

# Travel Report from RTCSA 2000 in Cheju Island, South Korea

Anders Wall and Markus Lindgren  
Mälardalen Real-Time research Centre  
Department of Computer Engineering  
Mälardalen University, VÄSTERÅS  
{[awl](mailto:awl@mdh.se), [mle](mailto:mle@mdh.se)}@mdh.se

## Introduction

We went to Korea and RTCSA 2000 in order to present three papers. Sweden was well represented with 10 papers out of 70, including six papers from MRTC in Västerås. The conference was located at Cheju island and lasted for three days. Before the start of RTCSA there was a Sweden-Korean workshop at the same location, lasting for two days

## Cheju Island

Cheju is the biggest island in Korea with approx. 500 000 inhabitants. It is located south from the mainland in the Yellow Sea. Cheju is most famous for being a resort where Korean honeymooners spend their vacation. It was a very interesting place to visit as the culture and the food was rather different from what we are used to. Even though tourism is one of the island main sources of income, it is almost not exploited at all.

Our slightly different appearance, compared to Koreans, seemed to amuse most of the local inhabitants, since many of them started smiling and giggling as soon they spotted us. Quite amusingly, it was equally easy to make ourselves understood when speaking Swedish. (Perhaps they all take Swedish classes?)

## Sweden-Korean workshop

The purpose of the workshop was for Sweden and Korean researchers to get together and find areas for possible collaborations, but also to simply get to know the "other" sides work.

The workshop lasted over one and a half-day, with about equal number of presentations from each country. (The half day that's "missing" was spent on a short sight seeing tour of Cheju Island.)

Each presentation was allotted 30 minutes (including time for questions). The Korean researchers had a quite strong focus on reducing power consumption in embedded systems; an issue which today is very important and probably will be so for quite some time. They attack this problem by both software and hardware tricks. In my opinion the hardware approach is the best.

## Talks

One talk was on a special device for measuring the power consumption. The device was able to measure the power consumption between two different (processor, RAM, etc.) states with only two or three sample points (others require many more).

Sheanyun Lee had performed an investigation of how "eor" instructions with different input data consume differing amounts of power. In essence the hamming distance between consecutive instructions decides the power consumption. (Rather low-level work and it seems quite hard to make significant changes to power consumption by optimizing the instruction layout from a compiler, which seemed to be the goal of this work.) Andreas Ermedahl who is currently a visiting researcher at Seoul National University was co-author of the paper.

One talk was on modifying frequency and voltage to the processor while still being able to make some predictions on real-time performance. (Basically the same result as was previously published in RTSS'98.) In

my opinion this seems to be a better approach towards reducing power consumption. Quite surprisingly they did not seem to be supported by industry (at least not Sheayun's work).

From the Swedish side execution time analysis dominated (Jakob, Ebbe, och Jan G). Hans reported on our effort on moving towards reliability estimates rather than timing guarantees. There was also one talk from Lund on their efforts on making Java suitable for embedded systems.

### ***Conclusion/Result***

Besides a greater insight into each others work, I don't think this workshop will have any immediate effects. (Perhaps Hans has some more input on this topic?)

## **RTCSA 2000**

The conference can be concluded as “yet another conference on real-time systems”, thus the community still has its focus on guaranteeing temporal behavior, i.e. scheduling. Nevertheless, some indications on a shift of focus could be perceived, if not in the proceedings so in the keynote speech given by Al Mok from University of Texas, Austin. He hinted at that we have to investigate how to design- and maintain systems. In my opinion, he touched on a very important issue although he ended up falling into the “scheduling-pit” again. His main point was that we must schedule systems such as they are *robust* to changes in execution times and frequency. Such changes can, for instance, arise from changes to a more effective hardware. Another example is when new functionality is to be added and we need to release some utilization from the system for that by decreasing the frequency of some of the tasks that already exist in the system. Moreover, the same conclusion regarding trends could be drawn from some of the informal discussions we had offline.

The trend that has been going on for a while now, namely soft real-time systems, was continued. A lot of people are targeting consumer applications such as multimedia, and personal communication.