Cognitive and Social Support for Learning Java API

Yunwen Ye
April 24, 2006
University of Colorado at Boulder
Guest Lecture at Design, Learning and Collaboration
I have a dream…

In 1983, 85% of the code has been repeated by someone else in the world. (Capers Jones, 1984)

Every time when I am trying to re-invent the wheel, an agent brings me the existing wheel that I can use immediately
I have yet another dream…

All the information I need is always at my desktop, virtual or real; and the needed information only
Software reuse

- **Definition**
  - Creating new software systems with existing artifacts

- **Reusable artifacts**
  - **Code artifacts**
    - macros, functions, *methods, classes*, subsystems, systems (Open Source Software)
  - **Non-code artifacts**
    - analyses, designs, test plans and cases, domain models
  - **Knowledge**
    - program idioms, program plans, design patterns, software architecture styles, domain knowledge

- **Reuse repository systems**
  - Supporting reuse activities
Why reuse?

- Increased productivity
  - Reduced development time
  - Reduced cognitive load
  - Reduced testing time
- Increased quality
  - Fewer bugs
- Enhanced evolvability and maintainability
Reuse process (sLCMS)
Understanding the cognitive issues in reuse

- Cognitive engineering:
  - Apply what is known from cognitive science to the design and construction of tools that assists cognitive activities of human beings

- Bridge two gulfs between users and tools
Execution gulf

- Bridging the gap from the goal to the tool
  - Intention Formation
    - Users decide to do something with an internal specification of the task created from their goal.
  - Action Specification
    - Users externalize the internal specification into a sequence of specified actions.
  - Action Execution
    - The actions are executed with the tool.

© 2006 Yunwen Ye
Evaluation gulf

- Bridging the gap from tool output to goal
  - System Perception
    - Users perceive the output of the tool.
  - Interpretation
    - Users interpret the perceived output.
  - Evaluation
    - Users compare the interpretation with the original goal.

© 2006 Yunwen Ye
A cognitive model of reuse

Development Environment

form

reuse intentions

formulate

reuse queries

retrieve

repository

integrate

chosen comps

choose

retrieved comps

Retrieval by Reformulation
A cognitive model of reuse

No attempt to reuse

formulate

reuse queries

repository

form

reuse intentions

Development Environment

choose

chosen comps

integrate

Unable to use in the current context

Unable to understand the components

Retrieve by Reformulation

Vocabulary mismatch

Existence of components
Research problems

- No attempt to reuse *(Location)*
  - Information islands
    - Not aware of the existence of reusable components
  - Perceived low reuse utility (benefits/cost)
    - High cost of locating components

- Unable to locate the component *(Location)*
  - Situation model vs. system model

- Unable to use the component *(Comprehension)*
User’s knowledge about a reuse repository

L4: System Model

L3: Belief

L2: Vaguely Known

L1: Well Known
Growth of Java class library

The growth of Java API classes and interfaces

<table>
<thead>
<tr>
<th>Version &amp; release year</th>
<th>No. of classes and interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java 1.0 (1996)</td>
<td>212</td>
</tr>
<tr>
<td>Java 1.1 (1997)</td>
<td>504</td>
</tr>
<tr>
<td>Java 1.2 (1998)</td>
<td>1781</td>
</tr>
<tr>
<td>Java 1.3 (2000)</td>
<td>2130</td>
</tr>
<tr>
<td>Java 1.4.0 (2002)</td>
<td>2723</td>
</tr>
<tr>
<td>Java 1.5.0 (2004)</td>
<td>3279</td>
</tr>
</tbody>
</table>
Functionality of development tools

- Features
- Cognitive capability
- Cognitive friction point
## Libraries used in STeP_IN

<table>
<thead>
<tr>
<th>Library Name</th>
<th>Class</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>activation.jar</td>
<td>38</td>
<td>261</td>
</tr>
<tr>
<td>bcel-5.1.jar</td>
<td>373</td>
<td>3093</td>
</tr>
<tr>
<td>commons-collections-3.1.jar</td>
<td>446</td>
<td>4021</td>
</tr>
<tr>
<td>commons-dbcp-1.2.1.jar</td>
<td>44</td>
<td>935</td>
</tr>
<tr>
<td>commons-pool-1.2.jar</td>
<td>25</td>
<td>277</td>
</tr>
<tr>
<td>jun547.jar</td>
<td>2640</td>
<td>18412</td>
</tr>
<tr>
<td>mail.jar</td>
<td>240</td>
<td>1966</td>
</tr>
<tr>
<td>postgresql.jar</td>
<td>82</td>
<td>1216</td>
</tr>
<tr>
<td>resolver.jar</td>
<td>29</td>
<td>298</td>
</tr>
<tr>
<td>StPL75.jar</td>
<td>175</td>
<td>1384</td>
</tr>
<tr>
<td>xercesImpl.jar</td>
<td>784</td>
<td>7463</td>
</tr>
<tr>
<td>xml-ParserAPIs.jar</td>
<td>207</td>
<td>1748</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>5083</strong></td>
<td><strong>41074</strong></td>
</tr>
</tbody>
</table>
No attempt to reuse

- No attempt to reuse is the most significant barrier to reuse (Frakes & Fox, 1996)
Proposed solution

- Active component repository systems
  - Overcoming the limits of browsing and searching
  - Supporting information delivery

- Benefits
  - Reusing unknown components
  - Reduced locating cost
  - Seamless integration with programming environment
Challenges in active reuse repository systems

L3: Belief
L2: Vaguely Known
L1: Well Known
L4: System Model

Task-relevant information
/** This class simulates the process of card dealing. Each card is represented with a number from 0 to 51. The program should produce a list of 52 cards, as results from a human card dealer. */
public class CardDealer {
    static int[] cards = new int[52];
    static {
        for (int i = 0; i < 52; i++) cards[i] = i;
    }
    /** Create a random number between two limits. */
    public static int getRandomNumber (int from, int to) {

