

Conference Report

Human Centric Computing Languages and Environments (HCC) 2003

Tim Wright

Auckland, New Zealand, 28-31 October, 2003

In 2003, HCC was held in the southern hemisphere for the first time—in Auckland, New Zealand. Like last year, I stayed in a backpackers, and unlike last year my luggage arrived on time. Other people were not so fortunate: due to the fires in California, at least four people arrived without their luggage.

The conference was 4 days long with a workshop on the first day. There were two keynote presentations, by a Microsoft researcher and an architect. Unfortunately, the conference was a two-stream conference so I could not see all presentations, but my presentation was well recieved.

Keynote Presentations

There were two keynote presentations:

Mary Czerwinski from Microsoft Research talked about two research areas: the data mountain and the task gallery. She said that there will be a version of the data mountain in the next version of Windows. The task gallery is their replacement for the start bar. They found that users get a 15 to 30% increase in productivity by simply using a multiple monitor machine, but the Windows start bar does not scale well over two screens. Other research she performed found that people consider large displays more public than small displays.

Murk Burry from RMIT University closed the conference with Antoni Gaudi's computational legacy. He has been using aeronautical design

software to complete the Sagrada Familia Church — conventional architectural software is not powerful enough.

Papers

Many papers at the conference were related to my research. Highlights were two descriptions of programming environments for children: *Icicle* and *Viscuit*. An interesting new concept was End User Software Engineering, and Andrey Ko presented a model of programming errors.

Icicle: Robert Sheehan spoke about parallism in *Icicle*. *Icicle* is a programming by demonstration system in a visual simulation domain. Users specify before and after rules. *Icicle* differs from existing systems, like Stagecast, because it has an animated representation.

Robert argued that there are two types of parallism: intra-object parallism and inter-object parallism. *Icicle* provides intra-object parallism by letting users drop rules on top of each other, and animates the resulting compound rule. *Icicle* also can fire multiple rules simtaneously, providing inter-object parallism.

Viscuit is another programming by demonstration system for visual simulations using before and after rewrite rules. *Viscuit* differs because it uses fuzzy rule rewriting logic, where the first half of a rule will match if the objects are not the correct rotation or distance from each other. *Viscuit* examines how close the match is, how the match

differs from the original rule, and modifies the after rule accordingly.

WYSIWYT: What you see is what you test

End User Software Engineering: trying to get end-users to use some software engineering tricks.

Model of Programming Errors

My Presentation

My presentation was on the second day of papers. It was recieved well.

There was time for two questions. One person asked for a clarification, and the other wondered if my taxonomy could be extended to cover modelling langauges as well as programming environments. I believe it can, and think a couple of paragraphs about this will be a worthwhile addition to my thesis.

After my presentation many people came and talked to me. These discussions ranged from people sharing their experiance building programming environments. One person even said that they had the problems I was talking about, and my presentation has given them the vocabulary to explain the problems to others. Another person wanted a copy of the presentation. I have since emailed it to them.

Conclusion

It was a very good conference that I enjoyed throughly. I spoke to many interesting people, and they seemed to wnjoy my research.