PACEM: Cooperative Control for Citywide Energy Management

Jacob Beal, and Hal Abelson

August 22, 2008

Abstract

Proto/Amorphous Cooperative Energy Management (PACEM) controls energy demand across an entire metropolitan area by coordinating the behavior of individual electrical appliances. The goal of this project is to demonstrate the feasibility of PACEM by showing smooth demand shaping and additive scaling using cheap, decentralized communication hardware, both in simulation and on a test system of 50-100 devices. PACEM's fine-grained control, taking each customer's preferences and needs into account, rests on a foundation of novel algorithms, system designs, and compilation technologies coming from our previous work in spatial computing. This foundation allows us to create distributed programs that run robustly even on a rapidly changing population of millions of devices. This new technology will allow PACEM to be deployed incrementally without any changes to the existing power grid, with positive incentives for both utility companies and customers to participate. PACEM has the potential to smooth out peak consumption, decrease the frequency of power failures and brown-outs, and increase participation in conservation efforts, yet still create virtually no inconvenience for any customer.

1 Project Description

Imagine this: it is a hot summer day and you are sitting at home when your phone rings: "Power company calling. Energy consumption is high right now: would you mind turning off your air conditioner?"

You reply, "What if instead I raise the temperature a few degrees, turn off some lights, and read a book rather than watching TV?"

"We really need less ACs. But if you do that and you and your neighbor take turns with AC, it's OK."

"Sure, I know she'll cooperate. I'll turn mine off this hour, and she'll turn hers of the next."

"Thanks! I'll let you know when you can run it freely again, and we'll cut your next bill a little."

A person's energy needs are often fairly flexible: the temperature could change a few degrees, the laundry get done later, and that light be turned off. If customers and utilities were constantly negotiating with one another, they could use this flexibility to shape energy demands, smoothing peak consumption and making failures and brown-outs less likely.

People will not do this. You don't want to sit and wait for calls from your utility, and they don't want to handle millions of negotiations. Most people interact with their utility once a month through their bill, and almost never with their neighbors.

But what if your appliances knew your preferences and negotiated for you? What if they were also making deals with your neighbors ("I'll run my AC this hour, you run yours the next") and people in distant parts of the city ("I'll do my laundry later so that the hospital can keep its AC running on full.")?

This is the vision for Proto/Amorphous Cooperative Energy Management (PACEM). Building on our Proto language for spatial computing, we propose to prototype a distributed energy management system in which millions of appliances across a metropolitan area cooperatively shape energy demand on a time scale of minute to hours in response to requests from utilities, yet still take the preferences and needs of every individual customer into account.

Advantages of PACEM The powerful advantages of our approach make it likely that PACEM can be widely deployed within five years:

- *Painless for customers:* Customers express flexibility during installation, in exchange for a discount. Once installed, it is largely invisible.
- *High-speed shaping of energy demand:* Utilities control customer behavior to smooth out peaks in demand and reduce the likelihood of power failures and brown-outs.

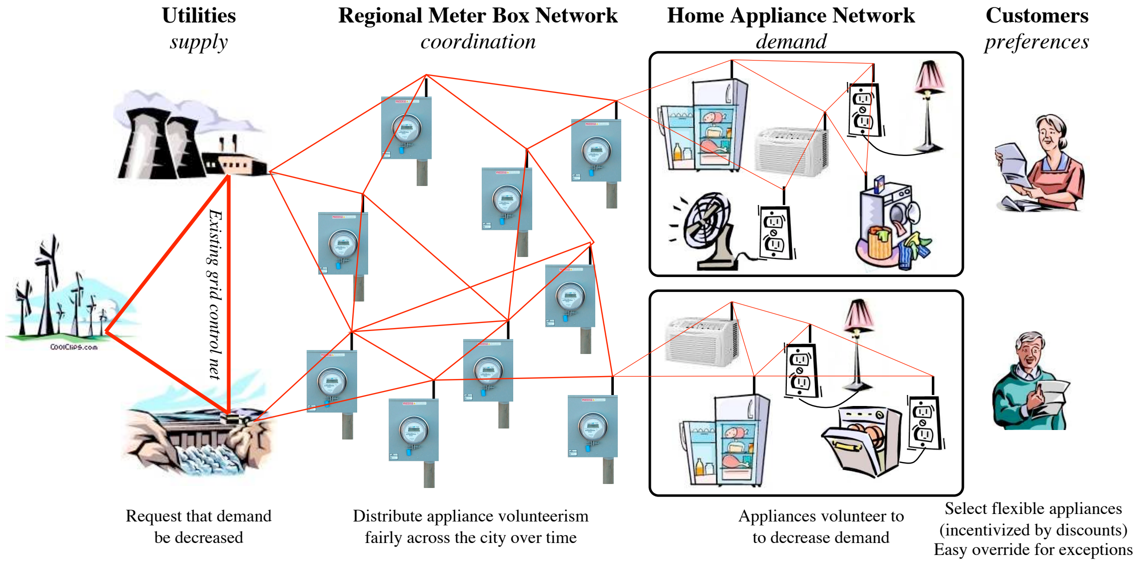


Figure 1: Overview of PACEM: utilities supply power and request decreases in demand. Customers specify their flexibility in exchange for lower energy prices. In each home, smart appliances and outlets communicate to decide which will volunteer to decrease demand. Homes connect to a wireless mesh network of devices mounted on meter boxes, which manages the overall demand and distributes volunteerism fairly.

