Computing for Global Development: Is it Computer Science Research?

Kentaro Toyama
Microsoft Research India
kentoy@microsoft.com

Muneeb Ali
Princeton University
muneeb@princeton.edu

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

Poverty and the associated sufferings remain a global challenge, with over a billion people surviving on less than a dollar a day. Technology, applied appropriately, can help improve their lives. Despite some clear examples of technical research playing a key role in global development, there is a question that repeatedly arises in this area: *can technologies for developing regions be considered a core area of computer science research?* In this note, we examine some of the arguments on both sides of this question, deliberately avoid answering the question itself (for the lack of community consensus), and provide some suggestions for the case where the answer is in the affirmative.

Categories and Subject Descriptors:
A.1 [Introductory and Survey]

General Terms: Design, Economics, Human Factors

Keywords: Developing Regions, Global Development

1. INTRODUCTION

The last decade has seen interest in applying information and communication technologies for global development. Research in this multidisciplinary area, often abbreviated ICTD, involves a broad range of activities in which commoditized electronic technologies, such as the personal computer and the mobile phone, are utilized for global socio-economic development, particularly in the world's developing countries. ICTD draws interest from multiple disciplines like sociology, economics, political science, engineering, and computer science to name a few. We should differentiate between ICTD and "technical ICTD": The former is broadly interdisciplinary; the latter – for the lack of a better name – refers to the aspects most relevant to computer scientists and engineers. (In this note, we often use ICTD to mean technical ICTD, unless otherwise indicated.)

Although televisions, radios, and landline phones are not outside of the scope of ICTD, more recent technologies, such as personal computers, mobile phones, and wireless networks are what dominate many ICTD projects [1]. Technologies developed for the first world have often been a poor fit in these areas [2], due to issues of cost, infrastructure, physical environment, and social factors, and there is a need for technology research specifically aimed at developing regions. This is the gap that technical ICTD aims to fill. For a more detailed introduction to ICTD, see [1], [2], and [3].

CCR readers are likely to have perked up at the mention of mobile phones and wireless networks, and indeed, there are many opportunities for these and related technologies in developing regions. ICTD also poses new research challenges for systems and networking researchers, given the unique constraints of applications e.g., there have been several streams of work in long-distance WiFi to connect remote rural villages to urban centers or the Internet [4], [5]. The TIER group at UC Berkeley has led much of this work, and as they explicitly prescribe, the challenge lies in leveraging cheap, off-the-shelf hardware, while adapting the network stack to handle long-distance transmissions at reasonable bandwidth [2]. Their work has been successfully deployed by an eye hospital in southern India that uses the technology to allow remote consultations at its rural outreach clinics [4].

Another example is work by the Tetherless Computing Lab at University of Waterloo, in providing a sneaker-net solution that allows remote sites to maintain replicated data and provide email access to disconnected endpoints, assuming intermittent transport of a USB flash drive between them [6]. Although the work is still relatively new, such a system is bound to be useful to any number of non-profit organizations that use computers in branch offices, but have infrastructural or financial difficulty in maintaining high-bandwidth connections.

It is relatively easy to argue that ICTD research can be beneficial for people in developing regions. But is it computer science research? This question is of sufficient interest to a growing number of computer scientists that the Computing Research Association recently sponsored a two-day workshop around exactly this question [7]. Much of what we discuss below reflects our discussions at the workshop, though the ideas expressed here are our own and not necessarily the consensus of the workshop participants.

2. RESEARCH MERIT

In this section we present some arguments for and against the adoption of ICTD research by computer science.

2.1 Why Computer Science?

Perhaps the most convincing case, for considering ICTD as computer science research, is simply that computer science research problems are routinely encountered in ICTD. Over the last few years (ICTD is still relatively young), solutions to these problems have been published in NSDI, SenSys, MobiCom, HotMobile, HotNets, and WWW, just in areas relevant to the SIGCOMM community, and in other computer science areas such as SIGCHI (computer-human interaction) and IJCAI (artificial intelligence). Clearly, some technical ICTD papers have enough scientific contribution that they are published at the "traditional" computer science venues. Is it possible to carve out the technical aspects of ICTD and make it a mainstream research area of computer science? More computer science researchers are needed in ICTD, and it would be a shame if they only pursue it as a side activity, instead of primary research.

