# Source Code Concept Location With Multilingual Support

Nuno Carvalho[1] Alberto Simões[2] José João Almeida[1]

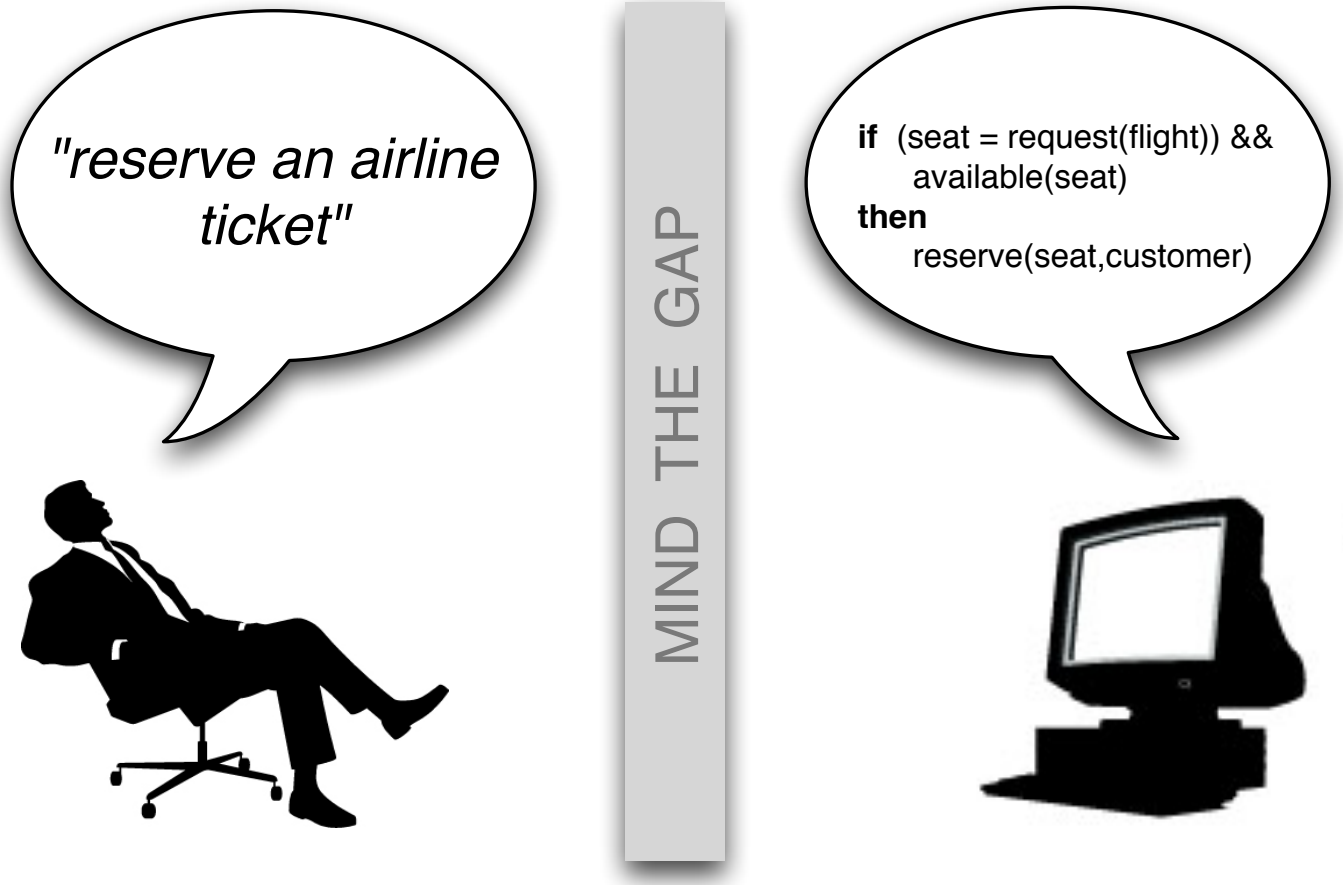
[1]Departamento de Informática, Universidade do Minho
{narcarvalho,jj}@di.uminho.pt

