Experimental Results on Anaphors in a Programming Language

Sebastian Lohmeier
Technische Universität Berlin
sl@monochromata.de

56. StuTS, November 21, 2014

Anaphors in English ...

1 Keith was giving a lecture in London.

2a He was **taking his car** there overnight.

3 **The car** had recently been overhauled.

(Garrod and Sanford, 1982)

- Direct anaphors co-refer to a referent of a previously mentioned antecedent.

Anaphors in English ...

1 Keith was giving a lecture in London.

2b He was **driving** there overnight.

3 **The car** had recently been overhauled.

(Garrod and Sanford, 1982)

- Direct anaphors co-refer to a referent of a previously mentioned antecedent.
- An indirect anaphor refers to a referent that is related but not identical to the referent of its previously mentioned anchor.
- The relation (e.g. thematic role, part, ...) is retrieved from long-term memory instead of being expressed in the text.

Anaphors in English ...

1 Keith was giving a lecture in London.

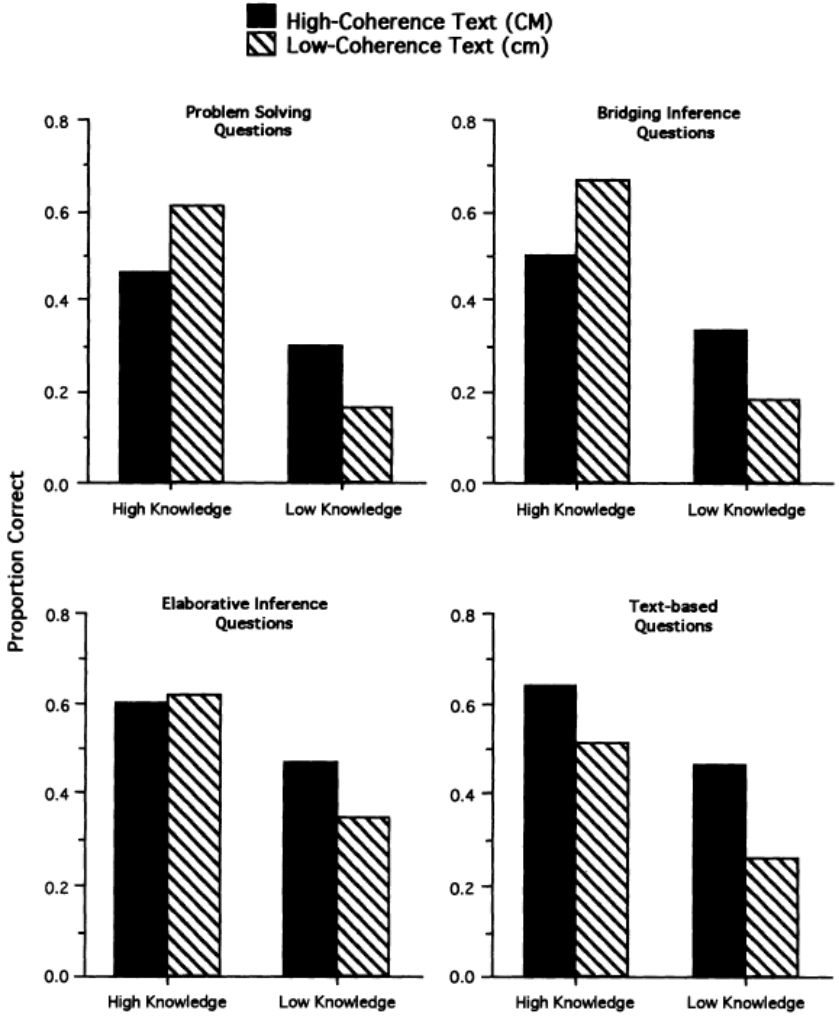
2b He was **driving** there overnight.

3 **The car** had recently been overhauled.

(Garrod and Sanford, 1982)

