Climate Change: A Grand Software Challenge

Steve Easterbrook
Department of Computer Science
University of Toronto
140 St George Street, Toronto, Ontario, Canada
sme@cs.toronto.edu

ABSTRACT

Software is a critical enabling technology in nearly all aspects of climate change, from the computational models used by climate scientists to improve our understanding of the impact of human activities on earth systems, through to the information and control systems needed to build an effective carbon-neutral society. Accordingly, we, as software researchers and software practitioners, have a major role to play in responding to the climate crisis. In this paper we map out the space in which our contributions are likely to be needed, and suggest a possible research agenda.

Categories and Subject Descriptors

D.2.0 [**Software Engineering**]: General

1. INTRODUCTION

Climate change is likely to be the defining issue of the 21st century. The science is unequivocal - concentrations of greenhouse gases are rising faster than at any previous era in the earth's history, and the impacts are already evident [1]. Future impacts are likely to include a reduction of global food and water supplies, more frequent extreme weather events, sea level rise, ocean acidification, and mass extinctions [10]. In the next few decades, serious impacts are expected on human health from heat stress and vector-borne diseases [2].

Unfortunately, the scale of the systems involved makes the problem hard to understand, and hard to solve. For example, the additional carbon in greenhouse gases tends to remain in atmosphere-ocean circulation for centuries, which means past emissions commit us to further warming throughout this century, even if new emissions are dramatically reduced [12]. The human response is also very slow - it will take decades to complete a worldwide switch to carbon-neutral energy sources, during which time atmospheric concentrations of greenhouse gases will continue to rise. These lags in the system mean that further warming is inevitable, and catastrophic climate disruption is likely on the business-as-usual scenario.

Hence, we face a triple challenge: *mitigation* to avoid the worst climate change effects by rapidly transitioning the world to a low-carbon economy; *adaptation* to re-engineer the infrastructure of modern society so that we can survive and flourish on a hotter planet; and *education* to improve public understanding of the inter-relationships of the planetary climate system and human activity systems, and of the scale and urgency of the problem.

These challenges are global in nature, and pervade all aspects of society. To address them, researchers, engineers, policymakers, and educators from many different disciplines need to come to the table and ask what they can contribute. In the short term, we need to deploy, as rapidly as possible, existing technology to produce renewable energy[8] and design government policies and international treaties to bring greenhouse gas emissions under control. In the longer term, we need to complete the transition to a global carbon-neutral society by the latter half of this century [1]. Meeting these challenges will demand the mobilization of entire communities of expertise.

Software plays a major role, both as part of the problem and as part of the solution. A large part of the massive growth of energy consumption in the past few decades is due to the manufacture and use of computing and communication technologies, and the technological advances they make possible. Energy efficiency has never been a key requirement in the development of software-intensive technologies, and so there is a very large potential for efficiency improvements [16].

But software also provides the critical infrastructure that supports the scientific study of climate change, and the use of that science by society. Software allows us to process vast amounts of geoscientific data, to simulate earth system processes, to assess the implications, and to explore possible policy responses. Software models allow scientists, activists and policymakers to share data, explore scenarios, and validate assumptions. The extent of this infrastructure is often invisible, both to those who rely on it, and to the general public [6]. Yet weaknesses in this software (whether real or imaginary) will impede our ability to make progress in tackling climate change. We need to solve hard problems to improve the way that society finds, assesses, and uses knowledge to support collective decision-making.

In this paper, we explore the role of the software community in addressing these challenges, and the potential for software infrastructure to bridge the gaps between scientific disciplines, policymakers, the media, and public opinion. We also identify critical weaknesses in our ability to develop and validate this software infrastructure, particularly as traditional software engineering methods are poorly adapted to the construction of such a vast, evolving knowledge-intensive software infrastructure.

2. RECENT LESSONS

In the past year, climate change has rarely been out of the news. Record-breaking temperatures across much of the world in the last few months capped a record-breaking decade [7]; world leaders met in Copenhagen to design a successor to the Kyoto Protocol (and largely failed to make progress), climate legislation wound its way through Congress in the US, and a computer hacker publicized thousands of emails from the University of East Anglia, to show climate scientists apparently behaving unprofessionally.