[2]Centro de Estudos Humanísticos, Universidade do Minho
ambs@ilch.uminho.pt

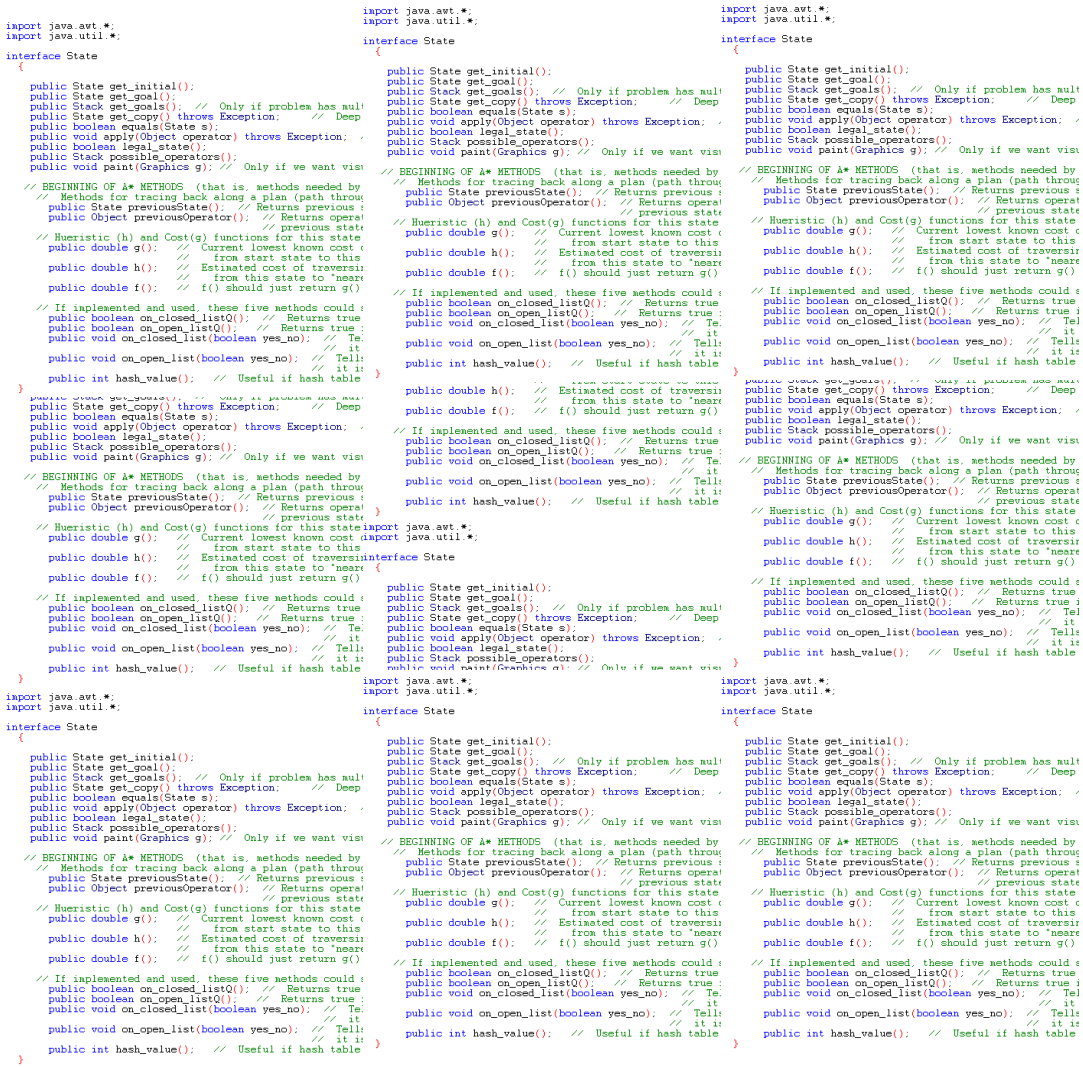
Per-Fide Workshop
November 22nd, 2011

- today, everything depends on information systems
- humans need to teach computers how to perform tasks
- humans write programs
- programs are written using programming languages
- programs require care
  - have bugs, add features, improvement, ...
  - complex, expensive, time consuming task
- humans devise ways to quicker and better understand programs
- reverse engineering of software – Program Comprehension
- concept location is a common task
  - identify parts of the programs responsible for implementing real life concepts