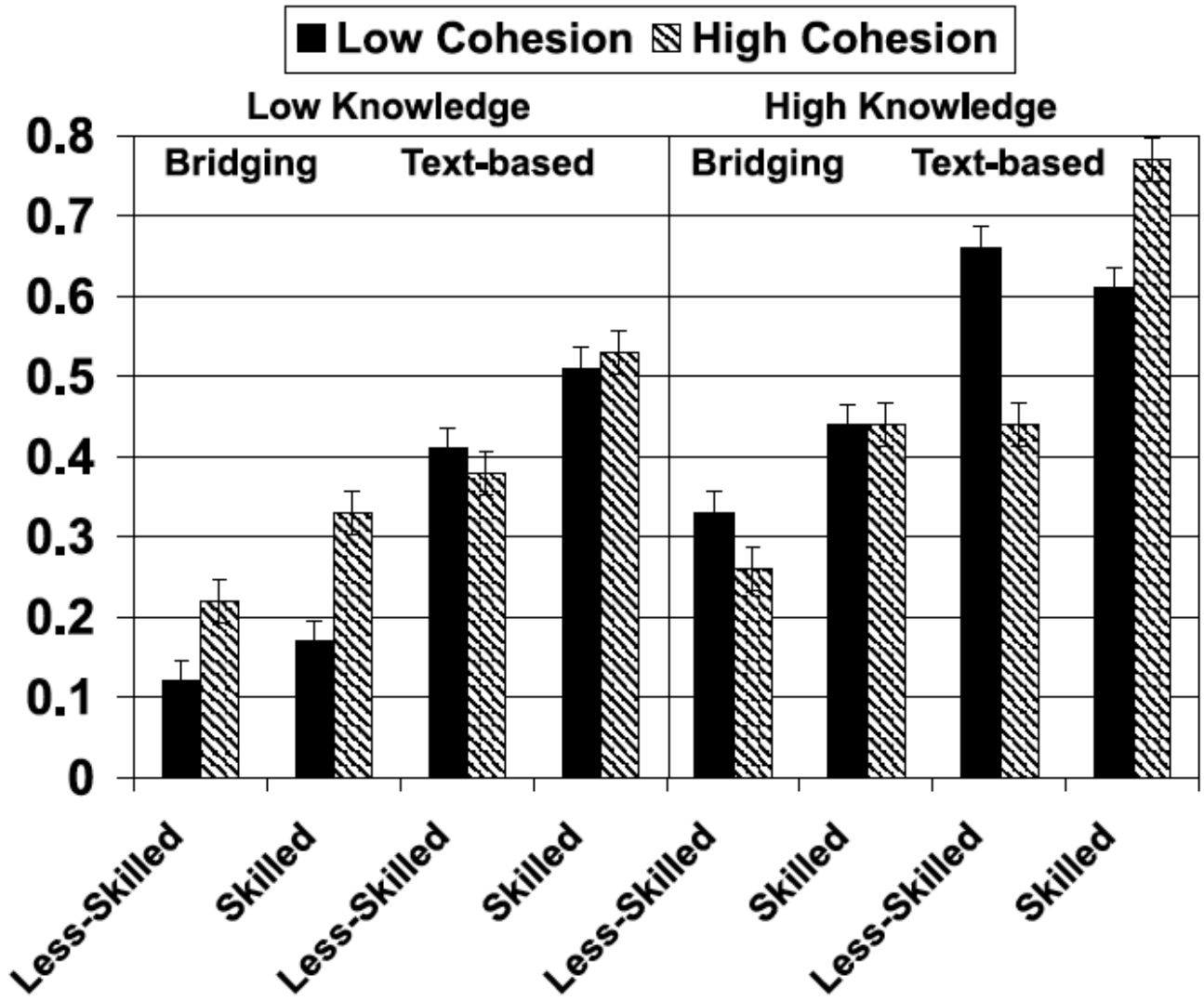
- Garrod and Terras (2000): Regression-path duration equivalent for direct and indirect anaphoric uses of frequently related words
- McNamara et al. (1996): Some readers answer difficult questions better after reading difficult text with low cohesion, cf. indirect anaphors (RCE: reverse cohesion effect)
- O'Reilly and McNamara (2007): RCE does not occur with high reading skill (reversed RCE)

Anaphors in English ...



Reverse cohesion effect in correct answers
(McNamara et al., 1996, 29)

Anaphors in English ...



Reversed reverse-cohesion effect in correct answers
(O'Reilly and McNamara, 2007, 135)

Anaphors in English and in Java

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

Anaphors in English and in Java

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

Anaphors in English and in Java

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

Anaphors in English and in Java

- “Semantic” reference to objects reduces local variables and qualified expressions
- Shortens source code
- Indirect anaphors incomprehensible without background knowledge
- Might improve comprehension with background knowledge
- There are different proposals for anaphors in programming (Knöll et al., 2011; Knöll and Mezini, 2006; Lohmeier, 2011; Lopes et al., 2003).
- Experimental evaluation is still outstanding, yet.

Procedure and Materials

1. Program comprehension skill questionnaire
2. Introduction to anaphors
3. 2x20 tasks: read source code, answer yes-no question
4. 20 comprehension questions
5. 5 minutes to write short summary of the code
6. Post-test questions and de-briefing

Procedure and Materials

Simplified examples of the materials:

```
/**
 * The backend of a JavaSpace implementation
 */
public class OutriggerImpl {

  private static OutriggerImpl Instance;
  private LeaseRenewalManager lrm;

  // ...
```

Q: Does each OutriggerImpl instance have a Lease?
A: No, OutriggerImpl declares no field of type Lease.

Procedure and Materials (Control Condition)

```
public static void main(String[] args) {

  RegistrarLocator locator = new RegistrarLocator();
  Instance = new OutriggerImpl();

  ServiceRegistration registration =
      locator.getRegistrar().register(Instance);
  Log.log("Got service id:
      "+registration.getServiceID());

  Instance.lrm.renewUntil(registration.getLease(),
      FOREVER);
}
```

Q: Is the LeaseRenewalManager part of the ServiceRegistrar?
A: No, the LeaseRenewalManager is part of the OutriggerImpl instance.

