TC-ECBS newsletter

October 2006 Issue

TC-ECBS newsletter is produced as a focal point for news concerning the (related) activities of the IEEE Computer Society Engineering of Computer-Based Systems technical committee. If you have any items for the next issue please contact **ecbs-news@computer.org**.

Forthcoming: ECBS 2007

Raising the Expectations of Computer-Based Systems

At Doubletree Hotel Tucson at Reid Park in Tucson, Arizona, U.S.A March 26th-29th, 2007.

The 14th Annual ECBS will be hosted by the University of Arizona in Tucson, AZ U.S.A March 26th-29th, 2007.

The conference website is available at http://www.ece.arizona.edu/~ecbs07/

What is the Engineering of Computer Based Systems? The emerging discipline of ECBS is devoted to the design, development, deployment, and analysis of complex systems whose behaviour is largely determined or controlled by computers. CBSs are characterized by functional, performance, and reliability requirements that mandate the tight integration of information processing and physical processes. ECBS encompasses many facets: system modelling, requirements specification, simulation, architectures, communications, safety, security, reliability, software, hardware, human computer interfacing, system integration, verification and validation and project management.

ECBS is the integration of several engineering disciplines including software, hardware, and communications into a complete systems engineering discipline. The conference provides a bridge between industry and academia; the program will provide a balanced view of academic research and industrial developments.

SCOPE: Contributions are sought in two major categories: advances in fundamental ECBS technologies and reports of solutions that further ECBS practice in application domains. The meeting's theme represents the increasing expectations of stakeholders upon computer-based systems in terms of performance, security, reliability, safety, usability. As builders and researchers of CBSs, are we achieving these expectations? What are we doing well, where do we need to improve?

One or more of the topics listed below should be reflected in the submitted papers - whereas the main focus should be the engineering of computer-based systems.

o Architectures

o Autonomic Systems

o Codesign

o Component-Based System Design

o Design Evolution

o Distributed Systems Design

o ECBS Infrastructure (Tools, Environments)

o Education & Training

o Embedded RealTime Software Systems

o Lifecycle Processes and Process Evolution

o Integration Engineering

o Model-Based System Development

o Modeling and Analysis of Complex Systems

o Open Systems

o Product-Families Models and Architectures

o Reengineering & Reuse

o Reliability, Safety, Dependability, Security

o Requirements Elicitation and Analysis

o Standards

o System on a Chip

o System Assessment, Testing and Metrics

o Verification & Validation

Reports of practical solutions, trends and new system characteristics for ECBSs may include application domains such as: Aerospace Systems, Command and Control, Continuous and Discrete Manufacturing, Environmental Systems, Instrumentation and Control Applications, Internet Technology and Applications, Intelligent Highway-Vehicle Systems (IHVS), Medical Systems, Telecommunication.

WORKSHOPS:

ECBS 2007 will feature the following workshops:

1. Domain-Specific Approaches to Model-Based Development

http://www.theoinf.tu-ilmenau.de/~riebisch/mbd/

2. Engineering of Autonomic & Autonomous Systems

www.ulster.ac.uk/ease/

3. Embedded Systems Worskshop: theory and practice

The workshops include short presentations and a forum for focused discussions. Traditionally, at ECBS conferences, workshops are strongly connected to the conference sessions to enable continued in-depth discussions.

POSTER SESSIONS: Posters and abstracts presenting work in progress are invited for a poster session. Accepted abstracts will be published in the proceedings. Graduate students are especially welcome to participate.

DOCTORAL SYMPOSIUM: The Doctoral Symposium provides a forum for PhD candidates to present work in progress and get feedback from the research community.

INDUSTRIAL TRACK: The industrial track provides a forum for short papers on results of industrial research and development.

SUBMISSION OF PAPERS: Full papers and Doctoral Symposium papers must be 6 to max 10 pages in Computer Society Format, poster abstracts 1 to 2 pages and industrial track papers 2 to 4 pages.

Submission of papers due:

10 November, 2006

Submission of doctoral symposium paper:

15 November, 2006

Submission of poster abstracts due:

15 November, 2006

Submission of industrial track papers due:

15 November, 2006

Notification:

11 December, 2006

Camera-ready version due (if accepted):

06 January, 2007

Early registration deadline:

26 February, 2007

Submission dates for workshop papers may differ, see separate Calls for Papers on the workshop websites.

General Chair. Jerzy W. Rozenblit (ecbs07@ece.arizona.edu)

Program Co-Chairs: John Leaney, Tim O'Neill, Jianfeng Peng

Europe Co-Chairs: Matthias Riebisch, Peter Tabeling; Asia Pacific Co-Chairs: Tim O'Neill, John Leaney.; Publicity Chair: Brian Ten Eyck; Local Arrangements Chair: Rozanne Canizales

STEERING COMMITTEE

Jerzy Rozenblit, Jianfeng Peng, John Leaney, Matthias Riebisch, Peter Tabeling , Stephanie White, Byron Purves, Roy Sterritt, Jonah Lavi, Miroslav Sveda, Wilhelm Rossak.

TC-ECBS newsletter

THINKING ABOUT SYSTEMS

By **Bryon Purves**

Byron Purves, TC ECBS chair, writes about his experiences as a System Engineer.

As a teenager I wanted to be a chemist. I had my laboratory in a shed in the back garden. But when it came time to go to university, due a comedy of errors, I found myself, without much choice, a student in Physics and Mathematics. And the mathematics subject included what, at the time, was called Numerical Mathematics. But it was really Computer Science in disguise. So in 1964 I wrote my first program, to calculate the prime numbers, and in 1966, forty years ago, I was a member of the first graduating class in England with a degree that included Computer Science. That little comedy of errors was one of those lucky breaks of fate, and it led me into what has been, so far, a career that has included a sequence of very interesting projects.