    }
}

/** An example for getInt written by yunwen "Fri Oct 5 14:00:58 2001" */
import com.objectspace.jgl.util.*;
/** Roll a die and print the probability of each number's occurrence. */
public class DiceRoller {
    final static int times = 10000;
    public static void main (String args[]) {
        int[] distribution = new int[6];
        int p;
        for (int i = 0; i < times; i++) {
            p = Randomizer.getInt (1, 6);
            distribution[p-1]++;
        }
        System.out.println("(Number, Occurrences, Probability)"");
    }
}

*CB-Example*/
/home/yunwen/java/examples/DiceRoller.java (JDE)--L10--Top--
1 0.89 getInt Generate a random number using the default generator
2 0.78 getLong Generate a random number using the default generator
3 0.78 nextInt Generates an even number
4 0.77 nextLong Generates a given number
-1:2** RCLI-display** 10-05 02:08 PM 0.97 (ReusableCode)
com.objectspace.jgl.util.Randomizer::nextInt(int lo, int hi)
/** This class simulates the process of card dealing. Each card is represented with a number from 0 to 51. And the program produces a list of 52 cards, as it is resulted from a human card dealer */

```java
public class CardDealer1 {
    static int [] cards=new int[52];
    static {
        for (int i=0; i<52; i++) cards[i]=i;
    }
    /**< Create a random number between two limits */
```
CodeBroker: An active reuse repository system

```java
/** This class simulates the process of card dealing. Each card is represented with a number from 0 to 51. And the program produces a list of 52 cards, as it is resulted from a human card dealer */
public class CardDealer1 {
    static int[] cards = new int[52];
    static {
        for (int i = 0; i < 52; i++) cards[i] = i;
    }
    /** Create a random number between two limits */
    public static int getRandomNumber(int from, int to) {
```
CodeBroker: An active reuse repository system

```java
public class CardDealer1 {
    static int[] cards = new int[52];
    static {
        for (int i = 0; i < 52; i++) cards[i] = i;
    }
    /** Create a random number between two limits */
    public static int getRandomNumber(int from, int to) {
```

```java
    public static int getInt(int lo, int hi)
        Generate a random number using the default generator.

    See Also:
        nextInt

    public static long getRandom(long hi)
        Generate a random number using the default generator.

    See Also:
        nextLong
```
CodeBroker: An active reuse repository system

```java
public class CardDealer {
    static int[] cards = new int[52];
    static {
        for (int i = 0; i < 52; i++) cards[i] = i;
    }
    /** Create a random number between two limits */
    public static int getRandomNumber(int from, int to) {
    }

    /** CardDealer.java 10-05 02:08 PM 0.97 (JDE)--L10--All----------
    */
    An example for getInt written by yunwen "Fri Oct 5 14:00:58 2001"*/
    import com.objectspace.jgl.util, *;
    /** Roll a die and print the probability of each number's occurrence */
    public class DiceRoller {
        final static int times = 10000;
        public static void main(String args[]) {
            int[] distribution = new int[6];
            int p;
            for (int i = 0; i < times; i++) {
                p = Randomizer.getInt(1, 6);
                distribution[p - 1]++;
            }
            System.out.println("(Number, Occurrences, Probability)";
            CB-Example*/(/home/yunwen/java/examples/DiceRoller.java) (JDE)--L10--Top-
            1 0.89 getInt Generate a random number using the default generator
            2 0.78 getNext Generate a random number using the default generator
            3 0.78 nextInt Generates an int value between
            4 0.77 nextLong Generates a long value between
            =1;** *RCI--display* 10-05 02:08 PM 0.97 (ReusableComponentInfo)--L1--Top-
            com.objectspace.jgl.util.Randomizer::int getInt(int lo, int hi)
```
Inferring the task

- Plan recognition
  - Actions → Inferred goal → Suggested actions or information

- Similarity analysis
Similarity analysis in CodeBroker

Current situation

Create a random number between two limits

int <- int x int

Signature Matching
Information Retrieval

Fetcher

Listener

Situation A

Generate a random number using the default generator

int <- int x int

Fetcher

Listener
The rationale

- Three aspects of a program
  - Concept
    - The functionality of the program
    - Semantic information
    - Revealed in comments, identifiers, ...
  - Constraint
    - Execution environment
    - Syntactic information
    - Revealed in signatures, protocols, ...
  - Code
    - The implementation

- The assumption
  - Similar concept + compatible signature → reusable code
Basic information retrieval (IR) techniques

- Information retrieval: Finding similar documents based on the commonality of terms
  - Documents and queries are represented by term vectors
    \[ D_j = (f_1, j, f_2, j, \ldots, f_N, j) \]
  - Similarity is the distance between two vectors
    \[
    \text{Similarity}(Q, D) = \sqrt{\sum_{i=1}^{n} Q[i] \times D[i]} / \sqrt{\sum_{i=1}^{n} Q[i]^2 \times \sum_{i=1}^{n} D[i]^2}
    \]

Term space: (factor information help human operation retrieval system)

<table>
<thead>
<tr>
<th>Contents</th>
<th>Vector</th>
<th>Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q human factors in information retrieval system</td>
<td>(1 1 0 1 0 1 1)</td>
<td></td>
</tr>
<tr>
<td>D1 factor factor factor human human retrieval system</td>
<td>(3 0 0 2 0 1 1)</td>
<td>7/75^{0.5} = 0.80</td>
</tr>
<tr>
<td>D2 information operation retrieval retrieval</td>
<td>(0 1 0 0 1 2 0)</td>
<td>0.55</td>
</tr>
<tr>
<td>D3 factor help help retrieval</td>
<td>(1 0 2 0 0 1 0)</td>
<td>0.37</td>
</tr>
</tbody>
</table>
LSA: Improved IR

