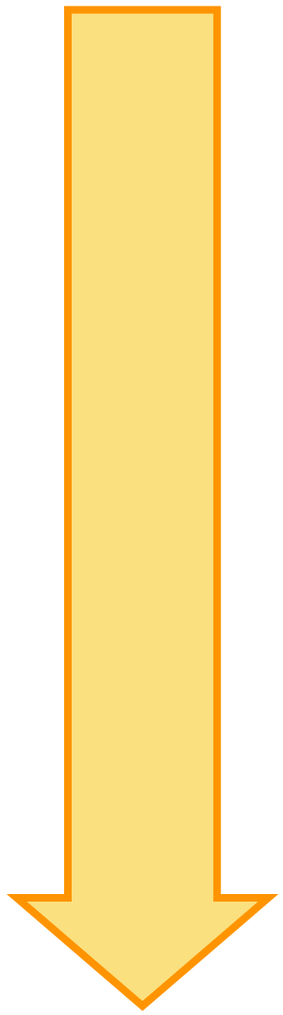
```
public static void bubbleSort(int[] array) {

    for (int i = array.length - 1; i > 0; i--) {
        for (int j = 0; j < i; j++) {
            if (array[j] > array[j + 1]) {
                int temp = array[j];
                array[j] = array[j + 1];
                array[j + 1] = temp;
            }
        }
    }
}
```

What would be a landmark and what would be a beacon?

Waypoints


LONGITUDE: 7.045454746632007
LATITUDE: 49.25785151296508
©2007 Google -Imagery ©2007 Cnes/Spot Image -Terms of Use

Learning an environment



Landmark knowledge

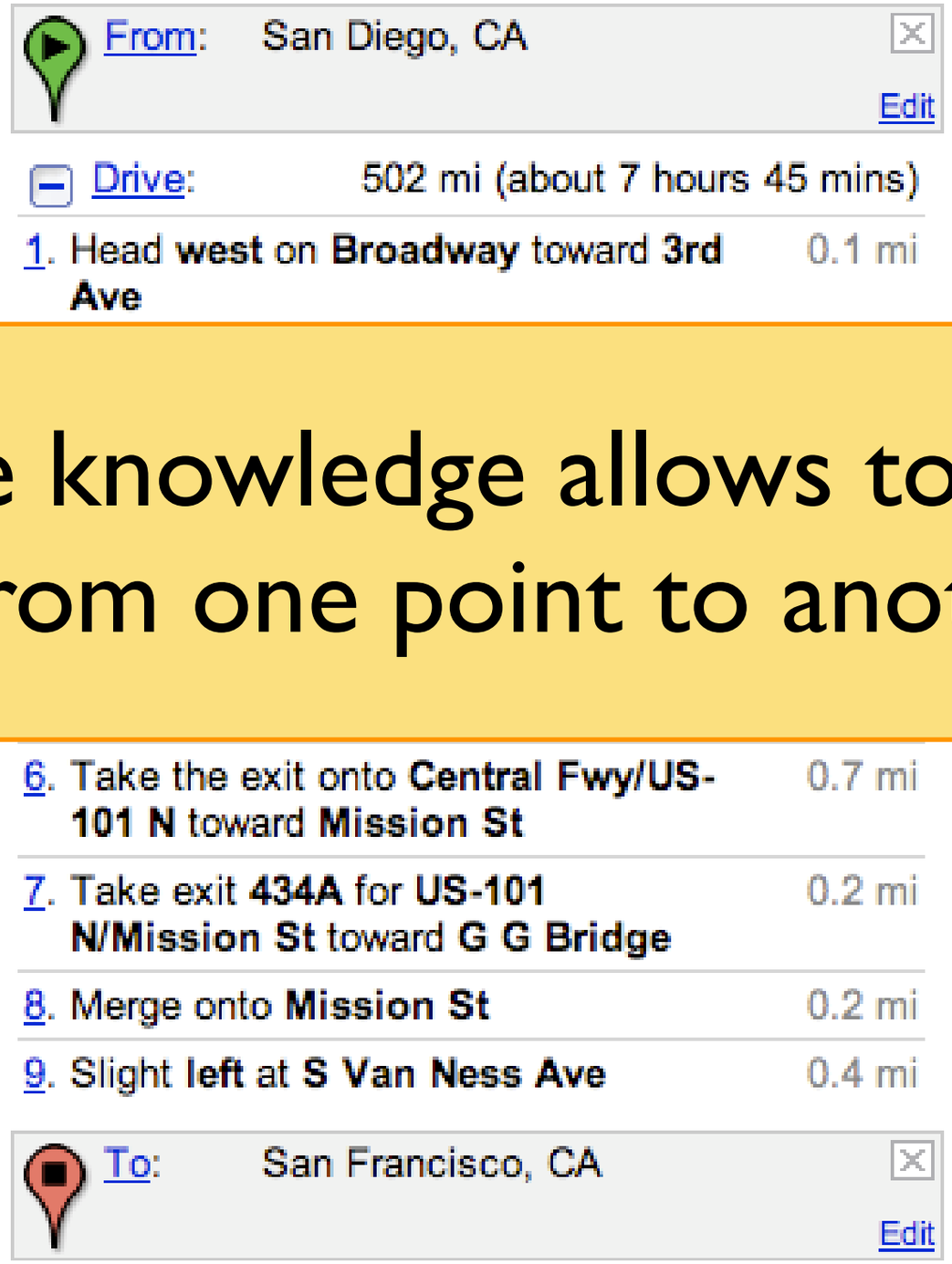
Route knowledge

Survey knowledge

Route knowledge

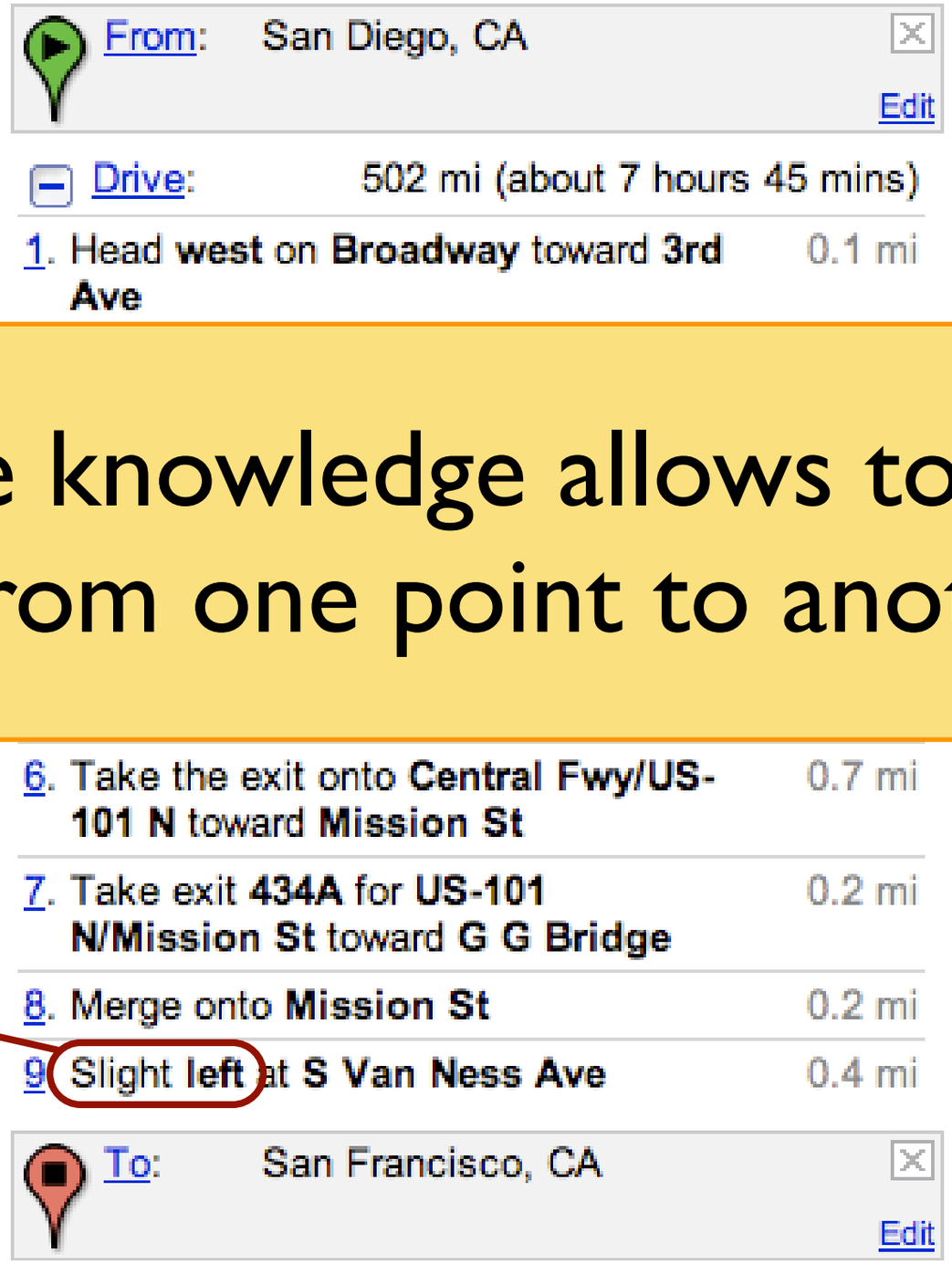
Route knowledge allows to navigate from one point to another.

Route knowledge



Route knowledge allows to navigate from one point to another.

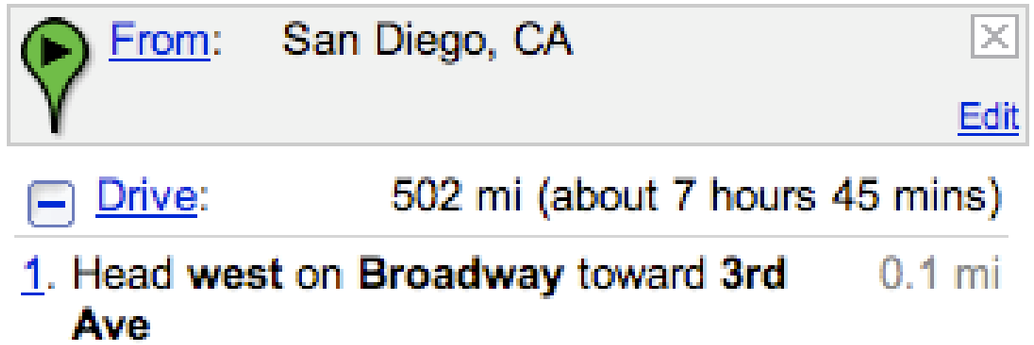
Route knowledge



Route knowledge allows to navigate from one point to another.

Navigational actions

Route knowledge



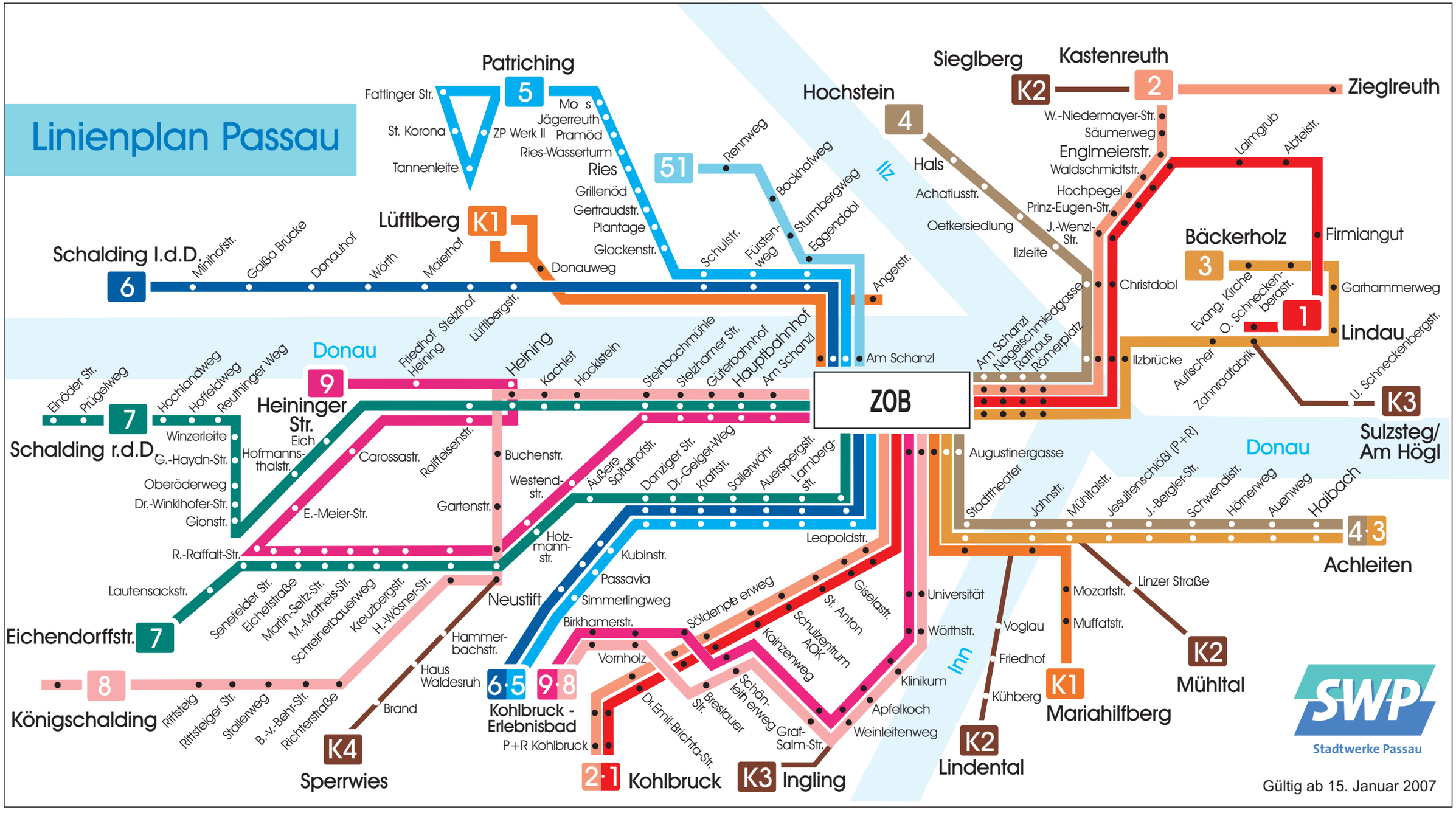
Route knowledge allows to navigate from one point to another.

Navigational actions

Landmarks

Route knowledge

Route knowledge



Survey knowledge

Survey knowledge is analogous to a map. except that it allows the navigator to adopt the most convenient perspective.

Spatial properties, but distortions can occur.

Elements in cognitive maps

Paths: channels for navigator movement

street, canal, transit line

Edges: indicate district limits

fence, river

Districts: reference points

neighborhood

Nodes: focal points for travel

town square, buildings

Landmarks: “special” reference points

statue, tower

What makes a landmark?

What makes a landmark?

Memorizable

Memory for location

What makes a landmark?

Significant height

Visible

Memorizable

Complex shape

Large, visible signs

Memory for location

What makes a landmark?

