Lecture 26

Empirical Studies of Clone Evolution
Clone Genealogies

Today's Agenda (1)

- Class Presentation
  - Meiru Che
  - Amal Banerjee
- Course Evaluation
  - I need a volunteer to collect and deposit course evaluation forms.

Today's Agenda (2)

- Discussion on practical implications of SE research
- Discussion on "An Empirical Study of Clone Genealogies"

Recap of CCFinder

- CCFinder is a robust and scalable clone detector.
- It transforms a program to a parameterized token sequence using language dependent transformation rules.
- It then use a suffix tree algorithm to find common contiguous subsequences.
- Its case studies show that CCFinder can be applied to industrial size programs.

Class Presentations

- Advocate: Meiru
- Skeptic: Amal

Course-Instructor Survey

- Instructor's Name: Kim, Miryung
  - This survey is for the instructor, not TA.
- Course Abbreviation and Number: EE382V Software Evolution
- Course Unique Number: 16730
- Semester and Year: Spring 2009

Discussion - Refactoring

- What is a definition of refactoring?

Discussion - Information Hiding

- What did you learn from the class activity on refactoring?
  - (1) What do you need to consider before restructuring a program?

Discussion - Information Hiding

- What did you learn from the class activity on refactoring?
  - (2) What do you need to consider after restructuring a program?

Discussion - Information Hiding

- What is the Information Hiding Principle?
-

Discussion - Information Hiding

- How can you apply the Information Hiding Principle to your software design process?
-

Program Differencing

- Which tool do you current use to compare program versions?

- Why is program differencing important in software evolution research?

Program Differencing

- In this colurse, you have studied many different types of program differencing tools, such as diff, AST-based diff, Jdiff, UMLDiff, and LogicalStructuralDiff.
  - (1) Pick one of the above tools and describe its key ideas and benefits of using it.

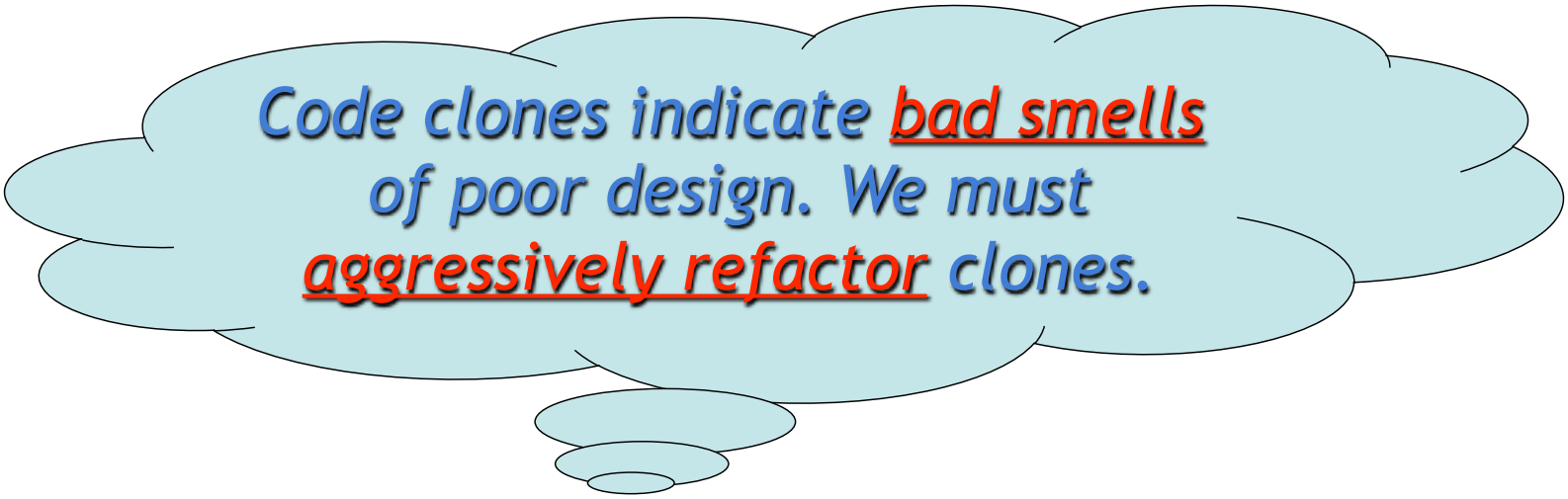
Program Differencing

- In this colurse, you have studied many different types of program differencing tools, such as diff, AST-based diff, Jdiff, UMLDiff, and LogicalStructuralDiff.
  - (2) How will you apply these key ideas in the absence of the program differencing tool that can run on your codebase?

