Behavior protocols – experience with SPEEDO

Viliam Holub

DISTRIBUTED SYSTEMS RESEARCH GROUP
http://nenya.ms.mff.cuni.cz

CHARLES UNIVERSITY PRAGUE
Faculty of Mathematics and Physics

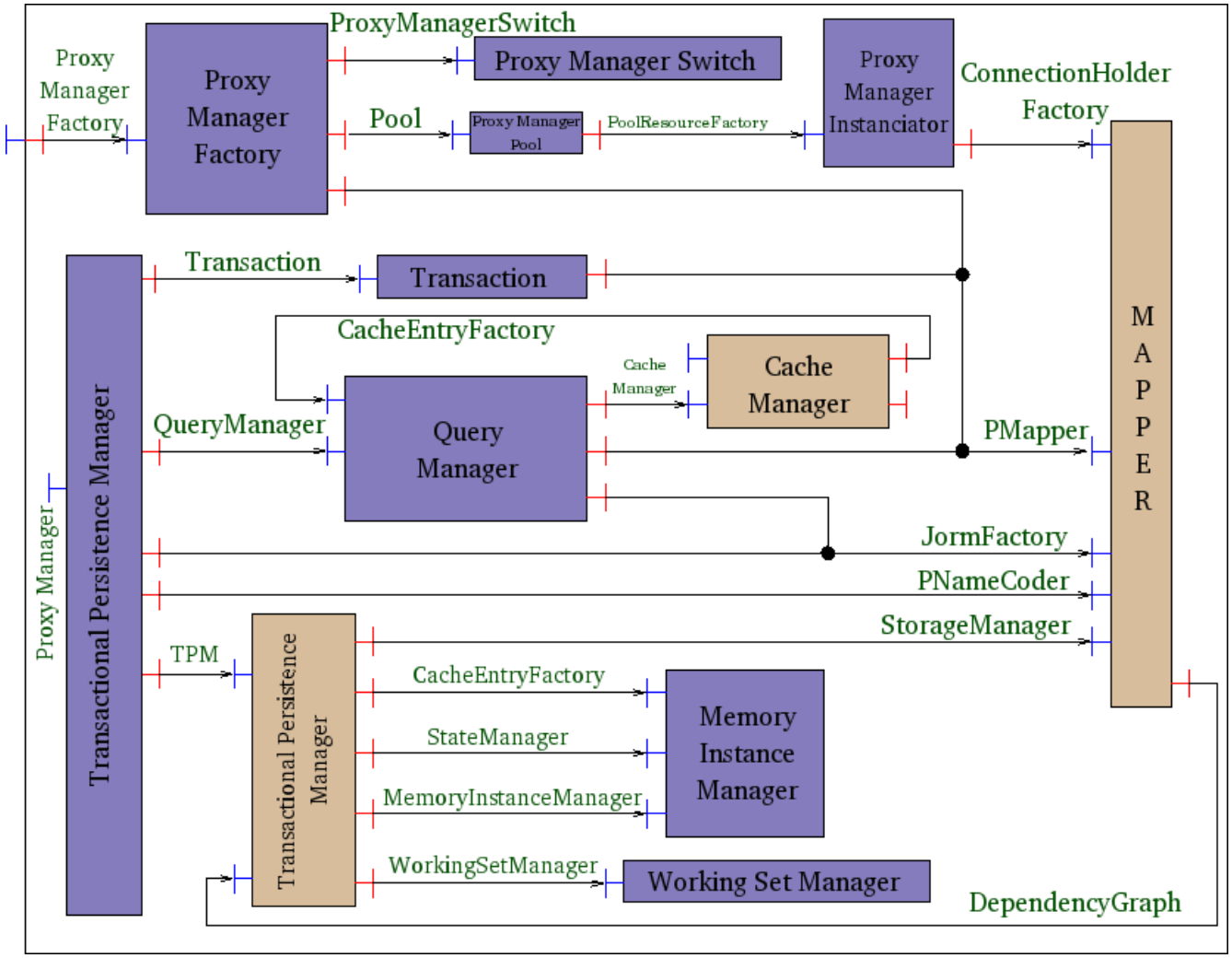
Outline

- SPEEDO overview
  - Used as a case study to asses viability of behavior protocols
- Behavior extraction – pros&cons
- Proposed improvements
  - Preprocessor
  - Synchronization
  - Parameterized protocols

SPEEDO overview I

- On of the ObjectWeb projects
- Using Fractal component model
  - Libraries – Perseus Cache, Persistency, Concurrency, Dependency, Pool
- 43000 lines of core JAVA sources
  - Speede core functionality
- >97000 lines of JAVA sources
  - Core + test suites
- >125000 lines total
- Protocols constructed
  - for those 43000 lines of core
  - 2 month

SPEEDO overview II



Extracting BP – pros

- Straightforward
- Fast

```
?tpm_tpm.commitWS {
  (
    NULL
    +
    (
      !tpm_tpm.rollbackWS;
      [EXIT]
    )
  );
  (
    !tpm_cm_cm.unfix;
    (
      NULL
      +
      (
        !tpm_cm_cm.finalize;
        !mapper.endWS;
        [EXIT]
      );
    );
    !mim.isUnexported;
    !mim.makeUnbound
    +
    (
      (
        !mim.makeClean;
        !mim.setReferenceState
      )
      +
      (
        !mim.getReferenceState
        (
          !mim.setReferenceState
          +
          !mim.destroyState
        )
```

Extracting BP – cons

- Prone to bugs
  - Specially true for large components
  - automated extractor?
- Internal methods have to be expanded
  - when calling a requires interface
  - preprocessor?
- Nested interface calls modeled as parallel calls
  - Even though is a nested visit of a single thread
  - parameterized protocols?
- Multithread code is not fully supported
  - Modeled as multiple instances of an interface
  - lock, synchronized interface?
- Exceptions
  - Not supported

Proposed solutions

- *(i) Automated extraction*
  - Challenge: design a tool
    - User advises needed
    - No desire to produce a final result
- *(ii) Preprocessor*
  - syntax slightly extended (macro idea)

```
Name1 < !i2.a; !i3.b; !i2.c; > // macro def


?i1.a { Name1; !i2.a;  Name2 < !i3.a; !i3.b; > }
+
?i1.b { !i3.c; Name2; }
```

Proposed solutions

- *(iii) Lock, synchronized interface*
  - Independent component
  - Automatically generated
  - Parameterized iterative algorithms

Proposed solutions

- *(iv) Parameterized protocols, exceptions*
  - Still an open issue

The end