George E. Danner
Industrial Science, LLC
NKS 2006 Conference
The Fairmont, Washington DC
June 16, 2006

Ladies and Gentlemen, thank you for coming. By now many of you have had a full day at the conference, and you've seen yet another year of steady growth in the NKS community. I'm amazed at the energy and creativity of the researchers here – some really important work is always going on.

My own focus is a bit different from what you've seen. The bulk of the research in NKS is about advancing the science – the collective body of knowledge about simple programs and how they behave. I come from a business background, so my interest is in the real-world application of NKS to solve problems.

It is different, but it should not be unexpected. If we look back in history, we see that almost every important scientific discovery progresses through two eras. The first is naturally the understanding of the discovery itself; the second is the harnessing of that discovery for useful purposes. While the present state of NKS is still arguably in the first era, it isn't too early for people like me to be thinking about the exploitative side of simple programs, and that is exactly what I have been doing for the last three years. And so it is appropriate for me to report back to you what I have found along this journey, and to pass a few ideas back to you to consider.

Now I run a consulting practice in corporate strategy, and as such we get handed all sorts of very interesting and decidedly complex problems. We don't specialize in any one industry – we've done work in retail, manufacturing, government, telecommunications, energy – you name it. This has given us a rather unique front row seat on how complexity shows up in the real world. And of all the problems we've witnessed, there are some very durable patterns that emerge:

1. Dynamics. Systems seem to exhibit short term and long term behavior that is decidedly different. (example)
2. There are many "moving parts" to real world systems, each contributing to a collective whole that is deducible only by observing the system in total. (example)
3. The levers of control on the system are not intuitive, and require experimentation or simulation to understand. (example)
4. The most fundamental of all activities is search.

Now some of these characteristics may start to sound familiar to you if you've been working in NKS for a while. There are lots of simple programs, for example, that show behavior in its early stages of evolution that is quite different from its later stages. Small, localized interactions of simple programs lead to an interesting system-wide result.

Controlling the outcome of a simple program is not intuitively obvious. One has to run every possibility of a class of simple programs to understand the patterns of behavior.

So the characteristics of simple programs match up rather squarely to the characteristics of real world systems. While not conclusive, this should give us some motivation to take this to the next step: can NKS shed light on complex systems in the real world? I think this is an open question, and a very important one, and needs some healthy debate around the community. My job as it stands today is to find fodder for that debate: to do enough experimentation to get threads of activity going on real world problems. (as a point of departure from my main remarks here I am actually looking for a research team at a university that can help me with some development work along these lines).

As I've mentioned in previous talks on this subject, I believe that NKS can be applied to real world problems on two levels. First and foremost, there is direct modeling of systems – say creating a simple program that mimics some market or sub-economy or even company, and therefore is effective at revealing insight as to the future behavior of that system under defined conditions. Here is an example I'd like to show you:

<<Anadarko Model>>

Second, and this is perhaps more subtle, is the methodology of NKS – how NKS looks at problems. I've often called this Meta-NKS because it isn't so much about simple programs and what they do as it is a general problem solving approach.

There are a few signature features of meta-NKS. One is observation - simply looking at representations of systems. This may sound outright trivial, but from where I come, it is not. You would be amazed at how few real measurements are made in the business world. Raw data, we have that. But data placed in some kind of visual representation where color, shape, and scale have meaning – that is much rarer. So in business we are very primitive in the ways that we observe our own data. You have much to teach us.

In fact I've become quite a student of visualization of late and I would like nothing more than to see the fields of data visualization and NKS come together in some symbiotic way. Many of you know the work of Edward Tufte – he and others like him could dramatically strengthen the way we visualize the fingerprints of simple programs.

The other feature is enumeration – cataloging all possible variations of a system, then searching through these enumerations, automatically or manually, to find patterns of interest. Again, the business world is way behind: most models I've seen draw upon one singular view of the future, or at most a handful of scenarios. Extreme conditions are rarely considered, yet are at the heart of every known failure in modern business.

The owners of these real-world problems, the people who are at the front lines of business complexity, have much to learn simply by mimicking the generalized solution approaches that are the hallmarks of NKS.

Then there is the more direct approach – using simple programs as models or tools to derive insight about a system under study. In the field I'm in, I'm often asked if Optimization is feasible using NKS. I actually posed this question to Stephen in 2003, and at that time he was skeptical that NKS and optimization could embrace each other, but he followed that by saying that he wasn't entirely sure why.

There are two schools of thought here. One is that NKS not practical for optimization, because you are asking it to solve a closed-form problem. Not only that, the search spaces for optimization problems are typically intractable on practical computers.

Fair enough, but what about the Wolfram Tones site? One might think that solutions – in this case sounds that are pleasing – would be rare in a vast ocean of very poor solutions. It turns out that this is not the case. So things like Wolfram Tones call into question these assumptions about NKS not being appropriate for this or that – including optimization. This is a second research thread that I am working on.

One thing is for sure – if someone does devise an optimization method using NKS, it will not be just some variant of existing approaches – like GAs or LPs – rather it will be a radically different way to do optimization – and that in and of itself will be interesting to several disciplines.

So what is it going to take to bring the two communities together – work in pure NKS and those that own complex problems in the real world? I posed a suggestion in the last conference, and I'll re-iterate it here, with some updates. A solution could be stylized problem forms – that is, problems that strip away the parochial details of a given industry or organization, yet contain characteristics that are readily mappable to real problems. On the other hand, these problem forms are abstract enough to be of interest to researchers. It isn't the very best example, but certainly one can think about the Traveling Salesman Problem as a rough analogy.

So I've been busy in the last couple of years working on devising problem forms that meet these criteria, and I've come up with one – I'll call it the Easter Egg Problem. Here is the setup: <<discuss>>

Jason Cawley has been working with me on this for some time, and he's made some very helpful suggestions. One is to set up an experiment whereby we "cheat" in our favor by giving ourselves lots of a-priori information about the location of eggs. We could also "double-cheat" by making our simple programs not so simple. Cheating ensures that will be some solution that works – like starting from a known point. Then we gradually relax the amount of a-priori information to see precisely where along that spectrum we lose our ability to construct adequate solutions.

Now my suspicion is that we will find that the threshold for the need to cheat to produce good solutions will be surprisingly low. If that is the case, that means that there is some inherent capability of simple programs to conduct effective searches. And if that is true, that means that simple programs can give us disproportionate leverage on the initial

information we have about complex systems. The implication here is that a company faced with a very complex market or product decision has some hope of understanding the ramifications of that decision much better by simulating it. Moreover implementing the solution is one of having the many small parts of the company follow simple rules to produce a collective result. This could be a very dramatic finding. We are still working on this, and I hope to be able to report some progress back to you all very soon.

In conclusion, let me just say how exciting it is to be living in this era. In the business world we have extensive access to data. We have problems whose yet-to-be-found solutions are worth millions to billions. And the standard of precision around solutions is far lower than what many of you may be used to. It is often sufficient to have solutions that are "better than average" over the range of possible solutions. So there is a far lower bar that we have to meet, and that should be encouraging to those of you interested in this field. Bridging the gap between the real world and NKS could be one of the most important developments in the community in the next few years. I encourage you to stay tuned.

Thank you very much.