Clone Genealogy

- An Empirical Study of Code Clone Genealogies, Kim et al. ESEC/FSE 2005
  - Studies of code clone evolution
  - Mining software repositories research
  - Its study results challenged one of the most widely-held conventional wisdom about clones.

Conventional Wisdom




```
public void updateFrom  (Class c ) {
      String cType = Util.makeType(c.Name());
      if (seenClasses.contains(cType)) {
         return;
      }
      seenClasses.add(cType);
      if (hierarchy != null) {
         ....
      }
      ...
```

```
public void updateFrom  (ClassReader cr ) {
      String cType =CTD.convertType (c.Name());
      if (seenClasses.contains(cType)) {
         return;
      }
      seenClasses.add(cType);
      if (hierarchy != null) {
        ....
      }
      ...
```

Our Previous Study of Copy and Paste Programming Practices at IBM

[Kim et al. ISESE2004]

- Even skilled programmers often **create and manage** code clones with clear intent.
  - Programmers cannot refactor clones because of **programming language limitations.**
  - Programmers **keep** and **maintain clones** until they realize how to abstract the common part of clones.
  - Programmers often **apply similar changes** to clones.

Research Questions

How do clones evolve over time?

- consistently changed?
- long-lived (or short-lived)?
- easily refactorable?

Previous Studies of Code Clones

- automatic clone detection
  - **lexical, syntactic (AST or PDG), metric, etc.**
- studies of clone coverage ratio
  - **gcc (8.7%), JDK (29%), Linux (22.7%), etc.**
- studies of clone coverage change
  - **changes of clone coverage in Linux [Antoniol+02], [Li+04]**

These studies do not answer how individual clones changed with respect to other clones.

