Activation and the Comprehension of Indirect Anaphors in Source Code

Sebastian Lohmeier
Technische Universität Berlin
sl@monochromata.de

25. PPIG, June 26, 2014

Indirect Anaphors in OOP

Psycholinguistic Experiments

Experiment
Design
Hypotheses
Apparatus and Procedure
Materials
Relevance

Discussion

References

Indirect Anaphors in OOP

```
private void foo() {
   new File(".").list();
   System.out.println(.File);
}
```

Indirect Anaphors in OOP

```
private void foo() {
   new File(".").list();
   System.out.println(.String[]);
}
```

Indirect Anaphors in OOP

```
private void foo() {
   new File(".").list();
   System.out.println(.String[]);
}

private void connected(Socket socket) {
   .InputStream.available();
}
```

Indirect Anaphors in OOP

```
private void foo() {
   new File(".").list();
   System.out.println(.String[]);
}

private void connected(Socket socket) {
   .InputStream.available();
}

private void connected(Socket socket) {
   socket.inputStream.available();
}
```
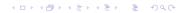
Indirect Anaphors in OOP

```java
private void foo() {
   new File(".").list();
   System.out.println(.String[]);
}

private void connected(Socket socket) {
   .InputStream.available();
}

private void connected(Socket socket) {
   socket.getInputStream().available();
}
```

Psycholinguistic Experiments

- Garrod and Terras (2000): regression- + fixation duration (on word following) direct/indirect anaphors for high-activation instruments (e.g. WRITE + PEN) equivalent, significant difference for low-activation instruments (e.g. WRITE + CHALK)
- Rayner et al. (1995): gaze durations for low-frequency words reduced by 50ms after the 3rd encounter in a text
- O'Reilly and McNamara (2007): readers with high background knowledge but low reading skill answered text-based questions better with low cohesion texts than with high cohesion texts („reverse cohesion effect", maybe caused by skimming i.e. overlooking relevant details)

Experiment: Design

- Programmers – professionals and students – read Java source code with/out anaphors and answer design-level questions afterwards
- 4 groups to balance sequence effects
  1T+2C, 1C+2T, 2T+1C, 2C+1T
- 4 independent vars
  - target condition (T: with vs. C: without anaphors),
  - program comprehension skill (high vs. low score),
  - activation of relation used for indirect anaphors (high or low, manipulated via task sequence within a block),
  - question type (text-based or inference questions in comprehension questionnaire)
- 3 dependent vars:
  - error rate in comprehension questions,
  - regression-path duration for (word following) anaphor
  - task duration

Experiment: Hypotheses

A. Regression-path reading-time on an indirect anaphor or the following word will be shorter, the more active the underspecified relation.

B. If a target relation is highly activated, regression-path reading-times will be equivalent for both types of target expressions.

C. For programmers with low program comprehension skill and for highly activated relations, the error rate for text-based comprehension questions is expected to be lower for the test tasks with indirect anaphors than for the control tasks with local variables and qualified expressions.

D. For highly active relations, duration could be lower for test tasks than for corresponding control tasks.

E. Alternatively, task duration could be higher for test tasks with anaphors than for corresponding control tasks.

Experiment: Apparatus and Procedure

+ Eclipse IDE with disabled editor and navigation via buttons

Experiment: Apparatus and Procedure

- ► Program comprehension skill questionnaire
- ► Introduction to anaphors and anaphors reference
- ► 2x20x code reading tasks with yes-no questions
- ► 20 open-ended comprehension questions
- ► 5 minute summary in bullet points
- ► Post-test questions

Experiment: Materials: PC Skill Questionnaire

How familiar are participants with identifying design-level
relations in (undocumented) source code?

1. Self-rating vs. all other programmers (top 10%, ...)
2. Familiarity with and attitude towards
   2.1 Extending undocumented software
   2.2 Using (undocumented) open source APIs/frameworks
   2.3 Using/knowing design patterns and refactorings
3. Implementation and documentation of 10k LOC projects

Experiment: Materials

```
package net.jini.core.lookup;

public interface Service {
  public Service.ID getServiceID();

  public class ID implements Serializable {
    public long mostSig;
    public long leastSig;

    public ID(long mostSig, long leastSig) {
      this.mostSig = mostSig;
      this.leastSig = leastSig;
    }

    public long getMostSignificantBits() {
      return mostSig;
    }

    public long getLeastSignificantBits() {
      return leastSig;
    }
  }
}
```

Experiment: Materials

Does a Service have an Entry instance?

No, services themselves are not in general associated with entries.

Experiment: Materials

```
package net.jini.space;

public interface JavaSpace extends Service {

  public Lease write(Entry entry, long lease)
    throws RemoteException;

  public Entry take(Entry tmpl, long timeout)
    throws RemoteException;

  public EventRegistration addListener(Entry tmpl,
      RemoteEventListener listener, long lease)
    throws RemoteException;
}
```

Experiment: Materials

Does a JavaSpace have a Service.ID?

Yes, JavaSpace has a Service.ID, like all other services.

Experiment: Materials

12 items later ...

Experiment: Materials

```
public void distributeVouchers(JavaSpace space) {
  try {
    String creator = (String)config.getEntry(SERVER,
        "voucherCreator", String.class);
    String title = (String)config.getEntry(SERVER,
        "voucherTitle", String.class);
    String description = (String)config.getEntry(SERVER,
        "voucherDescription", String.class);
    float value = (Float)config.getEntry(SERVER,
        "voucherValue", Float.class);
    (Location[])config.getEntry(SERVER, "voucherLocations",
        Location[].class);
    for(Location location: .Location[]) {
      UuidFactory.generate();
      new Voucher(.Uuid, creator, title, description,
          value, .Location);
      writeVoucher(.JavaSpace, .Voucher);
    }
  } catch (Throwable t) {
    Log.log(Level.SEVERE, "Failed to distribute vouchers to
        JavaSpace "+.Service.ID, .Throwable);
  }
}
```

Experiment: Materials

Does a Voucher have a Uuid?

Yes, a Voucher has a Uuid.

Experiment: Materials

```
private void writeVoucher ( JavaSpace space , Voucher voucher )
    {
  try {
    long leaseDurationMs = 24*60*60*1000;
    space.write(.Voucher , leaseDurationMs );
    Log.log(Level.INFO , "Added voucher at "+.Location+" to
        JavaSpace "
      +.Service.ID );
  } catch ( RemoteException e) {
    Log.log(Level.SEVERE , "Failed to write voucher to space
        "+.Service.ID , .RemoteException );
  }
}
```

Experiment: Materials

The log entry mentions a Location: was this Location obtained from the JavaSpace?

No, the Location is part of the Voucher written to the JavaSpace.

Experiment: Relevance

A Do activation levels affect reading times of IAs?

B Are there IAs that are not harder to understand than local variables/qualified expressions?

C Can IAs improve comprehension?

D,E How do IAs affect task duration?

- Following B+C: can IDEs show IAs on demand?

Discussion

*

References I

Garrod, S. and Terras, M. (2000). The contribution of lexical and situational knowledge to resolving discourse roles: Bonding and resolution. *Journal of Memory and Language*, 42:526–544.

O'Reilly, T. and McNamara, D. S. (2007). Reversing the reverse cohesion effect: Good texts can be better for strategic, high-knowledge readers. *Discourse Processes*, 43(2):121–152.

Rayner, K., Raney, G. E., and Pollatsek, A. (1995). Eye movements and discourse processing. In Lorch, R. F. and O'Brien, E. J., editors, *Sources of coherence in reading*, pages 9–35. Erlbaum, Hillsdale.