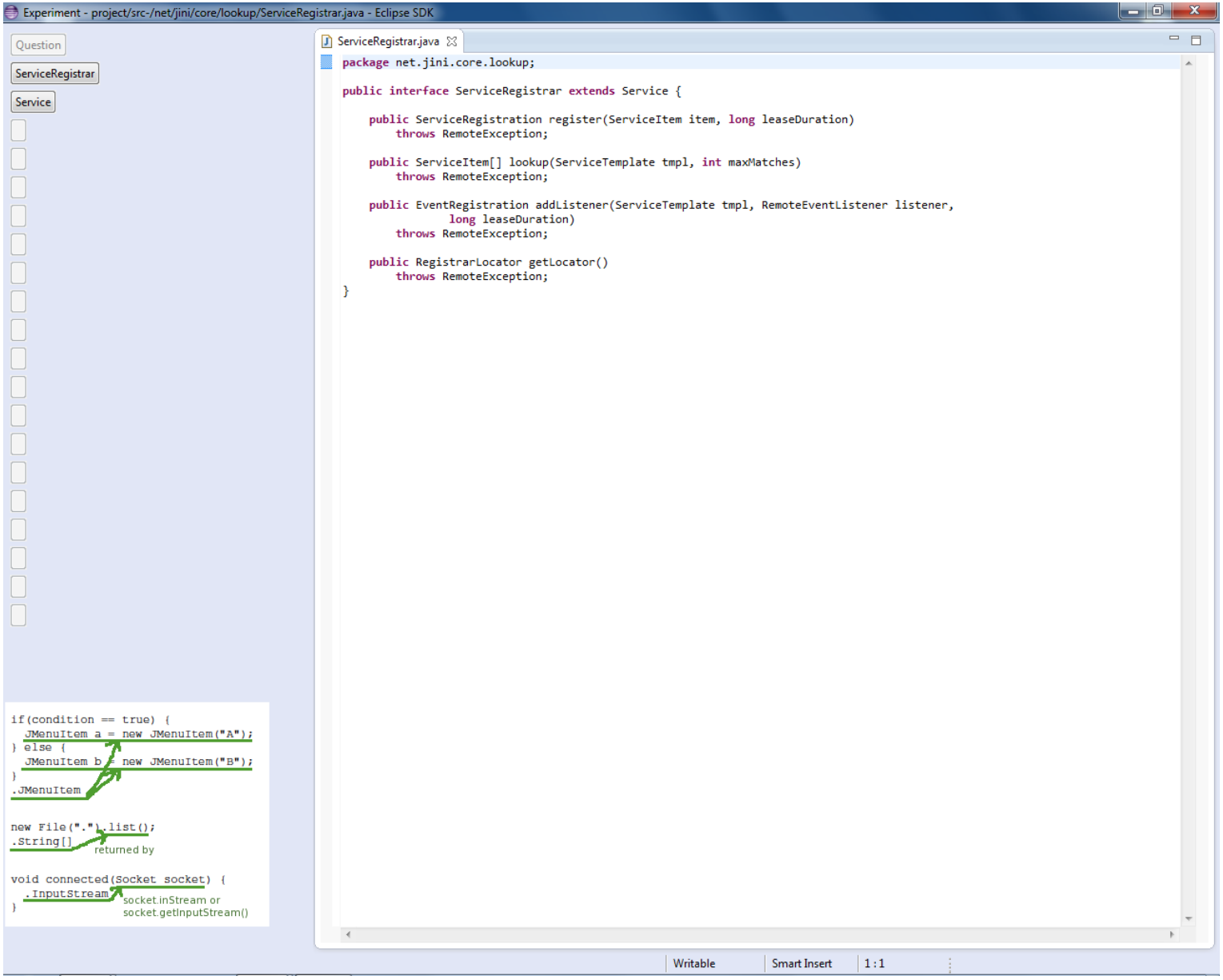
Procedure and Materials (Test Condition)

```
public static void main(String[] args) {

  new RegistrarLocator();
  Instance = new OutriggerImpl();

  .ServiceRegistrar.register(.OutriggerImpl);
  Log.log("Got service id:
      "+.ServiceRegistration.getServiceID());

  .LeaseRenewalManager.renewUntil(.Lease, FOREVER);
}
```

Q: Is the LeaseRenewalManager part of the ServiceRegistrar?
A: No, the LeaseRenewalManager is part of the OutriggerImpl instance.

