Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2010 Symposium

ISBN
978-0-309-16362-0

190 pages
6 x 9
PAPERBACK (2010)

National Academy of Engineering

FRONTIERS OF ENGINEERING

Reports on Leading-Edge Engineering from the 2010 Symposium

NATIONAL ACADEMY OF ENGINEERING
OF THE NATIONAL ACADEMIES

THE NATIONAL ACADEMIES PRESS
Washington, D.C.
www.nap.edu

THE NATIONAL ACADEMIES PRESS • 500 Fifth Street, N.W. • Washington, DC 20001

NOTICE: This publication has been reviewed according to procedures approved by a National Academy of Engineering report review process. Publication of signed work signifies that it is judged a competent and useful contribution worthy of public consideration, but it does not imply endorsement of conclusions or recommendations by the NAE. The interpretations and conclusions in such publications are those of the authors and do not purport to represent the views of the council, officers, or staff of the National Academy of Engineering.

Funding for the activity that led to this publication was provided by IBM, The Grainger Foundation, Air Force Office of Scientific Research, Defense Advanced Research Projects Agency, Department of Defense-DDR&E Research, Microsoft Research, and Cummins Inc. This material is also based upon work supported by the National Science Foundation under Grant No. 0903867. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

International Standard Book Number-13: 978-0-309-16362-0
International Standard Book Number-10: 0-309-16362-5

Additional copies of this report are available from The National Academies Press, 500 Fifth Street, N.W., Lockbox 285, Washington, DC 20055; (800) 624-6242 or (202) 334-3313 (in the Washington metropolitan area); Internet, **http://www.nap.edu**

Printed in the United States of America

ORGANIZING COMMITTEE

ANDREW M. WEINER (*Chair*), Scifres Family Distinguished Professor of Electrical and Computer Engineering, Purdue University
ALI BUTT, Assistant Professor, Department of Computer Science, Virginia Tech
MARK BYRNE, Mary & John H. Sanders Associate Professor, Department of Chemical Engineering, Auburn University
DILMA DA SILVA, Research Staff Member, Advanced Operating Systems Group, IBM T.J. Watson Research Center
DANIEL ELLIS, Associate Professor, Department of Electrical Engineering, Columbia University
MICHEL INGHAM, Technical Group Supervisor, Flight Software Systems Engineering and Architectures Group, Jet Propulsion Laboratory
YOUNGMOO KIM, Assistant Professor, Department of Electrical and Computer Engineering, Drexel University
JACOB LANGELAAN, Assistant Professor, Department of Aerospace Engineering, Pennsylvania State University
BABAK PARVIZ, Associate Professor, Department of Electrical Engineering, University of Washington

Staff

JANET R. HUNZIKER, Senior Program Officer
ELIZABETH WEITZMANN, Program Associate

Preface

This volume highlights the papers presented at the National Academy of Engineering's 2010 U.S. Frontiers of Engineering Symposium. Every year, the symposium brings together 100 outstanding young leaders in engineering to share their cutting-edge research and technical work. The 2010 symposium was held September 23–25, and hosted by IBM at the IBM Learning Center in Armonk, New York. Speakers were asked to prepare extended summaries of their presentations, which are reprinted here. The intent of this book is to convey the excitement of this unique meeting and to highlight cutting-edge developments in engineering research and technical work.

GOALS OF THE FRONTIERS OF ENGINEERING PROGRAM

The practice of engineering is continually changing. Engineers today must be able not only to thrive in an environment of rapid technological change and globalization, but also to work on interdisciplinary teams. Cutting-edge research is being done at the intersections of engineering disciplines, and successful researchers and practitioners must be aware of developments and challenges in areas that may not be familiar to them.

At the 2-1/2–day U.S. Frontiers of Engineering Symposium, 100 of this country's best and brightest engineers, ages 30 to 45, have an opportunity to learn from their peers about pioneering work being done in many areas of engineering. The symposium gives early career engineers from a variety of institutions in academia, industry, and government, and from many different engineering disciplines, an opportunity to make contacts with and learn from individuals they would not meet in the usual round of professional meetings. This networking may lead to collaborative work and facilitate the transfer of new techniques and approaches. It is hoped that the exchange of information on current developments in many fields of engineering will lead to insights that may be applicable in specific disciplines and thereby build U.S. innovative capacity.

The number of participants at each meeting is limited to 100 to maximize opportunities for interactions and exchanges among the attendees, who are chosen through a competitive nomination and selection process. The topics and speakers for each meeting are selected by an organizing committee of engineers in the same 30- to 45-year-old cohort as the participants. Different topics are covered each year, and, with a few exceptions, different individuals participate.

Speakers describe the challenges they face and communicate the excitement of their work to a technically sophisticated but non-specialized audience. Each speaker provides a brief overview of his/her field of inquiry; defines the frontiers of that field; describes experiments, prototypes, and design studies that have been completed or are in progress, as well as new tools and methodologies, and limitations and controversies; and summarizes the long-term significance of his/her work.

THE 2010 SYMPOSIUM

The four general topics covered at the 2010 meeting were: cloud computing, engineering and music, autonomous aerospace systems, and engineering inspired by biology. The Cloud Computing session described how this disruptive technology changes the way users design, develop, deploy, utilize, and disseminate applications and data. Following an overview presentation on the potential of cloud computing, there were talks on the challenges of providing transparent interfaces to the users while maintaining massive scale, developing robust cloud applications, and the environmental ramifications of cloud computing.

Technology has strongly influenced music since the first musical instruments and continues to do so in a variety of ways. In the Engineering and Music session, presentations covered advances in very large-scale music information retrieval, non-mainstream ways that people outside the engineering community are using technology to create music, the use of laptop computers in collaborative live performance, and utilizing mathematics to analyze and better understand music as well as incorporating mathematical representations into visualizations for live performance.

Autonomous Aerospace Systems was the focus of the third session, which included presentations on techniques for enabling "intelligence" in autonomous systems through probabilistic models of the environment and the integration of human operators in the control/planning loop, challenges for automation posed by NASA's current and future space missions, the role of health awareness in systems of multiple autonomous vehicles, and automation and autonomy in the deployment of the next generation air transportation system.

The symposium concluded with the session Engineering Inspired by Biology, which highlighted the diverse role biology is playing in contemporary engineering. Talks focused on engineering challenges in the analysis of genetic variation, gene expression, and function; engineering biomimetic peptides for targeted drug delivery; and using biomolecules for actuation as motor-powered devices within systems.

In addition to the plenary sessions, the participants had many opportunities to engage in informal interactions. On the first afternoon of the meeting, participants broke into small groups for "get-acquainted" sessions during which individuals presented short descriptions of their work and answered questions from their colleagues. This helped attendees get to know more about each other relatively early in the program. On the second afternoon, there were tours of the IBM T.J. Watson Lab in Yorktown Heights and the IBM Industry Solutions Lab in Hawthorne.

Every year, a distinguished engineer addresses the participants at dinner on the first evening of the symposium. The speaker this year was Dr. Bernard S. Meyerson, vice president for innovation at IBM, who gave a talk on the topic, *Radical Innovation to Create a Smarter Planet*.

NAE is deeply grateful to the following organizations for their support of the 2010 U.S. Frontiers of Engineering Symposium: IBM, The Grainger Foundation, Air Force Office of Scientific Research, Defense Advanced Research Projects Agency, Department of Defense-DDR&E Research, National Science Foundation, Microsoft Research, and Cummins Inc. NAE would also like to thank the members of the Symposium Organizing Committee (p. iv), chaired by Dr. Andrew M. Weiner, for planning and organizing the event.

Contents

CLOUD COMPUTING

ENGINEERING AND MUSIC

AUTONOMOUS AEROSPACE SYSTEMS

ENGINEERING INSPIRED BY BIOLOGY

APPENDIXES

CLOUD COMPUTING

Introduction

Ali R. Butt
Virginia Tech

Dilma Da Silva
IBM Research

Cloud computing is emerging as a disruptive technology that will change the way users, especially scientists and engineers, design, develop, deploy, use, and disseminate their applications and data. By decoupling lower-level computer system details from application development, and freeing users to focus on their technical and scientific missions, cloud computing is likely to have a profound impact on our lives.

Computer-based simulations and applications are considered a "third-pillar" of scientific discovery, which complements the traditional pillars of theory and experimentation. Currently, these simulations and applications, which require significant investment in the acquisition and maintenance of system infrastructure, are used only by seasoned computer scientists. Cloud computing promises to lower the entry barrier and allow for the easy integration of knowledge gained from scientific observation and for predictions of future responses or outcomes.

The speakers in this session highlight some recent advances in technologies that are shaping the modern cloud-computing paradigm. Their talks cover a wide range of "cloud aspects," from designing innovative computer systems to how such systems can be used and configured in an energy-efficient way.

Armando Fox (UC-Berkeley) begins with an overview of how next-generation clouds should look. Based on user feedback and a survey of requirements, he discusses the major trends as computer scientists work toward realizing future clouds and making them amenable to wide-scale use and adaptation, enabling the democratization of supercomputing. Next, Luiz Andre Barroso (Google) describes the basics of cloud computing—how such systems are realized, the challenges to providing transparent interfaces to users while maintaining unfathomable scale, and support for user applications in a seamless, world-wide “supercomputer” (i.e., the cloud).

In the third talk, YY Zhou (UC-San Diego) describes the challenges of building robust applications in the cloud. Finally, Parthasarathy Ranganathan (HP Labs) describes the environmental and energy implications of using hundreds of thousands of computing nodes at a central location. He also discusses how building architecture and software design can be done in ways that reduce the carbon footprint of the supporting cloud infrastructure.

Opportunities and Challenges of Cloud Computing

Armando Fox
University of California, Berkeley

Computer science is moving forward so quickly and is so focused on its recent history that we are often surprised to learn that visionary ideas were articulated long before the technology for their practical implementation was developed. The following vision of "utility computing" is excerpted from an overview of the pioneering and highly influential MULTICS computing system (Corbató and Vyssotsky, 1965):

> One of the overall design goals is to create a computing system which is capable of meeting almost all of the present and near-future requirements of a large computer utility. Such systems must run continuously and reliably 7 days a week, 24 hours a day in a way similar to telephone or power systems, and must be capable of meeting wide service demands . . . [T]he importance of a multiple access system operated as a computer utility is that it allows a vast enlargement of the scope of computer-based activities, which should in turn stimulate a corresponding enrichment of many areas of our society.

Today, 45 years later, that vision appears close to becoming reality. In 2008, Amazon announced the availability of its Elastic Compute Cloud (EC2), making it possible for anyone with a credit card to use the servers in Amazon's datacenters for 10 cents per server hour with no minimum or maximum purchase and no contract (Amazon AWS, 2008b). Amazon has since added options and services and reduced the base price to 8.5 cents per server hour.) The user is charged for only as long as he/she uses the computer rounded up to the next hour.

The essence of cloud computing is making datacenter hardware and software available to the general public on a pay-as-you-go basis. Every user enjoys the illusion of having virtually infinite capacity available instantaneously on demand. Hence the term *utility computing* is used to describe the "product" sold by a cloud-computing provider.

Of course, by 2008, many companies, such as Google Search and Microsoft Hotmail, were already operating extensive "private clouds" that delivered proprietary SaaS (software as a service). These companies had found it necessary to develop the programming and operational expertise to run such installations.

In contrast, EC2 was the first truly low-cost utility computing that was not bundled with a particular SaaS application. Users of EC2 were allowed to deploy applications of their choice, which greatly increased the popularity of the system. Private-cloud operators Google and Microsoft soon followed suit and now provide public-cloud services in addition to their proprietary services.

At first, skeptics were hard pressed to believe that Amazon could operate such a service at a profit. But, as leading software architect James Hamilton observed (2008), because of economies of scale, the costs of bandwidth, storage, and power for warehouse-scale datacenters are five to seven times cheaper than for medium-sized datacenters (see Table 1). With Amazon's retail-to-consumer operational expertise, the company found a profitable way to pass these savings along to individual users.

COST ASSOCIATIVITY AND ELASTICITY

The cloud-computing service model, which represents a radical departure from conventional information technology (IT), enables *fundamentally new* kinds of computation that were previously infeasible. For example, in 2008, the National Archives released 17,481 pages of documents, including First Lady Hillary Clinton's daily schedule of activities. Peter Harkins, a senior engineer at *The Washington Post*, using 200 computers in EC2 for less than nine hours, produced a searchable corpus of the documents and made it publicly available on the World Wide Web less than a day later (Amazon AWS, 2008b). The server time cost

TABLE 1 Comparative Economies of Scale in 2006 for a Medium-Sized Datacenter (~1,000 servers) and a Warehouse-Scale Datacenter (~50,000 servers)

Technology	Medium-Sized Data Center	Warehouse-Scale Data Center	Ratio
Network	\$95 per Mbit/sec/month[a]	\$13 per Mbit/sec/month	7.1
Storage	2.20 per GByte/month[b]	\$0.40 per GByte/month	5.7
Administration	1 administrator per ≈140 servers	1 administrator for > 1,000 servers	7.1

[a] Mbit/sec/month = megabit per second per month.
[b] GByte/month = gigabyte per month.
Source: Hamilton, 2008.

Harkins less than $150—the same cost as using a single server for 1,800 hours, and far less than the cost of purchasing a single server outright. Being able to use 200 servers for nine hours for the same price as using one server for 1,800 hours is an unprecedented new capability in IT that can be called *cost associativity*.

That same year, 2008, programmers at the Web startup company Animoto developed an application to create music videos from a user's photo collection. When that application was made available to the more than 200 million users of Facebook, it became so popular so quickly that the number of users doubled every 12 hours for the next three days, causing the number of servers to increase from 50 to 3,500. After the peak subsided, demand fell to a much lower level, and the unnecessary servers were released.

Elasticity, the ability to add and remove servers in minutes, rather than days or weeks, is also unprecedented in IT. Elasticity is financially appealing because it allows actual usage to closely track demand on an hour-by-hour basis, thereby transferring the *risk* of making a poor provisioning decision from the service operator to the cloud-computing provider.

But elasticity is even more important for handling *spikes* and *data hot spots* resulting from unexpected events. During the terrorist attacks of September 11, 2001, for example, viewer traffic on the CNN website increased by an order of magnitude in just 15 minutes (LeFebvre, 2001). In another case, when entertainer Michael Jackson died unexpectedly in 2009, the number of Web searches about Jackson spiked to nearly 10 times the average so suddenly that Google initially mistook the event for a malicious attack on its search service.

According to Tim O'Reilly, founding editor of O'Reilly Media, a leading technical publisher, the ability to deal with sudden surges is particularly important for mobile applications that "respond in real time to information provided either by their users or by non-human sensors" (quoted in Siegele, 2008). In other words, these services are accessible to the more than 50 percent of the world population equipped with cell phones, the most ubiquitous Internet access devices.

OPPORTUNITIES AND CHALLENGES

Scaling Down

Before the advent of cloud computing, scaling up was considered a permanent change, because it usually meant buying and installing new hardware. Consequently, extensive research was conducted on scaling up systems without taking them offline. The idea of subsequently scaling them down—and then possibly back up again—was not even considered.

Since cloud computing involves borrowing machines from a shared pool that is constantly upgraded, scale-up and scale-down are likely to mean that hardware will be more heterogeneous than in a conventional datacenter. Research is just beginning on software, such as scalable consistency-adjustable data storage

(SCADS), which can gracefully scale down as well as up in a short time (Armbrust et al., 2009).

At the other extreme, fine-grained pricing may enable even cheaper utility computing during demand troughs. California power companies have already introduced demand-based pricing models in which power is discounted during off-peak times. By analogy, Amazon EC2 has introduced a new mechanism whereby otherwise unused machines are made available at a discounted rate on a "best-effort" basis. However, the user might be forced to give up the machine on short notice if demand increases and a priority customer is willing to pay a premium for it.

This leads to a relatively new situation of clusters whose topologies and sizes can change at any time and whose cycles may be "reclaimed" on short notice for higher priority applications. Research on scheduling frameworks, such as Mesos, is addressing how applications on cloud computing can deal gracefully with such fluctuations (Hindman et al., 2010).

The ability to scale down also introduces new motivations for improving the energy efficiency of IT. In traditional research proposals, energy costs are usually absorbed into general institutional overhead. With cloud computing, a customer who uses fewer machines consumes less energy and, therefore, pays less. Although warehouse-scale datacenters are now being built in locations where cheaper power (e.g., hydroelectric power) is available (Table 2), the pay-as-you-go model of cloud computing introduces a direct financial incentive for cloud users to reduce their energy usage.

Several challenges, however, may interfere with this opportunity for "greener" IT. Unfortunately, today's servers consume nearly half as much energy when they are idle as when they are used. Barroso and Hölzle (2007) have argued that we will need design improvements at all levels, from the power supply to energy-aware software, to achieve "energy proportional" computing in which the amount of energy consumed by a server is proportional to how much work it does.

TABLE 2 Price of Kilowatt Hours (kWh) of Electricity

Cents per kWh	Region	Factors
3.6	Idaho	Hydroelectric power; no long-distance transmission
10.0	California	Long-distance transmission; limited transmission lines in Bay Area; no coal-fired electricity allowed in the state
18.0	Hawaii	Fuel must be shipped to generate electricity

Source: EIA, 2010.

Better and Faster Research

Cost associativity means that "embarrassingly parallel" experiments—experiments that require many trials or tasks that can be pursued independently—can be accelerated to the extent that available cloud resources allow. For example, an experiment that requires 100,000 trials of one minute each would take more than two months to complete on a single server. Cost associativity makes it possible to harness 1,000 cloud servers for two hours for the same cost. Researchers in the RAD Lab working on datacenter scale computing now routinely run experiments involving hundreds of servers to test out their ideas at realistic scale. Before cloud computing, this was impossible for any university laboratory.

Tools like Google's MapReduce (Dean and Ghemawat, 2004) and the open-source equivalent, Hadoop, give programmers a familiar data-parallel "building block" and encapsulate the complex software engineering necessary for handling the challenges of resource scheduling and responding to machine failures in the cloud environment. However, because many problems cannot be easily expressed as MapReduce tasks, other frameworks, such as Pig, Hive, and Cascading, have emerged that provide higher level languages and abstractions for cloud programming.

Indeed, Amazon's recently-introduced "Elastic MapReduce" service, which provides a "turnkey" version of the MapReduce framework, allows jobs to be written using not only those frameworks, but also statistical modeling packages, such as R. On the level of cloud infrastructure itself, the goal of the Berkeley BOOM project (*boom.cs.berkeley.edu*) is to simplify the creation of new cloud programming frameworks by applying principles from declarative networking.

Progress is being made on all of these fronts, and some new systems are in regular use in production environments. However, the artifacts and ecosystem comprising them are still a long way from "turnkey" systems that will allow domain-expert programmers to seamlessly combine the abstractions in their applications.

HIGH-PERFORMANCE COMPUTING

The scientific and high-performance computing (HPC) community has recently become more interested in cloud computing. Compared to SaaS workloads, which rely on request-level parallelism, HPC workloads typically rely on thread- or task-level parallelism, making them more communication-intensive and more sensitive to communication latency. These properties make HPC workloads particularly vulnerable to "performance noise" artifacts introduced by the pervasive use of virtualization in cloud environments (Armbrust et al., 2010b).

Legacy scientific codes often rely on resource-scheduling approaches, such as gang scheduling and make assumptions about the network topology that connects the servers. Such design decisions make sense in a statically provisioned environment but not for cloud computing. Thus, not surprisingly, early benchmarks of existing HPC applications on public clouds were not encouraging (Evangelinos and Hill, 2008; Walker, 2008).

However, cloud providers have been quick to respond to the potential HPC market, as illustrated by Amazon's introduction in July 2010 of "Cluster Compute Instances" tuned specifically for HPC workloads. Experiments at the National Energy Research Scientific Computing (NERSC) Laboratory at Lawrence Berkeley Laboratory measured an 8.5X performance improvement on several HPC benchmarks when using this new type of instance compared to conventional EC2 instances. Amazon's own measurements show that a "virtual cluster" of 880 HPC instances can run the LINPACK linear algebra benchmark faster than the 145th-fastest supercomputer in the world, as measured by Top500.com. These results have encouraged more scientists and engineers to try cloud computing for their experiments. Installations operated by academic/industrial consortia, such as the Google/IBM/NSF CluE cluster that runs Hadoop (NSF, 2009), Yahoo's M45 cluster (*http://labs.yahoo.com/Cloud_Computing*), and OpenCirrus (*opencirrus.org*), are other examples of cloud computing for scientific research.

Even if the running time of a problem is slower on cloud computing than on a dedicated supercomputer, the total time-to-answer might still be shorter with cloud computing, because unlike traditional HPC facilities, the user can provision a "virtual supercomputer" in the cloud instantly rather than waiting in line behind other users (Foster, 2009).

Longtime HPC veteran Dan Reed, now head of the eXtreme Computing Group (XCG) at Microsoft Research, also believes cloud computing is a "game changer" for HPC (West, 2009). He points out that while cloud infrastructure design shares many of the challenges of HPC supercomputer design, the much larger volume of the cloud infrastructure market will influence hardware design in a way that traditional HPC has been unable to do.

TRANSFERS OF BIG DATA

According to Wikipedia, the Large Hadron Collider could generate up to 15 petabytes (15×10^{15} bytes) of data per year, and researchers in astronomy, biology, and many other fields routinely deal with multi-terabyte (TB) datasets. A boon of cloud computing is its ability to make available tremendous amounts of computation on-demand with large datasets. Indeed, Amazon is hosting large public datasets for free, perhaps hoping to attract users to purchase nearby cloud computing cycles (Amazon AWS, 2008a).

The key word here is *nearby*. Transferring 10 TB over a network connection at 20 megabits per second—a typical speed observed in measurements of long-haul bandwidth in and out of Amazon's S3 cloud storage service (Garfinkel, 2007)—would take more than 45 days and incur transfer charges of $100 to $150 per TB.

In the overview of cloud computing by Armbrust et al. (2010b), we therefore proposed a service that would enable users to instead ship crates of hard drives containing large datasets overnight to a cloud provider, who would physically incorporate them directly into the cloud infrastructure. This idea was based on experience with this method by the late Jim Gray, the Turing Award-winning computer scientist who was recently instrumental in promoting the use of large-scale computation in science and engineering. Gray reported using this technique reliably; even if disks are damaged in transit, well-known RAID-like techniques could be used to mitigate the effects of such failures (Patterson, 2003).

Shortly after the overview was published, Amazon began offering such a service and continues to do so. Because network cost/performance is improving more slowly than any other cloud computing technology (see Table 3), the "FedEx a disk" option for large data transfers is likely to become increasingly attractive.

TABLE 3 Update of Gray's Costs of Computing Resources from 2003 to 2008

	Wide-area (long-haul) Network Bandwidth/Month	**CPU Hours (all cores)**	**Disk Storage**
Item in 2003	1 Mbps WAN[a] link	2 GHz CPU, 2 GB DRAM	200 GB disk, 50 Mb/s transfer rate
Cost in 2003	$100/month	$2,000	$200
What $1 buys in 2003	1 GB	8 CPU hours	1 GB
Item in 2008	100 Mbps WAN link	2 GHz, 2 sockets, 4 cores/socket, 4 GB DRAM	1 TB disk, 115 MB/s sustained transfer
Cost in 2008	$3,600/month	$1,000	$100
What $1 buys in 2008	2.7 GB	128 CPU hours	10 GB
Cost/performance improvement	2.7x	16x	10x
Cost to rent	$0.27–$0.40	$2.56	$1.20–$1.50
What $1 buys on AWS[b] in 2008	$0.10–$0.15/ GB × 3 GB	128 × 2 VMs@ $0.10 each	$0.12–$0.15/ GB-month × 10 GB

[a] WAN = wide-area (long-haul) network
[b] AWS = Amazon Web Services
Source: Armbrust et al., 2010a.

LICENSING AND CLOUD PROVIDER LOCK-IN

Amazon's EC2 represents one end of a spectrum in that its utility computing service consists of a bare-bones server built around the Intel x86 processor architecture. Cloud users must provide all of the software themselves, and open-source building blocks, such as the Linux operating system, are popular starting points. However, scientific and engineering research also frequently requires the use of proprietary software packages, such as Matlab.

Although some publishers of proprietary software (including Matlab) now offer a pay-as-you-go licensing model like the model used for the public cloud, most software is still licensed in a "cloud-unfriendly" manner (e.g., per seat or per computer). Changing the structure of software licenses to approximate the public cloud pricing model is a nontechnical but real obstacle to the increased use of the cloud in scientific computing.

In addition, if other providers, such as Google AppEngine or Microsoft Azure, provide value-added software functionality in their clouds, users might become dependent on such software to the point that their computing jobs come to require it. An example is Google AppEngine's automatic scale-up and scale-down functionality, which is available for certain kinds of user-deployed applications. If such applications were migrated to a non-Google platform, the application authors might have to create this functionality themselves.

The potential risk of "lock-in" to a single provider could be partially mitigated by standardizing the application programming interfaces and data formats used by different cloud services. Providers could then differentiate their offerings by the quality of their implementations, and migration from one provider to another would result in a possible loss of performance, rather than a loss of functionality. The Data Liberation Front, a project started by a group of Google engineers, is one group that is actively pursuing data standardization.

CONCLUSION

In 1995, researchers at Berkeley and elsewhere had argued that networks of commodity workstations (NOWs) offered potential advantages over high-performance symmetrical multiprocessors (Anderson et al., 1995). The advantages would include better scalability, cost-effectiveness, and potential high availability through inexpensive redundancy.

At that time software could not deal with important aspects of NOW architecture, such as the possibility of partial failure. Nevertheless, the economic and technical arguments for NOW seemed so compelling that, over the course of several years, academic researchers and commercial and open-source software authors developed tools and infrastructure for programming this idiosyncratic architecture at a much higher level of abstraction. As a result, applications that once took years for engineers to develop and deploy on a NOW can be prototyped today by Berkeley undergraduates as an eight-week course project.

Given this rapid evolution, there is good reason to be optimistic that in the near future computer-based scientific and engineering experiments that take weeks today will yield results in a matter of hours. When that time arrives, the necessity of purchasing and administering one's own supercomputer or computer cluster (and then waiting in line to use it) will seem as archaic as text-only interfaces do today.

ACKNOWLEDGMENTS

This work was done at the University of California Berkeley Reliable Adaptive Distributed Systems Laboratory (RAD Lab), and much of what is reported here builds on a survey article written by Michael Armbrust, the present author, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Support was provided by Sun Microsystems, Google, Microsoft, Amazon Web Services, Cisco Systems, Cloudera, eBay, Facebook, Fujitsu, Hewlett-Packard, Intel, Network Appliance, SAP, VMWare, and Yahoo! Inc., as well as matching funds from the State of California MICRO Program (Grants 06-152, 07-010, 06-148, 07-012, 06-146, 07-009, 06-147, 07-013, 06-149, 06-150, and 07-008), the National Science Foundation (Grant CNS-0509559), and the University of California Industry/University Cooperative Research Program (UC Discovery) Grant COM07-10240.

REFERENCES

Amazon AWS. 2008a. Public Data Sets on AWS. Available online at *http://aws.amazon.com/publicdatasets/*.

Amazon AWS. 2008b. AWS Case Study: Washington Post. Available online at *http://aws.amazon.com/solutions/case-studies/washington-post*.

Anderson, T.E., D.E. Culler, and D. Patterson. 1995. A case for NOW (networks of workstations). IEEE Micro 15(1): 54–64.

Armbrust, M., A. Fox, D.A. Patterson, N. Lanham, B. Trushkowsky, J. Trutna, and H. Oh. 2009. SCADS: scale-independent storage for social computing applications. In CIDR Perspectives 2009. Available online at *http://www.db.cs.wisc.edu/cidr/cidr2009/Paper_86.pdf*.

Armbrust, M., A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. 2010a. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report EECS-2009-28, EECS Department, University of California, Berkeley. Available online at *www.abovetheclouds.cs.berkeley.edu*.

Armbrust, M., A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. 2010b. A view of cloud computing. Communications of the ACM 53(4): 50–58.

Barroso, L.A., and U. Hölzle. 2007. The case for energy-proportional computing. IEEE Computer 40(12): 33–37.

Corbató, F.J., and V.A. Vyssotsky. 1965. Introduction and overview of the multics system. P. 185 in Proceedings of the Fall Joint Computer Conference, 1965. New York: IEEE.

Dean, J., and S. Ghemawat. 2004. MapReduce: Simplified data processing on large clusters. Pp. 137–150 in Proceedings of the 6th USENIX Symposium on Operating Systems Design and Implementation (OSDI '04), December 5–8, 2004, San Diego, Calif. Berkeley, Calif.: USENIX.

EIA (Energy Information Administration). 2010. State Electricity Prices, 2006. Available online at *http://www.eia.doe.gov/neic/rankings/stateelectricityprice.htm*.

Evangelinos, C., and C.N. Hill. 2008. Cloud computing for parallel scientific HPC applications: feasibility of running coupled atmosphere-ocean climate models on Amazon's EC2. In First ACM Workshop on Cloud Computing and its Applications (CCA'08), October 22–23, 2008, Chicago, Ill. New York: ACM.

Foster, I. 2009. What's faster—a supercomputer or EC2? Available online at *http://ianfoster.typepad.com/blog/2009/08/whats-fastera-supercomputer-or-ec2.html*.

Garfinkel, S. 2007. An Evaluation of Amazon's Grid Computing Services: EC2, S3 and SQS. Technical Report TR-08-07. Harvard University. Available online at *http://simson.net/clips/academic/2007.Harvard.S3.pdf*.

Hamilton, J. 2008. Internet-Scale Service Efficiency. Presentation at 2nd Large-Scale Distributed Systems and Middleware (LADIS) Workshop, September 15–17, 2008, White Plains, NY. Available online at *http://www.cs.cornell.edu/courses/cs5410/2008fa/Slides/MSCloud.pdf*.

Hindman, B., A. Konwinski, M. Zaharia, A. Ghodsi, A.D. Joseph, R.H. Katz, S. Shenker, and I. Stoica. 2010. Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center. Technical Report UCB-EECS-2010-87. University of California, Berkeley. Available online at *http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-87.html*.

LeFebvre, W. 2001. CNN.com: Facing a World Crisis. In Proceedings of the 15th Conference on Systems Administration (LISA 2001), San Diego, Calif., December 2–7, 2001. Berkeley, Calif.: USENIX.

NSF (National Science Foundation). 2009. National Science Foundation Awards Millions to Fourteen Universities for Cloud Computing Research. Available online at *http://www.nsf.gov/news/news_images.jsp?cntn_id=114686&org=NSF*.

Patterson, D. 2003. A Conversation with Jim Gray. ACM Queue 4(1). Available online at *http://queue.acm.org/detail.cfm?id=864078*.

Siegele, L. 2008. A Survey of Corporate IT: Let It Rise. The Economist (October 23). Web edition only.

Walker, E. 2008. Benchmarking Amazon EC2 for high performance scientific computing. ;login 33(5): 18–23. Available online at *http://www.usenix.org/publications/login/2008-10/openpdfs/walker.pdf*.

West, J. 2009. Twins Separated at Birth: Cloud Computing, HPC and How Microsoft is Trying to Change How We Think About Scale. Available online at *http://www.hpcwire.com/features/Twins-Separated-at-Birth-41173917.html.*

Warehouse-Scale Computing: The Machinery That Runs the Cloud

Luiz André Barroso
Google

As high-bandwidth Internet connectivity becomes more ubiquitous, an increasing number of applications are being offered as Internet services that run on remote data-center facilities instead of on a user's personal computer. The two classes of machines enabling this trend can be found on the very small and very large ends of the device spectrum. On the small end, mobile devices focus on user interaction and Internet connectivity, but with limited processing capabilities. On the large end, massive computing and storage systems (referred to here as w*arehouse-scale computers* [WSCs]) implement many of today's Internet (or Cloud) services, (Barroso and Hölzle, 2009).

Cost efficiency is critical for Internet services because only a small fraction of these services result directly in revenue; the rest comes mostly from online advertising. WSCs are particularly efficient for popular computing and data-intensive online services, such as Internet searches or language translations. Because a single search request may query the entire Web, including images, videos, news sources, maps, and product information, such services require a computing capacity well beyond the capabilities of a personal computing device. Thus, they are only economically feasible when amortized over a very large user population.

In this article I provide a brief description of the hardware and software in WSCs and highlight some of their key technical challenges.

HARDWARE

WSC hardware consists of three primary subsystems: computing equipment per se, power-distribution systems, and cooling infrastructure. A brief description of each subsystem follows below.

Computing Systems

WSCs are built of low-end or mid-range server-class computers connected in racks of 40 to 80 units by a first-level networking switch; each switch connects in turn to a cluster-level network fabric that ties together all of the racks. The clusters, which tend to be composed of several thousand servers, constitute the primary units of computing for Internet services. WSCs can be composed of one or many clusters. Storage is provided either as disk drives connected to each server or as dedicated file-serving appliances.[1]

The use of near PC-class components is a departure from the supercomputing model of the 1970s, which relied on extremely high-end technology, and reflects the cost sensitivity of the WSC application space. Lower-end server components that can leverage technology and development costs in high-volume consumer markets are therefore highly cost efficient.

Power Distribution

Peak electricity needs of computing systems in a WSC can be more than 10 MW—roughly equivalent to the average power usage of 8,000 U.S. households. At those levels, WSC computing systems must tap into high-voltage, long-distance power lines (typically 10 to 20 kilovolts); the voltage level must then be converted down to 400 to 600 volts, the levels appropriate for distribution within the facility.

Before power is distributed to computing equipment, however, it is fed to an uninterruptible power supply (UPS) system that acts as an energy supply buffer against utility power failures. UPS systems are designed to support less than a minute of demand, since diesel generators can jump into action within 15 seconds of a utility outage.

Cooling Infrastructure

Virtually all energy provided to computing equipment becomes heat that must be removed from the facility so the equipment can remain within its designed operating temperature range. This is accomplished by air conditioning units inside the building that supply cold air (18 to 22°C) to the machinery, coupled by a liquid coolant loop to a cooling plant situated outside the building. The cooling plant uses chiller or cooling towers to expel heat to the environment.

[1] Storage systems based on FLASH memory technology (sometimes called solid-state drives, or SSDs) are just beginning to be considered for WSC systems as an intermediary layer between DRAM and magnetic disk drives.

Relative Costs

The capital costs of the three main subsystems of a WSC vary depending on the facility design. The cost of non-computing components is proportional to peak power delivery capacity, and cooling infrastructure is generally more expensive than the power-distribution subsystem. If high-end energy-efficient computing components are used, computing system costs tend to be dominant. If lower end, less energy-efficient computing components are used, cooling and power-distribution system costs usually predominate. Energy, therefore, affects WSC costs in two ways: (1) directly through the price of the amount of electricity consumed; and (2) indirectly through the cost of cooling and power plants.

Design Challenges

Designing a WSC represents a formidable challenge. Some of the most difficult issues are deciding between scale-up (e.g., bigger servers) and scale-out (e.g., more servers) approaches and determining the right aggregate capacity and performance balance among the subsystems. For example, we may have too much CPU firepower and too little networking bandwidth.

These decisions are ultimately based on workload characteristics. For example, search workloads tend to compute heavily within server nodes and exchange comparatively little networking traffic. Video serving workloads do relatively little processing but are network intensive. An Internet services provider that offers both classes of workloads might have to design different WSCs for each class or find a common sweet spot that accommodates the needs of both. Common designs, when possible, are preferable, because they allow the provider to dynamically re-assign WSC resources to workloads as business priorities change, which tends to happen frequently in the still-young Internet services area.

Energy Efficiency

Given the impact of energy on overall costs of WSCs, it is critical that we understand where energy is used. The data-center industry has developed a metric, called power usage effectiveness (PUE), that objectively characterizes the efficiency of non-computing elements in a facility. PUE is derived by measuring the total energy that enters a facility and dividing it by the amount consumed by the computing equipment. Typical data centers are rather inefficient, with PUEs hovering around 2 (one Watt used, one Watt wasted). State-of-the-art facilities have reported PUEs as low as 1.13 (Google, 2010); at such levels, the energy-efficiency focus shifts back to the computing equipment itself.

Mobile and embedded devices have been the main targets of low-power technology development for decades, and many of the energy-saving features that make their way to servers had their beginnings in those devices. However, mobile systems have focused on techniques that save power when components are idle, a feature that is less useful for WSCs, which are rarely completely idle. Therefore, energy-efficient WSCs require energy proportionality, system behavior that yields energy-efficient operation for a range of activities (Barroso and Hölzle, 2007).

SOFTWARE

The software that runs on WSCs can be broadly divided into two layers: infrastructure software and workloads. Both are described below.

Infrastructure Software

The software infrastructure in WSCs includes some basic components that enable their coordinated scheduling and use. For example, each Google WSC cluster has a management software stack that includes a scheduling master and a storage master, and corresponding slaves in each machine. The scheduling master takes submitted jobs and creates job-task instances in various machines. Enforcing resource allocations and performance isolation among tasks is accomplished by per-machine scheduling slaves in coordination with the underlying operating system (typically a Linux-based system). The role of storage servers is to export local disks to cluster-wide file-system users.

Workloads

WSC workloads can include thousands of individual job tasks with diverse behavior and communication patterns, but they tend to fall into two broad categories: data processing and online services. Data processing workloads are large-batch computations necessary to analyze, reorganize, or convert data from one format to another. Examples of data-processing workloads might include stitching individual satellite images into seamless Google Earth tiles or building a Web index from a large collection of crawled documents. The structure of these workloads tends to be relatively uniform, and the keys for high performance are finding the right way to partition them among multiple tasks and then place those tasks closer to their corresponding data. Programming systems, such as MapReduce (Dean and Ghemawat, 2004), have simplified the building of complex data-processing workloads.

Web search is the best example of a demanding online-services workload. For these workloads, keeping users happy means providing very quick response times. In some cases, the system may have to process tens of terabytes of index data to respond to a single query. Thus, although high processing throughput is a requirement for both data-processing and online-services workloads, the latter have much stricter latency constraints per individual request. The main challenge for online-services workloads it to provide predictable performance by thousands of cooperating nodes on sub-second timescales.

Programming Challenges

Similar to the hardware-design problem for WSCs, the complexity of software development for a WSC hardware platform can be an obstacle for both workload and infrastructure software developers. The complexity derives from a combination of scale and limits of electronic technology and physics. For example, a processor accessing its local memory can do so at rates of more than 10 gigabytes per second, but accessing memory attached to another processor in the facility may only be feasible at rates that are slower by orders of magnitude.

WSC software designers must also be able to cope with failures**.** Two server crashes per year may not sound particularly damaging, but if the system runs on 5,000 servers it will see approximately one failure every hour. Programming efficient WSC workloads requires handling complex performance trade-offs and creating reliable systems in the face of high failure rates.

WRAP UP

The rapid increase in the popularity of Internet services as a model for provisioning computing and storage solutions has given rise to a new class of massive-scale computers outside of the traditional application domain of supercomputing-class systems. Some of the world's largest computing systems are the WSCs behind many of today's Internet services. Building and programming this emerging class of machines are the subjects of some of the most compelling research being conducted today on computer systems.

REFERENCES

Barroso, L.A., and U. Hölzle. 2007. The case for energy-proportional computing. IEEE Computer, December 2007.

Barroso, L.A., and U. Hölzle. 2009. The Datacenter as a Computer—An Introduction to the Design of Warehouse-Scale Machines. Synthesis Series on Computer Architecture. San Rafael, Calif.: Morgan & Claypool Publishers.

Dean, J., and S. Ghemawat. 2004. MapReduce: Simplified Data Processing on Large Clusters. In OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, California, December, 2004.

Google. 2010. Google Data Center Efficiency Measurements. Available online at *http://www.google.com/corporate/green/datacenters/measuring.html*.

Developing Robust Cloud Applications

Yuanyuan (YY) Zhou
University of California, San Diego

Despite possible security and privacy risks, cloud computing has become an industry trend, a way of providing dynamically scalable, readily available resources, such as computation, storage, and so forth, as a service to users for deploying their applications and storing their data. No matter what form cloud computing takes—public or private cloud, raw cloud infrastructure, or applications (software) as a service—it provides the benefits of utility-based computing (i.e., computing on a pay-only-for-what-you-use basis).

Cloud computing can provide these services at reduced costs, because cloud service is paid for incrementally and scales with demand. It can also support larger scale computation, in terms of power and data storage, without the configuration and set-up hassles of installing and deploying local, large-scale clusters. Cloud computing also has more mobility, because it provides access from wherever the Internet is available. These benefits allow IT users to focus on domain-specific problems and innovations.

More and more applications are being ported or developed to run on clouds. For example, Google News, Google Mail, and Google Docs all run on clouds. Of course, these platforms are also owned and controlled by the application service provider, namely Google, which makes some of the challenges discussed below easier to address.

Many applications, especially those that require low costs and are less sensitive to security issues, such as Amazon Elastic Computing Cloud (EC2) and Amazon Machine Images (AMIs), have moved to public clouds. Since February 2009, for example, IBM and Amazon Web Services have allowed developers to use Amazon EC2 to build and run a variety of IBM platform technologies. Because developers can use their existing IBM licenses on Amazon EC2, software developers have been able to build applications based on IBM software within Amazon EC2. This new "pay-as-you-go" model provides development and production instances of IBM DB2, Informix Dynamic Server, WebSphere Portal, Lotus Web Content Management, and Novell's SUSE Linux operating system on EC2.

With this new paradigm in computation, cost savings, and other benefits, cloud computing also brings unique challenges to building robust, reliable applications on clouds. The first major challenge is the change in mindset to the unique characteristics (e.g., elasticity of scale, transparency of physical devices, unreliable components, etc.) of deploying and running an application in clouds. The second challenge is the development of frameworks and tool sets to support the development, testing, and diagnosis of applications in clouds.

In the following sections, I describe how the traditional application development and execution environment has changed, the unique challenges and characteristics of clouds, the implications of cloud computing for application development, and suggestions for easing the move to the new paradigm and developing robust applications for clouds.

DIFFERENCES BETWEEN CLOUDS AND TRADITIONAL PLATFORMS

Although there are many commonalities between traditional in-house/local-execution platforms and clouds, there are also characteristics and challenges that are either unique or more pronounced in clouds. Some short-term differences that will disappear when cloud computing matures are discussed below.

