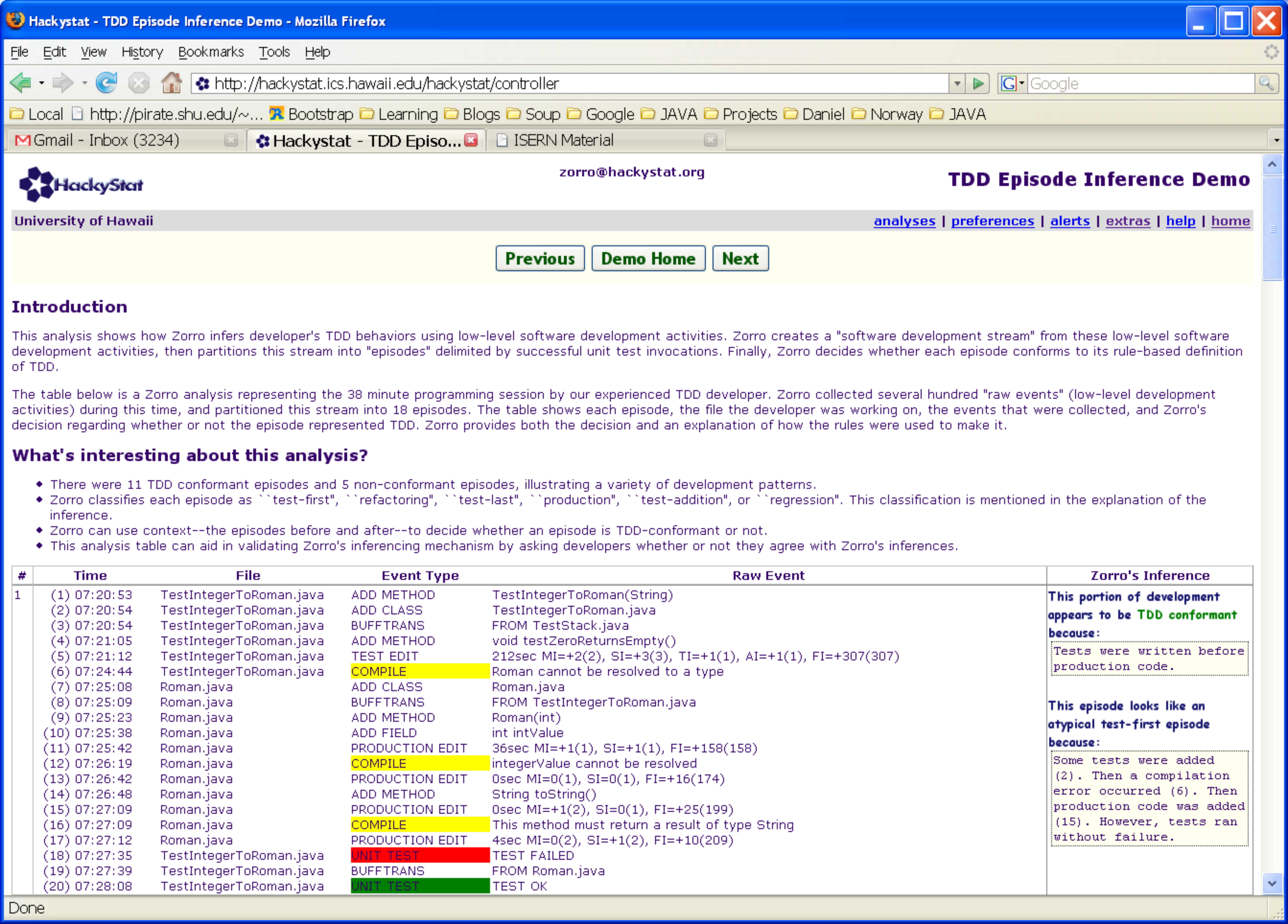

Hackystat - TDD Episode Inference Demo - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://hackystat.ics.hawaii.edu/hackystat/controller

Local http://pirate.shu.edu/~... Bootstrap Learning Blogs Soup Google JAVA Projects Daniel Norway JAVA

Gmail - Inbox (3234) | Hackystat - TDD Episo... | ISERN Material

HackyStat zorro@hackystat.org **TDD Episode Inference Demo**

University of Hawaii analyses | preferences | alerts | extras | help | home

Previous | Demo Home | Next

Introduction

This analysis shows how Zorro infers developer's TDD behaviors using low-level software development activities. Zorro creates a "software development stream" from these low-level software development activities, then partitions this stream into "episodes" delimited by successful unit test invocations. Finally, Zorro decides whether each episode conforms to its rule-based definition of TDD.