The latter event led to many sensational claims about scientists fabricating data, refusing access to data and program code, and conspiring to subvert the peer-review process. While a series of inquiries have subsequently shown these allegations to be false [13], there remain many lingering questions about the role of software in the computational sciences, and the broader question of how people get access to trustworthy information about climate science.

Software quality is a particular concern. Climate scientists build a variety of software tools to support their work. At the heart of the field are the Global Circulation Models (GCMs) that simulate the atmosphere, oceans, cryosphere and biosphere, to study the processes of climate change on a global scale, and generate future projections used in the IPCC assessments [14]. Less glamourous, but equally important, a large number of data handling and analysis tools are used for processing the raw observational data and the results of simulation runs, and for sharing climate data with the broader scientific community. Most of this software is built by the climate scientists themselves, who have little or no training in software engineering. As a result the quality of this software varies tremendously: The GCMs tend to be exceptionally well engineered [5], while some data processing tools are barely even tested.

Openness and reproducibility are crucial issues for this community. Scientists need to validate one another's results, and repeat experiments. The datasets are huge—climate data centers handle terabytes per day—and extremely complex, with frequent missing and noisy data; and have to be reprocessed for analysis at different physical and temporal scales. Because of the complexity of the data processing, open access to the raw data is almost irrelevant. To understand and use the data, one would need to know its entire provenance, including the processing steps that have been applied, the tools (with their settings) that were used, the rationale for each processing step, and any known quality issues and assumptions. Current data analysis frameworks only capture a fraction of this information, and even then, tend to separate it from the datasets. Current meta-data efforts in the earth systems modeling community [4] aim to tackle this challenge, but are still struggling with standardization of metadata descriptors; the ability to capture full data provenance is still a pipe dream. Meanwhile, scientists are not generally motivated to provide free and open access to the datasets, lest the effort they have invested in creating and analyzing them go unrecognized. Reproducibility is difficult because the large numerical routines are often tied to a particular supercomputer architecture and sensitive to small perturbations in the computations, so that changes in the hardware, compiler, or configuration files often make it impossible to re-run an old experiment.

The earth system modeling community understands many of these challenges, and is in the process of building the software infrastructure needed to better support modeling, data analysis and data sharing [3]. However, outside of this community, these challenges are rarely studied and poorly understood. Standard software engineering techniques often don't apply, because scientific software is built to solve problems for which the answers aren't known in advance, and only approximate solutions are possible. Similarly, standard software quality metrics may not apply well, because scientific validity doesn't correspond directly with traditional notions of software correctness [11]—as George Box said, "all models are wrong, but some are still useful". Testing this software is a major challenge.

Another lesson from recent portrayals of climate science in the media is that much better communication is needed of results of climate science, how those results are obtained, and what the implications are. The news media made serious mistakes reporting the science, often misrepresenting how science is done, and failing to put the latest events into the broader context. The rapid spread of misinformation (and sometimes wild allegations), especially through the blogosphere, demonstrated that we lack the tools to assess the credibility of sources of information, and the trustworthiness of the people commenting on it. Search engines provide instant access to a variety of information sources, but their ranking systems are unable to sort out politically-motivated misinformation from honest accounts of the science, nor are they able to rank search results for credibility.

Ironically, over a decade in which the scientific community has become steadily more certain and more pessimistic about climate change, the general public has become steadily more uncertain and polarized.

3. THE SOFTWARE COMMUNITY'S ROLE

For the software research community, we can frame the challenge as follows. How can we, as experts in software technology, and as the creators of future software tools and techniques, apply our particular knowledge and experience to the challenge of climate change? How can we understand and exploit the particular intellectual assets of our community—our ability to:

- think computationally;
- understand and model complex inter-related systems and systems-of-systems;
- analyze and prioritize multi-stakeholder requirements;
- build useful abstractions and problem decompositions;

- manage and coordinate large-scale open source design communities;
- study and understand evolutionary forces at play in technical infrastructures;
- identify, diagnose and repair bugs in socio-technical systems;
- incrementally refine existing systems and deploy enhancements during operational use;
- build the information systems and knowledge management tools that empower effective decision-making;
- develop and verify complex control systems;
- create user-friendly and task-appropriate interfaces to complex information and communication infrastructures.