Statelessness and Server Failures

Because one of the major benefits of cloud computing is lower cost, cloud service providers are likely to use cost-effective hardware/software that is also less robust and less reliable than people would purchase for in-house/local platforms. Thus, the underlying infrastructure may not be configured to support applications that require very reliable and robust platforms.

In the past two to three years, there have been many service outages in clouds. Some of the most widely known outages have caused major damage, or at least significant inconvenience, to end users. For example, when Google's Gmail faltered on September 24, 2009, even though the system was down for only a few hours, it was the second outage that month and followed a disturbing sequence of outages for Google's cloud-based offerings for search, news, and other applications in the past 18 months. Explanations ranged from routing errors to problems with server maintenance. Another example is the outage on Twitter in early August 2009 that lasted throughout the morning and into early afternoon and probably angered serious "twitterers."

Ebay's PayPal online payments system also failed a few times in August 2009; outages lasted from one to more than four hours, leaving millions of customers unable to complete transactions. A network hardware problem was reported to be the culprit. PayPal lost millions of dollars, and merchants lost unknown amounts. Thomas Wailgum of CIO.com reported in January 2009, that Salesforce.com had suffered a service disruption for about an hour on January 6 when a core network device failed because of memory allocation errors.

General public service providers have also experienced outages. For example, Rackspace was forced to pay out $2.5 to $3.5 million in service credits to customers in the wake of a power outage that hit its Dallas data center in late June 2009. Amazon S3 storage service was knocked out in summer 2008; this was followed by another outage in early 2009 caused by too many authentication requests.

Lack of Transparency and Control (Virtual vs. Physical)

Because clouds are based on virtualization, applications must be virtualized before they can be moved to a cloud environment. Thus, unlike local platforms, cloud computing imposes a layer of abstraction between applications and physical machines/devices. As a result, many assumptions and dependencies on the underlying physical systems have to be removed, leaving applications with little control, or even knowledge of, the underlying physical platform or other applications sharing the same platform.

Network Conflicts with Other Applications

For in-house data grids, it is a good idea to use a separate set of network cards and put them on a dedicated VLAN, or even their own switch, to avoid broadcast traffic between nodes. However, application developers for a cloud may not have this option. To maximize usage of the system, cloud service providers may put many virtual machines on the same physical machine and may design a system architecture that groups significant amounts of traffic going through a single file server, database machine, or load balancer. For example, so far there is no equivalent of network-attached shared storage on Amazon. In other words, cloud application developers should no longer assume they will have dedicated network channels or storage devices.

Less Individualized Support for Reliability and Robustness

In addition to the absence of a dedicated network, I/O devices are also less likely to find cloud platforms that provide individualized guarantees for reliability and robustness. Although some advanced, mature clouds may provide several levels of reliability support in the future, this support will not be fine-grained enough to match individual applications.

Elasticity and Distributed Bugs

The main driver for the development of cloud computing is for the system to be able to grow as needed and for customers to pay only for what they use (i.e., elasticity). Therefore, applications that can dynamically react to changes in workload are good candidates for clouds. The cost of running an application on a cloud is much lower than the cost of buying hardware that may remain idle except in times of peak demand.

If a good percentage of your workloads have already been virtualized, then they are good candidates for clouds. If you simply port the static images of existing applications to clouds, you are not taking advantage of cloud computing. In effect, your application will be over-provisioned based on the peak load, and you will have a poorly used environment. Moving existing enterprise applications to the cloud can be very difficult simply because most of them were not designed to take advantage of the cloud's elasticity. Distributed applications are prone to bugs, such as deadlocks, incorrect message ordering, and so on, all of which are difficult to detect, test, and debug.

Elasticity makes debugging even more challenging. Developers of distributed applications must think dynamically to allocate/reclaim resources based on workloads. However, this can easily introduce bugs, such as resource leaks or tangling links to reclaimed resources. Addressing this problem will require either software development tools for testing and detecting these types of bugs or new application development models, such as MapReduce, which would eliminate the need for dynamic scaling up and down.

Lack of Development, Execution, Testing, and Diagnostic Support

Finally, one of the most severe, but fortunately short-term, challenges is the lack of development, testing, and diagnostic support. Most of today's enterprise applications were built using frameworks and technologies that were not ideal for clouds. Thus, an application that works on a local platform may not work well in a cloud environment. In addition, if an application fails or is caught up in a system performance bottleneck caused by the transparency of physical configuration/layout or other applications running on the same physical device/hardware, diagnosing and debugging the failure can be a challenge.

IMPROVING CLOUDS

Cloud computing is likely to bring transformational change to the IT industry, but this transformation cannot happen overnight—and it certainly cannot happen without a plan. Both application developers and platform providers will have to work hard to develop robust applications for clouds.

Application developers will have to adopt the new paradigm. Before they can evaluate whether their applications are well suited, or at least have been revised properly to take advantage of the elasticity of clouds, they must first understand the reasons for, and benefits of, moving to clouds. Second, since each cloud platform may be different, it is important that application developers understand the platform's elasticity model and dynamic configuration method. They must also keep abreast of the provider's evolving monitoring services and service level agreements, even to the point of engaging the service provider as an ongoing operations partner to ensure that the demands of the new application can be met.

The most important thing for cloud platform providers is to provide application developers with testing, deployment, execution, monitoring, and diagnostic support. In particular, it would be useful if applications developers have a good local debugging environment as well as testing platforms that can help with programming and debugging programs written for the cloud.

Unfortunately, experience with debugging on local platforms does not usually simulate real cloud-like conditions. From my personal experience and from conversations with other developers, I have come to realize that most people face problems when moving code from their local servers to clouds because of behavioral differences such as those described above.

CLOUD COMPUTING ADOPTION MODEL

A cloud computing model, proposed by Jake Sorofman in an October 20, 2008 article on the website, *Dr. Dobb's: The World of Software Development*, provides an incremental, pragmatic approach to cloud computing. Loosely based on the Capability Maturity Model (CMM) developed by the Software Engineering Institute (SEI) at Carnegie Mellon University, this Cloud Computing Adoption Model (Figure 1) proposes five steps for adopting the cloud model: (1) Virtualization—leveraging hypervisor-based infrastructure and application virtualization technologies for seamless portability of applications and shared server infrastructure; (2) Exploitation—conduct controlled, bounded deployments using Amazon EC2 as an example of computing capacity and a reference architecture; (3) Establishment of foundations—determine governance, controls, procedures, policies, and best practices as they begin to form around the development and deployment of cloud applications. In this step, infrastructures for developing, testing, debugging, and diagnosing cloud applications are an essential part of the foundation to make the cloud a mainstream of computing; (4) Advancement—scale up the volume of cloud applications through broad-based deployments in the cloud; and (5) Actualization—balance dynamic workloads across multiple utility clouds.

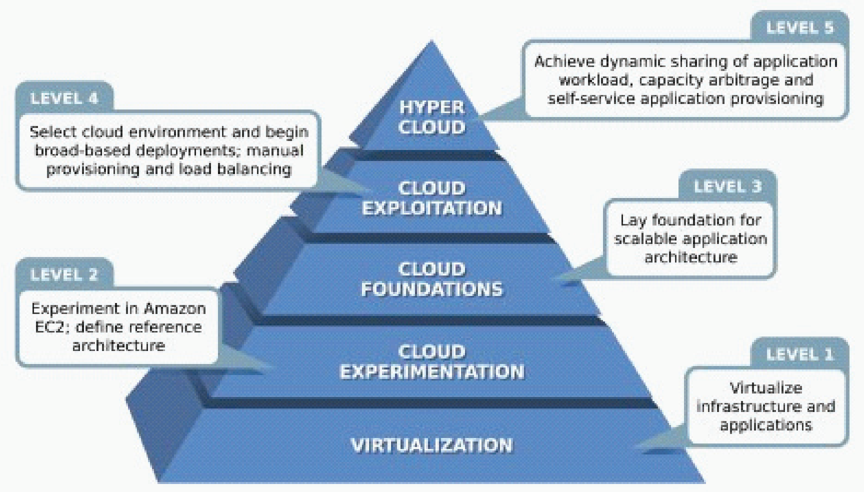

LEVEL 5
Achieve dynamic sharing of application workload, capacity arbitrage and self-service application provisioning
HYPER CLOUD
LEVEL 4
Select cloud environment and begin broad-based deployments; manual provisioning and load balancing
CLOUD EXPLOITATION
LEVEL 3
Lay foundation for scalable application architecture
CLOUD FOUNDATIONS
LEVEL 2
Experiment in Amazon EC2; define reference architecture
CLOUD EXPERIMENTATION
LEVEL 1
Virtualize infrastructure and applications
VIRTUALIZATION

FIGURE 1 Cloud Computing Adoption Model. Source: *http://www.rpath.com/corp/cloud-adoption-model.*

Green Clouds: The Next Frontier

PARTHASARATHY RANGANATHAN
Hewlett Packard Research Labs

We are entering an exciting era for computer-systems design. In addition to continued advances in performance, next-generation designs are also addressing important challenges related to power, sustainability, manageability, reliability, and scalability. At the same time, new combinations of emerging technologies (e.g., photonics, non-volatile storage, and 3D stacking), and new workloads (related to cloud computing, unstructured data, and virtualization) are presenting us with new opportunities and challenges. The confluence of these trends has led us to rethink the way we design systems—motivating holistic designs that cross traditional design boundaries.

In this article, we examine what this new approach means for the basic building blocks of future systems and how to manage them. Focusing on representative examples from recent research, we discuss the potential for dramatic (10 to 100X) improvements in efficiency in future designs and the challenges and opportunities they pose for future research.

PREDICTING THE FUTURE OF COMPUTING SYSTEMS

What can we predict for computing systems 10 years from now? Historically, the first computer to achieve terascale computing (10^{12}, or one trillion computing operations per second) was demonstrated in the late 1990s. About 10 years later, in mid-2008, the first petascale computer was demonstrated at 1,000 times more performance capability. Extrapolating these trends, one can expect an exascale computer by approximately 2018. That is a staggering *one million trillion* computing operations per second and a thousand-fold improvement in performance over any current computer.

Moore's law (often described as the trend that computing performance doubles every 18 to 24 months) has traditionally helped predict performance challenges, for terascale and more recently petascale computing, but the transition from petascale to exascale computing is likely to pose some new challenges we need to address going forward.

CHALLENGES

The Power Wall

The first challenge is related to what is commonly referred to as the *power wall*. Power consumption is becoming a key constraint in the design of future systems. This problem is manifested in several ways: in the amount of electricity consumed by systems; in the ability to cool systems cost effectively; in reliability; and so on.

For example, recent reports indicate that the electricity costs for powering and cooling cloud datacenters can be millions of dollars per year, often more than was spent on buying the hardware (e.g., Barroso and Hölzle, 2007)! IDC, an industry analyst firm, has estimated that worldwide investment in power and cooling was close to $40 billion last year (Patel, 2008).

This emphasis on power has begun to have a visible impact on the design of computing systems, as system design constraints are shifting from optimizing performance to optimizing energy efficiency or performance achieved per watt of power consumed in the system. This shift has been partly responsible for the emergence of multi-core computing as the dominant way to design microprocessors.

In addition, recognition has been growing that designers of energy-efficiency optimized systems must take into consideration not only power consumed by the computing system, but also power consumed by the supporting equipment. For example, for every watt of power consumed in the server of a datacenter, an additional half to one watt of power is consumed in the equipment responsible for power delivery and cooling (often referred to as the burdened costs of power and cooling, or power usage effectiveness [PUE] [Belady et al., 2008]).

Sustainability

*S*ustainability is also emerging as an important issue. The electricity consumption associated with information technology (IT) equipment is responsible for 2 percent of the total carbon emissions in the world, more than the emissions of the entire aviation industry. More important, IT is increasingly being used as the tool of choice to address the remaining 98 percent of carbon emissions from non-IT industries (e.g., the use of video conferencing to reduce the need for travel or the use of cloud services to avoid transportation or excess manufacturing costs) (Banerjee et al., 2009).

One way to improve sustainability is to consider the total life cycle of a system—including both the supply and demand side. In other words, in addition to the amount of energy used in *operating* a system, it is important to consider the amount of energy used in *making* the system.

Manageability

Sustainability is just one of the new "*ilities*" that pose challenges for the future. Another key challenge pertains to *manageability*, which can be defined as the collective processes of deployment, configuration, optimization, and administration during the life cycle of an IT system.

To illustrate this challenge, consider, as an example, the potential infrastructure in a future cloud datacenter. On the basis of recent trends, one can assume that there will be five global datacenters with 40 modular containers each, 10 racks per container, 4 enclosures per rack, and 16 blade servers per enclosure. If each blade server has two sockets with 32 cores each and 10 virtual machines per core, this cloud vendor will have a total of 81,920,000 virtual servers to operate its services. Each of the more than 80 million servers, in turn, will require several classes of operations—for bring-up, day-to-day operations, diagnostics, tuning, and other processes, ultimately including retirement or redeployment of the system. Although a lot of work has been done on managing computer systems, manageability on such a large scale poses new challenges.

Reliability

Trends in technology scaling circuit level and increased on-chip integration at the micro-architectural level lead to a higher incidence of both transient and permanent errors. Consequently, new systems must be designed to operate reliably and provide continued up-time, even when they are built of unreliable components.

Business Trends

Finally, these challenges must be met within the constraints of recent business trends. One important trend is the emphasis on reducing total costs of ownership for computing solutions. This often translates to a design constraint requiring the use of high-volume commodity components and avoiding specialization limited to niche markets.

OPPORTUNITIES

We believe that the combination of challenges—low power, sustainability, manageability, reliability, and costs—is likely to influence how we think about system design to achieve the next 1,000-fold increase in performance for the next decade. At the same time, we recognize that interesting opportunities are opening up as well.

Data-Centric Workloads

A fundamental shift has taken place in terms of data-centric workloads. The amount of data being created is increasing exponentially, much faster than Moore's law predicted. For example, the size of the largest data warehouse in the Winter Top Ten Survey has been growing at a cumulative annual growth rate of 173 percent (Winter, 2008). The amount of online data is estimated to have increased nearly 60-fold in the last seven years, and data from richer sensors, digitization of offline content, and new applications like Twitter, Search, and others will surely increase data growth rates. Indeed, it is estimated that only 5 percent of the world's off-line data has been digitized or made available through online repositories so far (Mayer, 2009).

The emergence and rapid increase of data as a driving force in computing has led to a corresponding increase in data-centric workloads. These workloads focus on different aspects of the data life cycle (capture, classify, analyze, maintain, archive, and so on) and pose significant challenges for the computing, storage, and networking elements of future systems.

Among these, an important recent trend (closely coupled with the growth of large-scale Internet web services) has been the emergence of complex analysis on an immense scale. Traditional data-centric workloads like web serving and online transaction processing (e-commerce) are being superseded by workloads like real-time multimedia streaming and conversion; history-based recommendation systems; searches of texts, images, and even videos; and deep analysis of unstructured data (e.g., Google Squared).

Emerging data-centric workloads have changed our assumptions about system design. These workloads typically operate at larger scale (hundreds of thousands of servers) and on more diverse data (e.g., structured, unstructured, rich media) with input/output (I/O) intensive, often random data-access patterns and limited locality. Another characteristic of data-centric workloads is a great deal of innovation in the software stack to increase scalability and commodity hardware (e.g., Google MapReduce/BigTable).

Improvements in Throughput, Energy Efficiency, Bandwidth, and Memory Storage

Recent trends suggest several potential technology disruptions on the horizon (Jouppi and Xie, 2009). On the computing side, recent microprocessors have favored multi-core designs that emphasize multiple simpler cores for greater throughput. This approach is well matched with the large-scale distributed parallelism in data-centric workloads. Operating cores at near-threshold voltage has been shown to significantly improve energy efficiency. Similarly, recent advances in networking, particularly related to optics, show a strong growth in bandwidth for communication among computing elements at various levels of the system.

Significant changes are also expected in the memory/storage industry. Recently, new non-volatile RAM (NVRAM) memory technologies have been demonstrated that significantly reduce latency and improve energy efficiency compared to Flash and Hard Disk. Some of these NV memories, such as phase-change RAM (PCRAM) and Memristors, have shown the potential to replace DRAM with competitive performance and better energy efficiency and technology scaling. At the same time, several studies have postulated the potential end of DRAM scaling (or at least a significant slowing down) over the next decade, which further increases the likelihood that DRAM will be replaced by NVRAM memories in future systems.

INVENTING THE FUTURE— CROSS-DISCIPLINARY HOLISTIC SYSTEM DESIGN

We believe that the confluence of all these trends—the march toward exascale computing and its associated challenges, opportunities related to emerging large-scale distributed data-centric workloads, and potential disruptions from emerging advances in technology—offers us a unique opportunity to rethink traditional system design.

We believe that the next decade of innovation will be characterized by a holistic emphasis that cuts across traditional design boundaries—across layers of design from chips to datacenters; across different fields in computer science, including hardware, systems, and applications; and across different engineering disciplines, including computer engineering, mechanical engineering, and environmental engineering.

We envision that in the future, rather than focusing on the design of single computers, we will focus on the design of computing elements. Specifically, future systems will be (1) composed of simple building blocks that are efficiently co-designed across hardware and software and (2) composed together into computing ensembles, as needed and when needed. We refer to these ideas as designing *disaggregated dematerialized system elements* bound together by a *composable ensemble management* layer. In the discussion below, we present three illustrative examples from our recent research demonstrating the potential for dramatic improvements.

Cross-Layer Power Management

In the past few years, interest has surged in enterprise power management. Given the multifaceted nature of the problem, the solutions have correspondingly focused on different dimensions. For example, some studies have focused on average power reduction for lower electricity costs while others have focused on peak power management for lower air conditioning and power-delivery costs.

Previous studies can also be categorized based on (1) their approaches (e.g., local resource management, distributed resource scheduling, virtual machine migration); (2) options for controlling power (e.g., processor voltage scaling, component sleep states, turning systems off); (3) specific levels of implementation—chip, server, cluster, or datacenter level—hardware, software, or firmware; and (4) objectives and constraints of the optimization problem—for example, whether or not we allow performance loss and whether or not we allow occasional violations in power budgets.

In the future, many (or all) of these solutions are likely to be deployed together to improve coverage and increase power savings. Currently, emergent behavior from the collection of individual optimizations may or may not be globally optimal, or even stable, or even correct! A key need, therefore, is for a carefully designed, flexible, extensible coordination framework that minimizes the need for global information exchange and central arbitration.

In this first example, we explain how a collaborative effort between computer scientists, thermo-mechanical engineers, and control engineering experts led to a novel coordination solution. The solution is summarized in Figure 1 and is further elaborated in Raghavendra et al. (2008). Briefly, this design is based on carefully connecting and overloading the abstractions in current implementations to allow individual controllers to learn and react to the effect of other controllers, the same way they would respond to variations in workload demand. This enables formal mathematical analysis of stability and provides flexibility to dynamic changes in the controllers and system environments. A specific coordination architecture for five individual solutions using different techniques and actuators to optimize for different goals at different system levels across hardware and software demonstrates that a cross-layer solution can achieve significant advantages in correctness, stability, and efficiency over existing state of the art.

Although illustrative design has shown the potential of a cross-disciplinary approach to improving power management for the cloud, many more opportunities have yet to be explored. Specifically, how do we define the communication and coordination interfaces to enable federated architectures? How do we extend solutions to adapt to application-level semantics and heterogeneity in the systems space (Kansal et al., 2009)? How do we design federation at the larger scale typical of cloud systems? Finally, although our discussions have focused on power management, the “intersecting control loops” problem is representative of a larger class of management problems—how architectures generalize to broader resource management domains.

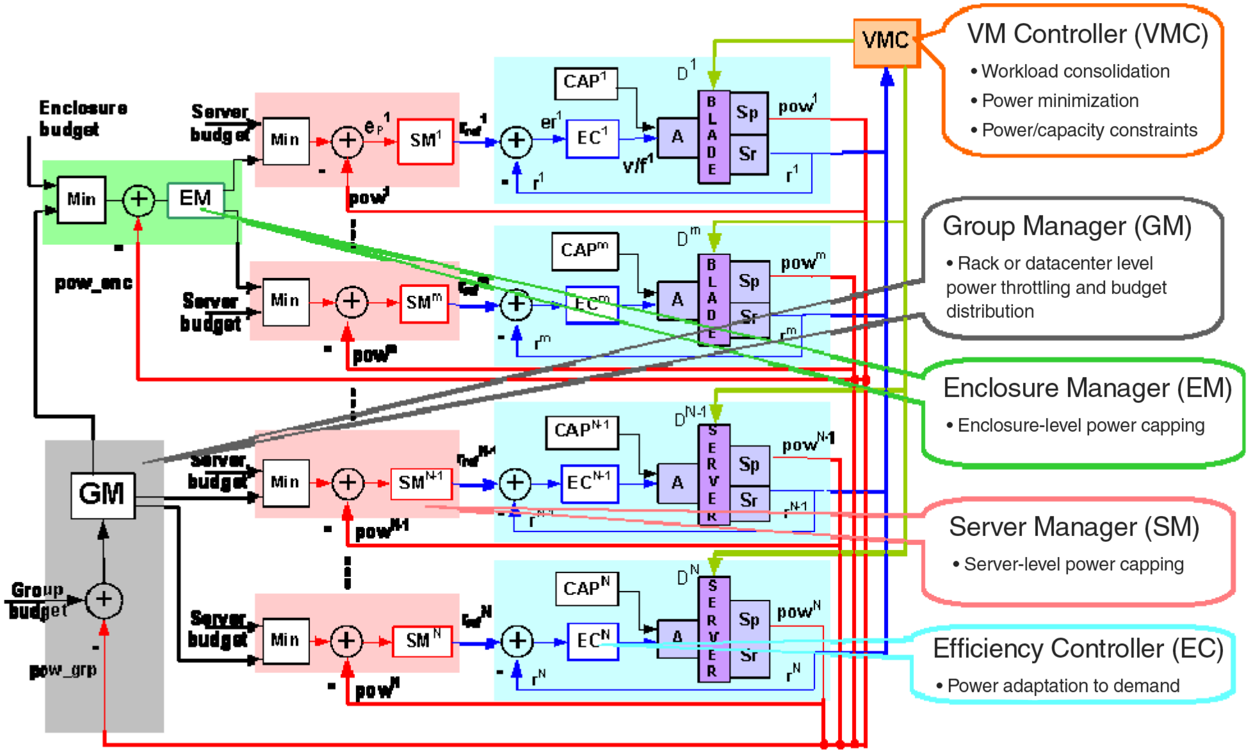


FIGURE 1 A coordinated power-management architecture. The proposed architecture coordinates different kinds of power-management solutions (multiple levels, approaches, time constants, objective functions, and actuators). Key features include (a) a control-theoretic core to enable formal guarantees of stability and (b) intelligent overloading of control channels to include the impact of other controllers, reduce the number of interfaces, and limit the need to access global data.

Dematerialized Datacenters

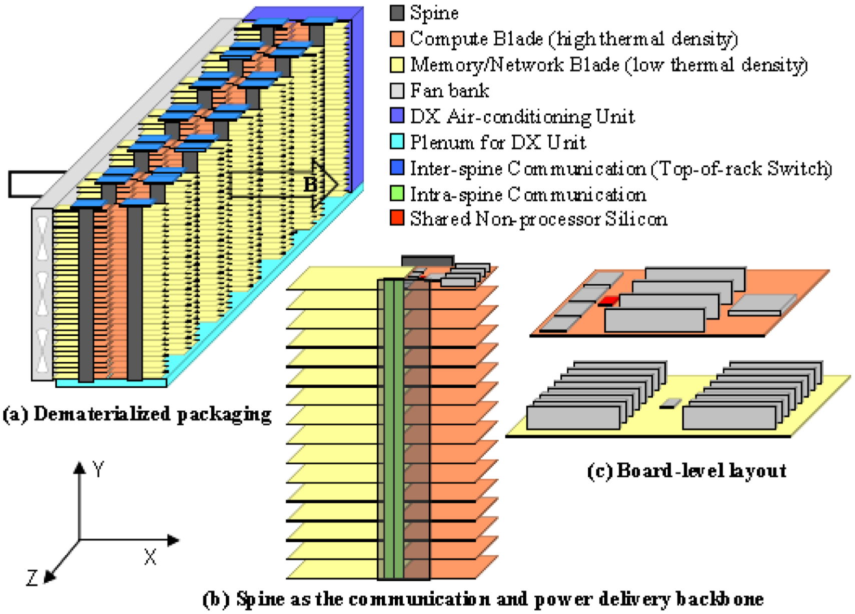
Our second example is a collaborative project by computer scientists, environmental engineers, and mechanical engineers to build a sustainability-aware new datacenter solution. Unlike prior studies that focused purely on operational energy consumption as a proxy for sustainability, we use the metric of *life-cycle exergy destruction* to systematically study the environmental impact of current designs for the entire life cycle of the system, including embedded impact factors related to materials and manufacturing.

A detailed description of exergy is beyond the scope of this article, but briefly, exergy corresponds to the available energy in a system. Unlike energy, which is neither created nor destroyed (the first law of thermodynamics), exergy is continuously consumed in the performance of useful work by any real entropy-generating process (the second law of thermodynamics). Previous studies have shown that the destruction (or consumption) of exergy is representative of the irreversibility associated with various processes. Consequently, at a first-level of approximation, exergy can be used as a proxy to study environmental sustainability.

Studying exergy-efficient designs leads to several new insights (Chang et al., 2010). First, focusing on the most efficient system design does not always produce the most sustainable solution. For example, although energy-proportional designs are optimal in terms of operational electricity consumption, virtual machine consolidation is more sustainable than energy proportionality in some cases. Next, the ratio of embedded exergy to total exergy has been steadily increasing over the years, motivating new optimizations that explicitly target embedded exergy (e.g., recycling or dematerialization). Finally, performance and embedded, operational, and infrastructure exergy are not independent variables. Sustainability must be addressed holistically to include them all.

Based on insights provided by the study just described, we propose a new solution (Figure 2) that is co-designed across system architecture and physical organization/packaging. This solution includes three advances that work together to improve sustainability: (1) new material-efficient physical organization, (2) environmentally efficient cooling infrastructures, and (3) effective design of system architectures to enable the reuse of components.

A detailed evaluation of our proposed solution, which includes a combination of sustainability models, computational fluid-dynamics modeling, and full-system computer architecture simulation, demonstrates significant improvements in sustainability, even compared to an aggressive future configuration (Meza et al., 2010). The proposed design illustrates the opportunities that lie ahead. New silicon-efficient architectures, system designs that explicitly target up-cycling, and data-centers with renewable energy sources are the subjects of on-going research that can bring us closer to truly sustainable datacenters (Patel, 2008).


■ Spine
🟧 Compute Blade (high thermal density)
🟨 Memory/Network Blade (low thermal density)
□ Fan bank
🟪 DX Air-conditioning Unit
🟦 Plenum for DX Unit
🟦 Inter-spine Communication (Top-of-rack Switch)
🟩 Intra-spine Communication
🟥 Shared Non-processor Silicon
B
Y
X
Z

(a) Dematerialized packaging

(b) Spine as the communication and power delivery backbone

(c) Board-level layout

FIGURE 2 Conceptual sketch of a design for a dematerialized green datacenter. This specific design illustrates a container for cloud workloads that incorporates several optimizations co-designed with each other, including (1) new material-efficient physical design, (2) component reuse enabled by a disaggregated system architecture, (3) sharing among collections of systems to reduce the amount of material used in building the system, (4) environmentally efficient cooling that leverages ambient air, and (5) thermal density clustering for lower cooling exergy.

From Microprocessors to Nanostores

The third example is a cross-disciplinary collaboration among device physicists, computer engineers, and systems software developers to design a disruptive new system architecture for future data-centric workloads (Figure 3). Leveraging the memory-like and disk-like attributes of emerging non-volatile technologies, we propose a new building block, called a nanostore, for data-centric system design (Ranganathan, in press).

A nanostore is a single-chip computer that includes 3-D stacked layers of dense silicon non-volatile memory with a layer of compute cores and a network interface. A large number of individual nanostores can communicate over a simple interconnect and run a data-parallel execution environment like MapReduce to support large-scale distributed data-centric workloads. The two most impor-

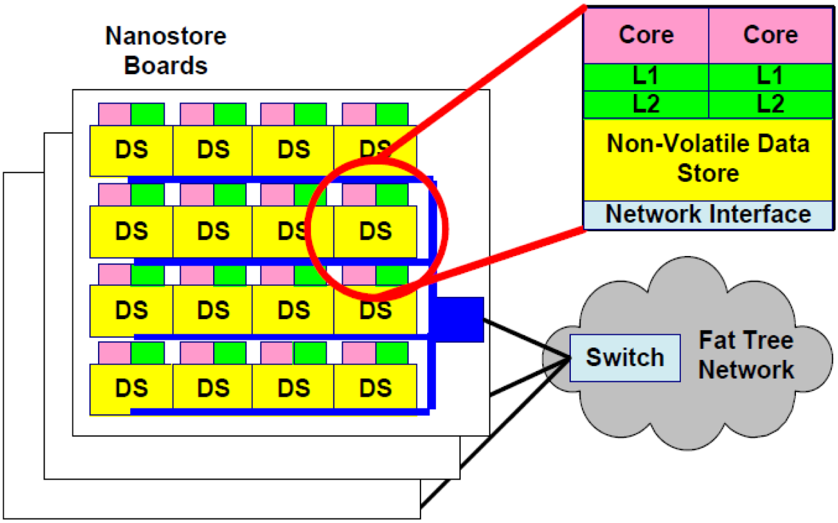


FIGURE 3 The combination of emerging data-centric workloads and upcoming non-volatile and other technologies offer the potential for a new architecture design—"nanostores" that co-locate power-efficient compute cores with non-volatile storage in the same package in a flatter memory hierarchy.

tant aspects of nanostores are (1) the co-location of power-efficient computing with a single-level data store, and (2) support for large-scale distributed design. Together, they provide several benefits.

The single-level data store enables improved performance by providing faster data access (in latency and bandwidth). Energy efficiency is also improved by the flattening of the memory hierarchy and the increased energy efficiency of NVRAM over disk and DRAM. The large-scale distributed design, which increases parallelism and overall data/network bandwidth, allows for higher performance. This design also improves energy efficiency by partitioning the system into smaller elements that can leverage more power-efficient components (e.g., simpler cores).

Our results show that nanostores can achieve orders of magnitude higher performance with dramatically better energy efficiency. More important, they have the potential to be used in new architectural models (e.g., leveraging a hierarchy of computes [Ranganathan, in press]) and to enable new data-centric applications that were previously not possible. Research opportunities include in-systems software optimizations for single-level data stores, new endurance optimizations to improve data reliability, and architectural balance among compute, communication, and storage.

CLOSING

Although the research described in these examples shows promising results, we believe we have only scratched the surface of what is possible. Opportunities abound for further optimizations, including for hardware-software co-design (e.g., new interfaces and management of persistent data stores [Condit et al., 2009]) and other radical changes in system designs (e.g., bio-inspired "brain" computing [Snider, 2008]).

Overall, the future of computing systems offers rich opportunities for more innovation by the engineering community, particularly for cross-disciplinary research that goes beyond traditional design boundaries. Significant improvements in the computing fabric enabled by these innovations will also provide a foundation for innovations in other disciplines.

REFERENCES

Banerjee, P., C.D. Patel, C. Bash, and P. Ranganathan. 2009. Sustainable data centers: enabled by supply and demand side management. Design Automation Conference, 2009: 884–887.

Barroso, L.A., and U. Hölzle. 2007. The case for energy proportional computing. IEEE Computer 40(12): 33–37.

Belady, C., A. Rawson, J. Pefleuger, and T. Cader. 2008. Green Grid Data Center Power Efficiency Metrics: PUE and DCIE. Green Grid white paper #6. Available online at *www.greengrid.org*.

Chang, J., J. Meza, P. Ranganathan, C. Bash, and A. Shah. 2010. Green Server Design: Beyond Operational Energy to Sustainability. Paper presented at HotPower 2010, Vancouver, British Columbia, October 3, 2010.

Condit, J., E.B. Nightingale, C. Frost, E. Ipek, D. Burger, B. Lee, and D. Coetzee. 2009. Better I/O Through Byte-Addressable, Persistent Memory. Presented at Symposium on Operating Systems Principles (SOSP '09), Association for Computing Machinery Inc., October 2009.

Jouppi, N., and Y. Xie. 2009. Emerging technologies and their impact on system design. Tutorial at the International Symposium on Low Power Electronics and Design, 2009.

Kansal, A., J. Liu, A. Singh, R. Nathuji, and T. Abdelzaher. 2009. Semantic-less Coordination of Power Management and Application Performance. In Hotpower 2009 (co-located with SOSP 2009), USENIX, October 10, 2009.

Mayer, M. 2009. The Physics of Data. Talk given at Xerox PARC, August 13, 2009. Available online at *http://www.parc.com/event/936/innovation-at-google.html*.

Meza, J., R. Shih, A. Shah, P. Ranganathan, J. Chang, and C. Bash. 2010. Lifecycle-Based Data Center Design. Presented at ASME International Mechanical Engineering Congress & Exposition, Vancouver, British Columbia, November 14–17, 2010.

Patel, C. 2008. Sustainable IT Ecosystem. Keynote address at the 6th USENIX Conference on File and Storage Technologies, February 26–29, 2008, San Jose, California.

Raghavendra, R., P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu. 2008. No "Power" Struggles: Coordinated Multi-level Power Management for the Data Center. In Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Seattle, Wash., March 2008.

Ranganathan, P. In press. From microprocessor to nanostores: rethinking system building blocks for the data-centric era. IEEE Computer.

Snider, G., 2008. Memristors as Synapses in a Neural Computing Architecture. Presented at Memristor and Memristive Systems Symposium, University of California, Berkeley, November 21, 2008.

Winter. R. 2008. Why are data warehouses growing so fast? An Update on the Drivers of Data Warehouse Growth. Available online at *http://www.b-eye-network.com/view/7188*.

Engineering and Music

Introduction

Daniel Ellis
Columbia University

Youngmoo Kim
Drexel University

Music plays a vital role in every culture on Earth, contributing to the quality of life for billions of people. From the very beginning, engineering and technology have strongly influenced the development of music and musical instruments. The power and potential of this relationship is exemplified in the work of great multidisciplinary thinkers, such as Leonardo Da Vinci, Benjamin Franklin, and Alexander Graham Bell, whose innovations were inspired by their passions for both fields.

In the past decade, there has been a revolution in music—including the aggregation of enormous digital music libraries, the use of portable digital devices to listen to music, and the growing ease of digital distribution. At the same time, advanced tools for creating, manipulating, and interacting with music have become widely accessible. These changes are reshaping the music industry, which has moved far beyond the sale of recordings into personalized music search and retrieval, fan-generated remixes and "mash-ups," and interactive video games.

The rapid proliferation of digital music has also given rise to an explosion of music-related information, and the new field of music information retrieval has been focused on finding methods for managing this data. In the future, listening to music will be transformed by systems that can locate pieces selected from a practically unlimited pool of available music, and fine-tuned to satisfy the mood and preferences of the listener at that moment.

To reconcile quantitative signal content with the complex and obscure perceptions and aesthetic preferences of listeners, music information retrieval requires unprecedented collaboration between experts in signal processing, machine learning, data management, psychology, sociology, and musicology. In the first presentation in this session, Brian Whitman (The Echo Nest) describes advances in the field that combine audio features and myriad music-related data sources to derive metrics for complex judgments, such as similarities among artists and personalized music recommendations.

The next speaker, Douglas Repetto (Columbia University) is the founder of DorkBot, a collection of local groups using technology in non-mainstream ways, usually classified in the category of "art," for want of a better name. He reviews how contemporary composers explore the limits of technology in their art, and the wider experiences of people in the "maker" community who practice what is clearly engineering, but outside of traditional engineering institutions.

Engineering advances have transformed the creative palette available to composers and musicians. Sounds that cannot be produced by physical instruments can be generated electronically, and modern laptop computers have sufficient processing power to perform complex syntheses and audio transformations. Applications of these technologies in collaborative live performance have been pioneered by the Princeton Laptop Orchestra (PLOrk), co-founded by Dan Trueman (Princeton University). In his presentation, he details the technologies that have been developed and used by PLOrk and the orchestra's ongoing efforts to use music technology to engage and energize undergraduate and K–12 students.

The relationship between music and mathematics has fascinated people for many centuries. From the ancient Greeks who considered music a purely mathematical discipline to the serialist composers of the 20th century who relied on numeric combinations to drive compositional choices, countless attempts have been made to derive and define a formal relationship between the two fields. Elaine Chew (University of Southern California) describes her use of mathematics to analyze and understand music and how she incorporates mathematical representations into visualizations for live performance.

Very Large Scale Music Understanding

Brian Whitman
The Echo Nest Corporation

Scientists and engineers around the world have been attempting to do the impossible—and yet, no one can question their motives. When spelled out, "understanding music" by a computational process just feels offensive. How can music, something so personal, something rooted in context, culture, and emotion, ever be labeled by an autonomous process? Even an ethnographical approach—surveys, interviews, manual annotation—undermines the raw effort of musical artists, who will never understand, or even, perhaps, take advantage of what might be learned or created through this research. Music by its very nature resists analysis.

In the past 10 years, I've led two lives—one as a "very long-tail" musician and artist and the other as a scientist turned entrepreneur who currently sells "music intelligence" data and software to almost every major music-streaming service, social network, and record label. How we got from one to the other is less interesting than what it might mean for the future of expression and what I believe machine perception can actually accomplish.

In 1999, I moved to New York City to begin graduate studies at Columbia working on a large "digital government" grant parsing decades of military documents to extract the meaning of acronyms and domain-specific words. At night I would swap the laptops in my bag and head downtown to perform electronic music at various bars and clubs.

As hard as I tried to keep my two lives separate, the walls between them quickly came down when I began to ask my fellow performers and audience members how they learn about music. They responded, "We read websites," "I'm on a discussion board," "A friend e-mailed me some songs," and so on. Obviously, simultaneously with the concurrent media frenzy on peer-to-peer networks

(Napster was just ramping up), a real movement in *music discovery* was underway. Technology had been helping us acquire and make music, but all of a sudden it was being used to communicate and learn about it as well.

With the power to communicate with millions and the seemingly limitless potential of bandwidth and attention, even someone like me could be noticed. So, suitably armed with a background in information retrieval and an almost criminal naiveté about machine learning and signal processing, I quit my degree program and began to concentrate full time on the practice of what is now known as "music information retrieval."

MUSIC INFORMATION RETRIEVAL

The fundamentals of music information retrieval derive from text retrieval. In both cases, you are faced with a corpus of unstructured data. For music, these include time-domain samples from audio files and score data from the compositions. Tasks normally involve extracting readable features from the input and then developing a model from the features. In fact, music data are so unstructured that most music-retrieval tasks began as blind "roulette wheel" predictions: "Is this audio file rock or classical?" (Tzanetakis and Cook, 2002) or "Does this song sound like this one?" (Foote, 1997).

The seductive notion that a black box of some complex nature (usually with hopeful success stories embedded in their names [e.g., neural networks, Bayesian belief networks, support vector machines]) might untangle a mess of audio stimuli in a way that approaches our nervous and perceptual systems' response is intimidating enough. That problem is so complex and so hard to evaluate that it distracts researchers from the much more serious elephantine presence of the emotional connection that underlies the data.

The science of music retrieval is rocked by a massive advance in signal processing or machine learning that solves the problem of label prediction. We can now predict the genre of a song with 100 percent accuracy. The question is what that does for the musician and what it does for the listener. If I knew a song I hadn't heard yet was predicted to be "jazz" by a computer, this might save me the effort of looking up the artist's information, who probably spent years defining his/her expression in terms of or despite these categories. But the jazz label doesn't *tell me anything* about the music, about what I'll feel when I hear it, about how I'll respond or how it will resonate with me individually or in the global community. In short, we had built a black box that could neatly delineate other black boxes but was of no benefit to the very human world of music.

The way out of this feedback loop is to somehow automatically understand reaction and context the same way we do when we actually perceive music. The ultimate contextual-understanding system would be able to gauge my personal reaction and mindset to a piece of music. It would not only know my history and my influences, but would also understand the larger culture around the musical content.

We are all familiar with the earliest approaches to contextual understanding of music—collaborative filtering, a.k.a. "people who buy this also buy this" (Shardanand and Maes, 1995)—and we are just as familiar with its pitfalls. Sales- or activity-based recommenders only know about you in relation to others—their meaning of your music is not what you like but what you've shared with an anonymous hive. The weakness of these filtering approaches becomes apparent when you talk to engaged listeners: "I always see the same bands," "There's never any new stuff," or "This thing *doesn't know me*."

My reaction to the senselessness of filtering approaches was to return to school and begin applying my language-processing background to music—reading about music and not just trying to listen to it. The idea was that, if we could somehow approximate even 1 percent of the data that communities generate about music on the Internet—they review it, they argue about it on forums, they post about shows on their blogs, they trade songs on peer-to-peer networks—we could begin to model large-scale cultural reactions (Whitman, 2005). Thus, in a world of music activity, we would be able to autonomously and anonymously find a new band, for example, that collaborative filtering would never touch (because there are not enough sales data yet) and acoustic filtering would never "get" (because what makes them special is their background or their fan base or something else impossible to calculate from the signal).

THE ECHO NEST

With my co-founder, whose expertise is in musical approaches to signal analysis (Jehan, 2005), I left the academic world to start a private enterprise, "The Echo Nest." We now have 30 people, a few hundred computers, one and a half million artists, and more than ten million songs. Our biggest challenge has been the very large scale of the data. Each artist has an Internet footprint, on average thousands of blog posts, reviews, and forum discussions, all in different languages. Each song is comprised of thousands of "indexable" events, and the song itself might be duplicated thousands of times in different encodings. Most of our engineering work involves dealing with this huge amount of data. Although we are not an infrastructure company, we have built many unique data storage and indexing technologies as a byproduct of our work.

We began the company with the stated goal of indexing everything about music. And over the past five years we have built a series of products and technologies that take the best and most practical aspects of our music-retrieval dissertations and package them cleanly for our customers. The data we collect are necessarily unique. Instead of storing data on relationships between musicians and listeners, or only on popular music, we compute and aggregate a sort of Internet-scale cache of all possible points of information about a song, artist, release, listener, or event. We sell a music-similarity system that compares two songs based on their acoustic and cultural properties. We provide data (automatically generated) on tempo, key, and timbre to mobile applications and streaming services. We track artists' "buzz" on the Internet and sell reports to labels and managers.