- Latent semantic analysis
  - Addressing the vocabulary mismatch problem (people use different names to refer to the same concept)
  - Creating a semantic space with a large amount of documents

\[
\begin{align*}
\begin{pmatrix}
  w_{1,1} & w_{1,2} & \cdots & w_{1,M} \\
  w_{2,1} & w_{2,2} & \cdots & w_{2,M} \\
  \vdots & \vdots & \ddots & \vdots \\
  w_{N,1} & w_{N,2} & \cdots & w_{N,M}
\end{pmatrix}

&= 
\begin{pmatrix}
  t_{1,1}^{(0)} & t_{1,2}^{(0)} & \cdots & t_{1,r}^{(0)} \\
  t_{2,1}^{(0)} & t_{2,2}^{(0)} & \cdots & t_{2,r}^{(0)} \\
  \vdots & \vdots & \ddots & \vdots \\
  t_{N,1}^{(0)} & t_{N,2}^{(0)} & \cdots & t_{N,r}^{(0)}
\end{pmatrix}

\times
\begin{pmatrix}
  s_{1,1} & 0 & \cdots & 0 \\
  0 & s_{2,2} & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & s_{r,r}
\end{pmatrix}

\times
\begin{pmatrix}
  d_{1,1}^{(0)} & d_{1,2}^{(0)} & \cdots & d_{1,M}^{(0)} \\
  d_{2,1}^{(0)} & d_{2,2}^{(0)} & \cdots & d_{2,M}^{(0)} \\
  \vdots & \vdots & \ddots & \vdots \\
  d_{r,1}^{(0)} & d_{r,2}^{(0)} & \cdots & d_{r,M}^{(0)}
\end{pmatrix}
\end{align*}
\]

Reducing the singular vectors

\[
\hat{X} = 
\begin{pmatrix}
  t_{1,1} & t_{1,2} & \cdots & t_{1,k} \\
  t_{2,1} & t_{2,2} & \cdots & t_{2,k} \\
  \vdots & \vdots & \ddots & \vdots \\
  t_{N,1} & t_{N,2} & \cdots & t_{N,k}
\end{pmatrix}

\times
\begin{pmatrix}
  s_{1,1} & 0 & \cdots & 0 \\
  0 & s_{2,2} & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & s_{k,k}
\end{pmatrix}

\times
\begin{pmatrix}
  d_{1,1} & d_{1,2} & \cdots & d_{1,M} \\
  d_{2,1} & d_{2,2} & \cdots & d_{2,M} \\
  \vdots & \vdots & \ddots & \vdots \\
  d_{k,1} & d_{k,2} & \cdots & d_{k,M}
\end{pmatrix}
\]
Probabilistic IR model

- Adding weights to each term
  \[ D_j = (t_{1, j}, t_{2, j}, ..., t_{N, j}) \]
  \[ t_{i,j} = TRW_i * f_{i,j} \]

- Term Relevance Weight
  \[ TRW_i = \log \left( \frac{p_i \times (1-q_i)}{q_i \times (1-p_i)} \right) \]
  \[ p_i \] Probability of the term appearing in relevant documents
  \[ q_i \] Probability of the term appearing in irrelevant documents
Weighting schema in CodeBroker

\[ sim(Q, D_j) = \sum_{i=1}^{T} (\log \frac{N - n_i + 0.5}{n_i + 0.5}) \frac{(k_1 + 1)tf_{i,j}}{K + tf_{i,j}} \frac{(k_3 + 1)qtf_i}{k_3 + qtf_i} \]

- \( N \) is the number of components
- \( n_i \) is the number of components whose documents contain the term \( ti \)
- \( T \) is the number of terms in the component collection
- \( tf_{i,j} \) is the frequency of term \( ti \) in the document of the component \( Dj \)
- \( qtf_i \) is the frequency of term \( ti \) in the query \( Q \)

\[ K = k_1 ((1 - b) + b \cdot dl_j / avdl) \]

- \( k_1, k_3, b \) are empirically determined parameters depending on the nature of the document collection. In CodeBroker, \( k_1 \) is set to 1.2, \( k_3 \) to 1.0, and \( b \) to 0.75.

- \( dl_j \) is the length of document \( D_j \)
- \( avdl \) is the average length of all documents in the collection
Signature matching determines the constraint compatibility

- Reusable components must be compatible in signature
  - Signature is the syntactic interface of a module (method and class)
  - Improving the precision of retrieval

- Method level match
  - Exact match
    - \( \text{Type}_1 \times \text{Type}_2 \rightarrow \text{Type}_3 \)
    - \( \text{Type}_A \times \text{Type}_B \rightarrow \text{Type}_C \)
    - \( \Leftrightarrow \ \text{Type}_1=\text{Type}_A \ \text{AND} \ \text{Type}_2=\text{Type}_B \ \text{AND} \ \text{Type}_3=\text{Type}_C \)
  - Relaxed match
    - Generalization / Specialization / Reorder
      - \( \text{string} \times \text{int} \rightarrow \text{int} \) matches (relaxed) \( \text{long} \times \text{string} \rightarrow \text{long} \)
Presenter: tailoring the delivery to larger context and user

**User model**: list of components known to the user

**Discourse model**: list of components from uninterested domains
Discourse models: Improving task-relevance

- Discourse models capture the larger context of programming activities
  - Representing the interaction history between programmers and CodeBroker
  - Removing irrelevant components
  - Negative discourse models: specifying what is not of interest to programmers
- Example:
  ```
  (((“java.util.zip”) ;; a package
  (“java.awt” (“CardLayout”))) ;; a class
  ```
User models: User-specific delivery