Significant height

Visible

Memorizable

Complex shape

Large, visible signs

Bright exterior

Unique
color, texture

Memory for
location

Visible

Expensive materials

Navigation in codespace

Many similarities to physical world:

Manifestation (can be viewed and manipulated)

Identifiable constructs (objects)

Various levels of abstraction (hierarchies)

Direction

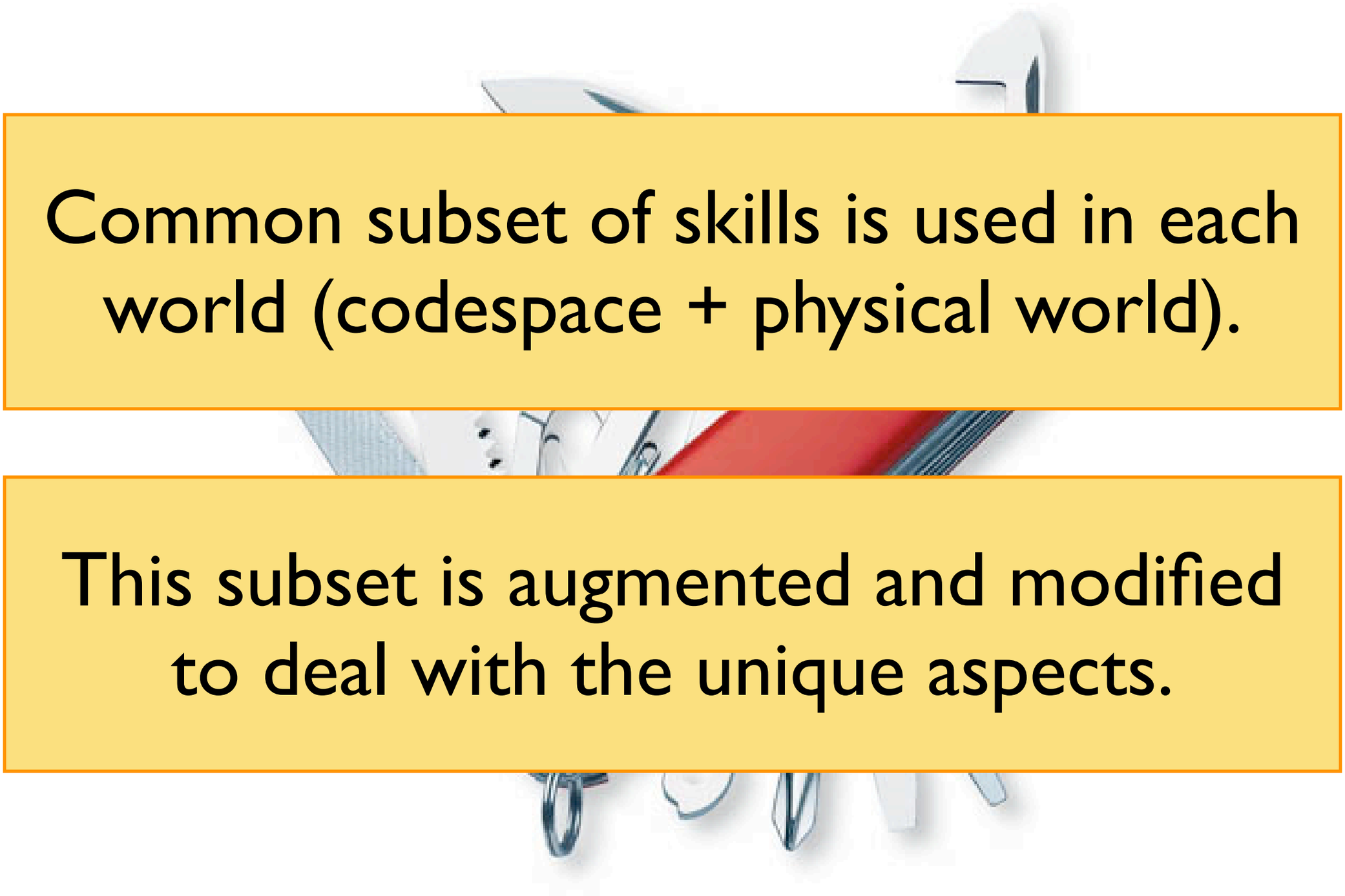
Distance (transit technqiues)

Swiss army knife hypothesis

Swiss army knife hypothesis

Common subset of skills is used in each world (codespace + physical world).

Swiss army knife hypothesis



Common subset of skills is used in each world (codespace + physical world).

This subset is augmented and modified to deal with the unique aspects.

Software navigation

Where do I start?

Software navigation

Where do I start?

(Self location)

Where am I?

Software navigation

Where do I start?

(Self location)
Where am I?

When's the next turn?
(Where do I have to look/change next?)

Software navigation

Where do I start?

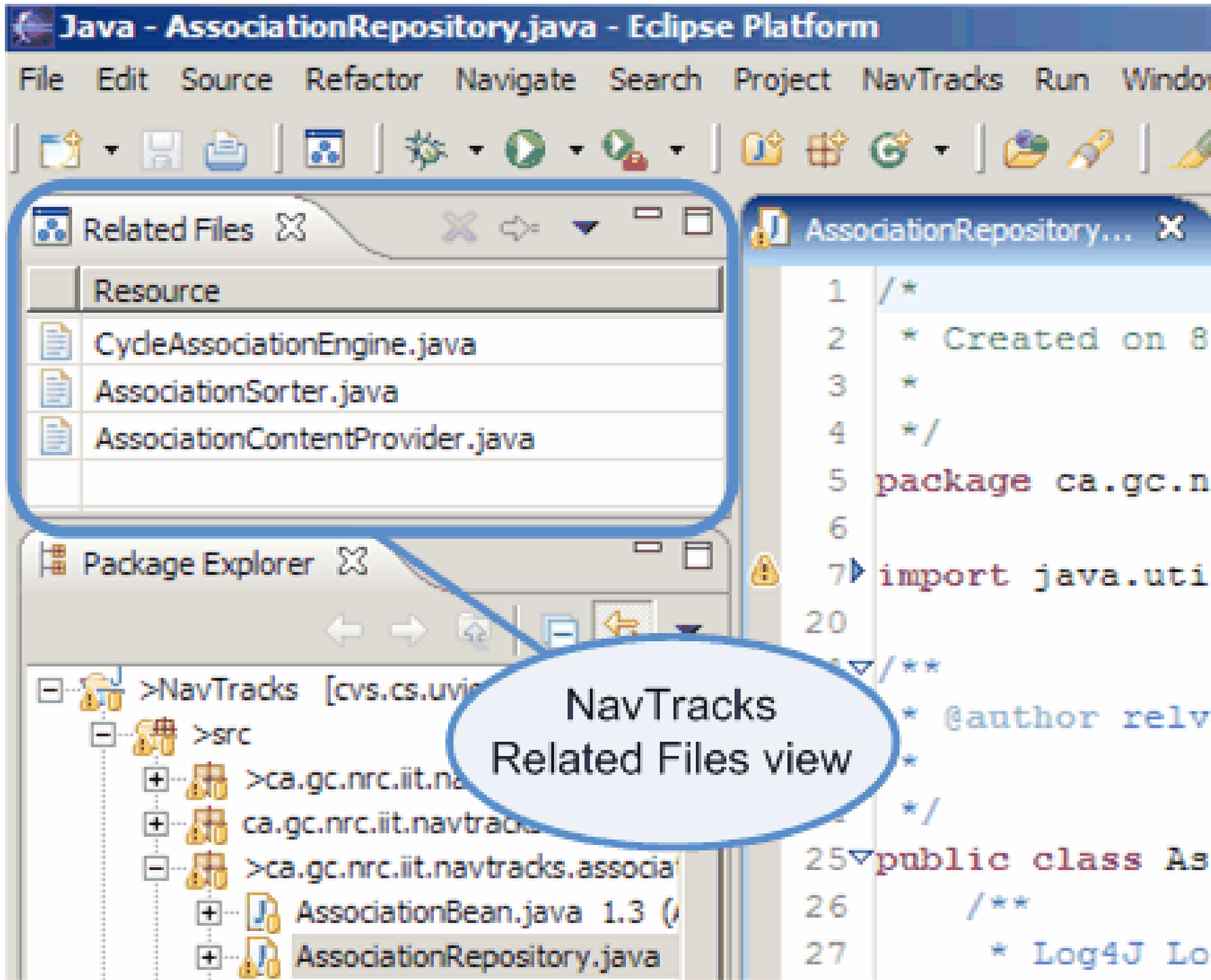
(Self location)
Where am I?

When's the next turn?
(Where do I have to look/change next?)

Have I reached?
(Have I completed my task?)

NavTracks

Keep track of navigation history to support browsing



Singer et al. (ICSM 2005)

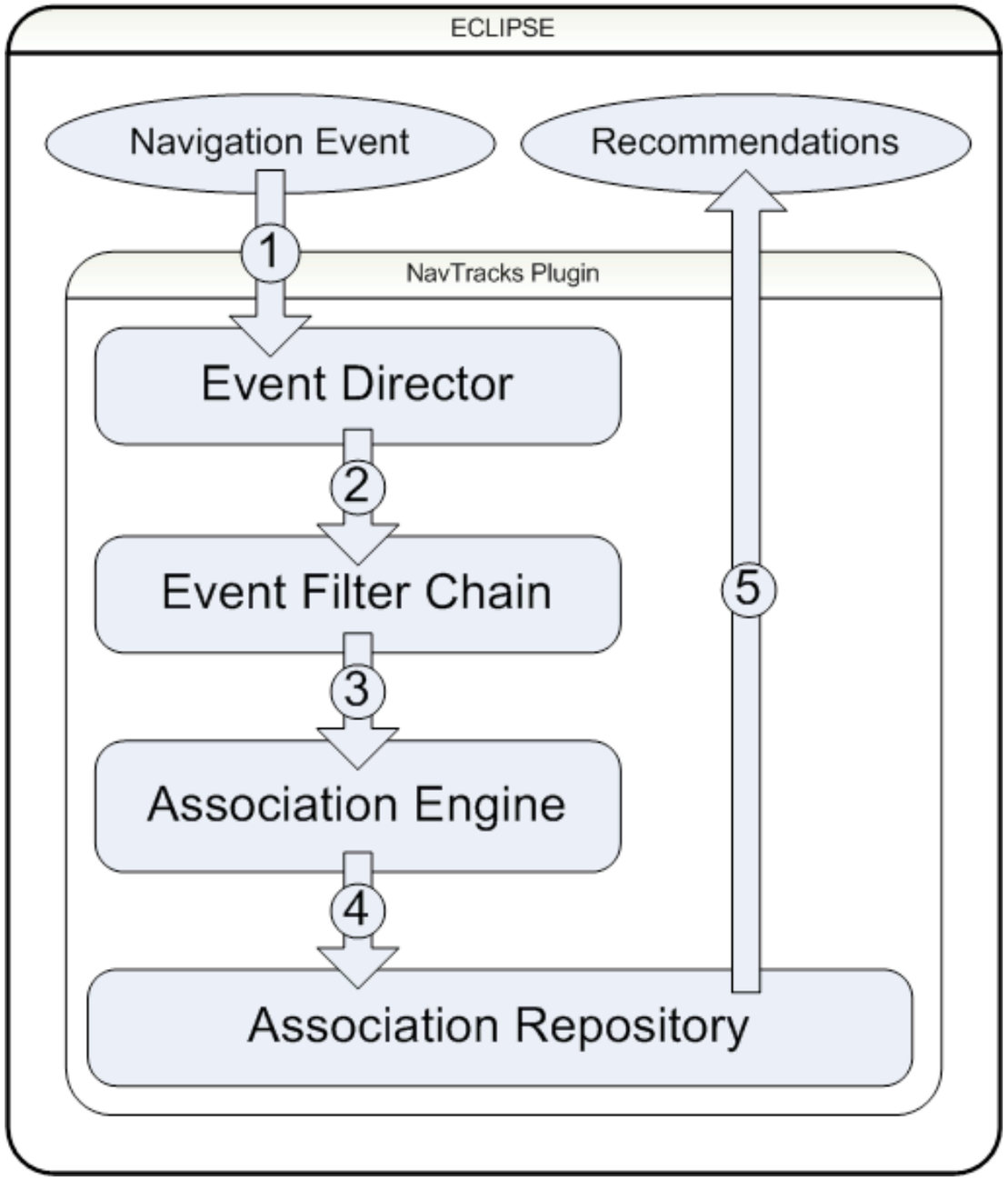
Requirements

Non-disruptive. Should not disrupt nor affect the performance of developers.

Current patterns. Prefer locally available data over historical traces (e.g., version archives).

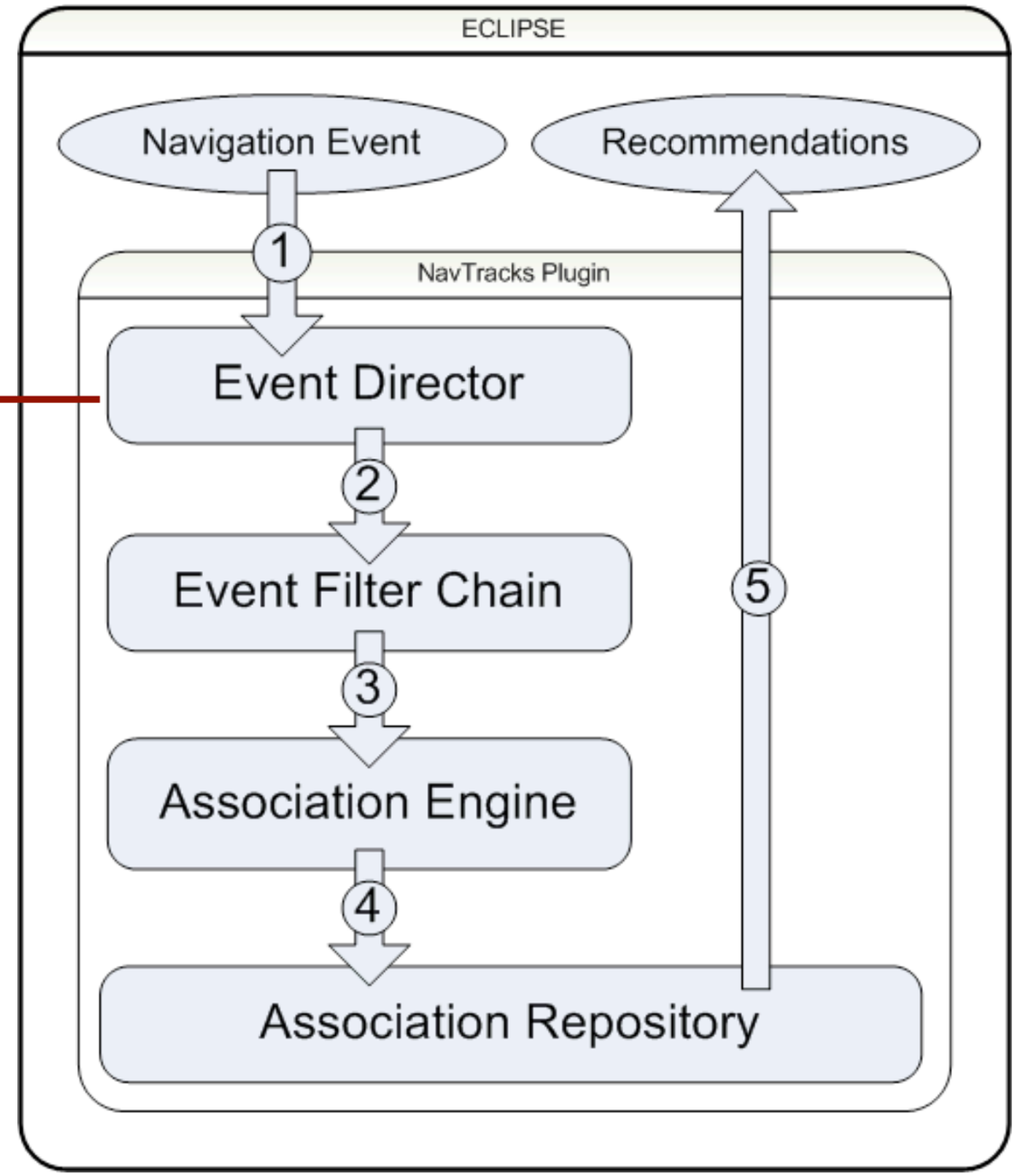
Approximate but efficient. Like Google, be efficient and accurate most of the time.

