



# PROGRAMMING LANGUAGES LAB



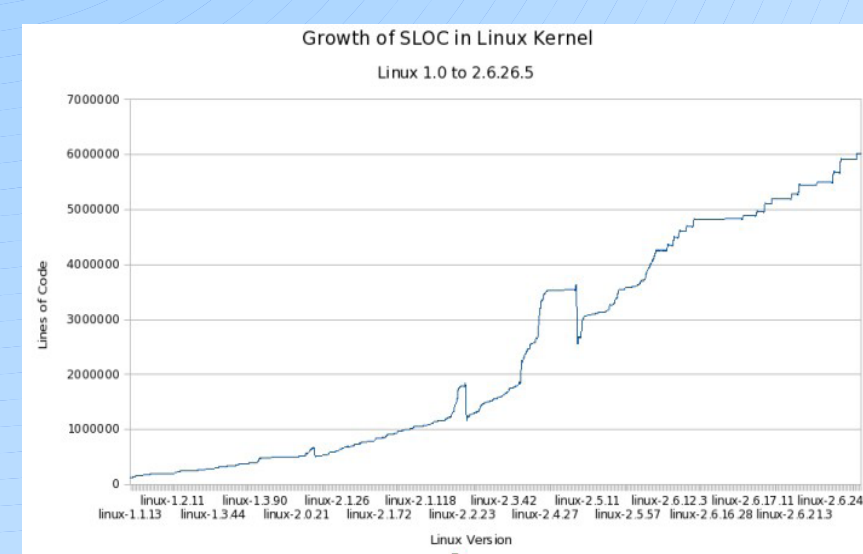
## Lab Members

**Faculty:**  
Dr. Deepak D'Souza  
Dr. K V Raghavan

**Students:**  
Amogh Margoor  
Aravind Acharya N  
Anab De  
Girish M R  
Pranav Modi  
Raghavendra K R  
Sachin Kale  
Vasanta K

