Software Developer Behavior Simulator

COMP-522 Project (Fall 2004)

Philippe Nguyen

Presentation Overview

- Introduction
- Functional Description
- Design
- Evaluation
- Limitations
- Concluding Remarks

Introduction

- Importance of human behavior in the practice of software development
- More and more research is aimed at understanding how developers conduct certain tasks
- Empirical studies are expensive
- Based on M. Robillard’s work on the effectiveness of developers

Goals

- Primary:
  - Build a simulator that models a software developer’s behavior performing a code navigation task
- Scope:
  - Establish a working framework upon which more sophistication can be added

Functional Description

- Inputs:
  - Source model
  - Concern description
  - Level of developer
  - Seed element
- Output:
  - Investigation transcript
    - Sequence of {Element, Activity} tuples

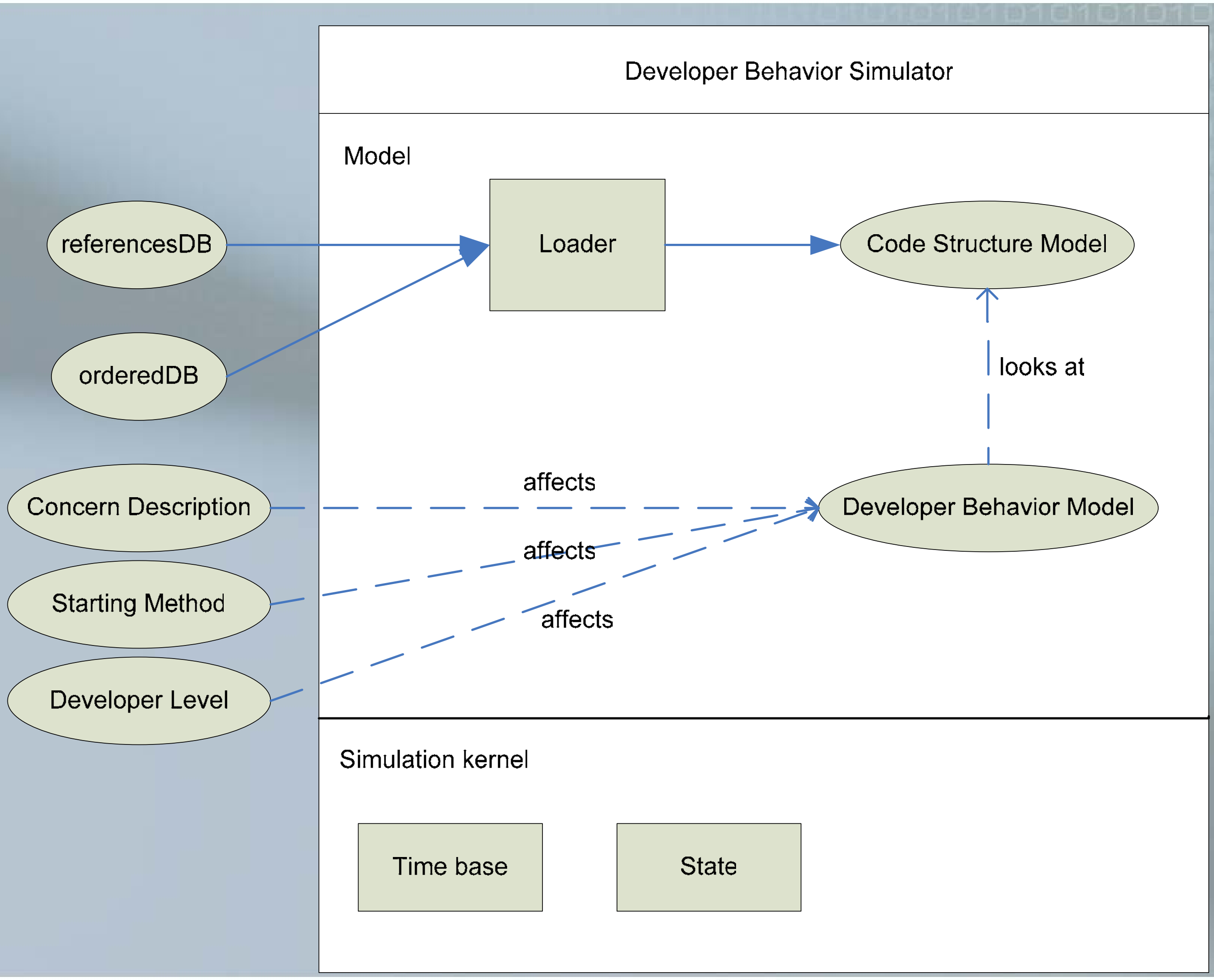
Challenges

- Level of smartness of the behavior model
- Termination
- Quantification of qualitative descriptions
- Define simulator:
  - Time-base
  - State
  - Transition function

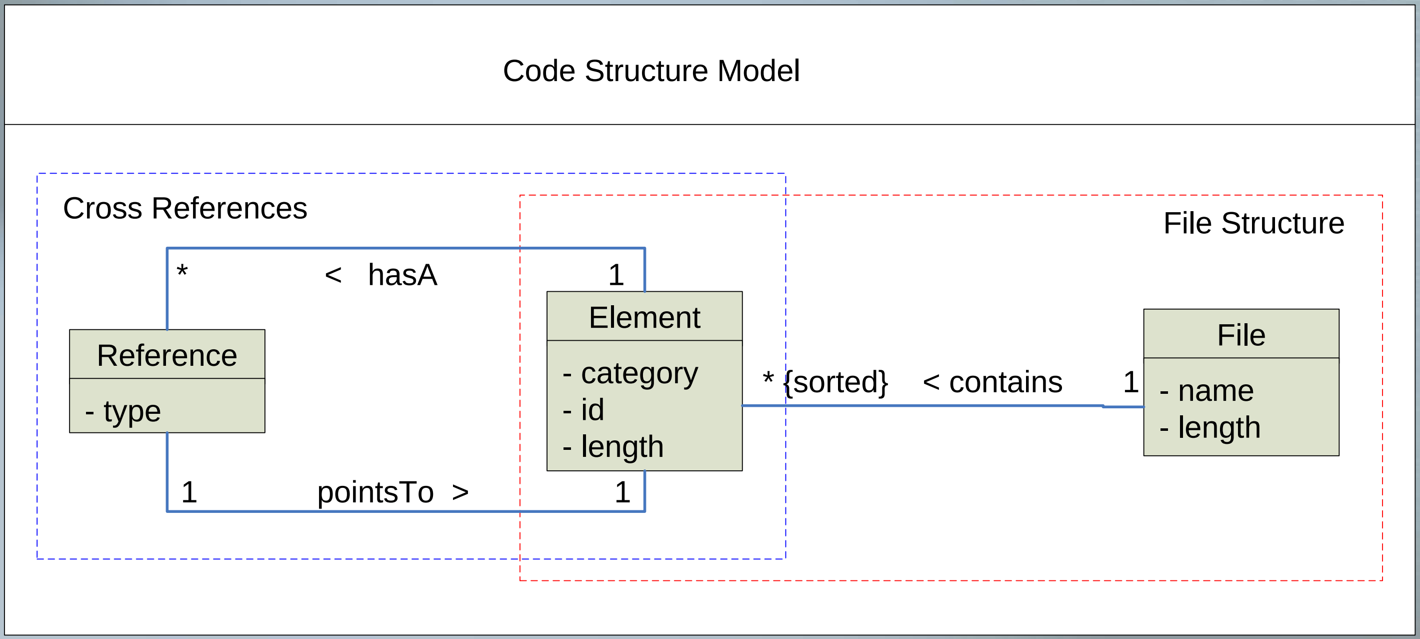
Design

- Non-deterministic decision generator
- Probabilistic choices
- Assign the proper probability to each possible choice in order to simulate a real developer as closely as possible

