A Framework and Methodology for Studying the Causes of Errors in Software Systems

Andrew J. Ko and Brad A. Myers

J. Vis. Lang. Comput. 16, Feb 2005

Summary by Prof. Thomas LaToza

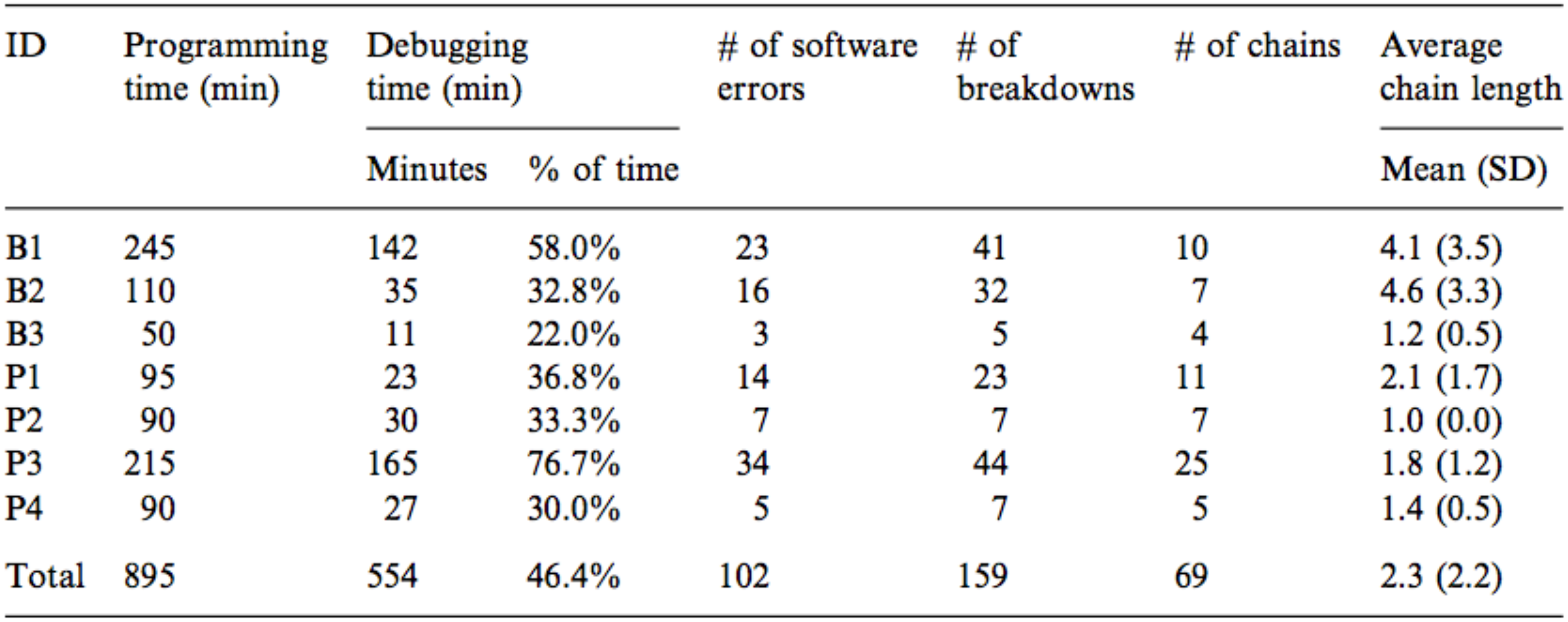
SWE 795, Spring 2017

Software Engineering Environments

What causes defects?