# Natural Language VS Formal Languages

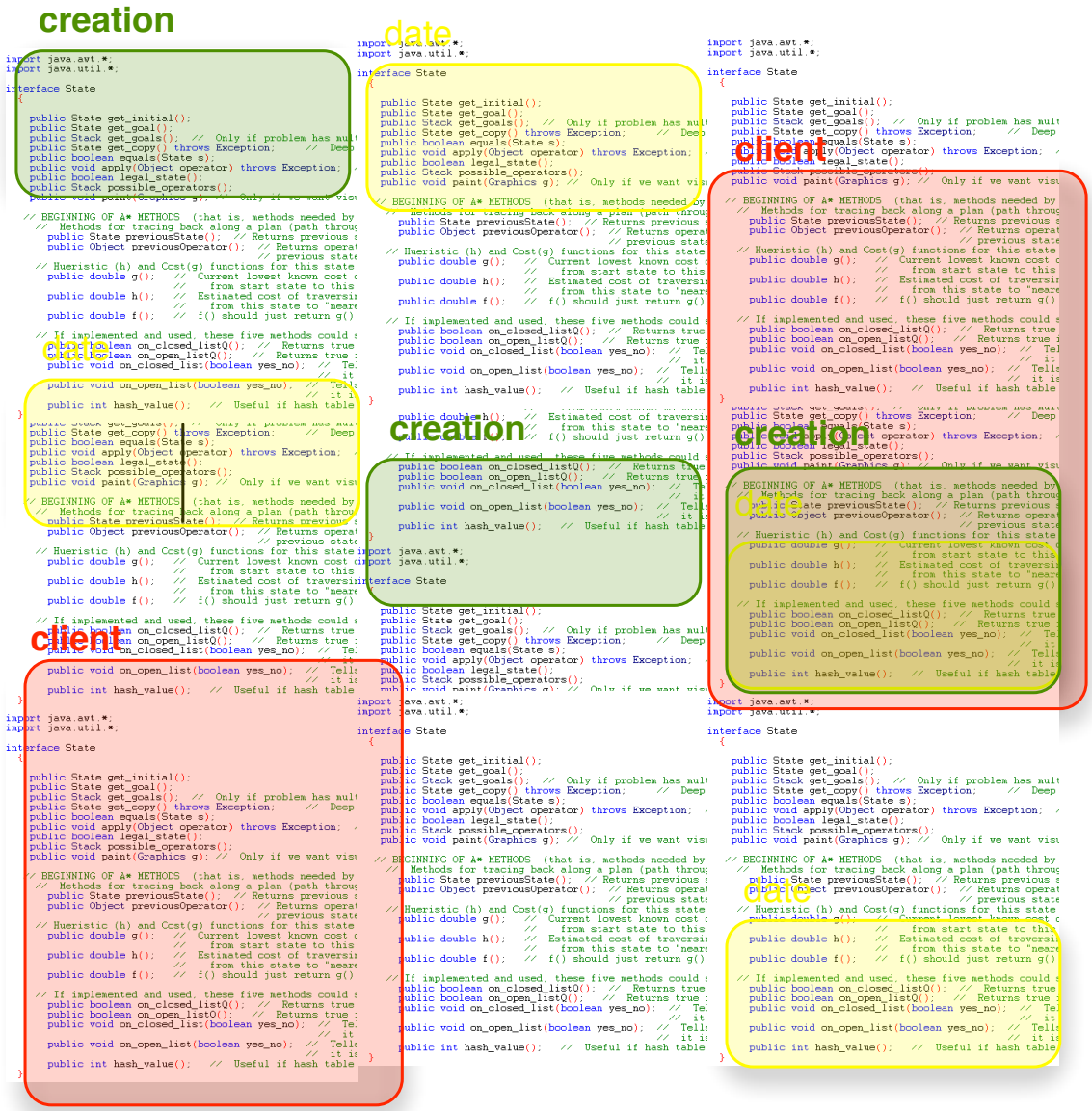


# Source Code

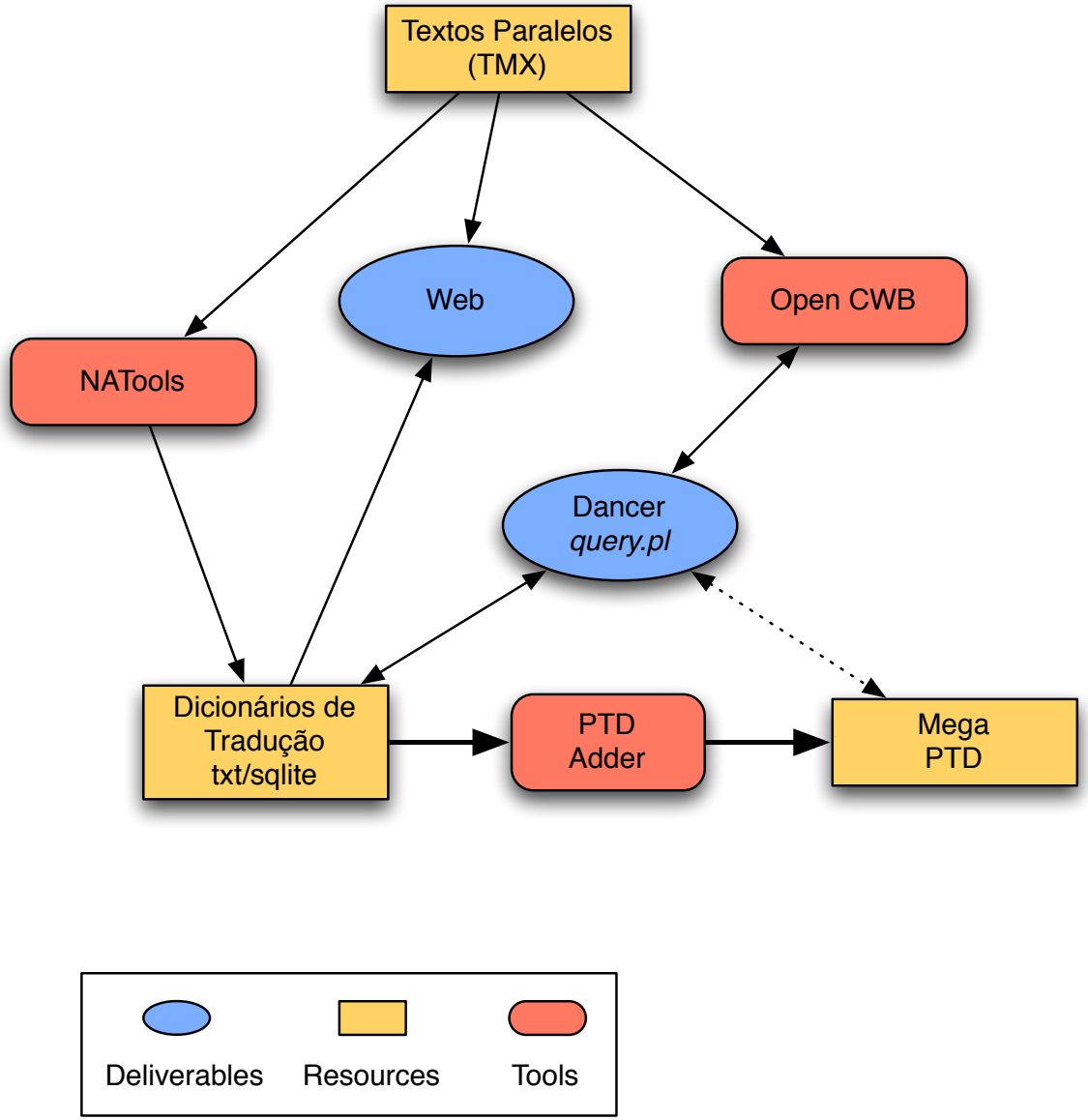


```
import java.awt.*;
import java.util.*;

interface State
  {

    public State get_initial();
    public State get_goal();
    public Stack get_goals();  //  Only if problem has mult
    public State get_copy() throws Exception;      //  Deep
    public boolean equals(State s);
    public void apply(Object operator) throws Exception;  /
    public boolean legal_state();
    public Stack possible_operators();
    public void paint(Graphics g); //  Only if we want visu

   // BEGINNING OF A* METHODS  (that is, methods needed by
     //  Methods for tracing back along a plan (path throug
       public State previousState();  // Returns previous s
       public Object previousOperator();  // Returns operat
                                          // previous state
     // Hueristic (h) and Cost(g) functions for this state
       public double g();   //  Current lowest known cost (
                            //    from start state to this
       public double h();   //  Estimated cost of traversin
                            //    from this state to "near
       public double f();   //  f() should just return g()

     // If inplemented and used, these five methods could s
       public boolean on_closed_listQ();  //  Returns true
       public boolean on_open_listQ();    //  Returns true
       public void on_closed_list(boolean yes_no);  //  Te
                                                    //  it
       public void on_open_list(boolean yes_no);  //  Tells
                                                  //  it is
       public int hash_value();   //  Useful if hash table
  }
```

