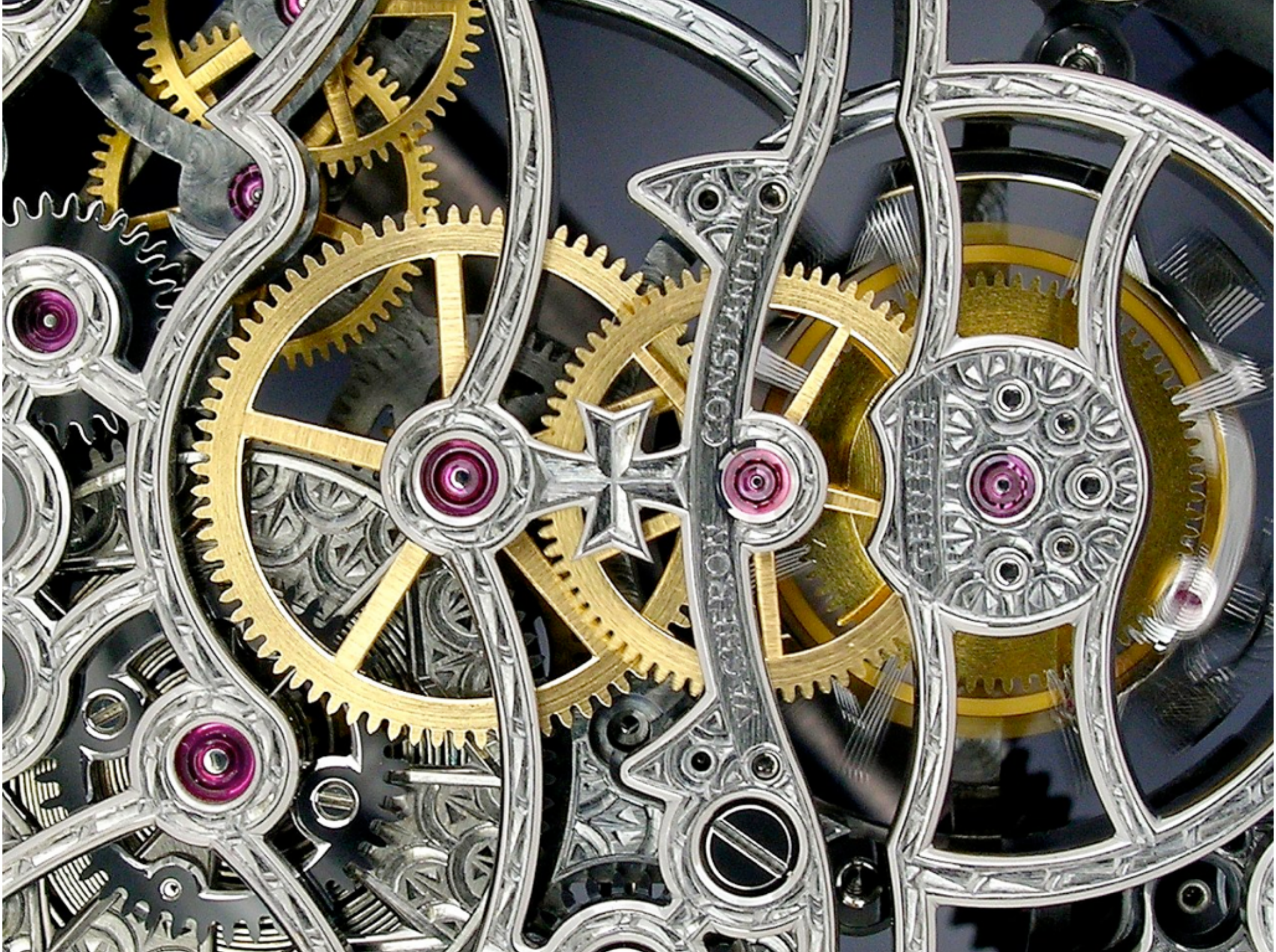
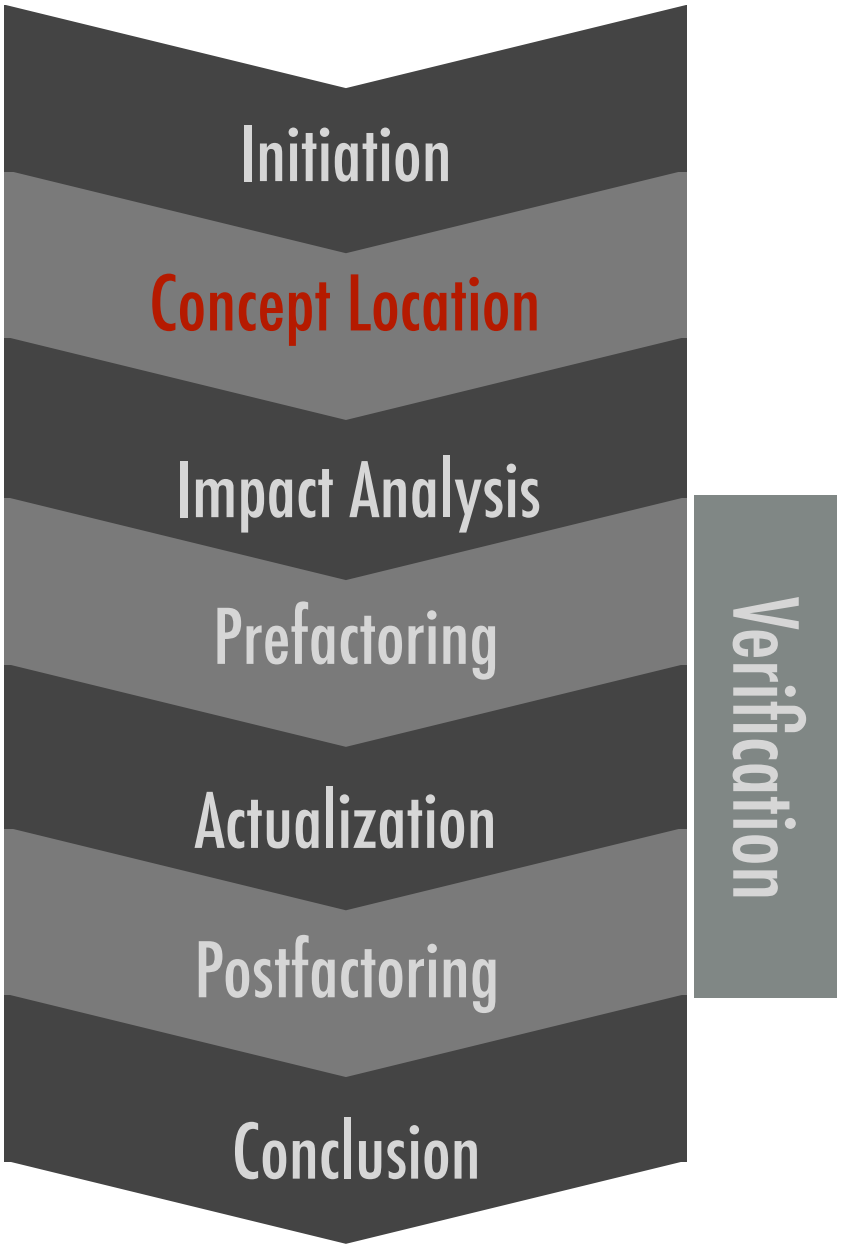
Concepts and Concept Location

Prof. Romain Robbes


CONSTANTIN
GENEVE

Concept location finds a starting point



The programmer looks for the module(s) that needs to be changed to implement the change request.