- User models represent programmers’ knowledge on the component repository
  - A list of known components
  - Example:
    ```
    ("java.applet" ("Applet" ("getParameterInfo")))
    ("java.io" ("File" ("exists"
                          "11/02/00" "11/10/00"
                          "11/11/00")
                          ("isAbsolute"
                          "11/01/00" "11/10/00"
                          "11/11/00"))
    )
    ```
  - Components contained in user models are **not** delivered
Incremental discourse modeling and user modeling

- **Initial** user models
  - Created by analyzing existing user programs

- **Adaptive** user models
  - CodeBroker updates user models automatically when it detects the use of a component in the editor

- **Adaptable** user models and discourse models
  - Using the Skip Components Menu associated with each delivered component
Models in CodeBroker

Component Repository

Existing programs

Session 1
Task 1
Task 2

Session 2
Task 1
Retrieval-by-reformulation

- A process for software developers to incrementally develop reuse queries
- Delivered components help developers become familiar with the vocabulary and structure of the repository
  - Change the way of writing the query
  - Limit the search scope by specifying (un)interested packages and classes
The cycle of delivery-browsing-searching

- Delivered components are results of information reconnaissance
- Possible actions after the delivery
  - The needed component is delivered
    → Choose the needed one through browsing
  - Too many components are delivered
    → Filter the delivered components
  - The needed one is not delivered
    → Search again through retrieval-by-reformulation
Evaluation experiments

- **Experiment goals:**
  - Observe the effectiveness of CodeBroker in encouraging programmers to reuse
  - Analyze the effectiveness of task inference, discourse models, and user models

- **12 experiments with 5 subjects**
  - Implementing an assigned task with CodeBroker

<table>
<thead>
<tr>
<th>Subjects</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Years of prog. in general</td>
<td>3-4</td>
<td>5-6</td>
<td>8</td>
<td>10+</td>
<td>10+</td>
</tr>
<tr>
<td>Java skill (self-evaluation)</td>
<td>4</td>
<td>7</td>
<td>7-8</td>
<td>10</td>
<td>7</td>
</tr>
</tbody>
</table>
## System assessment

<table>
<thead>
<tr>
<th>Sub</th>
<th>No</th>
<th>total</th>
<th>delivered</th>
<th>breakdown of deliveries</th>
<th>triggered</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>unanticipated (L4-L3)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>anticipated but unknown (L3)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>vaguely known (L2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>1</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>S2</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>S3</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S4</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>S5</td>
<td>11</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sum</td>
<td>57</td>
<td>20</td>
<td>9</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

vaguely known (L2)  
unanticipated (L4-L3)  
anticipated but unknown (L3)  
triggered
The STeP_IN system: a SocioTechnical Platform for in situ Networking
Three proximities for knowledge collaboration

- **Cognitive proximity**
  - Defines the transferability and combinability of knowledge

- **Structural proximity**
  - Provides communication channels for knowledge to flow

- **Relational proximity**
  - Determines the motivation to participate in knowledge collaboration
Dynamic community

- A dynamic community is a small group of knowledge workers that forms ad hoc in support of a particular user working on a particular task, and dissembles as the task is finished.

- Dynamic communities support situated knowledge collaboration by mobilizing positive forces in all three proximities.
Set of people \( \Psi = \{A, B, C, D, E, M, N, O, P, Q\} \)
Set of information \( \Phi = \{\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \eta, \theta\} \)
Relation between information

\[ \mathbb{II} = \{ (\alpha, \beta), (\alpha, \gamma), (\beta, \varepsilon), (\beta, \delta), (\gamma, \theta), (\zeta, \eta) \} \]
Relation between people and information

\( PI = \{(B, \alpha), (C, \alpha), (M, \alpha), (B, \beta), (D, \beta), (E, \gamma), (N, \gamma), (D, \delta), (P, \varepsilon), (O, \zeta), (Q, \eta), (Q, \theta)\} \)
Knowledge Work Space

Relation between people

\[ PP = \{ (A, B), (A, C), (A, D), (A, E), (A, O), (A, P), \\
(D, O), (E, N), (E, Q), (M, P), (M, N), (O, Q) \} \]
Knowledge Work Space

\[ KWS = \{ (a, \beta), (a, \gamma), (\beta, \varepsilon), (\beta, \delta), (\gamma, \theta), (\zeta, \eta) \}, \]
\{ (B, a), (C, a), (M, a), (B, \beta), (D, \beta), (E, \gamma), (N, \gamma), (D, \delta), (P, \varepsilon), (O, \zeta), (Q, \eta), (Q, \theta) \}, \}
\{ (A, B), (A, C), (A, D), (A, E), (A, O), (A, P), (D, O), (E, N), (E, Q), (M, P), (M, N), (O, Q) \} \} \]
The forming process of a DynC
Triggering event for $\text{Dyn}(A, \alpha)$
From information to information
From information to people (experts)
From people to people
\(Dyn(A, \alpha) = \{A, B, C, D, E\}\)
$\text{Dyn}(A, \alpha) = \{A, B, C, D, E\}$
Task-specific and member-specific

DynC(A, α) = {A, B, C, D, E}

DynC(A, β) = {A, B, C, D, P}

DynC(N, α) = {E, N, M}
DynC formation support subsystem

Dynamic Community Formation

1. User & info
2. Locate relevant info
3. Identifying experts
4. Selecting experts
5. DynC members

Information repository
- Relation between info
- Relation between info & people
- Relation between people
STeP_IN: programming with external knowledge resources

- Individualized search for API methods
- Accumulating and showing use examples
- Getting help from peers (a DynC approach)
  - Who are the experts?
    - Experts can only be identified after task is known
  - Who is willing to help?
    - Utilizing existing social relations
Technical Profile initialization