The heart of The Echo Nest remains true to our original idea. We strongly believe in the power of data to enable new music experiences. Because we crawl and index everything, we can level the playing field for all types of musicians by taking advantage of the information provided to us by any community on the Internet. Work in music retrieval and understanding requires a sort of wide-eyed passion combined with a large dose of reality. The computer is never going to fully understand what music is about, but if we sample from the right sources often enough and on a large enough scale, the only thing standing in our way is a leap of faith by listeners.

REFERENCES

Foote, J.T. 1997. Content Based Retrieval of Music and Audio. Pp. 138–147 in Multimedia Storage and Archiving Systems II, edited by C.-C.J. Kuo, S-F. Chang, and V. Gudivada. Proceedings of SPIE, Vol. 3229. New York: IEEE.

Jehan, T. 2005. Creating Music by Listening. Dissertation, School of Architecture and Planning, Program in Media Arts and Sciences, Massachusetts Institute of Technology.

Shardanand, U., and P. Maes. 1995. Social Information Filtering: Algorithms for Automating 'Word of Mouth.' Pp. 210-217 in Proceedings of ACM (CHI)'95 Conference on Human Factors in Computing Systems. Vol. 1. New York: ACM Press.

Tzanetakis, G., and P. Cook. 2002. Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing 10(5): 293–302.

Whitman, B. 2005. Learning the Meaning of Music. Dissertation, School of Architecture and Planning, Program in Media Arts and Sciences, Massachusetts Institute of Technology.

Doing It Wrong[1]

Douglas Repetto
Columbia University

Culture is a cumulative and expansive phenomenon; creative communities are in constant flux as their members build on the past, conduct experiments, and fuse bits and pieces of the local and the exotic, the old and the new. Even ancient traditions, seemingly eternal, had precursors. No culture ever appears whole or finished; a culture is always the work of communities, which, consciously or not, shape it to fit their contemporary environment.

It can be tempting to frame conversations about art and music in terms of masterpieces, greatest hits, stars, creative genius, and so on. Works of art are seen as singular objects, the results of exceptional actions by heroic individuals. Masterpieces are somehow definitive, answering questions or offering lessons about creativity. But this is generally not only an inaccurate and unproductive way of thinking about what being creative means, but also a clear path to creative paralysis.

Cultural change, especially on long time scales, is unpredictable. Looking back, it may be temping to draw curves connecting artists or movements to one another, to see particular works or traditions as signposts indicating changes to come or the last gasp of a movement or idea on the way out. But these are, at best, approximations. Hindsight gives the illusion of purposeful movement, of considered progress toward a desired result, and renown or endurance are often mistaken for markers of creative fitness.

[1] Although this paper focuses on experimental music making, the ideas are equally applicable to other creative pursuits, such as visual art, dance, and writing. I believe there are useful analogies and metaphors that link experiments in the arts with topics in science and engineering, but I am not going to try to make those links explicit. Hopefully something here will be compelling to the reader in the context of her own work.

But consider this question: in what way does the ubiquitous presence of Mozart in elevators and dentists' offices provide meaningful guidance to a contemporary human being embarking on a creative life? No one can deny that Mozart reached a pinnacle of creative achievement, but to say that Mozart created works of musical genius says nothing about what we should do today, what music is, or how it can be made.

I take it as axiomatic that the value of a creative work is only partly determined by its material and perceptual qualities. Physiological responses, as well as cultural, social, and intellectual responses are all part of the equation. To paraphrase Brian Whitman of The Echo Nest, to think that a computational analysis of acoustic musical features leads to an understanding of the music sets the bar for "understanding" extremely low (Whitman, 2005). Examples abound: limited editions are valued more highly than unlimited editions; the paintings of Jackson Pollack-like robots are not acclaimed or collected by museums (Piquepaille, 2007); note-perfect Led Zeppelin cover bands do not fill Madison Square Garden with screaming fans; high-tech forensic techniques and boatloads of cash have been dispatched in an attempt to determine whether or not Leonardo drew a small sketch of a pretty young girl (Grann, 2010).

"Value" in these cases can usually be read, at least in part, as monetary value, but focusing exclusively on monetary value misses the point. We cannot know how a work will be valued in the future, what effect, if any, it will have on its own culture or on the culture that follows it.

Although musical innovators throughout history would have articulated these ideas differently, I believe they shared the central tenets that creative acts require deviations from the norm and that creative progress is born not of optimization but of variance. More explicit contemporary engagement with these ideas leads one to the concept of creative research, of music making with goals and priorities that are different from those of their traditional precursors—perhaps sonic friction, in addition to ear-pleasing consonances, for example, or "let's see what happens" rather than "I'm going to tell you a story."

The spirit of "let's see what happens" pervades much contemporary experimental music-making. Here's a very small, personal sample of works I find compelling in the context of musical research, of deviating from the norm, of "doing it wrong."

+++++

"I am sitting in a room" by Alvin Lucier (1990)

This 1969 work on tape has been very influential for several generations of experimental musicians and composers. Lucier recorded himself reading a text describing what he's doing and why. He then played that recording into the room and re-recorded it, after which he played that re-recording into the room and re-re-recorded it, and so on for many iterations. A simple, elegant idea with a surprisingly rich and lovely outcome. And for romantics, a note of poignancy is added to the relationship between reading and effect because Lucier has a slight stutter:

> I am sitting in a room different from the one you are in now. I am recording the sound of my speaking voice and I am going to play it back into the room again and again until the resonant frequencies of the room reinforce themselves so that any semblance of my speech, with perhaps the exception of rhythm, is destroyed. What you will hear, then, are the natural resonant frequencies of the room articulated by speech. I regard this activity not so much as a demonstration of a physical fact, but more as a way to smooth out any irregularities my speech might have.

"I Am Sitting in a Different Room" by Stina Hasse (2010)

Stina Hasse, a Danish student in one of my classes last year, had access to an anechoic chamber (a room in which there is almost no echo or resonance). She decided to make a sort of inverted version of Lucier's piece by translating the text into Danish and expanding it and then using the re-re-recording process to reproduce her own voice. She expected that the experiment would reveal flaws in the design of the anechoic chamber, small resonances that the room's creators had been unable to extinguish. But what happened was both unexpected and wonderful. Instead of bringing out the resonant frequencies of the room (of which there are almost none), it brought out the technological resonances of the equipment she used (the electronic noise of the digital recorder, the acoustic coloration of the microphones, the inevitable hisses and clicks of the physical world).

"51 melodies" by Larry Polansky (1991)

For almost 30 years, Polansky has been playing with the idea of "morphological mutation functions," techniques for smoothly changing musical parameters. In "51 Melodies," a 12-minute composition for two guitars and rhythm section, various flavors of mutation have been applied to the two guitar melodies to bring them in and out of harmonic and rhythmic sync as they move from a source melody in unison at the beginning of the piece to a target melody in unison at the very end. One of the things I like most about this piece is that the guitar parts are very difficult, and the guitarists work hard to play them exactly as notated. They put a lot of hard work into playing precisely notated music that is completely bonkers and nearly incoherent. A work like "51 Melodies" is a classic target for "my six-year-old could do that"-type derision. Its surface features are not particularly "pretty," and appreciation of the music is enhanced by an interest in the conceptual process behind its creation.

"Zero Waste" by Nick Didkovsky (2004)

This is a work for solo piano created on the fly, as the pianist plays. It starts by presenting two algorithmically generated measures of music to the pianist, who then proceeds to play the rather difficult music as accurately as possible. A computer system records the performance, translates it, as best as it can, into rational musical notation, and presents the new notation as the next two measures to be played. Similar to a live performance version of the Lucier piece, "Zero Waste" is a feedback loop that brings out resonances in a system. In this case, the system is the physiological makeup and sight-reading skills of the performer coupled with the performance capture and analysis capabilities of the computer. In performance the developing score is often projected behind the performer, allowing the audience to track the process visually as well as sonically.

"face shock/face copy" by Daito Manabe (2009)

Manabe has been exploring using electrical musical signals to stimulate facial muscles, facial muscle signals to create musical signals, and the transfer of facial gestures between performers via electrical stimulation. If your six-year-old is doing this, you are one lucky parent!

REFERENCES

Didkovsky, N. 2004. Recent Compositions and Performance Instruments Realized in Java Music Specification Language. Pp. 746–749 in Proceedings of the International Computer Music Conference. San Francisco, Calif.: International Computer Music Association.

Grann, D. 2010. The Mark of a Masterpiece. *The New Yorker*, July 12, 2010.

Hasse, S. 2010. I Am Re-Recording in a Room. Personal correspondence.

Lucier, A. 1990. I am Sitting in a Room. New York: Lovely Music, Ltd.

Manabe, D. 2009. face shock/face copy. Personal correspondence.

Piquepaille, R. 2007. A Robot that Paints like Jackson Pollock. Available online at *http://www.zdnet.com*. (Among many others.)

Polansky, L. 1991. 51 Melodies, for two electric guitars and rock band (or any melody instruments). Lebanon, N.H.: Frog Peak Music.

Whitman, B. 2005. Learning the Meaning of Music. Dissertation, School of Architecture and Planning, Program in Media Arts and Sciences, Massachusetts Institute of Technology.

Digital Instrument Building and the Laptop Orchestra

Daniel Trueman
Princeton University

Musical and technological innovations have long gone hand in hand. Historically, this relationship evolved slowly, allowing significant time for musicians to live with and explore new possibilities and enough time for engineers and instrument builders to develop and refine new technologies. The roles of musician and instrument builder have typically been separate; the time and skills required to build an acoustic instrument are typically too great for the builder to also master that instrument, and vice versa.

Today, however, the terms of the relationship have changed. As digital technologies have greatly accelerated the development of new instruments, the separation between these roles has become blurred. However, at the same time, social contexts for exploring new instruments have developed slowly, limiting opportunities for musicians to make music together over long periods of time with these new instruments.

The laptop orchestra is a socially charged context for making music together with new digital instruments while simultaneously developing those instruments. As the role of the performer and builder have merged and the speed with which instruments can be created and revised has increased, the development of musical instruments has become part of the performance of music. In this paper, I look at a range of relevant technical challenges, including speaker design, human-computer interaction (HCI), digital synthesis, machine learning, and networking, in the musical and instrument-building process.

CHALLENGES IN BUILDING DIGITAL INSTRUMENTS

Digital building of musical instruments is a highly interdisciplinary venture that touches on disparate disciplines. In addition to obvious engineering concerns, such as HCI, synthesis, speaker design, and so on, the fields of perception, cognition, and both musical and visual aesthetics also come into play. The basic feedback loop between a player and a generic (acoustic, electronic, or digital) musical instrument is illustrated in Figure 1. In Figure 2, the instrument is "exploded" to reveal the components that might make up a digital musical instrument.

Traveling along the feedback loop in Figure 2, we can see relevant challenges, including sensor design and configuration, haptic feedback systems, computational problems with feature extraction, mapping and synthesis, and amplifier and speaker design. Add to these the ergonomic and aesthetic design of the instrument itself.

Building a compelling digital instrument involves addressing all of these challenges simultaneously, while taking into account various practical and musical concerns, such as the size, weight, and visual appeal of the instrument, its physicality and sound quality, its ease of setup (so it might be "giggable"), and its reliability. The most active researchers in this field are typically either musicians with significant engineering skills or engineers with lifelong musical training.

DIGITAL INSTRUMENT BUILDING IN PRACTICE

My own bowed-sensor-speaker-array (BoSSA) illustrates one solution to the digital instrument problem (Trueman and Cook, 2000). BoSSA (Figure 3) consists of a spherical speaker array integrated with a number of sensors in a violin-inspired configuration. The sensors, which measure bow position, pressure, left-hand finger position, and other parameters, provide approximately a dozen streams of control information. Software is used to process and map these streams to a variety of synthesis algorithms, which are then amplified and sent out through the 12 speaker drivers on the speaker surface.

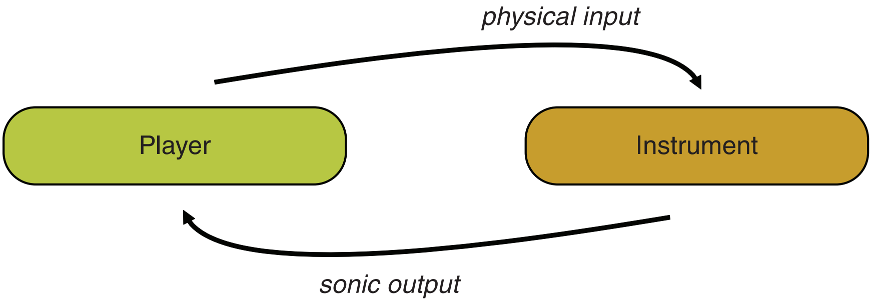


FIGURE 1 Basic player/instrument feedback loop.

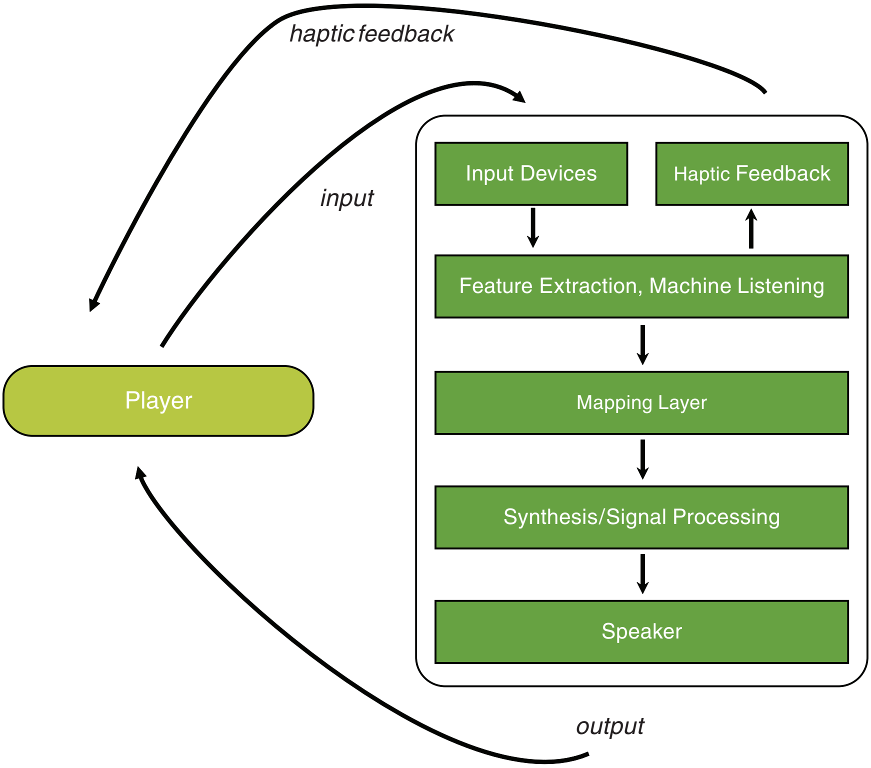


FIGURE 2 Player/digital-instrument feedback loop.

BoSSA, a fairly low-tech, crude instrument made with off-the-shelf sensor and speaker components, is more of a proof of concept than a prototype for future instruments. Its most compelling feature is the integration of sensor and speaker technology into a single localized, intimate instrument.

In recent years, sensor technologies for these kinds of instruments have become much more refined and commercially viable. For example, the K-bow (Figure 4) is a wireless sensor-violin bow that hit the markets just this past year (McMillen, 2008). In terms of elegance and engineering, it surpasses earlier sensor bows (including mine), and if it turns out to be commercially viable, it gives us hope that these kinds of experimental explorations may gain broader traction.

Another set of instruments, built by Jeff Snyder (2010), integrates speaker and sensor technology directly into acoustic resonators so thoroughly that, at first glance, it is not apparent that these are not simply acoustic instruments (Figure 5).

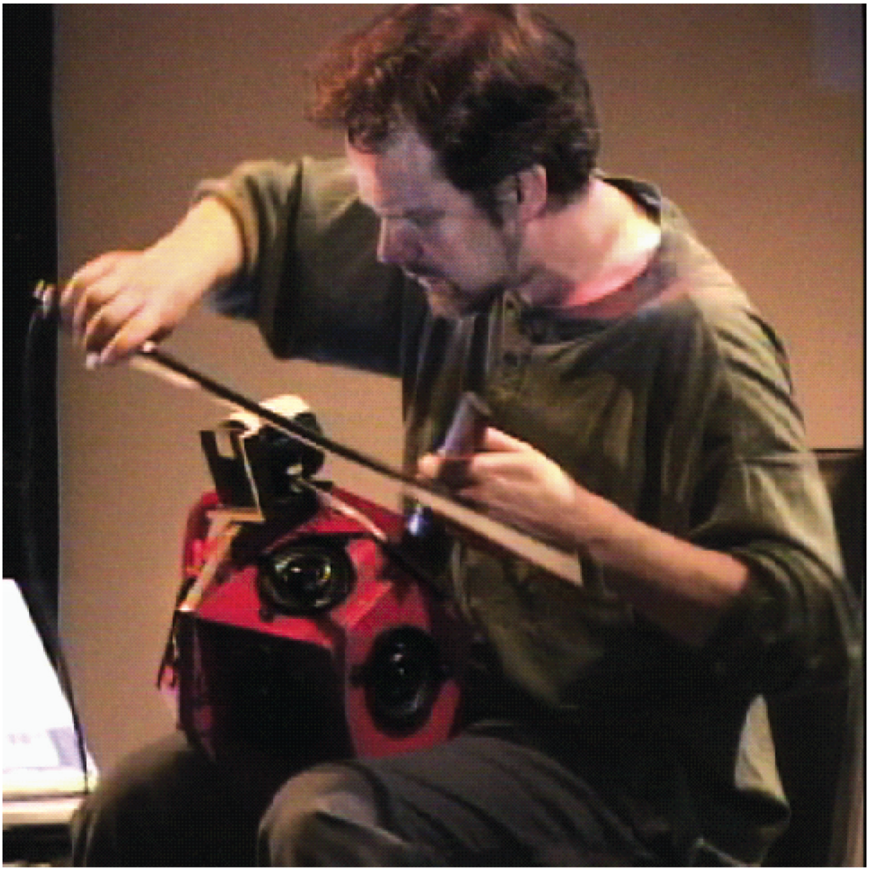


FIGURE 3 The Bowed Sensor Speaker Array.

However, none of these instruments integrates any kind of active haptic feedback. Although the importance of haptic feedback remains an open question, I am convinced it will become more important and more highly valued as digital musical instrument design technologies continue to improve. The physical feedback a violinist receives through the bow and strings is very important to his/her performance technique, as well as to the sheer enjoyment of playing. Thus an ability to digitally manipulate this interface is, to say the least, intriguing.

Researchers developing new haptic technologies for music have demonstrated that they enable new performance techniques. For instance, Edgar Berdahl and his colleagues (2008) at Stanford have developed a haptic drum (among other instruments) that enables a one-handed roll, which is impossible on an acoustic drum (Figure 6).

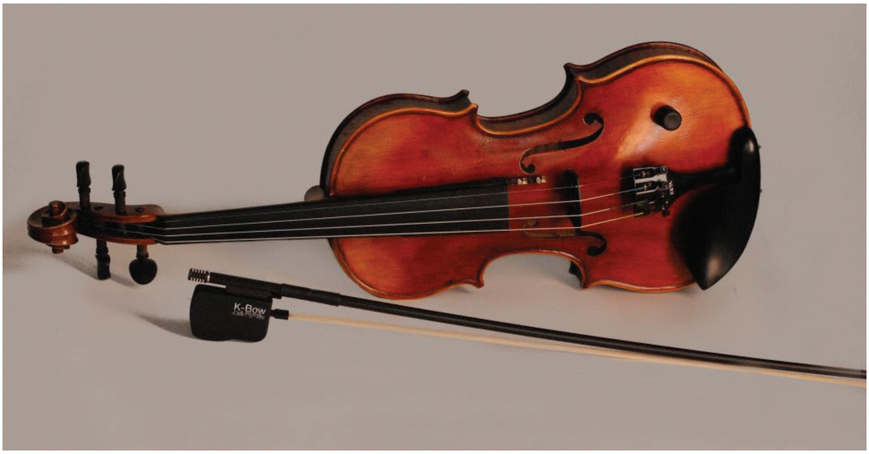


FIGURE 4 The K-bow. Photo courtesy of Keith McMillen.

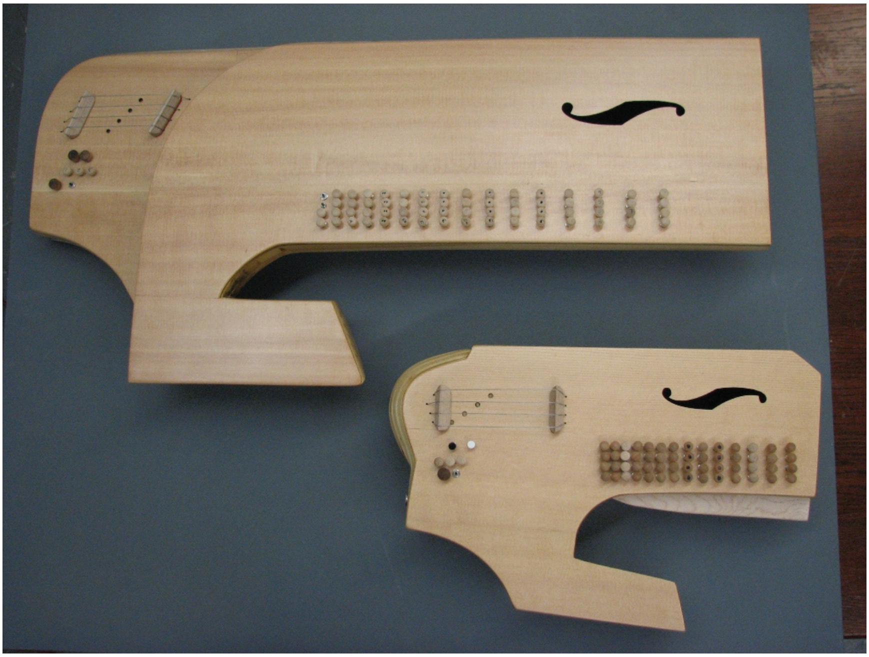


FIGURE 5 The Snyder Contravielles. Photo courtesy of Jeff Snyder.

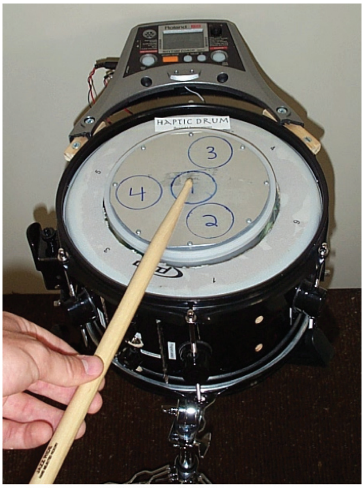


FIGURE 6 Edgar Berdahl's haptic drum. Photo courtesy of Edgar Berdahl.

Perhaps the most compelling aspect of these instruments is in the mapping layer. Here, relationships between body and sound that would otherwise be impossible can be created, and even changed instantaneously. In fact, the mapping layer itself can be dynamic, changing as the instrument is played.

As described earlier, however, creating mappings by manually connecting features to synthesis parameters is always challenging, sometimes practically impossible. One exciting development has been the application of machine-learning techniques to the mapping layer. Rebecca Fiebrink has created an instrument called the Wekinator that gives the musician a way to rapidly and dynamically apply techniques from the Weka machine-learning libraries (Witten and Frank, 2005) to create new mappings (Fiebrink, 2010; Fiebrink et al., 2009).

For instance, a musician might create a handful of sample correspondences (e.g., gesture G with the controller should yield sound S, as set by particular parameter values) to create a data set from the sensor features that can be used by machine-learning algorithms of various sorts to create mapping models that can then be performed. The player can then explore these new models, see how they sound and feel, add new data to the training set, and retrain or simply start over until he/she finds a model that is satisfying. This application of machine learning is unusual in that the end result is usually not known at the outset. Instead, the solutions provided by machine learning feed back into a creative process, finally resulting in an instrument that would have been impossible to imagine fully beforehand. Ultimately, machine learning and the Wekinator are both important in facilitating the creative process.

PLAYING WELL WITH OTHERS

From a musical perspective, computers are terrific at multiplication—they can make a lot from a little (or even from nothing at all).. As a result, much of laptop music is solo, one player who can make a lot of sound, sometimes with little or no effort. One of the main goals of the Princeton Laptop Orchestra (Figure 7) is to create a musically, socially charged environment that can act as a counterweight to this tendency (Trueman, 2007). By putting many musicians in the same room and inviting them to make music together, we force instrument builders and players (often one and the same person) to focus on responsive, subtle instruments that require constant effort and attention, that can turn on a dime in response to what others do, and that force players to break a sweat. The laptop orchestra is, in a sense, a specific, constrained environment within which digital musical instruments can evolve and survive; those that engage us as musicians and enable us to play well with others survive.

While a laptop orchestra is, like any other orchestra, a collection of musical instruments manned by individual players, the possibility of leveraging local wireless networks for musical purposes becomes both irresistible and novel. For instance, a "conductor" might be a virtual entity that sends pulsed information over the network. Or, musicians might set the course of an improvisation on the fly via some sort of musically coded text-messaging system (Freeman, 2010). Instruments themselves might be network enabled, sharing their sensor data with other instruments or players (Weinberg, 2005).

Humans are highly sensitive to variations in timing, and wireless networks are often not up to the challenge. For instance, when sending pulses over the network every 200 milliseconds (ms), humans hear jitter if the arrival times vary by more than 6 ms (a barely noticeable difference); some researchers have concluded that expressive variability occurs even within that window (Iyer, 1998). Preliminary research has been done on the usability of wireless routers instead of computers (Cerqueira and Trueman, 2010), but a perfect wireless router for musical purposes has yet to be found.

CLOSING THOUGHTS

Musical instruments are subjective technologies. Rather than optimal solutions to well defined problems, they reflect individual and cultural values and are ongoing ad hoc manifestations of a social activity that challenges our musical and engineering abilities simultaneously. Musical instruments frame our ability to hear and imagine music, and while they enable human expression and communication, they also are, in and of themselves, expressive and communicative. They require time to evaluate and explore, and they become most meaningful when they are used in a context with other musicians. Therefore, it is essential to view instrument building as a fluid, ongoing process with no "correct" solu-

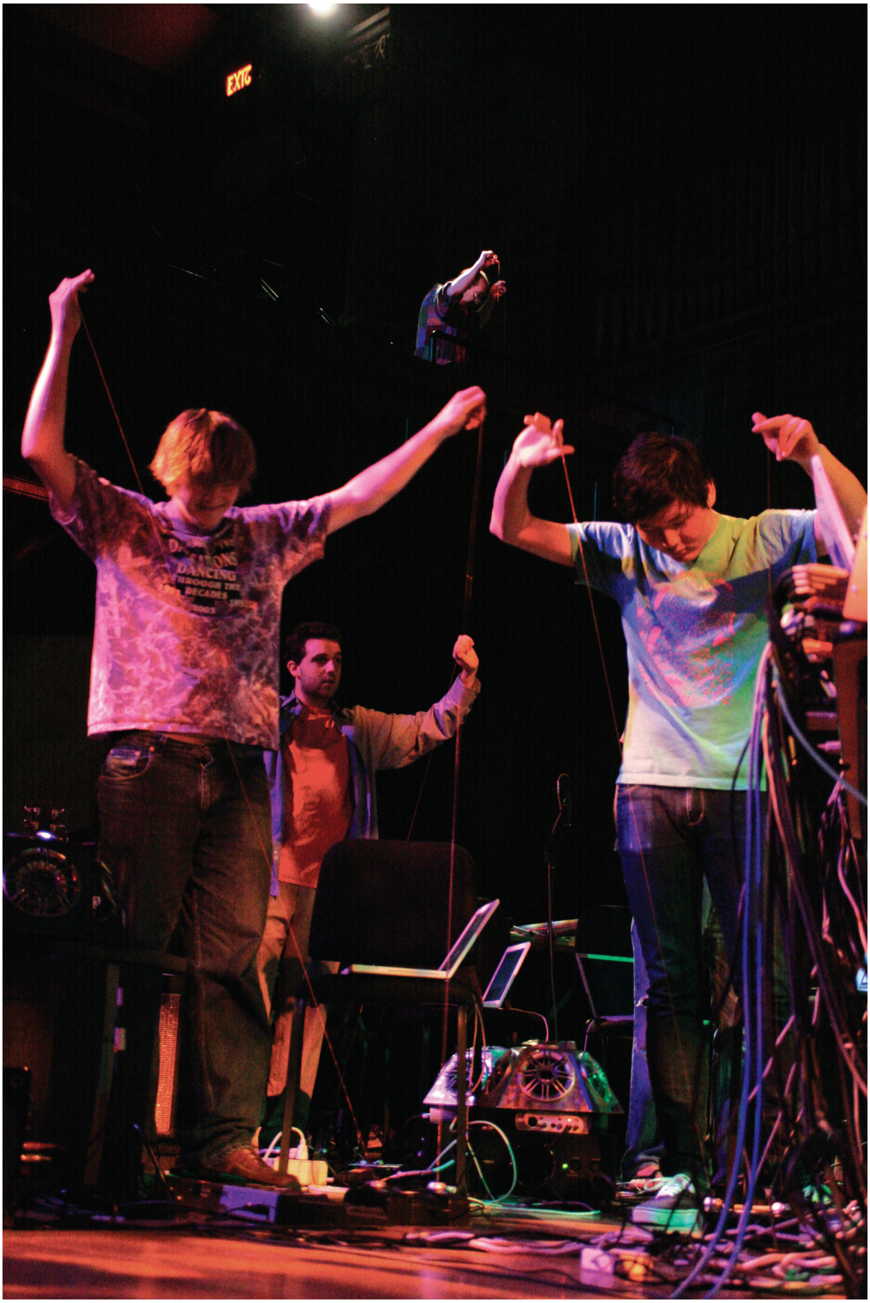


FIGURE 7 The Princeton Laptop Orchestra. Photo courtesy of Lorene Lavora.

tions, a process that requires an expansive context in which these instruments can inspire and be explored.

REFERENCES

Berdahl, E., H.-C. Steiner, and C. Oldham. 2008. Practical Hardware and Algorithms for Creating Haptic Musical Instruments. Eighth International Conference on New Interfaces for Musical Expression, Genova, Italy.

Cerqueira, M., and D. Trueman. 2010. Laptop Orchestra Network Toolkit. Senior Thesis, Computer Science Department, Princeton University.

Fiebrink, R. 2010. Real-Time Interaction with Supervised Learning. Atlanta, Ga.: ACM CHI Extended Abstracts.

Fiebrink, R., P.R. Cook, and D. Trueman. 2009. Play-along Mapping of Musical Controllers. Proceedings of the International Computer Music Conference (ICMC), Montreal, Quebec.

Freeman, J. 2010. Web-based collaboration, live musical performance, and open-form scores. International Journal of Performance Arts and Digital Media 6(2) (forthcoming).

Iyer, V. 1998. Microstructures of Feel, Macrostructures of Sound: Embodied Cognition in West African and African-American Musics. Ph.D. Dissertation, Program in Technology and the Arts, University of California, Berkeley.

McMillen, K. 2008. Stage-Worthy Sensor Bows for Stringed Instruments. Eighth International Conference on New Interfaces for Musical Expression Conference, Genova, Italy.

Snyder, J. 2010. Exploration of an Adaptable Just Intonation System. D.M.A Dissertation, Department of Music**,** Columbia University.

Trueman, D. 2007. Why a laptop orchestra? Organised Sound 12(2): 171–179.

Trueman, D., and P. Cook. 2000. BoSSA: the deconstructed violin reconstructed. Journal of New Music Research 29(2): 121–130.

Weinberg, G. 2005. Local performance networks: musical interdependency through gestures and controllers. Organised Sound 10(3).

Witten, I., and E. Frank. 2005. Data Mining: Practical Machine Learning Tools and Techniques. 2nd ed. San Francisco: Morgan Kaufmann.

Demystifying[1] Music and Its Performance

Elaine Chew
University of Southern California

The mathematical nature of music and its imminently quantifiable attributes make it an ideal medium for studying human creativity and cognition. The architecture of musical structures reveals principles of invention and design. The dynamics of musical ensemble offer models of human collaboration. The demands of musical interaction challenge existing computing paradigms and inspire new ones.

Engineering methodology is integral to systematic study, computational modeling, and a scientific understanding of music perception and cognition, as well as to music making. Conversely, understanding the in-action thinking and problem solving integral to music making and cognition can lead to insights into the mechanics of engineering discovery and design. Engineering-music research can also advance commercial interests in Internet radio, music recommendation and discovery, and video games.

The projects described in this article, which originated at the Music Computation and Cognition Laboratory at the University of Southern California, will give readers a sense of the richness and scope of research at the intersection of engineering and musical performance. The research includes computational music cognition, expression synthesis, ensemble coordination, and musical improvisation.

[1] The title is inspired by George Bugliarello's description of "science and engineering as great untanglers of myths" (Bugliarello, 2003).

ANALYZING AND VISUALIZING TONAL STRUCTURES IN REAL TIME

Most of the music we hear is tonal music, that is, tones (or pitches) organized in time that generate the perception of different levels of stability. The most stable pitch in a musical segment is known as the tonal center, and this pitch gives the key its name. Computational modeling of key-finding dates back to the early days of artificial intelligence (Longuet-Higgins and Steedman, 1971). A popular method for key-finding, devised by Krumhansl and Schmuckler (described in Krumhansl, 1990), is based on the computation of correlation coefficients between the duration profile (vector) of a query stream and experimentally derived probe tone profiles.

Theoretically Efficient Algorithms

In 2000, the author proposed a spiral array model for tonality consisting of a series of nested helices representing pitch classes and major and minor chords and keys. Representations are generated by successive aggregations of their component parts (Chew, 2000). Previous (and most ongoing) research in tonality models has focused only on geometric models (representation) or on computational algorithms that use only the most rudimentary representations. The spiral array attempts to do both. Thus, although the spiral array has many correspondences with earlier models, it can also be applied to the design of efficient algorithms for automated tonal analysis, as well as to the scientific visualization of these algorithms and musical structures (Chew, 2008).

Any stream of notes can generate a center of effect (i.e., a center of gravity of the notes) in the spiral array space. The center of effect generator (CEG) key-finding algorithm based on the spiral array determines the key by searching for the key representation nearest the center of effect of the query stream. The interior-point approach of the CEG algorithm makes it possible to recognize the key quickly and provides a framework for designing further algorithms for automated music analysis.

A natural extension of key-finding is the search for key (or contextual) boundaries. Two algorithms have been proposed for finding key boundaries using the spiral array—one that minimizes the distance between the center of effect of each segment and its closest key (Chew, 2002) and one that finds statistically significant maxima in the distance between the centers of effect of consecutive music segments, without regard to key (Chew, 2005).

The inverse problem of pitch-spelling (turning note numbers, or frequency values, into letter names for music manuscripts that can be read by humans) is essential to automated music transcription. Several variants of a pitch-spelling algorithm using the spiral array have been proposed, such as a cumulative window (Chew and Chen, 2003a), a sliding window (Chew and Chen, 2003b), and a multi-window bootstrapping method (Chew and Chen, 2005).

Robust Working Systems

Converting theoretically efficient algorithms into robust working systems that can stand up to the rigors of musical performance presents many challenges. Using the Software Architecture for Immersipresence framework (François, in press), many of the algorithms described above have been incorporated into the Music on the Spiral Array Real-Time (MuSA.RT) system—an interactive tonal analysis and visualization software (Chew and François, 2003).

MuSA.RT has been used to analyze and visualize music by Pachelbel, Bach, and Barber (Chew and François, 2005). These visualizations have been presented in juxtaposition to Sapp's keyspaces (2005) and Toiviainen's self-organizing maps (2005). MuSA.RT has also been demonstrated internationally and was presented at the AAAS Ig Nobel session in 2008.

Figure 1 shows MuSA.RT in concert at the 2008 ISMIR conference at Drexel University in Philadelphia. As the author plays the music (in this case, Ivan Tcherepnin's *Fêtes–Variations on Happy Birthday)* on the piano, MuSA.RT performs real-time analysis of the tonal patterns. At the same time, visualizations of the tonal structures and trajectories are projected on a large screen, revealing the evolution of tonal patterns—away from C major and back—over a period of more than ten minutes.

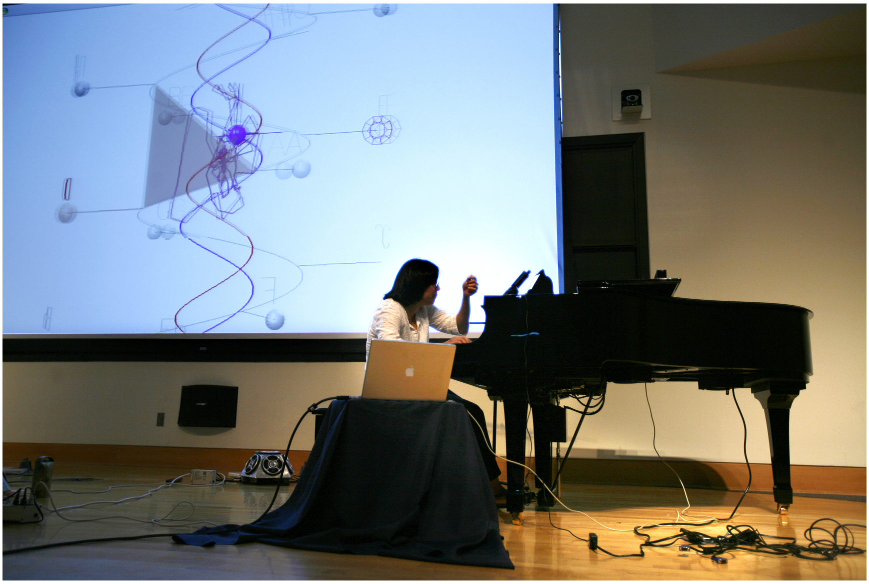


FIGURE 1 MuSA.RT in concert at the 2008 ISMIR conference at Drexel University. Source: © The Philadelphia Enquirer. Reprinted with permission.

In MuSA.RT version 2.7, the version shown in Figure 2 (as well as Figure 1), the pitch-class helix and pitch names are shown in silver, the major/minor triad helices are hidden, and the major/minor key helices, which are shown in red and blue, respectively, appear as a double helix in the structure's core. When a note is sounded, silver spheres appear on the note name, a short-term center of effect tracks the local context, and the triad closest to it lights up as a pink/blue triangle. A long-term center of effect tracks the larger scale context, and a sphere appears on the closest key, the size of which is inversely proportional to the center of effect-key distance. Lighter violet and darker indigo trails trace the history of the short-term and long-term centers of effect, respectively

An analysis of humor in the music of P.D.Q. Bach (a.k.a. Peter Schickele) by Huron (2004) revealed that many of Schickele's humorous effects are achieved by violating expectations. Using MuSA.RT to analyze P.D.Q. Bach's *Short-Tempered Clavier*, we visualized excessive repetition, as well as incongruous styles, improbable harmonies, and surprising tonal shifts (all of which appeared as far-flung trajectories in the spiral array space) (Chew and François, 2009). Figure 3 shows visualizations of a few of these techniques.

By using the sustain pedal judiciously, and by accenting different notes through duration or stress, a performer can influence the listener's perception of tonal structures. The latest versions of MuSA.RT take into account pedal effects in the computing of tonal structures (Chew and François, 2008). Because center of effect trails react directly to the timing of note soundings, no two human performances result in exactly the same trajectories.

We are currently working on extending spiral array concepts to a higher dimension to represent tetrachords (Alexander et al., in preparation). The resulting pentahelix has a direct correspondence to the orbifold model (a geometric model

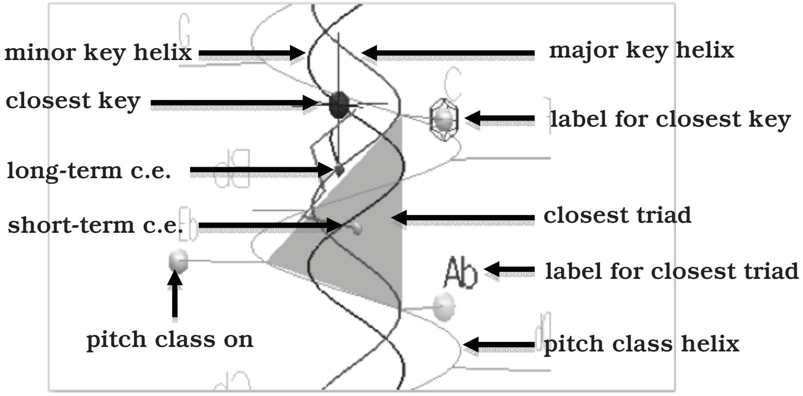
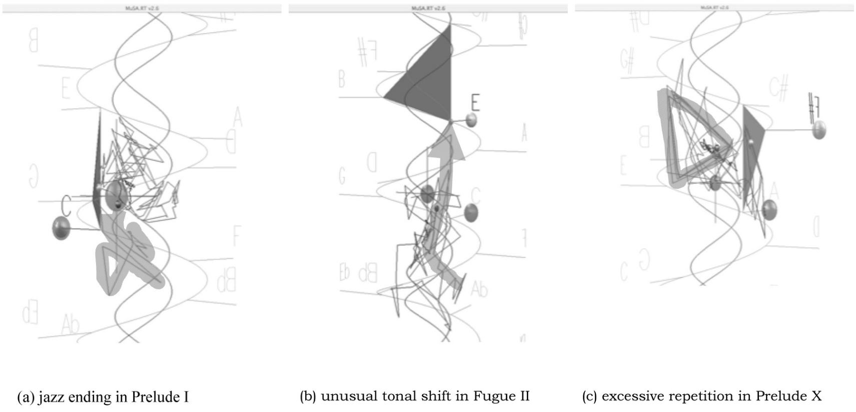


FIGURE 2 Components of the spiral array in MuSA.RT labeled. Color figure available online at *http://www.nap.edu/catalog.php?record_id=13043*.



(a) jazz ending in Prelude I (b) unusual tonal shift in Fugue II (c) excessive repetition in Prelude X

FIGURE 3 Humor devices in P.D.Q. Bach's Short-Tempered Clavier visualized in MuSA.RT. Source: Adapted from Chew and François, 2009.

for representing chords using a topological space with an orbifold structure) (Callendar et al., 2008; Tymoczko, 2006).

