

Developer Conference

August 11 2007

Introduction to Resource Oriented Computing
using NetKernel



Dr. Peter. J. Rodgers
Co-founder of 1060 Research
and co-architect of NetKernel

Venue Hotel Green Park,
Greenlands, Begumpet,
Hyderabad

Schedule 11:00 AM – 1:00 PM Introduction to ROC
 1:00 PM – 2:00 PM Lunch Break
 2:00 PM – 4:00 PM Discussions

Reservation:

Send an email to hyderabad2007@1060research.com

or register at <http://1060research.com/hyderabad/>

Introduction to Resource Oriented Computing and NetKernel.

The Problem

As we move to new architectural models such as SOA and distributed systems, we deal with the impact of constrained brittle software everyday, even if we are too focused on the details to see it.

For example, XML messages are passed between systems providing a very flexible medium for information exchange. But, when software developers reach into their toolkits for help processing these messages, all they find are ways to bind the messages to low level code and so they inadvertently destroy the very flexibility provided by XML.

We find a similar story with databases. Relational databases are built on a set-theoretic foundation in which one thing is clearly evident - a query operation on a table always results in a table. However, when it comes time to process this information using current language tools the binding issue raises its ugly head once again and the elegance and power of the set approach is lost in a tangle of objects.

Software projects are failing because business information's natural fluidity and constant change is not served by inflexible and brittle code.

And, it will get worse. Why?

We have another problem. Welcome to the n-core generation. We can no longer rely on pure CPU speed increases for performance gains. Because of physical limitations, chip designers are going wide - they are providing multi-core systems instead of faster single CPUs. But most of our software can't take advantage of these new chips.

Is there a solution?

When software is built within a logical model on top of an operating-system-like abstraction running on a micro-kernel, dramatic things happen. Developers no longer think about threads. Threads are a physical level issue handled transparently by the micro-kernel. Software is no longer brittle, now software can be constructed using logically identified resources and services - objects, libraries, and other small scale details are hidden in the physical level by the micro-kernel. Software is larger scale and the amount of code required to build an application goes down by a factor of ten to one hundred.

Performance goes up and stays up because a dynamic virtual system can self-tune to match changing workloads. This sounds like the ideal solution for software development - move away from the small scale details and step up onto a new and more powerful abstraction, just as the IT industry has done throughout its history.

Will this work? Fortunately, the answer is yes. A growing number of companies are building their projects this way and repeatably find they reap the benefits of more flexible, faster systems written using significantly less code in dramatically shortened time frames. Systems scale linearly with compute cores, developers don't even think about threads. Finally, we can build our information bridges with cranes and box girders.

We call this new model **Resource Oriented Computing**.

Learn more. Attend the developer conference.

Reserve your seat at the conference by sending an email to

hyderabad2007@1060research.com