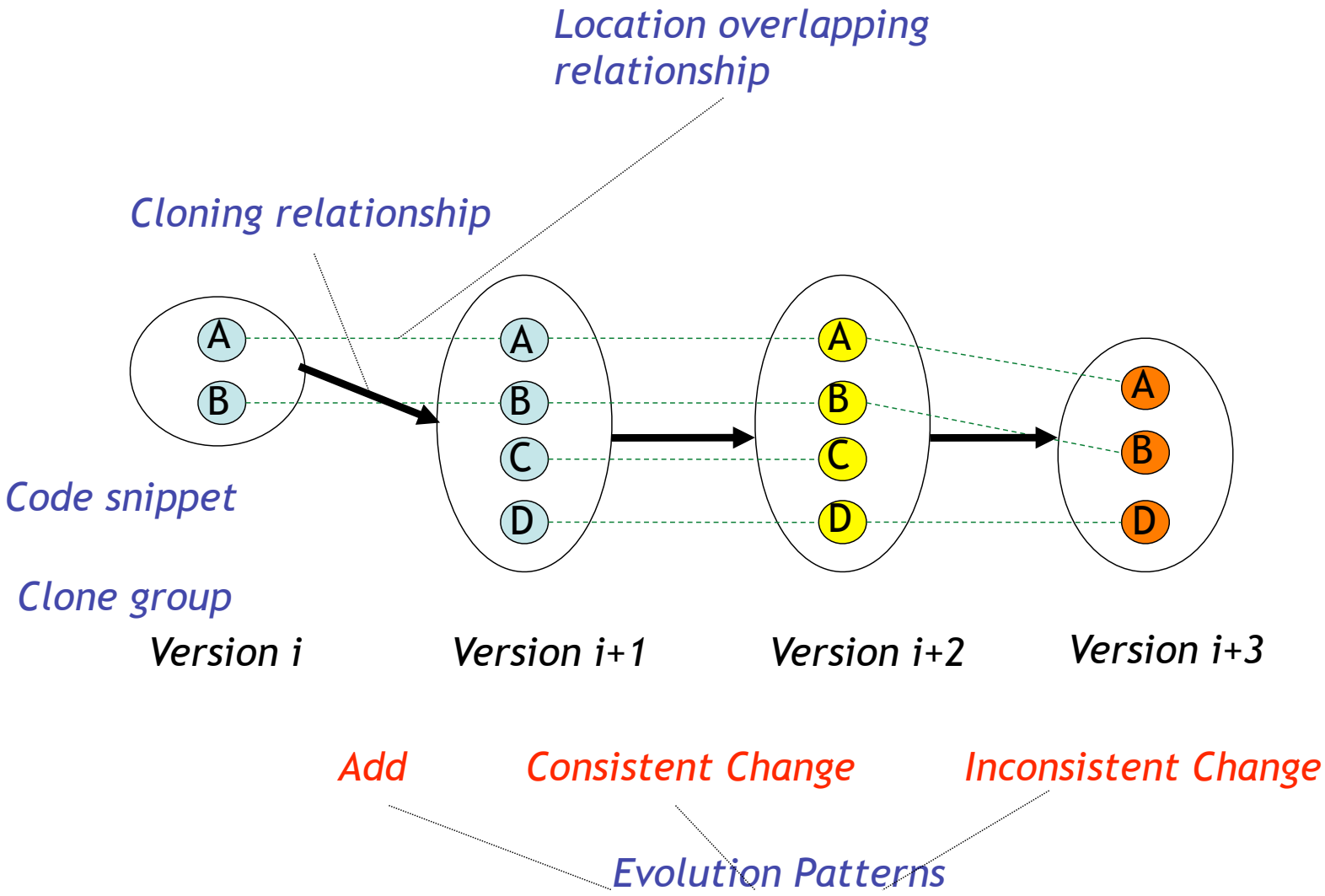
Outline

☑motivation

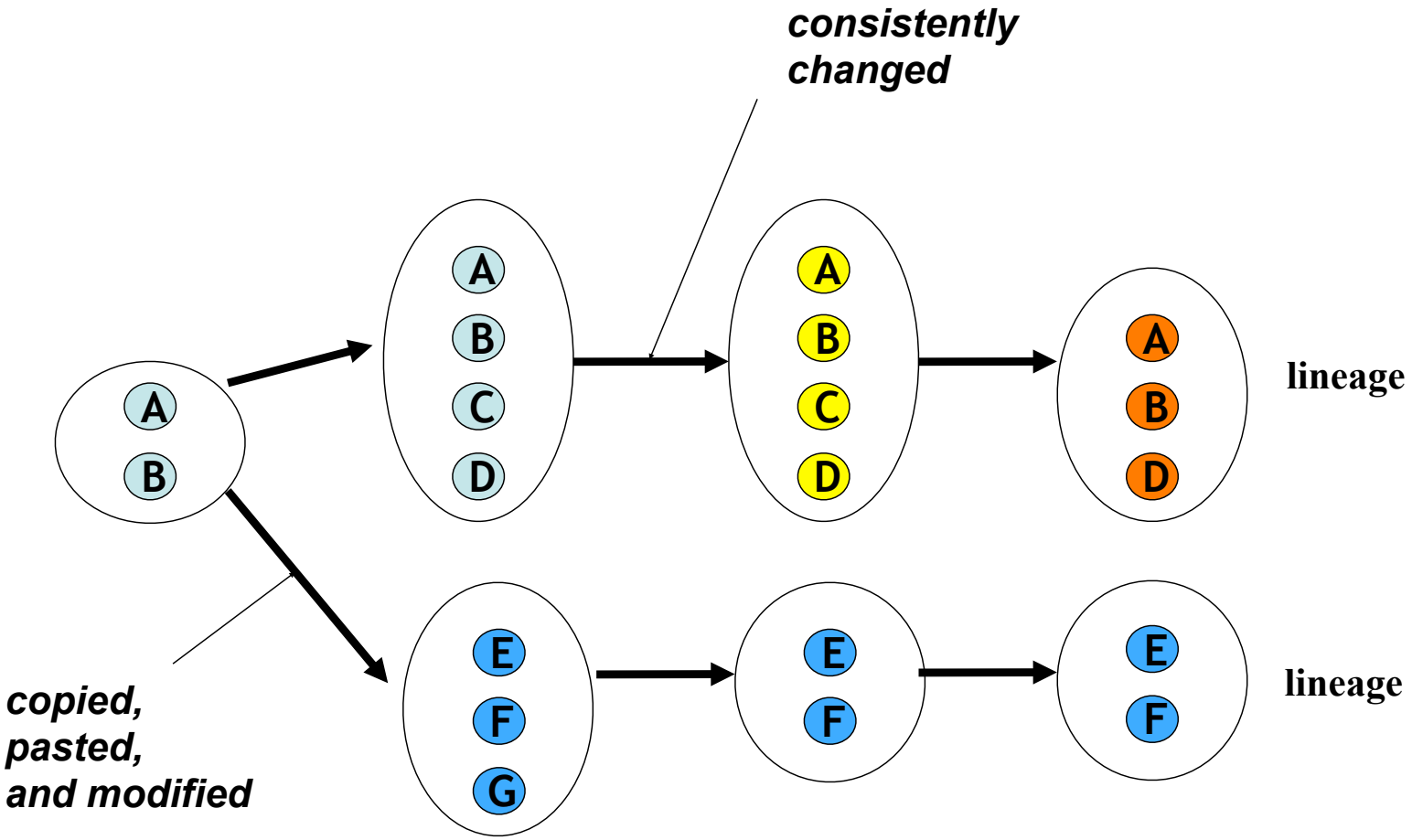
❑clone genealogy : model and tool

❑study procedure and results