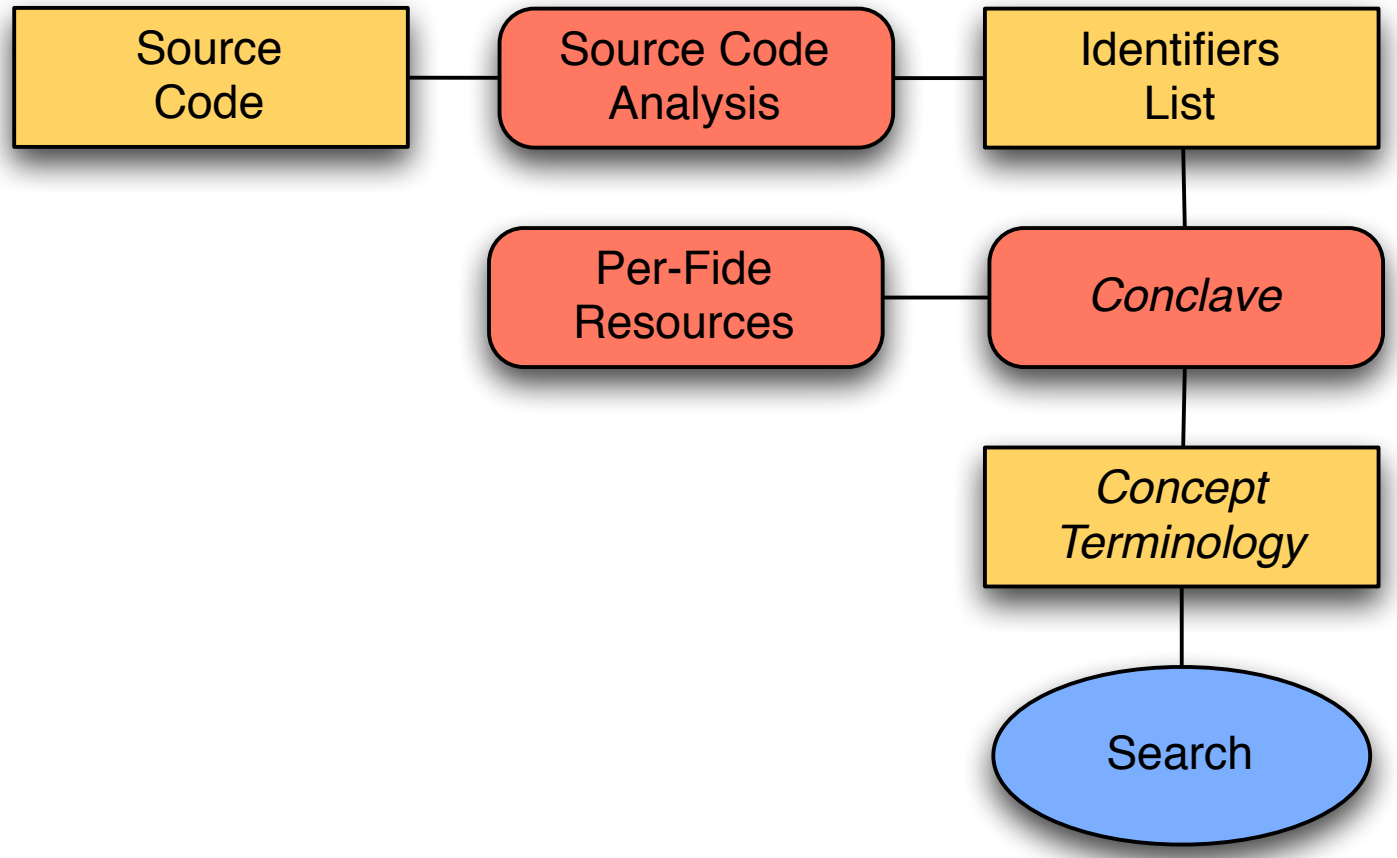
# Source Code Implements Concepts



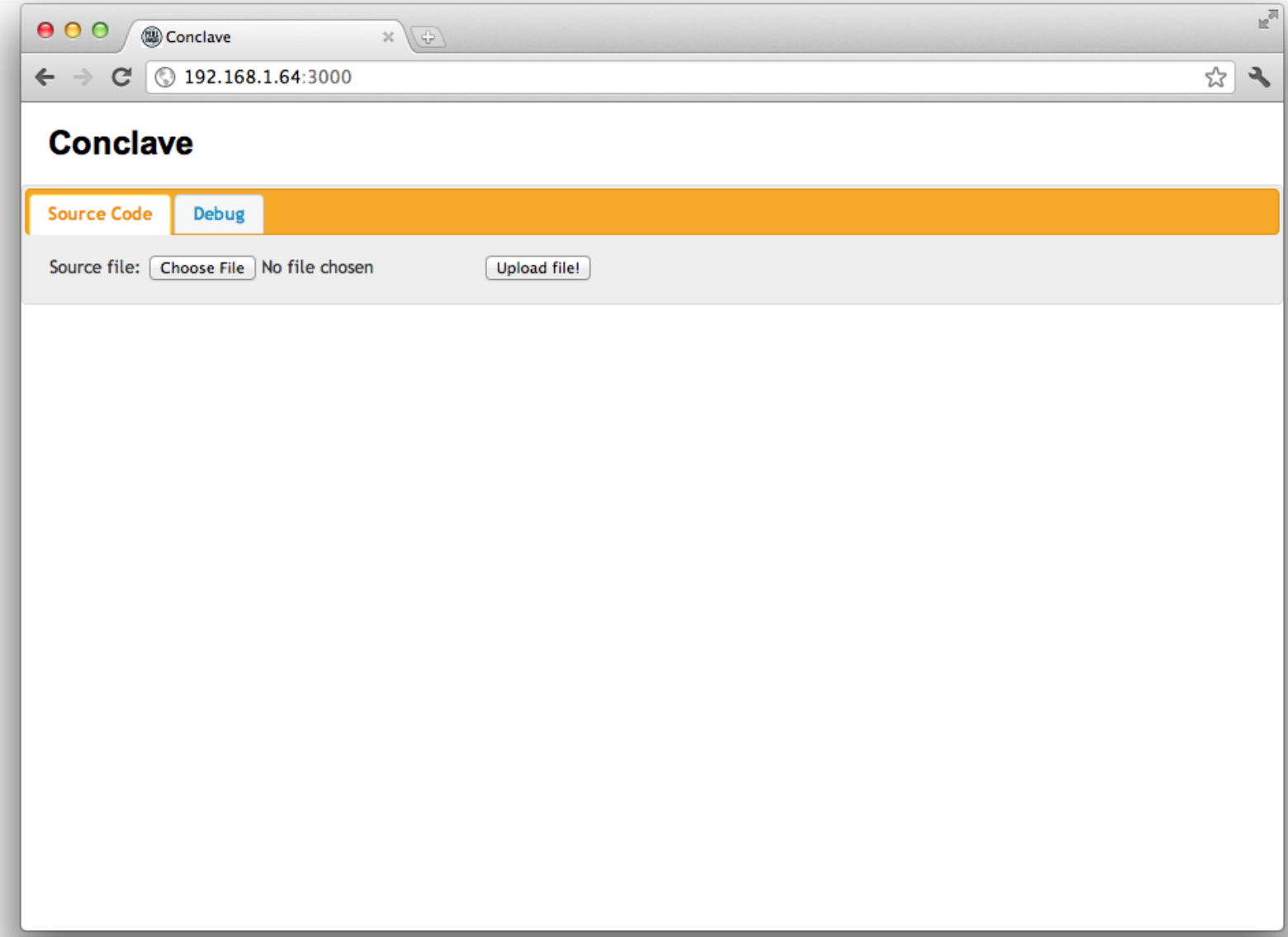
# Per-Fide Environment



