At the end of 2002, I attended a workshop in Edinburgh sponsored by the UK Computing Research Committee. The meeting’s topic was Grand Challenges for Computing Research. It was modeled on a US initiative, the Computing Research Association’s Conference on Grand Challenges in Computer Science and Engineering, held in summer 2002.

Both events aimed to generate from computer scientists and other researchers a range of generally accepted long-term, large-scale challenges for computing research in the twenty-first century. The term Grand Challenge sounds very beguiling. It suggests a group of people working together in pursuit of a common goal—one that they believe is valuable and achievable within a predicted timescale.

**Defining a Grand Challenge**

When the UK workshop was announced, the organizers tried to outline a Grand Challenge’s essential characteristics. A Grand Challenge should be driven by scientific curiosity. It should address fundamental issues concerning the subject’s nature and limits. Ambition is another requirement—to build something that no one has seen before and to go beyond what is currently possible. This will likely require the development of tools, techniques, and methods that are unknown at this point. The challenge must excite us, because it needs to generate enthusiastic support from a community of researchers and developers. It must also capture the imagination of the wider public and be recognized as worthwhile by those in other disciplines. A good challenge will require cooperation between groups but will also foster a healthy sense of competition. It will decompose into intermediate objectives that will still be important achievements if the overall project fails. Finally, a good challenge needs a halting condition—it should be obvious when and to what extent the challenge has or has not been met.

Grand Challenges have been set in other areas of human endeavor, sometimes with spectacular effects. President Kennedy’s commitment to put a human on the moon by the end of the ’60s is one example, although it had a military and political imperative that went well beyond the science and engineering. Mapping the human genome is another example of a challenge successfully met. At various times, challenges have not been met. Efforts in the ’70s to cure cancer within a decade failed. Nevertheless, the challenge has undoubtedly led to many medical and biological advances.

**AI challenges old and new**

Computer science, of course, has witnessed equivalent efforts, many of which have had a strong AI and intelligent systems flavor. One long-standing challenge was to build a championship chess program. When the first programs won significant tournaments, they certainly attracted considerable interest. But it was Deep Blue’s defeat of Garry Kasparov that people noticed. This was actually a triumph of hardware and carefully crafted search algorithms—what I have called elsewhere “Brute Force with Insight” (Nov./Dec. 2001 IEEE Intelligent Systems). You can debate how far this has really transformed our understanding of what is important in building complex AI systems. However, I believe that massive computing power directed with a light touch toward interesting parts of a problem space can yield remarkable results.

Other past AI Grand Challenges have either failed or remain unfulfilled dreams and aspirations. These include the 1960s ambition of automatic translation from Russian to English (although Web services are now commonly deployed that do at least an initial rough translation between languages). Perhaps the greatest unmet challenge is that of building HAL within X years. For many of us, this was partly why we took up AI. So, it is interesting to consider how many of this new batch of computer science Grand Challenges are variants on this theme.

The 2002 US Grand Challenge workshop had an invited group of researchers develop examples of challenges. One example was safety.net, a ubiquitous-computing infra-

---

**URLs**

**CRA Conference on Grand Research Challenges in Computer Science and Engineering:**
www.cra.org/Activities/grand.challenges/index.html

**Grand Challenges for Computing Research:**
http://umbriel.dcs.gla.ac.uk/NeSC/general/esi/events/Grand_Challenges
structure for managing the full range of disaster relief responses and recovery. Another was a Teacher for Every Learner—a software proxy for each of us, our very own intellectual mentor, customized to our individual learning requirements and attainment. A dependency Grand Challenge attempted to tackle the whole issue of trusted systems. Yet another looked to conquer the problems of complexity that are increasingly apparent in our computing infrastructures. How do we build, manage, and maintain systems with billions of parts? Finally, researchers advocated the concept of a cognitive assistant, many elements of which featured in Ron Brachman’s article (“Systems That Know What They’re Doing”) in our last issue.

The UK workshop received over 100 submissions to its call for challenge statements. The authors of around 60 of these attended the workshop; their submissions were organized into four panels: software engineering, ubiquitous computing, human-oriented computing, and biologically inspired modeling. Each panel then worked up one or two consolidated challenges. Final topics included

- Dependable-systems evolution
- Scalable computing for ubiquitous systems
- Healthcare coordination across a city
- An infrastructure for information management over a human lifetime
- A complete digital model of the nematode *C. elegans*

Interestingly, the two workshops had much in common in four areas: dependability, ubiquitous computing, integrated emergency response infrastructures, and digital cognitive assistants. In all these areas we can quickly discern the need for contributions from AI and IS researchers.

Both workshops aimed to mobilize the computer science research community behind bold ambitions and to present challenges that no single organization could tackle.

**Challenges gone wrong**

This is interesting and inspirational stuff, but perhaps we should pause and consider the various ways a Grand Challenge can go wrong. In an amusing speech at the UK workshop, Jayadev Misra (University of Texas) argued that you can classify any Grand Challenge into one of four basic categories. The first is challenges that are simply fantastic. These typically depend on a series of other Grand Challenges, all of which must succeed at a crucial point in time.

The second category is driven by the recognition that a discipline is in imminent danger of collapse. Researchers prepare a tedious polemic that critiques the lack of scientific rigor in their field. The answer, they argue, is to eschew any kind of short-term engineering approach. What is needed are penetrating questions regardless of whether there is any prospect of answering them.

A third category comprises challenges that are collections of pet projects promoted by professors and bundled into a larger proposal. Here, the main integration tool is the stapler. One tactic in this type of challenge is to include all potential rivals as collaborators to increase the chances of funding.

Finally, we must not neglect the managerial Grand Challenge, where all the components are available and the challenge is to put them together. Here, we rely on project plans and milestones, deliverables and success Gantt charts.

Nevertheless, it seems to me that the simple articulation of a Grand Challenge can help inspire a whole generation of researchers. Misra mentions that one of his colleagues, when asked for papers that proposed grand visions for computing in its early days, suggested Vannevar Bush’s “As We May Think” (July 1945 *Atlantic Monthly*). I too have heard this mentioned on many occasions as an inspirational article.

In “Systems That Know What They’re Doing,” Brachman articulates a worthy Grand Challenge. It is a system that would learn from experience, one that could be instructed and could show improvement. Such a system could explain what it was doing and why. It would have a degree of reflection such that it knows the limits of what it can prove for itself and is prepared to ask for help when it must. It could cope with surprise, learning from the unexpected. The vision here is reminiscent of HAL and sets out challenges at the heart of AI.

Of course, one of the outcomes of any Grand Challenge that actually gets going and receives funding is to generate a vigorous debate about whether the effort is fundamentally ill-conceived. We can trace a number of significant changes in AI’s view of itself to arguments around whether certain sorts of Grand Challenge were the right way to proceed. These sorts of debates can be beneficial in their own right as we question the assumptions and goals, limits and scope of our discipline. Thus, whatever our track record for realizing such ambitions, it’s important that we continue to set ourselves Grand Challenges.