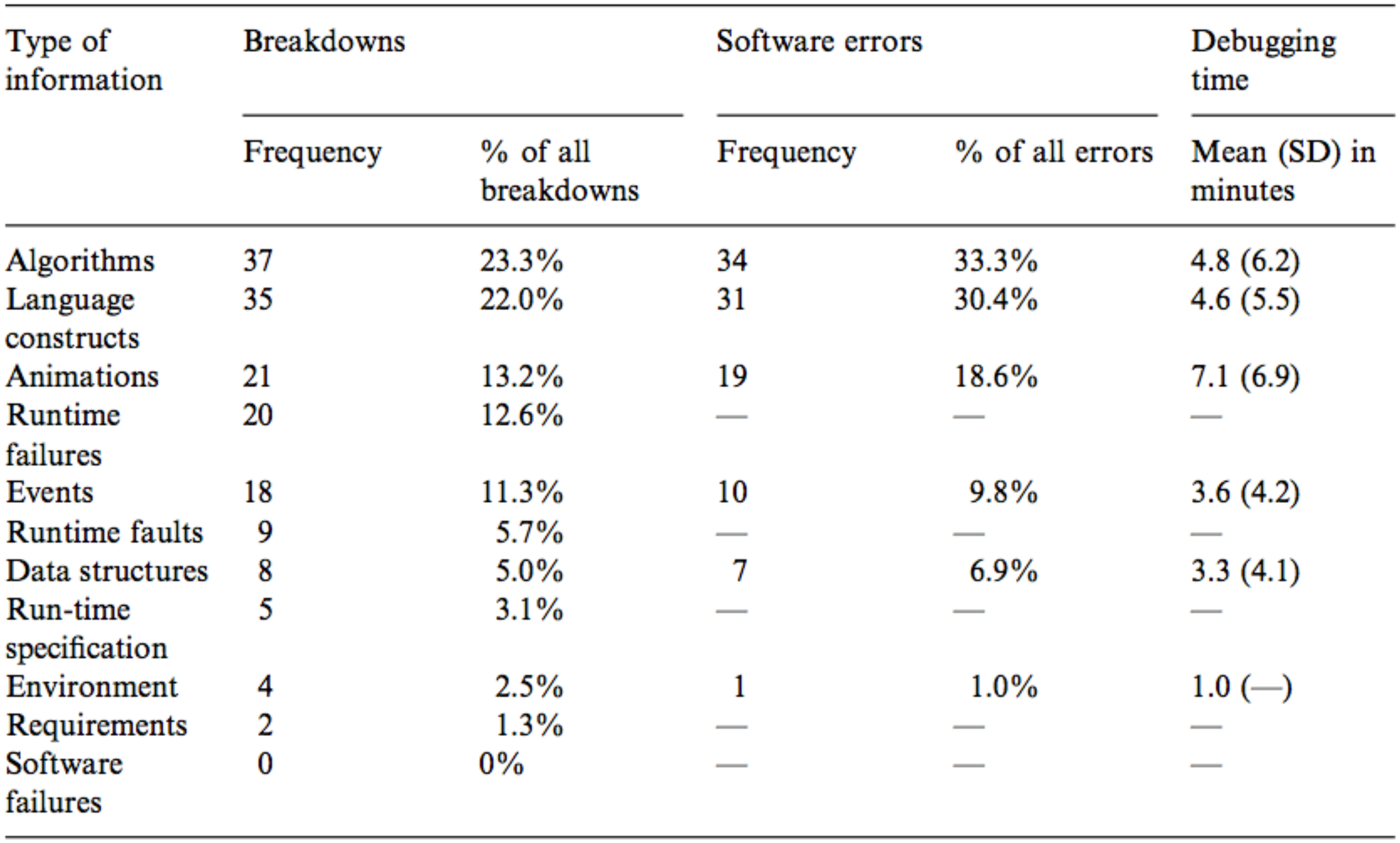
- Conducted a study to examine common breakdowns in programmer's cognition
- Observed 4 *novice* developers for 90 - 215 mins ea. working on tasks with Alice programming environment
- Identified and categorized breakdowns to understand their causes

Debugging time, errors, & breakdowns



ID	Programming time (min)	Debugging time (min)		# of software errors	# of breakdowns	# of chains	Average chain length
		Minutes	% of time				Mean (SD)
B1	245	142	58.0%	23	41	10	4.1 (3.5)
B2	110	35	32.8%	16	32	7	4.6 (3.3)
B3	50	11	22.0%	3	5	4	1.2 (0.5)
P1	95	23	36.8%	14	23	11	2.1 (1.7)
P2	90	30	33.3%	7	7	7	1.0 (0.0)
P3	215	165	76.7%	34	44	25	1.8 (1.2)
P4	90	27	30.0%	5	7	5	1.4 (0.5)
Total	895	554	46.4%	102	159	69	2.3 (2.2)

Frequency of breakdowns by information type



Type of information	Breakdowns		Software errors		Debugging time
	Frequency	% of all breakdowns	Frequency	% of all errors	Mean (SD) in minutes
Algorithms	37	23.3%	34	33.3%	4.8 (6.2)
Language constructs	35	22.0%	31	30.4%	4.6 (5.5)
Animations	21	13.2%	19	18.6%	7.1 (6.9)
Runtime failures	20	12.6%	—	—	—
Events	18	11.3%	10	9.8%	3.6 (4.2)
Runtime faults	9	5.7%	—	—	—
Data structures	8	5.0%	7	6.9%	3.3 (4.1)
Run-time specification	5	3.1%	—	—	—
Environment	4	2.5%	1	1.0%	1.0 (—)
Requirements	2	1.3%	—	—	—
Software failures	0	0%	—	—	—

Some frequent causes of defects

- Breakdowns in in implementing expressions (33%)
  - Often difficult to craft boolean expressions correctly
- Breakdowns in debugging (18%)
  - Often generated only a singled incorrect hypothesis about cause of error (biased reviewing)
- Breakdowns in reuse (7%)
  - Pasted code without adequately adapting it to context