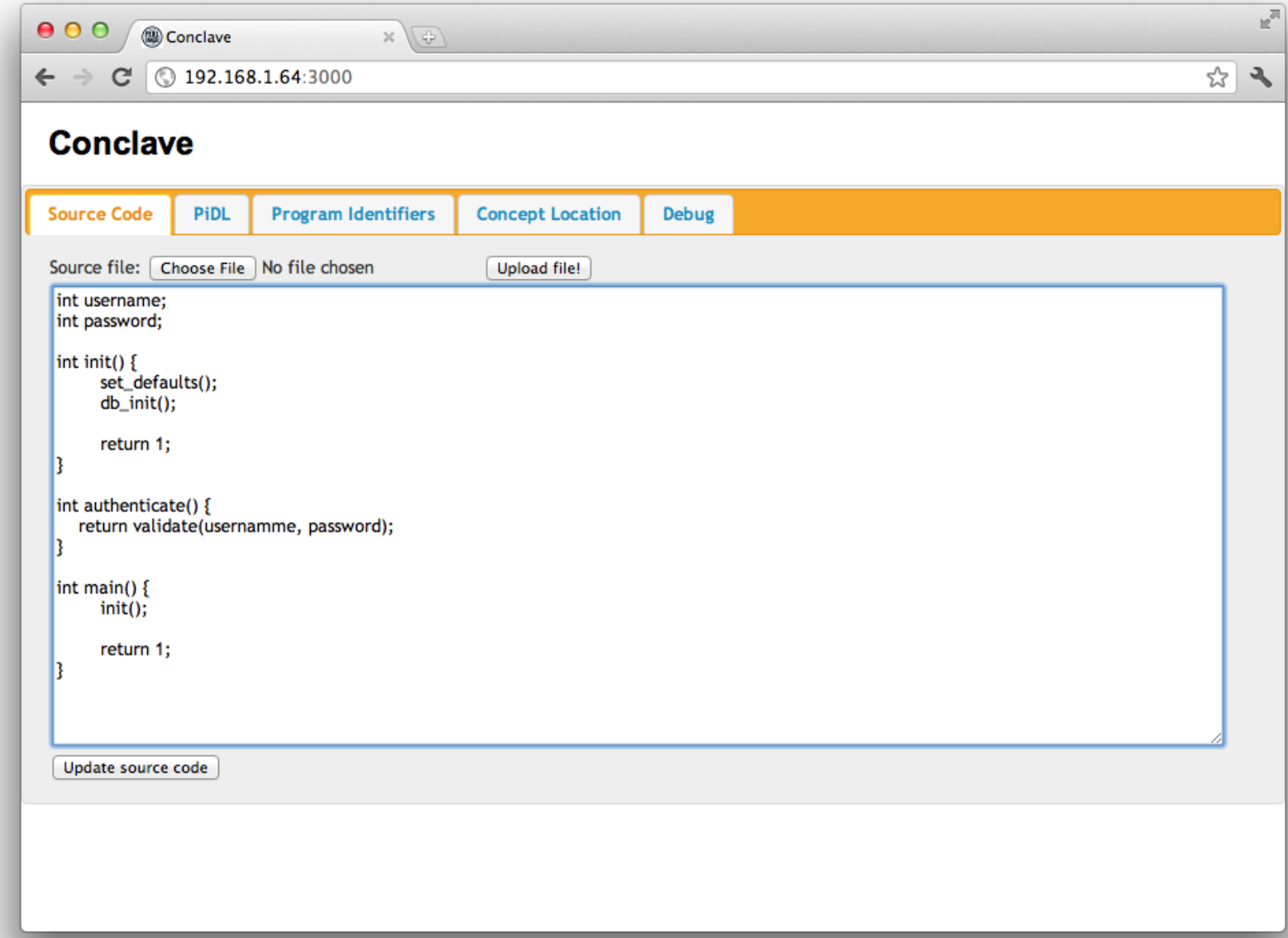
# Workflow


Source Code
Source Code Analysis
Identifiers List
Per-Fide Resources
Conclave
Concept Terminology
Search