Apparatus and Participants

Apparatus and Participants



Apparatus and Participants

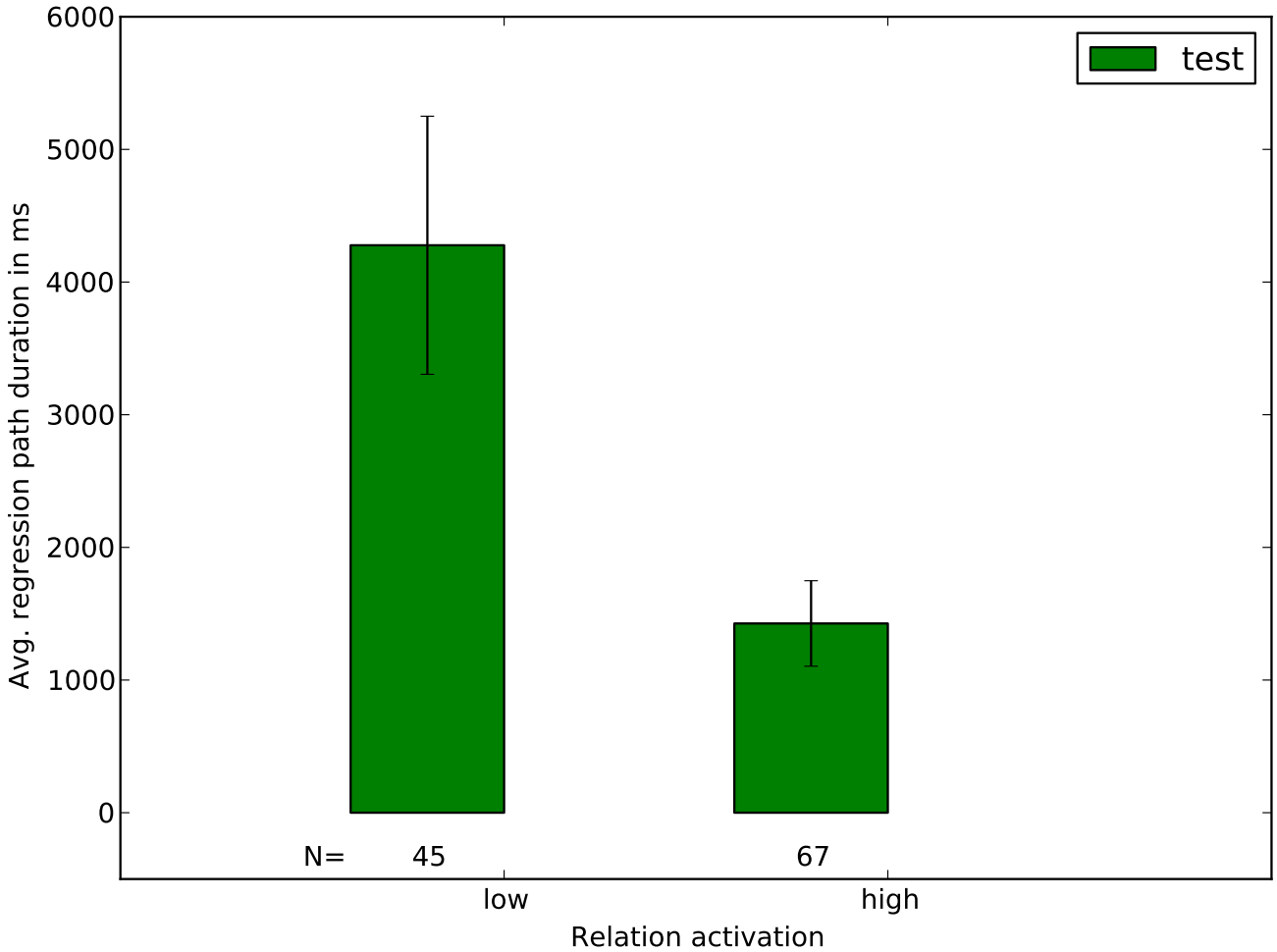
- 19 participants (3 female, 16 male)
- Age 24 to 46
- Students of Computer Science and Human Factors as well as professional programmers
- Vision: normal and corrected
- Native language: 18x German, 1x Dutch

Experimental design

- 4 Groups with different material configurations
  1. T:01-20 + C:21-40
  2. C:01-20 + T:21-40
  3. T:21-40 + C:01-20
  4. C:21-40 + C:01-20
- 4 independent vars
  - 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),
  - question type (text-based or inference questions in comprehension questionnaire)
- 3 dependent vars:
  - error rate in comprehension questions,
  - regression-path duration for target word, i.e. anaphors and replaced expressions
  - task duration