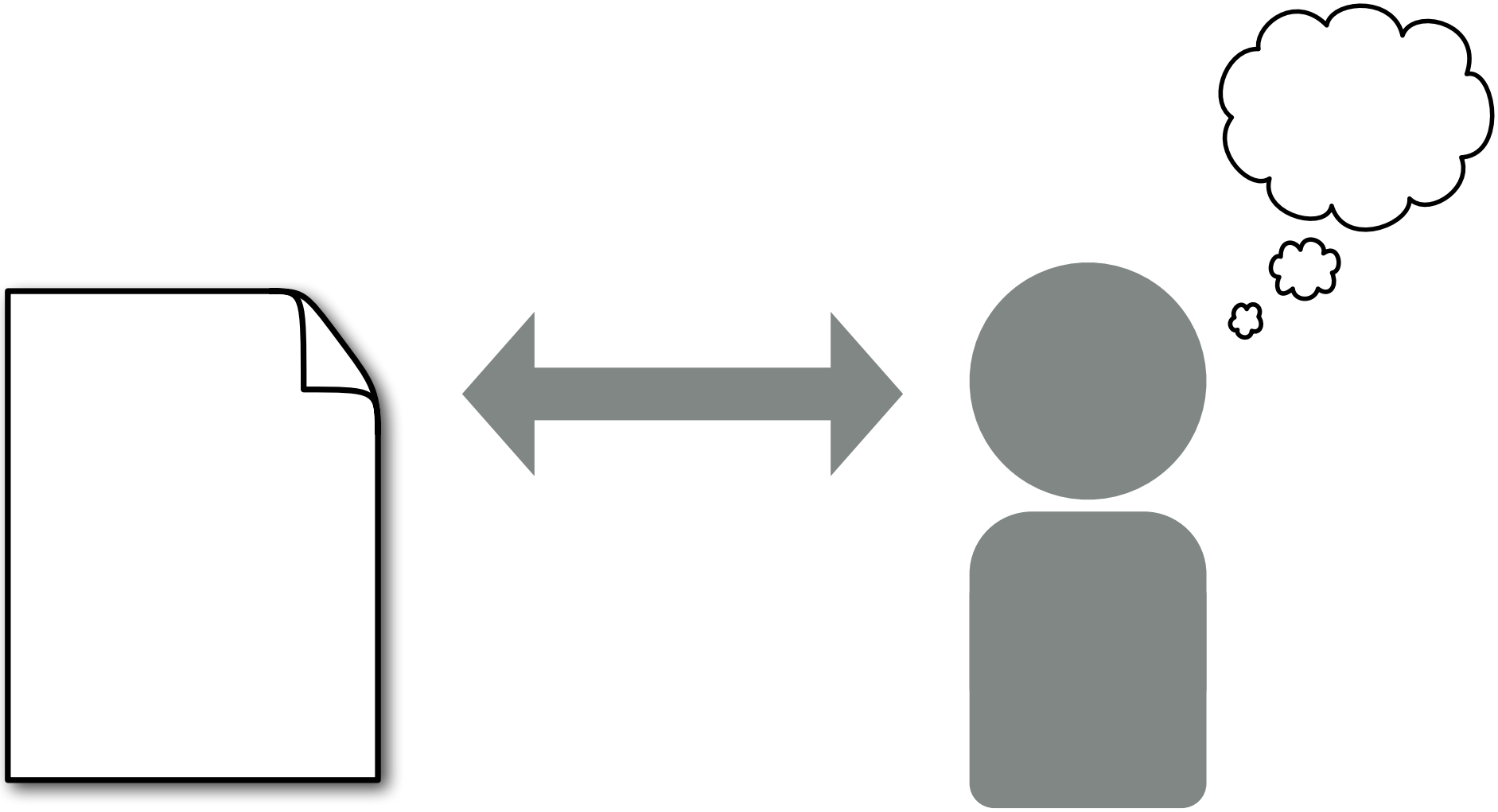
A starting point, to be refined via impact analysis.

Outline

What are concepts
Concept location
Extracting significant concepts
Concept location via Grep
Examples
Concept location via Dependency Search
Examples

Concepts

Concepts are the bridge between the program and the stakeholders' vision of it



The role of concept location

Concept location is needed whenever a change is to be made.

Change requests are most often formulated in terms of domain concepts:

- Ex: “Correct error that arises when trying to paste a text”
- The programmer must find in the code the locations where the concept “paste” is located

