

Backfiring COSMIC size from Java and C++ code

Malcolm S Jenner

(M.S.Jenner@blueyonder.co.uk)

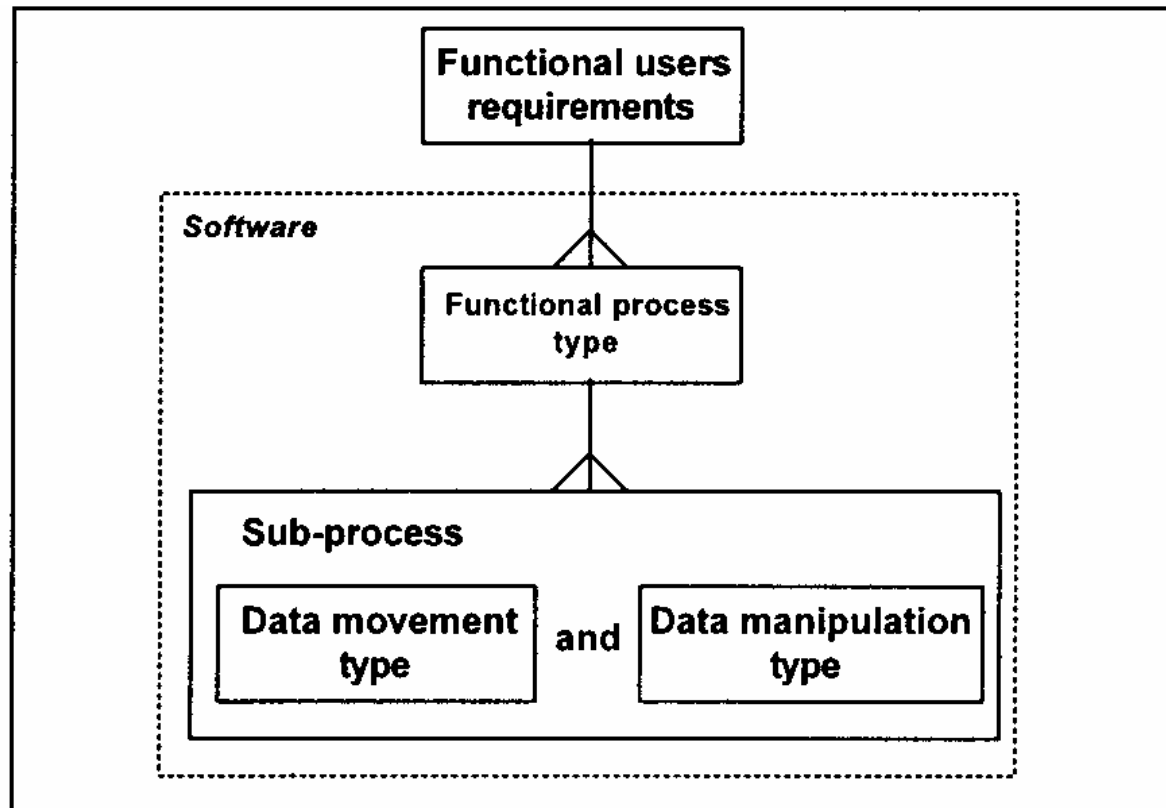
COSMIC from Code

1. UML and COSMIC size
2. UML to code
3. Code to COSMIC size

1. UML and COSMIC size

COSMIC FFP 2.0

- Structural model



Comparison of Concepts (from Bévo et al)

COSMIC FFP 2.0

UML

Data group

Class

Data attribute

Object attribute

Functional process

Use Case (or scenario)