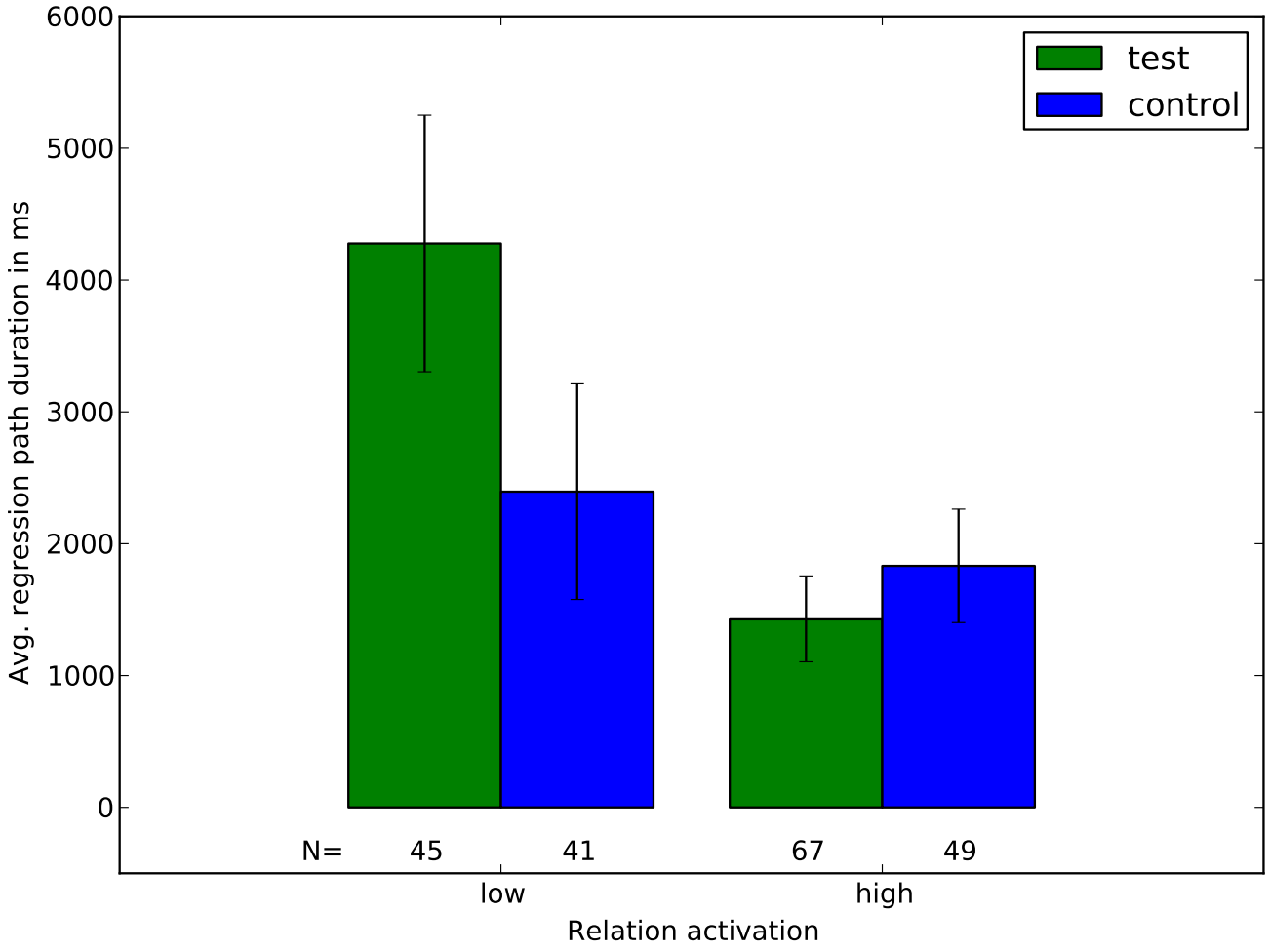
Hypotheses and Results: A

A Regression-path duration on an indirect anaphor will be shorter, the more active – i.e. more recently and frequently presented – the underspecified relation.



Hypotheses and Results: B

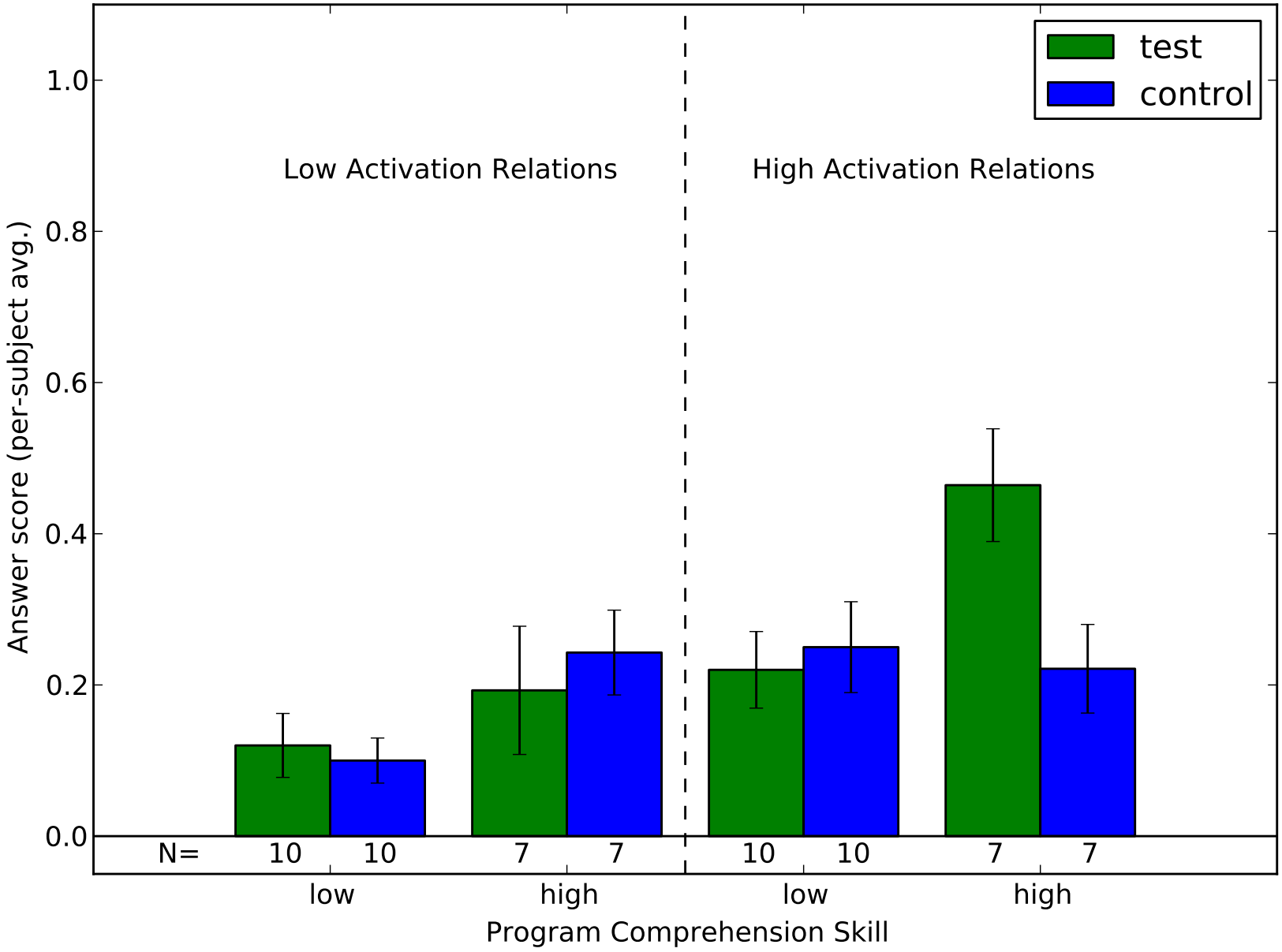
B Regression-path durations on target words in control and test group will be identical for highly activated relations.