<table>
<thead>
<tr>
<th>package</th>
<th>class</th>
<th>method</th>
<th># of definitions</th>
<th># of references</th>
<th>check</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang</td>
<td>StringBuffer</td>
<td>java.lang.String</td>
<td>0</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>java.lang</td>
<td>StringBuffer</td>
<td>java.lang.StringBuffer</td>
<td>0</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>java.lang</td>
<td>StringBuffer</td>
<td>StringBuffer()</td>
<td>0</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td><em>p.co.sra.smalltalk</em></td>
<td><em>STObject</em></td>
<td><em>STObject</em>()</td>
<td>1</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>java.lang</td>
<td>Object</td>
<td>Object()</td>
<td>0</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>java.awt</td>
<td>Point</td>
<td>Point(int, int)</td>
<td>0</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>java.io</td>
<td>PrintStream</td>
<td>void</td>
<td>0</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td><em>p.co.sra.smalltalk</em></td>
<td><em>Stimage</em></td>
<td><em>Stimage</em>()</td>
<td>1</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td><em>p.co.sra.smalltalk</em></td>
<td><em>STRectangle</em></td>
<td><em>STRectangle</em>(int,)</td>
<td>1</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>java.lang</td>
<td>Object</td>
<td>java.lang.Class</td>
<td>0</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>java.util</td>
<td>Vector</td>
<td>int size()</td>
<td>0</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td><em>p.co.sra.smalltalk</em></td>
<td><em>Stimage</em></td>
<td>int width()</td>
<td>1</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td><em>p.co.sra.smalltalk</em></td>
<td><em>Stimage</em></td>
<td>int height()</td>
<td>1</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>java.lang</td>
<td>StringBuffer</td>
<td>java.lang.StringBuffer</td>
<td>0</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td><em>p.co.sra.smalltalk</em></td>
<td><em>SmalltalkException</em></td>
<td><em>SmalltalkException</em>()</td>
<td>1</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td><em>p.co.sra.smalltalk</em></td>
<td><em>SmalltalkException</em></td>
<td><em>SmalltalkException</em>()</td>
<td>1</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>java.util</td>
<td>ArrayList</td>
<td>int size()</td>
<td>0</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>java.lang</td>
<td>String</td>
<td>boolean</td>
<td>0</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td><em>p.co.sra.smalltalk</em></td>
<td><em>SmalltalkTestExar</em></td>
<td><em>SmalltalkTestExar</em>()</td>
<td>1</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td><em>p.co.sra.smalltalk</em></td>
<td><em>STObject</em></td>
<td><em>STObject</em>()</td>
<td>1</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>java.util</td>
<td>Vector</td>
<td>java.lang.Object</td>
<td>0</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td><em>p.co.sra.smalltalk</em></td>
<td><em>STBlockClosure</em></td>
<td><em>STBlockClosure</em>()</td>
<td>1</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>java.awt.event</td>
<td>WindowAdapter</td>
<td>WindowAdapter()</td>
<td>0</td>
<td>14</td>
<td></td>
</tr>
</tbody>
</table>
Social Profile Initialization

![Image of Social Profile Initialization](image_url)
Individualized search
Search engine for Java programs

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>generate random numbers</td>
<td>Generates a random value from this distribution.</td>
</tr>
<tr>
<td>reSeed</td>
<td>Reseeds the random number generator with the current seed.</td>
</tr>
<tr>
<td>nextSecureInt</td>
<td>Generates a random int value uniformly distributed.</td>
</tr>
<tr>
<td>nextSecureLong</td>
<td>Generates a random long value uniformly distributed.</td>
</tr>
<tr>
<td>reSeedSecure</td>
<td>Reseeds the secure random number generator with the supplied seed.</td>
</tr>
<tr>
<td>nextSample</td>
<td>Uses a 2-cycle permutation shuffle to generate a random sample.</td>
</tr>
<tr>
<td>reSeedSecure</td>
<td>Reseeds the secure random number generator with the supplied seed.</td>
</tr>
<tr>
<td>nextHexString</td>
<td>Generates a random string of hex characters of length n.</td>
</tr>
<tr>
<td>getNextValue</td>
<td>Generates a random value from this distribution.</td>
</tr>
</tbody>
</table>
| nextInt | Generates a random int value uniformly distributed.
Search engine for Java programs

![Image of Codebroker interface]

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>nextValue</code></td>
<td>Generates a random value from the distribution</td>
<td></td>
</tr>
<tr>
<td><code>reSeed</code></td>
<td>Reseeds the random number generator with the current state</td>
<td></td>
</tr>
<tr>
<td><code>nextSecureInt</code></td>
<td>Generates a random int value uniformly distributed</td>
<td></td>
</tr>
<tr>
<td><code>nextSecureLong</code></td>
<td>Generates a random long value uniformly distributed</td>
<td></td>
</tr>
<tr>
<td><code>reSeedSecure</code></td>
<td>Reseeds the secure random number generator with the supplied seed</td>
<td></td>
</tr>
<tr>
<td><code>nextInt</code></td>
<td>Uses a 2-cycle permutation shuffle to generate a value in the range</td>
<td></td>
</tr>
<tr>
<td><code>reSeedSecure</code></td>
<td>Reseeds the secure random number generator with the supplied seed</td>
<td></td>
</tr>
<tr>
<td><code>nextHexString</code></td>
<td>Generates a random string of hex characters of length</td>
<td></td>
</tr>
<tr>
<td><code>nextValue</code></td>
<td>Generates a random value from the distribution</td>
<td></td>
</tr>
</tbody>
</table>
Search engine for Java programs

Specifying packages or classes not to search
# Search engine for Java programs

![Codebroker web interface](image)