ICTD research not only impacts global development, but can also advance "traditional" computer science. Two examples that come to mind are WildNet [4] and HashCache [8]. Although motivated by applications in developing regions, these works fundamentally changed our view that a) 802.11 can be a viable long-distance technology and b) large disks can be efficiently used to index web caches, respectively. An ICTD context can mean different technology and/or cost constraints for exploring a classic research issue. Also, ICTD results may prove useful in other contexts when a) the technology constraints and/or needs shift (e.g., the Indian Simputer bears similarity to today's iPhone [9]) or b) new application of the technology is found in another context e.g., delay tolerant networks can be useful both in ICTD and for space communication in planetary-scale networks.

ICTD researchers are obsessed with real impact and share common methodological perspectives. Most feel that the research cannot be done without significant time spent in "the field", interacting with potential users to understand their environment and constraints, and iteratively testing solutions with target communities. These are unique cultural aspects of ICTD that bind the community together, much in the way of the systems community's insistence on building actual systems or the graphics community's pursuit of beautiful imagery and animations.

Finally, there is a frequently recurring set of challenges and technical problems that repeat across application areas. Much ICTD research is concerned with low cost, power efficiency, and unreliable infrastructure, since money is rarely abundant among users and those who serve them. Issues with unreliable infrastructure mean that systems must be exceptionally robust, and take power outages and intermittent bandwidth as part of normal operation. Also, devices fail often due to high levels of dust and fluctuations in power. Human constraints such as illiteracy or inexperience with technology also place unique constraints on technology, particularly on user interfaces.

2.2 Why Not Computer Science?

It is clear that there are computer science research problems in ICTD, so the question is not whether individual ICTD projects contain computer science problems. Rather, it is a question of whether the combination of problems has some sort of conceptual coherence, and whether together they constitute a unique area of computer science not adequately covered by existing subareas. The issue is difficult to articulate in the abstract, so we note three reasons why ICTD research, taken as a whole, is often *not* considered mainstream computer science.

First, ICTD is driven by application, by definition "for development". In the informal hierarchy of computer science, less technical and more socially-applied areas like human-computer interaction (HCI) are often relegated to the lower levels and have historically struggled to gain acceptance as core computer science research. In ICTD, computer science is applied to agriculture, education, microfinance, healthcare, political activism, livelihood enhancement, and so on. ICTD succeeds not based on the merit of the technology it generates, but based on its impact on the application domain. In the end, what matters is crops saved, illnesses cured, children taught, and employment secured – not the typical concerns of most computer scientists, at least in their capacity as researchers.

Second, and related, ICTD is thoroughly interdisciplinary, and interdisciplinarity is also viewed suspiciously by academic disciplines that believe themselves to be pursuing "pure" research in a well-circumscribed field. One issue with interdisciplinary work is that problems seen to be legitimate, even crucial, to the area often don't contain enough research content in any one area to satisfy the contributing disciplines. Also, even if there is enough research content in one area, solving that part of the problem may be only a small portion of the larger problem being addressed. As is the case with some work in systems research, its not clear that a core technical contribution is really valuable without building the whole system. Unlike systems research, however, "the whole system" might include non-technical components requiring social, cultural, economic, and political efforts, as well.

Consider the problem of collecting healthcare data across a heterogeneous geography through a digital medium, an oft-cited problem that remains unsolved in the general case. Solving the problem is likely to require a mix of economics, political science, sociology, medicine, engineering, and computer science. But, the computer science part of the work might only be incremental tweaks to known algorithms in distributed systems, mobility, security, privacy, data aggregation, and human-computer interaction. Moreover, a growing body of evidence in the ICTD community suggests that technical problems are a small fraction of a complete solution – technology is not the bottleneck. It might turn out, for example, that setting up the right incentives has much more impact than devising the perfect communications protocol. Again, the individual researcher not only needs practical impact, but research contribution for advancing her career.