Architecture



Architecture

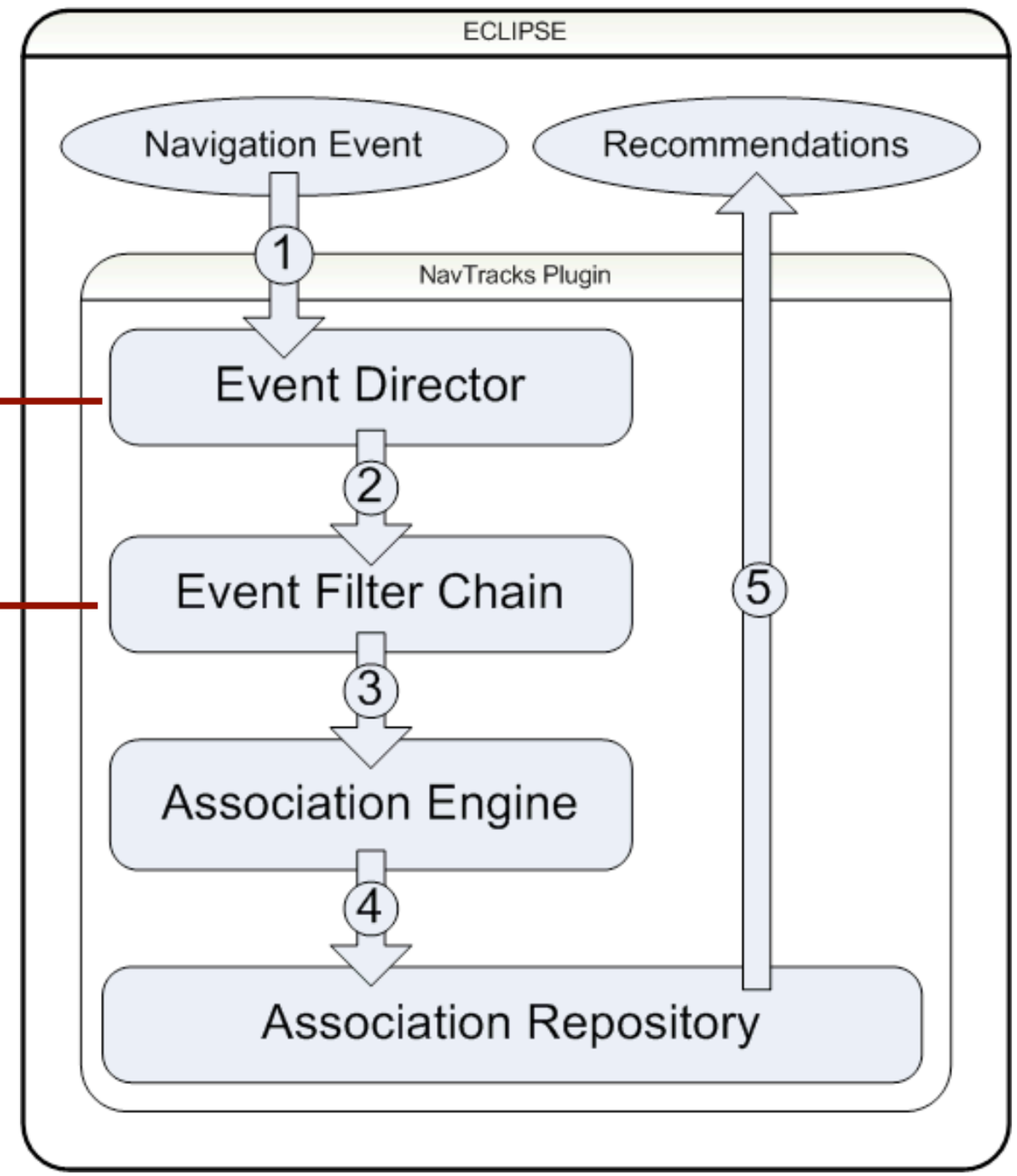
Record events



Architecture

Record events

Filter jitter and duplication

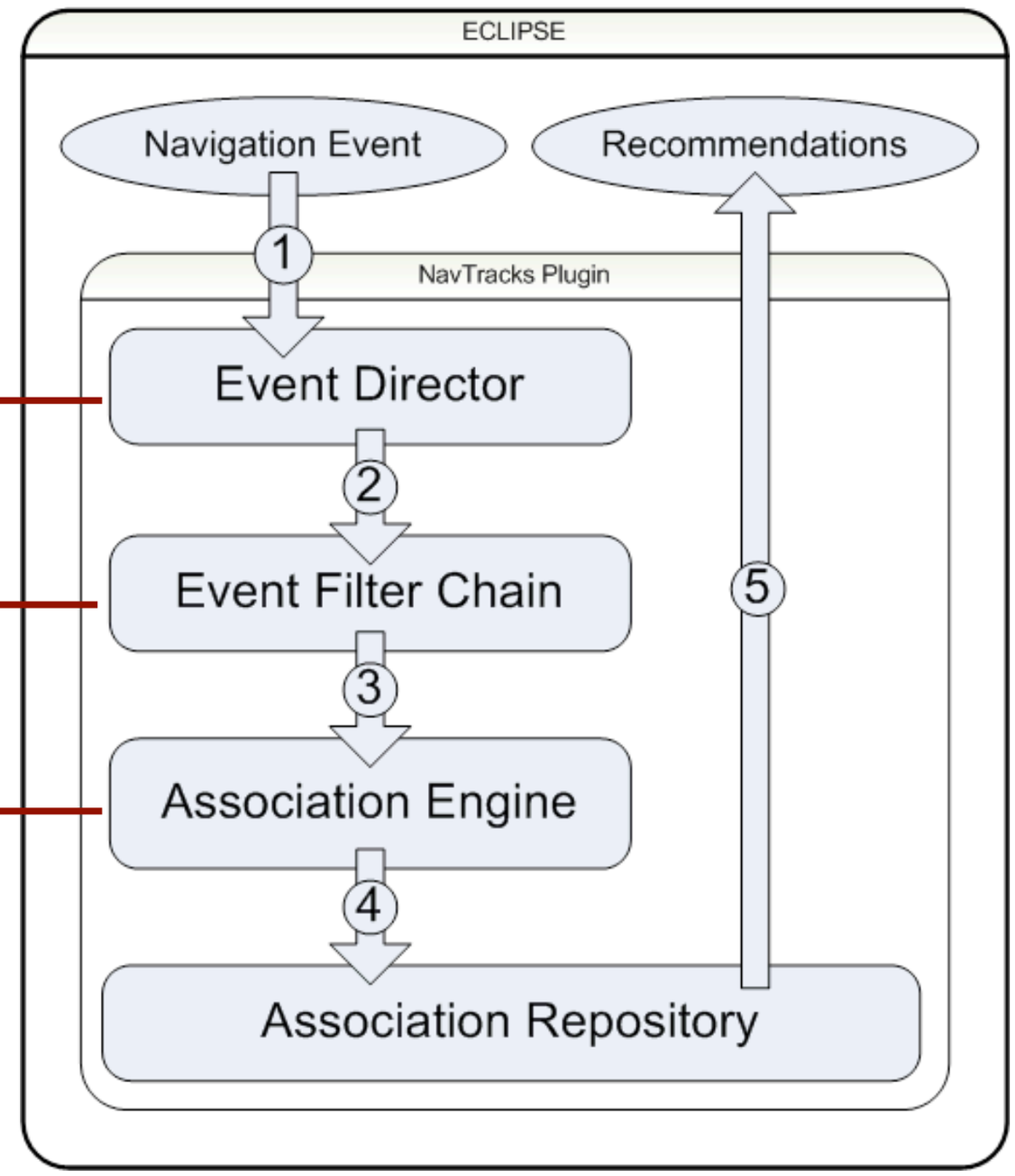


Architecture

Record events

Filter jitter and duplication

Compute associations



Associations

Event Step	Event Window (size n=4)	Associations Detected (minimum cycle length k=3)
Step 1:	A	No Cycles

Associations

Event Step	Event Window (size n=4)	Associations Detected (minimum cycle length k=3)
Step 1:	A	No Cycles
Step 2:	A C	No Cycles

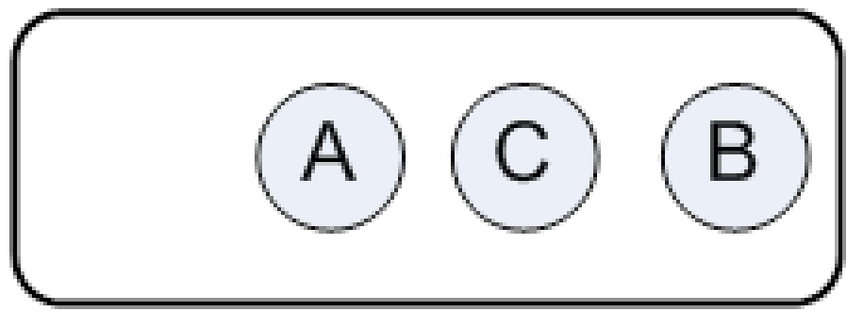
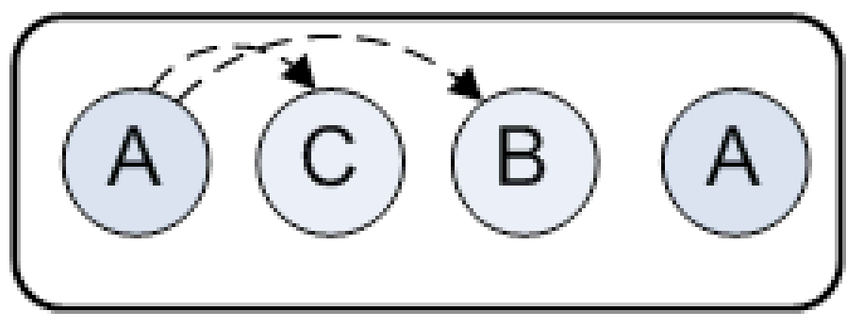
Associations

Event Step	Event Window (size n=4)	Associations Detected (minimum cycle length k=3)
Step 1:	A	No Cycles
Step 2:	A C	No Cycles
Step 3:	A C B	No Cycles

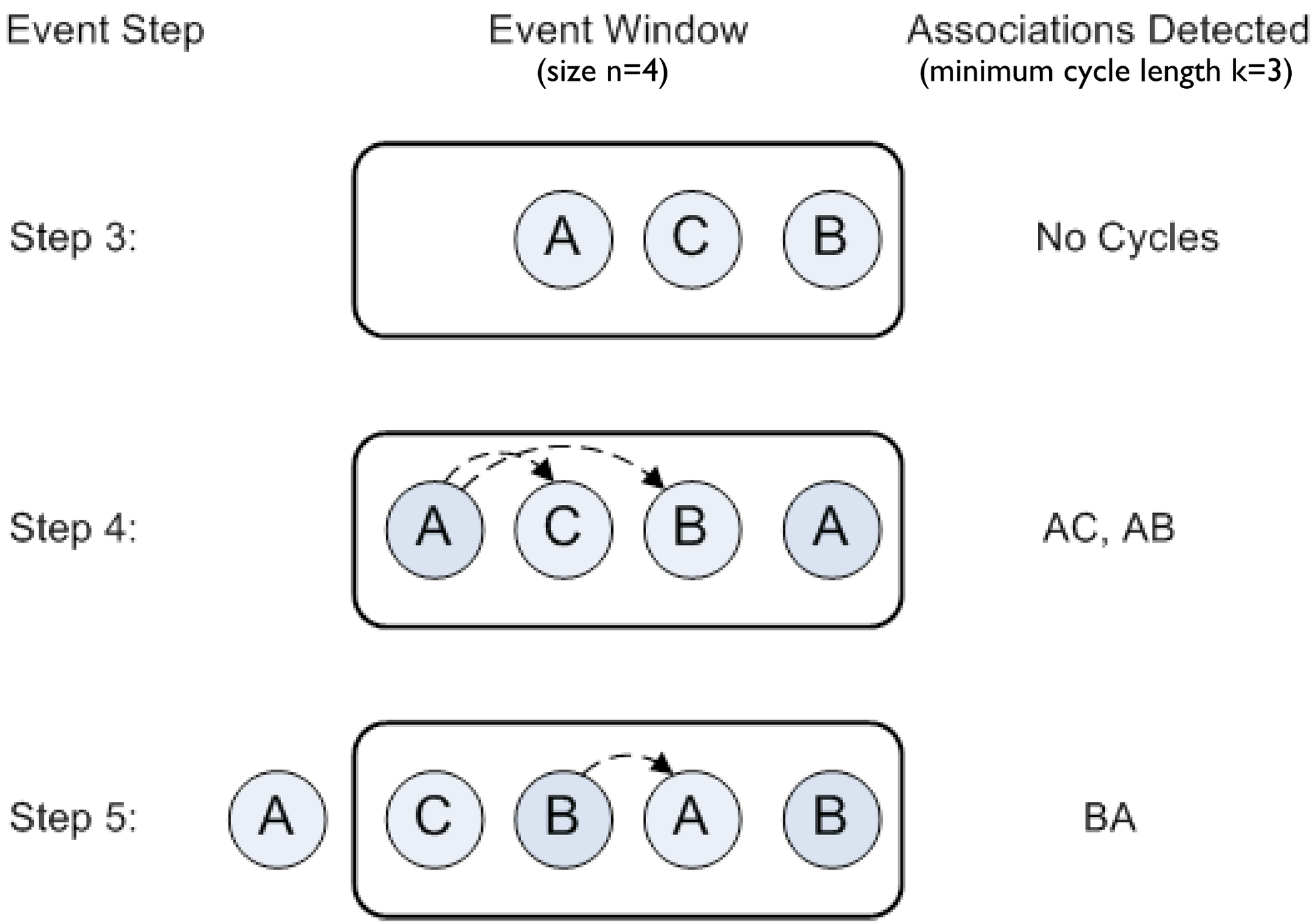
Associations

Event Step	Event Window (size n=4)	Associations Detected (minimum cycle length k=3)
Step 3:	A C B	No Cycles

Associations

Event Step	Event Window (size n=4)	Associations Detected (minimum cycle length k=3)
Step 3:		No Cycles
Step 4:		AC, AB

Associations



Continuous evaluation

Continuous evaluation and training process on the events of three developers: 36%, 35%, 16%

Event Class	D3	D1	D2	Average
2-6 occurrences	25%	23%	17%	22%
7-11 occurrences	52%	50%	32%	45%
11-16 occurrences	46%	51%	31%	43%
12-21 occurrences	60%	31%		45%
> 21 occurrences	43%	43%		43%

User study

Experiences and feedback from 5 developers.

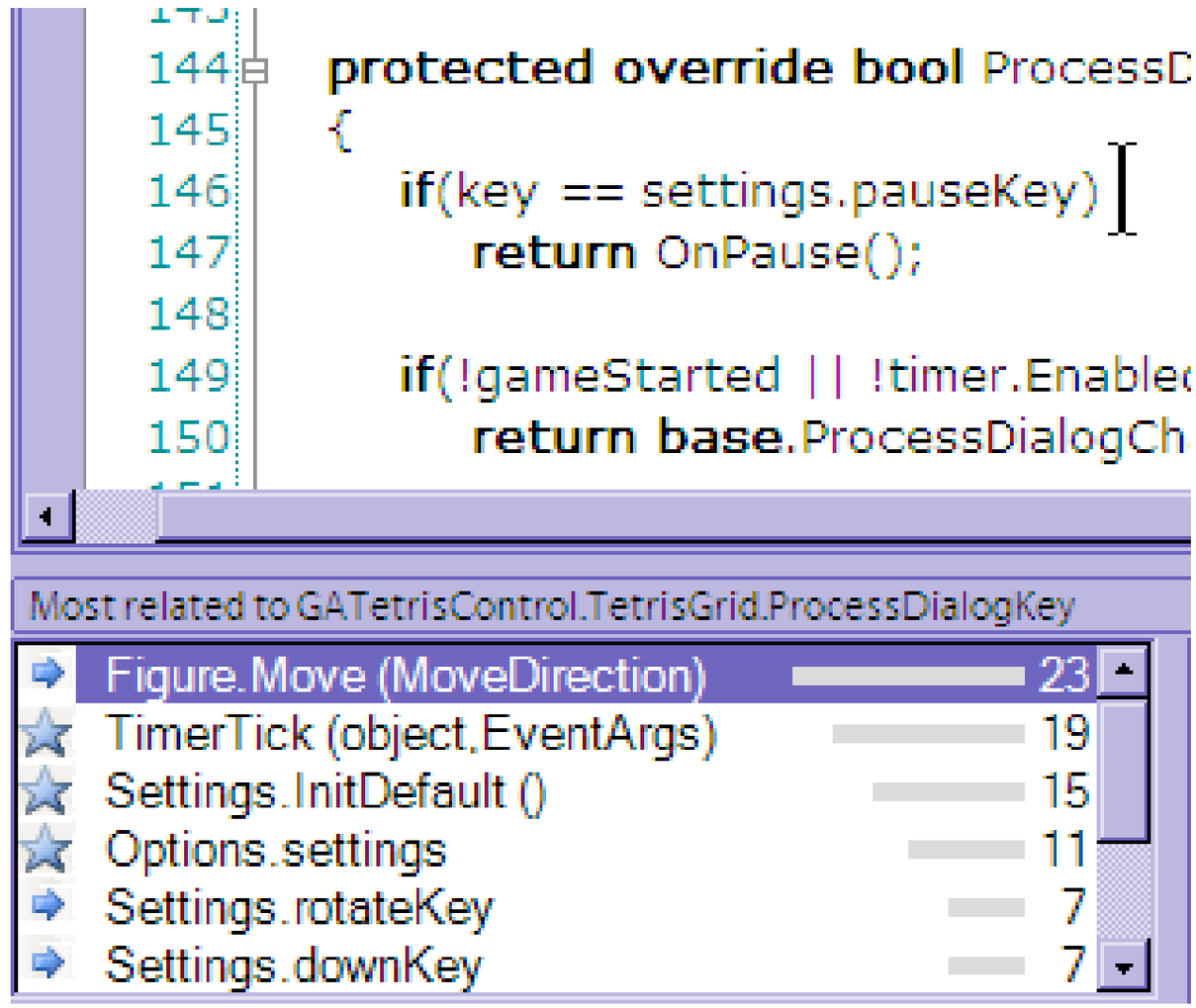
Newcomer & New system development use. Tool provides a memory aid for related files.