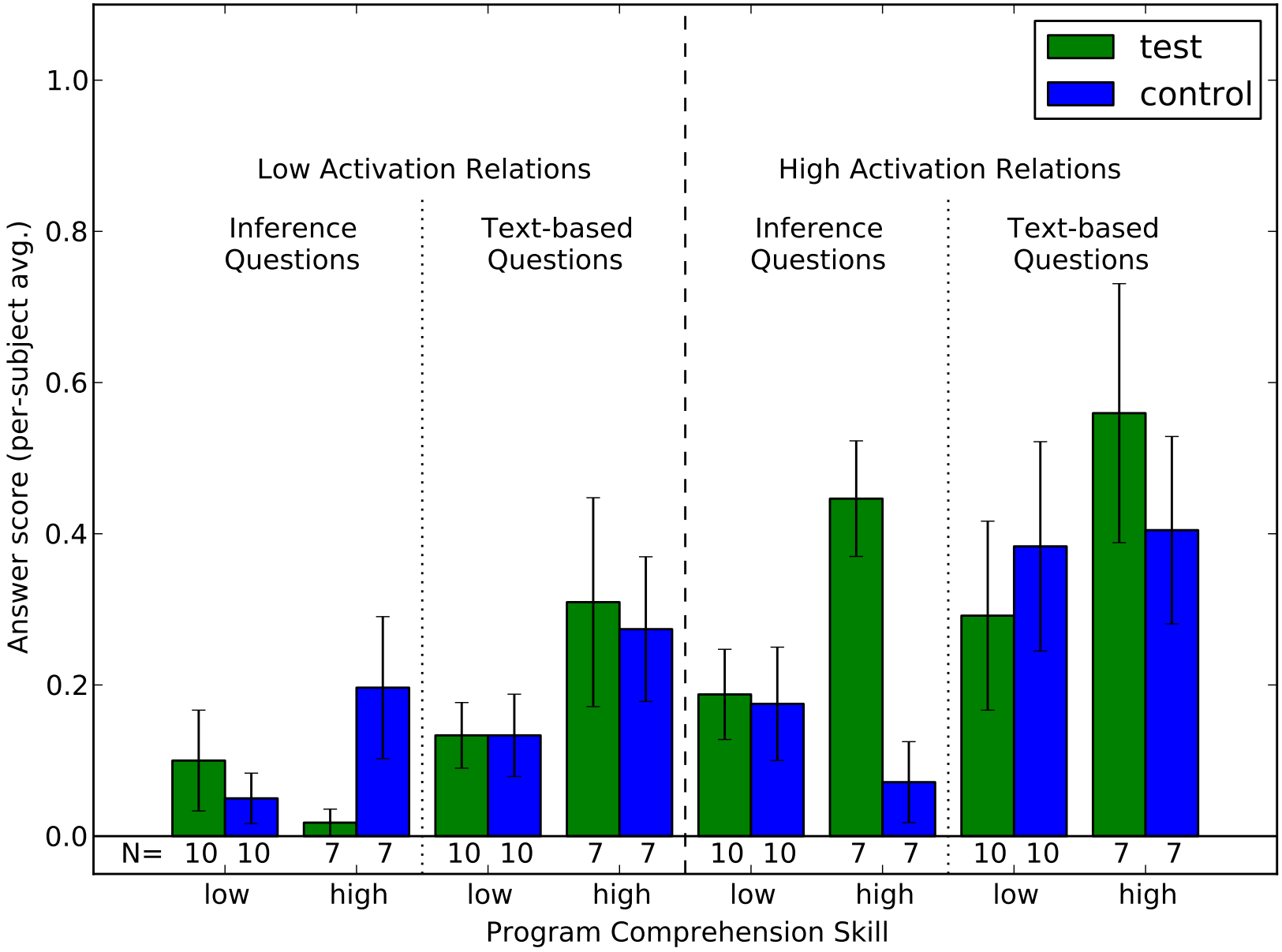
Hypotheses and Results: C

C For highly activated relations, less-skilled programmers will make fewer errors in text-based comprehension question for the test condition with (indirect) anaphors than for the control condition without anaphors.