Third, ICTD like some other application subfields lacks a clear definition of generic technical issues within a well-circumscribed context. Just in agriculture applications, there are networking problems (e.g., connecting remote villages to urban experts), speech problems (e.g., for building a Q&A system in multiple dialects), information-retrieval problems (e.g., permitting cross-lingual, geography-relevant database queries), computer-vision problems (e.g., diagnosing diseased crops via photographs), and so on. Furthermore, the technical content of each such problem is firmly within existing subareas of computer science.

With these counts against ICTD research, it's not surprising that the area finds it difficult to gain a foothold within some computer science departments.

3. CRITICAL FACTORS

Next, we consider some critical factors that may help establish ICTD in computer science circles, without necessarily arguing that this should happen.

3.1 Acceptance as Applied Science

Ironically, ICTD is struggling to establish itself within a field that has itself had a history of struggling to establish itself, namely computer science. With its roots in Goedel's Incompleteness Theorem and the Church-Turing Thesis, computer science was for many decades considered a subarea of mathematics, and applied mathematics. Howard Aiken, inventor of the Mark I computer, considered by many to be the first universal computer, faced tremendous challenges at Harvard University in establishing a department of computer science [10]. No doubt, detractors felt that computer science was too applied and too interdisciplinary to be elevated to its own area of inquiry. Computer science came into its own only after a rise in the number of exceptional thinkers in the area, as well as with the dramatic growth in its own relevance to society in general.

The analogy to computer science is apt, but perhaps not one that computer science departments themselves are willing to accept as a rationale for ICTD. So, instead, it's worth looking for other analogies that are more relevant.

We propose that the most relevant analogies are those of areas such as bioinformatics, computational physics, or education technologies, where computer science is in service of the goals of another discipline. Bioinformatics, in particular, appears to have had much success not only in becoming a mainstream computer science research area, but an "official" computer science subarea (for example, computer science textbooks routinely list it among CS sub-areas [11], [12]). We believe that following the footsteps of bioinformatics, can help ICTD establish itself within computer science.

3.2 Name and Definition

Names are important. "Computer science for biology" hasn't caught on, but "bioinformatics" and "computational biology" have. The ICTD research community currently lacks such an identity. ("ICTD" is what we use in this article, but it generally refers to the more interdisciplinary works. Technical ICTD doesn't yet have a unique moniker.) It doesn't help that the area is not easy to describe in a short phrase – the "D" for "development" is itself confusing in a computer science context where the word usually means software development. Yet, it's hard to avoid the word in the context of developing countries, global development, and socio-economic development. Some suggestions for a name for technical ICTD are "computing for global development," "global computing," and so on - none of them have unanimous support from the community. The sooner we converge on a simple, descriptive, and catchy name the better.

Converging on a standard name naturally leads us to the next question. What exactly is ICTD research? What is the line between research and philanthropy? What separates technical ICTD from more interdisciplinary ICTD? Here, we are not asking the more important question of what is good ICTD research, but rather a more primitive one concerning the boundaries of the subarea. Drawing these lines is non-trivial, given the multi-disciplinary nature of this space. One approach is to start defining what is *not* ICTD research.

3.3 Faculty Focus

In the university context, what is ultimately required for a field to become viable is simply enough faculty conducting research in the area. With faculty involvement, there will be research, papers, workshops and conferences, and support for graduate work (or at least, a determined effort to get it). And with critical mass comes more faculty involvement, as faculty influence hiring and departmental direction.

But, faculty focus is not only the end goal, but also one of the paths to legitimacy as a research area. That is, with more faculty who declare themselves to be working on ICTD, the goal of ICTD as a mainstream computer science research activity becomes closer. What does it take to achieve this? Actually, it seems very simple; we want to encourage (a) more existing CS faculty to engage in ICTD research, (b) more existing CS faculty to declare ICTD as their only or primary area of research, and (c) more new faculty in ICTD to be hired into CS departments.

Of these, engaging more faculty seems easiest to achieve through collaborations. If those of us doing research in ICTD engage other CS faculty, we can increase the number of faculty interested in the effort. Bioinformatics appears to have benefited greatly from significant collaborative pressure coming from the application domain; biologists wanted more input from computer scientists.