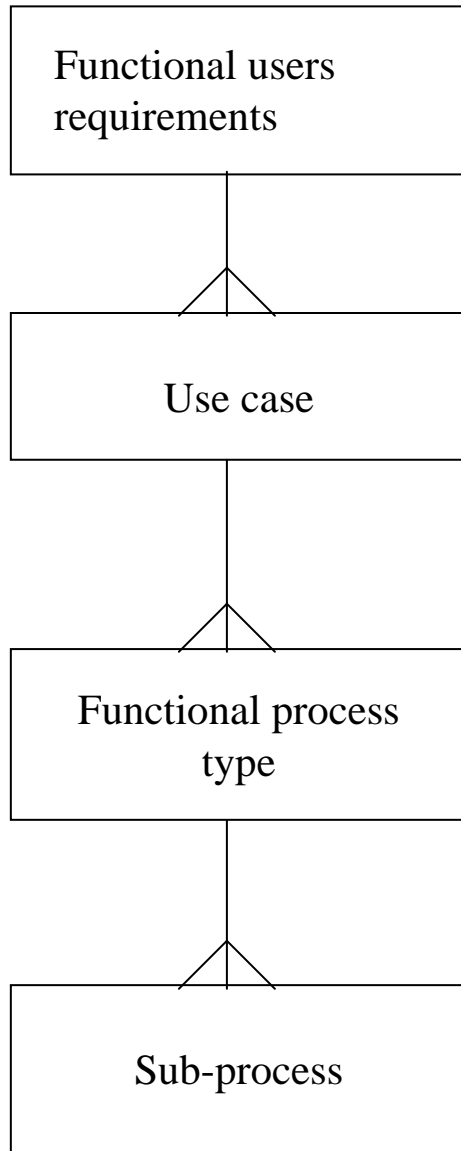
Use Cases

- What is a use case?
- Different partitioning
- Any complete set should give total specification

Sequence Diagrams

- Sequence diagram splits use case into component activities – each representing an interaction with the actor
- Activities correspond to functional processes
- Modified COSMIC structural model for use with UML:

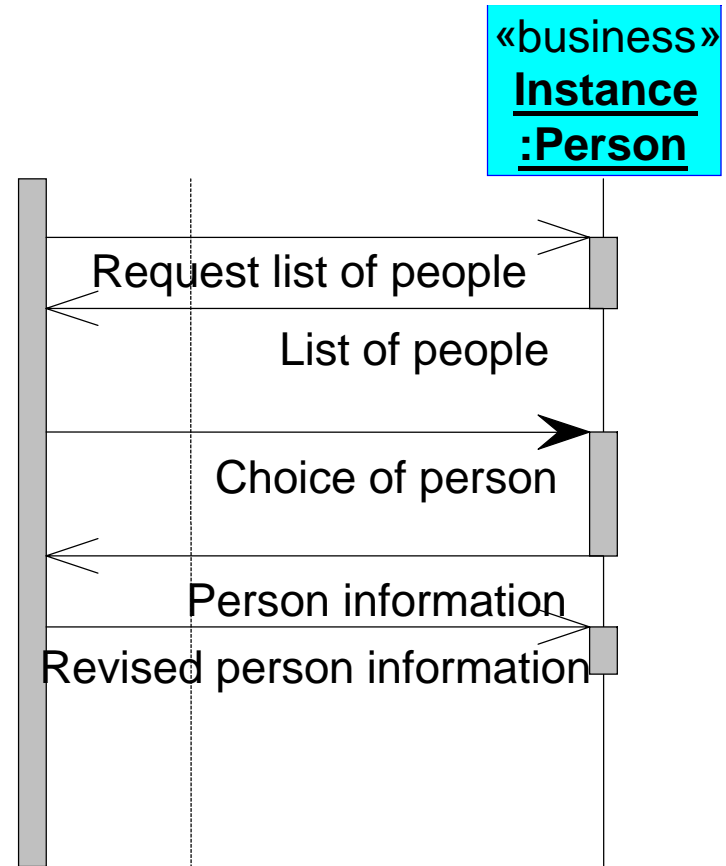
Modify person use case



Request list of people
Receive list of people

Choice of person

Receive details
Submit revised information



- Three functional processes
- Conceptually simpler as one use case
- Cfsus
 - 1E + 1R + 1X
 - 1E + 1R + 1X
 - 1E + 1W
- Total = 8

Interfaces

- Easier to count with interface object
- Omit optional return arrows for read operations
- Then number of arrows equals size in Cfsu
- E.g. Library example from Eriksson and Penker – “Return Item” use case

