Model builder. "I'm interested in tools that change the way science is done," Stephen Emmott says.

CREDIT: COURTESY OF MICROSOFT RESEARCH

COMPUTER MODELS

Coming Soon to a Lab Near You: Drag-and-Drop Virtual Worlds

Researchers at Microsoft hope to convince scientists that transparent, easy-to-tweak numerical simulations are as straightforward as clicking a mouse

CAMBRIDGE, UNITED KINGDOM—Techies love to hate Microsoft. They curse the "blue screen of death" that appears when a computer running the company's flagship Windows operating system crashes. They deride what they say are Windows's bloated code and security flaws. And they complain that the software giant is perpetually behind the curve on new technologies such as smart phones and tablet computers. In short, techies—many scientists included—are a tough audience.

So in 2003, Stephen Emmott could have been forgiven if he had walked the other way when Microsoft executives asked him to come aboard and help the company figure out what it should be doing in science. Emmott, then a neuroscientist at University College London who had worked previous stints at Bell Laboratories and NCR, accepted the challenge, provided he could build a cutting-edge computational sciences laboratory within Microsoft's research division to tackle knotty scientific challenges. If successful, the software the group created would help other scientists make broad impacts on their fields as well.

It's too early to say whether this strategy will make money for Microsoft in the long run. Indeed, for now, Emmott says that he and his colleagues plan to share their wares freely with the academic scientific community. But Emmott's vision is now in full gear. He spent his first year selling his ideas within the company and began hiring staff members. Now Microsoft Research's computational science lab has 40 Ph.D.s and students and continues to grow.

A couple of the researchers are software engineers—obviously Microsoft's stock in trade—but most come from disciplines as varied as ecology, neuroscience, mathematics, and developmental biology. Their hope, say Emmott and others, is to transform the way scientists study complex, ever-changing systems, such as the global carbon cycle and information processing inside cells. To do so, they're working to develop a suite of new software tools including novel programming languages that better represent biological systems and computer models that work across multiple scales, simulating carbon budgets at the levels of leaves, trees, and forests, for example. They're also striving to make those tools simple to use, thereby extending the types of studies that can be done by researchers who aren't full-time programmers. "I'm interested in tools that change the way science is done," Emmott says.

Prototype versions of several of these tools are now up and running and being put through their paces by researchers at Microsoft. One program, currently called Microsoft Computational Science Studio, contains components that are able to handle disparate types of data, quickly plug them into a model, and visualize the interactions. Other packages help biologists design and simulate DNA circuits for biological computers and manage wireless sensor networks for tracking animal behavior. Carol Barford, an ecologist at the University of Wisconsin, Madison, says she has used other software packages produced by academics to build and visualize complex models. She recently began working with Microsoft's software to investigate how future climate-change scenarios might affect agricultural production around the globe. "It's the slickest one I've ever seen," she says.

Capturing complexity

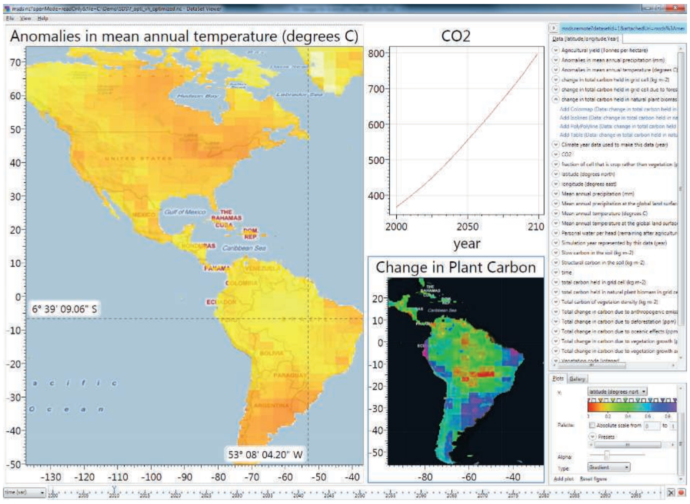
So why is a computer software company known primarily for its operating systems and business software mucking around with modeling the global carbon cycle and working to understand the human immune system? Sitting in his ground-floor office across the road from the University of Cambridge's famed Cavendish Laboratory where J. J. Thomson discovered the electron and James D. Watson and Francis Crick deciphered the structure of DNA, the 50-year-old neuroscientist spells out his thinking. For starters, Emmott says, science is "set to be the driver of our times." So progress on new computational tools and methods has the potential to make an impact on numerous fields. As well, he adds, scientific problems at the frontier of computing are perfect for honing talent and ideas that may lead to new or better Microsoft products.