## Why is research in programming languages important?

Increasing size and complexity of software

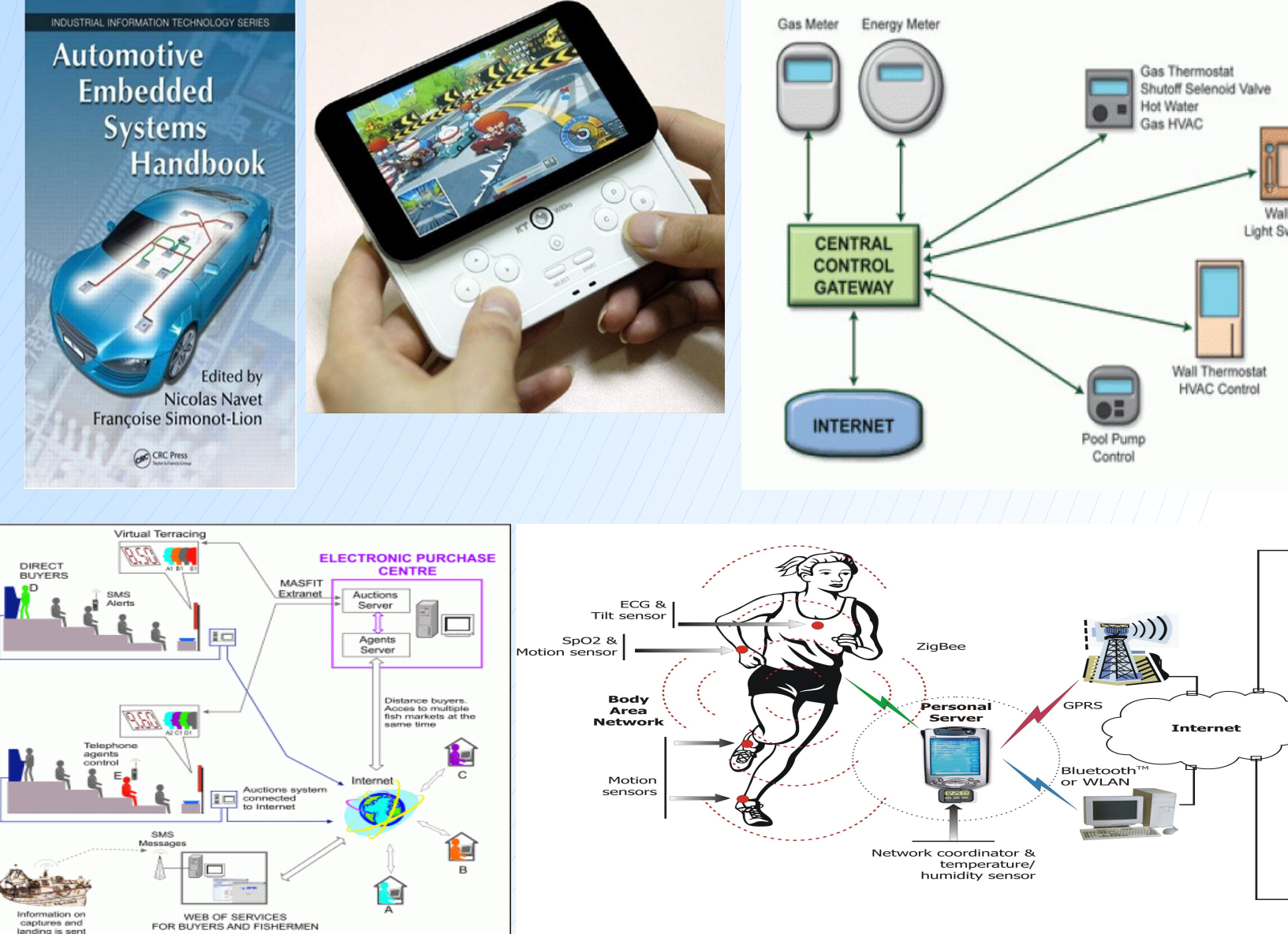


Need for reliable software



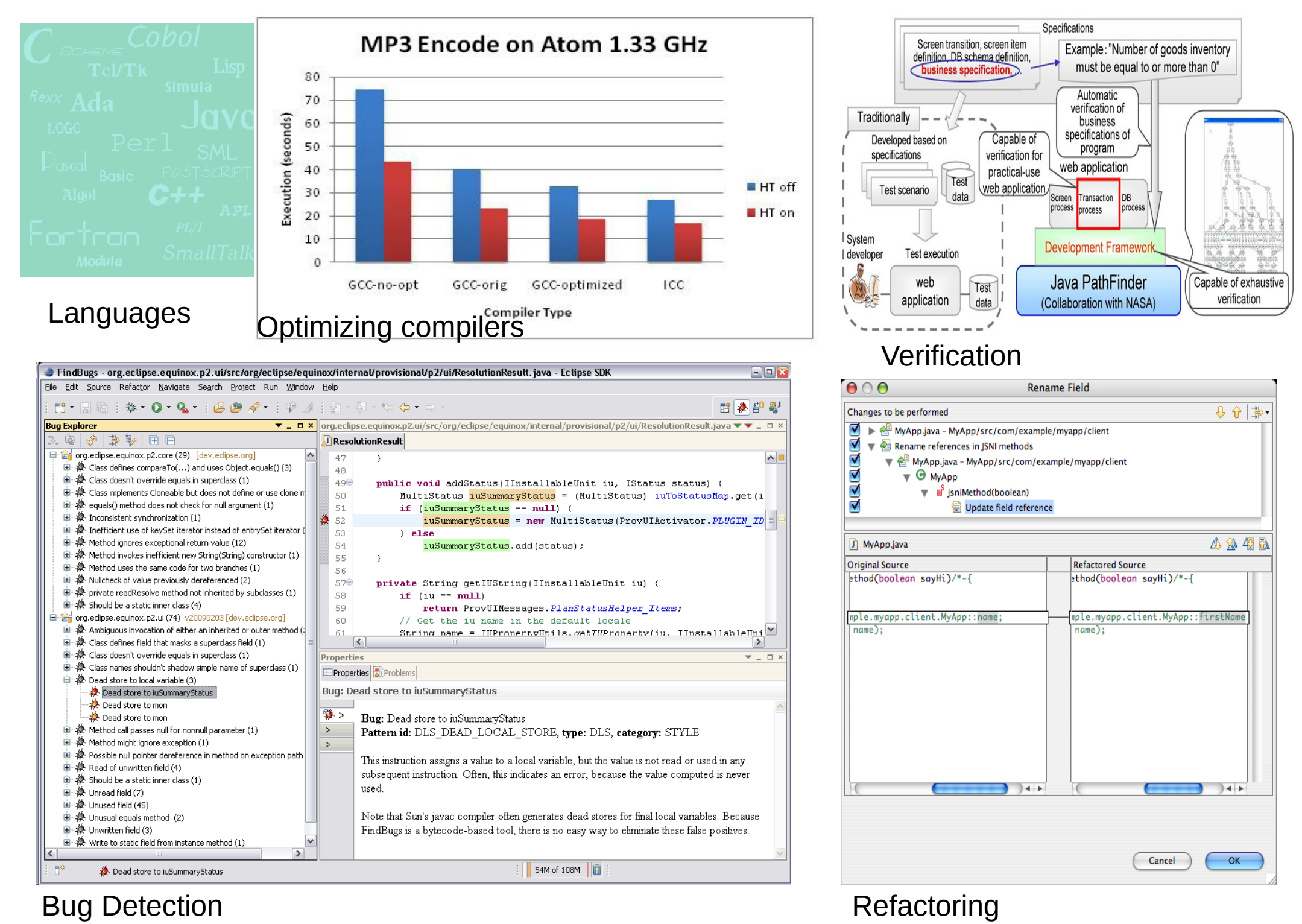
Ariane 5 explosion due to incorrect software