This is the start of the change

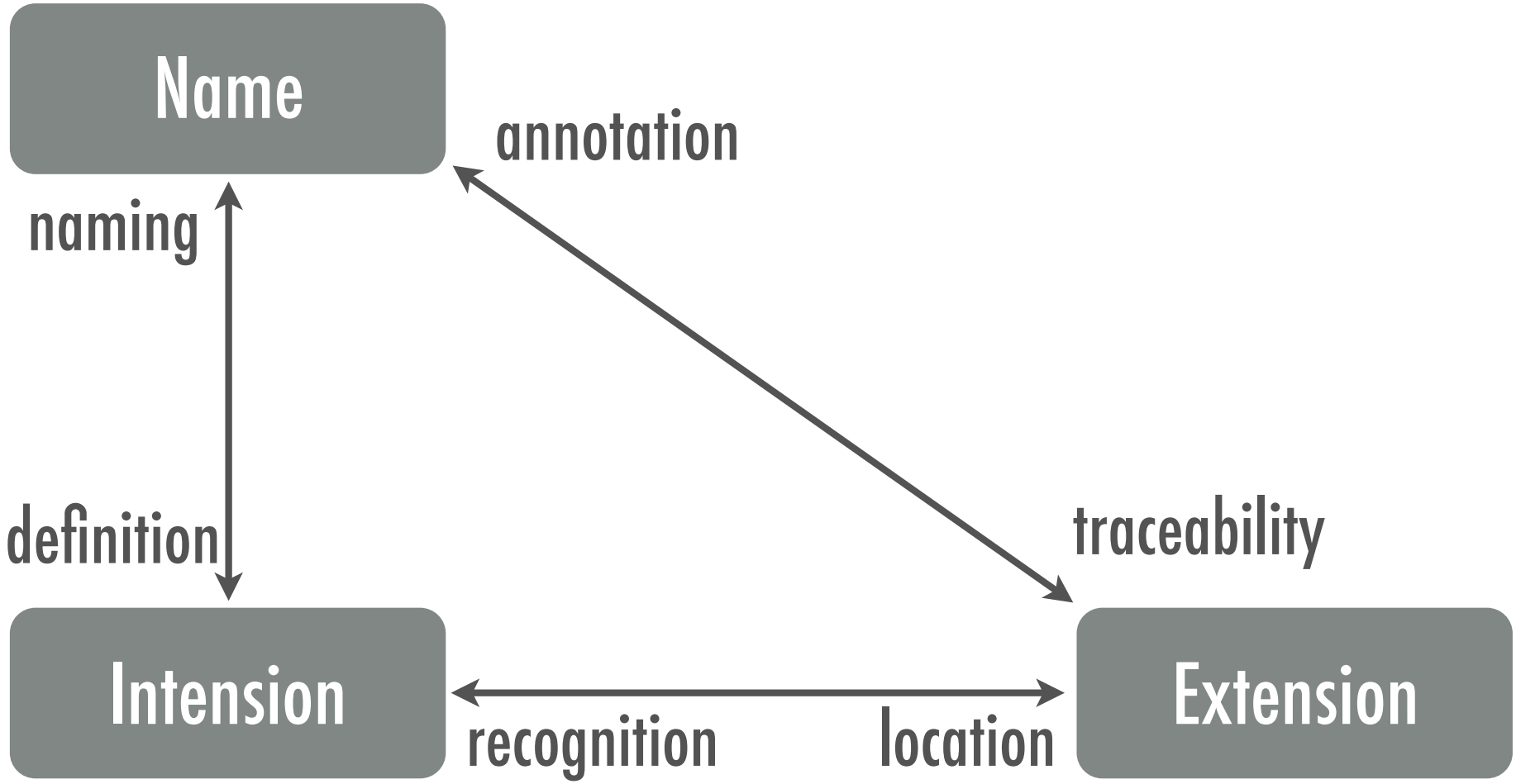
Concept location employs partial comprehension of source code

Large programs cannot be completely comprehended.

- Programmers seek the **minimum essential understanding** for a particular task
- Using an as-needed strategy, they understand how certain specific concepts are reflected in the code

Analogy: visiting a large city

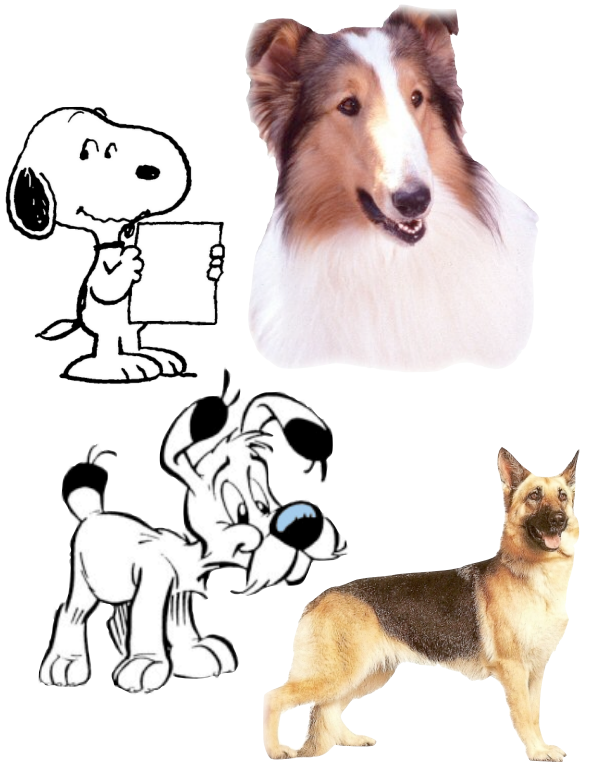
Each concept has three facets



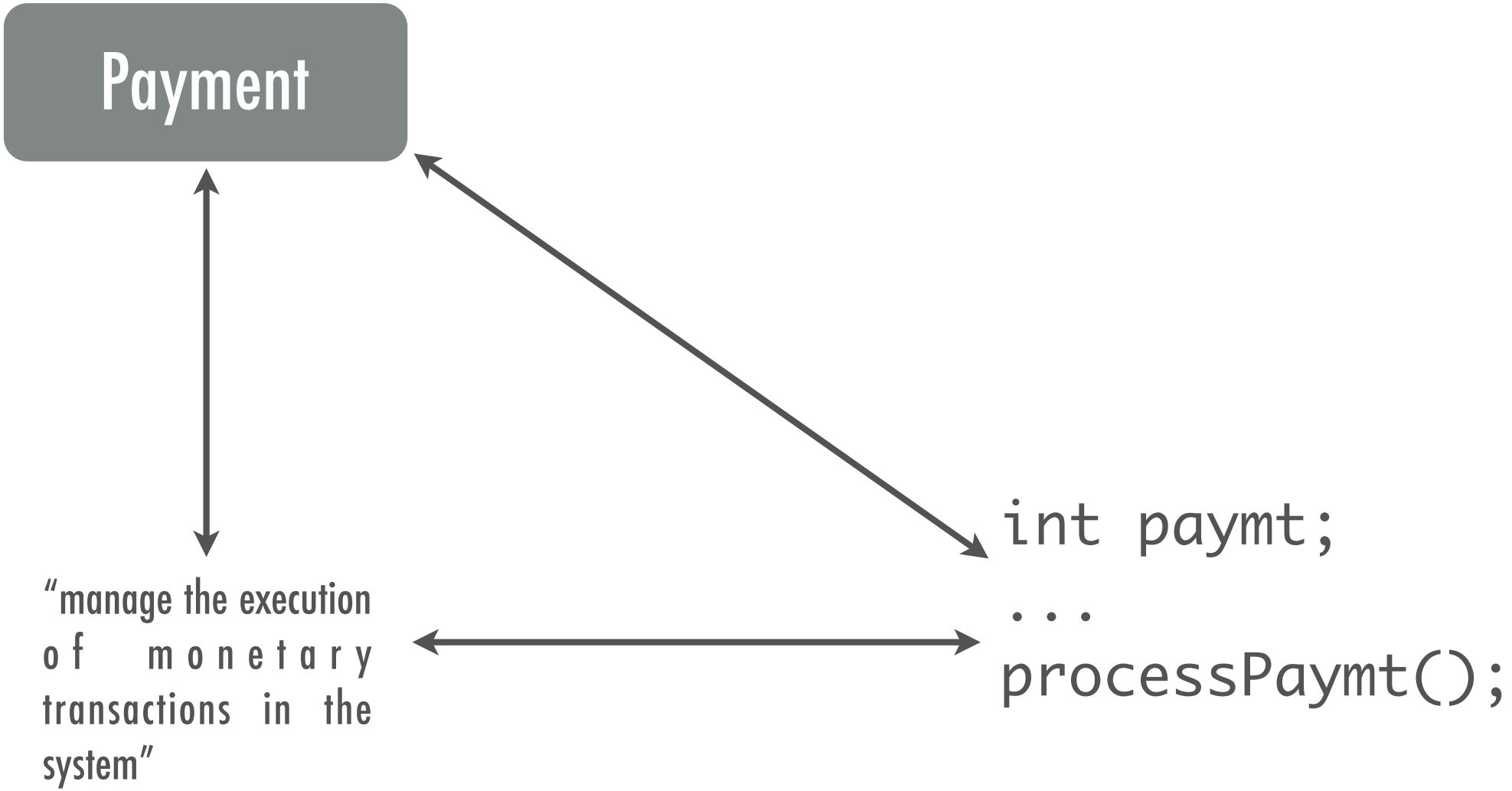
For instance ...