It will be encouraging to see existing, especially tenured, faculty declaring ICTD as their primary area of research. The more tenured faculty take interest, and proclaim ICTD as their primary area, the more the field gains validity.

Hiring is perhaps the area in which we have the least control, because there's a chicken-and-egg challenge; without the critical mass of ICTD-sympathetic faculty, ICTD faculty hires are less likely. Furthermore, ICTD research cannot be considered as mainstream computer science until junior faculty can get tenure based just on their ICTD research.

3.4 Publication Outlets

For a subfield to be recognized, a close second to faculty involvement is a quality publication outlet for research articles. As of this writing, technical ICTD lacks a flagship publication venue. It's not entirely clear whether there is a critical mass of strong research to merit a technical conference, but workshops such as NSDR [13] (held with SIGCOMM and SOSP) have been successful, and special tracks at WWW [14] and CHI [15] have shown that there is interest in other CS communities. As we look forward, again, there might be a chicken-and-egg challenge. You need a high-caliber conference to build a community, but you need critical mass for justifying a separate technical conference.

Furthermore, feedback from ICTD partners in healthcare and other areas suggests that journals are far more important, than conferences, in areas outside of computer science. They need journal publications for advancing their careers and currently technical ICTD lacks quality journals.

3.5 Funding

It's a fact of life that research agendas at universities are influenced by the availability of funds. If funding agencies like NSF or DARPA are not interested in funding a particular area, researchers will be limited in their ability to work in it. Currently, a common approach is to find alternate motivations, for similar research, to convince funding agencies e.g., by motivating a remote communication protocol in a military context instead. Identifying atypical funding resources and pushing the case for ICTD funding in front of government agencies is critical for the future of ICTD.

Luckily, such sources exist. Global development, despite its focus on the poor, is an enterprise that doesn't necessarily lack for money. Multilateral entities such as the World Bank and the United Nations, bilateral aid agencies such as Canada's International Development Research Center, and large foundations such as the Ford Foundation or the Gates Foundation are all interested in good research. To the extent that ICTD researchers can learn to navigate a different set of funding organizations, ICTD may turn out to be at least as well-funded as traditional areas of computer science.

3.6 Challenging Technical Problems

It's surprisingly difficult to find hard, technical problems that are unique to ICTD – often, the technical challenges are generic computer science research problems (e.g., better speech recognition). The portion that is relevant for ICTD is often limited to adaptation (e.g., what's the best way to train speech recognition engines quickly in local dialects?). It's not that challenging technical problems don't exist in ICTD, it's that they're often not obvious. To attract the best talent, you need technically hard problems that will not only impact global development, but also computer science.

Much of bioinformatics consists of string-matching, machine learning, and indexing problems that are well within the domain of other computer science sub-areas. It's not obvious at first glance that there are unique problems posed by biology as a domain. Yet, perhaps because these problems are acknowledged to be difficult technical problems that the area has gained acceptance. It would be helpful to understand how bioinformatics was first perceived by conservative computer science faculty in its early days.

3.7 Metrics and Tools

Common metrics can often unite a field. Technical ICTD does not currently have agreed-upon metrics. Are we trying to optimize the cost of bits transmitted? Are we increasing the useful computations performed per watt of power? Or, are we trying to decrease non-technical metrics such as disability-adjusted life years?

Also, technical ICTD lacks processes and tools. There is hardly any data on network conditions, specifications of typical computing devices, or limitations of available resources like memory and power. Current simulation environments don't accurately model conditions in developing regions (e.g., power characteristics and availability).

3.8 Increased General Interest

There is a direct correlation between public awareness, industry growth, increased funding, curious students, and general interest in university research. Certainly this was true for both the growth of computer science and bioinformatics. In effect, if the need for applied research is great enough, universities are willing to meet it. ICTD faculty may be able to attract really strong students – even to get students who would normally attend a higher-ranked school – because many students are truly passionate about the ICTD topic.