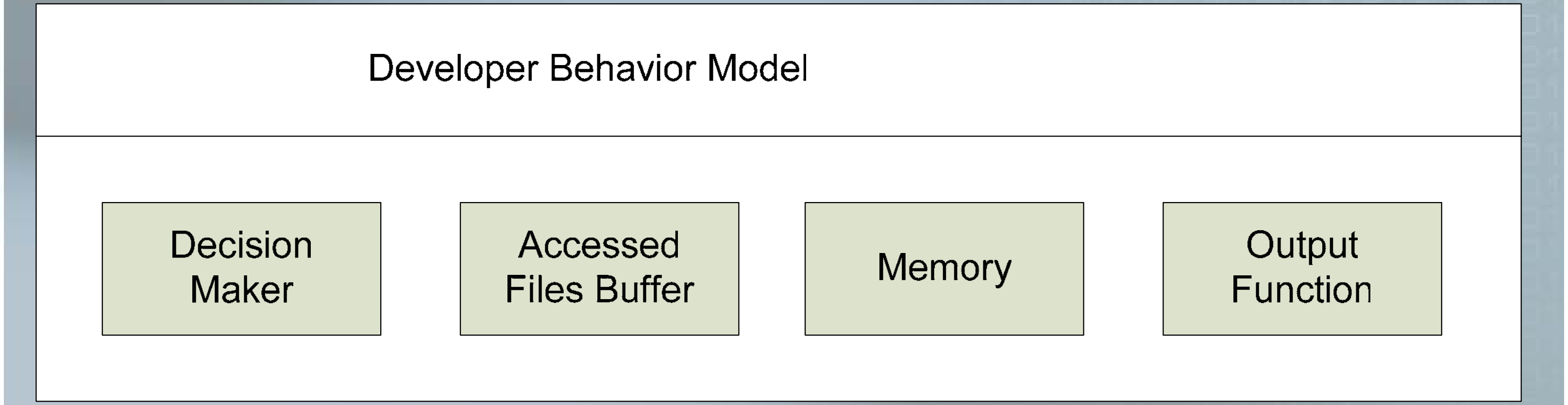
Simulator Architecture



Code Structure Model



Developer Behavior Model



Developer Behavior Model

Decision Maker

Accessed Files Buffer

Memory

Output Function

Simulation Kernel

- State
  - The current element examined
  - Accessed files buffer
  - Developer memory
- Time-base
  - Discrete
  - Ordering of the output investigation transcript encodes the time progression of the simulator

Evaluation

- Simulation runs on two real-world software systems:
  - jHotDraw
  - jEdit
- We expect:
  - Notable difference between random and other levels
  - Advanced: more Cs, less loops, wider breadth
- Demo

Limitations

- Performance & Scalability
  - Initial loading of source model in memory
- Level of sophistication
  - Selection of weight factors
  - Additional layers

Future Work

- Validate with real transcripts
- Add better AI
  - E.g. level of confidence
- Add sophistication to the code structure model
  - E.g. including fields, considering types of relations

Conclusion

- The primary goal was achieved
- Met challenges:
  - Definition of the simulator
  - Quantification of parameters
- Pending challenges:
  - Scalability
  - Smarter termination condition
  - Degree of sophistication

References

[1] M.P. Robillard, W. Coelho, and G.C. Murphy, *How Effective Developers Investigate Source Code: An Exploratory Study*, IEEE Transactions on Software Engineering, November 2004

[2] M.P Robillard and G.C Murphy, *Automatically Inferring Concern Code from Program Investigation Activities*, Proceedings of the 18th International Conference on Automated Software Engineering, pp. 225–234, IEEE Computer Society Press, October 2003.

[3] Rich Ackerman, “Vector Model Information Retrieval”, *Theory of Information Retrieval*, Florida State University, September 2003, URL: http://www.hray.com/5264/math.htm

Questions?