Emerging areas with programmable systems



Five of last ten Turing Awards are for research in PL!

## Notable contributions of PL research

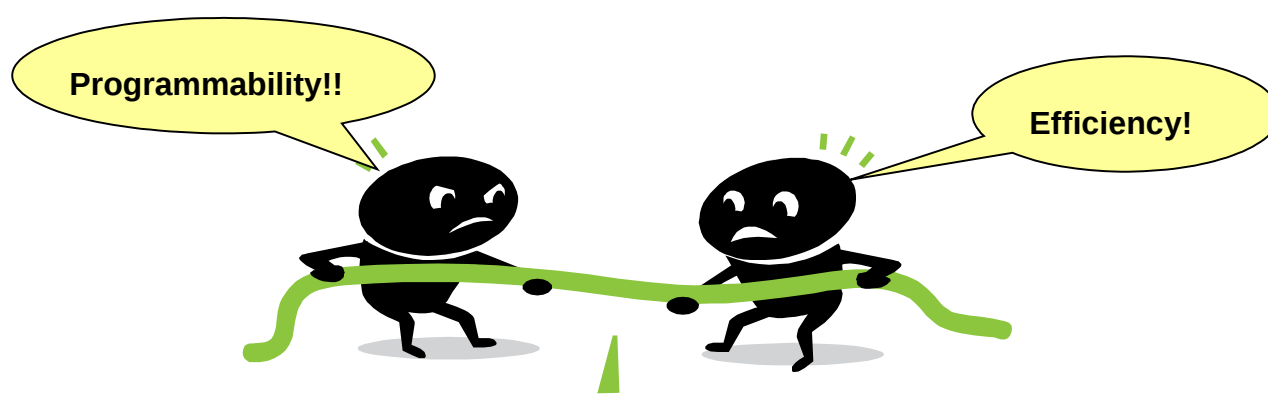


## Analysis of Shared-memory Concurrent Programs

```
x = y = 0;
x = 1; y = 1;
r1 = y; r2 = x;
```

Most processors may produce  $r1 == r2 == 0$ !

- ❖ Reordering needed for performance.
- ❖ May produce unexpected behaviors!



➤ What is the ideal semantics of shared-memory concurrent programs?  
➤ How to verify properties of programs with weak semantics?  
✓ Proposed an easy-to-understand operational semantics that also allows efficient implementation.  
✓ Developed a model-checker based on the semantics.



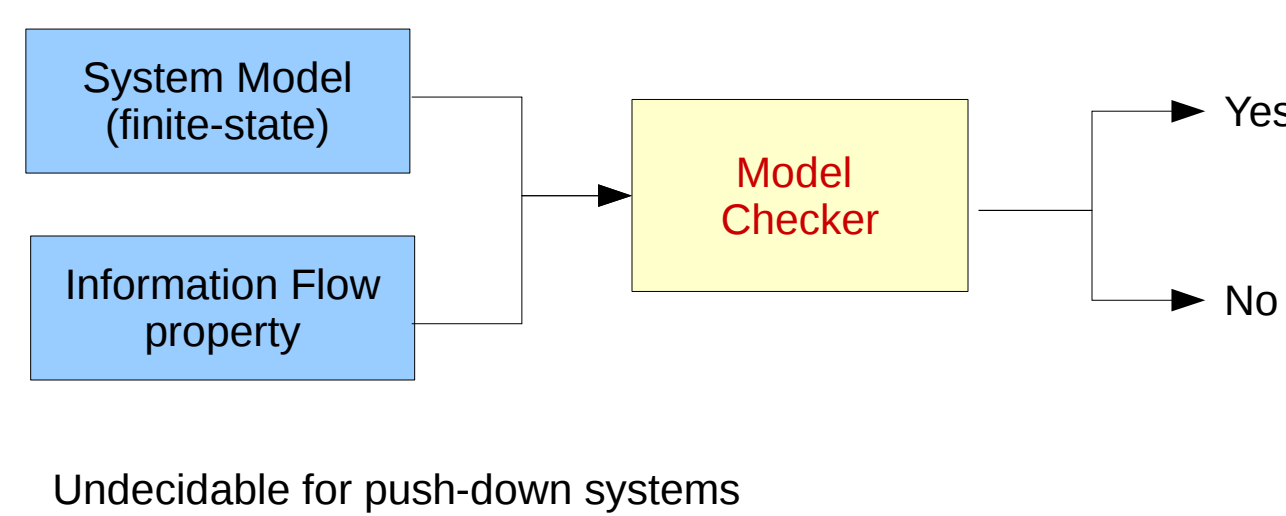
➤ Are well-synchronized programs easy to analyze?  
✓ Goal: Developing a technique for dataflow analysis for well-synchronized programs.

```
p = &a;
lock l; p = NULL; unlock l;
lock l; c = *p; unlock l;
p = &b; unlock l;
```