In short, the software community brings a unique set of skills related to the analysis and (re-)design of complex technical systems. How can we apply these intellectual strengths to make significant contributions to each of the challenges of climate change mitigation, adaptation and education?

Climate change is a systemic problem, and effective action requires a cross-disciplinary approach. In the natural sciences, the focus is on the physical processes in the atmosphere and biosphere that lead to climate change. In geography and environmental sciences, the focus is on impacts and adaptation of ecosystems and social systems. In economics, the focus is on the trade-offs around various policy instruments for controlling emissions of greenhouse gases. In various fields of engineering there is a push for development and deployment of new low-carbon technologies. Across all these disciplines, an important missing ingredient is the appropriate software infrastructure for collaboration and data sharing.

Systems thinking [17] is needed to understand better how social and psychological processes (human behaviour, peer pressure, the media, etc) interact with political processes (policymaking, leadership, voting patterns, etc), and how both are affected by our level of understanding of the physical processes of climate change. Computational thinking [18] is needed to understand how to redesign the knowledge infrastructure that allows information about all these processes to be factored into effective global action.

Progress will only be possible by understanding the needs of a diverse set of stakeholders: *scientists* who need tools to improve their understanding of earth systems; *educators* who need tools to reach broader communities to explain what the science tells us, and how the science is done; *journalists and science writers* who need access to knowledge to raise awareness of pressing issues; *policymakers*, who need tools to design, analyze and monitor a coordinated set of policies at international, national and regional levels; *political activists and non-governmental agencies* who need tools to coordinate their campaigns to put pressure on governments to act; *individuals and communities* who wish to reduce their carbon footprints and share information about strategies that work; and *engineers* who will develop and deploy alternative energy systems.

4. WHERE IS RESEARCH NEEDED?

These challenges were explored at two recent workshops on software research and climate change, the first at OOPSLA, in October 2009, and the second at ICSE in May 2010. These workshops identified three main research challenges, each of which has a set of related research problems:

4.1 Computer-Supported Collaborative Science

The first thrust is the challenge of developing new software infrastructure to support and accelerate inter-disciplinary work in climate science and its related disciplines. Examples of research in this space include:

- Software engineering tools/techniques for the development and optimization of earth system models. A particular challenge here is to make these models more readable, maintainable and portable, to accelerate the process of getting scientific ideas into working code, but without compromising their value as scientific instruments.
- Data management for data-intensive science. A challenge here is to provide the appropriate meta-data and semantics for end-users of climate data to understand what data exists, how it was obtained, what processing has been done to it, and what assumptions and limitations there are on the data.
- Open notebook science – the use of electronic notebooks and workflow tools for making research processes more transparent and repeatable, linked into social network tools for knowledge finding and mapping sources of expertise.

4.2 Software for collective decision making

The second thrust includes a wide array of information sharing tools to improve public understanding of science, through to decision support at multiple levels: individual, community, government, and inter-governmental. Examples of research in this space include:

- Simulations, games, and educational software to support public understanding of the science. Good visualizations play a central role in communicating the science of climate change to diverse audiences. Yet the scientific simulations are often built without concern for how the results might be communicated with broader audiences, while visualizations developed for non-scientists are often built without good connections to the latest science. Research in this space will bring together the latest science with expertise in visualization and information design, to develop interactive tools for a variety of non-specialist audiences.
- Reputation systems, to create new forms of quality control for web-based information sources. The challenge here is to take traditional processes used in peer-reviewed scientific literature and apply them to information sources and participants in online communities. A related problem is to apply such reputation systems to search engines, to include assessments of credibility and relevant expertise as a filter for search results.

- Collective intelligence tools that make use of crowd-sourcing techniques (e.g. Mechanical Turk, Wikipedia, Yahoo Answers) to improve the quality of evidence and analysis for action on climate change. A particular challenge is to make the steps used in quantitative and qualitative analysis visible and open, to allow massive collaboration, so that a broad community can collaboratively test and tweak the assumptions, see how those assumptions are used in the analysis, and to link steps in the analysis to sources of evidence.
- Decision support tools for carbon accounting, especially for regulatory and corporate decision-making. The challenge here is to bring accurate assessment of fossil fuel use fully into the decision-making process, so that carbon emissions can be managed as a tradeable resource.