Dog

"a domesticated carnivorous mammal that typically has a long snout, an acute sense of smell, and a barking, howling, or whining voice. "



Or ...

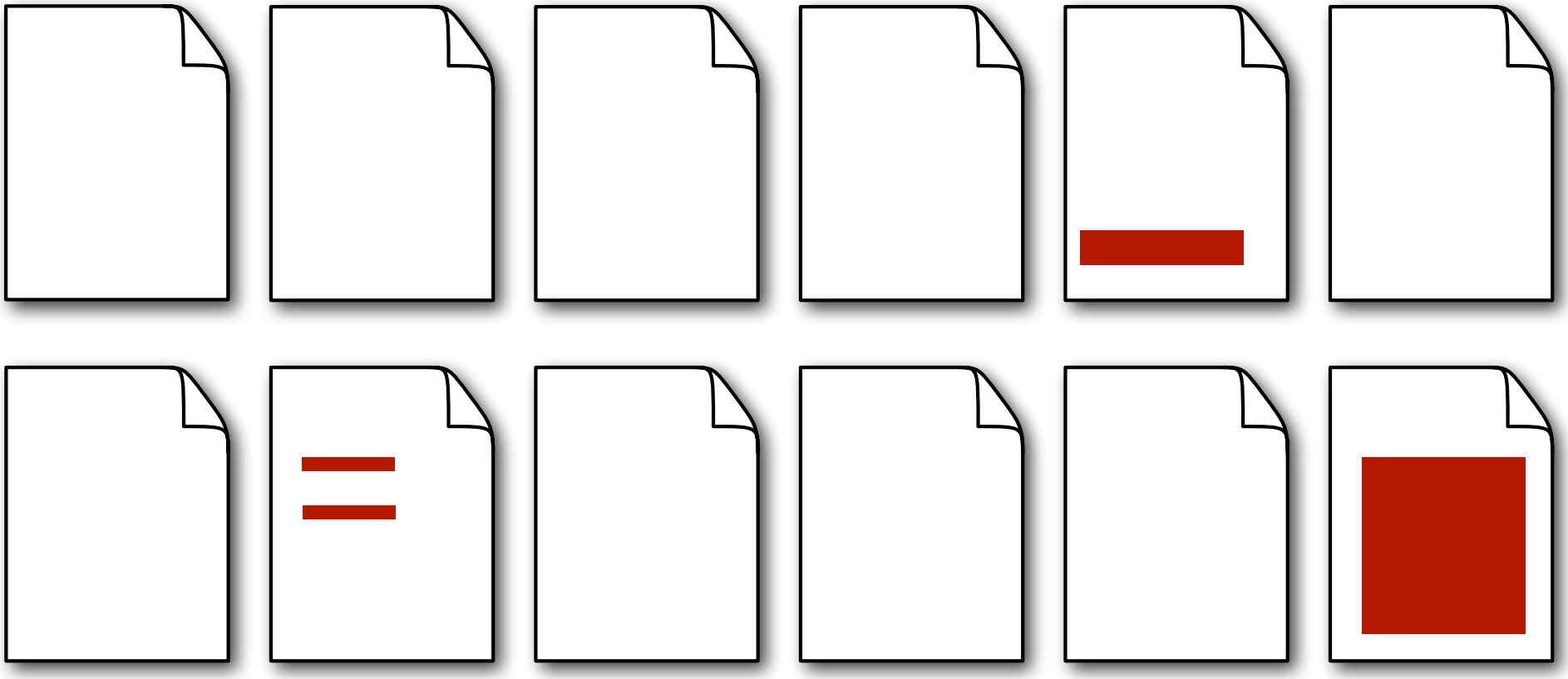

Payment
"manage the execution of monetary transactions in the system"
int paymt;
...
processPaymt();

There are six comprehension processes

location: intension → extension
recognition: extension → intension
naming: intension → name
definition: name → intension
annotation: extension → name
traceability: name → extension