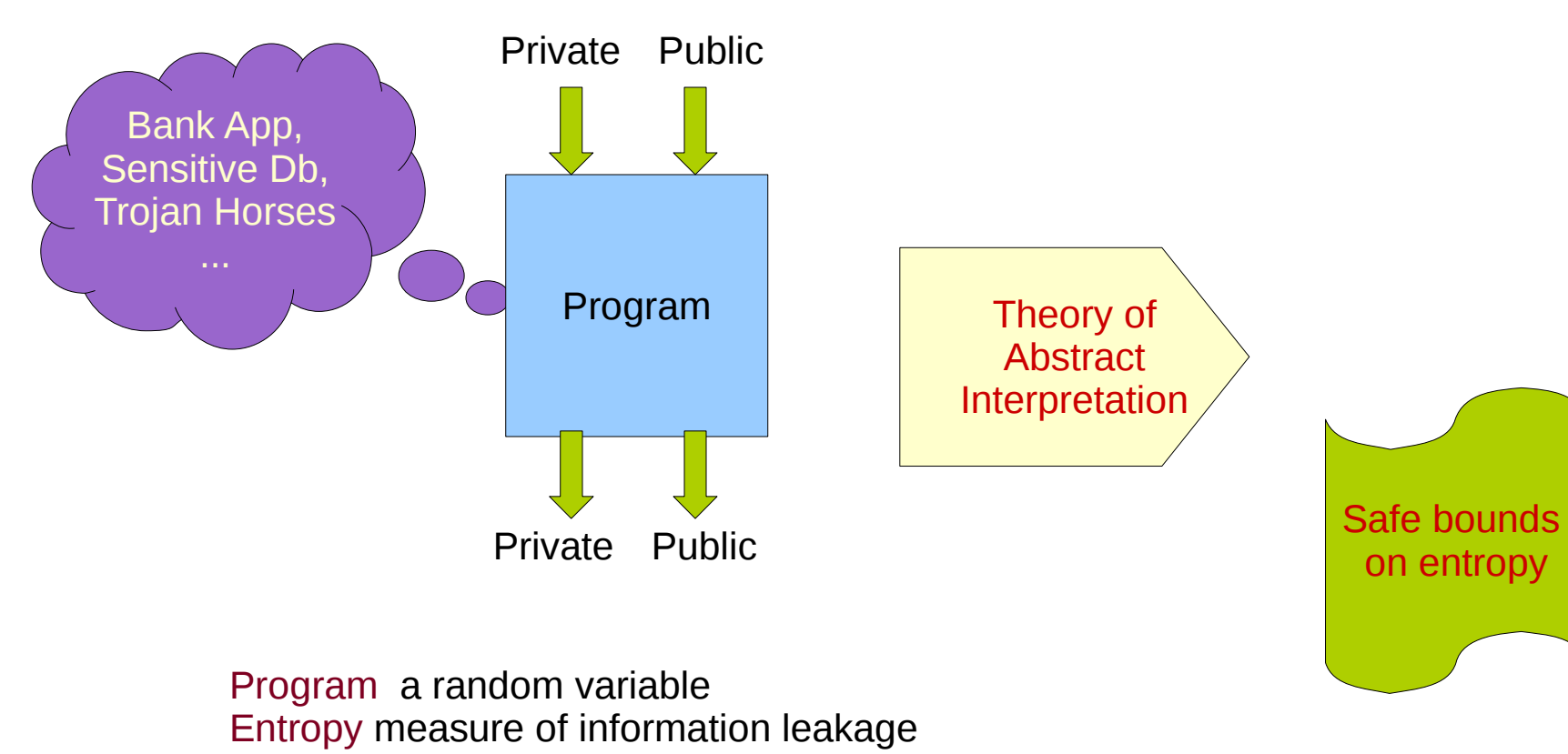
Given the program is well-synchronized, can we prove absence of null-deref efficiently?

