Semi-Automatic Garbage Collection for Mobile Networks

Elisa Gonzalez Boix  Tom Van Cutsem  Stijn Mostinckx
Jessie Dedecker  Wolfgang De Meuter  Theo D’Hondt
Programming Technology Laboratory - Vrije Universiteit Brussel - Belgium

Motivation: DGC in Mobile Networks

- I provide a payment service
- I provide printing services
- A request for printing
- A request for printing
- A request for printing

Volatile connections

- Device a
- Device b
- Device c

Context-dependent information

- Limited connectivity of the nodes.
- Inaccessible reference ≠ Broken reference.
- Semantics of the application are required to clear objects.
- Do not delete this reference, I’ll come back later.
- Do not need this reference at the end of the conference.

Read more?


Problem Statement

How long should the system wait for the connectivity of the devices?

- Application-dependent!
- Even context-dependent!

Who is responsible for garbage collection?

- Developer?
- System?

A combination of both: Semi-Automatic Garbage Collection

- Remote-references as a two-party contract.
- Referencing strategies express collection policies (remote object) + kind of rebinding (client object) upon disconnection.

Referencing Strategies

- Language support to apply a collection strategy to the remote reference.
- Express the disposability of a reference upon disconnection:
  - Temporal disposability
    - ...expressing disposability based on time constraints.
      - Weak References: always reclaim.
      - Temporal References: reclaim after time period.
      - Strong References: never reclaim.
  - Domain-specific disposability
    - ...expressing disposability based on context information.
  - ...some challenges
    - Resolving conflicts.
    - Indirect References.
    - Language support to annotate groups of remote references.

http://prog.vub.ac.be/amop