A good way to start that improvement is by making computer models simpler to navigate and understand. Computer models, of course, aim both to approximate the real world and to predict how it might change in the future. That's relatively straightforward when a model's key inputs, or parameters, are known. That is why engineers can land a rocket on the moon and construct bridges capable of withstanding gale-force winds. But there are a host of problems, called inverse problems, for which not all of the right parameters are known. For them, researchers must sift through vast amounts of observations to identify which set of parameters to plug into their models and their appropriate

Published by AAAS

values. To make matters more challenging, researchers are often also unclear about how a complex system's key parameters interact. That makes accurate model building and predictions dicey at best.

Take a complex climate model, for example. Over decades, researchers at labs around the world—including the Met Office Hadley Centre for Climate Prediction and Research in Exeter, U.K., and the National Center for Atmospheric Research (NCAR) in Boulder, Colorado—have built enormously complex general circulation models (GCMs) that predict the future state of Earth's climate by tracking how parameters such as wind currents, sea surface temperatures, polar ice cover, clouds, and rising greenhouse gas levels interact. As new observations revise, for example, the amount of light Earth's changing ice cover reflects into space, researchers can tweak that parameter in their model and run a new simulation to gauge the likely impact. Typically, however, this process is very slow. "It can take months or years to iterate current models," Emmott says.


Anomalies in mean annual temperature (degrees C)
70
60
50
40
30
20
10
0
-10
-20
-30
-40
-50
-130
-120
-110
-100
-90
-80
-70
-60
-50
-40
6° 39' 09.06" S
53° 08' 04.20" W
UNITED STATES
Hudson Bay
Labrador Sea
Gulf of Mexico
MEXICO
THE BAHAMAS
CUBA
DOM. REP
HONDURAS
Caribbean Sea
PANAMA
VENEZUELA
COLOMBIA
ECUADOR
BOLIVIA
PARAGUAY
ARGENTINA
Pacific Ocean
CO2
800
700
600
500
400
2000
2050
210
year
Change in Plant Carbon
20
10
0
-10
-20
-30
-40
-50
-80
-60
-40
File
View
Help
Data (latitude,longitude,Year)
Agricultural yield (Tonnes per hectare)
Anomalies in mean annual precipitation (mm)
Anomalies in mean annual temperature (degrees C)
change in total carbon held in grid cell (kg m-2)
change in total carbon held in grid cell due to fores
change in total carbon held in natural plant biomas
Add Colormap
Add Isolines
Add PolyPolyline
Add Table
Climate year data used to make this data (year)
CO2
fraction of cell that is crop rather than vegetation
latitude (degrees north)
longitude (degrees east)
Mean annual precipitation (mm)
Mean annual precipitation at the global land surfac
Mean annual temperature (degrees C)
Mean annual temperature at the global land surfac
Personal water per head (remaining after agricultur
Simulation year represented by this data (year)
Slow carbon in the soil (kg m-2)
Structural carbon in the soil (kg m-2)
time
total carbon held in grid cell (kg m-2)
total carbon held in natural plant biomass in grid ce
Total carbon of vegetation density (kg m-2)
Total change in carbon due to anthropogenic emis
Total change in carbon due to deforestation (ppm)
Total change in carbon due to oceanic effects (ppm
Total change in carbon due to vegetation growth
Vegetation code (integer)
Plots
Gallery
Palette
Absolute scale from
Presets
Alpha
Type
Gradient
Add plot
Reset figure
time (year)

Quick turnaround. Computational Science Studio is one of several new software tools aimed at making complex models easier to build, test, and refine.

Equally challenging is that such complex models are written in computer code that is impenetrable to most researchers outside the group responsible for updating it. That makes it difficult for researchers working in one particular area, such as tracking tree mortality rates in the Amazon, to get a sense of whether their data might influence the broader climate picture. That, in turn, slows the search for additional important parameters that might improve the models. "Now we have loads of data," says Rosie Fisher, an ecophysiologist at NCAR. "What we need are ways to quickly find the patterns that emerge from that data. It is a huge software problem. So it is very exciting that the Microsoft people are willing to look at this."

From office to lab

Why haven't others set their sights on such a goal before? Academic groups, Emmott explains, are adept at creating software tools and models to meet their own needs. But they typically don't have the time, money, or inclination to make them broadly useful to other researchers. "It was never really someone's job to do it," Emmott says. "The need was always to get your own research done rather than providing a service to the community." In the modeling arena, he adds, he hopes to streamline modeling software to make complex models far more accessible. "In essence, we want to do for modeling software what Microsoft programs such as Word and Excel did for business software," Emmott says.

That's where Computational Science Studio comes in. At the heart of the program—and others the lab is developing—is a software module code named Scientific Data Set (SDS), a sort of universal translator capable of recognizing and interpreting a wide variety of common data types, such as time series, satellite and medical images, and multidimensional numerical arrays. Users can also add new types of their own. The SDS allows "complete promiscuity" in working with virtually any type of data, Emmott says. With this ability, programmers then created software to allow virtually anyone to plug in different data sets and at the click of a button set up a model of how they interact. A visualization component renders the relationships, such as mapping out how different levels of deforestation in the Amazon rainforest would impact surface temperatures in Africa.