Wanderer use. Tool did not help for refactoring (repetitive copy, paste, and modify cycle).

Navigation use. Tool supported contextually different navigation paths.

Team Tracks

Team Tracks is used at Microsoft

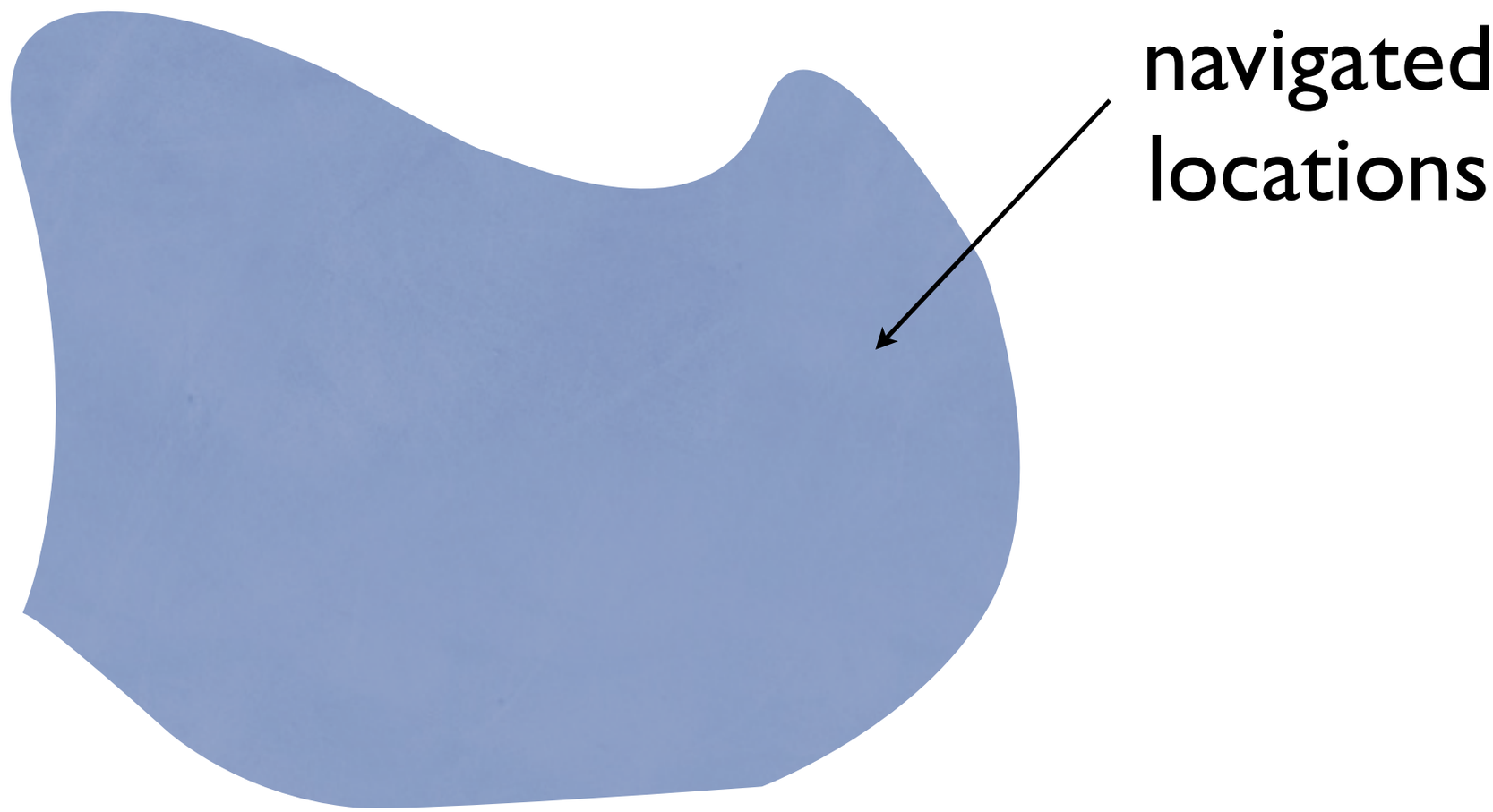


Team Tracks = NavTracks on Teams

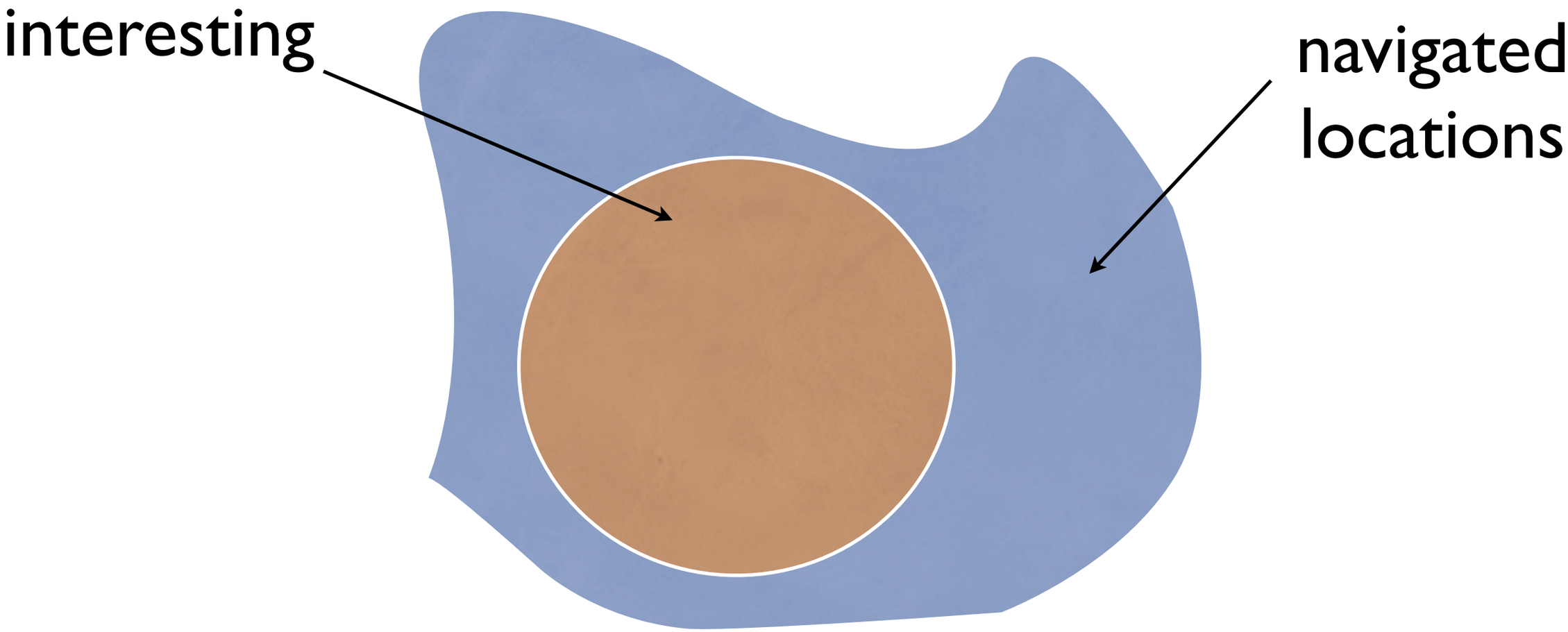
DeLine et al. (VL/HCC 2005, SOFTVIS 2005)