**Codebroker**

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Class Name</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>reSeed</td>
<td>org.apache.commons.math.random.RandomDataImpl</td>
<td>Reseed the random number generator with the current state.</td>
<td>none</td>
</tr>
<tr>
<td>reSeedSecure</td>
<td>org.apache.commons.math.random.RandomDataImpl</td>
<td>Reseed the secure random number generator with the current state.</td>
<td>none</td>
</tr>
<tr>
<td>nextSample</td>
<td>org.apache.commons.math.random.RandomDataImpl</td>
<td>Uses a 2-cycle permutation shuffle to generate a random element.</td>
<td>java.util.Collection c, int k</td>
</tr>
<tr>
<td>reSeedSecure</td>
<td>org.apache.commons.math.random.RandomDataImpl</td>
<td>Reseed the secure random number generator with the current state.</td>
<td>none</td>
</tr>
<tr>
<td>nextHexString</td>
<td>org.apache.commons.math.random.RandomData</td>
<td>Generates a random string of hex characters of length.</td>
<td>java.lang.String nextHexString(int len)</td>
</tr>
<tr>
<td>nextInt</td>
<td>org.apache.commons.math.random.RandomData</td>
<td>Generate a random int value uniformly distributed in the range.</td>
<td>java.util.Random nextInt(int lower, int upper)</td>
</tr>
<tr>
<td>nextLong</td>
<td>org.apache.commons.math.random.RandomData</td>
<td>Generate a random long value uniformly distributed in the range.</td>
<td>java.util.Random nextLong(long lower, long upper)</td>
</tr>
<tr>
<td>nextPermutation</td>
<td>org.apache.commons.math.random.randomEmpiricalDistribut</td>
<td>Uses a random permutation shuffle to generate a random permutation.</td>
<td>none</td>
</tr>
</tbody>
</table>
Search engine for Java programs

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getNextValue</td>
<td>Generates a random value from this distribution.</td>
</tr>
<tr>
<td>reSeed</td>
<td>Reseeds the random number generator with the supplied seed.</td>
</tr>
<tr>
<td>reSeedSecure</td>
<td>Reseeds the secure random number generator with the supplied seed.</td>
</tr>
<tr>
<td>nextSample</td>
<td>Uses a 2-cycle permutation shuffle to generate a random sample.</td>
</tr>
<tr>
<td>reSeedSecure</td>
<td>Reseeds the secure random number generator with the supplied seed.</td>
</tr>
<tr>
<td>nextHexString</td>
<td>Generates a random string of hex characters of length len.</td>
</tr>
</tbody>
</table>

Specifying packages or classes to search
Search engine for Java programs

![Image of Codebroker web interface](http://chum.cs.colorado.edu/cgi-bin/web/query.cgi?query=generate+random+numbers&model=range=0-200&language=en)

**Codebroker**

Search for: `generate random numbers`

Model: `+org.apache.commons.math.random.EmpiricalDistribut...`

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getNextValue</code></td>
<td>Generates a random value from this distribution.</td>
</tr>
<tr>
<td><code>getBinCount</code></td>
<td>Returns the number of bins.</td>
</tr>
<tr>
<td><code>load</code></td>
<td>Computes the empirical distribution from the provided data array.</td>
</tr>
</tbody>
</table>

© 2006 Yunwen Ye
## Search engine for Java programs

![Codebroker web interface - Microsoft Internet Explorer](image)

**Specifying packages at any hierarchical level**

<table>
<thead>
<tr>
<th>Package</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>org.apache.commons.math.random</td>
<td>getUniform()</td>
<td>Generates a random number uniformly distributed.</td>
</tr>
<tr>
<td>org.apache.commons.math.random</td>
<td>nextSecureInt()</td>
<td>Generates a random integer uniformly distributed.</td>
</tr>
<tr>
<td>org.apache.commons.math.random</td>
<td>nextSecureLong()</td>
<td>Generates a random long uniformly distributed.</td>
</tr>
<tr>
<td>org.apache.commons.math.random</td>
<td>reSeed()</td>
<td>Reseeds the random number generator with the supplied seed.</td>
</tr>
<tr>
<td>org.apache.commons.math.random</td>
<td>reSeedSecure()</td>
<td>Reseeds the secure random number generator with the supplied seed.</td>
</tr>
<tr>
<td>org.apache.commons.math.random</td>
<td>nextSample()</td>
<td>Uses a 2-cycle permutation shuffle to generate a random sample.</td>
</tr>
<tr>
<td>org.apache.commons.math.random</td>
<td>reSeedSecure()</td>
<td>Reseeds the secure random number generator with the supplied seed.</td>
</tr>
<tr>
<td>org.apache.commons.math.random</td>
<td>nextHexString()</td>
<td>Generates a random string of hex characters of length.</td>
</tr>
</tbody>
</table>

© 2006 Yunwen Ye
Extended Java API documents

```java
public Process exec(String command)
    throws IOException;

Executes the specified string command in a separate process.

Parameters:
  command - a specified system command.

Returns:
  a Process object for managing the subprocess.

Throws:
  SecurityException - if a security manager exists and its checkExec method doesn't allow creation of a subprocess.
  IOException - if an I/O error occurs
  NullPointerException - if command is null
  IllegalArgumentException - if command is empty

See Also:
```
Examples for

java.lang.Runtime
java.lang.Process exec(java.lang.String)

This example is provided by m-asada.
0 found it helpful.
0 found it not helpful.

```java
//executing an external unix command ls
try {
    String command = "ls";
    Process childprocess = Runtime.getRuntime().exec(command);
} catch (IOException e) { }
```

This example is ✔ Helpful  ❌ Not helpful

evaluate
Ask the experts

Requesting collaboration from experts on

```java
java.lang.Runtime
    .java.lang.Process exec(java.lang.String)
```

Subject: reading outcome of external command

Question: How can I read the output of the external command?
Mail sent to selected experts

m-asada has requested help on
java.lang.Runtime
Process exec (String)

Your email address will be automatically changed to your STeP_IN UserName when your reply is sent. Please don't include your signature if you want to remain to be known only by your UserName in STeP_IN.