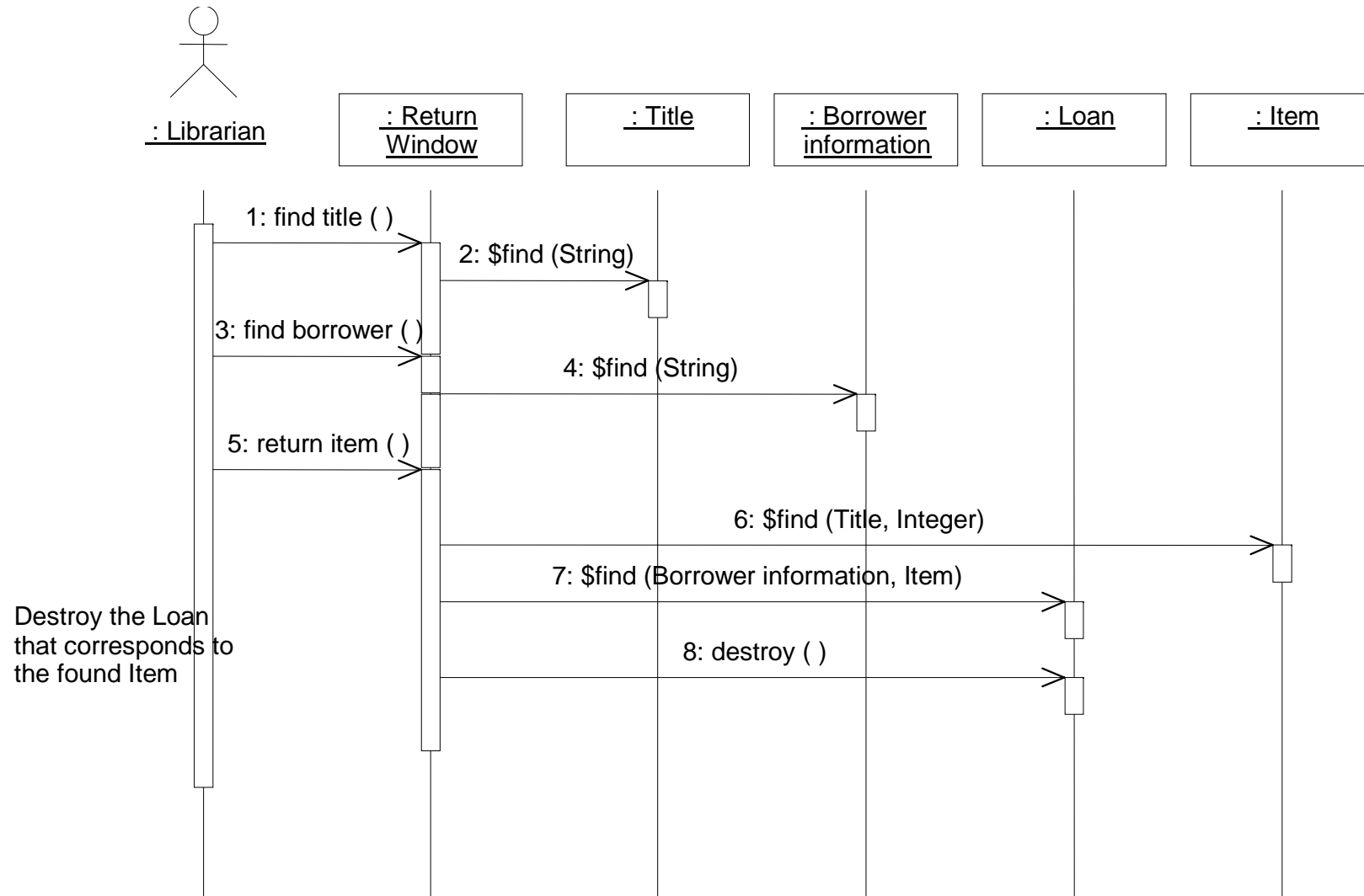


Figure 5

Return item

- Three actions – one functional process
 - $1E + 1R$
 - $1E + 1R$
 - $1E + 2R + 1W$

Automation

- Requires sequence diagrams to only have required information
 - Interface object
 - No returns from reads
- Then count arrows
- Depends on storage mechanism

Rational Rose

- Model stored as text file
- Locate start and finish of each Object InteractionDiagram
- Arrows identified by “origin” and “terminus”


```
(object InterMessView "" @87
  location      (0, 1116)
  label         (object SegLabel @88
    Parent_View @87
    location    (1271, 1072)
    quidu       "335CBD240032"
    anchor_loc  1
    nlines      1
    max_width   215
    justify     0
    label       "destroy ( )"
    pctDist     0.500000
    height      45
    orientation 0)
  client        @75
  supplier      @80
  Focus_Src     @79
  Focus_Entry   @82
  origin        (760, 1116)
  terminus      (1782, 1116)
  ordinal       7)
```

2. UML to code

Code operations

- Create object
- Modify attribute(s)
 - Enter data
 - Store data
- Retrieve attribute(s)
 - Obtain data
 - Return data
- Destroy object

COSMIC data movements

- Create business object
 - Write
- Modify attribute(s)
 - Write
- Retrieve attributes
 - Read
- Destroy object
 - Write