EXPERIENCING MUSIC PERFORMANCE THROUGH DRIVING

Not everyone can play an instrument well enough to execute expressive interpretations at will, but almost everyone can drive a car, at least in a simulation. The Expression Synthesis Project (ESP), based on a literal interpretation of music as locomotion, creates a driving interface for expressive performance that enables even novices to experience the kind of embodied cognition characteristic of expert musical performance (Figure 4).

In ESP (Chew et al., 2005a), the driver uses an accelerator and brake pedal to increase or decrease the speed of the car (music playback). The center line segments in the road approach at one per beat, thus providing a sense of tempo (beat rate and car velocity); this is shown on the speedometer in beats per minute. Suggestions to slow down or speed up are embedded in bends in the road and straight sections, respectively. Thus the road map, which corresponds to an interpretation of a musical piece, often reveals the underlying structure of the piece.

Despite the embedded suggestions, the user is free to choose her/his desired tempo trajectory. In addition, more than one road map (interpretation) can correspond to the same piece of music (Chew et al., 2006). As part of the system design (using SAI), a virtual radius mapping strategy ensures smooth tempo transitions (Figure 5), a hallmark of expert performance, even if the user's driving behavior is erratic (Liu et al., 2006).

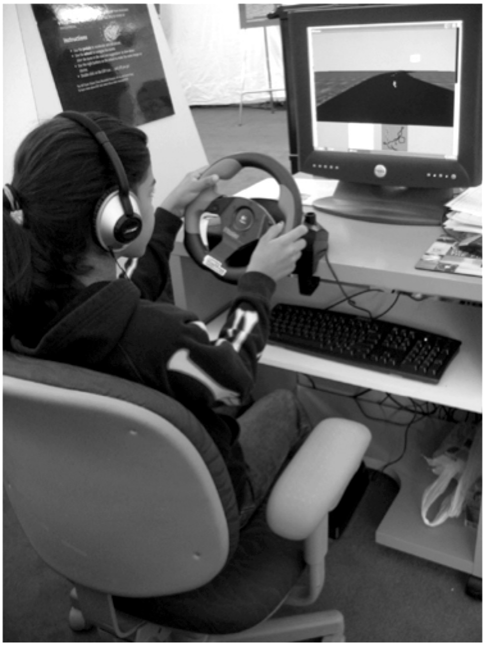


FIGURE 4 ESP at the USC Festival 125 Pavilion. Photo by Elaine Chew.

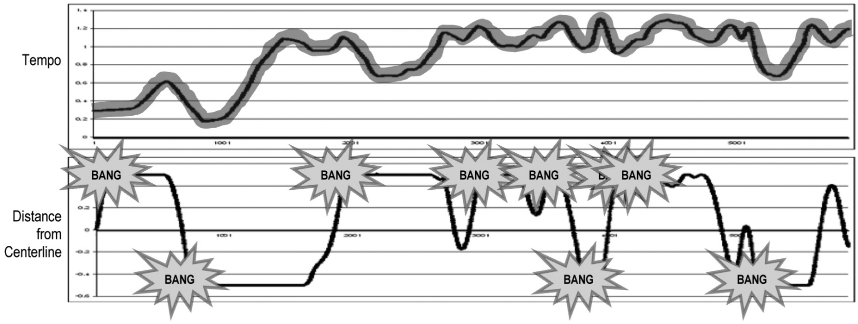


FIGURE 5 Virtual radius mapping ensures smooth tempo transitions despite erratic driving. Source: Adapted from Liu et al., 2006.

CHARTING THE DYNAMICS OF ENSEMBLE COORDINATION

Remote collaboration is integral to our increasingly global and distributed workplaces and society. Musical ensemble performance offers a unique framework through which to examine the dynamics of close collaboration and the challenges of human interaction in the face of network delays.

In a series of distributed immersive performance (DIP) experiments, we recorded the Tosheff Piano Duo performing Poulenc's *Piano Sonata for Four Hands* with auditory delays ranging from 0 milliseconds (ms) to 150 ms. In one experiment, performers heard themselves immediately but heard the other performer with delays. Both performers reported that delays of more than 50 ms caused them to struggle to keep time and compromised their interpretation of the music (Chew et al., 2004).

By delaying each performer's playing to his/her own ears so that it aligned with the incoming signal of the partner's playing (see Figure 6), we created a more satisfying experience for the musicians that allowed them to adapt to the conditions of delay and increased the delay threshold to 65 ms (Chew et al., 2005c), even for the *Final*, a fast and rhythmically demanding movement. Quantitative analysis showed a marked increase in the range of segmental tempo strategies between 50 ms and 75 ms and a marked decline at 100 ms and 150 ms (Chew et al., 2005b).

Most other experiments have treated network delay as a feature of free improvisation rather than a constraint of classical performance. A clapping experiment at Stanford showed that, as auditory delays increased, pairs of musi-

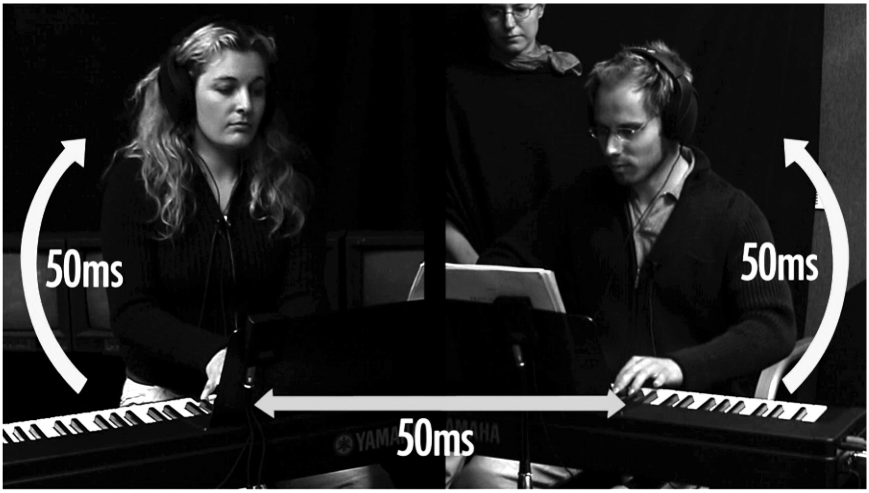


FIGURE 6 Delaying each player's sound to his/her own ears to align with incoming audio of the other player's sound.

cians slowed down over time. They sped up modestly when delays were less than 11.5 ms (Chafe and Gurevich, 2004). In similar experiments at the University of Rochester, Bartlette et al. (2006) found that latencies of more than 100 ms profoundly impacted the ability of musicians to play as a duo. Using more recent tools and techniques for music alignment and performance analysis, we can now conduct further experiments with the DIP files to create detailed maps of ensemble dynamics (Wolf and Chew, 2010).

ON-THE-FLY "ARCHITECTING" OF A MUSICAL IMPROVISATION

Multimodal interaction for musical improvisation (MIMI) was created as a stand-alone performer-centric tool for human-machine improvisation (François et al., 2007). Figure 7 shows MIMI at her concert debut earlier this year.

MIMI takes user input, creates a factor oracle, and traverses it stochastically to generate recombinations of the original input (Assayag and Dubnov, 2004). In previous improvisation systems (Assayag et al., 2006; Pachet, 2003), performers reacted to machine output without prior warning. MIMI allows for more natural interaction by providing visual cues to the origins of the music being generated and by giving musicians a 10-second heads up and review of the musical material.

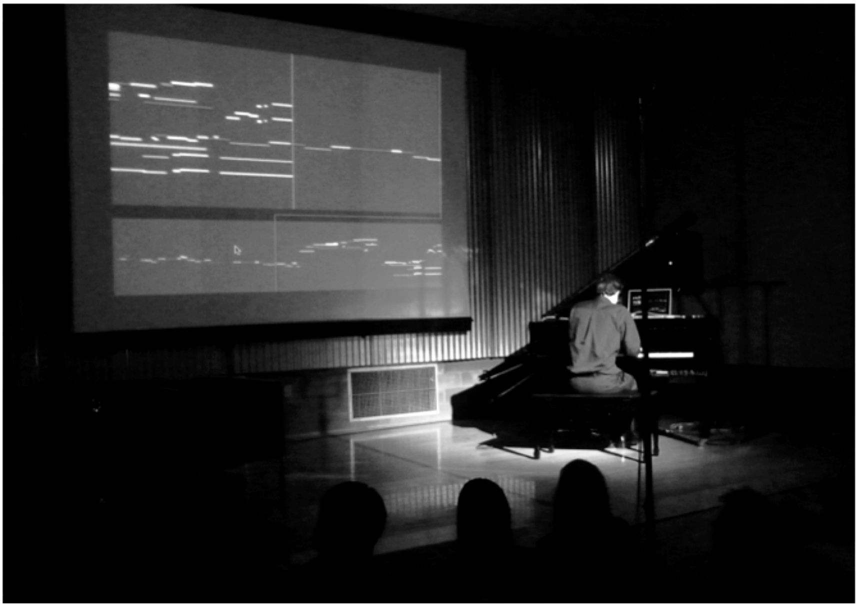


FIGURE 7 MIMI's concert debut at the People Inside Electronics Concert in Pasadena, California, with Isaac Schankler. Photo by Elaine Chew.

MIMI's interface allows the performer to decide when the machine learns and when the learning stops, to determine the recombination rate (the degree of fragmenting of the original material), to decide when the machine starts and stops improvising, the loudness of playback, and when to clear the memory (François et al., 2010). By tracking these decisions as the performance unfolds, we can build a record of how an improviser "architects" an improvisation. As work with MIMI continues and performance decisions are documented, our understanding of musical (and hence, human) creativity and design will improve.

OPEN COURSEWARE

Reviews of music-engineering research as open courseware can be found at *www-scf.usc.edu/~ise575* (Chew, 2006). Each website includes a reading list, presentations and student reviews of papers, and links to final projects. For the 2010 topic in Musical Prosody and Interpretation, highlights of student projects include Brian Highfill's time warping of a MIDI (musical instrument digital interface) file of *Wouldn't It Be Nice* to align with the Beach Boys' recording of the same piece; Chandrasekhar Rajagopal's comparison of guitar and piano performances of Granados' *Spanish Dance No. 5*; and Balamurali Ramasamy Govindaraju's charting of the evolution of vibrato in violin performance over time.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation (NSF) under Grants 0321377 and 0347988 and Cooperative Agreement EEC-9529152. Opinions, findings, and conclusions or recommendations expressed in this article are those of the author and not necessarily of NSF.

REFERENCES

Alexander, S., E. Chew, R. Rowe, and S. Rodriguez. In preparation. The Pentahelix: A Four-Dimensional Realization of the Spiral Array.

Assayag, G., and S. Dubnov. 2004. Using factor oracles for machine improvisation. Soft Computing 8(9): 604–610.

Assayag, G., G. Bloch, M. Chemillier, A. Cont, and S. Dubnov. 2006. Omax Brothers: A Dynamic Topology of Agents for Improvization Learning. In Proceedings of the Workshop on Audio and Music Computing for Multimedia, Santa Barbara, California.

Bartlette, C., D. Headlam, M. Bocko, and G. Velikic. 2006. Effect of network latency on interactive musical performance. Music Perception 24(1): 49–62.

Bugliarello, G. 2003. A new trivium and quadrivium. Bulletin of Science, Technology and Society 23(2): 106–113.

Callendar, C., I. Quinn, and D. Tymoczko. 2008. Generalized voice-leading spaces. Science 320(5874): 346–348.

Chafe, C., and M. Gurevich. 2004. Network Time Delay and Ensemble Accuracy: Effects of Latency, Asymmetry. In Proceedings of the 117th Conference of the Audio Engineering Society, San Francisco, California.

Chew, E. 2000. Towards a Mathematical Model of Tonality. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2000.

Chew, E. 2002. The Spiral Array: An Algorithm for Determining Key Boundaries. Pp. 18–31 in Music and Artificial Intelligence, edited by C. Anagnostopoulou, M. Ferrand, and A. Smaill. Springer LNCS/LNAI, 2445.

Chew, E. 2005. Regards on two regards by Messiaen: post-tonal music segmentation using pitch context distances in the spiral array. Journal of New Music Research 34(4): 341–354.

Chew, E. 2006. Imparting Knowledge and Skills at the Forefront of Interdisciplinary Research—A Case Study on Course Design at the Intersection of Music and Engineering. In Proceedings of the 36th Annual ASEE/IEEE Frontiers in Education Conference, San Diego, California, October 28–31.

Chew, E. 2008. Out of the Grid and into the Spiral: Geometric Interpretations of and Comparisons with the Spiral-Array Model. W. B. Hewlett, E. Selfridge-Field, and E. Correia Jr., eds. Computing in Musicology 15: 51–72.

Chew, E., and Y.-C. Chen. 2003a. Mapping MIDI to the Spiral Array: Disambiguating Pitch Spellings. Pp. 259–275 in Proceedings of the 8th INFORMS Computing Society Conference, Chandler, Arizona, January 8–10, 2003.

Chew, E., and Y.-C. Chen. 2003b. Determining Context-Defining Windows: Pitch Spelling Using the Spiral Array. In Proceedings of the 4th International Conference for Music Information Retrieval, Baltimore, Maryland, October 26–30, 2003.

Chew, E., and Y.-C. Chen. 2005. Real-time pitch spelling using the spiral array. Computer Music Journal 29(2): 61–76.

Chew, E., and A.R.J. François. 2003. MuSA.RT—Music on the Spiral Array. Real-Time. Pp. 448–449 in Proceedings of the ACM Multimedia ‘03 Conference, Berkeley, California, November 2–8, 2003.

Chew, E., and A.R.J. François. 2005. Interactive multi-scale visualizations of tonal evolution in MuSA. RT Opus 2. ACM Computers in Entertainment 3(4): 16 pages.

Chew, E. and A.R.J. François. 2008. MuSA.RT and the Pedal: the Role of the Sustain Pedal in Clarifying Tonal Structure. In Proceedings of the Tenth International Conference on Music Perception and Cognition, Sapporo, Japan, August 25–29, 2008.

Chew, E., and A.R.J. François. 2009. Visible Humour—Seeing P.D.Q. Bach's Musical Humour Devices in *The Short-Tempered Clavier* on the Spiral Array Space. In Mathematics and Computation in Music—First International Conference, edited by T. Noll and T. Klouche. Springer CCIS 37.

Chew, E., R. Zimmermann, A.A. Sawchuk, C. Kyriakakis, C. Papadopoulos, A.R.J. François, G. Kim, A. Rizzo, and A. Volk. 2004. Musical Interaction at a Distance: Distributed Immersive Performance. In Proceedings of the 4th Open Workshop of MUSICNETWORK, Barcelona, Spain, September 15–16, 2004.

Chew, E., A.R.J. François, J. Liu, and A. Yang. 2005a. ESP: A Driving Interface for Musical Expression Synthesis. In Proceedings of the Conference on New Interfaces for Musical Expression, Vancouver, B.C., Canada, May 26–28, 2005.

Chew, E., A. Sawchuk, C. Tanoue, and R. Zimmermann. 2005b. Segmental Tempo Analysis of Performances in Performer-Centered Experiments in the Distributed Immersive Performance Project. In Proceedings of International Conference on Sound and Music Computing, Salerno, Italy, November 24–26, 2005.

Chew, E., R. Zimmermann, A. Sawchuk, C. Papadopoulos, C. Kyriakakis, C. Tanoue, D. Desai, M. Pawar, R. Sinha, and W. Meyer. 2005c. A Second Report on the User Experiments in the Distributed Immersive Performance Project. In Proceedings of the 5th Open Workshop of MUSICNETWORK, Vienna, Austria, July 4–5, 2005.

Chew, E., J. Liu, and A.R.J. François. 2006. ESP: Roadmaps as Constructed Interpretations and Guides to Expressive Performance. Pp. 137–145 in Proceedings of the First Workshop on Audio and Music Computing for Multi-media, Santa Barbara, California, October 27, 2006.

François, A.R.J. In press. An architectural framework for the design, analysis and implementation of interactive systems. The Computer Journal.

François, A.R.J., E. Chew, and C.D. Thurmond. 2007. Visual Feedback in Performer-Machine Interaction for Musical Improvisation. In Proceedings of International Conference on New Instruments for Musical Expression, New York, June 2007.

François, A.R.J., I. Schankler, and E. Chew. 2010. MIMI4x: An Interactive Audio-Visual Installation for High-Level Structural Improvisation. In Proceedings of the Workshop on Interactive Multimedia Installations and Digital Art, Singapore, July 23, 2010.

Huron, D. 2004. Music-engendered laughter: an analysis of humor devices in P.D.Q. Bach. Pp. 700–704 in Proceedings of the International Conference on Music Perception and Cognition, Evanston, Ill.

Krumhansl, C. 1990. Cognitive Foundations of Musical Pitch. Oxford, U.K.: Oxford University Press.

Liu, J., E. Chew, and A.R.J. François. 2006. From Driving to Expressive Music Performance: Ensuring Tempo Smoothness. In Proceedings of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, Hollywood, California, June 14–16. 2006.

Longuet-Higgins, H.C., and M. Steedman. 1971. On interpreting Bach. Machine Intelligence 6: 221–241.

Pachet, F. 2003. The continuator: musical interaction with style. Journal of New Music Research 32(3): 333–341.

Sapp, C. 2005. Visual hierarchical key analysis. ACM Computers in Entertainment 3(4): 19 pages.

Toiviainen, P. 2005. Visualization of tonal content with self-organizing maps and self-similarity matrices. ACM Computers in Entertainment 3(4): 10 pages.

Tymoczko, D. 2006. The geometry of musical chords. Science 313(5783): 72–74.

Wolf, K., and E. Chew. 2010. Evaluation of Performance-to-Score MIDI Alignment of Piano Duets. In online abstracts of the International Conference on Music Information Retrieval, Utrecht, The Netherlands, August 9–13, 2010.

Autonomous Aerospace Systems

Introduction

MICHEL INGHAM
Jet Propulsion Laboratory

JACK LANGELAAN
Pennsylvania State University

Autonomous systems have become critical to the success of military and scientific missions. Vehicles like the Mars Exploration Rovers, which can autonomously drive through a cluttered environment to a goal and autonomously identify and extract features of scientific interest (e.g., dust devils and clouds) from images taken by onboard cameras, and the Boeing X-45A unmanned air vehicle (UAV), which demonstrated the first autonomous flight of a high-performance, combat-capable UAV and the first autonomous multi-vehicle coordinated flight, have reduced the level of human intervention from inner-loop control to high-level supervision.

However, human involvement is still a critical component of robotic systems. In some cases, it is necessary from a legal and arguably moral standpoint (e.g., in autonomous strike missions), but in most cases humans are necessary because of the limitations of current technology. For example, it is still impossible for a robot to navigate autonomously along a crowded sidewalk or for a robotic explorer to demonstrate initiative or "decide what is interesting." Even recovering from an error, such as a stuck wheel or an actuator fault, generally requires human intervention.

The presentations in this session focus on aspects of autonomy that will bring robotic systems from controlled devices that can function for a few minutes without human intervention to systems that can function autonomously for days or weeks in poorly characterized, or even unknown, environments. The speakers, who represent academia, government, and industry, cover both aeronautical and space autonomous systems. Their presentations have been organized to progress from a single vehicle (including human interaction with the vehicle) to teams of robots to the incorporation of autonomous unmanned air systems into the National Air Transportation System.

The first talk, by Mark Campbell (Cornell University), focuses on (1) techniques for enabling "intelligence" in autonomous systems through probabilistic models of the environment and (2) the integration of human operators into the control/planning loop. Chad Frost (NASA Ames Research Center) provides an overview of the challenges to increasing automation in NASA's current and future space missions, highlights examples of successful autonomous systems, and discusses some of the lessons learned from those experiences.

The subject of the third talk, by Stefan Bieniawski (Boeing Research and Technology), is the role of health awareness in multi-vehicle autonomous systems. He describes how such systems can address failures of components in a vehicle or the failure of a vehicle in the team. In the final presentation, Ella Atkins (University of Michigan) discusses the formidable challenges associated with the safe, efficient integration of unmanned air systems into airspace currently traveled by manned aircraft and the importance of automation and autonomy in the deployment of the next-generation air transportation system (NextGen).

Intelligent Autonomy in Robotic Systems

Mark Campbell
Cornell University

Automation is now apparent in many aspects of our lives, from aerospace systems (e.g., autopilots) to manufacturing processes (e.g., assembly lines) to robotic vacuum cleaners. However, although many aerospace systems exhibit some autonomy, it can be argued that such systems could be far more advanced than they are. For example, although autonomy in deep-space missions is impressive, it is still well behind autonomous ground systems. Reasons for this gap range from proximity to the hardware and environmental hardships to scientists tending not to trust autonomous software for projects on which many years and dollars have been spent.

Looking back, the adoption of the autopilot, an example of advanced autonomy for complex systems, aroused similar resistance. Although autopilot systems are required for most commercial flights today, it took many years for pilots to accept a computer flying an aircraft.

Factors that influence the adoption of autonomous systems include reliability, trust, training, and knowledge of failure modes. These factors are amplified in aerospace systems where the environment/proximity can be challenging and high costs and human lives are at stake. On deep-space missions, for example, which take many years to plan and develop and where system failure can often be traced back to small failures, there has been significant resistance to the adoption of autonomous systems.

In this article, I describe two improvements that can encourage more robust autonomy on aerospace missions: (1) a deeper level of intelligence in robotic systems; and (2) more efficient integration of autonomous systems and humans.

INTELLIGENCE IN ROBOTICS

Current robotic systems work very well for repeated tasks (e.g., in manufacturing). However, their long-term reliability for more complex tasks (e.g., driving a car) is much less assured. Interestingly, humans provide an intuitive benchmark, because they perform at a level of deep intelligence that typically enables many complex tasks to be performed well. However, this level of intelligence is difficult to emulate in software. Characteristics of this deep intelligence include learning over time, reasoning about and overcoming uncertainties/new situations as they arise, and developing long-term strategies.

Researchers in many areas are investigating the concept of deeper intelligence in robotics. For our purposes, we look into three research topics motivated in part by tasks that humans perform intelligently: (1) tightly integrated perception, anticipation, and planning; (2) learning; and (3) verified plans in the presence of uncertainties.

Tightly Integrated Perception, Anticipation, and Planning

As robotic systems have matured, an important advancement has been the development of high-throughput sensors. Consider, for example, Cornell's autonomous driving vehicle, one of only six that completed the 2007 DARPA Urban Challenge (DUC) (Figure 1). The vehicle (robot) has a perception system with a 64-scan lidar unit (100Mbits/sec), 4-scan lidar units (10MBits/sec), radars, and cameras (1,200Mbits/sec).

Although the vehicle's performance in the DUC was considered a success (Iagnemma et al., 2008; Miller et al., 2008), there were many close calls, several small collisions, and a number of human-assisted restarts. In fact, the fragility of practical robotic intelligence was apparent when many simple mistakes in perception cascaded into larger failures.

One critical problem was the mismatch between perception, which is typically *probabilistic* because sensors yield data that are inherently uncertain compared to the true system, and planning, which is *deterministic* because plans must be implemented in the real world. To date, perception research typically provides robotic planners with probabilistic "snapshots" of the environment, which leads to "reactive," rather than "intelligent," behaviors in autonomous robots.

Aerospace systems have similar problems. Figure 2 shows a cooperative unmanned air vehicle (UAV) system for searching out and tracking objects of interest (Campbell and Whitacre, 2007), such as tuna fish or survivors of hurricanes and fires. System failures include searching only part of an area, losing track of objects when they move out of sight (e.g., behind a tree or under a bridge), or vibrations or sensor uncertainty aboard the aircraft.

Overcoming these problems will require new theory that provides tighter linkage between sensors/probabilistic perception and actions/planning (Thrun et

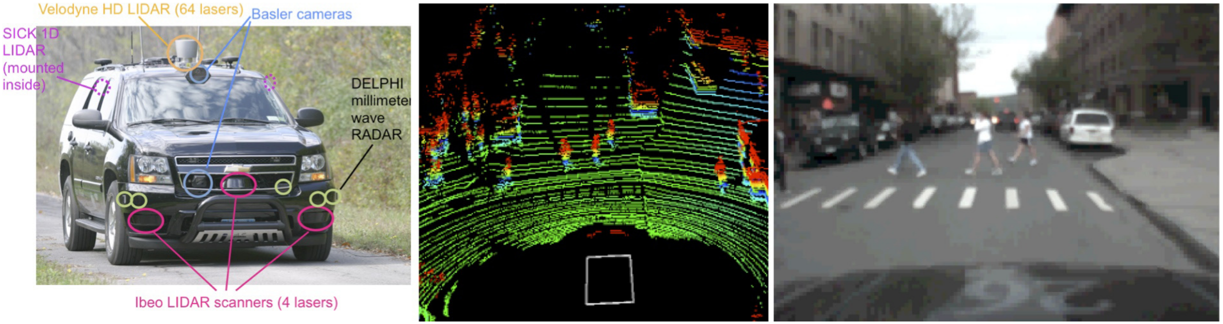

Velodyne HD LIDAR (64 lasers)
Basler cameras
SICK 1D LIDAR (mounted inside)
DELPHI millimeter wave RADAR
Ibeo LIDAR scanners (4 lasers)

FIGURE 1 The autonomous driving vehicle developed by Cornell University, which can handle large amounts of data intelligently. Left: Robot with multi-modal sensors for perceiving the environment. Middle: Screenshot of a 64-scan lidar unit. Right: Screenshot of a color camera.

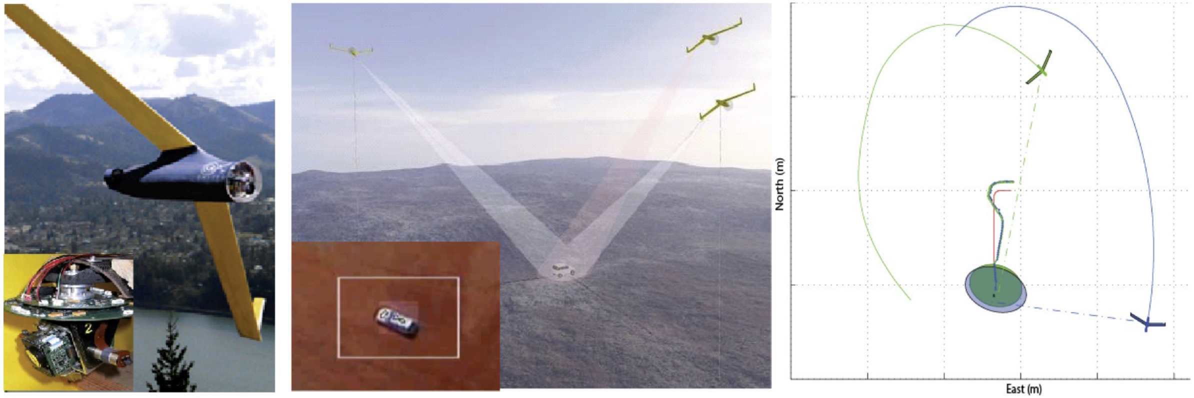


FIGURE 2 Multiple UAV system. Left: SeaScan UAV with a camera-based turret. Center: Notional illustration of cooperative tracking using UAVs. Right: Flight test data of two UAVs tracking a truck over a communication network with losses (e.g., dropped packets).

al., 2005). Given the high data throughput of the sensors on most systems, a key first step is to convert "data to information." This will require fusing data from many sensors to provide an accurate picture of the static, but potentially dynamic, environment, including terrain type and the identity and behaviors of obstacles (Diebel and Thrun, 2006; Schoenberg et al., 2010).

Take driving, for example. A human is very good at prioritizing relatively small amounts of information (i.e., from the eyes), as well as a priori learned models. If an object is far away, the human typically focuses on the "gist" of the scene, such as object type (Ross and Oliva, 2010). If an object is closer, such as when something is about to hit a car, the primary focus is on proximity, rather than type (Cutting, 2003). To ensure that large amounts of data are transformed into critical information that can be used in decision making, we need new representations and perception methods, particularly methods that are computationally tractable.

Plans must then be developed based on probabilistic information. Thus, the second essential step is to convert "information to decisions," which will require a new paradigm to ensure that planning occurs to a particular level of probability, while also incorporating changes in the environment, such as the appearance of objects (from the perceived information and a priori models). This is especially important in dynamic environments, where the behavior and motion of objects are strongly related to object type.

For autonomous driving (Figure 1), important factors for planning include the motion of other cars, cyclists, and pedestrians in the context of a map (Blackmore et al., 2010; Hardy and Campbell, 2010; Havlak and Campbell, 2010). For cooperative UAVs (Figure 2), important factors include the motion of objects and other UAVs (Grocholsky et al., 2004; Ousingsawat and Campbell, 2007). Although humans typically handle these issues well by relying on learned models of objects, including their motions and behaviors, developing robotic systems that can handle these variables reliably can be computationally demanding (McClelland and Campbell, 2010).

For single and cooperative UAV systems, such as those used for search and rescue or defense missions, data are typically in the form of optical/infra-red video and lidar. The necessary information includes detecting humans, locating survivors in clutter, and tracking moving cars—even if there are visual obstructions, such as trees or buildings. Actions based on this information then include deciding where to fly, a decision strongly influenced by sensing and coverage, and deciding what information to share (among UAVs and/or with ground operators). Ongoing work in sensor fusion and optimization-based planning have focused on these problems, particularly as the number of UAVs increases (Campbell and Whitacre, 2007; Ousingsawat and Campbell, 2007).

Learning

Humans typically drive very well because they learn *safely* over time (rules, object types and motion, relative speeds, etc.). However, for robots, driving well is very challenging, especially when uncertainties are prevalent. Consider Figure 3, which shows a map of the DUC course, with an overlay of 53 instances of emergency slamming of brakes by Cornell's autonomous vehicle. Interestingly, many of these braking events occurred during multiple passes near the same areas; the most frequent (18 times) took place near a single concrete barrier jutting out from the others, making it appear (to the perception algorithms) that it was another car (Miller et al., 2008).

Currently, a number of researchers exploring learning methods (e.g., Abbeel et al., 2010) have developed algorithms that learn helicopter dynamics/maneuver models over time from data provided by an expert pilot (Figure 3). Although learning seems straightforward to humans, it is difficult to implement algorithmically. New algorithms must be developed to ensure safe learning over time and adjust to new environments or uncertainties that have not been seen before (e.g., if at some point a car really did appear).

Verification and Validation in the Presence of Uncertainties

Current methods of validating software for autonomy in aerospace systems involve a series of expensive evaluation steps to heuristically develop confidence in the system. For example, UAV flight software typically requires validation first on a software simulator, then on a hardware-in-the-loop simulator, and then on flight tests. Fault-management systems continue to operate during flights, as required.

Researchers in formal logic, model checkers, and control theory have recently developed a set of tools that capture specification of tasks using more intuitive language/algorithms (Kress-Gazit et al., 2009; Wongpiromsarn et al., 2009). Con-

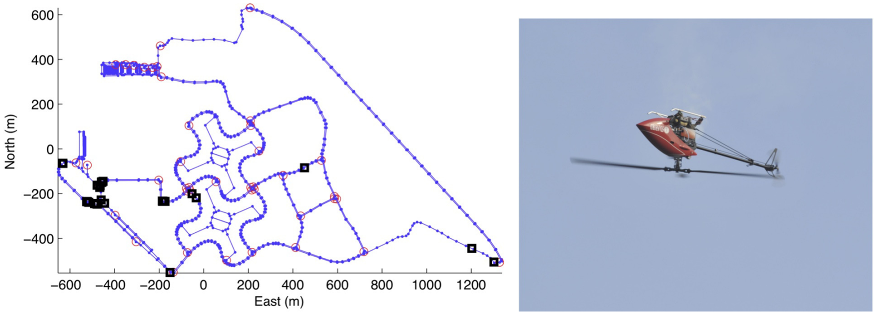


FIGURE 3 Left: Map of the DUC course (lines = map; circles = stop signs). Black squares indicate where brakes were applied quickly during the six-hour mission. Right: A model helicopter (operated by remote control or as an autonomous vehicle) in mid-maneuver. These complex maneuvers can be learned from an expert or by experimentation. Photo by E. Fratkin.

sider, for example, a car driving through an intersection with another car in the area (Figure 4). The rules of the road can be specified by logic, and controllers for autonomous driving can be automatically generated. Current research, however, typically addresses only simple models with little or no uncertainty.

New theory and methods will be necessary to incorporate uncertainties in perception, motion, and actions into a verifiable planning framework. Logic specifications must provide *probabilistic* guarantees of the high-level behavior of the robot, such as provably safe autonomous driving 99.9 percent of the time. These methods are also obviously important for aerospace systems, such as commercial airplanes and deep-space missions, where high costs and many lives are at risk.

INTERACTION BETWEEN HUMANS AND ROBOTS

Although interaction between humans and robots is of immense importance, it typically has a soft theoretical background. Human-robotic interaction, as it is typically called, includes a wide range of research. For example, tasks must be coordinated to take advantage of the strengths of both humans and robots; theory must scale well with larger teams; humans must not become overloaded or bored; and external influences, such as deciding if UAVs will have the ability to make actionable decisions or planetary rovers will be able to make scientific decisions, must be taken into consideration.

Efficient integration of autonomy with humans is essential for advanced aerospace systems. For example, a robot vacuuming a floor requires minimal interaction with humans, but search and tracking using a team of UAVs equipped with sensors and weapons is much more challenging, not only because of the complexity of the tasks and system, but also because of the inherent stress of the situation.

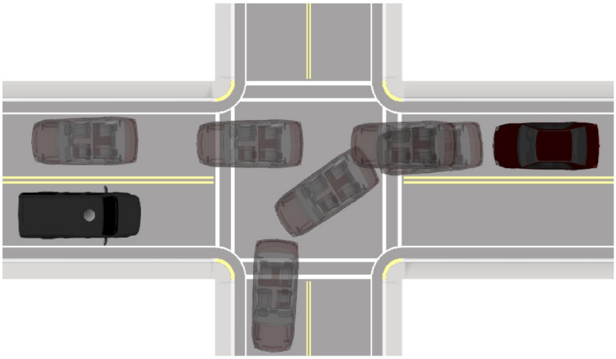


FIGURE 4 Example of using probabilistic anticipation for provably safe plans in autonomous driving.

The subject of interactions between humans and robots has many aspects and complexities. We look at three research topics, all of which may lead to more efficient and natural integration: (1) fusion of human and robotic information; (2) natural, robust, and high-performing interaction; and (3) scalable theory that enables easy adoption as well as formal analysis.

Fusion of Human and Robotic Information

Humans typically provide high-level commands to autonomous robots, but clearly they can also contribute important information, such as an opinion about which area of Mars to explore or whether a far off object in a cluttered environment is a person or a tree. Critical research is being conducted using machine-learning methods to formally model human opinions/decisions as sources of uncertain information and then fuse it with other information, such as information provided by the robot (Ahmed and Campbell, 2008, 2010).

Figure 5 shows a search experiment with five humans, each of whom has a satellite map overlaid with a density function that probabilistically captures the "location" of objects (Bourgault et al., 2008). The human sensor, in this case, is relatively simple: yes/no detection. A model of the human sensing process was developed by having the humans locate objects at various locations relative to their positions and look vectors. Intuitively, the ability to detect an object declines with increasing range and peripheral vision.

The fusion process, however, must also include uncertain measurements of the human's location and look vector. During the experiment, each human moved to a separate area, while fusing his/her own (uncertain) sensory information to create an updated density function for the location of objects. Fusion with information from other humans occurred when communication allowed—in this case,

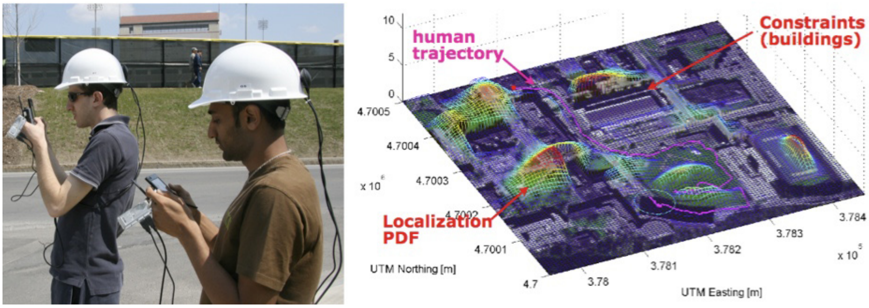


FIGURE 5 Search experiment with a network of five humans. Left: Humans with handheld PCs, local network, GPS, and compass. Right: Overlay of satellite imagery with a density of "probable" locations.

only at close range. Figure 5 shows the trajectory of one human's path and the real-time fused density of object location.

This experiment demonstrated initial decision modeling and fusion results, but the human decision was decidedly simple. To be useful, however, research, particularly in the area of machine learning, must model more complex outputs, such as strategic decisions over time or decisions made with little a priori data. New methods will be necessary to fuse information from many different sources, such as a human classifying items based on a discrete set of objects or pointing to a continuous area, or combinations of the two.

Natural, Robust, High-Performing Interaction

For effective teamwork by humans and robots, it is important to understand the strengths and weaknesses of both. Humans can provide critical strategic analyses but are subject to stress and fatigue, as well as boredom. In addition, they may have biases that must be taken into account (Parasuraman et al., 2000; Shah et al., 2009). Robots can perform repetitious tasks without bias or feelings. The strengths and weaknesses of both are, in effect, constraints in the design of systems in which humans and robots must work together seamlessly.

The most common interaction is through a computer, such as a mouse, keyboard, and screen. Sometimes, however, a human operator may be presented with more critical information for monitoring or decision making than he/she can handle. For example, the operator may be monitoring two UAVs during a search mission, and both may require command inputs at the same time. Or a human operator who becomes bored when monitoring video screens for hours at a time may not respond as quickly or effectively as necessary when action is required.

Taking advantage of recent commercial developments in computers that allow humans to interact with systems in many ways, current research is focused on multi-modal interaction. For example, Finomore et al. (2007) explored voice and chat inputs. Shah and Campbell (2010) are focusing on drawing commands on a tablet PC (Figure 6), where pixels are used to infer the "most probable" commands. The human operator can override a command if it is not correct, and the next most probable command will be suggested. Results of this study have shown a high statistical accuracy in recognizing the correct command by the human (Shah and Campbell, 2010).

More advanced systems are also being developed. Kress-Gazit and colleagues (2008) have developed a natural language parser that selects the appropriate command from spoken language and develops a provably correct controller for a robot. Boussemart and Cummings (2008) and Hoffman and Breazeal (2010) are working on modeling the human as a simplified, event-based decision maker; the robot then "anticipates" what the human wants to do and makes decisions appropriately. Although the latter approach is currently being applied only to simplified systems, it has the potential to improve team performance.

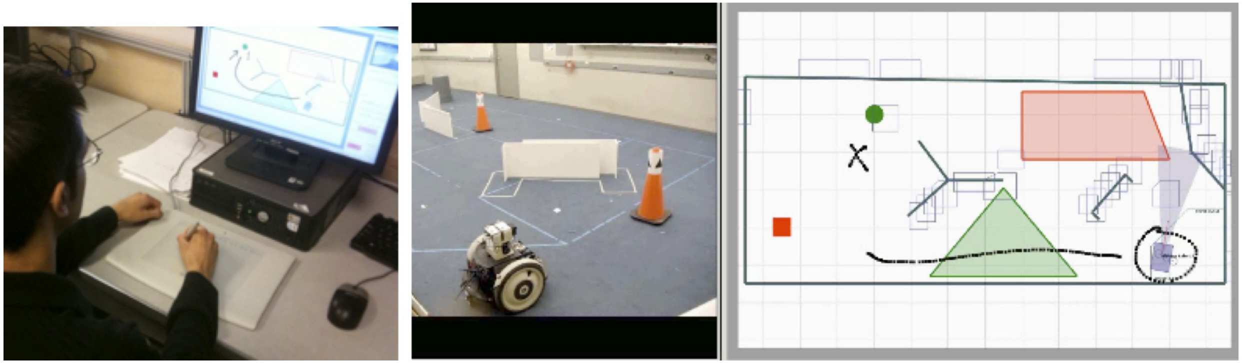


FIGURE 6 Human-drawn gesture commands for robots. Left: Human operator at a tablet-based computer. Center: A robot exploring an environment. Right: A screen shot showing how the human selected a robot, drew a potential path, and selected an area to explore.

Even non-traditional interfaces, such as commanding a UAV by brain waves, are being investigated (Akce et al., 2010).

Scalable Theory

A key constraint on the development of theory and implementations of teams of humans and robots is being able to scale up the theory to apply to large numbers (McLoughlin and Campbell, 2007; Sukkarieh et al., 2003). This is particularly important in defense applications, where hundreds, sometimes thousands of humans/vehicles must share information and plan together.

Most of the focus has been on hierarchical structures, but fully decentralized structures might also be effective (Ponda et al., 2010). Recent research has focused almost exclusively on large teams of cooperative vehicles, but given some level of human modeling, these methods could work for human/robot teams as well. The testing and adoption of these approaches, which will necessarily depend partly on cost and reliability, will continue to be challenging.

REFERENCES

Abbeel, P., A. Coates, and A.Y. Ng. 2010. Autonomous helicopter aerobatics through apprenticeship learning. International Journal of Robotics Research 29(7). doi: 10.1177/0278364910371999.

Ahmed, N., and M. Campbell. 2008. Multimodal Operator Decision Models. Pp. 4504–4509 in American Control Conference 2008. New York: IEEE.

Ahmed, N., and M. Campbell. 2010. Variational Bayesian Data Fusion of Multi-class Discrete Observations with Applications to Cooperative Human-Robot Estimation. Pp. 186–191 in 2010 IEEE International Conference on Robotics and Automation. New York: IEEE.

Akce, A., M. Johnson, and T. Bretl. 2010. Remote Tele-operation of an Unmanned Aircraft with a Brain-Machine Interace: Theory and Preliminary Results. Presentation at 2010 International Robotics and Automation Conference. Available online at *http://www.akce.name/presentations/ICRA2010.pdf*.

Blackmore, L., M. Ono, A. Bektassov, and B.C. Williams. 2010. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. IEEE Transactions on Robotics, 26(3): 502–517.

Bourgault, F., A. Chokshi, J. Wang, D. Shah, J. Schoenberg, R. Iyer, F. Cedano, and M. Campbell. 2008. Scalable Bayesian Human-Robot Cooperation in Mobile Sensor Networks. Pp. 2342–2349 in IEEE/RS International Conference on Intelligent Robots and Systems. New York: IEEE.