Navigation and changes

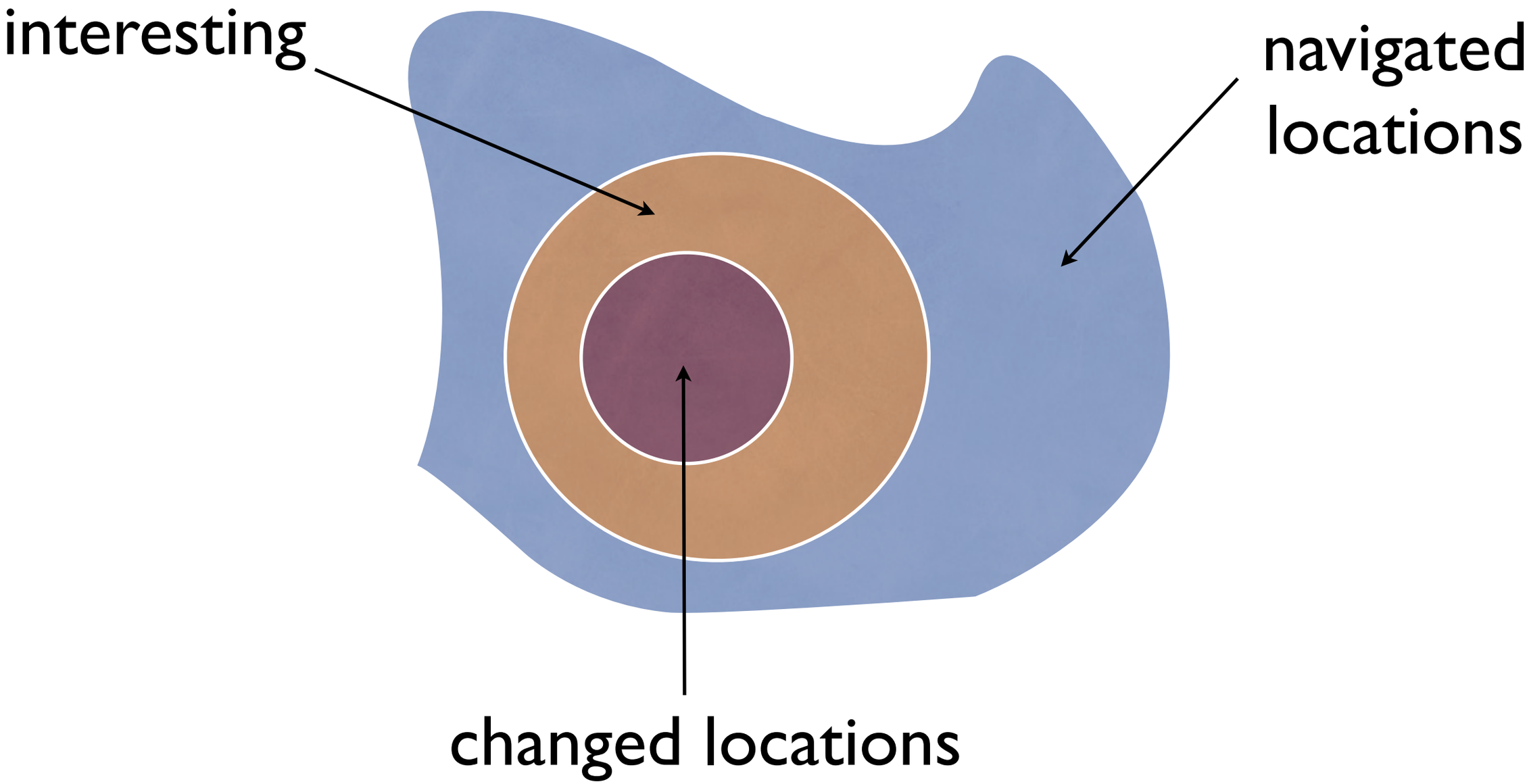
Navigation and changes



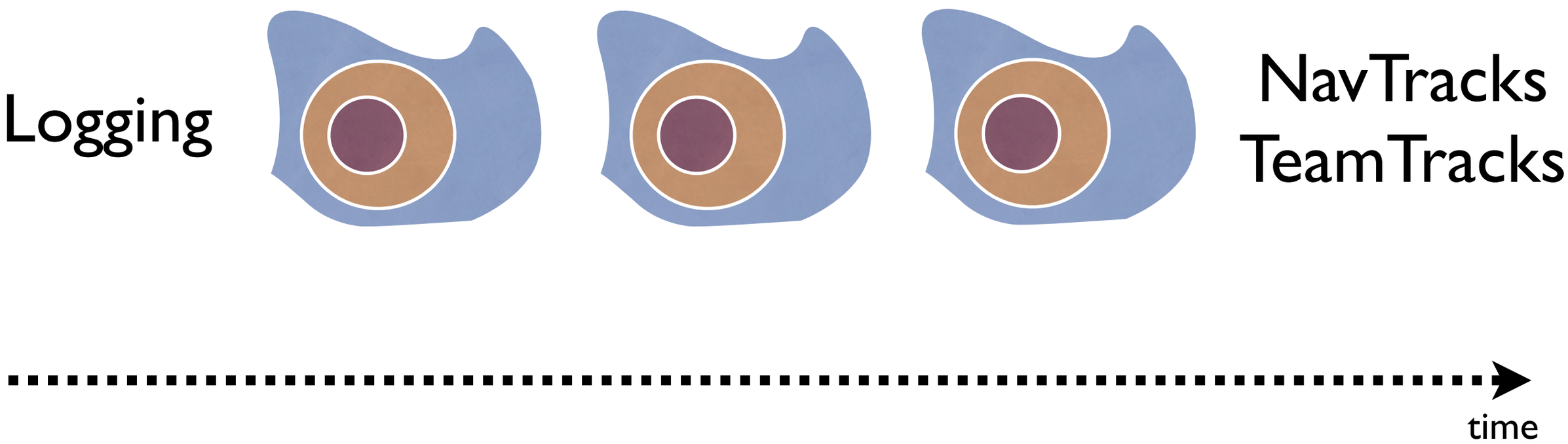
Navigation and changes



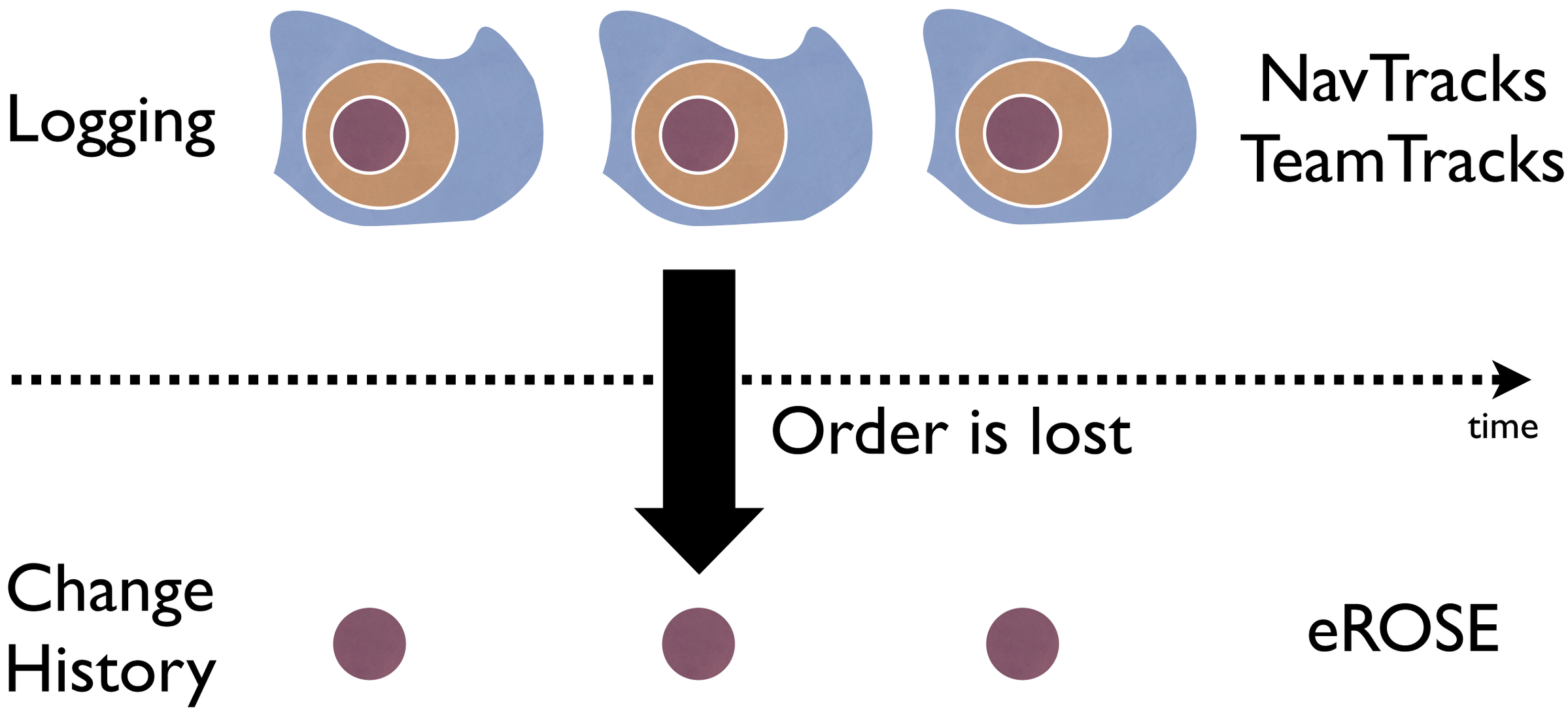
Navigation and changes



Navigation and changes



Navigation and changes



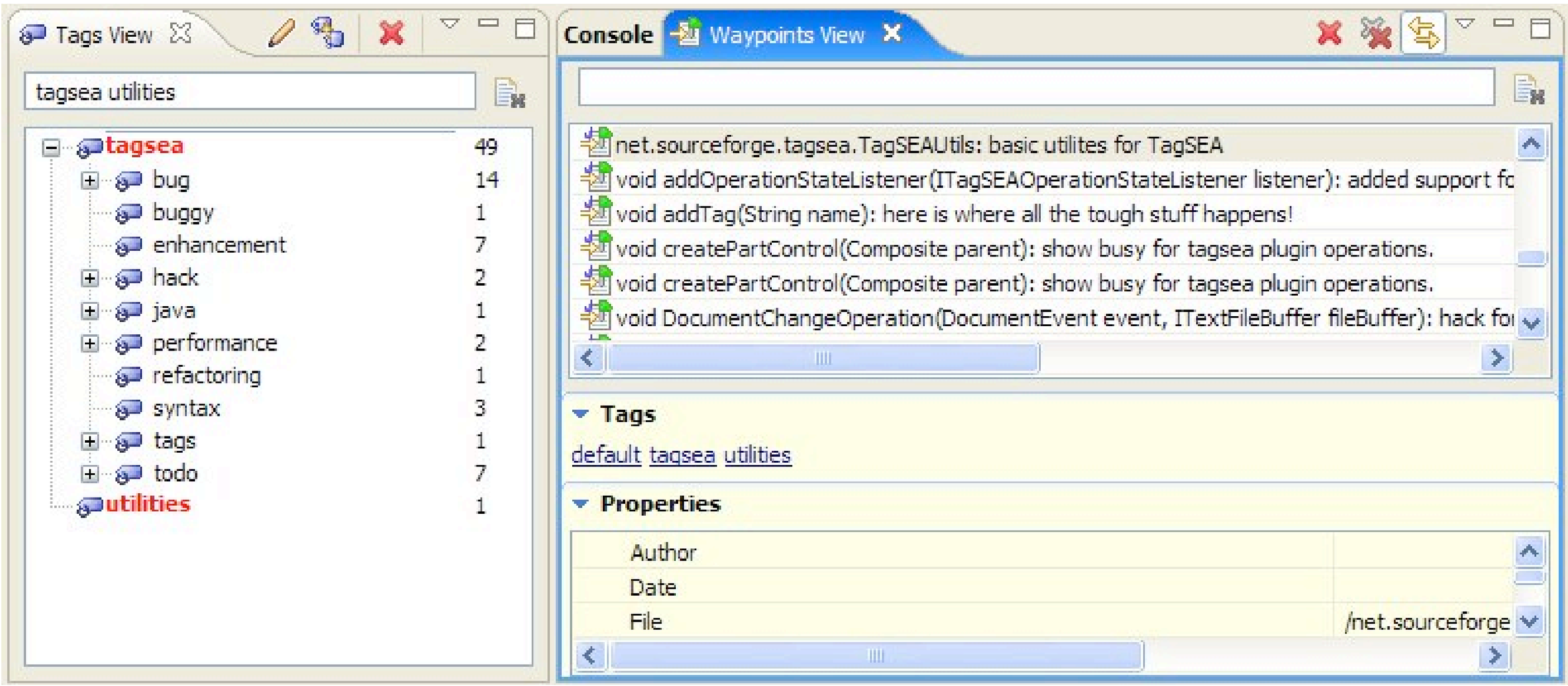
TagSea

DateMatcher.java | ExpressionMatcher... | TagTreeItemAdapte...

```
20 /**
21  * Simple utility class for TagSEA.
22  * @author Del Myers
23  */
24 //@tag tagsea utilities : basic utilites for TagSEA
25 public class TagSEAUtils {
26
27     /**
28      * Return code for a valid tag name.
29      */
30     public static final int TAG_NAME_VALID = 0;
31
32     /**
33      * Return code for a tag name with a syntax error.
34      */
35     public static final int TAG_NAME_SYNTAX_ERROR = 1;
36
```

Storey et al.
(CSCW 2006)

TagSea



TagSea = Waypoints + Social tagging

Suade

Suade addresses the concept assignment problem.

Guide developers towards interesting sections of code by analyzing the topology of a program's structural dependencies.

Robillard (ACM Distinguished Paper @ ESEC/FSE 2005)

The approach

Set of interest

fuzzy set

The approach

Set of interest

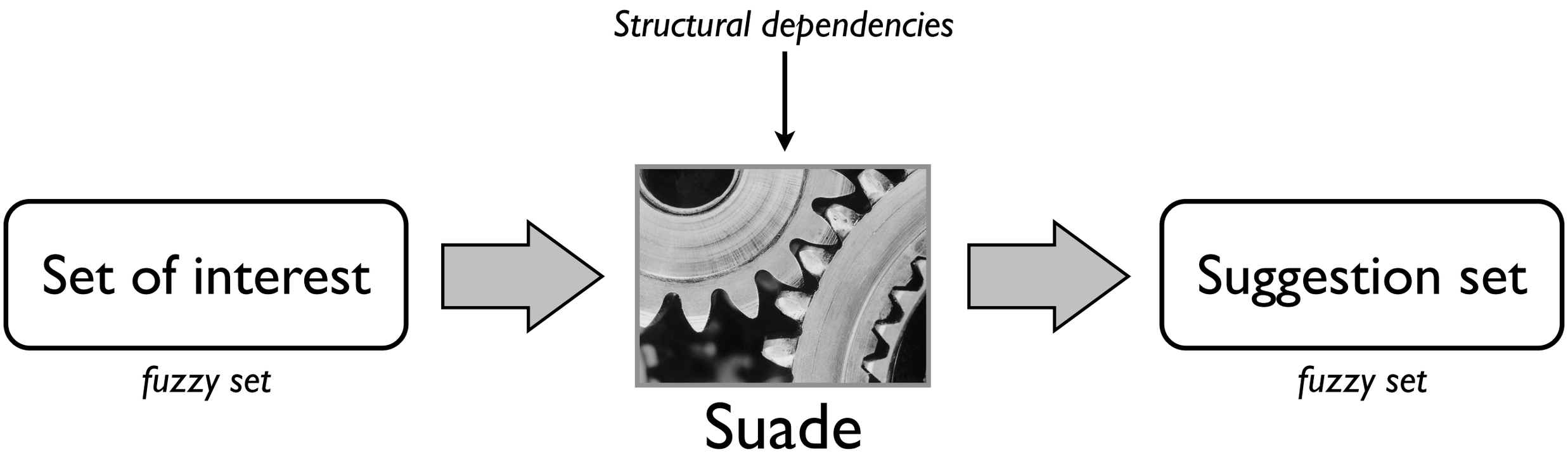
fuzzy set

Suade

Suggestion set

fuzzy set

The approach



The approach

Structural dependencies

Set of interest

fuzzy set

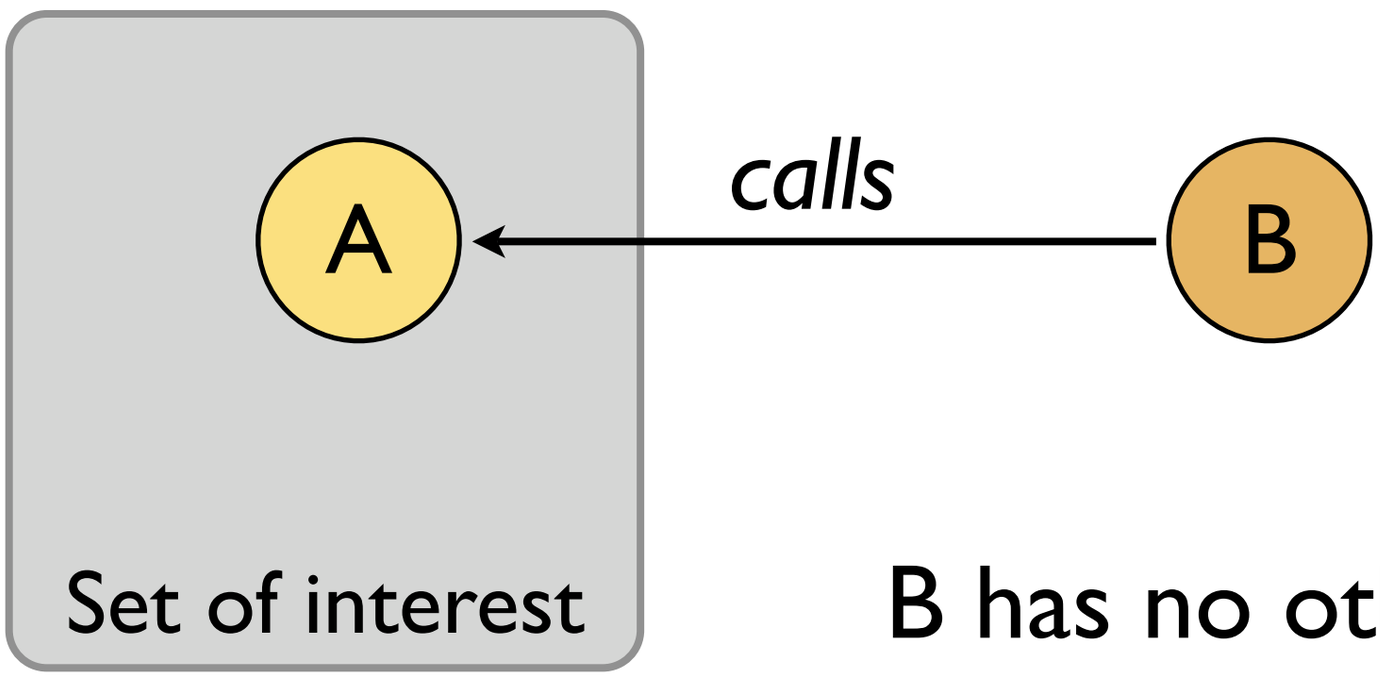
Suade

Suggestion set

fuzzy set

Specificity Reinforcement

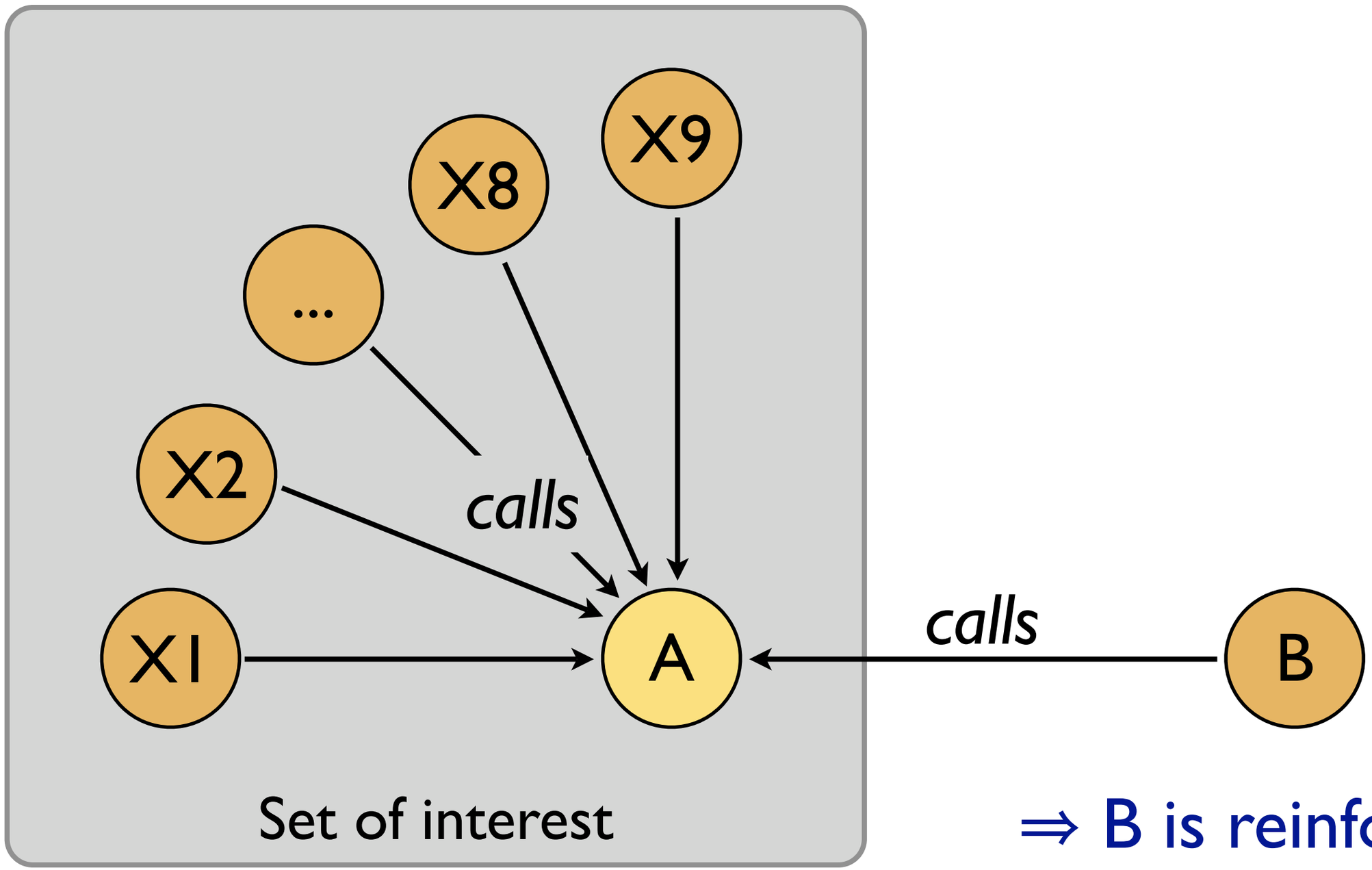
Specificity



B has no other calls!