--------Question Contents---------
How can I read the output of the external command?
Reply from experts

This is a message from yunwen sent to the DynC initiated by m-asada.

------Message Contents------
to read the output, you can do something like this

InputStream instr = process.getInputStream(); // process is the name of the command
int c;
while ((c=instr.read()) != -1) {
    DoSomethingWith((char) c);
}
instr.close();

Did I make myself clear?

On Dec 12, 2005, dyn+10@ktlsqj.sra.co.jp wrote:

m-asada has requested help on
java.lang.Runtime
Process exec (String)

Your email address will be automatically changed to your STeP_IN UserName when your reply is sent.
Please don't include your signature if you want to remain to be known only by your UserName in STeP_IN.
DynC evaluation

m-asada's Involved DynC

Active DynC you initiated

<table>
<thead>
<tr>
<th>DynC ID</th>
<th>Subject</th>
<th>Method</th>
<th>Start Date</th>
<th>Helpful</th>
<th>Not Helpful</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>さぶじゅくと</td>
<td>SecurityManager, int.classLoaderDepth()</td>
<td>2005/12/02 15:10:56</td>
<td>☑ helpful</td>
<td>☐ not helpful</td>
</tr>
<tr>
<td>8</td>
<td>SUBJECT</td>
<td>SecurityManager, void.checkList(int)</td>
<td>2005/12/02 15:08:28</td>
<td>☑ helpful</td>
<td>☐ not helpful</td>
</tr>
<tr>
<td>7</td>
<td>SUBJECT</td>
<td>SecurityManager, void.checkWrite(java.lang.String)</td>
<td>2005/12/02 15:07:13</td>
<td>☑ helpful</td>
<td>☐ not helpful</td>
</tr>
<tr>
<td>6</td>
<td>SUBJECT</td>
<td>SecurityManager, void.checkSetFactory()</td>
<td>2005/12/02 15:00:38</td>
<td>☑ helpful</td>
<td>☐ not helpful</td>
</tr>
</tbody>
</table>

This DynC was helpful; thank you for your help.
This DynC was not helpful; thank you for participation.
Discussion archive

Discussion Archive for: java.lang.Runtime
    java.lang.Process exec(java.lang.String)

DynCs:

- reading outcome of external command
  m-asada
  2005/12/13 08:29:43

Discussions:

From: m-asada
Subject: reading outcome of external command
Date: 2005/12/13 08:29:43

How can I read the output of the external command?

From: yunwen
Subject: Re: [DynC 10] reading outcome of external command
Date: 2005/12/13 08:43:30

to read the output, you can do something like this

```
InputStream instr = process.getInputStream(); // process is name of the command
int c;
while((c=instr.read()) != -1) {
    DoSomethingWith((char) c);
}
instr.close();
```

Did I make myself clear?

On Dec 12, 2005, at 4:29 PM, dync+10@ktlsvj.sra.co.jp wrote

> m-asada has requested help on
> java.lang.Runtime
DynC formation

- Expert identification
- Expert selection
Identifying experts

- Expert is a relative attribute
  - Only after a question is known, experts can be identified
- Creating user profiles by analyzing programs they have developed
  - Software developers who have used the method of interest are candidate experts
  - Links from methods to software developers
Expert selection

Level 1: Confirmed expertise
Level 2: Claimed expertise
Level 3: Inferred expertise
Level 4: Future expertise
Expert identification

Level 1: confirmed expertise

### vincent's Involved DynC

<table>
<thead>
<tr>
<th>DynC ID</th>
<th>Subject</th>
<th>Method</th>
<th>Start Date</th>
<th>End Date</th>
<th>Helpful/Not helpful</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>attributes of file after setReadOnly</td>
<td>File boolean setReadOnly()</td>
<td>2008/03/04</td>
<td>2008/03/04</td>
<td>helpful</td>
</tr>
<tr>
<td>8</td>
<td>help on exec</td>
<td>Runtime java.lang.Process exec(java.lang.String)</td>
<td>2008/03/02</td>
<td>2008/03/02</td>
<td><strong>helpful</strong></td>
</tr>
</tbody>
</table>

Send email to DynC #6
Expert identification

Level 2: claimed expertise

<table>
<thead>
<tr>
<th>Method</th>
<th>Author</th>
<th>Usage #</th>
<th>Declare</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang Object void &lt;init&gt;()</td>
<td>No</td>
<td>1</td>
<td>Expert, Not Declared, No Knowledge</td>
</tr>
<tr>
<td>java.lang Process InputStream getInputStream()</td>
<td>No</td>
<td>1</td>
<td>Expert, Not Declared, No Knowledge</td>
</tr>
<tr>
<td>java.lang Process OutputStream getOutputStream()</td>
<td>No</td>
<td>1</td>
<td>Expert, Not Declared, No Knowledge</td>
</tr>
<tr>
<td>java.lang Runtime Process exec(java.lang.String)</td>
<td>No</td>
<td>1</td>
<td>Expert, Not Declared, No Knowledge</td>
</tr>
<tr>
<td>java.lang Runtime getRuntime()</td>
<td>No</td>
<td>1</td>
<td>Expert, Not Declared, No Knowledge</td>
</tr>
<tr>
<td>java.lang StringBuffer void &lt;init&gt;()</td>
<td>No</td>
<td>1</td>
<td>Expert, Not Declared, No Knowledge</td>
</tr>
<tr>
<td>java.lang StringBuffer append(java.lang.String)</td>
<td>No</td>
<td>1</td>
<td>Expert, Not Declared, No Knowledge</td>
</tr>
<tr>
<td>java.lang StringBuffer</td>
<td>No</td>
<td>1</td>
<td>Expert, Not Declared, No Knowledge</td>
</tr>
</tbody>
</table>
Expert identification