The table below is a Zorro analysis representing the 38 minute programming session by our experienced TDD developer. Zorro collected several hundred "raw events" (low-level development activities) during this time, and partitioned this stream into 18 episodes. The table shows each episode, the file the developer was working on, the events that were collected, and Zorro's decision regarding whether or not the episode represented TDD. Zorro provides both the decision and an explanation of how the rules were used to make it.

What's interesting about this analysis?

- There were 11 TDD conformant episodes and 5 non-conformant episodes, illustrating a variety of development patterns.
- Zorro classifies each episode as ``test-first", ``refactoring", ``test-last", ``production", ``test-addition", or ``regression". This classification is mentioned in the explanation of the inference.
- Zorro can use context--the episodes before and after--to decide whether an episode is TDD-conformant or not.
- This analysis table can aid in validating Zorro's inferencing mechanism by asking developers whether or not they agree with Zorro's inferences.

#	Time	File	Event Type	Raw Event	Zorro's Inference
1	(1) 07:20:53	TestIntegerToRoman.java	ADD METHOD	TestIntegerToRoman(String)	**This portion of development appears to be TDD conformant because:** Tests were written before production code. **This episode looks like an atypical test-first episode because:** Some tests were added (2). Then a compilation error occurred (6). Then production code was added (15). However, tests ran without failure.
	(2) 07:20:54	TestIntegerToRoman.java	ADD CLASS	TestIntegerToRoman.java	
	(3) 07:20:54	TestIntegerToRoman.java	BUFFTRANS	FROM TestStack.java	
	(4) 07:21:05	TestIntegerToRoman.java	ADD METHOD	void testZeroReturnsEmpty()	
	(5) 07:21:12	TestIntegerToRoman.java	TEST EDIT	212sec MI=+2(2), SI=+3(3), TI=+1(1), AI=+1(1), FI=+307(307)	
	(6) 07:24:44	TestIntegerToRoman.java	COMPILE	Roman cannot be resolved to a type	
	(7) 07:25:08	Roman.java	ADD CLASS	Roman.java	
	(8) 07:25:09	Roman.java	BUFFTRANS	FROM TestIntegerToRoman.java	
	(9) 07:25:23	Roman.java	ADD METHOD	Roman(int)	
	(10) 07:25:38	Roman.java	ADD FIELD	int intValue	
	(11) 07:25:42	Roman.java	PRODUCTION EDIT	36sec MI=+1(1), SI=+1(1), FI=+158(158)	
	(12) 07:26:19	Roman.java	COMPILE	integerValue cannot be resolved	
	(13) 07:26:42	Roman.java	PRODUCTION EDIT	0sec MI=0(1), SI=0(1), FI=+16(174)	
	(14) 07:26:48	Roman.java	ADD METHOD	String toString()	
	(15) 07:27:09	Roman.java	PRODUCTION EDIT	0sec MI=+1(2), SI=0(1), FI=+25(199)	
	(16) 07:27:09	Roman.java	COMPILE	This method must return a result of type String	
	(17) 07:27:12	Roman.java	PRODUCTION EDIT	4sec MI=0(2), SI=+1(2), FI=+10(209)	
	(18) 07:27:35	TestIntegerToRoman.java	UNIT TEST	TEST FAILED	
	(19) 07:27:39	TestIntegerToRoman.java	BUFFTRANS	FROM Roman.java	
	(20) 07:28:08	TestIntegerToRoman.java	UNIT TEST	TEST OK	

Done