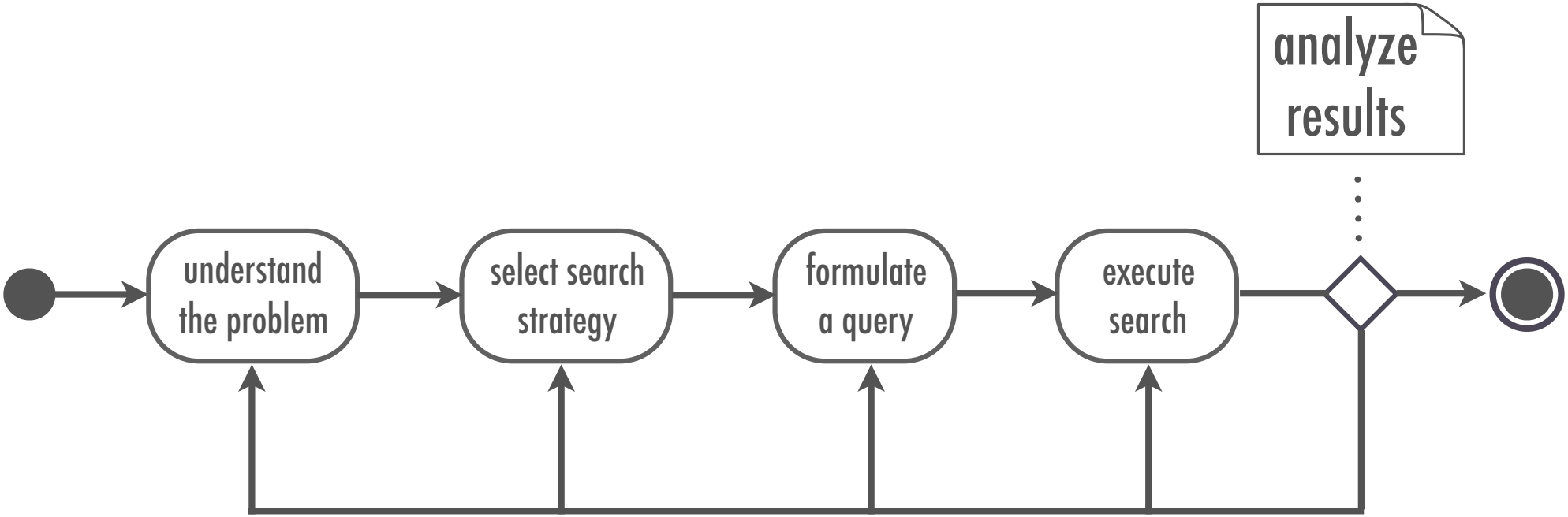
Concept Location

Concept location is the first step of software change



We search for one (or more) starting points

The concept location process



Search strategies start from the intension to the extension

source code only
documentation
requirements to design to code
execution traces
...

Implicit vs explicit concept extensions

explicit: directly appear in the code
implicit: implied by the code, no specific code location

page count in a word processor
vs.
file permissions
(implied by the file access API)

Extracting significant concepts

A change request can have many concepts

- analyze and understand the change request
- extract the concepts in the request
- filter communication with the programmer
- filter concepts unlikely to be present in the code
- rank remaining concepts by likelihood

Example

Implement a credit card payment

Communication with the programmer

Concept to be implemented

Significant concept

Example

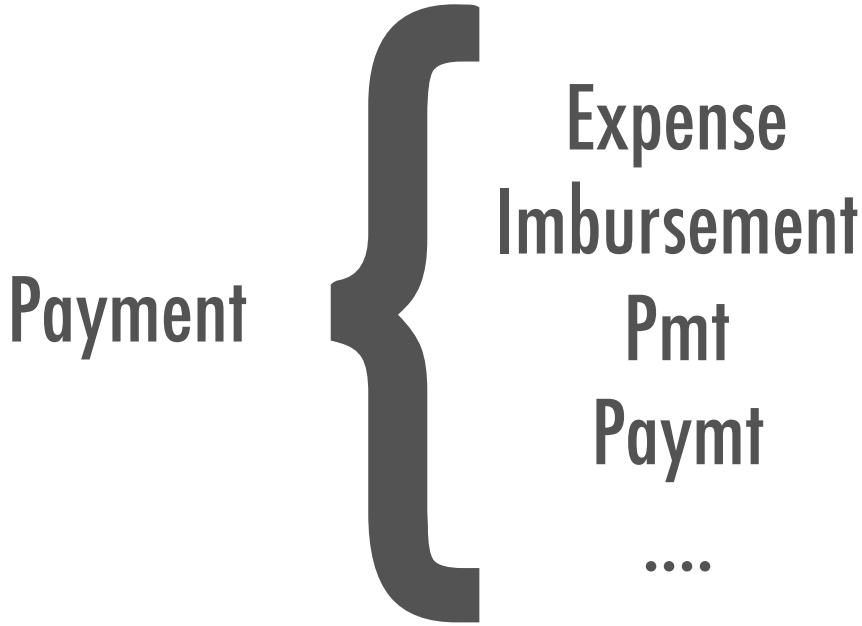
Fix the $!*%^#$ broken credit card payment

Communication with the programmer

Concept to be implemented

Significant concept

Homonyms, synonyms, make concept location more difficult



Concepts can be homonyms and bear the same name
(e.g. “x”, “item”, “pitbull”)

Concept location strategies

Concept location via grep

assumption: “concept extensions are labeled by concept names”