Level 3: inferred expertise

<table>
<thead>
<tr>
<th>Method</th>
<th>Author</th>
<th>Usage #</th>
<th>Declare</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang Object void &lt;init&gt;()</td>
<td>No</td>
<td>12</td>
<td><img src="expert.svg" alt="Expert" /> <img src="not-declared.svg" alt="Not Declared" /> <img src="no-knowledge.svg" alt="No Knowledge" /></td>
</tr>
<tr>
<td>java.lang Object java.lang.String toString()</td>
<td>No</td>
<td>2</td>
<td><img src="expert.svg" alt="Expert" /> <img src="not-declared.svg" alt="Not Declared" /> <img src="no-knowledge.svg" alt="No Knowledge" /></td>
</tr>
<tr>
<td>java.lang Runtime java.lang.Process exec(java.lang.String)</td>
<td>No</td>
<td>2</td>
<td><img src="expert.svg" alt="Expert" /> <img src="not-declared.svg" alt="Not Declared" /> <img src="no-knowledge.svg" alt="No Knowledge" /></td>
</tr>
<tr>
<td>java.lang String void &lt;init&gt;(java.lang.StringBuffer)</td>
<td>No</td>
<td>2</td>
<td><img src="expert.svg" alt="Expert" /> <img src="not-declared.svg" alt="Not Declared" /> <img src="no-knowledge.svg" alt="No Knowledge" /></td>
</tr>
<tr>
<td>java.lang String void &lt;init&gt;(byte[], java.lang.String)</td>
<td>No</td>
<td>1</td>
<td><img src="expert.svg" alt="Expert" /> <img src="not-declared.svg" alt="Not Declared" /> <img src="no-knowledge.svg" alt="No Knowledge" /></td>
</tr>
<tr>
<td>java.lang String char charAt(int)</td>
<td>No</td>
<td>4</td>
<td><img src="expert.svg" alt="Expert" /> <img src="not-declared.svg" alt="Not Declared" /> <img src="no-knowledge.svg" alt="No Knowledge" /></td>
</tr>
<tr>
<td>java.lang String</td>
<td>No</td>
<td>3</td>
<td><img src="expert.svg" alt="Expert" /> <img src="not-declared.svg" alt="Not Declared" /> <img src="no-knowledge.svg" alt="No Knowledge" /></td>
</tr>
</tbody>
</table>
Selecting experts

- From expert candidates, select those who are most likely to help
  - Those I have helped recently
  - Those I have interacted through emails
Selecting experts

4 relations

help<A,B,t>
friend<A,B>
exclude<A,B>
email<A,B>
Selecting experts

4 relations

help\langle A, B, t \rangle
friend\langle A, B \rangle
exclude\langle A, B \rangle
email\langle A, B \rangle

A (m-asada) participated in a DynC initiated by B (nisinaka) at time t
Selecting experts

4 relations

help<A,B,t>
friend<A,B>
exclude<A,B>
email<A,B>

A (m-asada) declares that he will always participate in B (nisinaka)'s DynC in the future
Selecting experts

4 relations

help\langle A,B,t\rangle
friend\langle A,B\rangle
exclude\langle A,B\rangle
e-mail\langle A,B\rangle

A (m-asada) delcares that he will never participate in B (yxy)'s DynC in the future
Selecting experts

4 relations

help\(\langle A,B,t\rangle\)

friend\(\langle A,B\rangle\)

exclude\(\langle A,B\rangle\)

email\(\langle A,B\rangle\)

The number of emails that A (nisinaka) has sent to B (m-asada)
Selecting experts based on 4 relations

For each person X in identified expert lists
1. If exclude<X, A>, X is removed from the list  
   because X declared he will never participate in A’s DynC
2. If friend<X, A>, X is selected  
   because X declared he will always participated in A’s DynC
3. If |help<A, X, t>| > |help<X, A, t>|, X is selected  
   because A has helped X more times than X did A
4. If help<A, X, t> and t is more recent than help<X, A, t>, X is selected  
   because A has recently helped X
5. Selecting from remaining experts on the order of email<X, A>  
   because X should have known A well if X has send many emails to A
6. Selecting from remaining experts those who got most help by other members
Social awareness communication

- Acknowledging publicly member participation to motivate
  - When experts response, members of the DynC know
  - Top contributors list

- Avoiding forced collaboration
Avoiding forced collaboration

- When request for help is sent to experts, recipients are hidden from requesters and other experts.

---

Image: Email message from m-asada asking for help with Java. The question is: How can I read the output of the external command?
Avoiding forced collaboration

- Selected experts can quietly withdraw from the DynC without anyone noticing

Notifying the formation of a new DynC
Avoiding forced collaboration

- Selected experts can quietly withdraw from the DynC without anyone noticing

Leave the current DynC

Notifying the formation of a new DynC
Avoiding forced collaboration

- Selected experts can quietly withdraw from the DynC without anyone noticing

Decline future participation in his/her DynC

Notifying the formation of a new DynC
Avoiding forced collaboration

- Selected experts can quietly withdraw from the DynC without anyone noticing

- Notify the formation of a new DynC on an API component

- Decline future participation of DynC on an API component
Social awareness communication

- Sending emails to ask the experts
  - Providing excuse space
    - A dose not know who are the recipients
  - Publicly acknowledgement
    - Any experts who answered the questions are made known
  - Easy exit
    - “Don’t bother me anymore about this problem”
    - “I don’t want to have more request emails from A”—establishing exclude<X, A>
Summary

- Better understanding of cognitive difficulties of component reuse
  - Unknown components
  - Low reuse utility
- A new type of component repository systems
  - Active component repository systems
- Integrating technology support with social support