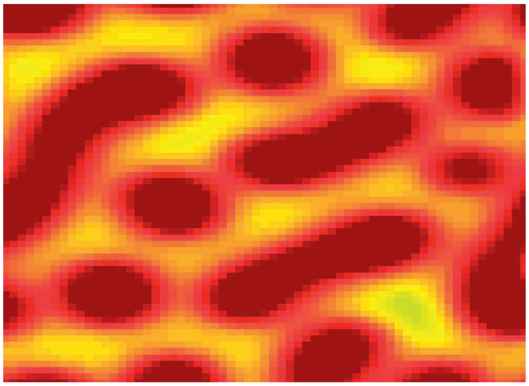
Last summer, Drew Purves, who heads Microsoft's Computational Ecology and Environmental Sciences (CEES) group, demonstrated the modeling package to Simon Lewis, an ecologist at the University of Leeds in the United Kingdom, and some of his colleagues. At the time, the demo showed how the drag-and-drop modeling software could plug in a wide range of data on biological processes—such as rates of photosynthesis and soil nitrogen fixation—and integrate them with changing CO_2 and temperature levels to show how a change in climate might affect the amount of carbon stored in forests. Among other things, the demo revealed how different deforestation rates could speed up or slow down temperature increases by 8 years by 2050. The model ran on a desktop computer in just a few minutes. "I was pretty impressed that they could be so computationally efficient," Lewis says.

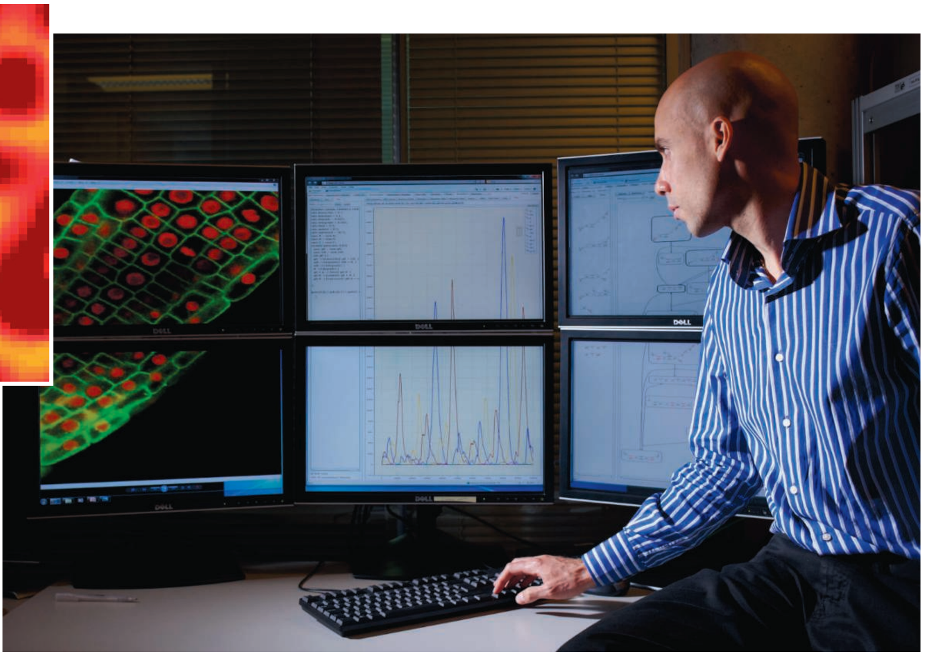
That efficiency could be vital to improving how current climate models handle the effect of biological feedbacks on future climate, for now one of the biggest uncertainties current models grapple with. For example, forests currently store as much carbon as is present in the atmosphere. Most climate modelers expect average temperatures to warm by between 1.6ºC and 4.3ºC by 2100 given midrange carbon-emissions scenarios. Less clear is how plants, from trees to grasses, savannas, and agriculture will respond. If the extra CO_2 in the atmosphere makes most plants grow faster, this could ameliorate some of the warming. Yet if the higher temperatures increase plant mortality, this could cause gigatons of carbon now stored in tree trunks and roots to wind up in the atmosphere and accelerate warming. Current models, known as dynamic global vegetation models, project widely different outcomes. By the year 2100, vegetation might be a carbon sink for 11 gigatons of carbon a year, or it might release an additional 6 gigatons of carbon every year beyond humanity's contribution.