Hypotheses and Results: C



Hypotheses and Results: C



Hypotheses and Results: C

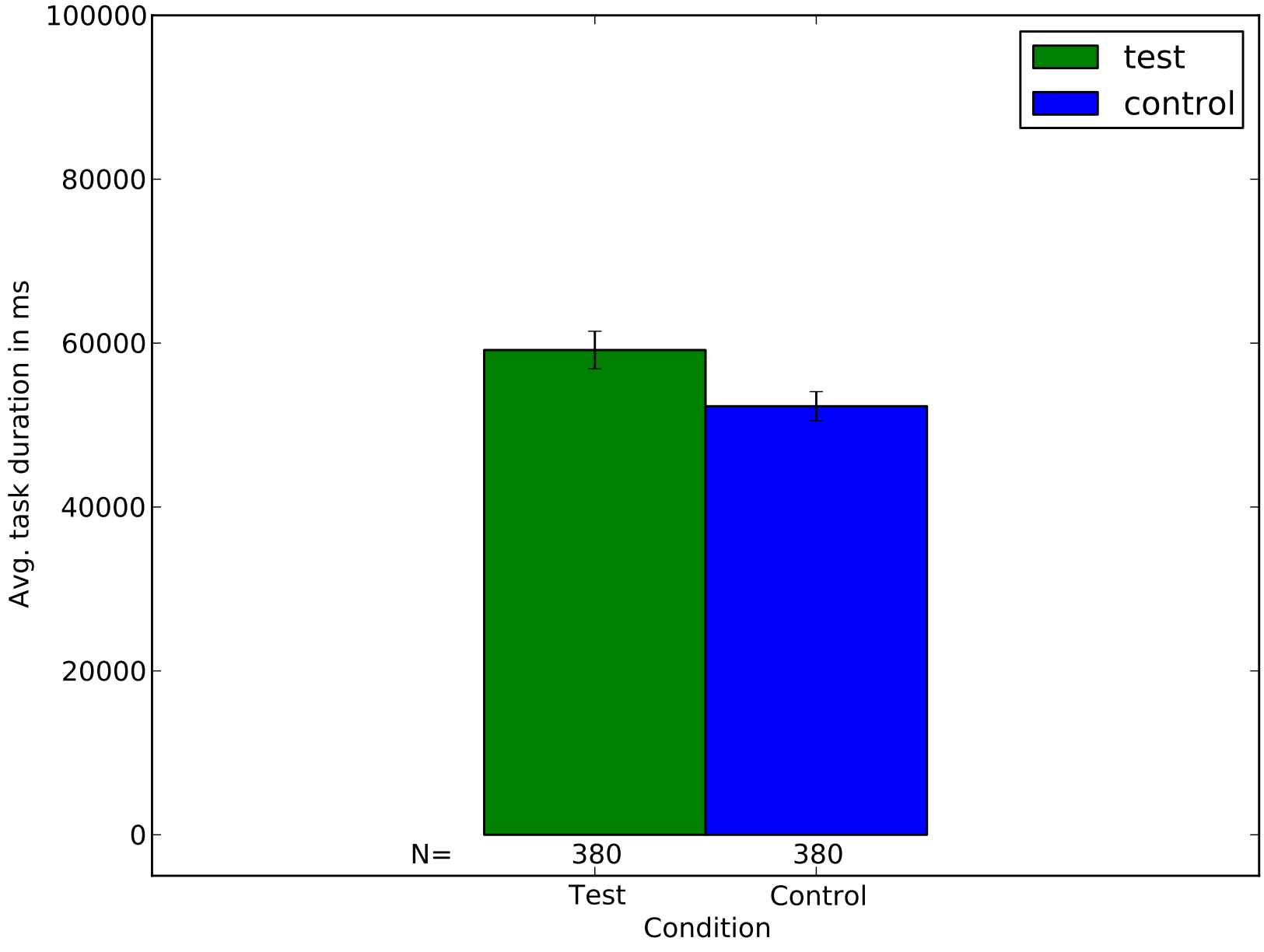
C' For highly activated relations, *high-skill* programmers made fewer errors in *inference-based* comprehension question for the test condition with (indirect) anaphors than for the control condition without anaphors.