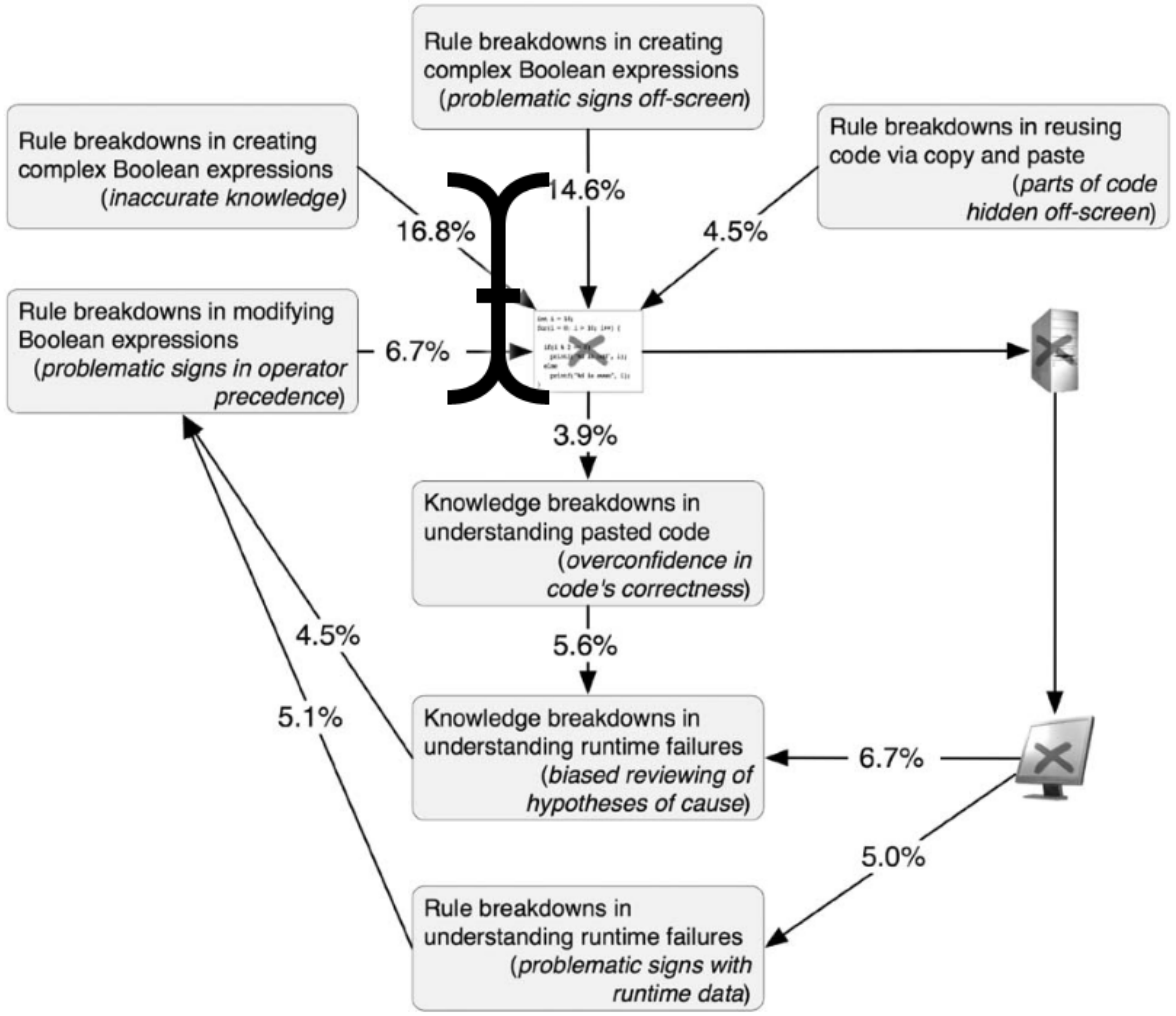

Rule breakdowns in creating complex Boolean expressions (*problematic signs off-screen*)
Rule breakdowns in creating complex Boolean expressions (*inaccurate knowledge*)
Rule breakdowns in reusing code via copy and paste (*parts of code hidden off-screen*)
14.6%
16.8%
4.5%
Rule breakdowns in modifying Boolean expressions (*problematic signs in operator precedence*)
6.7%
3.9%
Knowledge breakdowns in understanding pasted code (*overconfidence in code's correctness*)
5.6%
4.5%
5.1%
Knowledge breakdowns in understanding runtime failures (*biased reviewing of hypotheses of cause*)
6.7%
5.0%
Rule breakdowns in understanding runtime failures (*problematic signs with runtime data*)

Fig. 9. A model of the major causes of software errors in Alice during programmers' tasks. Each line represents a particular type of causal link between one type of breakdown and another, where the number on the line represents the proportion of the particular type of link out of all links in all chains. Note that we do not include numbers for the links between software errors, runtime faults, and runtime failures, since we only recorded software errors that led to runtime failures.

Questions for discussion

- Overall reaction to the paper
- How might these results differ for professional developers?
- How useful are these results in understanding causes of defects?
- What are the design implications for tools from these findings?