ICTD projects have enjoyed considerable coverage in the media recently (e.g., [16]). However, this coverage has not yet translated into concrete benefits for the ICTD community. We need a combination of ongoing PR about the field, to highlight successes in a way that it registers in the public mind and draws attention from faculty and funding agencies.

3.9 Patience

Many current ICTD researchers feel the challenges of pursuing this area today. However, ICTD as a research area is still very young – in fact, apart from a handful of isolated projects, it's hard to argue that technical ICTD research has been happening for more than six to seven years. Given such a short history, it might just be patience that is required.

Bioinformatics as a term was coined as early as 1978 [17], yet it hadn't blossomed as a field of research until relatively recently, with the mid-1990s just beginning to see wider spread interest. That's nearly twenty years from early conception to established subarea.

4. CONCLUSION

We considered the reasons for and against technical ICTD research gaining acceptance in computer science circles.

For those seeking to establish ICTD as a core research area within computer science, one question that arises is what precedent best applies. We argued that the best analogies are those applied subareas of computer science where computer science is clearly in service of another discipline, with bioinformatics being a key example. One route to success, therefore, might be to understand how bioinformatics gained a respectable reputation within computer science, as a way to light the path for technical ICTD. Among the issues to consider are name and definition, faculty focus, publication outlets, funding, unique technical challenges, new metrics and tools, and general public awareness.

Acknowledgements

We thank Jen Rexford for her feedback on an earlier draft.

5. REFERENCES

[1] K. Toyama and M. B. Dias, "Information and communication technologies for development," *IEEE Computer*, vol. 41, no. 6, pp. 22–25, June 2008.

[2] E. Brewer, M. Demmer, M. Ho, R. J. Honicky, J. Pal, M. Plauche, and S. Surana, "The challenges of technology research for developing regions," *IEEE Pervasive Computing*, vol. 5, no. 2, pp. 15–23, April-June 2006.

[3] M. B. Dias and E. Brewer, "How computer science serves the developing world," *Communications of the ACM*, vol. 52, no. 6, pp. 74–80, 2009.

[4] R. Patra, S. Nedevschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer, "WiLDNet: Design and implementation of high performance WiFi based long distance networks," in *USENIX NSDI 2007*, April 2007, pp. 87–100.

[5] B. Raman and K. Chebrolu, "Design and evaluation of a new MAC protocol for long-distance 802.11 mesh networks," in *ACM MobiCom 2005*, Aug-Sep 2005, pp. 156–169.

[6] "Vlink," 2009. [Online]. Available: http://blizzard.cs.uwaterloo.ca/tetherless/index.php/VLink

[7] "CCC Workshop on Computer Science and Global Development," August 2009. [Online]. Available: http://www.cra.org/ccc/globaldev.php

[8] A. Badam, K. Park, V. S. Pai, and L. L. Peterson, "HashCache: Cache storage for the next billion," in *USENIX NSDI 2009*, Boston, MA, April 2009, pp. 123–136.

[9] M. Ali and K. Langendoen, "TinyPC: Enabling low-cost internet access in developing regions," in *ACM NSDR 2007*, Kyoto, Japan, Aug. 2007, p. 177182.

[10] I. B. Cohen, G. W. Welch, and R. V. D. Campbell, Eds., *Makin' Numbers: Howard Aiken and the Computer.* MIT Press, 1999.

[11] N. B. Dale and J. Lewis, *Computer Science Illuminated.* Jones and Bartlett Publishers, Inc., 2002.

[12] D. Reed, *A Balanced Introduction to Computer Science.* Prentice-Hall, Inc., 2004.

[13] "ACM Workshop on Networked Systems for Developed Regions." [Online]. Available: http://nsdr.dritte.org/

[14] "Int'l World Wide Web Conference." [Online]. Available: http://www.iw3c2.org/conferences

[15] "CHI." [Online]. Available: http://sigchi.org/conferences/

[16] "MIT TechReview: 10 Emerging Technologies 2009." [Online]. Available: http://www.technologyreview.com/specialreports

[17] P. Hogeweg, "Simulating the growth of cellular forms," *Simulation*, vol. 31, no. 3, pp. 90–96, 1978.