Hypotheses and Results: D1 and D2

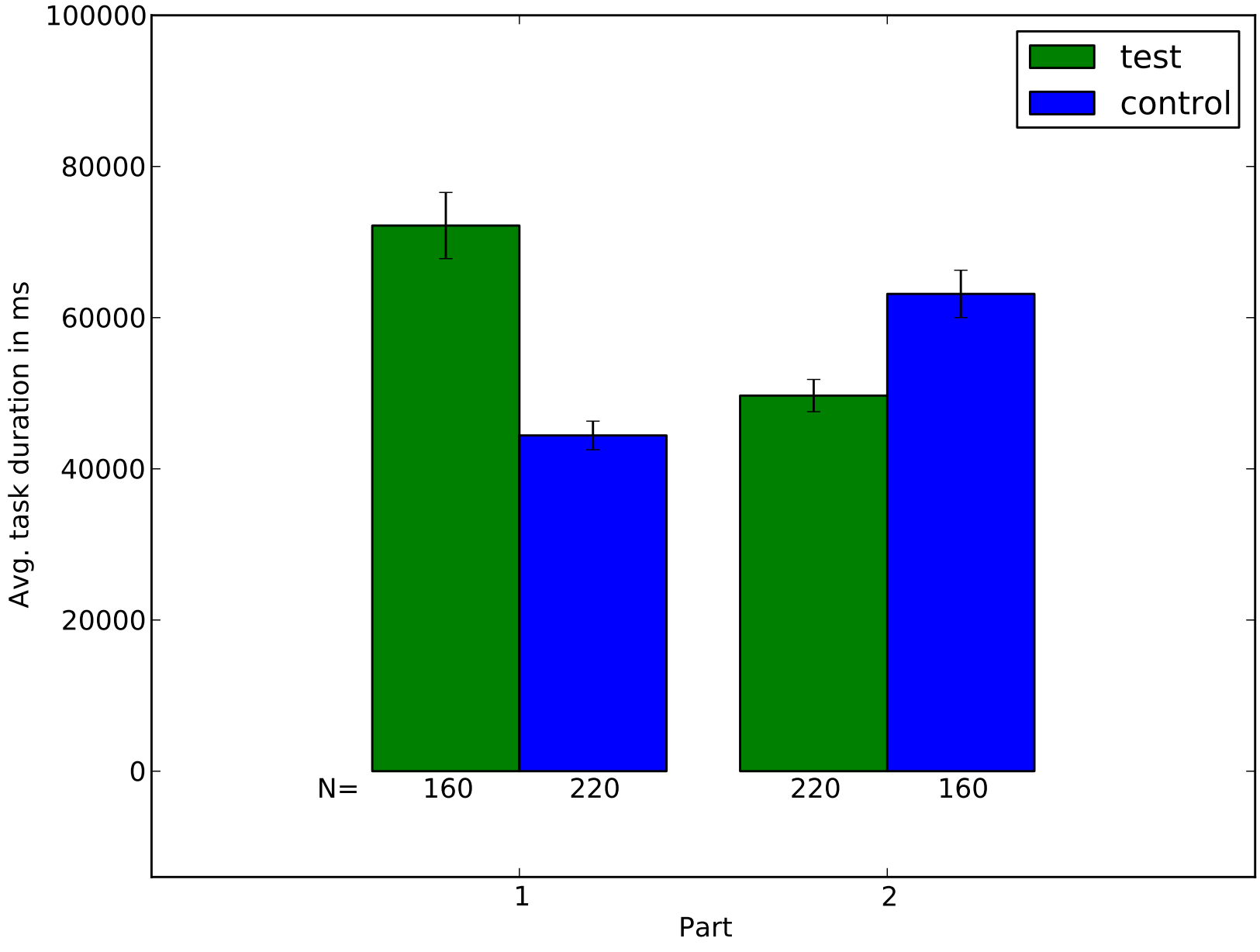
D1 At least for tasks with highly active relations, task duration could be lower for the test group with indirect anaphors than for the control group without them because under-specification reduces the amount of text to be read and indirect anaphors for dominant relations might not gazed at longer than direct anaphors.

D2 Alternatively, task duration could be higher for the test group than for the control group, if indirect anaphors are generally harder to understand than local variables.

Hypotheses and Results: D1 and D2



Hypotheses and Results: D1 and D2



Hypotheses and Results: Summary

A Regression-path duration on an indirect anaphor will be shorter, the more active – i.e. more recently and frequently presented – the underspecified relation.

B Regression-path durations on target words in control and test group could be identical for highly activated relations.

C' For highly activated relations, *high-skill* programmers made fewer errors in *inference-based* comprehension question for the test condition with (indirect) anaphors than for the control condition without anaphors.

Ds Effect on task duration not yet clear.

Discussion

- Eye tracking measures need re-calculation with error correction
- If the results still hold afterwards: great result
- Task duration might explain improved comprehension question performance (to check)
- Do short regression path durations reflect comprehension?

Discussion

- Yes-no-questions distracted from later comprehension tasks
- IDE support (highlighting and go-to-declaration) were requested and might improve program comprehension with anaphors

- Good basis for cognitive model to model fixation duration on indirect anaphors and task durations for feed-in eye tracking data
- If more complex source code is beneficial in some cases, source code could be personalised

Discussion

Q&A

References I

Garrod, S. and Sanford, A. J. (1982). Bridging inferences and the extended domain of reference. In Baddeley, A. and Long, J., editors, *Attention and Performance*, volume XI, pages 331–346. Erlbaum, Hillsdale, NJ.

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.

Knöll, R., Gasiunas, V., and Mezini, M. (2011). Naturalistic types. In *ONWARD '11: Proceedings of the 10th SIGPLAN symposium on New ideas, new paradigms, and reflections on programming and software*, pages 33–47.

Knöll, R. and Mezini, M. (2006). Pegasus: first steps toward a naturalistic programming language. In *Companion to the 21st ACM SIGPLAN symposium on Object-Oriented Programming, Systems, Languages & Applications (OOPSLA '06)*, pages 542–559, New York. ACM.

Lohmeier, S. (2011). Continuing to shape statically-resolved indirect anaphora for naturalistic programming.

Lopes, C. V., Dourish, P., Lorenz, D. H., and Lieberherr, K. (2003). Beyond AOP: toward naturalistic programming. *SIGPLAN Notices*, 38(12):34–43.

References II

McNamara, D. S., Kintsch, E., Butler-Songer, N., and Kintsch, W. (1996). Are good texts always better? Interactions of text coherence, background knowledge and levels of understanding in learning from text. *Cognition and Instruction*, 14(1):1–43.

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.