Eventually I got my Ph.D. in Physics. My thesis was in automatic speech recognition, and my coursework included, in addition to Physics, a collection of electrical and mechanical engineering. In retrospect, it would have been difficult to have chosen a more practical preparation for work in computer based systems.

In 1973 I joined The Boeing Company as an engineer in the Boeing Commercial Airplane Company, in the Propulsion and Noise staff. This was in the recovery period after a period of very, very deep layoffs. In the previous few years a very high proportion of the engineering work force had lost their jobs. Even though I had been assigned to a research unit, it was not easy to understand the environment. I was really overwhelmed with the complexity of industrial operations. I felt as though I would never understand how things worked. After an initial period of adjustment, my first assignment was to write a program to optimize the design of jet engine noise mufflers. I wrote it, (this is Fortran IV on punched cards) and it seemed to work reasonably well. But now I learned a very important lesson. I tested the software, and distributed it to the analysts. But, by mistake, I distributed an old version. Somehow I got things mixed up. With great embarrassment I recalled the wrong versions and replaced them. That was embarrassing.

So one of the first lessons I learned in industry was to control the configuration of my product.

I had a good time in Boeing Commercial Airplane Company, alternating between periods in which I developed analytical tools in a research group and periods in which I applied the new methods working in airplane programs. That was a wonderfully valuable arrangement, and I recommend it to anyone to arrange to work that way. Working in the airplane programs, I had to understand what was needed to make a commercial product, while my time in the research organization gave me the opportunity to keep up with the state of the art, and sometimes to advance it. So another lesson I learned early was the need for domain knowledge.

Unless you understand the application domain you are unlikely to do much useful. Actually you are likely to be solving the wrong problem, even if you do it well.

Something else I realized in this period was that a new problem did not always require a completely new solution. For part of this time I was interested in designing mufflers for jet engines. The driving problem was the community noise level around airports, and we had noise regulations to meet. From a computing standpoint, we needed to solve some difficult partial differential equations, and we needed numerical methods to do it. There was a huge amount of prior work in those areas, so it was possible to re-apply older methods in this new problem domain. These days we call it technology transfer.

So another useful lesson is to take advantage of the work of others. Practice technology transfer, or component re-use

To quote Sir Isaac Newton:

I have only stood on the shoulders of giants

We do not always need a new invention. The old invention might go a long way to solving our new problem. But this means that we have to put a lot of energy into understanding and knowing about those older methods. I suppose that formal education ought to be able to contribute to that. Professors take note. I don't know if it is still true, but in England we learned our science by following a historical approach, generally repeating the experiments that threw out the older theories.

So we also have to be willing to throw away the past.

After working in the noise reduction area for six years I came to the conclusion that everything I did to make the aircraft quieter, also made it heavier and reduced its range and payload. So we noise reducers were always in tension with those whose job it was to increase range and payload other positive performance features. This encouraged me to move to another part of the company. One of the really good things about working for a company like Boeing is its diversity. I went to work for Boeing Engineering and Construction Company (now long since gone out of business), to build control systems for electric utilities. This is when I really started to work at a system level. Our division delivered systems (hardware and software) for control of electric utility generation and transmission. The company sent me to school for a couple of weeks of intensive training, and so I became a power control systems engineer. The hardest part was learning the new alphabet soup, all of those abbreviations and acronyms. Every engineering area (and non-engineering area for that matter) has its own abbreviations and acronyms. They might even look the same as another area but mean something completely different. Whenever I switch to a new application domain, even though I find the underlying system principles to be the same, the language to describe them is different. The theoretical aspects of power systems were not terribly difficult. I was a physicist so I at least understood the theory of electricity and magnetism, and I had a background in numerical methods and software. Actually that put me in a better position than many electrical engineers. The tricky bit was to get a feel for alternating current. I remember being completely stumped when the power system engineers talked about have a generator sucking in vars. After a few months of working around the edges, my first real assignment was to respond to a Request for Proposal from Consolidated Edison in New York City.

Con Edison had written requirements for a system to automatically black out parts of New York City in order to prevent a total city blackout. I analyzed their requirements and went to New York to discuss my findings. I spent the next three days in a room in Manhattan with a dozen or so experienced system operators. These were the subject matter experts, or SMEs, as we call them. The first challenge was to explain what they had actually said they wanted. And they were surprised. The overwhelming response was, "Oh No, No! That's not really what we want." They had hired a consultant to write the specification for them, and that consultant had not really understood the need. So those next three days were spent eliciting the real requirements from the SMEs, as they understood them. The room was ideally suited for the purpose. There were blackboards around three sides of the room, and I used all of it. I used traditional flow chart diagramming to document what they wanted to happen. There was lots of heated discussion: New Yorkers seem to enjoy arguing, but we really did discover the requirements. There were several enablers here:

It would have been impossible to elicit the requirements just by writing down the words. Drawing pictures allowed more rapid agreement among the SMEs. *(Continued in next Issue of the TC-ECBS Newsletter).*

Your TC-ECBS EXCOM Contacts

Byron Purves TC chair
byron.purves@boeing.com

John Leaney, TC vice-chair
john.leaney@uts.edu.au

Tim O'Neill, TC recording secretary
Tim.ONeill@uts.edu.au

Roy Sterritt, TC 2nd vice-chair
r.sterritt@ulster.ac.uk

Join TC ECBS

Details can be found at the web site http://www.computer.org/tab/ where you can sign-up to TC-ECBS and three others free http://www.computer.org/TCsignup/index.htm