intension → name → extension

query can be a name or a pattern

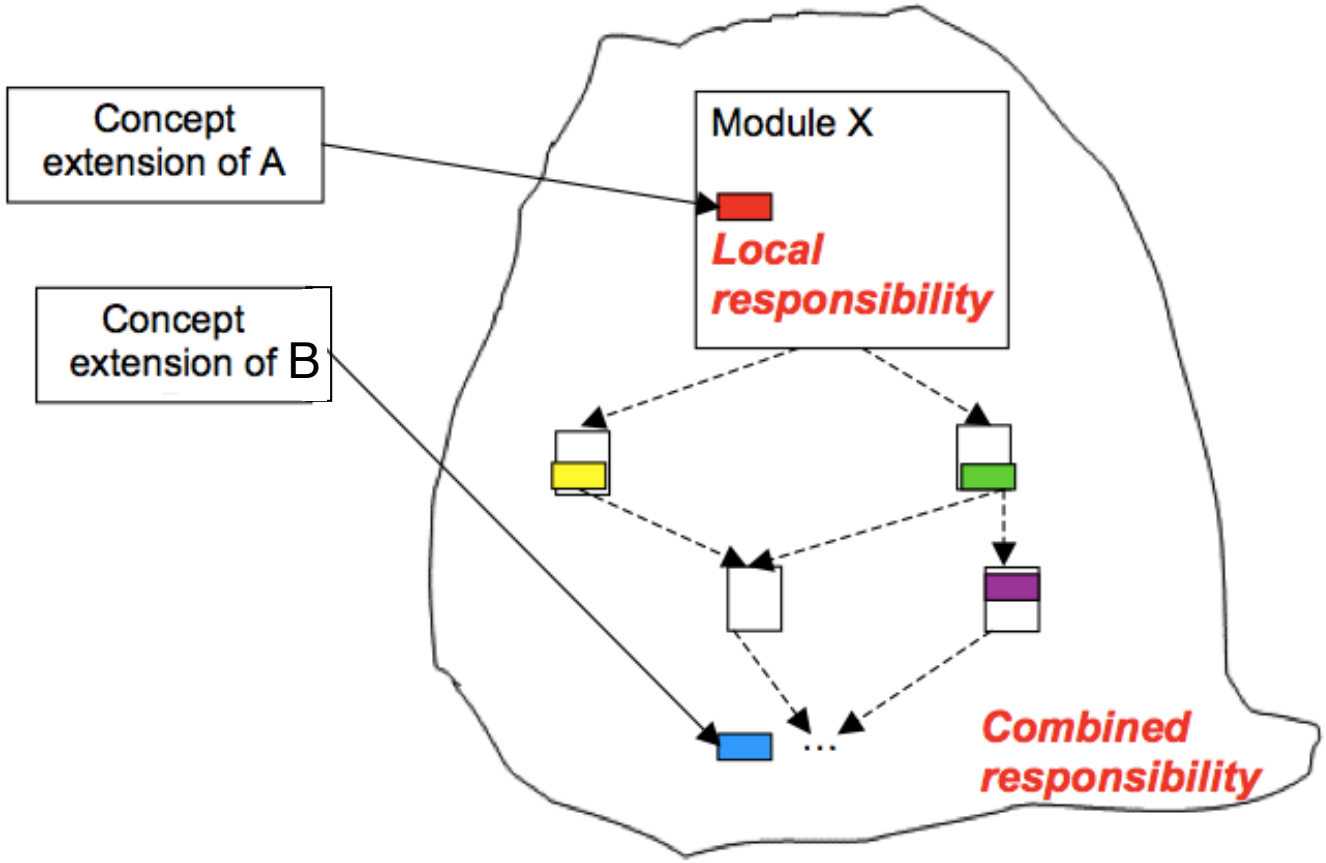
```
grep pay*
```

Failures when using grep

no matches: try an alternate query
wrong matches: in case of homonyms
too many matches: refine query

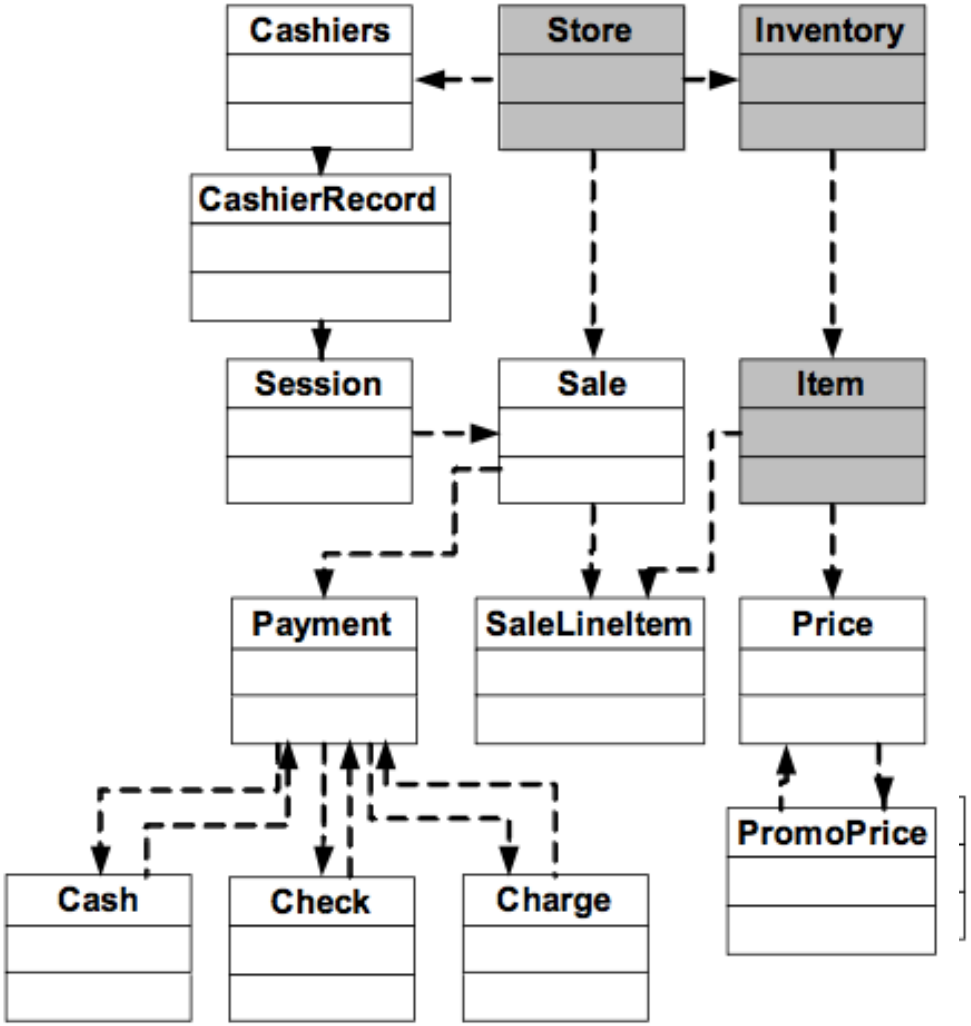
harder with implicit concepts
easier when code conventions are respected

