The End-to-End Use of Source Code Example: An Exploratory Study

Reid Holmes, Rylan Cottrell, Robert J. Walker, Jörg Denzinger

University of Washington, University of Calgary
End-to-End Reuse Process

Example Repository

Locate

Example Source Code

Select

Use example

Integrate

Developer’s Context
End-to-End Reuse Process

[Mcilroy 68]
[Fischer, et al. 91]
[Dusink & Katwijk 95]
[Frakes & Fox 96]
Example
Example

Interfaces

First 10 results return:
4 Interfaces.
5 Implementations.
   2 were specific to a particular library.
   3 had dependencies on other classes within its system.
1 Test (dependent on specific system).

Comments

Only 21,690 more to go!
Example

- 3 out of 10 were possibly relevant.
- Worst case, developer could have to reuse roughly 1882 lines of code.
End-to-End Reuse Process

Example Repository

Locate

Example Source Code

Select

Use example

Integrate

Developer’s Context
No End-to-End Tools Exist

- Locate
- Example Repository
- Example Source Code
- Select
- Integrate
- Developer’s Context
- Google Code Search, XFinder, XSnippet, Prospector
- PARSEWeb, Strathcona
- Jigsaw
Research Question

• Is pipelining existing tools enough?
Case Studies

• Four scenarios were chosen from the literature (PARSEWeb, Strathcona, XFinder, and XSnippet).
• Located examples using Strathcona and Google Code Search.
• Analyzed the first 5 returned results [Starke & Sillito 09].
• Integrated potentially relevant examples, manually and with the Jigsaw tool.
Lessons Learned

• Location
  – Results were ordered with respect to the query and not the end-to-end task.
  – Query does not capture the developer’s reuse context.

• Selection
  – Determining example relevant is difficult.
Lessons Learned Cont’d

• Integration
  – Manual. Identifying the dependencies the code example might have is non-trivial.
  – Jigsaw. Configuration time often took more time than the actual integration.
Moving Forward

1. Repositories become out of date.
   **Solution:** Maintainers need to not only keep source code up-to-date, but also remove out-of-date source code.

2. Queries need to capture the developer’s context.
   **Solution:** Syntax and semantics should be considered.

3. Example ordering is often ineffective.
   **Solution:** Ordered relative to their utility to the developer.
Moving Forward Cont’d

4. Attempting multiple integrations is tedious.
   **Solution:** Integration phase should be better supported.

5. Integration task are complex.
   **Solution:** Developer secondary goals should be considered.
Summary

• End-to-end reuse approach is not supported by any single tool.

• Current approaches using a collection of tools can inhibit the likelihood of successfully utilizing a source code example.