Boussemart, Y., and M.L. Cummings. 2008. Behavioral Recognition and Prediction of an Operator Supervising Multiple Heterogeneous Unmanned Vehicles. Presented at Humans Operating Unmanned Systems, Brest, France, September 3–4, 2008. Available online at *http://web.mit.edu/aeroastro/labs/halab/papers/boussemart%20-%20humous08-v1.3-final.pdf*.

Campbell, M.E., and W.W. Whitacre. 2007. Cooperative tracking using vision measurements on SeaScan UAVs. IEEE Transactions on Control Systems Technology 15(4): 613–626.

Cutting, J.E. 2003. Reconceiving Perceptual Space. Pp. 215–238 in Perceiving Pictures: An Interdisciplinary Approach to Pictorial Space, edited by H. Hecht, R. Schwartz, and M. Atherton. Cambridge, Mass.: MIT Press.

Diebel, J., and S. Thrun. 2006. An application of Markov random fields to range sensing. Advances in Neural Information Processing Systems 18: 291.

Finomore, V.S., B.A. Knott, W.T. Nelson, S.M. Galster, and R.S. Bolia. 2007. The Effects of Multimodal Collaboration Technology on Subjective Workload Profiles of Tactical Air Battle Management Teams. Technical report #AFRL-HE-WP-TP-2007-0012. Wright Patterson Air Force Base, Defense Technical Information Center.

Grocholsky, B., A. Makarenko, and H. Durrant-Whyte. 2004. Information-Theoretic Coordinated Control of Multiple Sensor Platforms. Pp. 1521–1526 in Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 1. New York: IEEE.

Hardy, J., and M. Campbell. 2010. Contingency Planning over Hybrid Obstacle Predictions for Autonomous Road Vehicles. Presented at IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, October 18–22, 2010.

Havlak, F., and M. Campbell. 2010. Discrete and Continuous, Probabilistic Anticipation for Autonomous Robots in Urban Environments. Presented at SPIE Europe Conference on Unmanned/Unattended Sensors and Sensor Networks, Toulouse, France, September 20–23, 2010.

Hoffman, G., and C, Breazeal. 2010. Effects of anticipatory perceptual simulation on practiced human-robot tasks. Autonomous Robots 28: 403–423.

Iagnemma, K., M. Buehler, and S. Singh, eds. 2008. Special Issue on the 2007 DARPA Urban Challenge, Part 1. Journal of Field Robotics 25(8): 423–860.

Kress-Gazit, H., G.E. Fainekos, and G.J. Pappas. 2008. Translating structured English to robot controllers. Advanced Robotics: Special Issue on Selected Papers from IROS 2007 22(12): 1343–1359.

Kress-Gazit, H., G.E. Fainekos, and G.J. Pappas. 2009. Temporal logic-based reactive mission and motion planning. IEEE Transactions on Robotics 25(6): 1370–1381.

McClelland, M., and M. Campbell. 2010. Anticipation as a Method for Overcoming Time Delay in Control of Remote Systems. Presented at the AIAA Guidance, Navigation and Control Conference, Toronto, Ontario, Canada, August 2–5, 2010. Reston, Va.: AIAA.

McLoughlin, T., and M. Campbell. 2007. Scalable GNC architecture and sensor scheduling for large spacecraft networks. AIAA Journal of Guidance, Control, and Dynamics 30(2): 289–300.

Miller, I., M. Campbell, D. Huttenlocher, A. Nathan, F.-R. Kline, P. Moran, N. Zych, B. Schimpf, S. Lupashin, E. Garcia, J. Catlin, M. Kurdziel, and H. Fujishima. 2008. Team Cornell's Skynet: robust perception and planning in an urban environment. Journal of Field Robotics 25(8): 493–527.

Ousingsawat, J., and M.E. Campbell. 2007. Optimal cooperative reconnaissance using multiple vehicles. AIAA Journal of Guidance Control and Dynamics 30(1): 122–132.

Parasuraman, R., T.B. Sheridan, and C.D. Wickens. 2000. A model for types and levels of human interaction with automation. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 30: 286–297.

Ponda, S., H.L. Choi, and J.P. How. 2010. Predictive Planning for Heterogeneous Human-Robot Teams. Presented at AIAA Infotech@Aerospace Conference 2010, Atlanta, Georgia, April 20–22, 2010.

Ross, M., and A. Oliva. 2010. Estimating perception of scene layout. Journal of Vision 10: 1–25.

Schoenberg, J., A. Nathan, and M. Campbell. 2010. Segmentation of Dense Range Information in Complex Urban Scenes. Presented at IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, October 18–22, 2010.

Shah, D., and M. Campbell. 2010. A Robust Sketch Interface for Natural Robot Control. Presented at IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, October 18–22, 2010.

Shah, D., M. Campbell, F. Bourgault, and N. Ahmed. 2009. An Empirical Study of Human-Robotic Teams with Three Levels of Autonomy. Presented at AIAA Infotech@Aerospace Conference 2009, Seattle, Washington, April 7, 2009.

Sukkarieh, S., E. Nettleton, J.H. Kim, M. Ridley, A. Goktogan, and H. Durrant-Whyte. 2003. The ANSER Project: data fusion across multiple uninhabited air vehicles. The International Journal of Robotics Research 22(7–8): 505.

Thrun, S., W. Burgard, and D. Fox. 2005. Probabilistic Robotics. Cambridge, Mass.: MIT Press.

Wongpiromsarn, T., U. Topcu, and R.M. Murray. 2009. Receding Horizon Temporal Logic Planning for Dynamical Systems. Available online at *http://www.cds.caltech.edu/~utopcu/images/1/10/WTM-cdc09.pdf*.

Challenges and Opportunities for Autonomous Systems in Space

Chad R. Frost
NASA Ames Research Center

With the launch of Deep Space 1 in 1998, the autonomous systems community celebrated a milestone—the first flight experiment demonstrating the feasibility of a fully autonomous spacecraft. We anticipated that the advanced autonomy demonstrated on Deep Space 1 would soon be pervasive, enabling science missions, making spacecraft more resilient, and reducing operational costs.

However, the pace of adoption has been relatively slow. When autonomous systems have been used, either operationally or as an experiment or demonstration, they have been successful. In addition, outstanding work by the autonomous-systems community has continued to advance the technologies of autonomous systems (Castaño et al., 2006; Chien et al., 2005; Estlin et al., 2008; Fong et al., 2008; Knight, 2008).

There are many reasons for putting autonomous systems on board spacecraft. These include maintenance of the spacecraft despite failures or damage, extension of the science team through "virtual presence," and cost-effective operation over long periods of time. Why, then, has the goal of autonomy not been more broadly adopted?

DEFINITION OF AUTONOMY

First, we should clarify the difference between *autonomy* and *automation*. Many definitions are possible (e.g., Doyle, 2002), but here we focus on the need to make choices, a common requirement for systems outside our direct, hands-on control.

An *automated system* doesn't make choices for itself—it follows a script, albeit a potentially sophisticated script, in which all possible courses of action have already been made. If the system encounters an unplanned-for situation, it stops and waits for human help (e.g. it "phones home"). Thus, for an automated system choices have either already been made and encoded, or they must be made externally.

By contrast, an *autonomous system* does make choices on its own. It tries to accomplish its objectives locally, without human intervention, even when encountering uncertainty or unanticipated events.

An ***intelligent*** *autonomous system* makes choices using more sophisticated mechanisms than other systems. These mechanisms often resemble those used by humans. Ultimately, the level of intelligence of an autonomous system is judged by the quality of the choices it makes. Regardless of the implementation details, however, intelligent autonomous systems are capable of more "creative" solutions to ambiguous problems than are systems with simpler autonomy, or automated systems, which can only handle problems that have been foreseen.

A system's behavior may be represented by more than one of these descriptions. A domestic example of such a system, iRobot's Roomba™ line of robotic vacuum cleaners, illustrates how prosaic automation and autonomy have become. The Roomba must navigate a house full of obstacles while ensuring that the carpet is cleaned—a challenging task for a consumer product. We can evaluate the Roomba's characteristics using the definitions given above:

- The Roomba user provides high-level goals (vacuum the floor, but don't vacuum here, vacuum at this time of day, etc.)
- The Roomba must make some choices itself (how to identify the room geometry, avoid obstacles, when to recharge its battery, etc).
- The Roomba also has some automated behavior and encounters situations it cannot resolve on its own (e.g., it gets stuck, it can't clean its own brushes, etc.).

Overall, the Roomba has marginal autonomy, and there are numerous situations it cannot deal with by itself. It is certainly not intelligent. However, it does have basic on-board diagnostic capability ("clean my brushes!") and a strategy (as seen in Figure 1) for vacuuming a room about whose size and layout it was initially ignorant. Roomba demonstrates the potential for the widespread use of autonomous systems in our daily lives.

AUTONOMY FOR SPACE MISSIONS

So much has been written on this topic that we can barely scratch the surface of a deep and rewarding discussion in this short article. However, we can examine a few recurring themes. The needs for autonomous systems depend, of course, on the mission. Autonomous operation of the spacecraft, its subsystems, and the science instruments or payload become increasingly important as the spacecraft

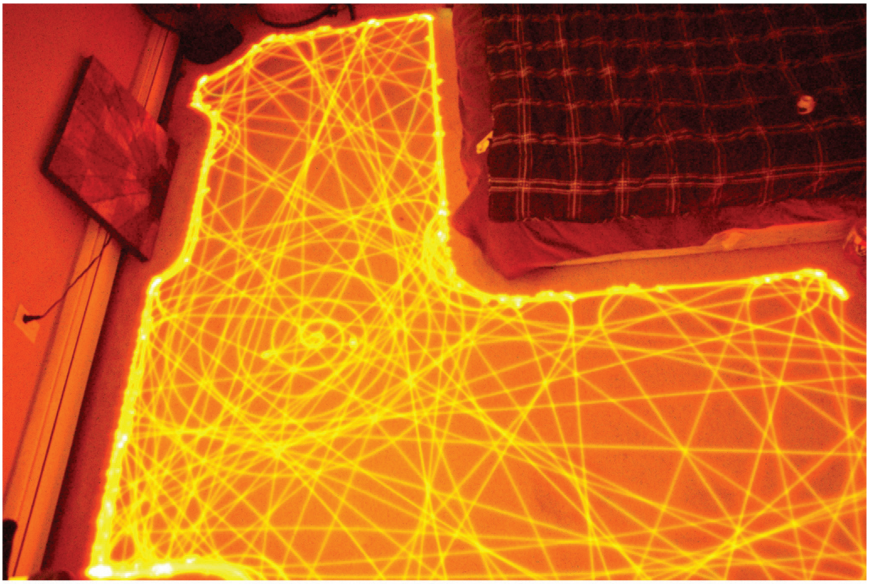


FIGURE 1 Long-exposure image of Roomba's path while navigating a room. Photo by Paul Chavady, used with permission.

is required to deal with phenomena that occur on time scales shorter than the communication latency between the spacecraft and Earth. But all spacecraft must maintain function "to ensure that hardware and software are performing within desired parameters, and [find] the cause of faults when they occur" (Post and Rose, undated).

Virtual Presence

"Virtual presence" is often cited as a compelling need for autonomy. Scientific investigation, including data analysis and discovery, becomes more challenging the farther away the scientific instruments are from the locus of control. Doyle (2002) suggests that ". . . a portion of the scientist's awareness will begin to move onboard, i.e., an observing and discovery presence. Knowledge for discriminating and determining what information is important would begin to migrate to the space platform." Marvin Minsky (1980), a pioneer of artificial intelligence, made the following observation about the first lunar landing mission in 1969:

> With a lunar telepresence vehicle making short traverses of one kilometer per day, we could have surveyed a substantial area of the lunar surface in the ten years that have slipped by since we landed there."

Add another three decades, and Minsky's observation takes on even greater relevance.

Traversing extraterrestrial sites is just one example of the ongoing "dirty, dull, dangerous" work that it is not cost-effective, practical, or safe for humans to do. Yet these tasks have great potential for long-term benefits. As Minsky pointed out, even if the pace of investigation or work is slower than what humans *in situ* might accomplish, the cumulative effort can still be impressive. The Mars Exploration Rovers are a fine example. In the six years since they landed, they have jointly traversed more than 18 miles and collected hundreds of thousands of images.

Thus, autonomous systems can reduce the astronauts' workload on crewed missions. An astronaut's limited, valuable time should not be spent verifying parameters and performing spacecraft housekeeping, navigation, and other chores that can readily be accomplished by autonomous systems.

Another aspect of virtual presence is robotic mission enablers, such as the extension of the spacecraft operator's knowledge onto the spacecraft, enabling "greater onboard adaptability in responding to events, closed-loop control for small body rendezvous and landing missions, and operation of the multiple free-flying elements of space-based telescopes and interferometers" (Doyle, 2002). Several of these have been demonstrated, such as the Livingstone 2, which provided on-board diagnostics on Earth Orbiter 1 (EO1) (Hayden et al., 2004) and full autonomy, including docking and servicing with Orbital Express (Ogilvie et al., 2008).

Common Elements of Autonomous Systems

The needs autonomous systems can fulfill can be distilled down to a few common underlying themes: mitigating latency (the distance from those interested in what the system is doing); improving efficiency by reducing cost and mass and improving the use of instrument and/or crew time; and managing complexity by helping to manage spacecraft systems that have become so complex they are difficult, sometime impossible, for humans on-board or on the ground to diagnose and solve problems. Nothing is flown on a spacecraft that has not "paid" for itself, and this holds true for the software that gives it autonomy.

SUCCESS STORIES

Where do we stand today in terms of *deployed* autonomous systems? Specifically, which of the technology elements that comprise a complex, self-sufficient, intelligent system have flown in space? There have been several notable successes. Four milestone examples illustrate the progress made: Deep Space 1, which flew the first operational autonomy in space; Earth Observing 1, which demonstrated autonomous collection of science data; Orbital Express, which autonomously carried out spacecraft servicing tasks; and the Mars Exploration Rovers, which have been long-term hosts for incremental improvements in their autonomous capabilities.

Deep Space 1. A Remote Agent Experiment

In 1996, NASA specified an autonomous mission scenario called the New Millennium Autonomy Architecture Prototype (NewMAAP). For that mission, a Remote Agent architecture that integrated constraint-based planning and scheduling, robust multi-threaded execution, and model-based mode identification and reconfiguration was developed to meet the NewMAAP requirements (Muscettola et al., 1998; Pell et al., 1996). This architecture was described by Muscettola in 1998:

> The Remote Agent architecture has three distinctive features: First, it is largely programmable through a set of compositional, declarative models. We refer to this as model-based programming. Second, it performs significant amounts of on-board deduction and search at time resolutions varying from hours to hundreds of milliseconds. Third, the Remote Agent is designed to provide high-level closed-loop commanding.

Based on the success of NewMAAP as demonstrated in the Remote Agent, it was selected as a technology experiment on the Deep Space 1 (DS1) mission. Launched in 1998, the goal of DS1 (Figure 2) was to test 12 cutting-edge technologies, including the Remote Agent Experiment (RAX), which became the first operational use of "artificial intelligence" in space. DS1 functioned completely autonomously for 29 hours, successfully operating the spacecraft and responding to both simulated and real failures.

Earth Observing 1. An Autonomous Sciencecraft Experiment

Earth Observing 1 (EO1), launched in 2000, demonstrated on-board diagnostics and autonomous acquisition and processing of science data, specifically, imagery of dynamic natural phenomena that evolve over relatively short time spans (e.g., volcanic eruptions, flooding, ice breakup, and changes in cloud cover).

The EO1 Autonomous Sciencecraft Experiment (ASE) included autonomous on-board fault diagnosis and recovery (Livingstone 2), as well as considerable autonomy of the science instruments and downlink of the resulting imagery and data. Following this demonstration, ASE was adopted for operational use. It has been in operation since 2003 (Chien et al., 2005, 2006).

A signal success of the EO1 mission was the independent capture of volcanic activity on Mt. Erebus. In 2004, ASE detected an anomalous heat signature, scheduled a new observation, and effectively detected the eruption by itself. Figure 3 shows 2006 EO1 images of the volcano.

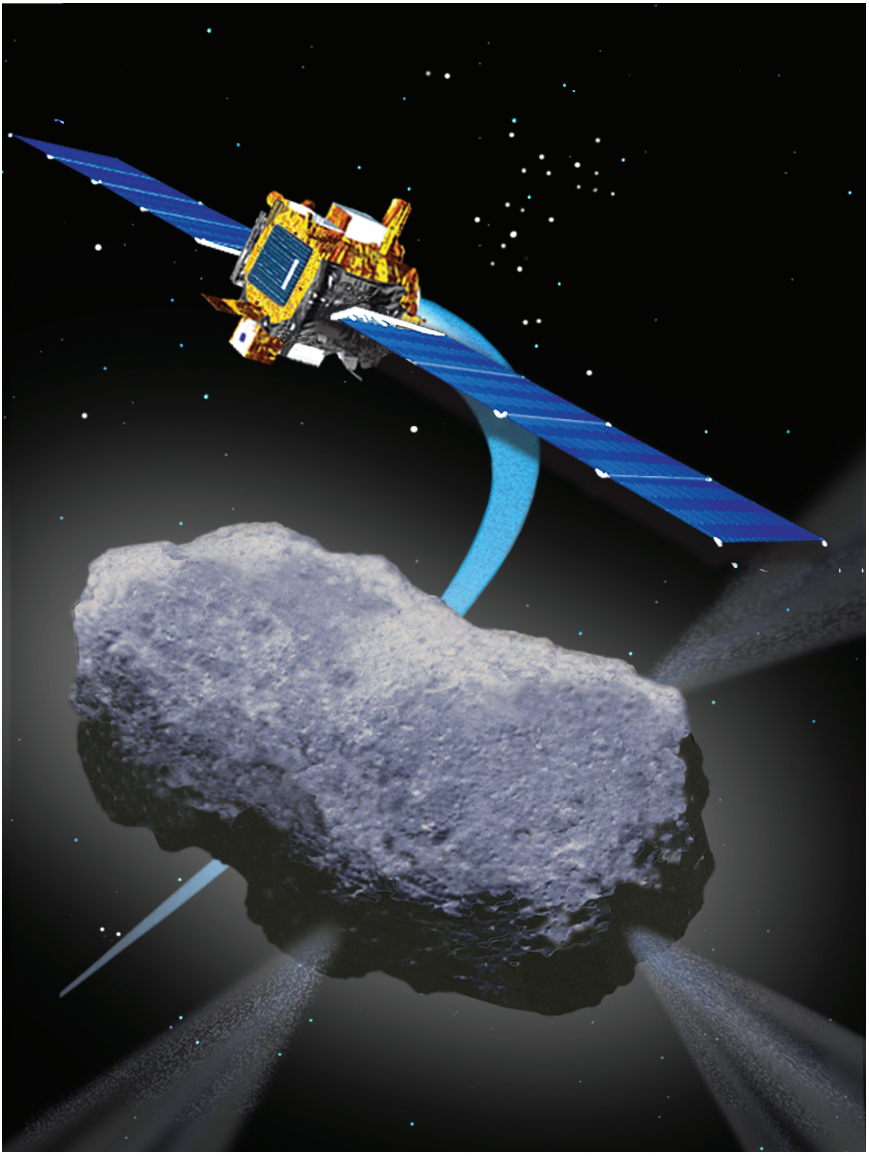


FIGURE 2 Deep Space 1 flew the Remote Agent Experiment, which demonstrated full spacecraft autonomy for the first time. Figure courtesy of NASA/JPL-Caltech.

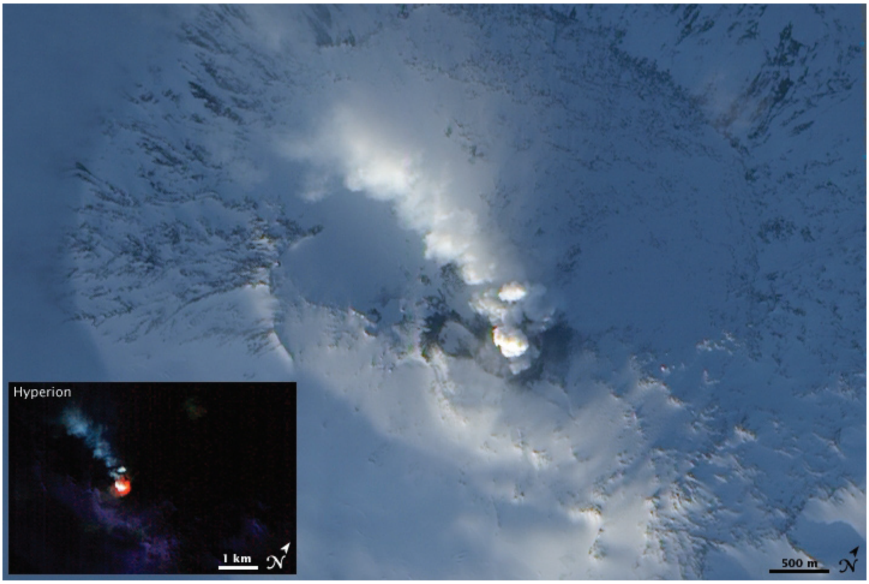


FIGURE 3 Images of volcanic Mt. Erebus, autonomously collected by EO1. NASA image created by Jesse Allen, using EO-1 ALI data provided courtesy of the NASA EO1 Team.

Orbital Express

In 2007, the Orbital Express mission launched two complimentary spacecraft, ASTRO and NextSat, with the goal of demonstrating a complete suite of the technologies required to autonomously service satellites on-orbit. The mission demonstrated several levels of on-board autonomy, ranging from mostly ground-supervised operations to fully autonomous capture and servicing, self-directed transfer of propellant, and automatic capture of another spacecraft using a robotic arm (Boeing Integrated Defense Systems, 2006; Ogilvie et al., 2008). These successful demonstrations showed that servicing and other complex spacecraft operations can be conducted autonomously.

Mars Exploration Rovers

Since landing on Mars in 2004, the Mars Exploration Rovers, *Spirit* and *Opportunity*, have operated with increasing levels of autonomy. An early enhancement provided autonomous routing around obstacles; another automated the process of calculating how far the rover's arm should reach out to touch a particular rock.

In 2007, the rovers were updated to autonomously examine sets of sky images, determine which ones showed interesting clouds or dust devils, and send only those images back to scientists on Earth. The most recent software has enabled *Opportunity* to make decisions about acquiring new observations, such as selecting rocks on the basis of shape and color, for imaging by the wide-angle navigation camera and detailed inspection with the narrow-field panoramic camera (Estlin et al., 2008).

REMAINING CHALLENGES

Despite the compelling need for spacecraft autonomy and the feasibility demonstrated by the successful missions described above, obstacles remain to the use of autonomous systems as regular elements of spacecraft flight software. Two kinds of requirements for spacecraft autonomy must be satisfied: (1) functional requirements, which represent attributes the software must objectively satisfy for it to be acceptable; and (2) perceived requirements, which are not all grounded in real mission requirements but weigh heavily in subjective evaluations of autonomous systems. Both types of requirements must be satisfied for the widespread use of autonomous systems.

Functional Requirements

From our experience thus far, we have a good sense of the overarching functional requirements for space mission autonomy. Muscettola et al. (1998) offer a nicely distilled set of these requirements (evolved from Pell et al., 1996).

“First, a spacecraft must carry out autonomous operations for long periods of time with no human intervention.” Otherwise, what’s the point of including autonomous systems? Short-term autonomy may be even worse than no autonomy at all. If humans have to step in to a nominally autonomous process, they are likely to spend a lot of time trying to determine the state of the spacecraft, how it got that way, and what needs to be done about it.

“Second, autonomous operations must guarantee success given tight deadlines and resource constraints.” By definition, if an unplanned circumstance arises, an autonomous system cannot stop and wait indefinitely either for human help or to deliberate on a course of action. The system *must* act, and it must do so expediently. Whether autonomy can truly guarantee success is debatable, but at least it should provide the highest likelihood of success.

“Third, since spacecraft are expensive and are often designed for unique missions, spacecraft operations require high reliability.” Even in the case of a (relatively) low-cost mission and an explicit acceptance of a higher level of risk, NASA tends to be quite risk-averse! Failure is perceived as undesirable, embarrassing, “not an option,” even when there has been a trade-off between an increased chance of failure and a reduction in cost or schedule. Autonomy must be perceived as reducing risk, ideally, without significantly increasing cost or schedule. Program managers, science principal investigators, and spacecraft engineers want (and need) an answer to their frequently asked question, "How can we be sure that your software will work as advertised and avoid unintended behavior?"

"Fourth, spacecraft operation involves concurrent activity by tightly coupled subsystems." Thus, requirements and interfaces must be thoroughly established relatively early in the design process, which pushes software development forward in the program and changes the cost profile of the mission.

Perceived Requirements

Opinions and perceptions, whether objectively based or not, are significant challenges to flying autonomous systems on spacecraft. Some of the key requirements and associated issues are identified below.

Reliability

As noted above, the question most frequently asked is whether autonomy software will increase the potential risk of a mission (Frank, 2008b). The question really being asked is whether the autonomous systems have the ability to deal with potentially loss-of-mission failures sufficiently to offset the added potential for software problems.

Complexity

Bringing autonomous systems onto a spacecraft is perceived as adding complexity to what might otherwise be a fairly straightforward system. Certainly autonomy increases the complexity of a simple, bare-bones system. This question is more nuanced if it is about degrees of autonomy, or autonomy versus automation. As spacecraft systems themselves become more complex, or as we ask them to operate more independently of us, must the software increase in sophistication to match? And, does sophistication equal complexity?

Cost

Adding many lines of code to support autonomous functions is perceived as driving up the costs of a mission. The primary way an autonomous system "buys" its way onto a spacecraft is by having the unique ability to enable or save the mission. In addition, autonomous systems may result in long-term savings by reducing operational costs. However, in both cases, the benefits may be much more difficult to quantify than the cost, thus, making the cost of deploying autonomous systems highly subjective.

Sci-Fi Views of Autonomy

This perceptive requirement boils down to managing expectations and educating people outside the intelligent-systems community, where autonomous systems are sometimes perceived as either overly capable, borderline Turing machines, or latent HAL-9000s ready to run amok (or worse, to suffer from a subtle, hard-to-diagnose form of mental illness). Sorry, but we're just not there yet!

ADDRESSING THE CHALLENGES

The principal challenge to the deployment of autonomous systems is risk-reduction (real or perceived). Improvements can be achieved in several ways.

Processes

Perhaps the greatest potential contributor to the regular use of autonomous systems on spacecraft is to ensure that rigorous processes are in place to (1) thoroughly verify and validate the software, and (2) minimize the need to develop new software for every mission. Model-based software can help address both of these key issues (as well as others) as this approach facilitates rigorous validation of the component models and the re-use of knowledge.

The model-based software approach was used for the Remote Agent Experiment (Williams and Nayak, 1996) and for the Livingstone 2 diagnostic engine used aboard EO1 (Hayden et al., 2004). However, processes do not emerge fully formed. Only through the experience of actually flying autonomous systems (encompassing both successes and failures) can we learn about our processes and the effectiveness of our methods.

Demonstrations

The next most effective way to reduce risk is to increase the flight experience and heritage of autonomous software components. This requires a methodical approach to including autonomous systems on numerous missions, initially as "ride-along" secondary or tertiary mission objectives, but eventually on missions dedicated to experiments of autonomy.

This is not a new idea. NASA launched DS1 and EO1 (and three other spacecraft) under the auspices of the New Millennium Program, which was initiated in 1995 with the objective of validating a slate of technologies for space applications. However, today there is no long-term strategy in place to continue the development and validation of spacecraft autonomy.

Fundamental Research

We have a long way to go before autonomous systems will do all that we hope they can do. Considerable research will be necessary to identify new, creative solutions to address the many challenges that remain. For example, it can be a challenge to build an integrated system of software to conduct the many facets of autonomous operations (including, e.g., planning, scheduling, execution, fault detection, and fault recovery) within the constraints of spaceflight computing hardware. Running on modern terrestrial computers, let alone much slower flight-qualified hardware, algorithms to solve the hard problems in these disciplines may be intractably slow.

Academic experts often know how to create algorithms that can theoretically run fast enough, but their expertise must be transformed into engineering discipline—practical, robust software suitable for use in a rigorous real-time environment—and integrated with the many other software elements that must simultaneously function.

Education

As we address the issues listed above, we must simultaneously educate principal investigators, project managers, and the science community about the advantages of autonomous systems and their true costs and savings.

OPPORTUNITIES

There are many upcoming missions in which autonomy could play a major role. Although so far, human spaceflight has been remarkably devoid of autonomy, as technologies are validated in unmanned spacecraft and reach levels of maturity commensurate with other human-rated systems, there is great potential for autonomous systems to assist crews in maintaining and operating even the most complex spacecraft over long periods of time. Life support, power, communications, and other systems require automation, but would also benefit from autonomy (Frank, 2008a).

Missions to the planets of our solar system and their satellites will increasingly require the greatest possible productivity and scientific return on the large investments already made in the development and launch of these sophisticated spacecraft. Particular destinations, such as the seas of Europa, will place great demands on autonomous systems, which will have to conduct independent explorations in environments where communication with Earth is difficult or impossible. Proposed missions to near-Earth objects (NEOs) will entail autonomous rendezvous and proximity operations, and possibly contact with or sample retrieval from the object.

A variety of Earth and space science instruments can potentially be made autonomous, in whole or in part, independently of whether the host spacecraft has any operational autonomy. Autonomous drilling equipment, hyperspectral imagers, and rock samplers have all been developed and demonstrated terrestrially in Mars-analog environments. EO1 and the Mars Exploration Rovers were (and are) fine examples of autonomous science systems that have improved our ability to respond immediately to transient phenomena.

This is hardly an exhaustive list of the possibilities, but it represents the broad spectrum of opportunities for autonomous support for space missions. Numerous other missions, in space, on Earth, and under the seas could also be enhanced or even realized by the careful application of autonomous systems.

CONCLUSIONS

Looking back, we have had some great successes, and looking ahead, we have some great opportunities. But it has been more than a decade since the first autonomous systems were launched into space, and operational autonomy is not yet a standard practice. So what would enable the adoption of autonomy by more missions?

- An infrastructure (development, testing, and validation processes) and code base in place, so that each new mission does not have to re-invent the wheel and will only bear a marginal increase in cost;
- A track record (“flight heritage”) establishing reliability; and
- More widespread understanding of the benefits of autonomous systems.

To build an infrastructure and develop a “flight heritage,” we will have to invest. Earth-based rovers, submarines, aircraft, and other “spacecraft analogs” can serve (and frequently do serve) as lower cost, lower risk validation platforms. Such developmental activities should lead to several flights of small spacecraft that incrementally advance capabilities as they add to the flight heritage and experience of the technology and the team.

Encouraging and developing understanding will depend largely on technologists listening to those who might someday use their technologies. Unless we hear and address the concerns of scientists and mission-managers, we will not be able to explain how autonomous systems can further scientific goals. If the autonomous-systems community can successfully do these things, we as a society stand to enter an exciting new period of human and robotic discovery.

REFERENCES

Boeing Integrated Defense Systems. 2006. Orbital Express Demonstration System Overview. Technical Report. The Boeing Company, 2006. Available online at *http://www.boeing.com/bds/phantom_works/orbital/mission_book.html*.

Castaño, R., T. Estlin, D. Gaines, A. Castaño, C. Chouinard, B. Bornstein, R. Anderson, S. Chien, A. Fukunaga, and M. Judd. 2006. Opportunistic rover science: finding and reacting to rocks, clouds and dust devils. In *IEEE Aerospace Conference*, *2006*. 16 pp.

Chien, S., R. Sherwood, D. Tran, B. Cichy, G. Rabideau, R. Castano, A. Davies, D. Mandl, S. Frye, B. Trout, S. Shulman, and D. Boyer. 2005. Using autonomy flight software to improve science return on Earth Observing One. Journal of Aerospace Computing, Information, and Communication 2(4): 196–216.

Chien, S., R. Doyle, A. Davies, A. Jónsson, and R. Lorenz. 2006. The future of AI in space. IEEE Intelligent Systems 21(4): 64–69, July/August 2006. Available online at *www.computer.org/intelligent*.

Doyle, R.J. 2002. Autonomy needs and trends in deep space exploration. In RTO-EN-022 "Intelligent Systems for Aeronautics," Applied Vehicle Technology Course. Neuilly-sur-Seine, France: NATO Research and Technology Organization. May 2002.

Estlin, T., R. Castano, D. Gaines, B. Bornstein, M. Judd, R.C. Anderson, and I. Nesnas. 2008. Supporting Increased Autonomy for a Mars Rover. In 9th International Symposium on Artificial Intelligence, Robotics and Space*,* Los Angeles, Calif., February 2008. Pasadena, Calif.: Jet Propulsion Laboratory.

Fong, T., M. Allan, X. Bouyssounouse, M.G. Bualat, M.C. Deans, L. Edwards, L. Flückiger, L. Keely, S.Y. Lee, D. Lees, V. To, and H. Utz. 2008. Robotic Site Survey at Haughton Crater. In Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space*,* Los Angeles, Calif., 2008.

Frank, J. 2008a. Automation for operations. In Proceedings of the AIAA Space Conference and Exposition*.* Ames Research Center.

Frank, J. 2008b. Cost Benefits of Automation for Surface Operations: Preliminary Results. In Proceedings of the AIAA Space Conference and Exposition. Ames Research Center.

Hayden, S.C., A.J. Sweet, S.E. Christa, D. Tran, and S. Shulman. 2004. Advanced Diagnostic System on Earth Observing One. In Proceedings of AIAA Space Conference and Exhibit, San Diego, California, Sep. 28–30, 2004.

Knight, R. 2008. Automated Planning and Scheduling for Orbital Express. In 9th International Symposium on Artificial Intelligence, Robotics, and Automation in Space*,* Los Angeles, Calif., February 2008. Pasadena, Calif.: Jet Propulsion Laboratory.

Minsky, M. 1980. Telepresence. OMNI magazine*,* June 1980. Available online at *http://web.media.mit.edu/~minsky/papers/Telepresence.html*.

Muscettola, N., P.P. Nayak, B. Pell, and B.C. Williams. 1998. Remote agent: to boldly go where no AI system has gone before. Artificial Intelligence 103(1–2): 5–47. Available online at: *http://dx.doi.org/10.1016/S0004-3702(98)00068-X*.

Ogilvie, A., J. Allport, M. Hannah, and J. Lymer. 2008. Autonomous Satellite Servicing Using the Orbital Express Demonstration Manipulator System. Pp. 25–29 in Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS'08*).*

Pell, B., D.E. Bernard, S.A. Chien, E. Gat, N. Muscettola, P.P. Nayak, M.D. Wagner, and B.C. Williams. 1996. A Remote Agent Prototype for Spacecraft Autonomy. In SPIE Proceedings, Vol. 2810, Space Sciencecraft Control and Tracking in the New Millennium*,* Denver, Colorado, August 1996.

Post, J.V., and D.D. Rose. Undated. A.I. In Space: Past, Present & Possible Futures. Available online at *http://www.magicdragon.com/ComputerFutures/SpacePublications/AI_in_Space.html*.

Williams, B.C., and P.P. Nayak. 1996. A Model-Based Approach to Reactive Self-Configuring Systems. Pp. 971–978 in Proceedings of the National Conference on Artificial Intelligence. Menlo Park, Calif.: Association for the Advancement of Artificial Intelligence.

Health Awareness in Systems of Multiple Autonomous Aerospace Vehicles

Stefan Bieniawski
Boeing Research & Technology

Significant investments have been made in the development of off-line systems for monitoring and predicting the condition and capabilities of aerospace systems, usually for the purpose of reducing operational costs. A recent trend, however, has been to include these technologies online and use the information they provide for real-time autonomous or semi-autonomous decision making. Health-based adaptations are common in systems that control critical functions, such as redundant flight control systems, but as the scope of systems has expanded—for instance, to systems with multiple vehicles—new challenges and opportunities continue to arise.

Recent studies have explored health-based adaptations at all levels (subsystem, system, and systems-of-systems layers) of a heterogeneous, multi-vehicle system. This emphasis on health awareness has the potential to address two needs: (1) to improve safety, overall system performance, and reliability; and (2) to meeting the expectations (situational awareness, override capability, and task or mission definition) of human operators, who are inevitably present.

One approach to evaluating complex, multi-vehicle systems is to use a subscale indoor flight-test facility where common real faults are manifested in different forms. This type of facility can handle a great many flight hours at low cost for a wide range of vehicle types and component technologies. The lessons learned from these tests and from the architecture developed to complete them are relevant for a large variety of aerospace systems.

This paper begins with a brief review of health awareness in aerospace vehicles and highlights of recent research. Key challenges are then discussed followed by a description of the integrated, experiment-based approach mentioned above. The paper concludes with a summary of lessons learned and opportunities for further research.

BACKGROUND

"Health management" in the context of aerospace systems can be defined as the use of measured data and supporting diagnostic and prognostic algorithms to determine the condition and predict the capability of systems and subsystems. The "condition," which provides insight into the current state of the system or subsystem, is primarily determined by diagnostic algorithms. As a notional example, determining condition might include measuring the voltage of a battery (diagnosis) and assessing its state (fully charged, partially charged, discharged). Determining "capability," which requires more sophisticated prognostic algorithms, might involve estimating the amount of charge or remaining time at a selected load level (prognosis). A more advanced capability would be an estimate of the number of remaining charge/discharge cycles.

Diagnostic algorithms are now commonly used in commercial and military aircraft and are a basic tool for many maintenance services. These technologies, which minimize the time aircraft must be out of service for maintenance, have tremendous value. Based on the extensive measurement suites available on existing aircraft, analyses are typically used for binary decision making (e.g., continue to use or replace). Although in some cases the information is downlinked in near real time, analyses are generally performed off-line at regular intervals.

Online diagnostic algorithms have only been used in limited situations for critical applications; these include real-time sensor integrity algorithms for managing redundancy in multichannel, fly-by-wire, flight control systems. Despite the limited use of these algorithms, their successes to date have illustrated the potential for health-based algorithms and decision making.

Ongoing research is being done on expanding the application of health-based diagnostic and "longer viewing" prognostic algorithms, which could significantly improve real-time decision making. Recent research has focused on how these technologies might be used in real time to augment decision making by autonomous systems and systems-of-systems. The research is divided into several categories: (1) sensors for providing raw data for algorithms; (2) diagnostic and prognostic algorithms for mining data and providing actionable condition and capability information; and (3) algorithms for using condition and capability data to make decisions. The resulting health-based adaptation can be made in various layers in a large-scale system or system-of-systems, ranging from subsystems (e.g., primary flight control or power management) to systems (e.g., individual vehicles) to systems-of-systems (e.g., multi-vehicle mission management).

CHALLENGES

Researchers have focused on identifying and addressing several key challenges. These include system complexity, the development of a system architecture, and a suitable evaluation environment.

System Complexity

In the large-scale systems of interest, there are subtle interactions between subsystems, systems, prototype algorithms, and the external environment. These interactions can lead to emergent behavior that makes it difficult to understand the contributions of various algorithms to overall system performance.

For instance, consider the effect of a new algorithm for ensuring the safe separation of aerial vehicles. How does this algorithm perform in the context of a large air traffic network in the presence of faults in various components and communication links? And how might these same technology elements be applicable to alternate missions such as search and rescue missions? A related issue is the development of suitable high-level system missions and associated metrics for the quantifiable evaluation of performance.

Development of an Adequate Architecture

A second challenge is to develop a system architecture that provides a framework for guiding and maturing technology components. Much of the development of existing algorithms is performed in isolation. Thus, even though it is based on excellent theoretical results, the consideration of peripheral effects in the complete system may be limited. An effective system architecture must be modular to allow the various technology elements to be implemented and evaluated in a representative context with one another.

Evaluation Environment

The third challenge is to ensure that the evaluation environment has sufficient complexity, scope, and flexibility to address the first two challenges. Simulations have some potential, but hands-on experiments with real hardware are essential to maturing technologies and addressing the challenges.

RECENT ADVANCES

Recent advances in motion-capture technology combined with continued developments in small-scale electronics can enable the rapid design and evaluation of flight vehicle concepts (Troy et al., 2007). These evaluations can be extended to the mission level with additional vehicles and associated software.

Boeing has been collaborating with other researchers since 2006 on the development of an indoor flight-test capability for the rapid evaluation of multi-vehicle flight control (Halaas et al., 2009; How et al., 2008; Saad et al., 2009). Several other researchers have also been developing multi-vehicle test environments, both outdoor (Hoffmann et al., 2004; Nelson et al., 2006) and indoor (Holland et al., 2005; Vladimerouy et al., 2004).

Boeing has focused on indoor, autonomous flight capability where the burden of enabling flight is on the system rather than on the vehicles themselves. This arrangement makes it possible for novel concepts to be flown quickly with little or no modification. This also enables the rapid increase in the number of vehicles with minimal effort.

Boeing has also focused on improving the health and situational awareness of vehicles (Halaas et al., 2009). The expanded-state knowledge now includes information related to the power consumption and performance of various aspects of the vehicle. Automated behaviors are implemented to ensure safe, reliable flight with minimal oversight, and the dynamics of these behaviors are considered in mission software. The added information is important for maximizing individual and system performance.

EXPERIMENTAL ENVIRONMENT FOR INTEGRATED SYSTEMS

To address the challenges mentioned above, Boeing Research & Technology has integrated component technologies into an open architecture with simplified subsystems and systems with sufficient fidelity to explore critical, emergent issues. Representative, simple systems consisting of small, commercially available vehicles are modified to include health awareness. These systems are then combined under a modular architecture in an indoor flight environment that enables frequent integrated experiments under realistic fault conditions. Sufficient complexity is introduced to result in emergent behaviors and interactions between multiple vehicles, subsystems, the environment, and operators. This approach avoids the inherent biases of simulation-based design and evaluation and is open to “real-world” unknown unknowns that can influence overall system dynamics.

Vehicle Swarm Technology Laboratory

Boeing Research & Technology has been developing the Vehicle Swarm Technology Laboratory (VSTL), a facility that provides an environment for testing a variety of vehicles and technologies in a safe, indoor, controlled environment (Halaas et al. 2009; Saad et al., 2009). This type of facility not only can accommodate a significant increase in the number of flight test hours available over traditional flight-test ranges, but can also decrease the amount of time required to first flight of a concept. The primary components of the VSTL include a position-reference system, vehicles and associated ground computers, and operator interface software. The architecture is modular and thus supports rapid integration of new elements and changes to existing elements.