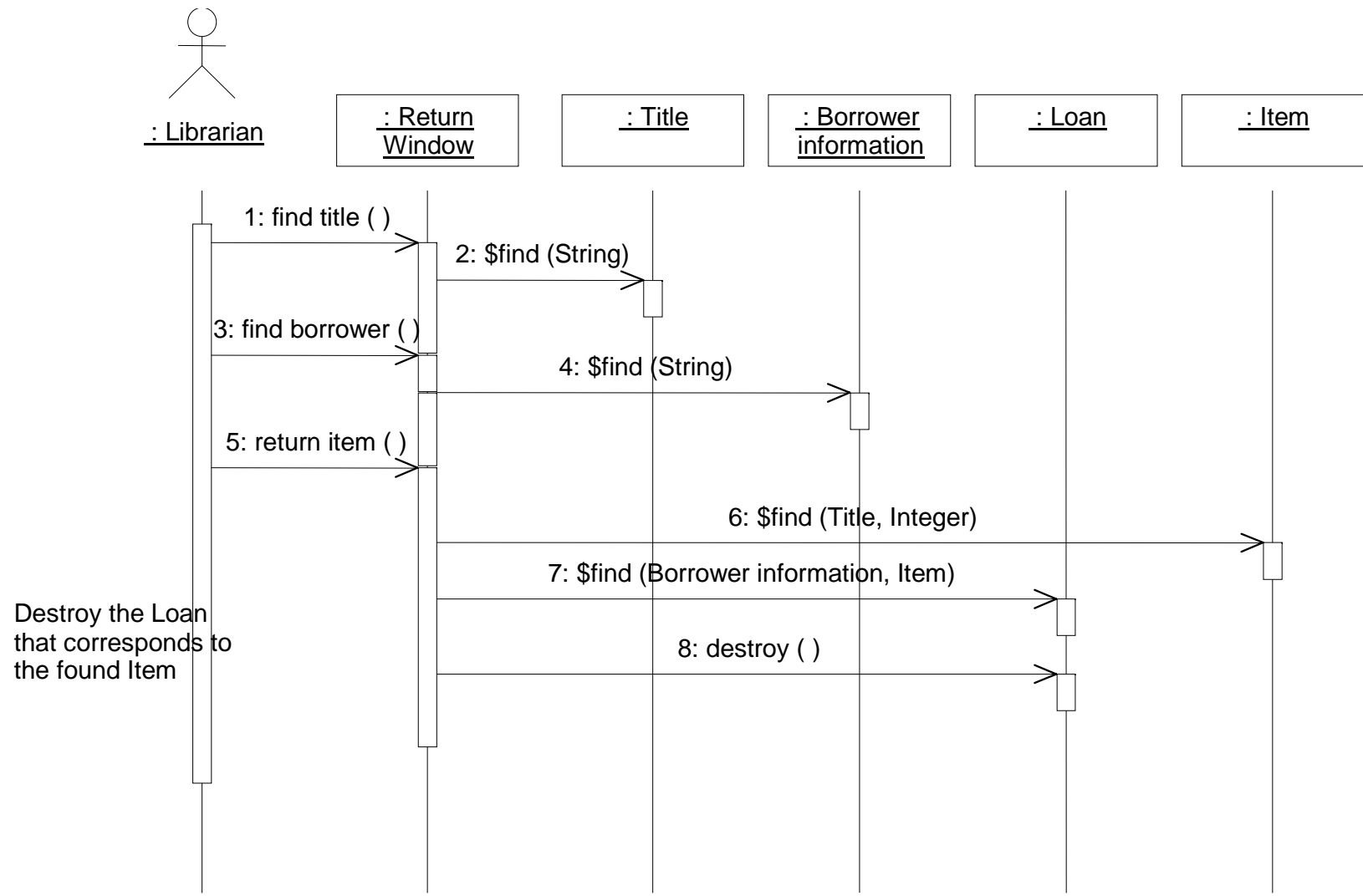


Figure 5

Interface object

- These provide for Entry and Exit of data
- Need to be created at start of Use Case
 - “Trigger” proposed by Azzouz & Abrain (2004) as addition to COSMIC data movements
 - ? Special case of Entry/Write

3. Code to COSMIC size

- Count:
 - All creations and destructions of objects
 - All calls to operations
 - All assignments to variable

- Problems:
 - May count implementation details as well as functional requirements
 - Possibly covers some data manipulation as well as data movements
- Are these problems?

References

- Assouz, S & Abrain, A, “A Proposed Measurement Role in the Rational Unified Process and its Implementation with ISO 19761: COSMIC-FFP”, SMEF-2004, Rome, Italy, 2004
(<http://www.lrgl.uqam.ca/publications.html>)

References (c't'd)

- Jenner, M S, “COSMIC FFP 2.0 AND UML: ESTIMATION OF THE SIZE OF A SYSTEM SPECIFIED IN UML - PROBLEMS OF GRANULARITY”, Proceedings, 4th European Conference on Software Measurement and ICT Control, Heidelberg, May 2001 pp 173-184

References (c't'd)

- Jenner, M S, “AUTOMATION OF COUNTING OF FUNCTIONAL SIZE USING COSMIC FFP IN UML”, (Proceedings, 12th International Workshop on Software Measurement, Magdeburg, October 2002)