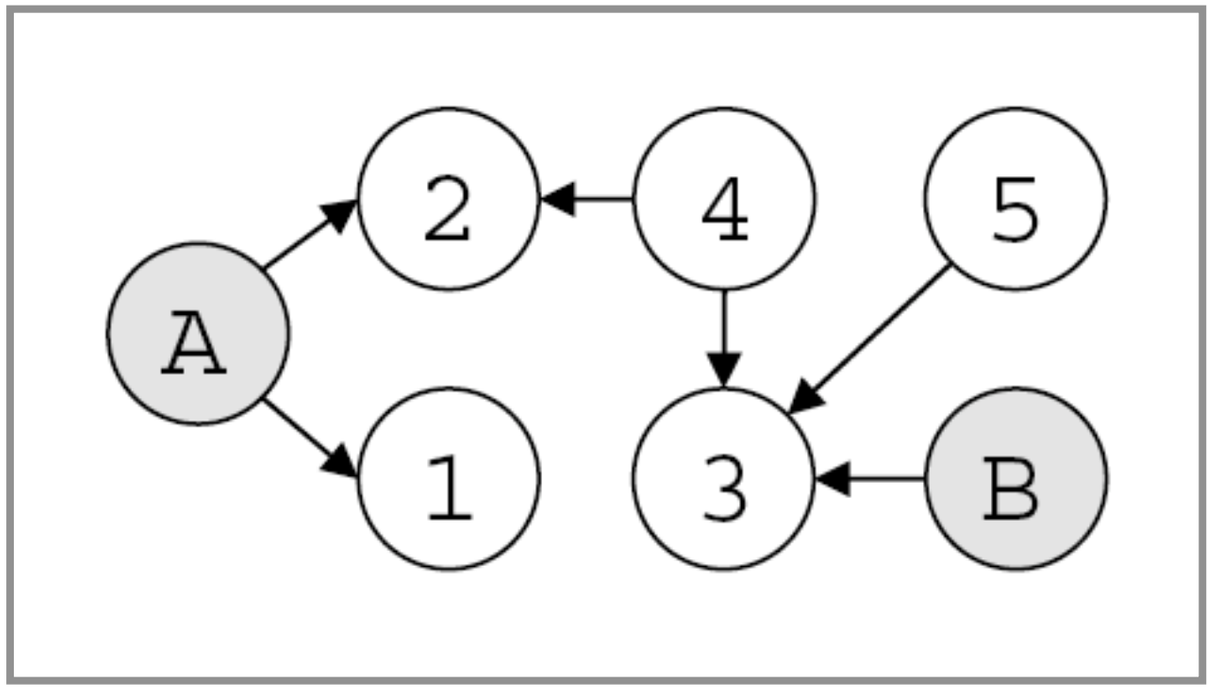
⇒ B is specific to the set of interest

Reinforcement



⇒ B is reinforced

Example for specificity



Distinguish between direct and transpose relations.

Multiple relations

Param: $\bar{I}$: Set of interest
Param: L: Set of relation types to analyze
Var: $\bar{S} = \{\}$: Suggestion set
Var: $\bar{T} = \{\}$: Temporary set
for all $l \in L$ **do**
    $\bar{T} = analyzeRelation(l, \bar{I})$
    $\bar{S} = \bar{S} \uplus \bar{T}$
end for
return $\bar{S}$

Multiple relations

Param: $\bar{I}$: Set of interest
Param: L: Set of relation types to analyze
Var: $\bar{S} = \{\}$: Suggestion set
Var: $\bar{T} = \{\}$: Temporary set
for all $l \in L$ **do**
    $\bar{T} = analyzeRelation(l, \bar{I})$
    $\bar{S} = \bar{S} \uplus \bar{T}$
end for
return $\bar{S}$

calls, called by, accesses, accessed by

Multiple relations

Param: $\bar{I}$: Set of interest
Param: L: Set of relation types to analyze
Var: $\bar{S} = \{\}$: Suggestion set
Var: $\bar{T} = \{\}$: Temporary set
for all $l \in L$ **do**
    $\bar{T} = analyzeRelation(l, \bar{I})$
    $\bar{S} = \bar{S} \uplus \bar{T}$
end for
return $\bar{S}$

calls, called by, accesses, accessed by

Suade uses a special union operator!!!

Analyze relations

Assumes: $P = (E, R)$: A program
Param: $\bar{I} = \{(x, \mu_{\bar{I}}(x)) \mid x \in E\}$: Set of interest
Param: $l \in \{r \mid (r, e_1, e_2) \in R\}$: Relation type to analyze
Param: $0 \leq \alpha \leq 1$: A calibration parameter
Var: $S_b \in E$: A (crisp) set of program elements
Var: $S_f \in E$: A (crisp) set of program elements
Var: $\bar{Z}$: The (fuzzy) set to be returned

Analyze relations

```
for all x ∈ Ī do
    S_b = {y | (l, x, y) ∈ R}
    for all s ∈ S_b do
        if s ∉ Ī then
            S_f = {y | (l^⊤, s, y) ∈ R}
            Z̄ = Z̄ ∪ {(s, (μ_Ī(x) · deg(1, S_b, Ī) · deg(0, S_f, Ī))^α)}
        end if
    end for
end for
return Z̄
```

Analyze relations

for all $x \in \bar{I}$ **do**
  $S_b = \{y \mid (l, x, y) \in R\}$
  for all $s \in S_b$ **do**
    if $s \notin \bar{I}$ **then**
      $S_f = \{y \mid (l^\top, s, y) \in R\}$
      $\bar{Z} = \bar{Z} \cup \{(s, (\mu_{\bar{I}}(x) \cdot deg(1, S_b, \bar{I}) \cdot deg(0, S_f, \bar{I}))^\alpha)\}$
    end if
  end for
end for
return $\bar{Z}$

direct degree

Analyze relations

for all $x \in \bar{I}$ **do**
  $S_b = \{y \mid (l, x, y) \in R\}$
  for all $s \in S_b$ **do**
    if $s \notin \bar{I}$ **then**
      $S_f = \{y \mid (l^\top, s, y) \in R\}$
      $\bar{Z} = \bar{Z} \cup \{(s, (\mu_{\bar{I}}(x) \cdot deg(1, S_b, \bar{I}) \cdot deg(0, S_f, \bar{I}))^\alpha)\}$
    end if
  end for
end for
return $\bar{Z}$

transposed degree

direct degree

Analyze relations

for all $x \in \bar{I}$ **do**
  $S_b = \{y \mid (l, x, y) \in R\}$
  for all $s \in S_b$ **do**
    if $s \notin \bar{I}$ **then**
      $S_f = \{y \mid (l^\top, s, y) \in R\}$
      $\bar{Z} = \bar{Z} \cup \{(s, (\mu_{\bar{I}}(x) \cdot deg(1, S_b, \bar{I}) \cdot deg(0, S_f, \bar{I}))^\alpha)\}$
    end if
  end for
end for
return $\bar{Z}$

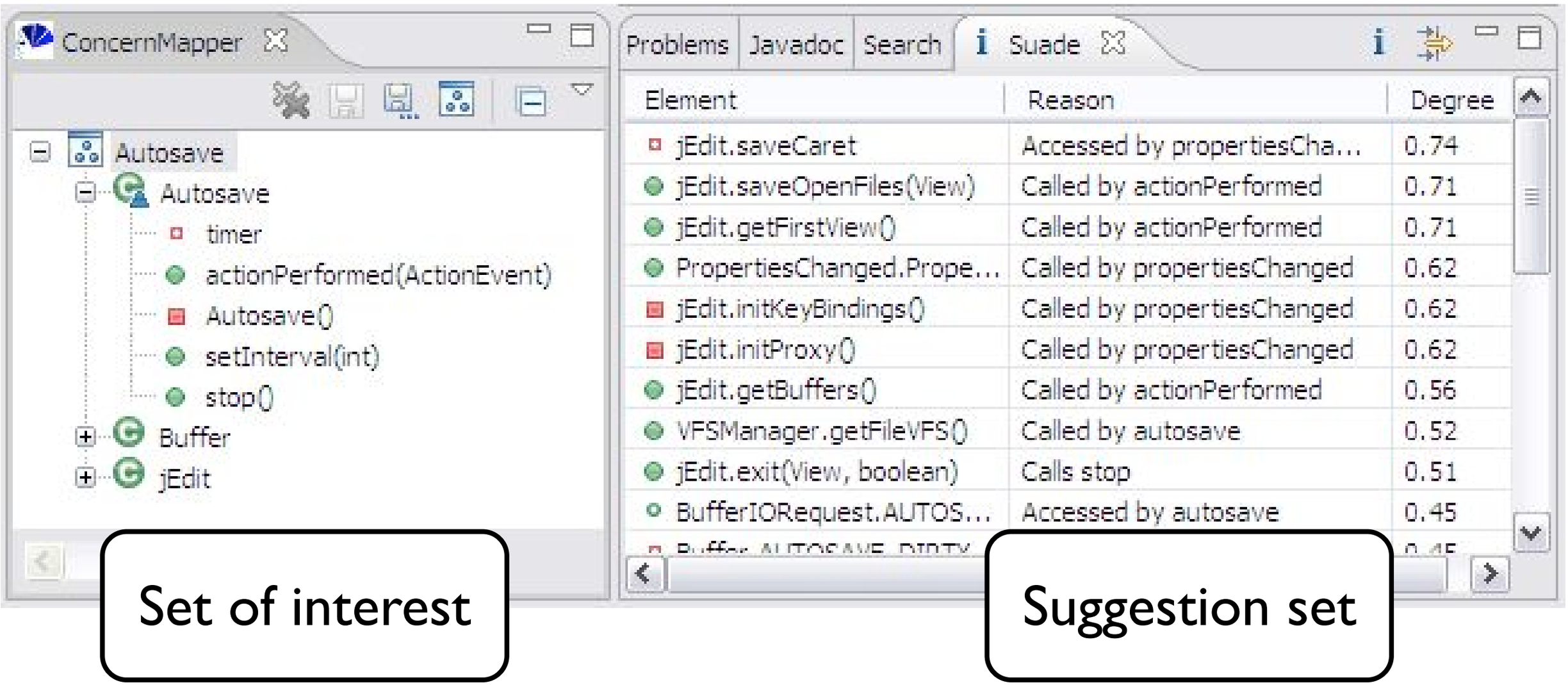
transposed degree

direct degree

$$deg(t, U, \bar{V}) = \frac{t + |U \cap \bar{V}|}{|U|}$$

Example

The Suade plug-in



Quantitative evaluation

Size of suggestion sets.

On average 4.8 suggestions for a singleton set.

Largest set contained 86 elements.

Parameterization of the algorithm.

Most useful values of α for $0.1 < \alpha < 0.7$.

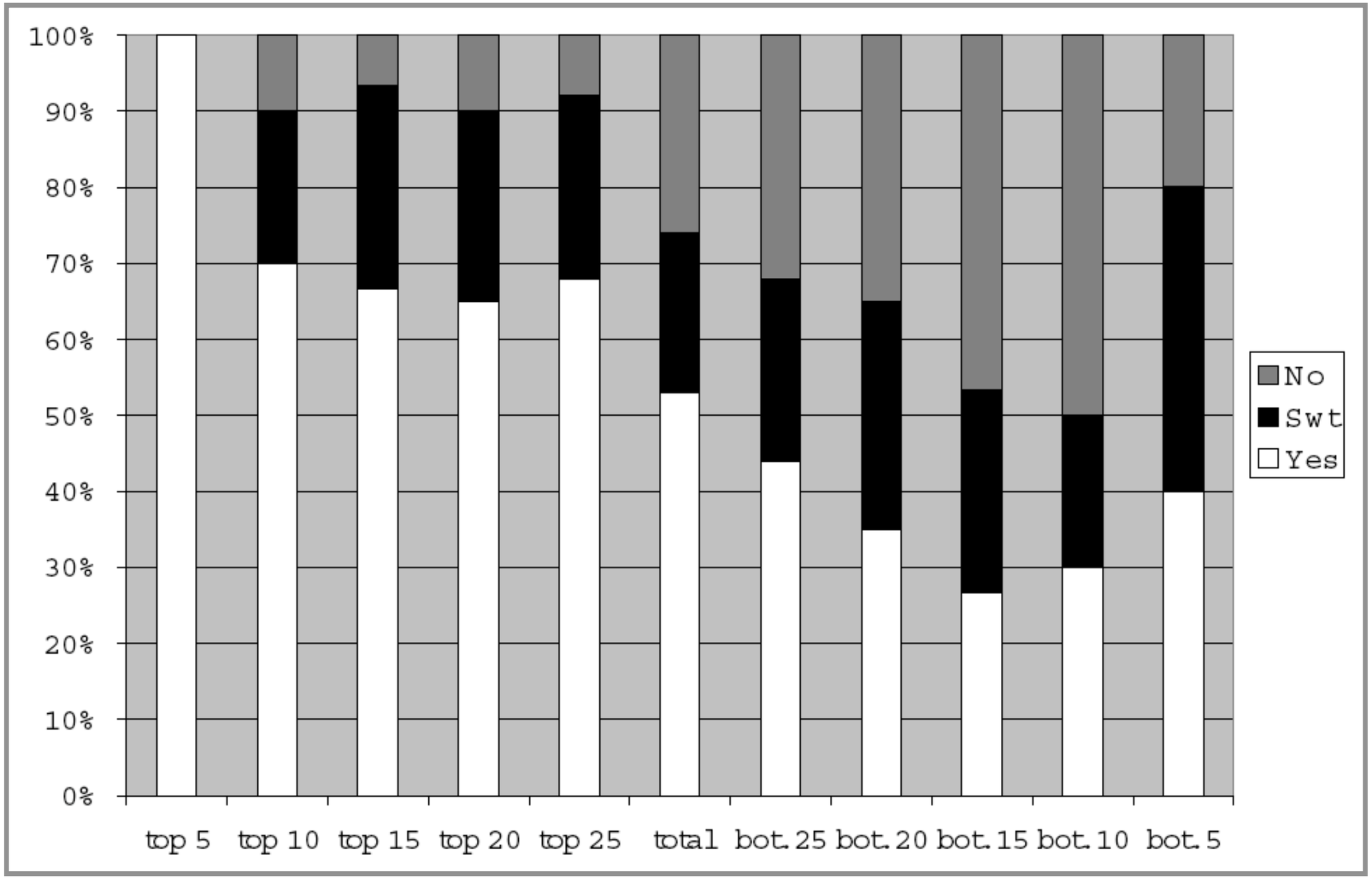
Degree should be viewed as relative measure.

Case study

File allocation concern of the Azureus system.
Set of interest: 4 members of DiskManagerImpl
Suggestion set: additional 54 elements

Evaluation by two experts (working as a team).
Is this element relevant to understand how files are allocated? ⇒ “Yes”, “No”, and “Somewhat”

Case study


100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
top 5
top 10
top 15
top 20
top 25
total
bot. 25
bot. 20
bot. 15
bot. 10
bot. 5
No
Swt
Yes

Mylar

Provide a task-focussed UI that reduces information overload by monitoring the work activity of developers.