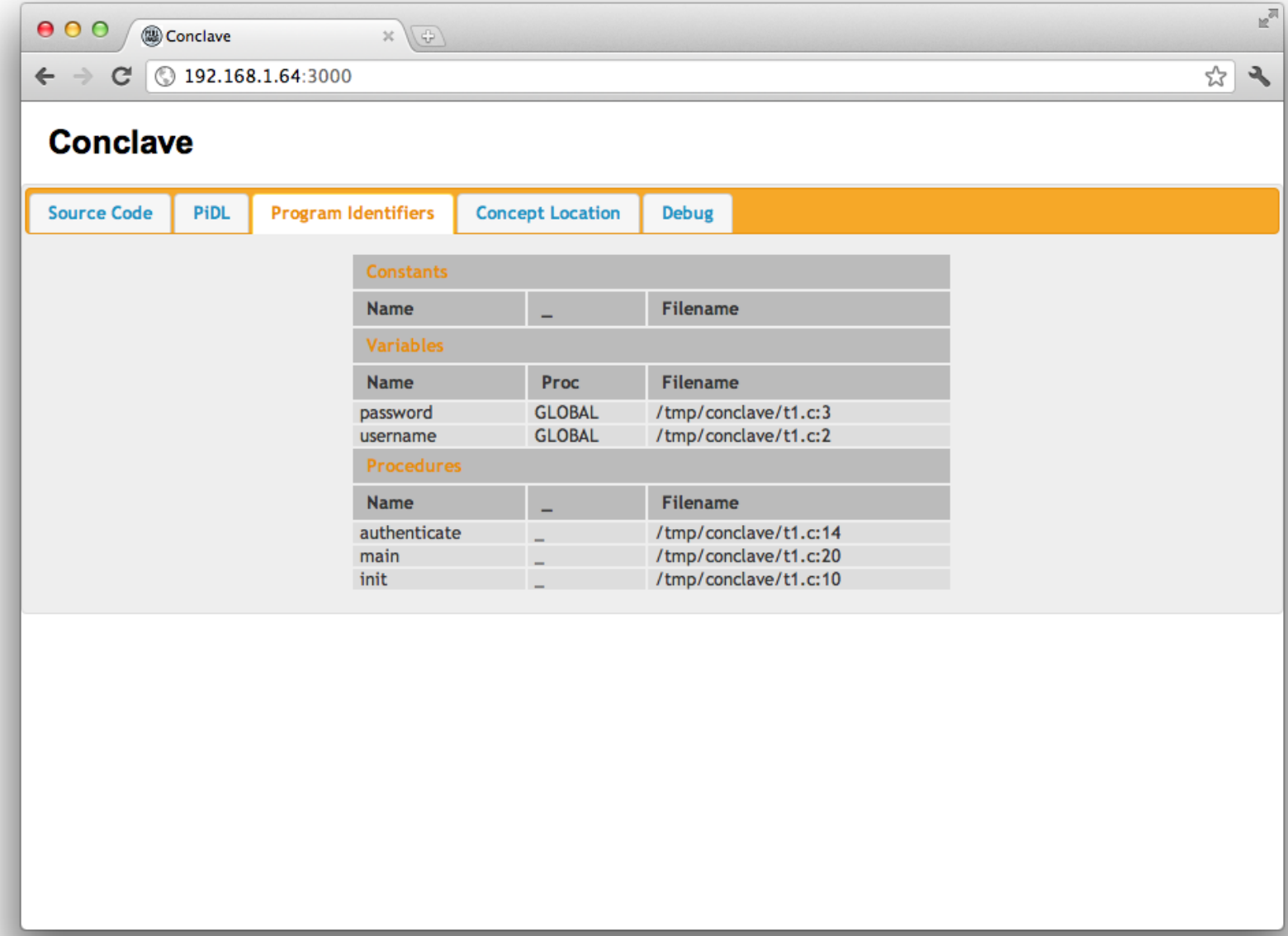
# Upload Source Code



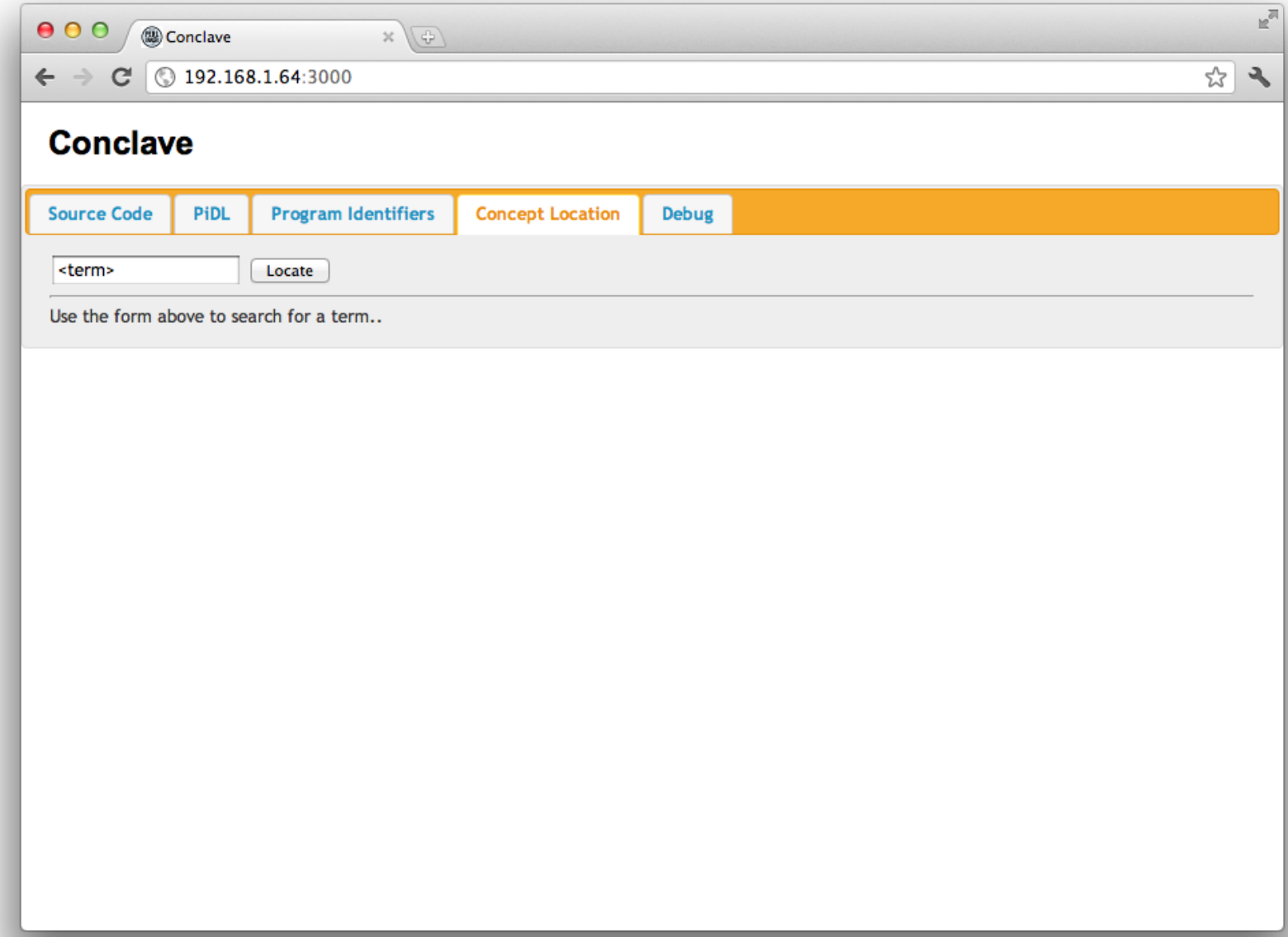
# Source Code


Conclave
192.168.1.64:3000
Conclave
Source Code
PiDL
Program Identifiers
Concept Location
Debug
Source file:
Choose File
No file chosen
Upload file!
int username;
int password;
int init() {
set_defaults();
db_init();
return 1;
}
int authenticate() {
return validate(usernamme, password);
}
int main() {
init();
return 1;
}
Update source code