## Analyzing Information Security

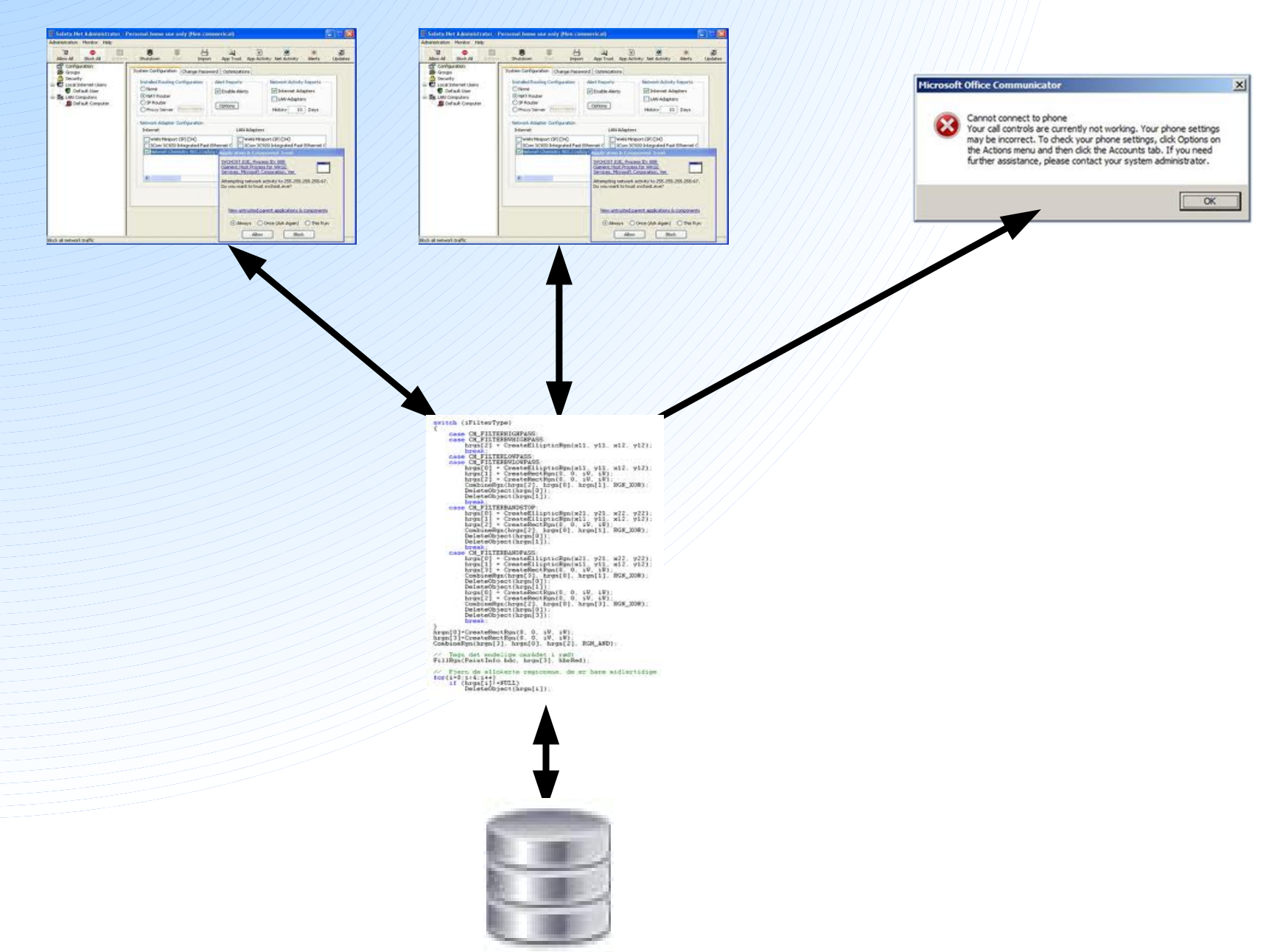
Model Checking Information Flow properties on abstract systems