Model of Clone Evolution



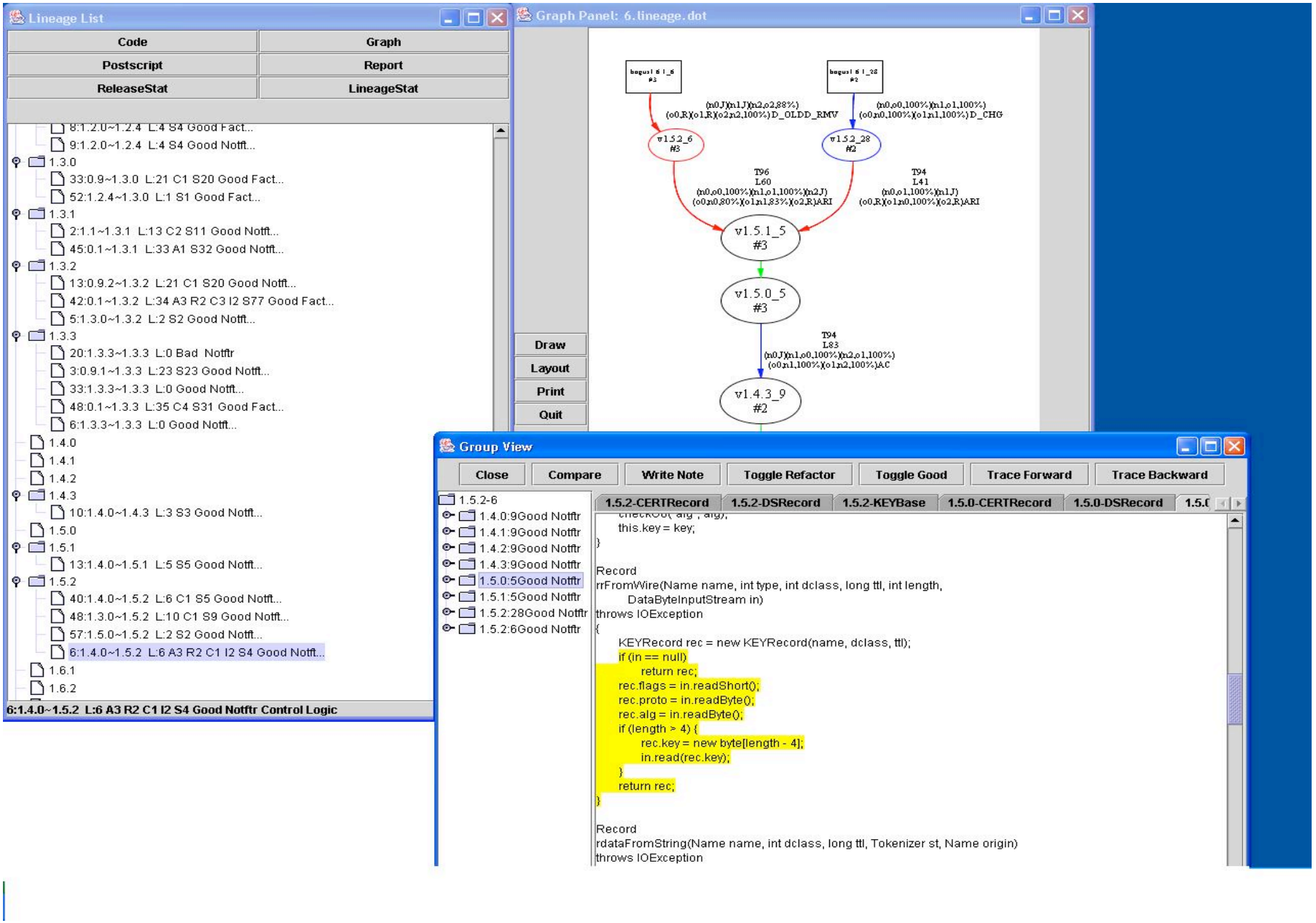
Clone genealogy is a set of clone groups connected by cloning relationships over time.