Concept location by dependency search

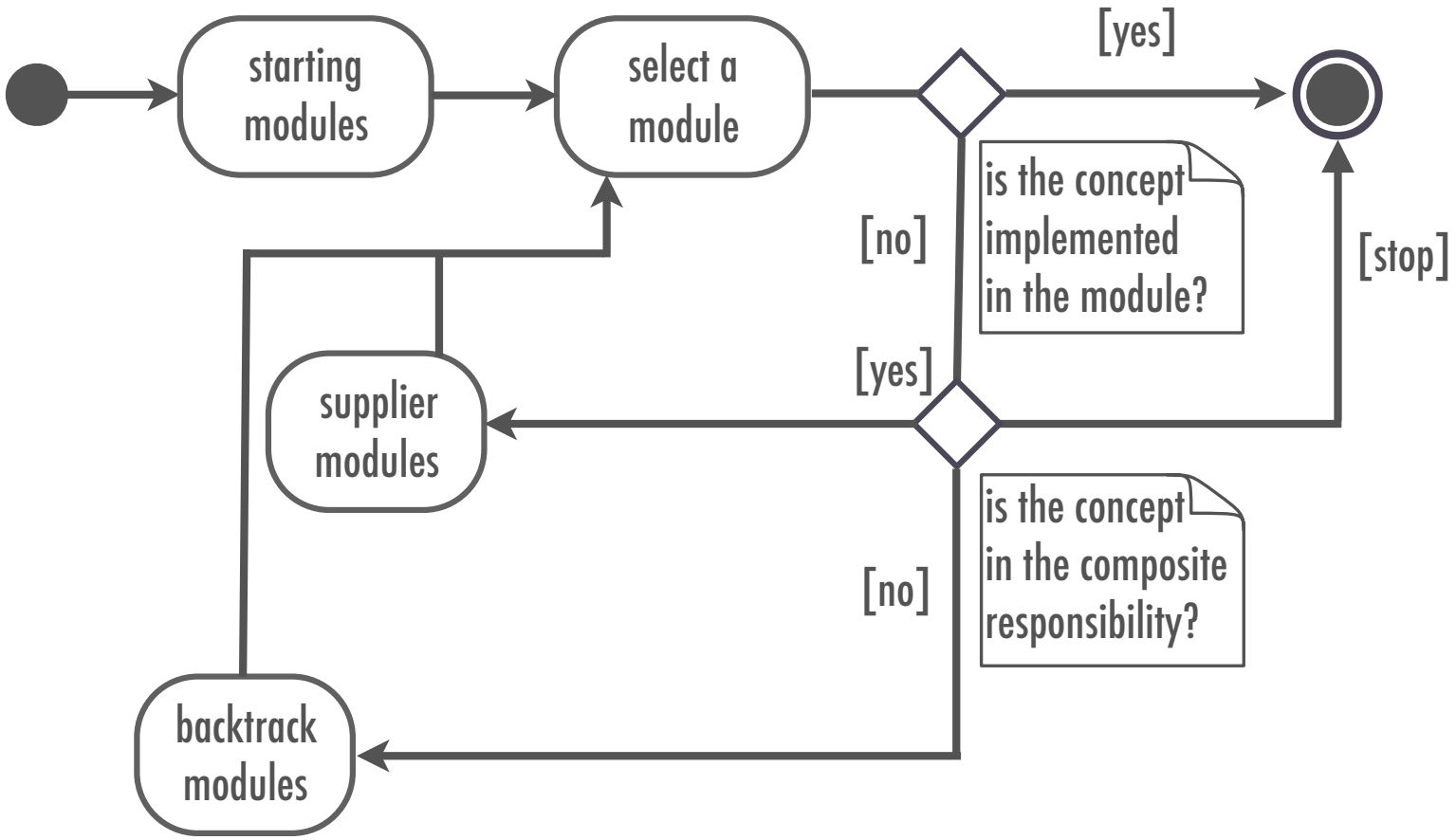


Local vs composite responsibility

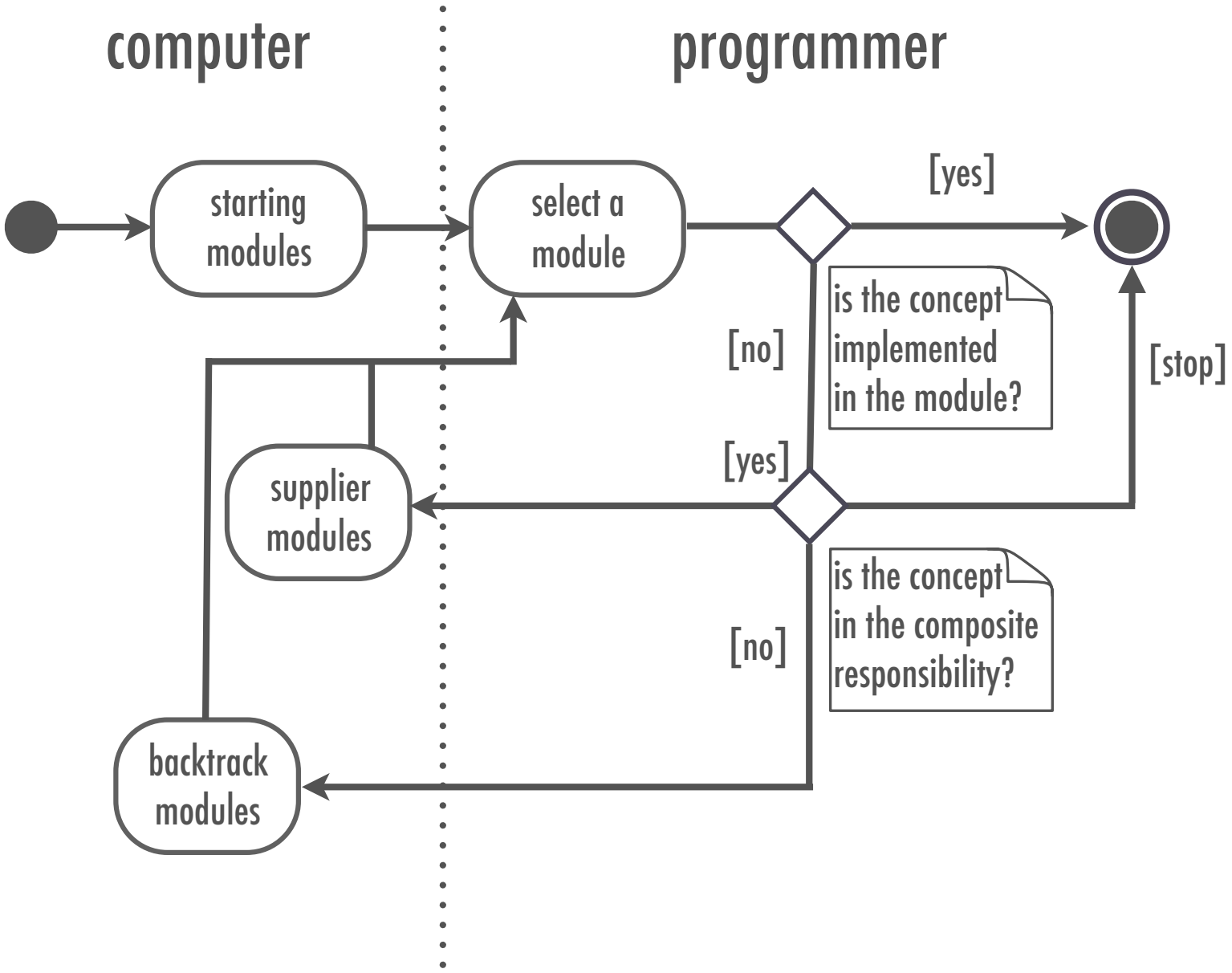
Dependency search uses class dependency graphs, or UML class diagrams



Dependency search process



Tool support



Other concept location strategies

Using information retrieval (LSI, LDA)
Using dynamic information
Using the debugger to confirm hypotheses

[Antoniol, Marcus, Poshyvanyk]