The position-reference system consists of a motion-capture system that emits coordinated pulses of light reflected by markers placed on the vehicles within viewing range of the cameras. Through coordinated identification by multiple cameras, the position and attitude of the marked vehicles is calculated and broadcast on a common network. This position-reference system has the advantages of allowing for the modular addition and removal of vehicles, short calibration time, and submillimeter and sub-degree accuracy.

The vehicles operated in the VSTL are modified, commercially available, remotely controlled helicopters, aircraft, and ground vehicles equipped with custom electronics in place of the usual onboard electronics. The custom electronics, which include a microprocessor loaded with common laboratory software, current sensors, voltage sensors, and a common laboratory communication system, enable communication with ground-control computers and add functionality. The ground computers execute the outer-loop control, guidance, and mission management functions.

A key component developed as part of the VSTL is improved vehicle self-awareness. A number of automated safety and health-based behaviors have been implemented to support simple, reliable, safe access to flight testing. Several command and control applications provide an interface between the operator and the vehicles. The level of interaction includes remotely piloted, low-level task control and high-level mission management. The mission management application was used to explore opportunities associated with health-based adaptations and obtain some initial information.

LESSONS LEARNED

Three missions were evaluated to determine the flexibility of the architecture and the indoor facility to test a variety of concepts rapidly. A specific metric was used for each mission to quantify performance. The first mission was non-collaborative and consisted of several vehicles repeatedly performing independent flight plans on conflicting trajectories. The metric was focused on evaluating flight safety and the performance of collision avoidance methodologies.

The second mission was an abstracted, extended-duration coordinated surveillance mission. The mission metric was associated with the level of surveillance provided in the presence of faults.

The third mission, an exercise to test the full capability of the architecture, highlighted the ability of vehicles and architecture to support a diversity of possible tasks. The mission involved the assessment of a hazardous area using multimodal vehicles and tasking. In addition, there were multiple human operators at different command levels. Success was measured as the completion of the tasks included in the mission and robustness in the presence of faults.

LESSONS LEARNED AND OPPORTUNITIES FOR FUTURE RESEARCH

Lessons Learned

The three experiments resulted in a number of lessons learned. The approach of integrating the various elements into a modular architecture and performing a range of simplified missions was validated; interactions among the various components exhibited complex behaviors, especially in the presence of faults; peripheral effects of inserting new technologies or algorithms were revealed; lower level functions, especially collision avoidance, need to be evaluated for a range of mission conditions; the role of operators, even in the essentially autonomous missions, was clear; in the presence of faults, sufficient situational awareness is necessary, as well as the ability to intervene if needed (although this capability existed, operators interacted with the system elements sometimes from the higher level command interface and sometimes from a lower level interface). These and other lessons indicate that further research will be necessary in several areas.

Areas for Future Research

First, we need a more formal framework for evaluating technologies and analyzing experimental results. This research should address the following questions: What tools can be developed to guide decisions about which technologies to insert? What is the risk of disrupting other functions? Second, we will need more research on interactions between systems and human operators, who are inevitably present and, thus, play a role in overall mission success: Can the influence of human operators be included in evaluating the overall potential benefit of a proposed technology?

We are hopeful that these and other questions that have emerged can be addressed using the capability and architecture that is already in place.

REFERENCES

Halaas, D.J., S.R. Bieniawski, P. Pigg, and J. Vian. 2009. Control and Management of an Indoor, Health Enabled, Heterogenous Fleet. Proceedings of the AIAA Infotech@Aerospace Conference and Exhibit and AIAA Unmanned . . . Unlimited Conference and Exhibit, AIAA-2009-2036, Seattle, Washington, 2009. Reston, Va.: AIAA.

Hoffmann, G., D.G. Rajnarayan, S.L. Waslander, D. Dostal, J.S. Jang, and C.J. Tomlin. 2004. The Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC). 23rd Digital Avionics System Conference, Salt Lake City, Utah, November 2004. New York: IEEE.

Holland, O., J. Woods, R. De Nardi, and A. Clark. 2005. Beyond swarm intelligence: the UltraSwarm. Pp. 217–224 in Proceedings of the 2005 IEEE Symposium, Pasadena, Calif., June 2005. New York: IEEE.

How, J., B. Bethke, A. Frank, D. Dale, and J. Vian. 2008. Real-time indoor autonomous vehicle test environment. IEEE Control Systems Magazine 28(2): 51–64.

Nelson, D.R., D.B. Barber, T.W. McLain, and R.W. Beard. 2006. Vector Field Path Following for Small Unmanned Air Vehicles. Pp. 5788–5794 in Proceedings of the 2006 American Control Conference, Minneapolis, Minn., June 14–16, 2006.

Saad, E., J. Vian, G. Clark, and S.R. Bieniawski. 2009. Vehicle Swarm Rapid Prototyping Testbed. Proceedings of the AIAA Infotech@Aerospace Conference and Exhibit and AIAA Unmanned . . . Unlimited Conference and Exhibit, AIAA-2009-1824, Seattle, Washington, 2009. Reston, Va.: AIAA

Troy, J.T., C.A. Erignac, and P. Murray. 2007. Closed-loop Motion Capture Feedback Control of Small-scale Aerial Vehicles. AIAA Paper 2007-2905. Infotech@Aerospace 2007 Conference and Exhibit, Rohnert Park, Calif., May 7–10, 2007. Reston, Va.: American Institute of Aeronautics and Astronautics.

Vladimerouy, V., A. Stubbs, J. Rubel, A. Fulford, J. Strick, and G. Dullerud. 2004. A Hovercraft Testbed for Decentralized and Cooperative Control. Pp. 5332–5337 in Proceedings of the 2004 American Control Conference, Boston, Mass., 2004.

Certifiable Autonomous Flight Management for Unmanned Aircraft Systems

ELLA M. ATKINS
University of Michigan

The next-generation air transportation system (NextGen) will achieve unprecedented levels of throughput[1] and safety by judiciously integrating human supervisors with automation aids. NextGen designers have focused their attention mostly on commercial transport operations, and few standards have been proposed for the burgeoning number of unmanned aircraft systems (UAS).[2] In this article, I describe challenges associated with the safe, efficient integration of UAS into the National Airspace System (NAS).

CURRENT AIRCRAFT AUTOMATION

Although existing aircraft autopilots can fly from takeoff through landing, perhaps the most serious technological impediment to fully autonomous flight is proving their safety in the presence of anomalies such as unexpected traffic, onboard failures, and conflicting data. Current aircraft automation is "rigid" in that designers have favored simplicity over adaptability. As a result, responding in emergency situations, particularly following events that degrade flight performance (e.g., a jammed control surface, loss of engine thrust, icing, or damage to the aircraft structure) requires the intervention and ingenuity of a human pilot or operator.

[1] In the context of NextGen, "throughput" is defined as the number of aircraft that can be moved through a particular airspace per unit time.

[2] The aerospace community has adopted the term "unmanned aircraft system" (UAS) to replace "unmanned air vehicle" (UAV), because a contemporary unmanned aircraft is a complex "system" of tightly integrated hardware and software that typically supports data acquisition, processing, and communication rather than cargo transport.

If the automation system on a manned aircraft proves to be insufficient, the onboard flight crew is immersed in an environment that facilitates decision making and control. Furthermore, modern aircraft rarely experience emergencies because their safety-critical systems are designed with triple redundancy.

ENSURING SAFETY IN UNMANNED AIRCRAFT SYSTEMS

To be considered "safe," UAS operations must maintain acceptable levels of risk to other aircraft and to people and property. An unmanned aircraft may actually fly "safely" throughout an accident sequence as long as it poses no risk to people or property on the ground or in the air. Small UAS are often considered expendable in that they do not carry passengers and the equipment itself may have little value. Thus, if a small UAS crashes into unimproved terrain, it poses a negligible risk to people or property.

UAS cannot accomplish the ambitious missions for which they are designed, however, if we limit them to operating over unpopulated regions. To reap the benefits of UAS, we must develop and deploy technologies that decrease the likelihood of a UAS encountering conditions that can lead to an incident or accident.

However, the recipe for safety on manned aircraft is impractical for small UAS. First, triple redundancy for all safety-critical systems would impose unacceptable cost, weight, and volume constraints for small aircraft. Second, although transport aircraft typically fly direct routes to deliver their payloads, surveillance aircraft are capable of dynamically re-planning their flight trajectories in response to the evolving mission or to observed data (e.g., the detection of a target to be tracked).

Finally, UAS are operated remotely, and operators are never directly engaged in a situation in which their lives are at risk. In fact, operators can only interact with a UAS via datalink, and "lost link" is currently one of the most common problems.[3]

SAFETY CHALLENGES FOR SMALL UNMANNED AIRCRAFT

With limited redundancy, highly dynamic routes, and strictly remote supervision, small UAS face formidable automation challenges. As the number of unmanned aircraft increases and as safety-oriented technology development continues to lag behind the development of new platforms, mission capabilities, and operational effi-ciency (e.g., one operator for multiple vehicles), it is becoming increasingly urgent that these issues be addressed. In addition, a large user base

[3] This issue was recently brought to our attention when a Fire Scout UAS aircraft lost its communication link and inappropriately flew quite close to restricted airspace around Washington, DC. (*http://www.nytimes.com/2010/08/26/us/26drone.html?_r=4&partner=rss&emc=rss*).

for UAS is emerging, which includes military and homeland security missions and commercial ventures.

Making the routine operation of unmanned aircraft safe wherever they are needed will substantially reduce the need for costlier manned flights that have a much greater adverse impact on the environment. However, for unmanned aircraft to operate near other aircraft or over populated areas, they must be capable of managing system failures, lost links, and dynamic routing, including collision avoidance, in a way that is "safe" for people and property.

We are currently working to augment autonomous decision making in the presence of actuator or sensor failures by expanding the definition of "flight envelope" to account for evolving physical, computational, perceptual, and environmental constraints. The flight envelope is traditionally defined by physical constraints, but under damage or failure conditions the envelope can contract. An autonomous flight controller must be capable of identifying and respecting these constraints to minimize the risk of loss-of-control as the aircraft continues on its mission or executes a safe emergency landing.

The autonomous flight manager can minimize risk by following flight plans that maximize safety margins first and then maximize traditional efficiency metrics (e.g., energy or fuel use). Thus flight plans for UAS may first divert the aircraft away from populated regions on the ground or densely occupied airspace and then decide whether to continue a degraded flight plan or end the mission through intentional flight termination or a controlled landing in a nearby safe (unpopulated) area. The key to certification of this autonomous decision making will be guaranteeing that acceptable risk levels, both real and perceived, are maintained.

ADDRESSING SAFETY CHALLENGES

In the discussion that follows, we look first at the problem of certifiable autonomous UAS flights in the context of current flight and air traffic management (ATM) technologies, which are primarily designed to ensure safe air transportation with an onboard flight crew. In this context, we also describe current and anticipated roles for automation and human operators.

Next, we characterize emerging UAS missions that are driving the need for fully autonomous flight management and integration into the NAS. Because loss-of-control is a major concern, I suggest an expanded definition of the flight envelope in the context of a real-life case study, the dual bird strike incident of US Airways Flight 1549 in 2009. That incident highlighted the need for enhanced automation in emergency situations for both manned and unmanned aircraft.

Finally, challenges to certification are summarized and strategies are suggested that will ultimately enable UAS to fly, autonomously, in integrated airspace over populated as well as rural areas.

FLIGHT AND AIR TRAFFIC MANAGEMENT: A SYSTEM-OF-SYSTEMS

In the NextGen NAS, avionics systems onboard aircraft will be comprised of a complex network of processing, sensing, actuation, and communication elements (Atkins, 2010a). UAS, whether autonomous or not, must be certified to fit into this system. All NextGen aircraft will be networked through datalinks to ATM centers responsible for coordinating routes and arrival/departure times.

The Federal Aviation Administration (FAA) and its collaborators have proposed a system-wide information management (SWIM) architecture (*www.swim.gov*) that will enable collaborative, flexible decision-making for all NAS users; it is assumed that all NextGen aircraft will be capable of accurately following planned 4-D trajectories (three-dimensional positions plus times), maintaining separation from other traffic, and sharing pertinent information such as GPS coordinates, traffic alerts, and wind conditions. Protocols for system-wide and aircraft-centric decision making must be established to handle adverse weather conditions, encounters with wake turbulence, and situations in which other aircraft deviate from their expected routes.

To operate efficiently in controlled NextGen airspace, all aircraft will be equipped with an onboard flight management system (FMS) that replicates current functionality, including precise following of the approved flight plan, system monitoring, communication, and pilot interfaces (Fishbein, 1995; Liden, 1994). Automatic Dependent Surveillance–Broadcast (ADS-B) systems will also communicate aircraft status information (e.g., position, velocity) to ensure collision avoidance. Without such equipment, it will be difficult to guarantee that traffic remains separated throughout flight, especially when manned and unmanned aircraft are involved.

Low-Cost Flight Management Systems

Small operators, from general and sports aviation to unmanned aircraft, will require low-cost options to the current FMS. Although advanced miniaturized electronics can make low-cost, lightweight FMS possible (Beard, 2010), producing and marketing these systems will require a concerted effort in the face of potentially slim profit margins and formidable validation and verification requirements.

The current FMS can devise and follow a flight plan from origin to destination airport. In the future, automation in both manned and unmanned aircraft is expected to include making and coordinating dynamic routing decisions based on real-time observations (e.g., weather), other traffic, or even mission goals (e.g., target tracking). Quite simply, we are rapidly moving toward collaborative human-machine decision making or fully autonomous decision making rather than relying on human supervisors of autonomous systems, particularly if operators are not onboard.

From Lost Link to Optional Link

Today's unmanned aircraft are flown by remote pilots/operators who designate waypoints or a sequence of waypoints, as well as a rendezvous location. However, as was mentioned above, communication (lost link) failure is a common and challenging unresolved issue for UAS. Addressing this problem will require that developers not only improve the availability of links, but simultaneously pursue technological advances that will render links less critical to safety.

As the level of autonomy increases to support extended periods of operation without communication links, UAS must be able to operate "unattended" for extended periods of time, potentially weeks or months, and to collect and disseminate data without supervision unless the mission changes.

Sense-and-Avoid Capability

Because human pilots cannot easily see and avoid smaller UAS, "sense and avoid" has become a top priority for the safe integration of UAS into NAS. A certified sense-and-avoid technology will provide another step toward fully autonomous or unattended flight management.

EMERGING UNMANNED MISSIONS

A less-studied but critical safety issue for UAS operations as part of NAS is maintaining safe operations in the presence of anomalies. Researchers are beginning to study requirements for autonomously carrying out UAS missions (Weber and Euteneuer, 2010) with the goal of producing automation technology that can be certified safe in both nominal and conceivable off-nominal conditions. In this section, we focus on the "surveillance" missions that distinguish UAS—particularly small unmanned aircraft that must operate at low cost in sparsely populated airspace—from traditional transport operations.

Traditional Transport Operations

Traditional transport aircraft have a single goal—to fly a human or cargo payload safely from an origin to a destination airport with minimal cost to the airline. The "best" routes are, therefore, direct, with vectors around traffic or weather as needed. Schedules can be negotiated up to flight time, and passengers and cargo carriers expect on-time delivery, as costs increase with delay. In the context of autonomous transport UAS (e.g., cargo carriers), issues include loss of facilities or adverse weather at the destination airport, failure or damage conditions (e.g., loss of fuel or power) that render the destination unreachable, and security issues that result in a system-wide change in flight plans (e.g., temporary flight restrictions).

Unmanned Surveillance Aircraft

Unlike traditional transport aircraft, the goal of surveillance unmanned aircraft may be to search a geographical region, to loiter over one or more critical sites, or to follow a surveillance target along an unpredictable route. A summary of potential commercial applications (Figure 1) that complement the myriad of military uses for surveillance flights, shows that surveillance and support are the primary emerging mission categories that will require the expansion of existing NAS protocols to manage dynamic routing and the presence of UAS in (1) uncontrolled, low-altitude airspace currently occupied primarily by general aviation aircraft and (2) congested airport terminal areas where traffic is actively managed (Atkins et al., 2009).

This will mean that UAS will mix with the full fleet of manned operations, ranging from sports and recreational aircraft operated by pilots with limited training to jets carrying hundreds of passengers. UAS missions also will overfly populated areas for a variety of purposes, such as monitoring traffic, collecting atmospheric data over urban centers, and inspecting sites of interest. Even small unmanned aircraft have the capacity to provide support for communication, courier services, and so on.[4]

Unmanned aircraft can work in formations that can be modeled and directed as a single entity by air traffic controllers. This capability can give controllers much more leeway in sequencing and separating larger sets of traffic than would be possible if all UAS flights were considered distinct.

UAS teams may also negotiate tasks but fly independent routes, such as when persistent long-term coverage is critical to a successful mission or when cooperative coverage from multiple angles is necessary to ensure that a critical ground target is not lost in an urban environment. Some activities may be scheduled in advance and prioritized through equity considerations (e.g., traffic monitoring), but activities related to homeland security or disaster response are unscheduled and may take priority even over airline operations.

Although the effects of high-altitude UAS must be taken into account by NAS, low-altitude aircraft operating over populated regions or in proximity to major airports will be the most challenging to accommodate in the NextGen NAS. UAS must, of course, be safe, but they must also be fairly accommodated through the extension of NAS metrics (e.g., access, capacity, efficiency, and flexibility) so they can handle operations when persistent surveillance over a region of interest is more important than equitable access to a congested airport runway.

[4] Large cargo carriers would also benefit from flying unmanned aircraft, which require only base personnel who would not have to be located at potential departure sites.

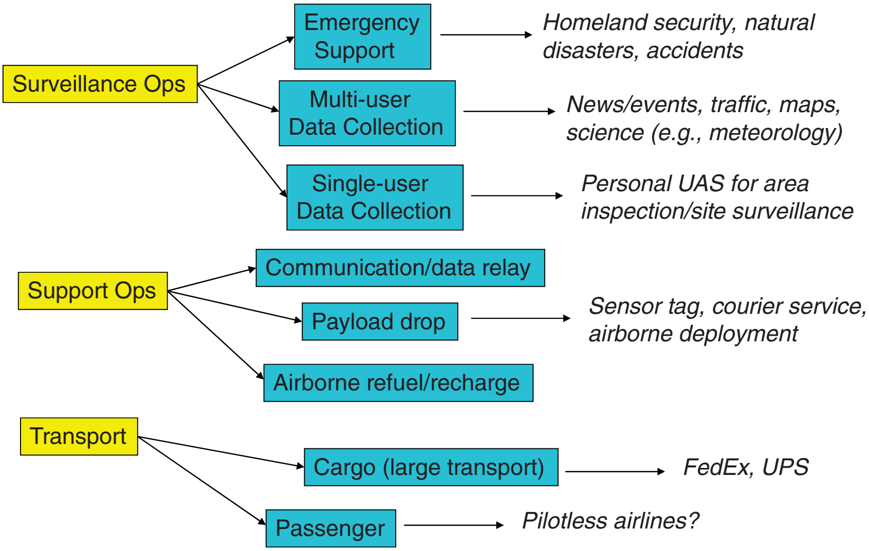

Surveillance Ops
Emergency Support
Homeland security, natural disasters, accidents
Multi-user Data Collection
News/events, traffic, maps, science (e.g., meteorology)
Single-user Data Collection
Personal UAS for area inspection/site surveillance
Support Ops
Communication/data relay
Payload drop
Sensor tag, courier service, airborne deployment
Airborne refuel/recharge
Transport
Cargo (large transport)
FedEx, UPS
Passenger
Pilotless airlines?

FIGURE 1 Emerging commercial applications for unmanned aircraft. Source: Atkins et al., 2009.

EXTENDING THE FLIGHT ENVELOPE TO MINIMIZE THE RISK OF LOSS-OF-CONTROL

Loss-of-control, the most frequent cause of aviation accidents for all vehicle classes, occurs when an aircraft exits its nominal flight envelope making it impossible to follow its desired flight trajectory (Kwatny et al., 2009). Current autopilot systems rely on intuitive, linearized, steady-flight models (Figure 2) that reveal how aero-dynamic stalls and thrusts constrain the flight envelope (McClamroch, in press).

To ensure the safe operation of UAS and to prove that autonomous system performance is reliable, an FMS for autonomous aircraft capable of *provably* avoiding loss-of-control in all situations where avoidance is possible will be essential. This will require that the autonomous system understand its flight envelope sufficiently to ensure that its future path only traverses “stabilizable” flight states (i.e., states the autonomous controller can achieve or maintain without risking loss-of-control).

Researchers are beginning to develop nonlinear system-identification and feedback control algorithms that offer stable, controlled flight some distance beyond the nominal “steady flight” envelope (Tang et al., 2009). Such systems could make it feasible for an autonomous system to “discover” this more expansive envelope (Choi et al., 2010) and continue stable operation despite anomalies

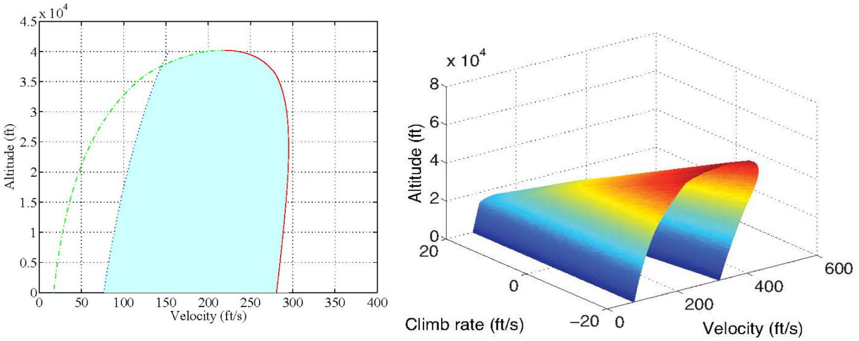


FIGURE 2 Examples of steady-level (left) and 3-D (right) traditional flight envelopes. Source: McClamroch, in press.

in the environment (e.g., strong winds) or onboard systems (e.g., control-surface failures or structural damage) that would otherwise lead to loss-of-control.

Flight Envelope Discovery

Figure 3 shows the flight envelope for an F-16 with an aileron jammed at 10 degrees. In this case, the aircraft can only maintain steady straight flight at slow speeds. The traversing curve shows an example of a flight envelope discovery process incrementally planned as the envelope is estimated from an initial high-speed turning state through stabilizable states to a final slow-speed straight state (Yi and Atkins, 2010).

This slow-speed, gentle-descent final state and its surrounding neighborhood are appropriate for final approach to landing, indicating that the aircraft can safely fly its approach as long as it remains within the envelope. Once the envelope has been identified, a landing flight plan guaranteed to be feasible under the condition of the control surface jam can be automatically generated.

Figure 4 illustrates the emergency flight management sequence of discovering the degraded flight envelope, selecting a nearby landing site, and constructing a feasible flight plan to that site. Although a runway landing site is presumed in the figure, an off-runway site would probably be selected for a small UAS that required little open space for landing. The sequence in Figure 4 mirrors the emergency procedures a pilot would follow when faced with degraded performance. Note that all of the steps in this process could be implemented autonomously with existing technology.

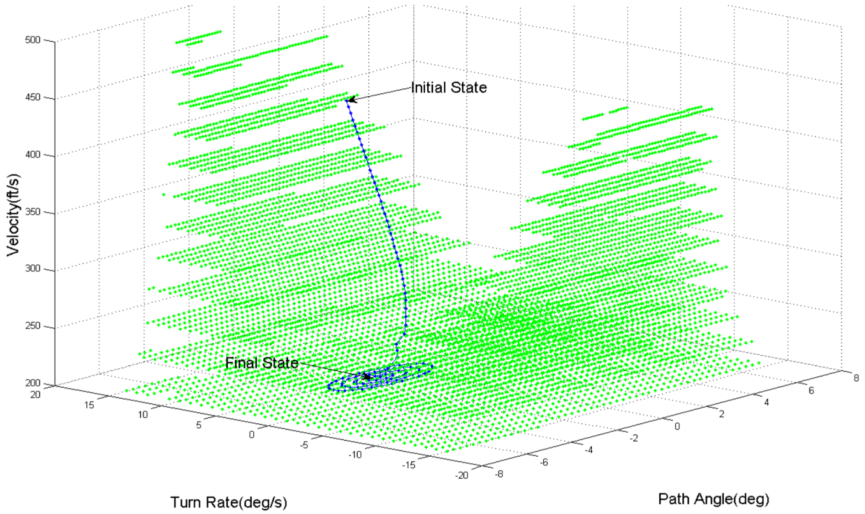


FIGURE 3 Trim-state discovery for an F-16 with a 10-degree aileron jam. Source: Yi and Atkins, 2010.

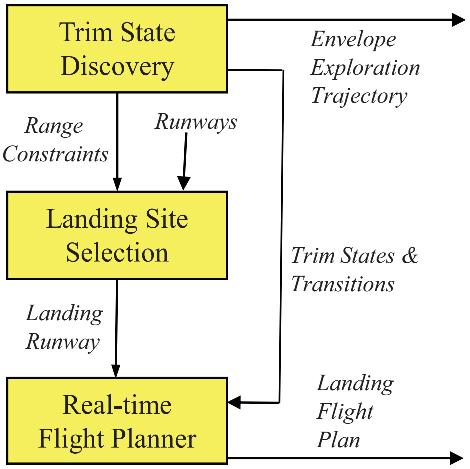


FIGURE 4 Simplified version of an emergency flight-planning sequence for a jet with degraded performance.

Autonomous Reaction to Off-Nominal Conditions

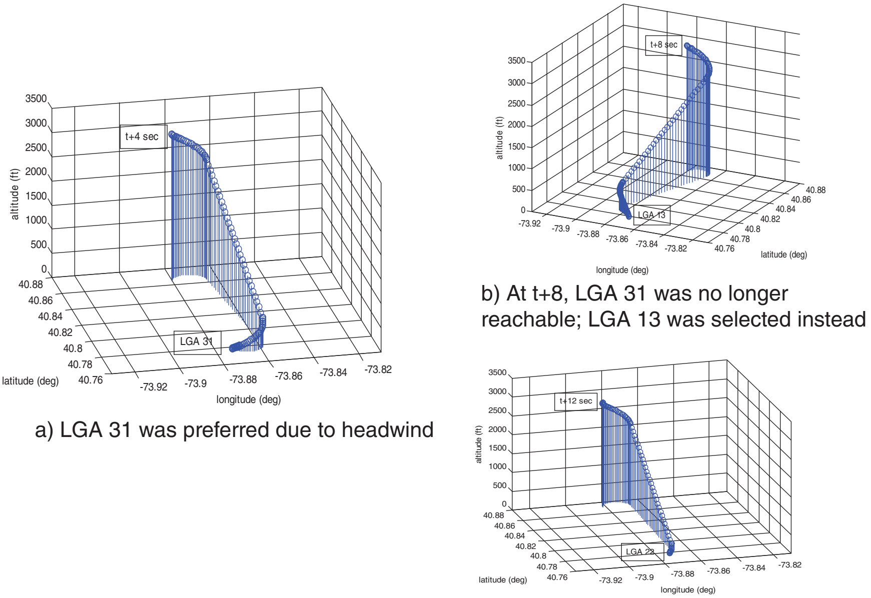
The remaining challenge is to prove that such an autonomous system is capable of recognizing and reacting to a sufficient range of off-nominal situations to be considered "safe" without a human pilot as backup. To illustrate how autonomous emergency flight management could improve safety, we investigated the application of our emergency flight planning algorithms to the 2009 Hudson River landing (Figure 5) of US Airways Flight 1549 (Atkins, 2010b).

About two minutes after the aircraft departed from LaGuardia (LGA) Airport in New York, it encountered a flock of large Canada geese. Following multiple bird strikes, the aerodynamic performance of the aircraft was unchanged, but propulsive power was no longer available because of the ingestion of large birds into both jet engines, which forced the aircraft to glide to a landing. In this event, the pilot aptly glided the plan to a safe landing on the Hudson River. All passengers and crew survived, most with no injuries, and the flight crew has been rightly honored for its exemplary performance.

In the case of Flight 1549, our adaptive flight planner first identified the glide (no-thrust) footprint from the coordinates at which thrust was initially lost. This analysis indicated that the aircraft could return to LGA as long as the return was initiated rapidly, before too much altitude was lost. Our landing site search algorithm prioritized LGA runway 31 as the best choice because of its headwind, but runways 13 and 22 were also initially reachable.

Figure 6 illustrates the feasible landing trajectories for Flight 1549 automatically generated in less than one second by our pre-existing engine-out flight planner adapted to Airbus A320 glide and turn capabilities. Notably, runway 31 was reach-

FIGURE 5 Post-landing photo of US Airways Flight 1549 in the Hudson River (*http://www.wired.com/images_blogs/autopia/2010/01/us_airways_1549_cropped.jpg*).



a) LGA 31 was preferred due to headwind

b) At t+8, LGA 31 was no longer reachable; LGA 13 was selected instead

c) At t+12, only LGA 22 was reachable

FIGURE 6 Feasible trajectories for Flight 1549 to return to LaGuardia Airport. Source: Atkins, 2010b.

able only if the turn back to LGA was initiated within approximately 10 seconds after the incident. Runways 13 and 22 were reachable for another 10 seconds, indicating that the pilot (or autopilot if available) did in fact have to initiate the return to LGA no more than approximately 20 seconds after thrust was lost.

We believe that if an automation aid had been available to rapidly compute and share the safe glide trajectory back to LGA, and if datalink coordination with air traffic control had been possible to facilitate clearing LGA departure traffic, Flight 1549 could have returned to LGA and avoided the very high-risk (albeit successful in this case) water landing. In short, this simple, provably correct "glide to landing" planning tool represents a substantial, technologically sound improvement over the level of autonomous emergency flight management available today and is a step toward the more ambitious goal of fully autonomous flight management.

CERTIFICATION OF FULLY AUTONOMOUS OPERATION

Every year the FAA is asked to certify a wide variety of unmanned aircraft for flight in the NAS. Although most unmanned operations are currently conducted over remote regions where risks to people and property are minimal, certification is and must continue to be based on guarantees of correct responses in nominal conditions, as well as contingency management to ensure safety. Although redundancy will continue to be key to maintaining an acceptable level of risk of damage to people and property in the event of failures, for UAS aircraft, triple redundancy architecture as is present in commercial transport aircraft may not be necessary because ditching the aircraft is often a viable option.

Safety certification is a difficult process that requires some trust in claims by manufacturers and operators about aircraft design and usage. Automation algorithms, however, can ultimately be validated through rigorous mathematical and simulation-based verification processes to provide quantitative measures of robustness, at least for envisioned anomalies in weather, onboard systems, and traffic.

Addressing Rigidity in Flight Management Systems

The remaining vulnerability of a fully autonomous UAS FMS is its potential rigidity, which could lead to an improper response in a truly unanticipated situation. The default method for managing this vulnerability has been to insert a human pilot into the aircraft control loop. However, with remote operators who have limited engagement with the aircraft, human intervention may not be the best way to offset automation rigidity. If that is the case, the certification of fully autonomous UAS FMS must be based on meeting or exceeding human capabilities, although assessing the human capacity for response will be challenging.

For remote unmanned aircraft, we can start by characterizing the bounds on user commands. Formal methods of validating and verifying automation algorithms and their implementations, as well as assessing their flexibility (rigidity), will also be essential. Simulation and flight testing will, of course, be necessary to gain trust, but we propose that simulation should be secondary to formal proofs of correctness when assessing the performance, robustness, and ultimately safety of autonomous UAS.

CONCLUSION

Ultimately, fully autonomous UAS operation will be both technologically feasible and safe. The only remaining issue will be overcoming public perceptions and lack of trust, which we believe can be mitigated by long-term exposure to safe and beneficial UAS operations.

REFERENCES

Atkins, E. 2010a. Aerospace Avionics Systems. Chapter 391 in Encyclopedia of Aerospace Engineering, edited by R. Blockey and W. Shyy. New York: Wiley

Atkins, E. 2010b. Emergency Landing Automation Aids: An Evaluation Inspired by US Airways Flight 1549. Presented at the AIAA Infotech@Aerospace Conference, Atlanta, Georgia, April 2010.

Atkins, E., A. Khalsa, and M. Groden. 2009. Commercial Low-Altitude UAS Operations in Population Centers. Presented at the American Institute of Aeronautics and Astronautics (AIAA) Aircraft Technology, Integration, and Operations Conference, Hilton Head, South Carolina, September 2009.

Beard, R. 2010. Embedded UAS Autopilot and Sensor Systems. Chapter 392 in Encyclopedia of Aerospace Engineering, edited by R. Blockey and W. Shyy. New York: Wiley.

Choi, H., E. Atkins, and G. Yi. 2010. Flight Envelope Discovery for Damage Resilience with Application to an F16. Presented at the AIAA Infotech@Aerospace Conference, Atlanta, Georgia, April 2010.

Fishbein, S.B. 1995. Flight Management Systems: The Evolution of Avionics and Navigational Technology. Santa Barbara, Calif.: Praeger.

Kwatny, H., J. Dongmo, B. Cheng, G. Bajpai, M. Yawar, and C. Belcastro. 2009. Aircraft Accident Prevention: Loss-of-Control Analysis. In Proceedings of the Guidance, Navigation, and Control Conference, Chicago, Ill., August 2009. Reston, Va.: AIAA.

Liden, S. 1994. The Evolution of Flight Management Systems. Pp. 157–169 in Proceedings of the IEEE Digital Avionics Conference. New York: IEEE.

McClamroch, N.H. In press. Steady Aircraft Flight and Performance.

Tang, L., M. Roemer, J. Ge, A. Crassidis, J. Prasad, and C. Belcastro. 2009. Methodologies for Adaptive Flight Envelope Estimation and Protection. In Proceedings of the Guidance, Navigation, and Control Conference, AIAA, Chicago, Ill., August 2009. Reston, Va.: AIAA.

Weber, R., and E. Euteneuer. 2010. Avionics to Enable UAS Integration into the NextGen ATS. Presented at AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario, August 2010.

Yi, G., and E. Atkins. 2010. Trim State Discovery for an Adaptive Flight Planner. Presented at the Aerospace Sciences Meeting, AIAA, Orlando, Fl., January 2010.

Engineering Inspired by Biology

Introduction

Mark Byrne
Auburn University

Babak Parviz
University of Washington

It has taken biological systems and physiological processes millions of years to evolve with the precise properties and functions they have today. Engineers have only recently developed an appreciation of the sophistication of biological systems, and they are looking to them for inspiration in the rational design of materials and systems. By studying and mimicking complex biological structures and processes, engineers can now design materials and devices with novel features and enhanced properties to help solve problems in a wide variety of disciplines, from health care to small-scale electromechanical devices. In this session, the presenters highlight bio-inspired, biomimetic, or bio-derived technologies and innovations and look ahead to what the future may hold in this field. The connecting thread among these talks is the diverse role biology plays in contemporary engineering, as bio-derived or bio-inspired technologies are pivotal to novel engineering solutions in a number of fields.

A revolution in health care is expected in the near future when low-cost genome sequencing for individuals becomes a reality. The first talk, by Mostafa Ronaghi (Illumina), highlights engineering challenges in the analysis of genetic variation, gene expression, and function. Addressing these challenges involves mimicking and exploiting biological recognition and/or function with detection at high fidelity. For example, single nucleotide discrimination through nanopores is possible under an applied field that mimics the highly versatile ion channel.

The challenges to bio-inspired engineering are many, but the benefits will be tremendous in determining mechanisms of disease, drug candidates, and clinical molecular diagnostics. Advances will lead not only to faster screening and detection of diseases, but also to the tailoring of therapeutics based on an individual's genetic predisposition to disease, or personalized medicine with individualized therapeutics. The efficient, effective delivery of therapeutics to the patient will be inherent in these developments. Thus, the next talk, by Efie Kokkoli (University of Minnesota), focuses on controlled, targeted drug delivery, specifically using biology in the design of targeted therapeutics. Delivering the optimal amount of therapeutic to the right place at the right time is a significant goal.

The final, capstone talk, by Henry Hess (Columbia University), is on how biomolecules can be used as motor-powered devices in systems, whether the system is the cell itself or whether biomolecules are used to provide an actuation mechanism on a micro/nano-electromechanical (MEMS/NEMS) device.

The Current Status and Future Outlook for Genomic Technologies

Mostafa Ronaghi
Illumina

Jeffrey Fisher
Applied Genomics Group, Research and Development

Genomics emerged as a scientific field after the invention of the original DNA sequencing technique by Fredrick Sanger (Sanger et al., 1977a,b). Sanger introduced a chemical method for reading about 100 nucleotides, which, at the time, took about six months of preparation. Thanks to a large community of scientists worldwide, Sanger's technique eventually evolved to become the technology of choice for sequencing. The draft sequencing of the first human genome took about 13 years to complete, and the project cost some $3 billion.

Pyrosequencing, the second alternative technology, is based on sequencing-by-synthesis, which could be parallelized to enable higher throughput by more than 100 fold (Ronaghi et al., 1996, 1998). Pyrosequencing was used to sequence thousands of microbial and larger genomes, including James Watson's genome.

In 2006, a private company (Illumina) introduced reversible dye-terminator sequencing-by-synthesis (Bentley, 2006). This technology has increased throughput by ~10,000 fold in the last four years and reduced the cost of sequencing a human genome to less than $10,000. The most recent system based on this chemistry allows sequencing of several human genomes in a single run.

In this article, we describe dye-terminator sequencing-by-synthesis and efforts to reduce costs even further. In addition, we discuss emerging applications and challenges to bringing genomics into the mainstream.

BACKGROUND

On the most fundamental level, sequencing the genome consists of just a handful of basic biochemical steps. The challenge is posed by the enormous scale of molecularly encoded information—two almost identical strands, each consisting of 3.2 billion base pairs of information for the human genome—that must be processed through those steps. Furthermore, a typical genome is read to 30X coverage, which means that each base pair is read on average 30 times (on separate strands of DNA), giving a total throughput per genome of 100 billion base pairs.

The processing and reading of these immense amounts of information has been made possible by the adoption of engineering-based approaches to massive parallelization of the sequencing reactions. All current-generation sequencing platforms coordinate chemical, engineering, and computation subsystems on an unprecedented scale (measured in information throughput) (Figure 1).

DNA SEQUENCING

Sequencing, which determines the arrangement of the four genetic bases (A, T, C, and G) in a given stretch of DNA, relies on four steps (Metzker, 2010; Pettersson et al., 2009; Shendure and Ji, 2008):

1. **Fragmentation**—breaking the genome into manageable segments, usually a few hundred base pairs long.
2. **Isolation**—capturing the segments in a way that keeps the signals they present distinct.
3. **Amplification**—although single-molecule techniques can theoretically proceed without this step, most systems apply some form of clonal amplification to increase the signal and accuracy of sequencing.
4. **Readout**—transforming the genetic information base by base into a machine-readable form, typically an optical (fluorescent) signal.

Although the field of genomics has evolved in recent years to include a variety of sequencing systems, including some that do not necessarily follow this exact pattern, the majority of commercial platforms use all four steps in one form or another.

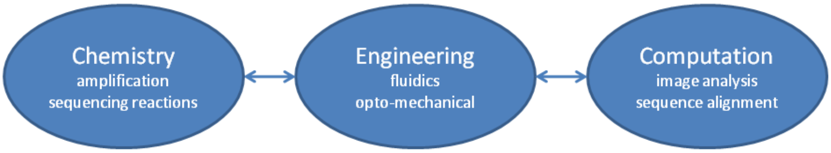


FIGURE 1 Modern sequencing requires highly coordinated subsystems to handle the throughput of massive amounts of information.

THE GENOME ANALYZER AND HISEQ SYSTEMS

The Illumina Genome Analyzer and HiSeq systems are examples of the massively parallel nature of the biochemical workflow described in the four steps listed above (Bentley et al., 2008). First a sample of DNA is fragmented into segments ~400 base pairs long, and oligonucleotides of known sequence are ligated to the ends. These ligated adapters function as "handles" for each segment, allowing it to be manipulated in downstream reactions. For example, they provide a means of trapping the DNA segment in the flow cell and later releasing it. They also provide areas where primers can bind for the sequencing reaction.

Next, the sample is injected into a flow cell containing a lawn of oligonucleotides that will bind to the adapters on the DNA segments (Figure 2a). The concentration is carefully controlled so that only one strand is present in a given area of the chip—representing the signal isolation step. The segment is then amplified in place by means of a substrate-bound polymerase chain reaction process (called bridge PCR), until each single segment has grown into a cluster of thousands of identical copies of the sequence (Figure 2b). A single flow cell finally contains several hundred million individual clusters. Although they are now larger than the initial single strand, the clusters remain immobile and physically separated from each other, making it possible to visually distinguish them during the readout step.

The genetic sequence is then transformed into a visual signal by synthesizing a complementary strand, one base at a time, using nucleotides with four separate color tags (Figure 3a). For each cycle (during which a single base per cluster is

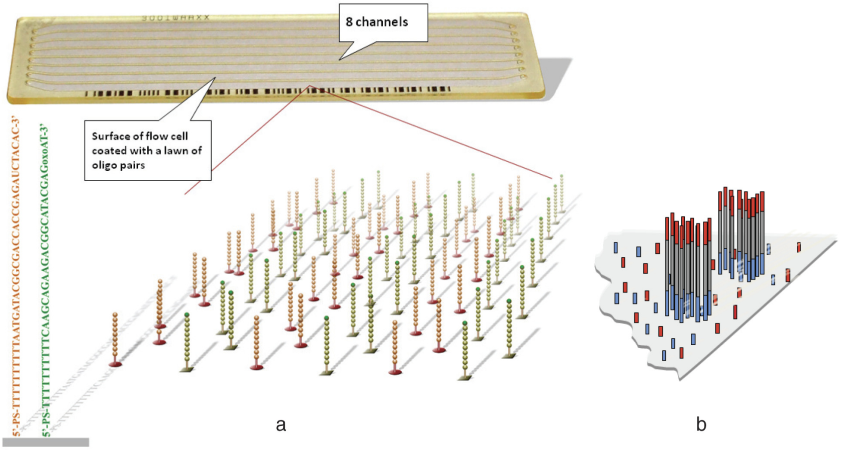

8 channels
Surface of flow cell coated with a lawn of oligo pairs
5'-PS-TTTTTTTTTTAATGATACGGCGACCACCGAGAUCTACAC-3'
5'-PS-TTTTTTTTTTCAAGCAGAAGACGGCATACGAGoxoAT-3'
a
b

FIGURE 2 (a) Adapter-ligated segments of DNA are loaded into a flow cell coated with a lawn of oligonucleotides to capture and bind DNA segments. (b) Once attached, the DNA is PCR amplified in place to form clonal clusters.