Clone Genealogy Extractor (CGE)

Given multiple versions of a program, V_k for $1 \leq k \leq n$.

- find clone groups in each version using CCFinder.
- find cloning relationships among clone groups of V_i and V_{i+1} using CCFinder.
- map clones of V_i and V_{i+1} using diff based algorithm.
- separate each connected component of cloning relationships (a clone genealogy).
- identify clone evolution patterns in each genealogy.


Lineage List
Code
Graph
Postscript
Report
ReleaseStat
LineageStat
8:1.2.0~1.2.4 L:4 S4 Good Fact...
9:1.2.0~1.2.4 L:4 S4 Good Notft...
1.3.0
33:0.9~1.3.0 L:21 C1 S20 Good Fact...
52:1.2.4~1.3.0 L:1 S1 Good Fact...
1.3.1
2:1.1~1.3.1 L:13 C2 S11 Good Notft...
45:0.1~1.3.1 L:33 A1 S32 Good Notft...
1.3.2
13:0.9.2~1.3.2 L:21 C1 S20 Good Notft...
42:0.1~1.3.2 L:34 A3 R2 C3 I2 S77 Good Fact...
5:1.3.0~1.3.2 L:2 S2 Good Notft...
1.3.3
20:1.3.3~1.3.3 L:0 Bad Notftr
3:0.9.1~1.3.3 L:23 S23 Good Notft...
33:1.3.3~1.3.3 L:0 Good Notft...
48:0.1~1.3.3 L:35 C4 S31 Good Fact...
6:1.3.3~1.3.3 L:0 Good Notft...
1.4.0
1.4.1
1.4.2
1.4.3
10:1.4.0~1.4.3 L:3 S3 Good Notft...
1.5.0
1.5.1
13:1.4.0~1.5.1 L:5 S5 Good Notft...
1.5.2
40:1.4.0~1.5.2 L:6 C1 S5 Good Notft...
48:1.3.0~1.5.2 L:10 C1 S9 Good Notft...
57:1.5.0~1.5.2 L:2 S2 Good Notft...
6:1.4.0~1.5.2 L:6 A3 R2 C1 I2 S4 Good Notft...
1.6.1
1.6.2
6:1.4.0~1.5.2 L:6 A3 R2 C1 I2 S4 Good Notftr Control Logic
Graph Panel: 6.lineage.dot
Draw
Layout
Print
Quit
v1.5.2_6
#3
v1.5.2_28
#2
v1.5.1_5
#3
v1.5.0_5
#3
v1.4.3_9
#2
Group View
Close
Compare
Write Note
Toggle Refactor
Toggle Good
Trace Forward
Trace Backward
1.5.2-6
1.4.0:9Good Notftr
1.4.1:9Good Notftr
1.4.2:9Good Notftr
1.4.3:9Good Notftr
1.5.0:5Good Notftr
1.5.1:5Good Notftr
1.5.2:28Good Notftr
1.5.2:6Good Notftr
1.5.2-CERTRecord
1.5.2-DSRecord
1.5.2-KEYBase
1.5.0-CERTRecord
1.5.0-DSRecord
this.key = key;
}
Record
rrFromWire(Name name, int type, int dclass, long ttl, int length,
DataByteInputStream in)
throws IOException
{
KEYRecord rec = new KEYRecord(name, dclass, ttl);
if (in == null)
return rec;
rec.flags = in.readShort();
rec.proto = in.readByte();
rec.alg = in.readByte();
if (length > 4) {
rec.key = new byte[length - 4];
in.read(rec.key);
}
return rec;
}
Record
rdataFromString(Name name, int dclass, long ttl, Tokenizer st, Name origin)
throws IOException

Outline

- ☑motivation
- ☑clone genealogy : model and tool
- ❑study procedure and results

Two Java Subject Programs

Program	*carol*	*dnsjava*
LOC	7878 ~ 23731	5756 ~ 21188
Duration	2 years 2 months	5 years 8 months
versions	37	224

versions: a set of check-in snapshots that increased or decreased the total lines of code clones