4.3 Green IT

The third challenge is the optimization of power consumption of software and all things controlled by software. Such research includes:

- Power aware computing, including better management of power in all computer systems, from mobile devices to cloud computing services and data centers.
- Smart controllers, to optimize and balance power consumption in everything that consumes power, and to provide appropriate feedback to users, to allow them to adjust their use of such devices for greater efficiency.
- Green Software Design, in which sustainability is treated as a first-class requirement in all stages of software development, so that environmental impact of new systems are reduced. A challenge here is to identify ways in which software and human activity intertwine, leading to inflexible systems that reduce the choices available for sustainability.

5. CONCLUSIONS

Climate change is a serious and urgent problem, and it demands a mobilization of effort across many different disciplines. None of the problems we have discussed in this paper can be solved by software alone, but software and computational thinking are critical components of the solution. We have identified three key areas where effort might be focussed: software to support the science of understanding climate change; software to support the global collective decision making; and software to reduce the carbon footprint of modern technology.

More importantly, we have argued for an ongoing discussion of how the skills and knowledge of the software community can be brought to bear in meeting the challenge of climate change. We argued that a massive mobilization of talent will be needed. Other disciplines are already developing disciplinary responses to this challenge (e.g. see [9, 15, 2]). It is time for the software community to step up to the plate.

6. ACKNOWLEDGMENTS

This work was supported by NSERC. Many thanks to Michael Tobis, Neil Ernst, Jorge Aranda, Jennifer Horkoff and Archer Batcheller for comments on earlier drafts.

7. REFERENCES

[1] I. Allison et al. *The Copenhagen Diagnosis, 2009: Updating the World on the Latest Climate Science.* The University of New South Wales Climate Change Research Centre (CCRC), Sydney, Australia, 2009.

[2] D. Campbell-Lendrum, R. Bertollini, M. Neira, K. Ebi, and A. McMichael. Health and climate change: a roadmap for applied research. *The Lancet*, 373:1663 – 1665, 2009.

[3] N. Collins et al. Design and implementation of components in the earth system modeling framework. *Int. J. High Perform. Comput. Appl.*, 19(3):341–350, 2005.

[4] R. Dunlap et al. Earth system curator: metadata infrastructure for climate modeling. *Earth Science Informatics*, 1:131–149, 2008.

[5] S. M. Easterbrook and T. C. Johns. Engineering the software for understanding climate change. *Computing in Science and Engineering*, 11:65–74, 2009.

[6] P. N. Edwards. *A Vast Machine: Computer Models, Climate Data, and the Politics of Global Warming.* MIT Press, Cambridge, Massachusetts, 2010.

[7] J. Hansen, R. Ruedy, M. Sato, and K. Lo. Global surface temperature change. in preparation, 2010.

[8] D. MacKay. *Sustainable Energy - without the hot air.* UIT Press, Cambridge, UK, 2009.

[9] J. Nagel, T. Dietz, and J. Broadbent. *Workshop on Sociological Perspectives on Global Climate Change, May 30-31, 2008.* National Science Foundation, 2009.

[10] M. Parry, O. Canziani, J. Palutikof, P. van der Linden, and C. Hanson, editors. *Contribution of Working Group II to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change.* Cambridge University Press, Cambridge, UK, 2007.

[11] J. Pipitone. Software quality in climate modelling. Master's thesis, Department of Computer Science, University of Toronto, 2010.

[12] V. Ramanathan and Y. Feng. On avoiding dangerous anthropogenic interference with the climate system: Formidable challenges ahead. *Proc. of the Nat. Acad. of Sciences*, 105(38):14245–14250, 2008.

[13] S. Reed. Oxburgh report clears controversial climate research unit. *Science*, 328:415, 2010.

[14] J. Slingo et al. Developing the next-generation climate system models: challenges and achievements. *Philosophical Transactions of the Royal Society A*, 367:815–831, 2009.

[15] J. Swim et al. *Psychology & Global Climate Change.* American Psychological Association, 2009.

[16] The Climate Group. *SMART 2020: Enabling the low carbon economy in the information age.* The Global eSustainability Initiative, Brussels, Belgium, 2008.

[17] G. M. Weinberg. *An Introduction to General Systems Thinking.* Dorset House, 2001.

[18] J. Wing. Computational thinking. *Communications of the ACM*, 49:33–35, 2006.