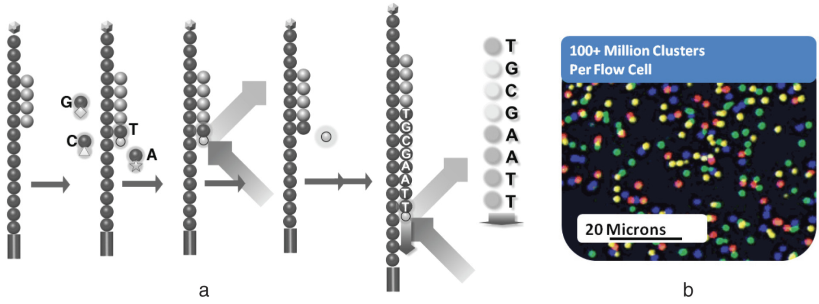


FIGURE 3 Sequencing-by-synthesis with reversible dye-terminator chemistry. During each round, (a) fluorescently-labeled nucleotides (with different colors for A, T, C, and G) are added to the flow cell. The nucleotide complementary to the next open position is then incorporated, and the entire flow cell is imaged, with the color of each cluster showing exactly which nucleotide was incorporated into that cluster during the current cycle. The label and terminator are then cleaved from the nucleotide so the reaction can begin anew. This process is repeated hundreds of times to read out a single contiguous sequence. The image map taken each cycle (b) is a four-color mosaic of clusters each emitting a color corresponding to the most recently incorporated nucleotide. The clusters are randomly distributed across the surface of the flow cell, but because they are immobilized at a fixed location, the progress of each can be followed from cycle to cycle. Color figure available online at *http://www.nap.edu/catalog.php?record_id=13043*.

read), DNA polymerase incorporates a single nucleotide that matches the next base on the template sequence. All four nucleotides, each carrying a different dye, are added in a mixture, but only one nucleotide is incorporated into the growing DNA strand. The incorporated nucleotide has a terminator group that blocks subsequent nucleotides from being added. The entire flow cell is then imaged, and the color of each cluster indicates which base was added for that sequence (Figure 3b). Finally, the terminator group and fluorophore are cleaved (i.e., chemically separated) from the nucleotide, and the cycle begins again.

This process is repeated until each cluster has been read 100 to 150 times. The segment can then be "flipped over," and another 100 to 150 bases of sequence information can be read from the other end. Thus, the total amount of information that can be garnered from a single flow cell is directly proportional to the number of clusters and the read length per cluster, both of which represent targets for improvement as we continually increase system throughput.

MOORE'S LAW AND GENOMICS

The often-quoted Moore's law posits that the number of transistors on an integrated circuit will double every 18 to 24 months, consequently reducing the cost per transistor (Figure 4). Sequencing costs have demonstrated a similar exponential decrease over time, but at an even faster pace.

One factor that has made this possible is that, unlike transistors, which have a density limited to improvements in the two-dimensional efficiency (surface area) of the chip, sequencing density can increase along a "third dimension," which is the read length. Therefore, each subsystem in Figure 1 can be improved to increase the total throughput of the system. Improvements in the chemistry have resulted in improved accuracy, longer read lengths, and shorter cycle times. In addition, by increasing both the area of the flow cell and the density of clusters, the total number of clusters has also been increased.

The engineering subsystem has doubled the throughput by using both the top and bottom of the flow cell for cluster growth. Cluster density has been increased by improving the optics and the algorithms that detect clusters. Total run time is regularly decreased by using faster chemistries, faster fluidics, faster optical scanning, and faster algorithms for image processing and base calling. On the one hand, improvements in each subsystem independently contribute to increases in throughput. On the other hand, an improvement in one system often becomes the leading driver for advances in the others.

FRONTIERS IN GENOMICS

The way forward lies in improving the technology so that it can be adapted to a broader range of applications. Three ways to achieve this are: (1) increasing

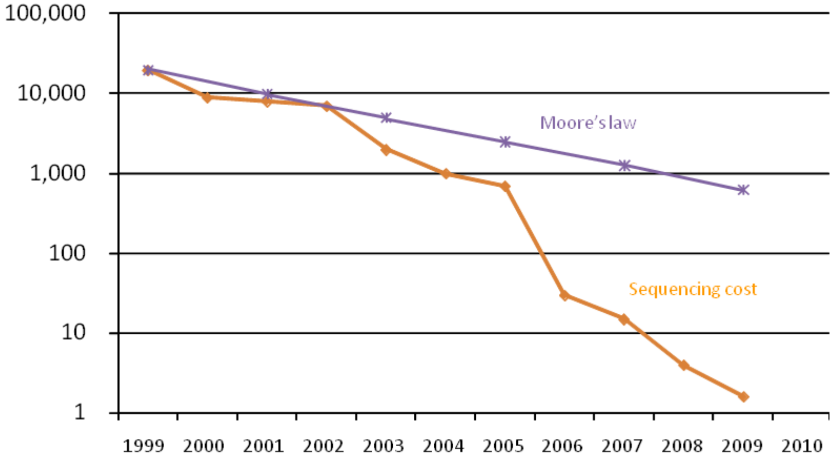


FIGURE 4 As the throughput of sequencing systems increases exponentially, the costs drop accordingly, outpacing the rate of change in Moore's law (arbitrary cost units along the y-axis).

accuracy to enable all diagnostic applications; (2) increasing the sensitivity of the system so that it can more robustly handle lower signal-to-noise ratios; and (3) increasing throughput to drive down costs.

Improving the Accuracy of Diagnoses

Using the methods described above, one can sequence an entire human genome starting with less than one microgram of DNA, about the amount of genetic material in fewer than 150 cells. However, there are other types of samples for which even this relatively modest amount of material is difficult to come by. For example, many researchers are beginning to look at the genomics of single cells—and not just one single cell, but processing small populations individually to evaluate the hetero-geneity in the group (Kurimoto and Saitou, 2010; Taniguchi et al., 2009; Walker and Parkhill, 2008). However, because a cell contains only about 6 picograms (pg) of genomic DNA and 10 pg of RNA, the corresponding signal is many orders of magnitude weaker than normal.

Another sample type that would benefit from improved assay sensitivity is a formalin-fixed, paraffin-embedded (FFPE) sample. FFPEs are histological tissue samples that have been chemically fixed to reduce degeneration, so they can be stained and examined under a microscope. Because there are huge archives of historical samples for which detailed patient outcomes are already known, researchers can use FFPE samples as resources to improve diagnosis by tracking down the genetic markers of disease. In addition, more accurate prognoses and more effective treatments are possible by studying the correlation between disease progression and genetic type in these earlier patients.

Unfortunately, fixing, staining, and storing FFPE samples can break down the genetic material, thus making sequencing or genotyping much more difficult. Nevertheless, the ability to use these samples and perform genomic analysis on them represents an invaluable resource for tracking down genetic contributions to disease and wellness (Bibikova et al., 2004; Lewis et al., 2001; Schweiger et al., 2009; Yeakley et al., 2005).

Increasing Sensitivity to Signal-to-Noise Ratios

In some cases, the signal itself is present at normal levels, but a much higher level of background noise drowns it out. For example, there has recently been a good deal of interest in studying the microbiome of different environments, such as soil, seawater, and the human gut (Gill et al., 2006; Turnbaugh et al., 2007; Woyke et al., 2009). In these cases, the genetic diversity of the sample can make it difficult to separate the components of different organisms.

Genomics also plays a vital role in the study of cancer (Balmain et al., 2003; Jones and Baylin, 2002; Stratton et al., 2009), which is defined by its genetic instability and pathology (Lengauer et al., 1998; Loeb, 1991, 2001). However, cells taken even from the same tumor can exhibit extreme genetic heterogeneity making increased sensitivity and detection key to distinguishing the often subtle differences that lead to one outcome as opposed to another. Sequencing this kind of sample requires much deeper coverage (7,200X read redundancy per base) than the typical 30X coverage for a homogenous sample.

Reducing Costs

Increases in throughput will affect the quantity of genetic information available, and the resultant decrease in cost will open up completely new markets, representing a qualitative shift in the ways in which genomics impacts our daily lives. When the cost of sequencing an entire genome is comparable to the current cost of analyzing a single gene, the market will experience a watershed moment as a flood of new applications for sequencing become possible. Diagnosis, prognosis, pharmacogenomics, drug development, agriculture—all will be changed in a fundamental way.

When whole-genome sequencing is priced in the hundreds of dollars, it will begin to be used all around us. It will become standard to have a copy of one's own genome. As *de novo* sequencing brings the genomes of an increasing variety of organisms into the world's data-bases, the study of biology will change from a fundamentally morphological classification system to genetically based classification. In agriculture, sequencing can act as an analog of a tissue-embedded radio frequency identification device (RFID); but instead of having to tag a sample with an electronic technology, we will simply extract some genetic material from a sample and sequence it, leading back to the very farm from which it came.

Today, it typically takes 12 years to bring a new drug to market, half of which is spent on discovery and half on approval; sequencing plays a role in both stages. During drug discovery, the pathways elucidated by genomic analysis lead to targeted development and shorter discovery cycles. The approval process will be facilitated by using genetic testing to define the patient populations involved in the testing of new drugs. Genetic testing will make it possible to account for genetic variation in a trial subject group when assessing efficacy and side effects. This will also lead to an improvement in treatment after a drug has been approved, as companion genetic tests for drugs will help doctors make informed decisions about how a drug might interact with a patient's genetic makeup.

CHALLENGES

Peripheral Systems

Achieving the improvements described above will require overcoming technical obstacles directly related to chemical, engineering, and computation modules of sequencing systems. However, some of the most significant bottlenecks to throughput are found not in sequencing itself, but in the peripheral (or ancillary) systems upstream and downstream of the process.

On the upstream side, for example, the rate at which samples are sequenced now outpaces the rate at which they can be prepared and loaded. At a conference last month, the Broad Institute described its ongoing efforts to increase the number of samples a technician can prepare each week from 12 or 15 to almost 1,000 by making sample preparation faster, cheaper, and with higher throughput (Lennon, 2010).

On the downstream end of the system we are beginning to bump up against throughput limits as well. Currently the HiSeq 2000 system produces about 40 GB (represented either as gigabases or gigabytes) of sequence information per day (*http://www.illumina.com/systems/ hiseq_2000.ilmn*). Information generated and accumulated at that rate cannot conveniently be handled by a local desktop computer. The storage, manipulation, and analysis of this information can only be done in the "cloud," whether by local servers, dedicated off-site servers, or third parties.

Although this amount of information can easily be transferred over network hardware, as sequencing systems advance to more than 1 terabyte per day, the physical infrastructure of data networks will begin to become a limiting factor. New algorithms and standards for non-lossy compression of whole-genome data sets into files recording an individual's genetic variations (single nucleotide polymorphisms [SNPs], copy number, etc.) from the reference genome will reduce the data burden a thousand-fold. However, even solving these kinds of bandwidth issues will not address the question of how one analyzes and uses the huge amount of information being generated. To put this in perspective, a single machine can now produce the same amount of data in one week as the Human Genome Project produced in 10 years.

Non-technical Challenges

Some of the most significant challenges facing mainstream genomics are decidedly non-technical in nature. Like many information-based fields, the pace of innovation is outstripping the rate at which legislation and regulation can keep up. Laws designed prior to the genomic revolution are being shoehorned to fit technologies and situations for which there are no clear precedents.

The regulatory landscape must be more clearly defined, so companies can move forward with confidence in leveraging innovations to improve people's lives. Simultaneously, we must raise public awareness of genomic technologies to dispel myths and promote a realistic, more accurate understanding of the importance of genomics to the health of both individuals and society as a whole.

SUMMARY

Genomics has emerged as an important tool for studying biological systems. Significant cost reductions in genomic sequencing have accelerated the adaptation of this technology for applications in a variety of market segments (e.g., research, forensics, consumer products, agriculture, and diagnostics). The most important factor in reducing cost is increasing throughput per day. We predict that the cost of sequencing an entire genome will drop to a few hundred dollars in the next few years as throughput rises with increasing density, longer read length, and shorter cycle time.

In a year or so, the cost of genome sequencing will be less than the cost of single-gene testing, which by itself has already brought significant cost savings to health care. We also predict that genome sequencing will soon become a standard part of medical practice and that in the next 15 years everybody in the Western world will be genome-sequenced.

REFERENCES

Balmain, A., J. Gray, and B. Ponder. 2003. The genetics and genomics of cancer. Nature Genetics 33(Suppl): 238–244.

Bentley, D.R. 2006. Whole-genome re-sequencing. Current Opinion in Genetics and Development 16(6): 545–552.

Bentley, D.R., et al. 2008. Accurate whole human genome sequencing using reversible terminator chemistry. Nature 456(7218): 53–59.

Bibikova, M., D. Talantov, E. Chudin, J. M. Yeakley, J. Chen, D. Doucet, E. Wickham, D. Atkins, D. Barker, M. Chee, Y. Wang, and J.-B. Fan. 2004. Quantitative gene expression profiling in formalin-fixed, paraffin-embedded tissues using universal bead arrays. American Journal of Pathology 165(5): 1799–1807.

Gill, S.R., M. Pop, R.T. DeBoy, P.B. Eckburg, P.J. Turnbaugh, B.S. Samuel, J.I. Gordon, D.A. Relman, C.M. Fraser-Liggett, and K. E. Nelson. 2006. Metagenomic analysis of the human distal gut microbiome. Science 312(5778): 1355–1359.

Jones, P.A., and S.B. Baylin. 2002. The fundamental role of epigenetic events in cancer. Nature Reviews. Genetics 3(6): 415–428.

Kurimoto, K., and M. Saitou. 2010. Single-cell cDNA microarray profiling of complex biological processes of differentiation. Current Opinion in Genetics and Development 20(5): 470–477.

Lengauer, C., K.W. Kinzler, and B. Vogelstein. 1998. Genetic instabilities in human cancers. Nature 396(6712): 643–649.

Lennon, N. 2010. Optimization of Sample Preparation for Next-Generation Sequencing. Presented at the Evolution of Next-Generation Sequencing Conference, September 27–29, 2010, Providence, Rhode Island.

Lewis, F., N.J. Maughan, V. Smith, K. Hillan, and P. Quirke. 2001. Unlocking the archive–gene expression in paraffin-embedded tissue. Journal of Pathology 195(1): 66–71.

Loeb, L.A. 1991. Mutator phenotype may be required for multistage carcinogenesis. Cancer Research 51(12): 3075–3079.

Loeb, L.A. 2001. A mutator phenotype in cancer. Cancer Research 61(8): 3230–3239.

Metzker, M.L. 2010. Sequencing technologies—the next generation. Nature Reviews. Genetics 11(1): 31–46.

Pettersson, E., J. Lundeberg, and A. Ahmadian. 2009. Generations of sequencing technologies. Genomics 93(2): 105–111.

Ronaghi, M., S. Karamohamed, B. Pettersson, M. Uhlén, and P. Nyrén. 1996. Real-time DNA sequencing using detection of pyrophosphate release. Analytical Biochemistry 242(1): 84–89.

Ronaghi, M., M. Uhlén, and P. Nyrén. 1998. A sequencing method based on real-time pyrophosphate. Science 281(5375): 363, 365.

Sanger, F., G.M. Air, B.G. Barrell, N.L. Brown, A.R. Coulson, C.A. Fiddes, C.A. Hutchison, P.M. Slocombe, and M. Smith. 1977a. Nucleotide sequence of bacteriophage phi x174 DNA. Nature 265(5596): 687–695.

Sanger, F., S. Nicklen, and A. R. Coulson. 1977b. DNA sequencing with chain-terminating inhibitors. Proceedings of the National Academy of Sciences of the United States of America 74(12): 5463–5467.

Schweiger, M.R., M. Kerick, B. Timmermann, M.W. Albrecht, T. Borodina, D. Parkhomchuk, K. Zatloukal, and H. Lehrach. 2009. Genome-wide massively parallel sequencing of formaldehyde fixed-paraffin embedded (FFPE) tumor tissues for copy-number- and mutation-analysis. PLoS ONE 4(5): e5548.

Shendure, J., and H. Ji. 2008. Next-generation DNA sequencing. Nature Biotechnology 26(10): 1135–1145.

Stratton, M.R., P.J. Campbell, and P.A. Futreal. 2009. The cancer genome. Nature 458(7239): 719–724.

Taniguchi, K., T. Kajiyama, and H. Kambara. 2009. Quantitative analysis of gene expression in a single cell by QPCR. Nature Methods 6(7): 503–506.

Turnbaugh, P.J., R.E. Ley, M. Hamady, C.M. Fraser-Liggett, R. Knight, and J.I. Gordon. 2007. The human microbiome project. Nature 449(7164): 804–810.

Walker, A., and J. Parkhill. 2008. Single-cell genomics. Nature Reviews Microbiology 6(3): 176–177.

Woyke, T., G. Xie, A. Copeland, J.M. González, C. Han, H. Kiss, J.H. Saw, P. Senin, C. Yang, S. Chatterji, J.-F. Cheng, J.A. Eisen, M.E. Sieracki, and R. Stepanauskas. 2009. Assembling the marine metagenome, one cell at a time. PLoS One 4(4): e5299.

Yeakley, J.M., M. Bibikova, E. Chudin, E. Wickham, J.B. Fan, T. Downs, J. Modder, M. Kostelec, A. Arsanjani, and J. Wang-Rodriguez. 2005. Gene expression profiling in formalin-fixed, paraffin-embedded (ffpe) benign and cancerous prostate tissues using universal bead arrays. Journal of Clinical Oncology 23(16S): 9526.

Engineering Biomimetic Peptides for Targeted Drug Delivery

Efrosini Kokkoli
University of Minnesota

Targeted drug delivery, the ability to deliver a drug to a specific site of disease, is the leading frontier in the pursuit of better strategies for selectively treating diseases with minimal side effects. One promising class of targeted-delivery vehicles are peptide-functionalized nanovectors. Biomimetic peptide-targeting ligands (peptides that mimic cell-binding domains of proteins) can be readily designed to bind selectively to a target (e.g., an adhesion receptor on the surface of a cell) with high affinity and specificity. Even more important, biomimetic peptides are accessible by chemical synthesis and relatively compact compared to antibodies and full proteins.

PEPTIDE-FUNCTIONALIZED LIPOSOMES

Liposomes are the most extensively studied drug-transport systems to date. A number of non-targeted liposome delivery systems are already FDA approved and are being used in a clinical setting. Liposomes range in diameter from approximately 50 nm to microns, although diameters of 100 to 200 nm are often desirable for drug delivery.

“Stealth” liposomes, often referred to as sterically stabilized liposomes, have short polyethylene glycol (PEG) polymer strands attached to a fraction of hydrophilic lipid headgroups. These PEG chains form a polymer brush on the surface of the liposome that, through steric repulsion, resists protein adhesion, and therefore clearance by the reticuloendothelial system (RES). An ongoing area of liposome drug-delivery research involves conjugating ligands, such as peptides, onto “stealth” liposomes to confer active as well as passive targeting capabilities on these drug carriers.

Like liposomes, polymeric drug-delivery vectors also encapsulate their cargo and shield it from degradation and clearance from the body. In addition, many of the same peptide-targeting ligands are being conjugated to polymeric delivery vehicles. However, an inherent conflict of design for most targeted-delivery nanovectors is that the surface is typically coated with a polymer brush to inhibit protein adhesion, and therefore clearance by RES, while at the same time ligands are installed on the surface to promote targeted adhesion.

Peptides have many of the same advantages as targeting ligands: they are small molecules; they can be efficiently chemically synthesized; they can achieve high specificity; and they are easily integrated into liposomes as peptide-amphiphiles (Tu et al., 2004). Peptide ligands can be designed to mimic protein-binding sites, or they can be identified, by phage display and other selection techniques, from large peptide libraries.

Today there are a multitude of peptide ligands for a wide range of target receptors; each of which exhibits different levels of binding specificity and affinity. Liposomes have been functionalized with different peptides: SP5-52, a peptide that binds to tumor vasculature; the basic fibroblast growth factor peptide (bFGFp), which specifically binds to FGFR-expressing cells, such as melanoma, breast cancer, and prostate cancer cells; the pentapeptide YIGSR, derived from the glycoprotein laminin, which has been shown to bind with high affinity to the laminin receptor over-expressed in human tumor cells; the NGR and APRGP peptides, which have been used as potential targeting moieties against tumor vasculature; and others (Pangburn et al., 2009).

A different strategy for delivering therapeutic loads to target cells is to use peptide sequences derived from protein transduction domains (PTDs), also called cell-penetrating peptides (CPPs). PTDs are short peptide sequences that mediate translocation across the cell membrane (Torchilin, 2008). Examples of PTDs include the Antennapedia peptide (Antp), the HIV-TAT (transactivator of transcription) peptide, poly-arginine peptides, and penetratin (Breunig et al., 2008). Cell uptake by PTD peptides appears to bypass the endocytic pathway, and there are different theories about the mechanism of CPP-mediated uptake (Torchilin, 2008).

The TAT peptide derived from HIV-TAT is a frequently used CPP (Torchilin, 2008) that has been conjugated to liposomes to improve the intercellular delivery of therapeutic loads (Kale and Torchilin 2007; Marty et al., 2004; Oba et al., 2007; Torchilin et al., 2003; Tseng et al., 2002). For example, Kale and Torchilin formulated a stealth liposomal delivery system with TAT conjugated on the surface of the particles. The liposomes were delivered to the tumor sites by the EPR effect and lost their PEG coating in the low pH tumor environment thus exposing the underlying TAT peptides, which were then able to mediate transport into the tumor cells (Kale and Torchilin, 2007).

Another class of targeting peptides are fusogenic peptides. The capacity of fusogenic peptides of natural (e.g., N-terminus of hemagglutinin subunit HA-2 of influenza virus) or synthetic (e.g., WEAALAEALAEALAEHLAEALAEALEALAA (GALA) or WEAKLAKALAKALAKHLAKALAKALKACEA (KALA)) origin has been exploited for the endosomal/lysosomal escape of several drug-delivery systems (Li et al., 2004; Plank et al., 1998). Fusogenic peptides assume a random coil structure at pH 7. Acidification triggers a conformational transition that allows their subsequent interaction with the lipid membrane, resulting in pore formation or the induction of membrane fusion and/or lysis (Breunig et al., 2008). The incorporation of synthetic membrane-active peptides into delivery systems has been found to improve the intracellular delivery of drugs, such as oligonucleotides, peptides, and plasmid DNA (Breunig et al., 2008).

Liposomes Functionalized with Collagen-Mimetic Peptides

Collagen-mimetic peptides have been developed to target the CD44 receptor, which is over-expressed in many tumor cells, and nanoparticles functionalized with the collagen-mimetic peptide ligands are endocytosed after the ligand binds to the CD44 receptor (Jiang et al., 2002; Tammi et al., 2001). Specifically, CD44 in metastatic melanoma is in the chondroitin sulfate proteoglycan (CSPG) modified form (Naor et al., 2002). CD44/CSPG receptors bind to a specific amino acid sequence from type IV collagen $\alpha 1(IV)_{1263\text{-}1277}$ (GVKGDKGNPGWPGAP), called IV-H1 (Chelberg et al., 1990; Fields et al., 1993; Lauer-Fields et al., 2003). More important, binding is highly dependant on the triple-helical structure of the sequence and the modified (CSPG) form of CD44 (Fields et al., 1993; Lauer-Fields et al., 2003; Malkar et al., 2002).

The IV-H1 peptide sequence was functionalized with different dialkyl tails to create collagen-like peptide-amphiphiles. Results showed that, although the IV-H1 peptide did not exhibit any positive ellipticity similar to a polyPro II helix, the peptide-amphiphiles investigated were all in triple-helical conformations (Yu et al., 1996). Moreover, the triple-helical peptide-amphiphiles were very stable. In this example, the self-assembly of the hydrophobic tails of the peptide-amphiphiles align the peptide strands and induce and/or stabilize the three-dimensional structure of the peptide headgroup into triple helices, giving rise to protein-like molecular architectures (Figure 1).

Previous studies have shown that a peptide-amphiphile with a peptide headgroup [$(\text{GP-Hyp})_4$-GVKGDKGNPGWPGAP-$(\text{GP-Hyp})_4$-NH_2] mimics the ™$1(IV)_{1263\text{-}1277}$ sequence in structure and function and specifically binds to CD44/CSPG (Lauer-Fields et al., 2003; Yu et al., 1996, 1998, 1999). When this peptide-amphiphile was incorporated into a stealth liposome and targeted to M14#5 metastatic melanoma cells, it promoted specific ligand/receptor interactions. Non-targeted liposomes showed no binding (Rezler et al., 2007).

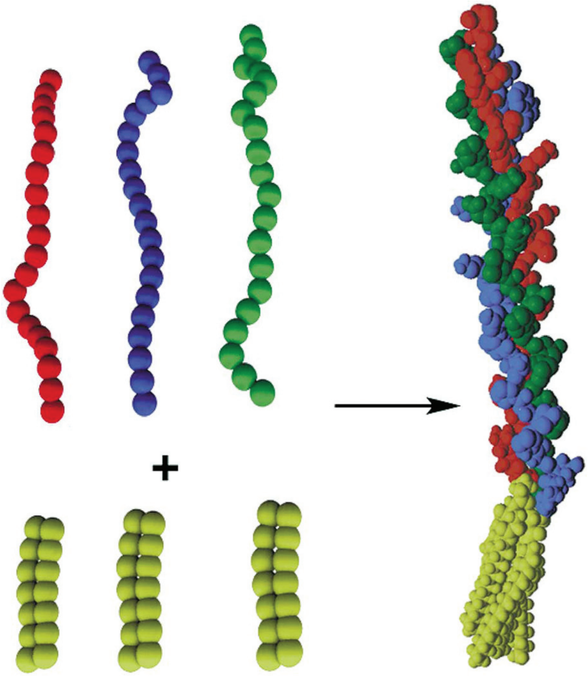


FIGURE 1 Structure of a peptide-amphiphile with triple-helical protein-like molecular architecture. Long-chain dialkyl ester lipid tails (top left) are connected to linear peptide chains (bottom left). The tails associate by hydrophobic interactions, inducing and/or stabilizing the 3-D structure of the peptide headgroup (right). Triple-helical molecular architecture is stabilized in the peptide-amphiphile. Color figure available online at *http://www.nap.edu/catalog.php?record_id=13043*. Source: Tirrell et al., 2002. Reprinted with permission from Elsevier.

PR_b-Functionalized Liposomes

Peptide ligands based on the tripeptide RGD (Arg-Gly-Asp) sequence are widely used in targeting research. The RGD sequence, located in the 10th type III repeat of the fibronectin molecule, which was originally identified as a cell-binding site in the extracellular matrix protein fibronectin, has been used as a targeting moiety on numerous occasions. Although RGD has been used with some success as a targeting moiety against integrins, it does not have the same adhesive properties as native fibronectin (Akiyama et al., 1995; Garcia et al., 2002; Yang et al., 2001).

A synergy amino acid sequence, Pro-His-Ser-Arg-Asn (PHSRN), located in the 9th type III repeat of fibronectin (Figure 2), has been shown to improve bind-

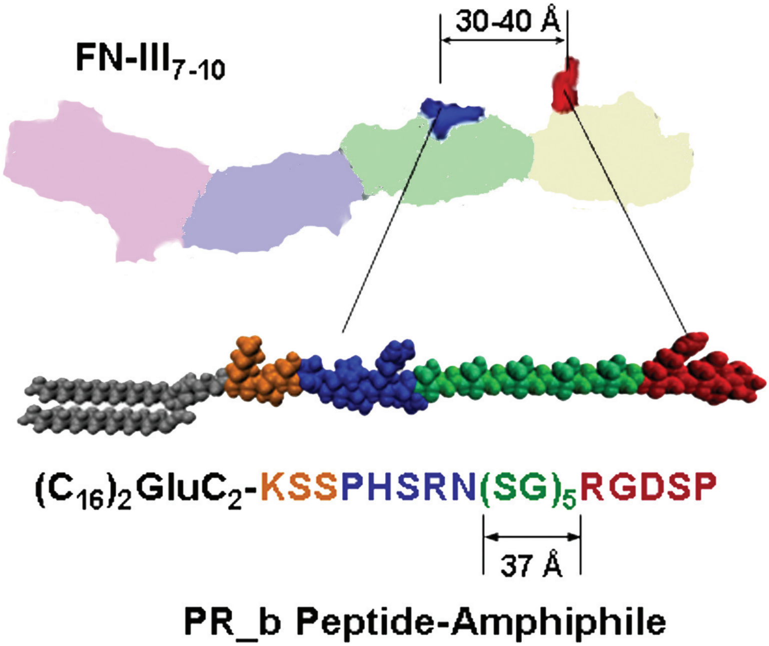


FIGURE 2 The four repeats of the fibronectin (FN) fragment III_{7-10} are shown: far left repeat for III_7 to far right for III_{10}. The synergy site PHSRN is in the III_9 repeat. The GRGDS is in the III_{10}. The schematic drawing of the PR_b peptide-amphiphile shows the four building blocks of the peptide headgroup: a KSS spacer, the PHSRN synergy site, a $(SG)_5$ linker, and the RGDSP binding site. When the PR_b peptide-amphiphile is used for the preparation of functionalized liposomes, the hydrophobic tail is part of the membrane, and the peptide headgroup is exposed at the interface. Color figure available online at *http://www.nap.edu/catalog.php?record_id=13043*.

ing affinity and is critical for specificity to the $\alpha_5\beta_1$ integrin (Aota et al., 1994; Leahy et al., 1996). Although various targeting moieties incorporating both the RGD and PHSRN sequences have been tested, most of these designs did not achieve the cell-adhesion densities supported by native fibronectin over similar time scales (Aucoin et al., 2002; Benoit and Anseth, 2005; Kao, 1999; Kim et al., 2002; Petrie et al., 2006).

Mardilovich and Kokkoli postulated that, for a small peptide to effectively mimic the $\alpha_5\beta_1$ binding site of fibronectin, the primary (RGD) and synergistic (PHSRN) binding sequences must be connected by a linker that approximates both the distance and hydrophobicity/hydrophilicity between the fibronectin sequences, which results in a neutral linker (Mardilovich and Kokkoli, 2004).

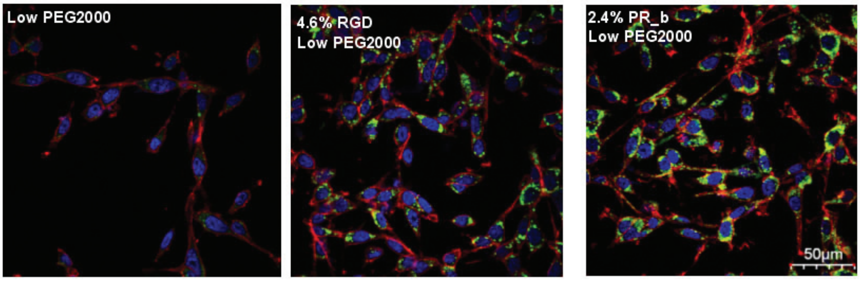


FIGURE 3 Confocal images that show internalization of targeted stealth liposomes to CT26 colon carcinoma cells. The images show the cell membrane, the nucleus, and the drug delivery systems shown between the nuclear region and the cell membrane. Different formulations with low densities of PEG2000 (2–3 percent) were incubated with CT26 at 37°C for 24 hours. The scale bar is 50 μm for all images. Color figure available online at *http://www.nap.edu/catalog.php?record_id=13043*. Source: Adapted from Garg et al., 2009.

Although previous attempts had been made to match the distance between the RGD and PHSRN sequences, they did not pay particular attention to the hydrophilicity/hydrophobicity of the linker.

Mardilovich and Kokkoli's efforts culminated in the design of a biomimetic peptide, named PR_b, which is now well established as a close mimic of the $\alpha_5\beta_1$ binding site in fibronectin and is a highly effective and specific targeting peptide (Mardilovich et al., 2006). PR_b has been shown to bind specifically to the $\alpha_5\beta_1$ integrin and to promote cell adhesion more effectively than similar peptides with hydrophobic or hydrophilic linkers, and even more effectively than fibronectin (Craig et al., 2008). When attached to a 16-carbon dialkyl tail to form a peptide-amphiphile (Figure 2), PR_b can easily be incorporated into a liposome.

Recently, stealth liposomes functionalized with PR_b were used for the targeted delivery of therapeutics to colon cancer cells (Garg and Kokkoli, 2011; Garg et al., 2009) and prostate cancer cells (Demirgöz et al., 2008). In these studies, PR_b-functionalized stealth liposomes loaded with a chemotherapy agent were more effective than RGD-functionalized stealth liposomes or non-targeted stealth liposomes in terms of cell adhesion, internalization, and cancer-cell toxicity (Figure 3).

CONCLUSION

A wide range of peptide-targeting ligands have been studied. The ones most frequently used in a variety of delivery systems, both liposomal and polymeric, are RGD, TAT, NGR, and bFGF. Installing a targeting ligand onto the surface of a delivery nanovector has been shown to have numerous advantages, such as increased cellular adhesion, internalization, and targeting. These advantages have now been confirmed by countless researchers.

REFERENCES

Akiyama, S.K., S. Aota, and K.M. Yamada. 1995. Function and receptor specificity of a minimal 20-kilodalton cell adhesive fragment of fibronectin. Cell Adhesion and Communication 3(1): 13–25.

Aota, S., M. Nomizu, and K. Yamada. 1994. The short amino acid sequence Pro-His-Ser-Arg-Asn in human fibronectin enhances cell-adhesive function. Journal of Biological Chemistry 269(40): 24756–24761.

Aucoin, L., C.M. Griffith, G. Pleizier, Y. Deslandes, and H. Sheardown. 2002. Interactions of corneal epithelial cells and surfaces modified with cell adhesion peptide combinations. Journal of Biomaterials Science. Polymer Edition 13(4): 447–462.

Benoit, D.S.W., and K.S. Anseth. 2005. The effect on osteoblast function of colocalized RGD and PHSRN epitopes on PEG surfaces. Biomaterials 26(25): 5209–5220.

Breunig, M., S. Bauer, and A. Goepferich. 2008. Polymers and nanoparticles: intelligent tools for intracellular targeting? European Journal of Pharmaceutics and Biopharmaceutics 68(1): 112–128.

Chelberg, M.K., J.B. McCarthy, A.P. Skubitz, L.T. Furcht, and E.C. Tsilibary. 1990. Characterization of a synthetic peptide from type IV collagen that promotes melanoma cell adhesion, spreading, and motility. Journal of Cell Biology 111(1): 261–270.

Craig, J.A., E.L. Rexeisen, A. Mardilovich, K. Shroff, and E. Kokkoli. 2008. Effect of linker and spacer on the design of a fibronectin mimetic peptide evaluated via cell studies and AFM adhesion forces. Langmuir 24(18): 10282–10292.

Demirgöz, D., A. Garg, and E. Kokkoli. 2008. PR_b-targeted PEGylated liposomes for prostate cancer therapy. Langmuir 24: 13518–13524.

Fields, C.G., D.J. Mickelson, S.L. Drake, J.B. McCarthy, and G.B. Fields. 1993. Melanoma cell adhesion and spreading activities of a synthetic 124-residue triple-helical “mini-collagen.” Journal of Biological Chemistry 268(19): 14153–14160.

Garcia, A.J., J.E. Schwarzbauer, and D. Boettiger. 2002. Distinct activation states of alpha5beta1 Integrin show differential binding to RGD and synergy domains of fibronectin. Biochemistry 41(29): 9063–9069.

Garg, A., and E. Kokkoli. 2011. pH-Sensitive PEGylated liposomes functionalized with a fibronectin-mimetic peptide show enhanced intracellular delivery to colon cancer cells. Current Pharmaceutical Biotechnology: in press.

Garg, A., A.W. Tisdale, E. Haidari, and E. Kokkoli. 2009. Targeting colon cancer cells using PEGylated liposomes modified with a fibronectin-mimetic peptide. International Journal of Pharmaceutics 366: 201–210.

Jiang, H., R.S. Peterson, W. Wang, E. Bartnik, C.B. Knudson, and W. Knudson. 2002. A requirement for the CD44 cytoplasmic domain for hyaluronan binding, pericellular matrix assembly, and receptor-mediated endocytosis in COS-7 cells. Journal of Biological Chemistry 277(12): 10531–10538.

Kale, A.A., and V.P. Torchilin. 2007. “Smart” drug carriers: PEGylated TATp-Modified pH-sensitive liposomes. Journal of Liposome Research 17(3–4): 197–203.

Kao, W.J. 1999. Evaluation of protein-modulated macrophage behavior on biomaterials: designing biomimetic materials for cellular engineering. Biomaterials 20(23–24): 2213–2221.

Kim, T.I., J.H. Jang, Y.M. Lee, I.C. Ryu, C.P. Chung, S.B. Han, S.M. Choi, and Y. Ku. 2002. Design and biological activity of synthetic oligopeptides with Pro-His-Ser-Arg-Asn (PHSRN) and Arg-Gly-Asp (RGD) motifs for human osteoblast-like cell (MG-63) adhesion. Biotechnology Letters 24(24): 2029–2033.

Lauer-Fields, J.L., N.B. Malkar, G. Richet, K. Drauz, and G.B. Fields. 2003. Melanoma cell CD44 interaction with the alpha 1(IV)1263-1277 region from basement membrane collagen is modulated by ligand glycosylation. Journal of Biological Chemistry 278(16): 14321–14330.

Leahy, D.J., I. Aukhil, and H.P. Erickson. 1996. 2.0 Å crystal structure of a four-domain segment of human fibronectin encompassing the RGD loop and synergy region. Cell 84(1): 155–164.

Li, W., F. Nicol, and J. Szoka. 2004. GALA: a designed synthetic pHresponsive amphipathic peptide with applications in drug and gene delivery. Advanced Drug Delivery Reviews 56: 967–985.

Malkar, N.B., J.L. Lauer-Fields, J.A. Borgia, and G.B. Fields. 2002. Modulation of triple-helical stability and subsequent melanoma cellular responses by single-site substitution of fluoroproline derivatives. Biochemistry 41(19): 6054–6064.

Mardilovich, A., and E. Kokkoli. 2004. Biomimetic peptide-amphiphiles for functional biomaterials: the role of GRGDSP and PHSRN. Biomacromolecules 5(3): 950–957.

Mardilovich, A., J.A. Craig, M.Q. McCammon, A. Garg, and E. Kokkoli. 2006. Design of a novel fibronectin-mimetic peptide-amphiphile for functionalized biomaterials. Langmuir 22(7): 3259–3264.

Marty, C., C. Meylan, H. Schott, K. Ballmer-Hofer, and R.A. Schwendener. 2004. Enhanced heparan sulfate proteoglycan-mediated uptake of cell-penetrating peptide-modified liposomes. Cellular and Molecular Life Sciences 61(14): 1785–1794.

Naor, D., S. Nedvetzki, I. Golan, L. Melnik, and Y. Faitelson. 2002. CD44 in cancer. Critical Reviews in Clinical Laboratory Sciences 39(6): 527–579.

Oba, M., S. Fukushima, N. Kanayama, K. Aoyagi, N. Nishiyama, H. Koyama, and K. Kataoka. 2007. Cyclic RGD peptide-conjugated polyplex micelles as a targetable gene delivery system directed to cells possessing $\alpha_v\beta_3$ and $\alpha_v\beta_5$ integrins. Bioconjugate Chemistry 18(5): 1415–1423.

Pangburn, T.O., M.A. Petersen, B. Waybrant, M.M. Adil, and E. Kokkoli. 2009. Peptide- and aptamer-functionalized nanovectors for targeted delivery of therapeutics. Journal of Biomechanical Engineering 131(7): 074005.

Petrie, T.A., J.R. Capadona, C.D. Reyes, and A.J. Garcia. 2006. Integrin specificity and enhanced cellular activities associated with surfaces presenting a recombinant fibronectin fragment compared to RGD supports. Biomaterials 27(31): 5459–5470.

Plank, C., W. Zauner, and E. Wagner. 1998. Application of membrane-active peptides for drug and gene delivery across cellular membranes. Advanced Drug Delivery Reviews 34: 21–35.

Rezler, E.M., D.R. Khan, J. Lauer-Fields, M. Cudic, D. Baronas-Lowell, and G.B. Fields. 2007. Targeted drug delivery utilizing protein-like molecular architecture. Journal of the American Chemical Society 129(16): 4961–4972.

Tammi, R., K. Rilla, J.-P. Pienimaki, D.K. MacCallum, M. Hogg, M. Luukkonen, V.C. Hascall, and M. Tammi. 2001. Hyaluronan enters keratinocytes by a novel endocytic route for catabolism. Journal of Biological Chemistry 276(37): 35111–35122.

Tirrell, M., E. Kokkoli, and M. Biesalski. 2002. The role of surface science in bioengineered materials. Surface Science 500(1–3): 61–83.

Torchilin, V.P. 2008. Tat peptide-mediated intracellular delivery of pharmaceutical nanocarriers. Advanced Drug Delivery Reviews 60(4–5): 548–558.

Torchilin, V.P., T.S. Levchenko, R. Rammohan, N. Volodina, B. Papahadjopoulos-Sternberg, and G.G.M. D'Souza. 2003. Cell transfection in vitro and in vivo with nontoxic TAT peptide-liposome-DNA complexes. Proceedings of the National Academy of Sciences 100(4): 1972–1977.

Tseng, Y.L., J.J. Liu, and R.L. Hong. 2002. Translocation of liposomes into cancer cells by cell-penetrating peptides penetratin and TAT: a kinetic and efficacy study. Molecular Pharmaceutics 62(4): 864–872.

Tu, R.S., K. Mohanty, and M.V. Tirrell. 2004. Liposomal targeting through peptide-amphiphile functionalization. American Pharmaceutical Review 7(2): 36–41.

Yang, X.B., H.I. Roach, N.M.P. Clarke, S.M. Howdle, R. Quirk, K.M. Shakesheff, and R.O.C. Oreffo. 2001. Human osteoprogenitor growth and differentiation on synthetic biodegradable structures after surface modification. Bone 29(6): 523–531.