Independent of which strategy you choose

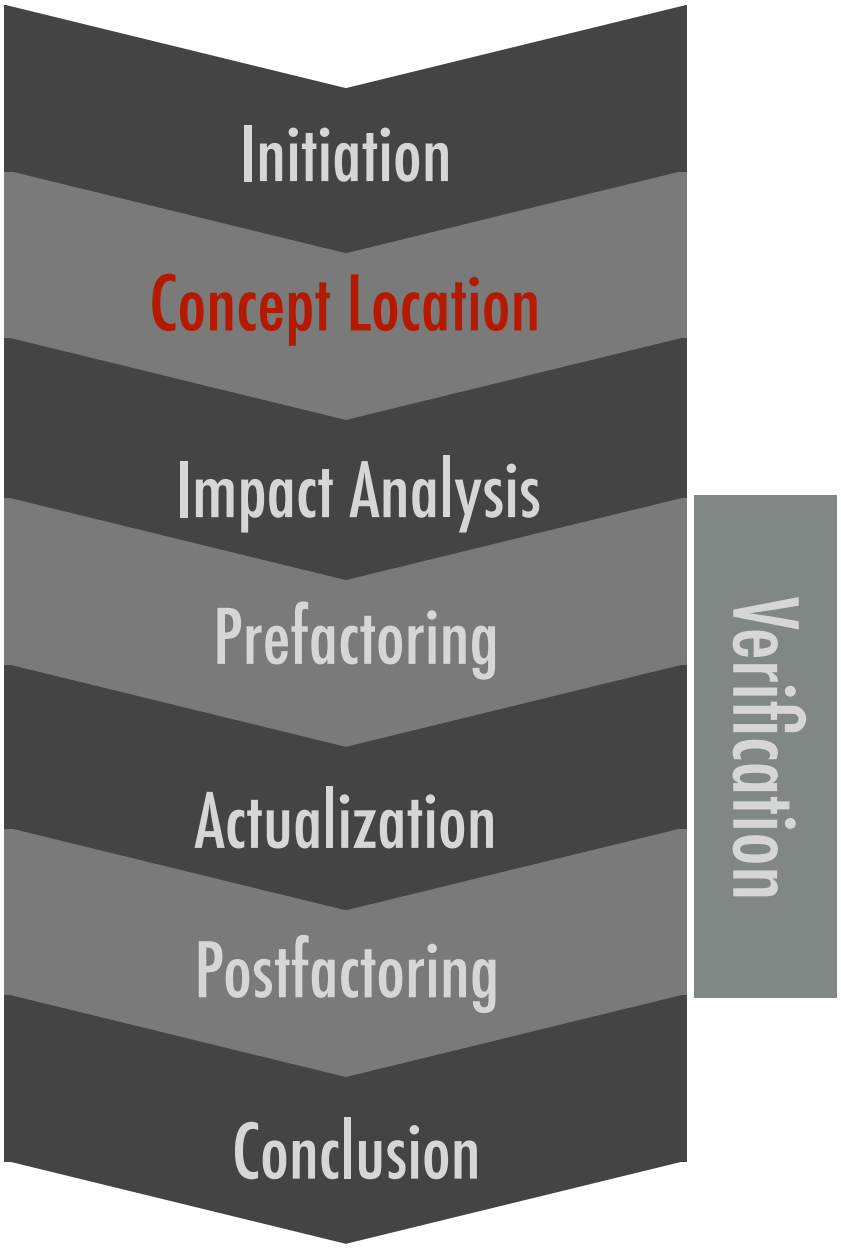
TAKE NOTES!

Generating knowledge about software is difficult and expensive ...
Don’t lose it!

Examples

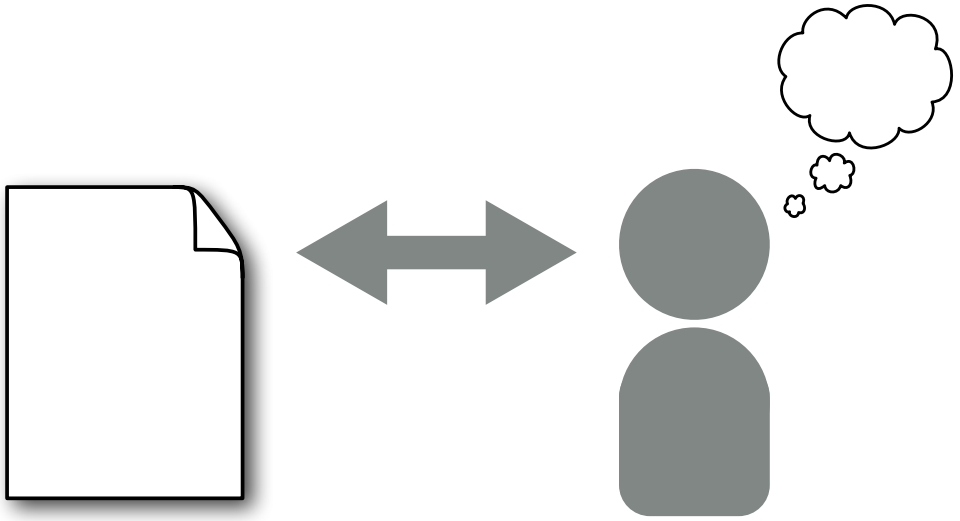
Conclusion

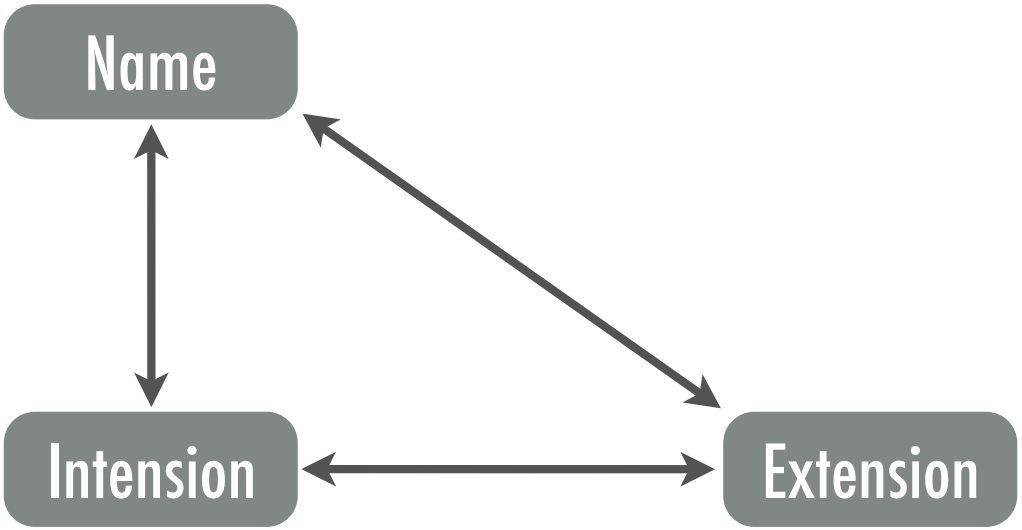
Concept location finds a starting point



The programmer looks for the module that needs to be changed to implement the change request

Concept location is a search





via grep
via dependencies
via dynamic
information