Quantifying Information Leakage of Programs

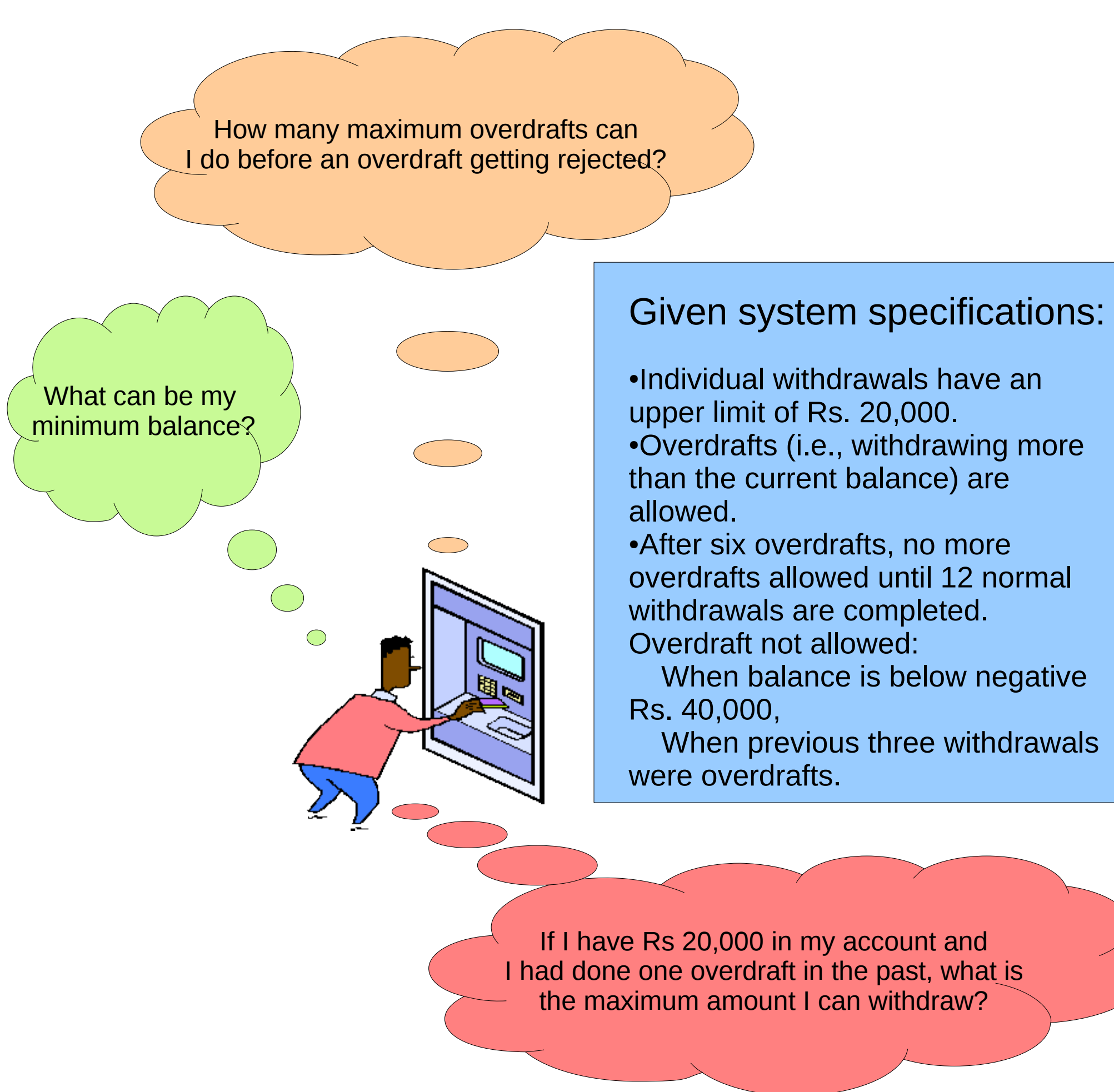


## Automated Testing of Database-Driven Systems



**Goal:** To automatically generate database instances and user inputs so that the program covers all paths within a certain bound.