Kersten and Murphy (AOSD 2005, FSE 2006)

Java - Figure.java -

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

DefaultRangeModel.java
DeferredUpdateManager.ja
DelegatingLayout.java
Ellipse.java
EllipseAnchor.java
EventDispatcher.java
EventListenerList.java
ExclusionSearch.java
FanRouter.java
Figure.java
FigureCanvas.java
FigureListener.java
FigureUtilities.java
FlowLayout.java
FocusBorder.java
FocusEvent.java
FocusListener.java
FocusTraverseManager.jav
FrameBorder.java
FreeformFigure.java
FreeformHelper.java
FreeformLayer.java
FreeformLayeredPane.java
FreeformLayout.java
FreeformListener.java
FreeformViewport.java
Graphics.java
GraphicsSource.java
GroupBoxBorder.java
IFigure.java
ImageFigure.java
ImageUtilities.java
InputEvent.java
KeyEvent.java
KeyListener.java
Label.java
LabelAnchor.java
LabeledBorder.java

Absolut... | Abstrac... | Abstrac... | Figure.... | 67

```
 */
protected boolean getFlag(int flag) {
    return (flags & flag) != 0;
}

/**
 * @see IFigure#getFont()
 */
public Font getFont() {
    if (font != null)
        return font;
    if (getParent() != null)
        return getParent().getFont();
    return null;
}

/**
 * @see IFigure#getForegroundColor()
 */
public Color getForegroundColor() {
    if (fgColor == null && getParent() != null)
        return getParent().getForegroundColor();
    return fgColor;
}

/**
 * Returns the border's Insets if the border is set. Otherwi
 * instance of Insets with all 0s. Returns Insets by referen
```

Problems | Search | Declaration | Javadoc

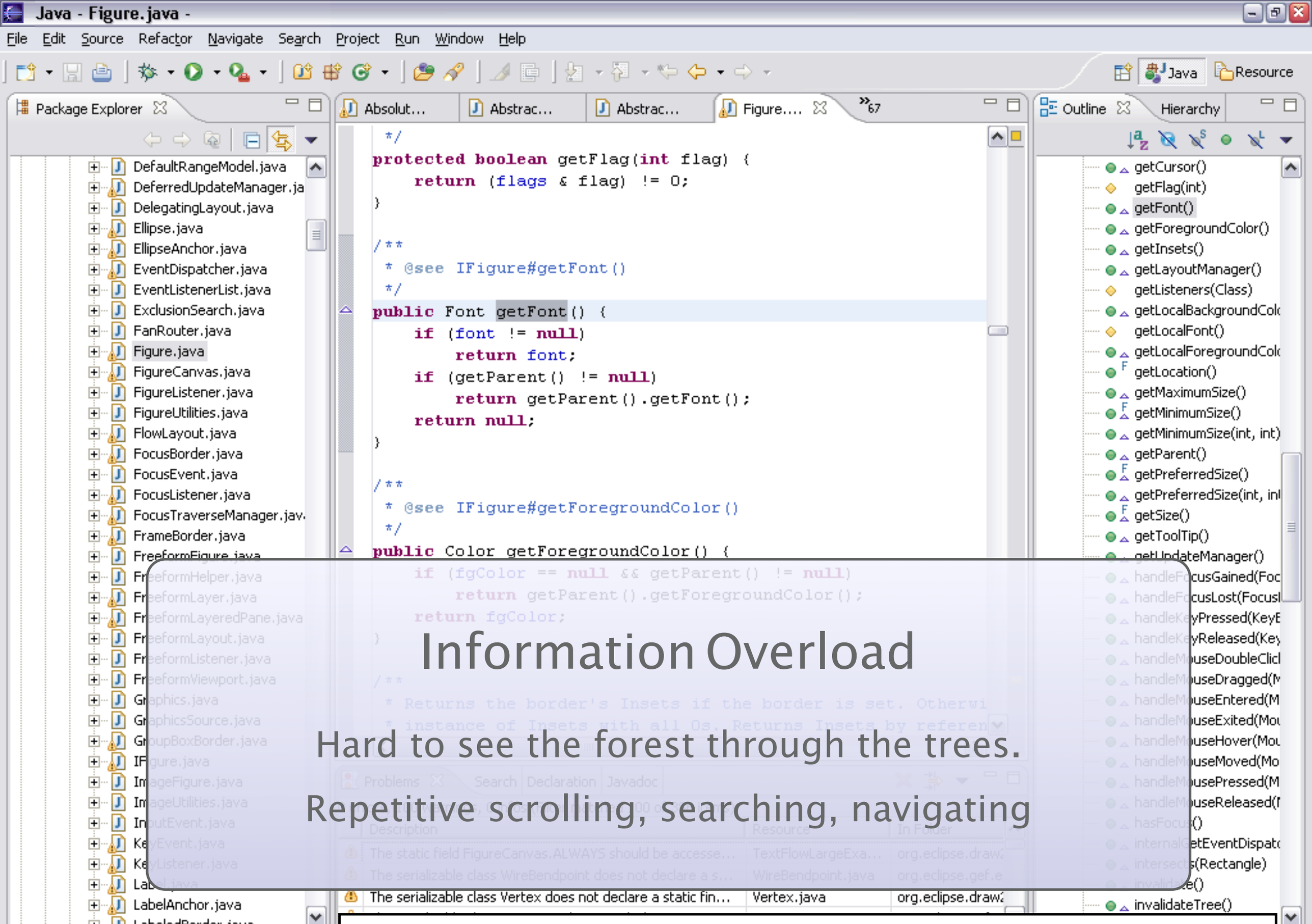
0 errors, 100 warnings, 0 infos (Filter matched 100 of 386 items)

Description	Resource	In Folder
The static field FigureCanvas.ALWAYS should be accesse...	TextFlowLargeExa...	org.eclipse.draw2
The serializable class WireBendpoint does not declare a s...	WireBendpoint.java	org.eclipse.gef.e
The serializable class Vertex does not declare a static fin...	Vertex.java	org.eclipse.draw2

Outline | Hierarchy

getCursor()
getFlag(int)
getFont()
getForegroundColor()
getInsets()
getLayoutManager()
getListeners(Class)
getLocalBackgroundCol
getLocalFont()
getLocalForegroundCol
getLocation()
getMaximumSize()
getMinimumSize()
getMinimumSize(int, int)
getParent()
getPreferredSize()
getPreferredSize(int, int
getSize()
getToolTip()
getUpdateManager()
handleFocusGained(Foc
handleFocusLost(Focusl
handleKeyPressed(KeyE
handleKeyReleased(Key
handleMouseDoubleClicl
handleMouseDragged(M
handleMouseEntered(M
handleMouseExited(Mou
handleMouseHover(Mou
handleMouseMoved(Mo
handleMousePressed(M
handleMouseReleased(
hasFocus()
internalGetEventDispat
intersects(Rectangle)
invalidate()
invalidateTree()

taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)



taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

Mylar Tasks - Figure.java -

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer | Hierarchy

- org.eclipse.draw2d
  - src
    - org.eclipse.draw2d
      - Figure.java
        - Figure
          - getFont
          - getLocalFont
          - getParent
  - Plug-in Dependencies
  - JRE System Library [jdk1.5.0]
    - rt.jar
      - java.awt
        - Font.class
          - Font
- org.eclipse.gef.examples.flow
  - src
    - org.eclipse.gef.examples.flow.ui
      - FlowEditor.java
        - FlowEditor
          - createActions
  - Plug-in Dependencies
  - JRE System Library [jdk1.5.0]

Font.class | Figure.java | FlowEditor.java | Font.class

```
protected boolean getFlag(int flag) {

/**
public Font getFont() {
    if (font != null)
        return font;
    if (getParent() != null)
        return getParent().getFont();
    return null;
}

/**
public Color getForegroundColor() {

/**
public Insets getInsets() {

/**
public LayoutManager getLayoutManager() {

/**
protected Iterator getListeners(Class clazz) {

/**
public Color getLocalBackgroundColor() {

/**
protected Font getLocalFont() {
```

Problems

0 errors, 169 warnings, 0 infos

Description	Java Element
The serializable class Figure does not declare a static final serialVe...	Figure

Outline | Task List

- Figure
  - getFont()
  - getLocalFont()
  - getParent()

Active Type Hierarchy

- Object
  - Figure
  - WorkbenchPart
    - EditorPart
      - GraphicalEditor
        - GraphicalEditorWithP
          - FlowEditor

Writable | Smart Insert | 600 : 13

taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

Mylar

See only what you're working on.

Aluminized film used to avoid blindness when staring at an eclipse.

Task Focused UI to avoid information blindness when staring at Eclipse.

taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

Context (Core)

- Degree-of-interest graph

taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

Context (Core)

- Degree-of-interest graph

taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

Context (Core)

- Degree-of-interest graph

taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

Context (Core)

- Degree-of-interest graph

taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

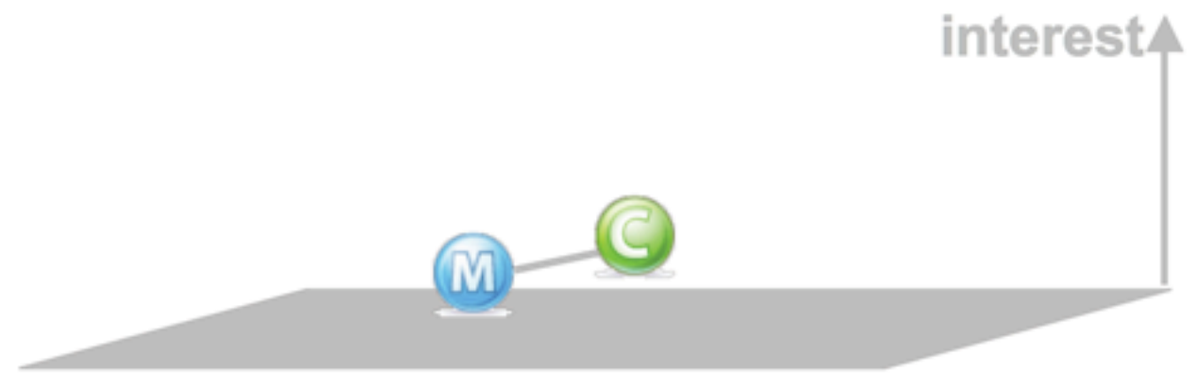
Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

Context (Core)

- Degree-of-interest graph



taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

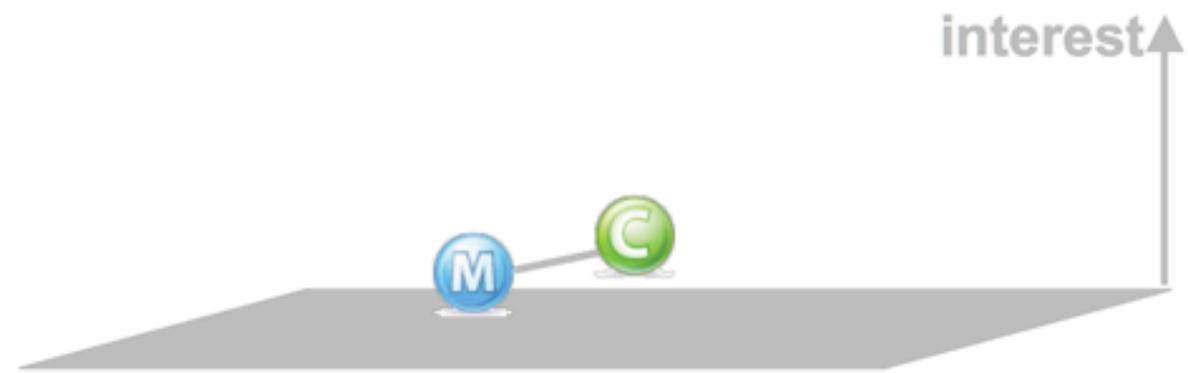
Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

Context (Core)

- Degree-of-interest graph



taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

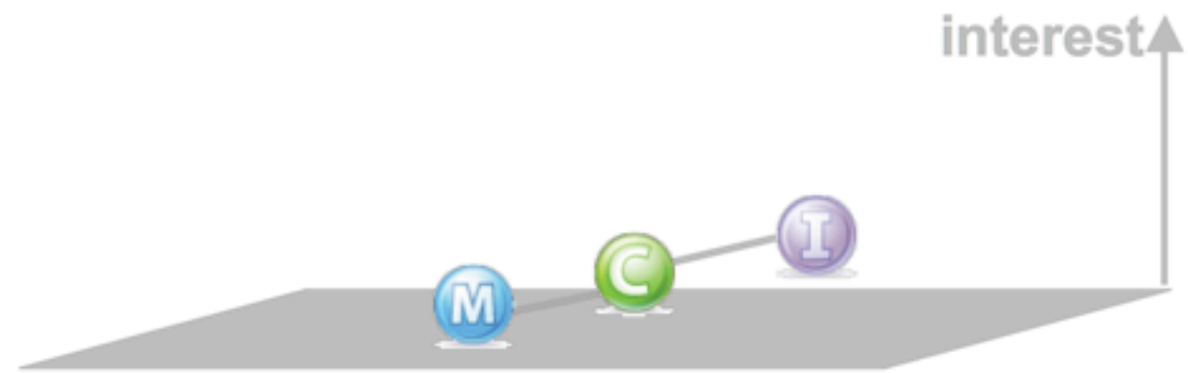
Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

Context (Core)

- Degree-of-interest graph



taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

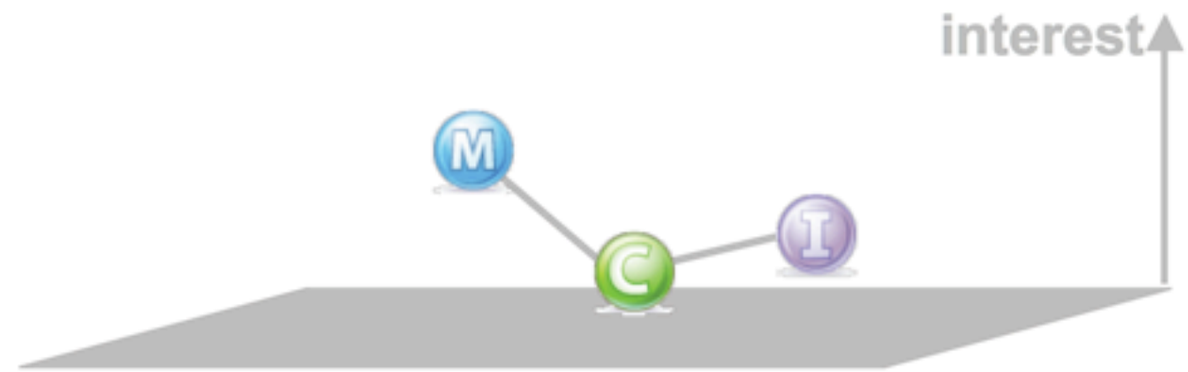
Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

Context (Core)

- Degree-of-interest graph



taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

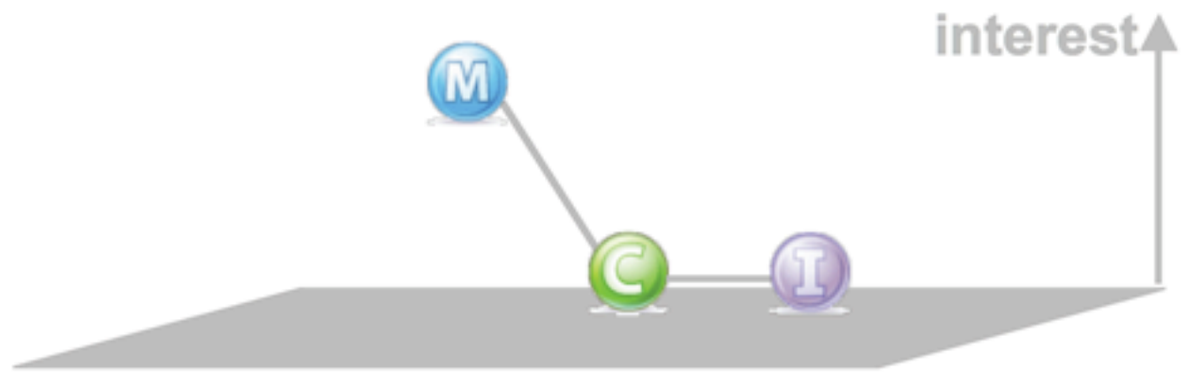
Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

Context (Core)

- Degree-of-interest graph



taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

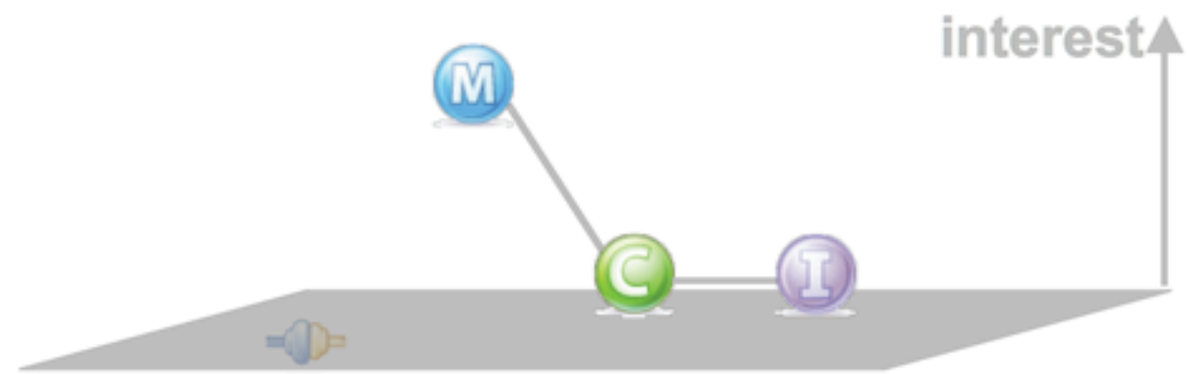
Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

