Driving towards the Computational Knowledge Economy

by Conrad Wolfram
Director of Strategic and International Development, Wolfram Research, Inc.

In May 2009, Wolfram|Alpha™ launched to worldwide excitement. It introduced the new concept of "Computational Knowledge Engine™"—working out custom answers to queries made, rather than picking out existing information like a traditional search engine. Think of the analogy of a research assistant who finds an answer specifically for you; whereas a search engine compares to a librarian who directs you to books or articles that might be relevant, but doesn't generate new answers or content.

In fact www.wolframalpha.com the website was just one window into the multi-decade Wolfram|Alpha project and its vision for all systematic knowledge: of curating, making computable and democratizing high-level use. This process is proving not only of use for publicly available information but also for inside business and government organizations. It's a way to structure and leverage knowledge assets.

Let's zoom out for a moment and think of what we mean by knowledge. I would argue that there are three facets. There's knowledge of base facts or data, knowledge of how and what to work out from them, then knowledge or judgment of what's important and what interpretation should be attached to the result. I'll call these facts, method and interpretation.

So far, the web and search has largely focused on finding base facts or opinion (where there are few facts, little method and lots of interpretation). And it's been spectacularly successful in democratizing this information retrieval—and some would say aspects of democracy itself.

But in almost all cases, you the human have to further process the results manually, mashing up the data with little systematic expertise to help. Say for example, you want to compare Korea's GDP to the UK's in € at today's rate. It's unlikely anyone has specifically precomputed the exact answer to this for you to search out so you'd have to find the GDPs, work out the exchange rates and plot them out manually. In short, there's a lot of work that the computer should be doing instead of you to get you to a useful answer.

Although computation itself is nothing new, Wolfram|Alpha's launch has put the spotlight on how it can help to democratize knowledge. One of its major achievements so far is to define and initiate an innovative practical process based on high-powered computation for making a knowledge or answer engine. It has encouraged us to think about the internet as active software for generating content rather than simply passively containing information. It's early days for this ambitious approach; there's a long way to go.

The approach is twofold: to curate the base facts, to make them computable; and to interpret human input so it becomes computable too. Then the critical steps are to select and apply useful computations and build custom answers to queries, with optimized representation on-the-fly. Crucially, human expertise has been pre-encoded up front to inform what's "useful" for a given type of user. For example, if you wanted an idea of how Microsoft Corporation compares to Google, it's important to have an expert editorialize which parameters to present—out of the hundreds of possibilities—and it's in this sense that democratizing human expertise is a key goal of Wolfram|Alpha.

Why now? Why Wolfram? Because we've been building high-powered computation technology—our software *Mathematica*®—for 23 years. Back then, few believed mathematical computation was of mainstream use. Now, not just technical professions but increasingly high-powered computation underlies all companies, governments and people's every day living.

But more than just betting on the growing importance of computation, we've always believed in separating the expertise of how (i.e. with which method) you solve a problem from what you want to solve—of automating computation and technical expertise. And key to achieving this was to find a general enough representation for everything we were encoding—functions, data, graphics, documents, interface etc. This one systematic, unified form is symbolic programming. Along the way we developed highly scalable web deployment technology, flexible report generation and a unique programming language.

Building and deploying Wolfram|Alpha with this *Mathematica* technology was crucial for delivering the four technical pillars of a computational knowledge engine: curated data, computational intelligence, natural language interface and automated presentation.

So much for the need and technology behind a knowledge engine. What of the consequences?

Firstly, the value of systematic information that isn't computable will decrease—as individualized and direct reprocessing by computer expertise becomes prevalent for everyone. Those who hold large bodies of knowledge or data therefore face a critical task of reformulation, systematic curation, and appropriate tagging to get their knowledge assets in shape.

Governments are a case in point. They collect huge quantities of data and collectively fund around $300 million of R&D. Yet we usually end up with "dead" summary documents and expertise or direct knowledge with a very small fraction of the population. We need to find a way to broaden the use and present an increasing fraction publicly, but in what form? It's only relatively recently that data and technical research output has become "electronic" rather than existing on paper; but the chosen electronic form is rarely computable and so further transformation is required.

By using this computability and applying expertise, you can publish technical results in a way that's useful to a far wider range of people that ever before. We have started work with a number of governments both to make their data computable and offer the wolframalpha.com channel for that data's public use. We've also started showing what can be done by publishing interactive applications rather than traditional static document presentation of ideas at demonstrations.wolfram.com. In the future we will see custom answers, which include interactive applications, automatically built, to represent knowledge and answer questions.

Corporations too have a lot of systematic knowledge—and perhaps have even more extra value to leverage from it. Their need is as much to improve internal information usage as external (a fact confirmed by the number of partnering projects our team is engaged with for using Wolfram|Alpha technology on private, internal information). While preformulated reports of information are standard, few organizations have the setup to get custom calculated answers to drive-by questions like “what was John Smith’s incentive last quarter” or “web hits by country divided by population of country”.

In the end, the key issue for companies and governments alike is the need to leverage their knowledge assets to the maximum. And key to achieving this is direct access by end users of the information to the processed results. Crucially, those users will often not know the mechanics of the processing needed—or even the exact form of the question to ask, but they do know when resulting information will help them. They want to be increasingly task rather than method orientated (while understanding what limitations the methods impose). Making the computer not other humans be the knowledge reprocessing intermediaries is crucial to democratizing knowledge and leveraging the depth and breadth of its use.

By encoding expertise one-level down, humans can operate at a higher level by standing on the automation of computers. An analogy might be managing an organization: you don’t need to know how do to everything that each staff member does, but you need to be able to understand what it means, what questions to ask and what outcome you’re looking for.

I’m now going to loop back and suggest that people should be differently trained in this new world of computational knowledge, that we should reformulate technical education of the future.

This surely sounds perverse. After all I’ve just explained how computers should be enabling everyone access to computational expertise—whether specifically trained in that field or not—so surely I cannot simultaneously be suggesting their retraining too?

The point is that technical education has been held back by the need to teach the mechanics of calculating by hand. That’s removed valuable time to teach conceptual understanding—both for formulating questions and interpreting results. It’s not that with current training, people won’t benefit enormously from the computational knowledge approach, it’s that the limitations as to what can be achieved will have contracted. Moving beyond the conventional issues of how easily information can be acquired or computed, the critical path will be what question to ask, what factors to include, what the answer means.

Adopting computer-based math will enable most of the 80% or so of math education time currently spent on calculating techniques to be refocused on to these aspects of mathematical understanding, and will provide enormous benefits to Knowledge Economies.

In fact I would go so far as to say that we will see a new era of “Computational Knowledge Economy” where the battleground for extra value has shifted up from applying base knowledge effectively, to computing, generating and applying derived knowledge effectively. Because computational expertise will be increasingly available to everyone, it will be an assumed, democratized resource to be exploited—not just the preserve of the specialist.

It’s exciting to see that computation for everyone has come of age. How do we shape our organizations to use it to the fullest?