## Verification of Requirement Specifications

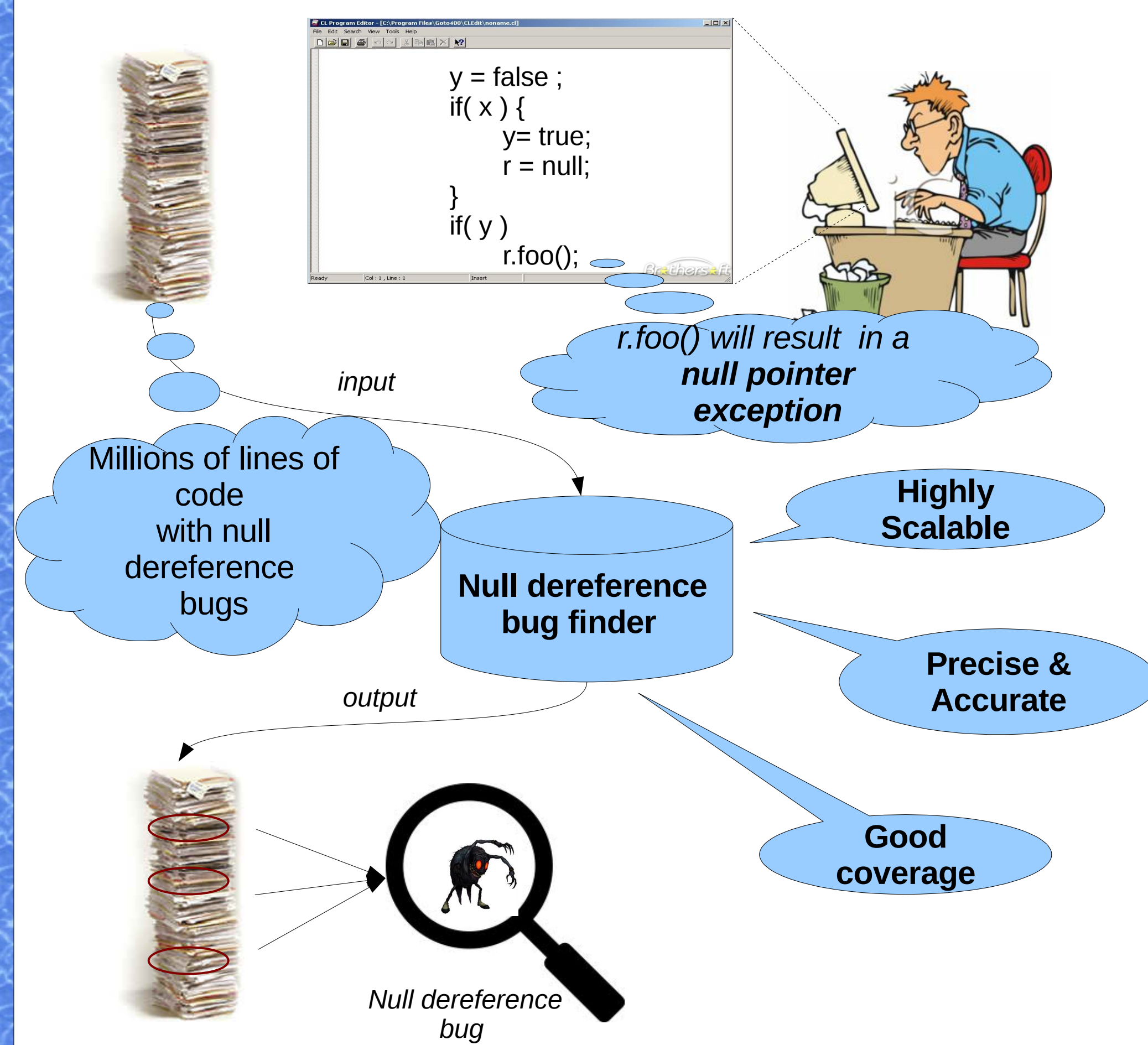


Given system specifications:

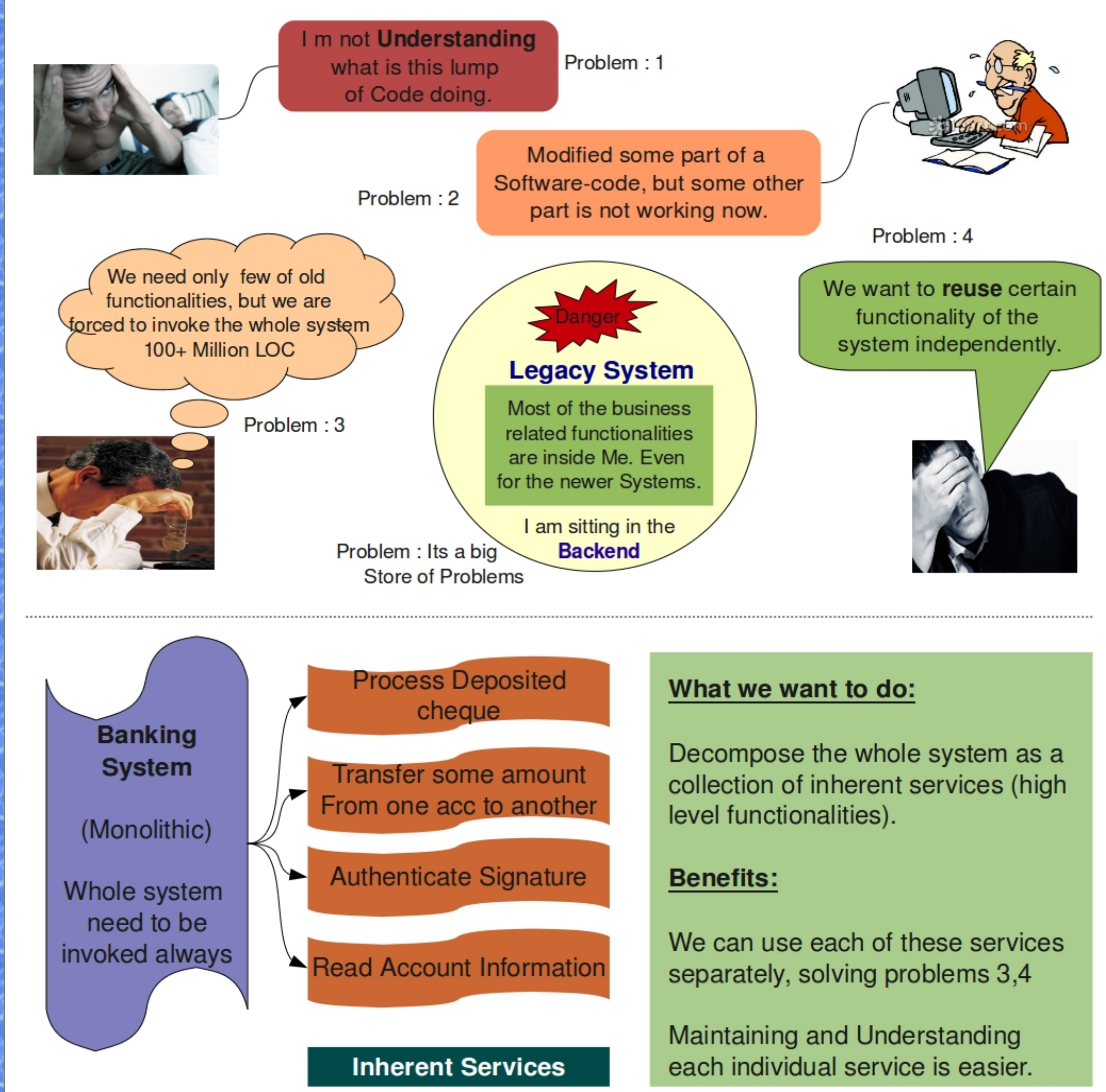
- Individual withdrawals have an upper limit of Rs. 20,000.
- Overdrafts (i.e., withdrawing more than the current balance) are allowed.
- After six overdrafts, no more overdrafts allowed until 12 normal withdrawals are completed.
- Overdraft not allowed:  
When balance is below negative Rs. 40,000,  
When previous three withdrawals were overdrafts.

If I have Rs 20,000 in my account and I had done one overdraft in the past, what is the maximum amount I can withdraw?

## Finding null dereference bugs in Java Programs



## Analysis of Software Systems



## Collaborations and Selected Projects

- ★ A Theory of Conflict-Tolerant Features (with General Motors ISL, Bangalore)
- ★ Constraint Temporal Logics (Indo-France project Timed-DISCOVERI)
- ★ Quantified Information Leakage in Programs
- ★ Java Memory Model Aware Software Validation (with Abhik Roychoudhury, National University of Singapore)
- ★ Mining Services From Monolithic Code (with Infosys)
- ★ Re-architecting legacy financial applications (with Tata Consultancy Services).



## Selected Publications

- ★ Dataflow Analysis for Data-race-Free Programs, Anab De, Deepak D'Souza and Rupesh Nasre, European Symposium on Programming, 2011
- ★ Model-Checking Information Flow Properties, Deepak D' Souza, Raveendra Holla, Raghavendra K. Ramesh and Barbara Sprick, Journal of Computer Security, 2011
- ★ A Case Study in Matching Service Descriptions to Implementations in an Existing System, Hari S Gupta, Deepak D' Souza, K V Raghavan and Girish M Rama, Int. Conference on Software Maintenance, 2010
- ★ Conflict-Tolerant Real-Time Specifications in Metric Temporal Logic, Sumesh Divakaran, Deepak D'Souza and Raj Mohan M., TIME, 2010
- ★ Analysing Message Sequence Graph Specifications, Deepak D'Souza, Joy Chakraborty and K. Narayan Kumar, ISO/IEC, 2010.
- ★ Deep Packet Inspection Using Message Passing Networks, Divya Jain, K. Vasanta Lakshmi and Priti Shankar, Recent Advances in Intrusion Detection, 2008
- ★ Conflict-Tolerant Real-Time Features, Deepak D'Souza, Madhu Gopinathan, S. Ramesh, and Prahlada Varadan Sampath, Int. Conference on the Quantitative Evaluation of Systems, 2008