Context (Core)

- Degree-of-interest graph



taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

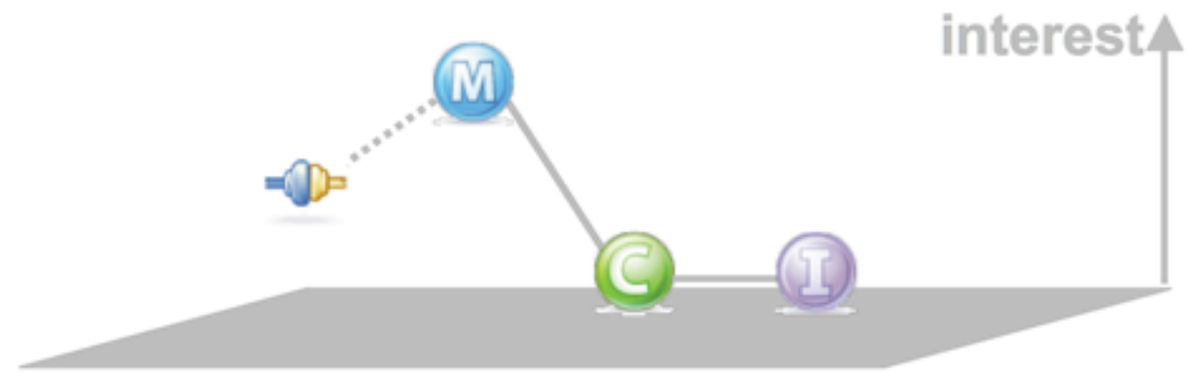
Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

Context (Core)

- Degree-of-interest graph
- Degree-of-separation scope



taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

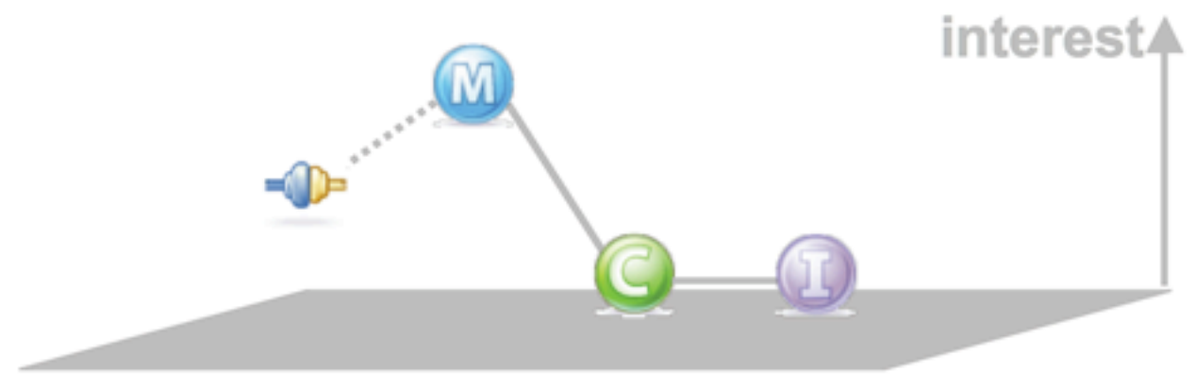
Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

Context (Core)

- Degree-of-interest graph
- Degree-of-separation scope
- Scaling factors



taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

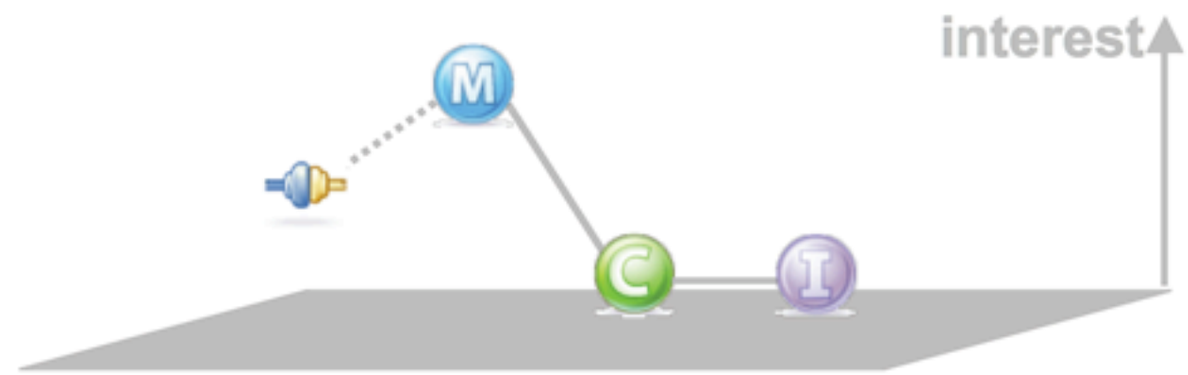
Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

Context (Core)

- Degree-of-interest graph
- Degree-of-separation scope
- Scaling factors



Projections (UI)

- Views, editors, files

taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

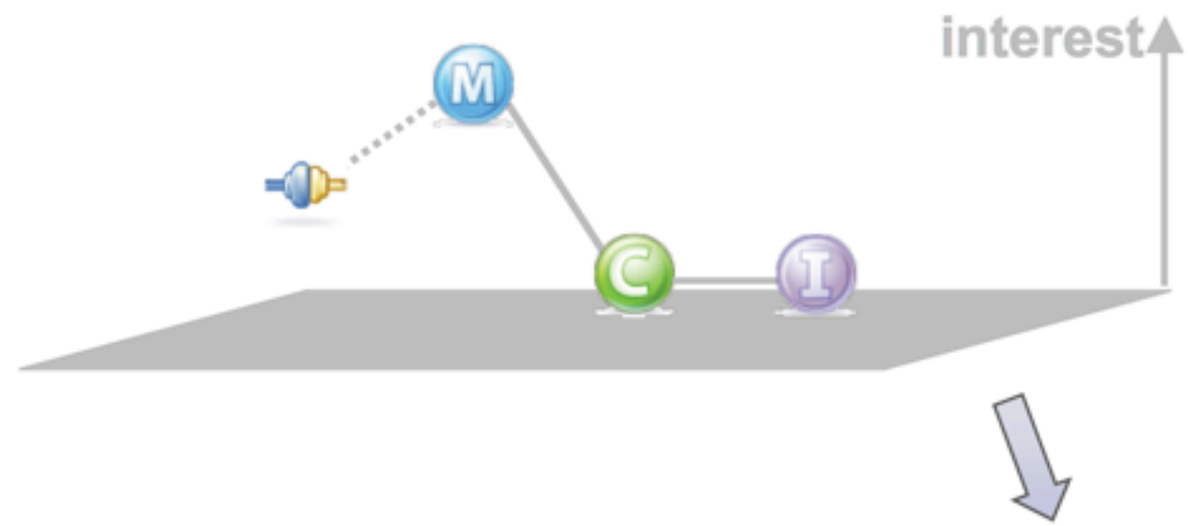
Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

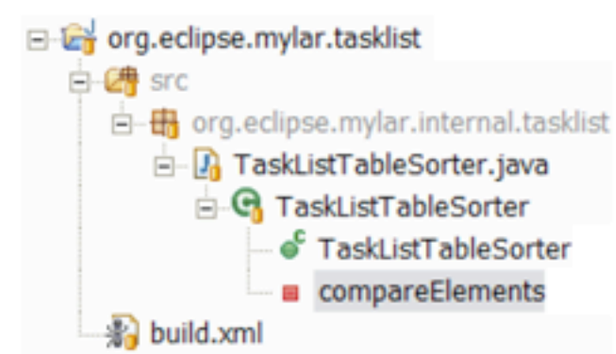
Context (Core)

- Degree-of-interest graph
- Degree-of-separation scope
- Scaling factors



Projections (UI)

- Views, editors, files



taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

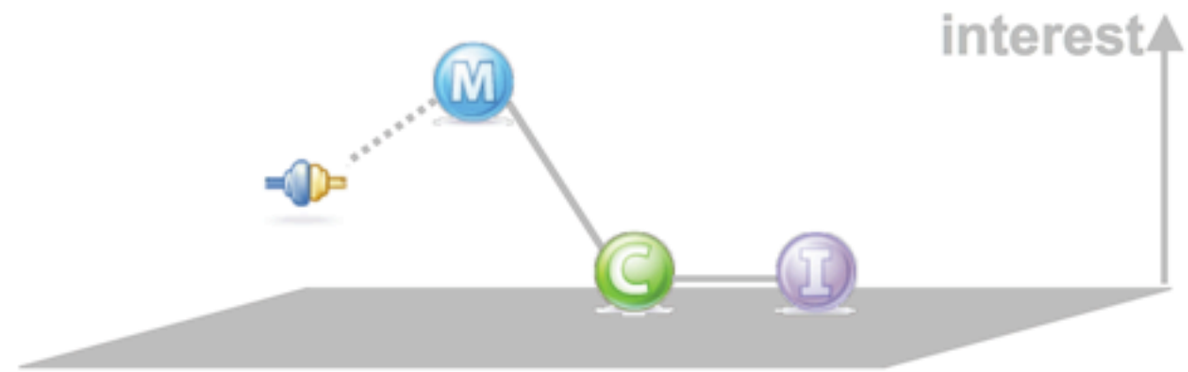
Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

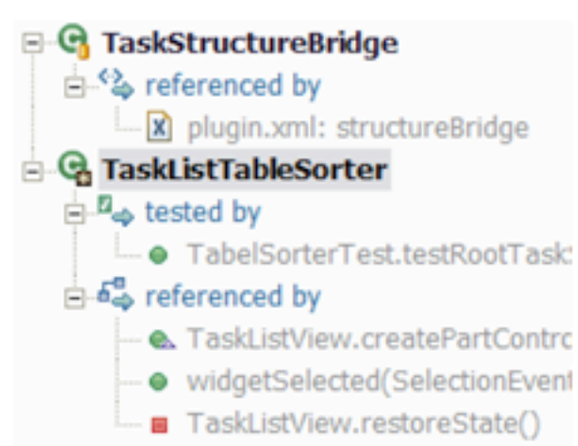
Context (Core)

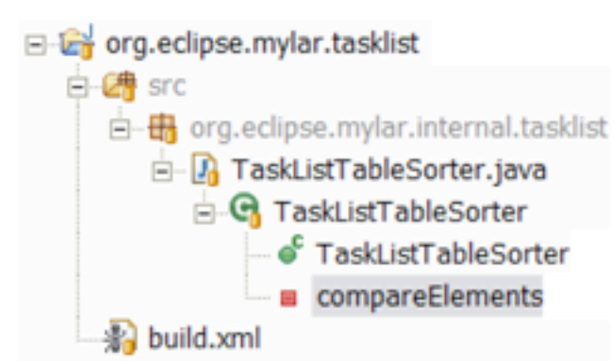
- Degree-of-interest graph
- Degree-of-separation scope
- Scaling factors



Projections (UI)

- Views, editors, files





taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

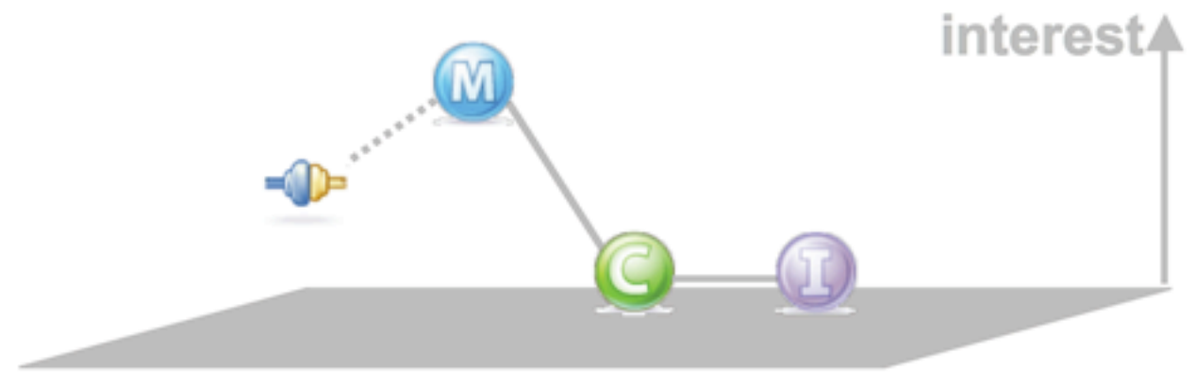
Mylar's context model

Interaction history

- InteractionEvent stream
- Origin, handle, type, date

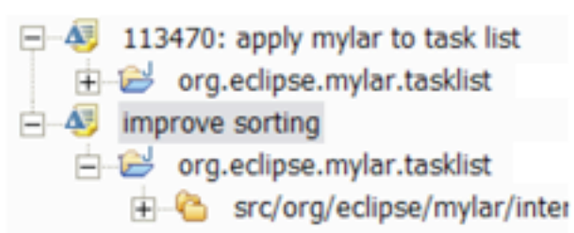
Context (Core)

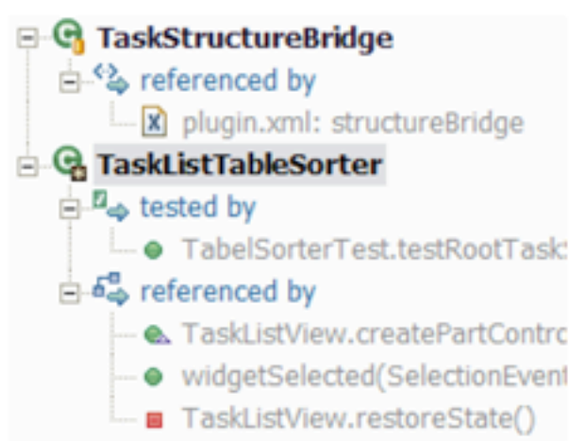
- Degree-of-interest graph
- Degree-of-separation scope
- Scaling factors

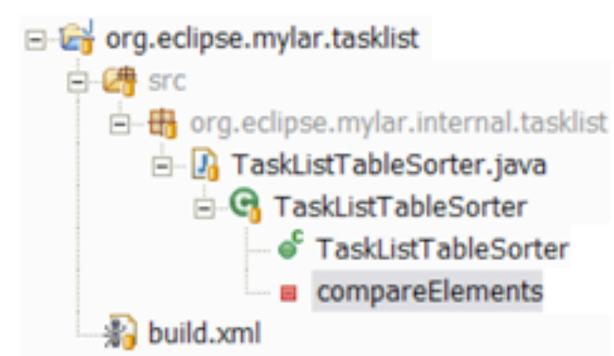


Projections (UI)

- Views, editors, files







taken from *Mylar: a Task Focused UI for Eclipse* (EclipseCon talk, Mar 2006)

Summary

- Landmarks, beacons, and waypoints
- Route knowledge vs. survey knowledge
- Support software navigation by using
  – past navigation traces (NavTracks)
  – structural dependencies (Suade)
- Task-focus reduces information overload (Mylar)

Literature

Design guidelines for landmarks to support navigation in virtual environments.
Norman Vinson.

Theoretical considerations on navigating codespace with spatial cognition .
Anthony Cox, Maryanne Fisher, and Philip O'Brien.

NavTracks: Supporting Navigation in Software Maintenance.
Janice Singer, Robert Elves, and Margaret-Anne D. Storey.

Automatic generation of suggestions for program investigation.
Martin P. Robillard

Using task context to improve programmer productivity.
Mik Kersten, Gail C. Murphy