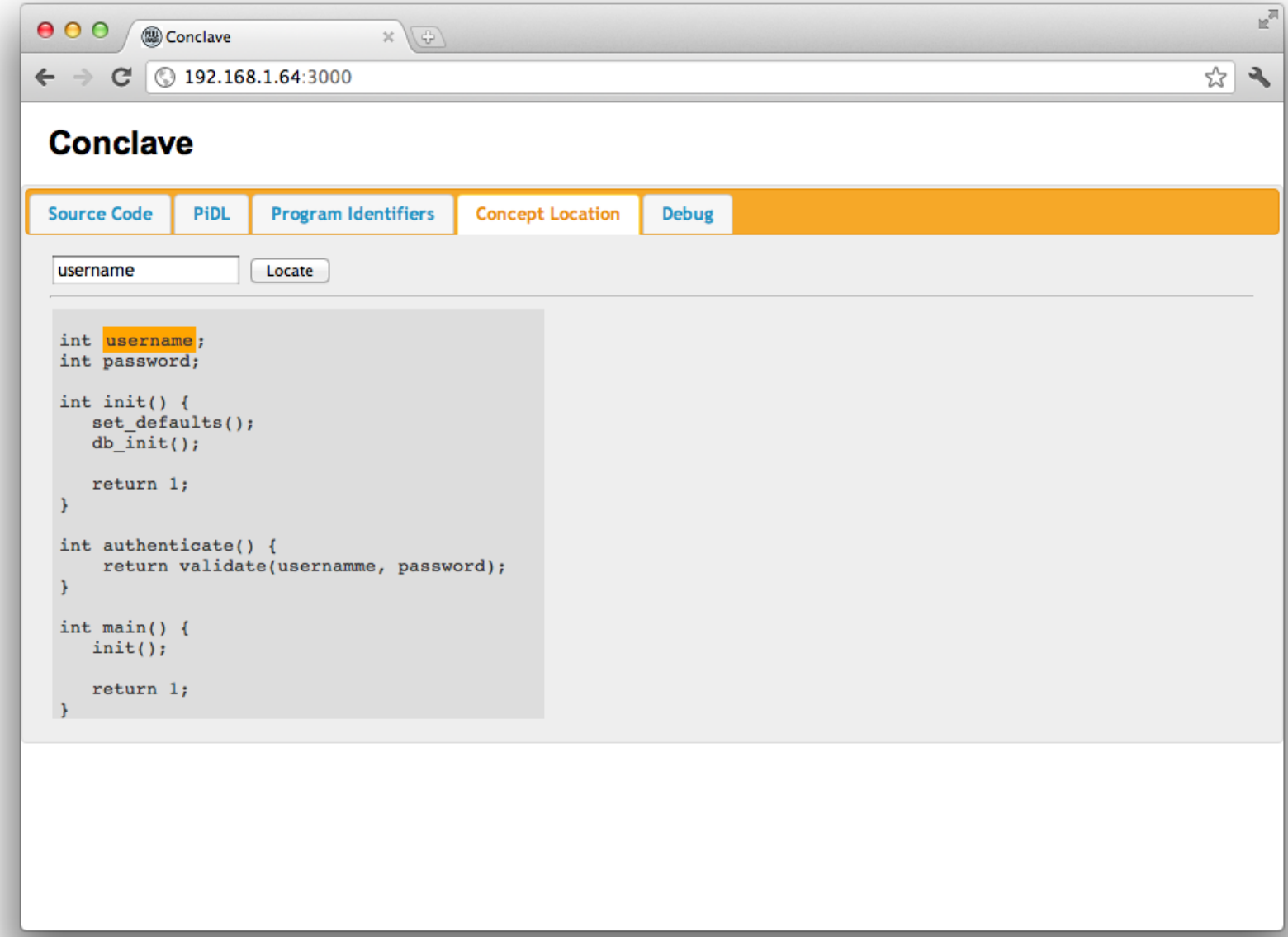
# Identifiers List Found



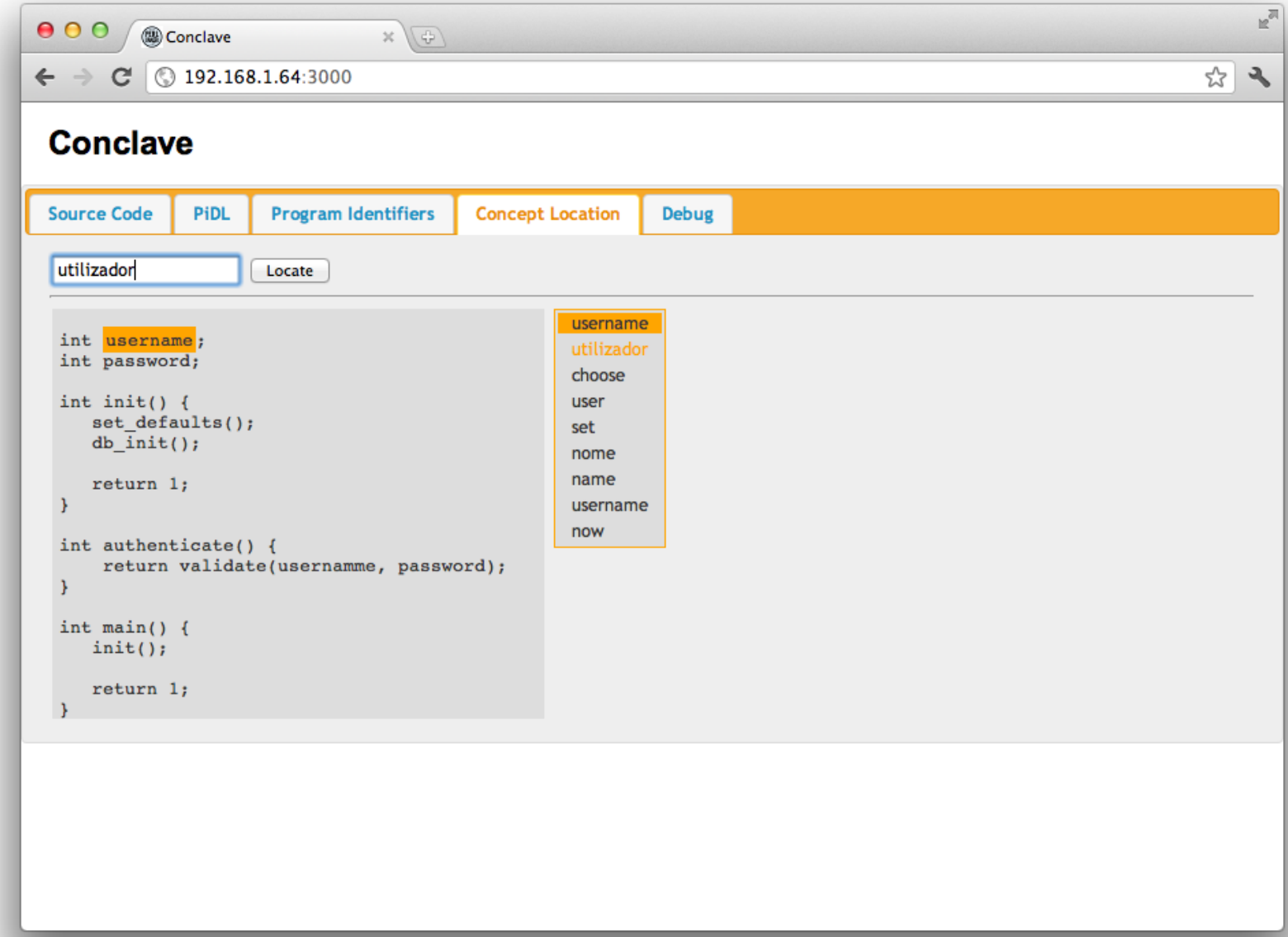
# Search Terms



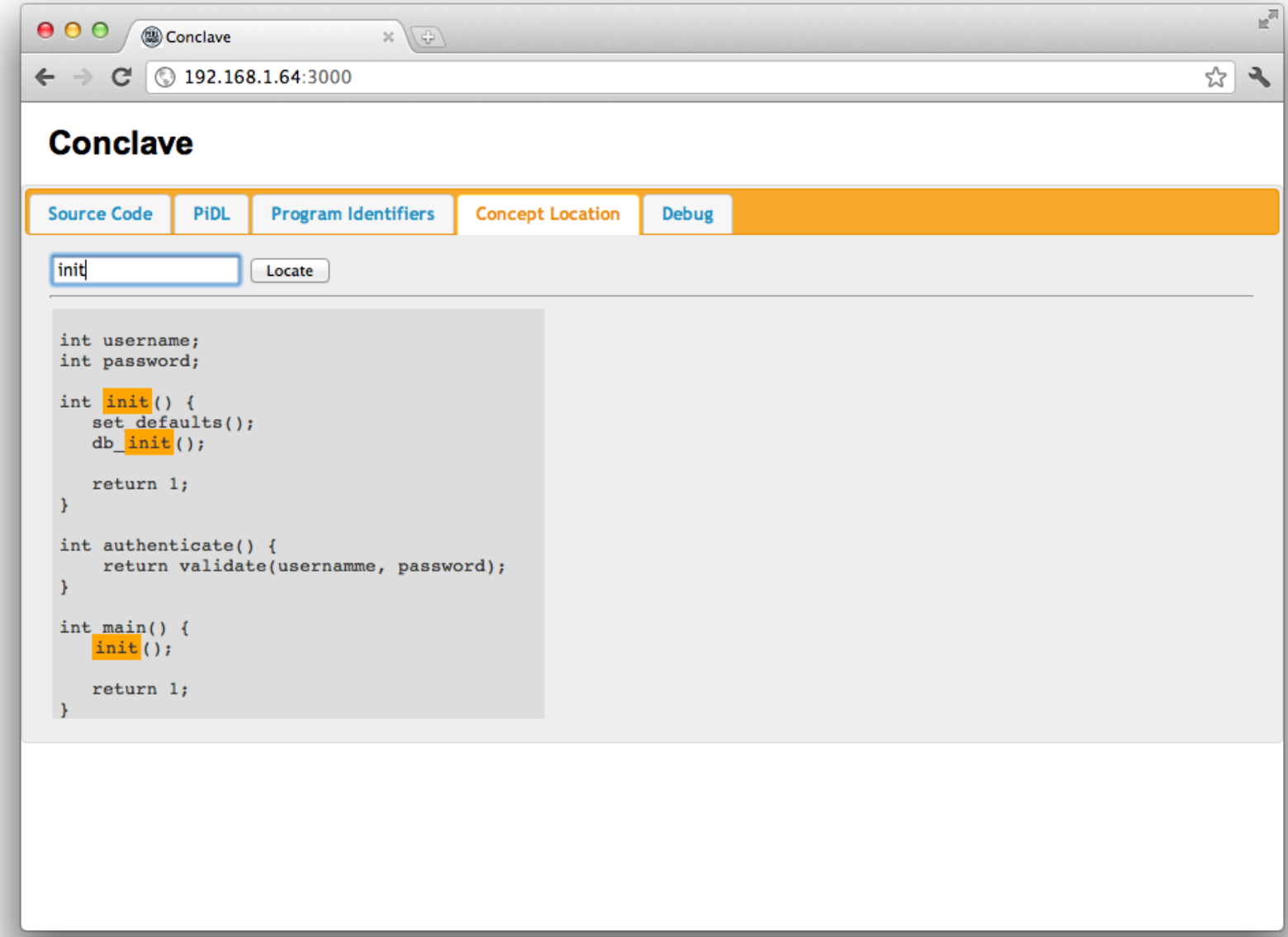
# Trivial Match



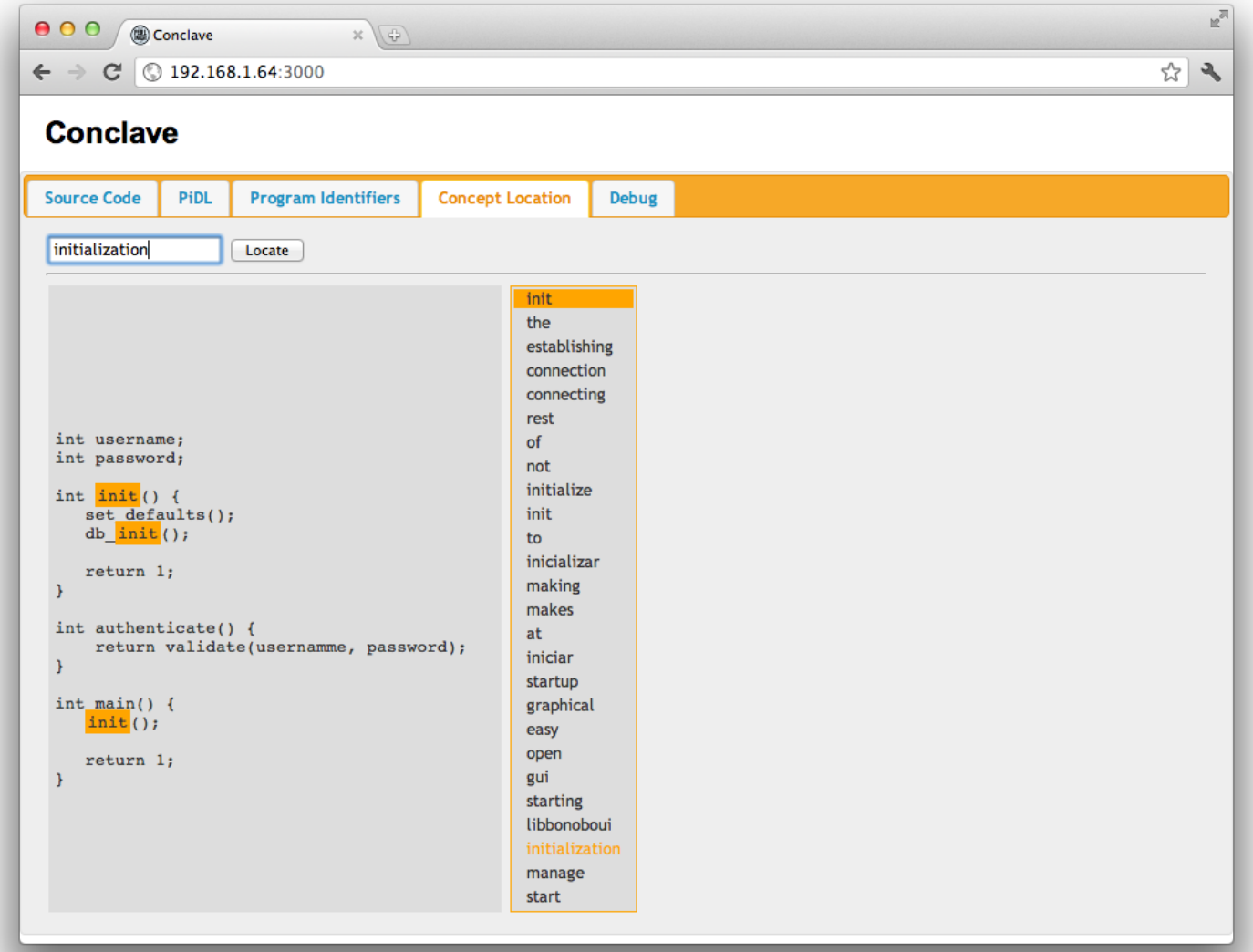
# Match In Another Language



# Another Trivial Match



# Match Another Term, Same Concept



# Conclusion

- share Per-Fide resources
  - these can be used in ways we don't think of (people have an incredible imagination)
  - others can take advantage of the projects' effort
- use of NLP resources in Program Comprehension
- clear advantages
  - more than multilingual support
  - more trust in translations, specialized corpora
  - look also for similar terms
  - (more)
- future work
  - enrich concepts terminology (based on comments for example)
  - more robust tool, interface improvements, ...
  - elegant way to share resources

Thank You!