Today's GCMs simulate detailed physical processes, such as ocean and atmospheric circulation. But they've had a harder time incorporating the myriad biological processes. Part of the challenge, Emmott explains, is that biological processes vary widely at different scales. At the level of an individual leaf and tree, rates of photosynthesis and nutrient uptake are key to understanding a tree's health and viability, whereas the competition of trees for light and how trees disperse their seeds becomes important at the level of a stand of forest.

Conventional models have struggled to incorporate such complexity across multiple scales. But the plug-and-play environment of the new software makes it a more manageable task. In fact, since their early demos to Lewis's group, among others, Purves and colleagues have constructed a more detailed carbon cycle model incorporating many bio-

CREDIT: COURTESY OF MICROSOFT RESEARCH





Programmer for life. Andrew Phillips (*right*) helped design software that tells how to engineer bacteria to grow in a "Turing pattern" (*above*).

CREDITS (TOP TO BOTTOM): COURTESY OF MICROSOFT RESEARCH; COURTESY OF MICROSOFT RESEARCH AND JIM HASELOFF, UNIVERSITY OF CAMBRIDGE

logical processes. The unpublished preliminary models predict that the carbon stored in vegetation by 2100 will fall within the range forecast by previous models.

The new carbon cycle model is far from the last word on the matter. Rather, the hope, Purves says, is that this ability to quickly build and test models will allow researchers, and entire research communities, to speed the cycle of improving their models. "One of the things we lack is the ability to explore a large number of scenarios," Purves says. Computational Science Studio and the lab's other new tools can help remedy that, says Matthew Smith, an ecologist in the CEES group. "The idea here is, you plug it in and ask if it is important," he says. "You can form your tests so much more quickly, and this allows you to cycle through them much faster." Equally important, the software should make it easier for researchers to test their ideas without becoming experts in writing code.

Another advantage of Microsoft's drag-and-drop modeling software is that it makes it easy to see what assumptions are built into the model, and it can even specify the degree of uncertainty in different components. Ultimately, Smith and Purves say, this type of a more generic and transparent modeling platform could help climate scientists and other groups compare their wares. "GCMs all have different data fed into them," Purves says. "We should take several different models and train them with the same data" and compare their outcomes, Purves adds. Eventually, that should reduce the models' collective uncertainties and improve their predictions.

Beyond climate

Emmott and his colleagues have set their sights on modeling far more than climate. They've also recently developed new programming languages and other tools for modeling complex biology. In one example, they've modeled a set of immune molecules known as the major histocompatibility complex class I. MHC-I molecules grab, replicate, and present small protein fragments known as peptides on the outer surface of cells. Immune sentries called T cells then inspect those peptides for foreign signatures common to viruses and other invaders and kill cells that might spread infection. Much is known about many of the key molecular MHC-I players, but the complexity of their interactions has prevented biologists from constructing a good model of how they behave in cells.

So Emmott and his colleagues used Computational Science Studio to plug in the key molecular players. The model enabled them to compare different theories of how the MHC-I system works. The prevailing view, Emmott explains, has been that a process known as peptide editing governs which peptides are presented to T cells and thus are most likely to generate an immune response. The team's latest model suggests that peptide editing indeed "accounts for a lot of the data," Emmott says. But the model gave an even better fit when the team added a secondary step, known as peptide filtering, in which a protein called tapasin recognizes foreign proteins and prioritizes which ones are displayed. This preliminary work also needs to be fleshed out before being published, Emmott says, but it underscores that plug-and-play models can test new ideas very quickly.

Not everyone at the lab is trying to simulate natural processes. Andrew Phillips, a computer scientist turned biologist, is leading a group developing computing languages and models for programming biological systems, from DNA strands to cells. In one project, Phillips and several colleagues created a new programming language for designing circuits in which tailored DNA strands interact to carry out a computation through a process called strand displacement. On 17 June 2009 in the *Journal of the Royal Society Interface*, Phillips and Microsoft colleague Luca Cardelli reported that they could use their setup to design simple logic gates and catalytic circuits, among other functions. They are testing the results with real DNA in test tubes.

In a second project, Phillips and colleagues created a programming language and models for designing genetic circuits that function inside cells. The team simply writes a program for a desired function, and the software will design the DNA strands needed for cells to pull it off. In one example, Phillips starts with an input that allows cells to express green fluorescent protein and writes a program to make a colony of cells in a petri dish express a pattern of colored regions known as a Turing pattern. The software then automatically generates the set of DNA sequences needed to produce the pattern. At this stage, the result is still an onscreen simulation, but Phillips and his colleagues are partnering with others to try to replicate it in real cell colonies.

As in other areas, Microsoft's computational scientists aren't alone in their efforts to push the envelope on synthetic biology. But the Cambridge team's new software languages and models could bring such work—which now requires heavy lifting by highly specialized labs—within reach of a far broader audience. If so, their stock among scientists could be on the rise.

–ROBERT F. SERVICE

Published by AAAS