Yu, Y.C., P. Berndt, M. Tirrell, and G.B. Fields. 1996. Self-assembling amphiphiles for construction of protein molecular architecture. Journal of the American Chemical Society 118(50): 12515–12520.

Yu, Y.C., V. Roontga, V.A. Daragan, K.H. Mayo, M. Tirrell, and G.B. Fields. 1999. Structure and dynamics of peptide-amphiphiles incorporating triple-helical proteinlike molecular architecture. Biochemistry 38(5): 1659–1668.

Yu, Y.C., M. Tirrell, and G.B. Fields. 1998. Minimal lipidation stabilizes protein-like molecular architecture. Journal of the American Chemical Society 120(39): 9979–9987.

Autonomous Systems and Synthetic Biology

Henry Hess
Columbia University

Autonomous systems have helped solve a variety of engineering challenges, from drastic changes in manufacturing processes since the 1950s to the exploration of space and oceans. Recently, autonomous systems conceived at the micro- and nanoscale levels are being used to address challenges in materials and biomedical applications. These microscopic devices and their applications are inspired by autonomous biological systems that operate both individually and collectively.

A prototypical autonomous biological device is a *Vibrio cholerae* bacterium that packs the ability to move, sense, target, adapt, and release active substances into just a few cubic centimeters. An example of a smart biological material, muscle, which is composed of autonomous systems, is hierarchically assembled from microscopic subunits, has the ability to exchange information with the environment via electrical and mechanical stimuli, and incorporates energy-conversion modules and self-healing abilities.

Both *V. cholerae* bacteria and muscle cells are significant achievements in engineering by evolution. Both can operate in diverse environments with low power consumption and limited computing resources.

These and other examples of microscopic systems with complex functionalities have been the inspiration for the emerging field of synthetic biology. A prominent research strategy in this new field—inspired by the successful large-scale integration of electronic circuits—is to focus on the design of standardized gene circuits that can serve as modules of complex programs to be executed by bacterial cells. This approach is akin to the delivery of a set of well organized blueprints to a contract manufacturer who then manufactures equipment according to the delivered specifications, learns from the experience of technicians to operate the equipment, and produces a product of interest.

A second strategy, which is the topic of this paper, is to develop the technical expertise to rationally design complex, interacting microscopic systems (Schwille and Diez, 2009). This approach builds on nanotechnology, as well as on increasingly complex *in vitro* experiments in cell biology. In these experiments, we replicate critical cellular functions to test our understanding of essential and auxiliary biological mechanisms and components.

The challenges in designing biomimetic systems using nanoscale building blocks include (1) controlling their operation in the presence of Brownian motion and other sources of noise, (2) integrating molecular information properly, (3) addressing lifetime and reliability issues, and (4) anticipating and using emergent phenomena.

KINESIN-POWERED MOLECULAR SHUTTLES

Kinesin motors are proteins that use ATP molecules as fuel to generate mechanical work (Howard, 2001). A microtubule assembled from thousands of tubulin proteins serves as a track for the kinesin motor. For each ATP molecule a kinesin motor hydrolyzes, it takes one step (i.e., moves 8 nanometers [nm]) along the microtubule track. The kinesin motor can advance against a force of about 5 piconewtons (pN), in the process converting more than 50 percent of the free energy of ATP hydrolysis into mechanical work.

Within cells, kinesin is primarily responsible for transporting molecular cargo from the center of the cell to the periphery. Biophysicists have developed the ability to observe and manipulate kinesin motors and the associated microtubule filaments outside the cell in so-called in vitro gliding assays. In these assays, kinesins are adhered to a surface, and fluorescently labeled microtubules are propelled by kinesin motors in the presence of ATP (Figure 1).

In addition to enabling the study of motor proteins, microtubules propelled by kinesin motors serve as a nanoscale transport system. By controlling the direction of the microtubules, the attachment and detachment of cargo to microtubules, and the supply of ATP fuel, microtubules can be induced to act as nanoscale delivery trucks, or "molecular shuttles" (Hess and Vogel, 2001). Assembled from biological components with unmatched functionality, these molecular shuttles can be used to explore design concepts for nanoscale systems and devices (Hess et al., 2002a,b).

Although individual molecular shuttles can be controlled (van den Heuvel et al., 2006), the inherent advantages of molecular devices can be better exploited by putting aside costly efforts to achieve individual control and accepting instead the autonomous operation of molecular shuttles in an externally directed "swarm."

A suitable analogy to this scenario is an anthill. Although in principle it is possible to induce an individual ant to perform a specific task, the anthill, as a complex system, relies on the emergence of useful actions that result from the autonomous

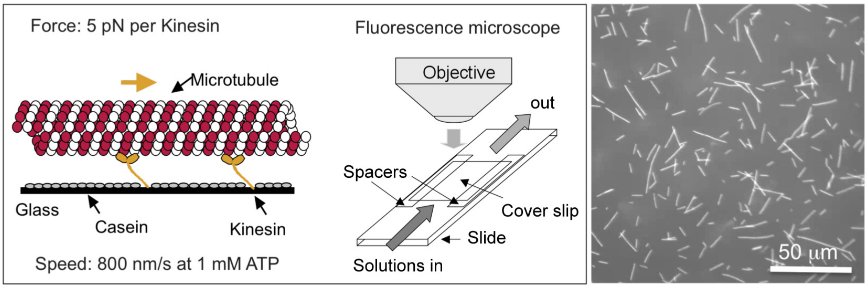


FIGURE 1 Surface-adhered kinesin motors can propel fluorescently labeled microtubules (diameter 25 nm) across a surface. A coating of casein proteins on the surface prevents the kinesin motor domains from attaching to the surface. The experiment is conducted in a cell composed of a cover slip, spacers, and a slide. When microtubule motion is observed with a fluorescence microscope, microtubules appear as fluorescent rods, and the kinesin motors on the surface are invisible.

decisions of individual ants. Figure 2 shows the assembly of "nanospools" from "sticky" microtubules, a striking example of emergence in swarms of molecular shuttles (Hess et al., 2005).

MECHANICAL ENGINEERING AT THE INTERFACE OF BIOPHYSICS AND SUPRAMOLECULAR CHEMISTRY

Precise measurements of nanometer lengths and molecular arrangements and interactions are the building blocks for the rational engineering of molecular shuttles. For example, fluorescence-interference contrast microscopy enables the measurement of the 20 nm "ground clearance" of molecular shuttles (Kerssemakers et al., 2009). Other experiments have determined the distribution of cargo-binding linker molecules on the shuttles and elucidated the complex, glue-like interaction between complementary linker molecules on a shuttle and its cargo (Pincet and Husson, 2005).

The combination of precise spatial information and detailed knowledge of molecular interactions enables us to predict emerging properties, such as an optimal velocity for cargo loading (Agarwal et al., 2009). This optimal velocity is a result of (1) the milliseconds required for the glue-like interaction between biotin and streptavidin to strengthen and (2) the velocity-dependence of the number of attempts to form such bonds.

The importance of such studies is that they transition from the reductionist analysis of molecular processes often practiced in biophysics and supramolecular chemistry to an engineering analysis of emerging phenomena resulting from

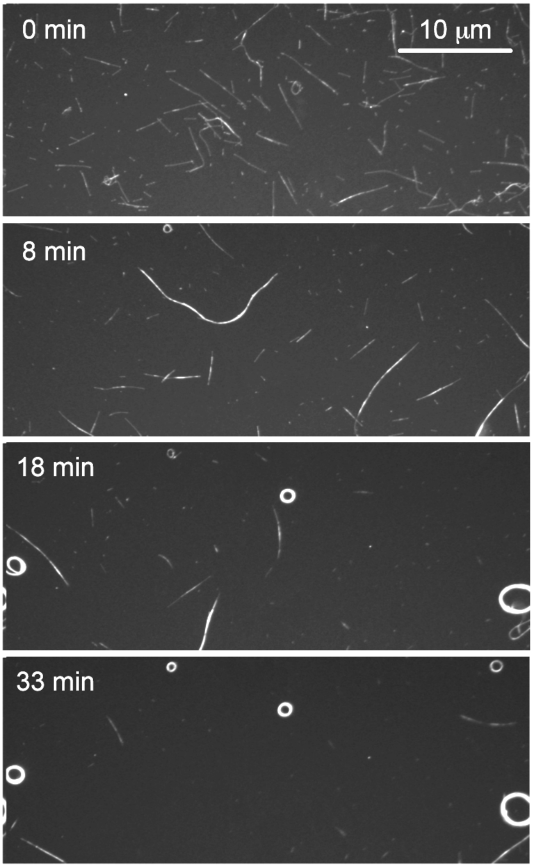


FIGURE 2 Biotin-functionalized microtubules rendered “sticky” by a partial coating of streptavidin self-assemble into nanowires and ultimately into nanospools.

the assembly of well understood building blocks into complex and artificial structures.

The challenge is made more daunting because classic engineering tools, such as technical drawings, have not yet been adapted to capture the dynamic nature of molecular structures. Questions such as whether the fluctuating reach of a linker molecule should be represented by its most likely or it most extended configuration (or both) are very important in a complex technical drawing (Figure 3) showing molecules of different sizes and flexibilities interacting on a wide range of time scales.

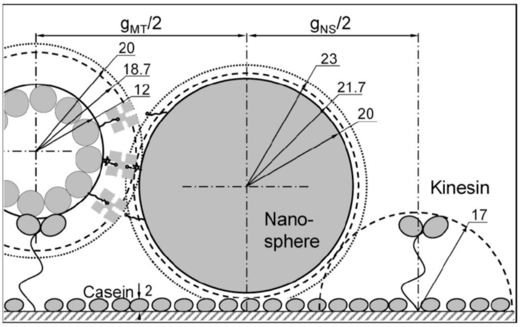


FIGURE 3 Understanding the interaction between a gliding microtubule functionalized with linker molecules (left) and a surface-adhered nanosphere coated with the complementary molecules requires a detailed analysis of the spatial arrangement. Lengths are measured in nanometers, dashed lines represent most likely positions, and dotted lines represent maximal extensions of biotin linkers and kinesin molecules. Adapted from Agarwal et al., 2009.

"SMART DUST" BIOSENSORS AS APPLICATIONS OF MOLECULAR SHUTTLES

Microorganisms, which have the ability to detect a variety of analytes with high specificity and sensitivity, process the incoming information, and communicate their measurements, excel at biosensing. From the perspective of an engineer, microorganisms act as microscopic sensor packages that are immersed into the sample (their environment) and collectively respond to analytes.

This concept is transferred to the engineering domain by "smart dust," which is an attempt to create highly integrated microscopic sensors in large numbers for remote detection scenarios (Kahn et al., 2000; Sailor and Link, 2005). With support from the DARPA Biomolecular Motors Program, five research teams working collaboratively pursued the creation of "smart dust" biosensors, whose core components are molecular shuttles (Bachand et al., 2009). In these sensors, antibody-functionalized molecular shuttles capture analytes, tag them with fluorescent particles, and transport them to a deposition zone for detection (Figure 4).

This design has several "biomimetic" aspects. First, the concept of smart dust per se is inspired by biological organisms. Second, the components of the molecular shuttles—kinesin motors, microtubules, and antibodies—are biological in origin. And third, the principle of operation, a swarm of unsophisticated, autonomous devices creating a detectable signal, is also bio-inspired.

The disadvantages of such a hybrid device are also closely related to its biological origins. Although maintaining the biological components in a relatively

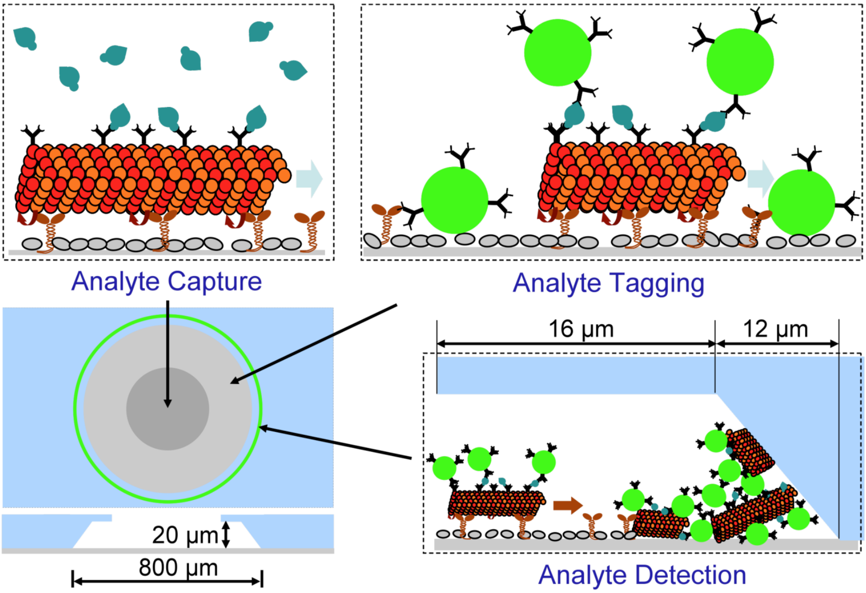


FIGURE 4 A "smart dust" biosensor based on molecular shuttles. Antibody-functionalized microtubules capture analyte molecules, such as protein biomarkers, in the center of a circular well and transport them across the surface. Collisions with antibody-coated fluorescent particles leads to particle capture if analyte is present. Eventually, shuttles reach the periphery of the well, where they accumulate and their cargo of analytes and fluorescent particles can be detected optically.

narrow temperature range at a given pH and with well defined buffer conditions is feasible in the laboratory, it cannot be guaranteed in the field. Therefore, although strategies for extending the lifetime and optimizing storage conditions have been explored (Boal et al., 2006; Seetharam et al., 2006), the fragility of the biological components is a key shortcoming. Successful proof-of-principle demonstrations of application concepts with hybrid devices will therefore require the development of synthetic components, such as synthetic molecular motors (Kay et al., 2007).

However, even when synthetic devices come close to achieving the performance of their natural counterparts, there is still an open question about whether there will be a fundamental trade-off between robustness and performance that will lead to similarly fragile synthetic devices. In addition, we do not know if biomolecular motors represent the pinnacle of achievable performance in terms of force, power, and energy efficiency, or if the use of synthetic materials and fabrication techniques will result in dramatic improvements in one or more metrics.

TRANSITIONING TO SYNTHETIC MATERIALS

In addition to the exciting research and potential applications of hybrid systems mentioned above, the performance and capabilities of synthetic systems are being steadily improved by ongoing efforts in the semiconductor industry to miniaturize energy sources, sensors, computing elements, and communications systems. This technological trend may enable the design of “nanomorphic cells” (Cavin and Zhirnov, 2008). The creation of entirely new classes of autonomous micro-devices with applications, for example, as smart therapeutic systems in medicine would significantly benefit the semiconductor industry.

Although a number of interesting avenues will be explored in the design of nanomorphic cells, my collaborators and I are primarily interested in providing mechanical work for locomotion using chemical fuel sources. The conversion of chemical energy into mechanical work with synthetic nanoscale devices is still in its infancy. The design of complex organic molecules capable of contraction or rotation when provided with fuel molecules has made significant progress but has not yet resulted in designs that can compete with biomolecular motors (Kay et al., 2007).

One successful approach has been to mimic bacterial motility by platinum-gold nanorods in a hydrogen peroxide solution (Hong et al., 2007). The nanorods catalyze the decomposition of the hydrogen peroxide, which in turn propels them forward. Surprisingly, the nanorods have a distinctly “chemotactic” response, moving toward higher hydrogen peroxide concentrations. The analysis of the process has informed our understanding of the mechanism supporting chemotaxis in bacteria.

On the basis of a similar combination of a “compartmentless” fuel cell and electroosmotic pumping, we were able to engineer a fully synthetic membrane that mimics the ability of cellular membranes to actively transport solutes using chemical energy harvested from the solution (Jun and Hess, 2010). A platinum electrode and a gold electrode on the surface of a polycarbonate membrane were electrically connected. When placed in a hydrogen peroxide solution, the fluid was pumped through the membrane (Figure 5).

In a sense, our efforts to create biomimetic functional materials and bio-inspired nanodevices are following the arc of development of human flight. First came the study of biological systems. This was followed by the creation of hybrid systems to elucidate the key principles of flight. Ultimately, synthetic flying machines were developed.

CONCLUSIONS

The engineering of molecular shuttles and other autonomous systems is an exciting aspect of nanotechnology in which progress relies heavily on the use of biological components. The assembly of biological building blocks into newly

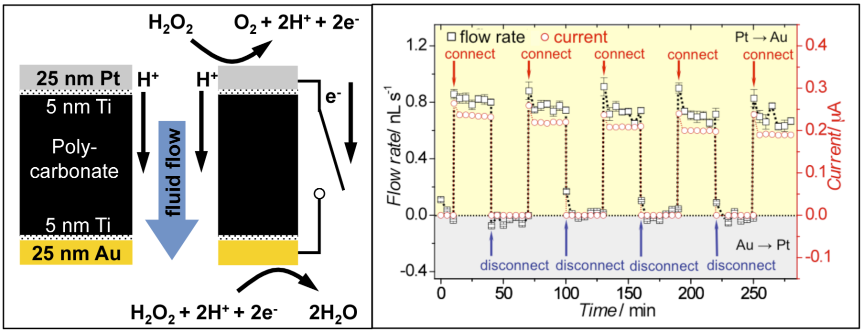


FIGURE 5 A biomimetic, self-pumping membrane (Jun and Hess, 2010). A thin plastic membrane with 1 µm diameter straight channels is coated with a thin film of platinum on one side and a thin film of gold on the other. When the membrane is submerged in a hydrogen peroxide solution and the two electrodes are electrically connected, the system acts as an integrated fuel cell and electroosmotic pump that uses the chemical energy of the hydrogen peroxide to power fluid flow across the membrane.

designed structures is a very stringent test of our assumptions about biological nanomachines and their interactions. As Richard Feynman said, "What I cannot create, I do not understand." The design process forces us to see biology through the eyes of an engineer and, in the process, to ask about friction, wear, fatigue, and other quintessential engineering questions.

REFERENCES

Agarwal, A., P. Katira, and H. Hess. 2009. Millisecond curing time of a molecular adhesive causes velocity-dependent cargo-loading of molecular shuttles. Nano Letters 9(3): 1170–1175.

Bachand, G.D., H. Hess, B. Ratna, P. Satir, and V. Vogel. 2009. "Smart dust'" biosensors powered by biomolecular motors. Lab on a Chip 9(12): 1661–1666.

Boal, A.K., H. Tellez, S.B. Rivera, N.E. Miller, G.D. Bachand, and B.C. Bunker. 2006. The stability and functionality of chemically crosslinked microtubules. Small 2(6): 793–803.

Cavin, R., and V. Zhirnov. 2008. Morphic architectures: atomic-level limits. Materials Research Society Symposium Proceedings 1067(Spring, Symposium B): B1001–B1002.

Hess, H., and V. Vogel. 2001. Molecular shuttles based on motor proteins: active transport in synthetic environments. Reviews in Molecular Biotechnology 82(1): 67–85.

Hess, H., J. Clemmens, J. Howard, and V. Vogel. 2002a. Surface imaging by self-propelled nanoscale probes. Nano Letters 2(2): 113–116.

Hess, H., J. Howard, and V. Vogel. 2002b. A Piconewton forcemeter assembled from microtubules and kinesins. Nano Letters 2(10): 1113–1115.

Hess, H., J. Clemmens, C. Brunner, R. Doot, S. Luna, K.H. Ernst, and V. Vogel. 2005. Molecular self-assembly of "Nanowires" and "Nanospools" using active transport. Nano Letters 5(4): 629–633.

Hong, Y., N.M.K. Blackman, N.Kopp, A. Sen, and D. Velegol. 2007. Chemotaxis of nonbiological colloidal rods. Physical Review Letters 99(17): 178103.

Howard, J. 2001. Mechanics of Motor Proteins and the Cytoskeleton. Sunderland, Mass.: Sinauer.

Jun, I.K., and H. Hess. 2010. A biomimetic, self-pumping membrane. Advanced Materials (in press).

Kahn, J.M., R.H. Katz, Y.H. Katz, and K.S.J. Pister. 2000. Emerging challenges: mobile networking for "smart dust." Journal of Communications and Networks 2(3): 188–196.

Kay, E.R., D.A. Leigh, and F. Zerbetto. 2007. Synthetic molecular motors and mechanical machines. Angewandte Chemie (International Edition) 46(1–2): 72–191.

Kerssemakers, J., L. Ionov, U. Queitsch, S. Luna, H. Hess, and S. Diez. 2009. 3D nanometer tracking of motile micro-tubules on reflective surfaces. Small 5(15): 1732–1737.

Pincet, F., and J. Husson. 2005. The solution to the streptavidin-biotin paradox: the influence of history on the strength of single molecular bonds. Biophysical Journal 89(6): 4374–4381.

Sailor, M.J., and J.R. Link. 2005. "Smart dust": nanostructured devices in a grain of sand. Chemical Communications (11): 1375–1383.

Schwille, P., and S. Diez. 2009. Synthetic biology of minimal systems. Critical Reviews in Biochemistry and Molecular Biology 44(4): 223–242.

Seetharam, R., Y. Wada, S. Ramachandran, H. Hess, and P. Satir. 2006. Long-term storage of bionanodevices by freezing and lyophilization. Lab on a Chip 6(9): 1239–1242.

van den Heuvel, M.G.L., M.P. De Graaff, and C. Dekker. 2006. Molecular sorting by electrical steering of microtubules in kinesin-coated channels. Science 312(5775): 910–914.

APPENDIXES

Contributors

Ella Atkins is an associate professor in the Department of Aerospace Engineering at the University of Michigan, where she is director of the Autonomous Aerospace Systems Lab. She has a joint appointment in the Department of Electrical Engineering and Computer Science. She received a Ph.D. in computer science and engineering from the University of Michigan. Her research focuses on the integration of strategic and tactical planning and optimization algorithms to enable robust operation in the presence of system failures and environmental uncertainties. A second research area is on optimization of and safety analysis in congested airspace.
http://aerospace.engin.umich.edu/people/faculty/atkins/

Luiz Andre Barroso is a distinguished engineer at Google Inc., where he has worked on a number of different areas, including cluster load balancing, finding related academic academic articles, failure analysis, RPC-level networking, server performance optimization, power provisioning, energy efficiency, and the design of Google's computing platform. He received a Ph.D. in computer engineering from the University of Southern California.
http://www.barroso.org/

Stefan Bieniawski is a senior flight sciences research engineer at Boeing Research & Technology. He received a Ph.D. in aeronautics and astronautics from Stanford University. His research is in flight control systems design and analysis, focusing on multi-disciplinary, unconventional, and collaborative control concepts. He is principal investigator for multiple internal and external research programs on advanced guidance and flight controls technologies.

Mark Campbell is an associate professor of mechanical and aerospace engineering at Cornell University. He is also associate director for graduate affairs and director of graduate studies for mechanical engineering. He received a Ph.D. from the Massachusetts Institute of Technology. Dr. Campbell is interested in control and autonomy for systems such as robotics, aircraft, and spacecraft. His research areas include autonomous robotics, human decision modeling, sensor fusion, nonlinear and hybrid estimation theory, integrated estimation and control, formation flying satellites, and structural dynamics and control.
http://www.mae.cornell.edu/index.cfm/page/fac/campbell.htm

Elaine Chew is an associate professor of electrical and industrial and systems engineering, and music, at the University of Southern California and founder and director of the Music Computation and Cognition Lab where she conducts and directs research on music and computing. An operations researcher and pianist by training, her research activities aim to explain and demystify the phenomenon of music and its performance through the use of formal scientific methods. As a performer, she designs and curates concerts featuring interactive scientific music visualizations and collaborates with composers to present eclectic post-tonal music. Dr. Chew received a Ph.D. from the Massachusetts Institute of Technology.
http://www-bcf.usc.edu/~echew

Armando Fox is an adjunct associate professor at the University of California, Berkeley, and a co-founder of the Berkeley Reliable Adaptive Distributed Systems Laboratory. He received a Ph.D. from UC Berkeley. His current research interests include applied statistical machine learning and cloud computing. He is a coauthor of the recently released position paper, “Above the Clouds: A Berkeley View of Cloud Computing,” and has frequently lectured on this topic.
http://www.eecs.berkeley.edu/Faculty/Homepages/fox.html

Chad Frost is supervisor for autonomous systems and robotics in the Intelligent Systems Division at NASA Ames Research Center, where his team develops and deploys technologies that will make aircraft safer, spacecraft more affordable, and robots smarter. He has degrees in aeronautical engineering from California Polytechnic State University.
http://ti.arc.nasa.gov/tech/asr/

Henry Hess is an associate professor in the Department of Biomedical Engineering at Columbia University. He received a Ph.D. in physics from Free University Berlin. His research is in the areas of nanobiotechnology, synthetic biology, and engineering at the molecular scale, in particular the design of active nanosystems incorporating biomolecular motors, the study of active self-assembly, and the investigation of protein-resistant polymer coatings.
http://www.bme.columbia.edu/fac-bios/hess/faculty.html

Efrosini Kokkoli is an associate professor in the Department of Chemical Engineering and Materials Science at the University of Minnesota. She received a Ph.D. in chemical engineering from the University of Illinois at Urbana-Champaign. Her research is in the rational design of novel biomimetic peptide-amphiphiles and aptamer-amphiphiles and their evaluation in different bionanotechnological applications such as targeted delivery of therapeutics and functionalized biomaterials for tissue engineering.
http://www.cems.umn.edu/about/people/faculty.php?id=20282

Bernard Meyerson is vice president for innovation at IBM and leads their Global University Relations function. He is also responsible for the IBM Academy, a self-governed organization of about 1,000 executives and senior technical leaders from across IBM. He is also a member of the CEO's Integration and Values Team, the senior executive group integrating the business activities of IBM's many disparate organizations and geographies. In 1992, Dr. Meyerson was appointed that year's sole IBM Fellow, IBM's highest technical honor, by IBM's chairman. Dr. Meyerson is a Fellow of the American Physical Society and IEEE and is a member of the National Academy of Engineering. He has received numerous technical and business awards for his work, which includethe Materials Research Society Medal, the Electrochemical Society Electronics Division Award, the IEEE Ernst Weber Award, the Electron Devices Society J. J. Ebers Award, and most recently the 2007 Lifetime Achievement Award from SEMI.

Parthasarathy Ranganathan is a distinguished technologist at Hewlett Packard Labs. He received a Ph.D. from Rice University. His research interests are in systems architecture and management, power management and energy-efficiency, and systems modeling and evaluation. He is currently the principal investigator for the exascale datacenter project at HP Labs that seeks to design and manage next-generation servers and datacenters.
http://www.hpl.hp.com/personal/Partha_Ranganathan/

Douglas Repetto is director of research at the Columbia University Computer Music Center. His work, including sculpture, installation, performance, recordings, and software, is presented internationally. He is the founder of a number of art/community-oriented groups including *dorkbot: people doing strange things with electricity*, *ArtBots: The Robot Talent Show*, *organism: making art with living systems*, and the *music-dsp* mailing list and website.
http://music.columbia.edu/~douglas

Mostafa Ronaghi is senior vice president and chief technology officer at Illumina, where he is responsible for leading internal research programs and evaluating new technologies for the company, which applies innovative technologies to the analysis of genetic variation and function. He has founded four life science companies:

Avantome, NexBio, ParAllele Bioscience, and Pyrosequencing AB. He received a Ph.D. from the Royal Institute of Technology in Sweden.
http://investor.illumina.com/phoenix.zhtml?c=121127&p=irol-govBio&ID=181235

Daniel Trueman is an associate professor of music at Princeton University as well as a composer and performer. He is co-founder and director of the Princeton Laptop Orchestra, an ensemble of laptop-ists with 6-channel spherical speakers and various control devices. He has been active as an experimental instrument designer and has built spherical speakers and the Bowed-Sensor-Speaker Array, among other things. He studied physics at Carleton College, composition and theory at the College Conservatory of Music in Cincinnati, and composition at Princeton University.
http://silvertone.princeton.edu/~dan/

Brian Whitman is co-founder and chief technical officer of The Echo Nest Corporation, a music intelligence company that automatically reads about and listens to the entire world of music for developers to build search, personalization, and interactive music applications. His research links community knowledge of music to its acoustic properties to learn the meaning of music. He received a Ph.D. from the Massachusetts Institute of Technology.
www.the.echonest.com

Yuanyuan Zhou is Qualcomm Chair Professor in the Department of Computer Science and Engineering at the University of California, San Diego and co-founder and CTO of Patterninsight, which was spun off from her research group. She received a Ph.D. from Princeton University. Her research focuses on next-generation computing issues, including energy and thermal management for datacenters, software dependability, and storage systems.
http://cseweb.ucsd.edu/~yyzhou

Program

NATIONAL ACADEMY OF ENGINEERING

2010 U.S. Frontiers of Engineering Symposium
September 23–25, 2010

Chair: Andrew M. Weiner, Purdue University

CLOUD COMPUTING

Organizers: Ali Butt and Dilma Da Silva

Opportunities and Challenges of Cloud Computing
Armando Fox

Warehouse-Scale Computing: The Machinery That Runs the Cloud
Luiz André Barroso

Developing Robust Cloud Applications
Yuanyuan (YY) Zhou

Green Clouds: The Next Frontier
Parthasarathy Ranganathan

***

ENGINEERING AND MUSIC

Organizers: Daniel Ellis and Youngmoo Kim

Very Large Scale Music Understanding
Brian Whitman

Doing It Wrong
Douglas Repetto

Digital Instrument Building and the Laptop Orchestra
Daniel Trueman

Demystifying Music and Its Performance
Elaine Chew

***

AUTONOMOUS AEROSPACE SYSTEMS

Organizers: Michel Ingham and Jack Langelaan

Intelligent Autonomy in Robotic Systems
Mark Campbell

Challenges and Opportunities for Autonomous Systems in Space
Chad R. Frost

Health Awareness in Systems of Multiple Autonomous Aerospace Vehicles
Stefan Bieniawski

Certifiable Autonomous Flight Management for Unmanned Aircraft Systems
Ella Atkins

***

ENGINEERING INSPIRED BY BIOLOGY

Introduction: Mark Byrne and Babak Parviz

The Current Status and Future Outlook for Genomic Technologies
Mostafa Ronaghi

Engineering Biomimetic Peptides for Targeted Drug Delivery
Efrosini Kokkoli

Autonomous Systems and Synthetic Biology
Henry Hess

***

DINNER SPEECH

Radical Innovation to Create a Smarter Planet
Bernard S. Meyerson

Participants

NATIONAL ACADEMY OF ENGINEERING

2010 U.S. Frontiers of Engineering Symposium
September 23–25, 2010

Charles Alpert
Research Scientist
Design Productivity Group, Systems
IBM

Ana Arias
Manager, Printed Electronic Devices Area
Electronic Materials and Devices Laboratory
Palo Alto Research Center, PARC

Ella Atkins**
Associate Professor
Department of Aerospace Engineering
University of Michigan

Debra Auguste
Assistant Professor of Biomedical Engineering
School of Engineering and Applied Sciences
Harvard University

Seth Bank
Assistant Professor
Department of Electrical and Computer Engineering
University of Texas at Austin

Luiz Andre Barroso**
Distinguished Engineer
Google

Stephane Bazzana
Research Manager, Biofuels Process Development
Biochemical Science and Engineering
DuPont Company

*Organizing Committee
**Speaker

Adam Berenzweig
Senior Software Engineer
Systems and Infrastructure
Google

Stefan Bieniawski**
Senior Flight Sciences Research Engineer
The Boeing Company

Brad Boyce
Principal Member of the Technical Staff
Multiscale Metallurgical Science and Technology
Sandia National Laboratories

David Boyd
Senior Research Fellow
Division of Engineering and Applied Science
California Institute of Technology

Jonathan Butcher
Assistant Professor
Department of Biomedical Engineering
Cornell University

Ali Butt*
Assistant Professor
Department of Computer Science
Virginia Tech

Mark Byrne*
Mary & John H. Sanders Associate Professor
Department of Chemical Engineering
Auburn University

Mark Campbell**
Associate Professor
Sibley School of Mechanical and Aerospace Engineering
Cornell University

Erick Cantu-Paz
Principal Scientist
Advertising Sciences
Yahoo! Labs

William Carter
Research Department Manager
Bio and Nanomaterials Technologies
HRL Laboratories

Robert Cassoni
Principal Engineer
Packaging Development R&D
Procter & Gamble Company

Jennifer Cha
Assistant Professor
Department of Nanoengineering
University of California, San Diego

William Chappell
Associate Professor
Department of Electrical and Computer Engineering
Purdue University

Elaine Chew**
Associate Professor
Electrical Engineering/Industrial and Systems Engineering
University of Southern California

Jerry Couretas
Technical Fellow and Senior Manager
Corporate Engineering and Technology
Lockheed Martin Corporation

Terence Critchlow
Chief Scientist
Computational Sciences and Mathematics
Pacific Northwest National Laboratory

Dilma da Silva*
Research Staff Member
Advanced Operating Systems Group
IBM T.J. Watson Research Center

Claus Daniel
Lead, Energy Storage Science and Technologies
Material Science and Technology Division/Physical Sciences
Oak Ridge National Laboratory

Eric Dashofy
Senior Member of the Technical Staff
Computer Systems Research Department
The Aerospace Corporation

Francis de los Reyes
Associate Professor
Department of Civil, Construction, and Environmental Engineering
North Carolina State University

Peter DiMaggio
Principal
Structural Engineering
Weidlinger Associates

Christopher Eckett
Deputy Program Leader
Pratt & Whitney Program Office
United Technologies Research Center

Hany Eitouni
Co-Founder and Director of Materials Development
Materials Department
Seeo

Daniel Ellis*
Associate Professor
Department of Electrical Engineering
Columbia University

Nicholas Fang
Assistant Professor
Department of Mechanical Science and Engineering
University of Illinois at Urbana-Champaign

Kevin Farinholt
R&D Engineer
Applied Engineering and Technology
Los Alamos National Laboratory

Andrew Fikes
Principal Software Engineer
Systems and Infrastructure
Google

Armando Fox**
Adjunct Associate Professor
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

Chad Frost**
Supervisor, Autonomous Systems and Robotics
Intelligent Systems Division
NASA Ames Research Center

Kevin Fu
Assistant Professor
Department of Computer Science
University of Massachusetts, Amherst

Di Gao
Assistant Professor
Department of Chemical and
Petroleum Engineering
University of Pittsburgh

Paul Gloeckner
Technical Advisor
Experimental Mechanics
Cummins Inc.

J. Michael Gray
Principal Mechanical Design Engineer
Pump Development
Medtronic

Mariah Hahn
Assistant Professor
Department of Chemical Engineering
Texas A&M University

A. John Hart
Assistant Professor
Department of Mechanical Engineering
and School of Art and Design
University of Michigan

Henry Hess**
Associate Professor
Department of Biomedical Engineering
Columbia University

Dean Ho
Assistant Professor
Departments of Mechanical and
Biological Engineering
Northwestern University

Dennis Hong
Associate Professor
Department of Mechanical Engineering
Virginia Tech

Gregory Huff
Assistant Professor
Department of Electrical and
Computer Engineering
Texas A&M University

Hillery Hunter
Research Staff Member
Exploratory System Architecture
IBM

Michel Ingham*
Technical Group Supervisor
Flight Software Systems Engineering
and Architectures Group
Jet Propulsion Laboratory

Nebojsa Jojic
Senior Researcher
Microsoft Research
Microsoft

Youngmoo Kim*
Assistant Professor
Department of Electrical and
Computer Engineering
Drexel University

Frederick Kish
Senior Vice-President
Optical Integrated Circuit Group
Infinera Corporation

John Kitching
Group Leader
Physics Laboratory
National Institute of Standards and
Technology

Jeffrey Kloosterman
Senior Principal Engineer
Energy Technology R&D
Air Products and Chemicals

Efrosini Kokkoli**
Associate Professor
Department of Chemical Engineering and Materials Science
University of Minnesota

Swaminathan Krishnan
Assistant Professor of Civil Engineeering and Geophysics
Division of Engineering and Applied Science
California Institute of Technology

Kevin Krizek
Associate Professor
Department of Civil Engineering and Planning and Design
University of Colorado, Denver

Sanjay Kumar
Assistant Professor
Department of Bioengineering
University of California, Berkeley

Aleksandar Kuzmanovic
Lisa Wissner Slivka and Benjamin Slivka Chair in Computer Science and Assistant Professor
Department of Electrical Engineering and Computer Science
Northwestern University

Diana Lados
Assistant Professor and Director, Integrative Materials Design Center
Department of Mechanical Engineering
Worcester Polytechnic Institute

Balasubramanian Lakshmanan
Lab Group Manager, Electrochemical Energy Research Lab
Global R&D
General Motors Company

Jacob Langelaan*
Assistant Professor
Department of Aerospace Engineering
Pennsylvania State University

Zhiqun Lin
Associate Professor
Department of Materials Science and Engineering
Iowa State University

Sergio Loureiro
Director, Materials Analysis, Mechanics & Processing
Materials & Processes Engineering
Pratt & Whitney

Ravi Madduri
Principal Software Development Specialist
Mathematics and Computer Science Division
Argonne National Laboratory

Leigh McCue-Weil
Assistant Professor
Department of Aerospace and Ocean Engineering
Virginia Tech

Scott McLaughlin
Vice President, Radar Engineering
Meteorological Systems Group
DeTect

Adrienne Menniti
Senior Wastewater Engineer
Water Business Group
CH2M HILL

Jeffrey Norris
Manager, Planning and Execution Systems
Systems and Software Division
Jet Propulsion Laboratory

Michele Ostraat
Senior Director, Center for Aerosol Technology
Engineering and Technology Unit
RTI International

Tomas Palacios
Emmanuel Landsman Associate Professor of Electronics
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

Edward Park
Chief Technology Officer
athenahealth

Babak Parviz*
Associate Professor
Department of Electrical Engineering
University of Washington

Annie Pearce
Assistant Professor
Myers-Lawson School of Construction
Virginia Tech

Ronald Polcawich
Team Lead, RF MEMS and mm-Scale Robotics
Army Research Laboratory

Shriram Ramanathan
Assistant Professor of Materials Science
School of Engineering and Applied Sciences
Harvard University

Parthasarathy Ranganathan**
Distinguished Technologist
Exascale Computing Lab
Hewlett Packard Research Labs

Venkatesh Rao
Entrepreneur-in-Residence
Scalable Computing and Web Technologies Laboratory
Xerox Research Center, Webster

Salil Rege
Senior Engineer
Process Solutions Technology Development Center
Cargill

Stanley Rendon
Research Specialist
Corporate Research Process Laboratory
3M Company

Douglas Repetto**
Director of Research
Computer Music Center
Columbia University

Daniel Ripin
Assistant Group Leader
Laser Technology & Applications Group
MIT Lincoln Laboratory

Justin Romberg
Assistant Professor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Mostafa Ronaghi**
Chief Technolgy Officer
Illumina

Klint Rose
Lead Research Engineer
Engineering Technology Division
Lawrence Livermore National Laboratory

Stergios Roumeliotis
Associate Professor
Department of Computer Science and Engineering
University of Minnesota

John Russell
Program Manager, Defense-Wide Manufacturing Science and Technology
Materials and Manufacturing Directorate
Air Force Research Laboratory

John Santini, Jr.
President & CEO
On Demand Therapeutics

Carolyn Seepersad
Assistant Professor
Department of Mechanical Engineering
University of Texas at Austin

Robert Sever
Manager
Biopharmaceuticals R&D
Praxair

David Sholl
Michael Tennenbaum Family Chair and GRA Eminent Scholar
School of Chemical and Biomolecular Engineering
Georgia Institute of Technology

Michael Siemer
President
Mydea Technologies

Joseph Sinfield
Assistant Professor
School of Civil Engineering
Purdue University

Ryan Starkey
Assistant Professor
Department of Aerospace Engineering Sciences
University of Colorado at Boulder

Desney Tan
Senior Researcher
Computational User Experiences
Microsoft Research

Yi Tang
Associate Professor
Department of Chemical and Biomolecular Engineering
University of California, Los Angeles

Seth Taylor
Laboratory Lead
Aerospace Research Laboratories
Northrop Grumman Aerospace Systems

Daniel Trueman**
Associate Professor
Department of Music
Princeton University

Srinivas Tummala
Principal Scientist
Process R&D Chemistry Technologies
Bristol-Myers Squibb Company

Greg VanWiggeren
Project Manager
Measurement and Sensors
Agilent Technologies

Jing Wan
Unconventional Resources Supervisor
Reservoir Engineering
ExxonMobil Development Company

Andrew Weiner*
Scifres Family Distinguished Professor of Electrical and Computer Engineering
School of Electrical and Computer Engineering
Purdue University

Sharon Weiss
Assistant Professor
Department of Electrical Engineering and Computer Science
Vanderbilt University

Thomas Wettergren
Senior Research Scientist
Staff of the Chief Technology Officer
Department of the Navy, Naval Undersea Warfare Center

Brian Whitman**
Co-Founder and CTO
The Echo Nest Corporation

Michael Williams
Lead Engineer, Artificial Intelligence Research and Development
Boeing Research and Technology

Siavash Yazdanfar
Senior Scientist
Applied Optics Laboratory
GE Global Research

Jun Ye
President and Chief Technology Officer
Brion Technologies

Zhiqiang (John) Zhai
Associate Professor
Department of Civil, Environmental, and Architectural Engineering
University of Colorado at Boulder

Yuanyuan Zhou**
Professor
Department of Computer Science and Engineering
University of California, San Diego

Dinner Speaker

Bernard Meyerson
Vice President for Innovation
IBM

Guests

Laura Adolfie
Director
STEM Development Office
Office of the Director, Defense Research and Engineering

Debbie Chachra
Associate Professor of Materials Science
Franklin W. Olin College of Engineering

William Hayden
Vice President
The Grainger Foundation

Todd Hylton
Program Manager
DARPA/DSO

Sohi Rastegar
Division Director
Office of Emerging Frontiers in
Research and Innovation
Directorate of Engineering
National Science Foundation

Phil Szuromi
Senior Editor
Science Magazine

IBM

Mark Dean
Vice President, Technical Strategy
and World Wide Operations

Tze-Chiang (T.C.) Chen
Vice President, Science and
Technology

Jia Chen
Program Director for Innovation

Lisa Kaiser
Staff

National Academy of Engineering

Lance Davis
Executive Officer

Janet Hunziker
Senior Program Officer

Elizabeth Weitzmann
Program Associate