- *Immediate benefits:* Per-user benefits should be non-zero and additive: twice as many users means at twice the control over demand.
- *Robust scaling:* Proto allows robust decentralized control that scales seamlessly from a neighborhood to a whole city and adapts to changes and failures without human intervention.

PACEM may have a large impact in several areas:

- *Increased grid robustness:* Shaping customer demand could help avoid failures and brown-outs.
- *Management of insufficient supply:* When supply is insufficient or unreliable, PACEM can prioritize critical customers (e.g. hospitals) and enforce community decisions about energy sharing.
- *Automated Conservation:* Customer energy decisions move from operational ("Should I turn on this light?") to institutional ("How important is the light to me?") where long-term thinking is easier (c.f. [16]). If a customer decides to conserve, it becomes their default behavior, rather than something requiring frequent thought.
- *Increased Energy Awareness:* Information flowing through the PACEM system can be exposed to interested customers, giving them a better understanding of how energy is being consumed in their house, their neighborhood, and their city. Increased awareness alone has been shown to cause up to 15% reduction in energy consumption[8]

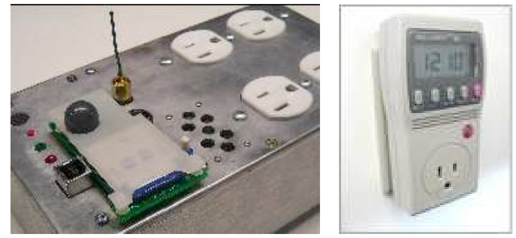
Less immediately, PACEM might also help implement other energy projects requiring large-scale coordination, such as decentralized generation.

1.1 System Overview

We envision a network that automatically matches a utility's requests for demand regulation with ways customers are willing to decrease demand (Figure 1). The utility sends supply data and regulation requests to a wireless mesh network of nodes hooked into electric meter boxes. Each meter box talks to a network of smart outlets and appliances in the household or business that it regulates. Customers encode how they are willing to decrease demand by plugging appliances into smart outlets and in the settings of smart appliances.

Deployment and Maintenance While this proposal is limited to technological work, here is one scenario for how PACEM could be deployed.

Figure 2: A possible "smart outlet" adapter to plug over an existing outlet. A three-way switch sets willingness to be shut off, which the "Not now!" button overrides. The outlet displays its mode with a multicolor LED, and communicates wirelessly with other outlets, appliances, and its meter box.



(a) ResEnv Plug node[10] (b) Kill-a-Watt

Figure 3: Some existing monitoring outlets.

Customers opt into PACEM to receive a discount on energy prices. When a customer joins, the utility adds a node to their meter box (which automatically integrates with the citywide network) and provides them with smart outlets (Figure 2) that plug over existing outlets, much like existing energy monitors (Figure 3). If deployment becomes widespread, large appliances like refrigerators may come with PACEM built in. The outlets form an ad hoc wireless network keyed to talk only with one another and their meter box.

Customers set their demand flexibility by choosing what to plug into smart outlets and setting which outlets regulate first. Once plugged in, the network needs no supervision and is effectively invisible.

Maintenance is simplified by the use of Proto (see below), which means there are no critical central nodes and that the network and running programs adapt smoothly to the addition and removal of nodes. The critical contractual link between utility and customer is the meter box node, which can be maintained and monitored for security along with the electric meter.

Demand Management When the utility company requests a decrease in demand, the meter box network spreads that request across the metropolitan area, distributes it into home networks, ensuring that the flexibility demanded of each customer is as small as possible and distributed fairly over time.

Within the home appliance network, smart outlets and appliances volunteer to reduce demand for a period of hours, prioritizing according to the user's preferences and subject to a one-hour override if the customer hits a "Not now!" button. Demand reductions accumulate slowly, to avoid overcorrection and the potential for oscillation. To reward customers for flexibility, their bill is decreased based on the amount that their appliances volunteer.

Different appliances have different tolerances: for example, lights and electric heaters can be abruptly switched off, but that treatment might damage the compressor of a refrigerator. Thus, there will need to be special smart outlets for a few different categories of appliance. A small amount of integration with an appliance can also greatly improve the efficacy of demand reduction: computers can be switched into low-power mode, heating and cooling systems desynchronized and their goals shifted by a few degrees, dryer cycles postponed, etc. This may be achieved first through accessories for the smart outlets and eventually by building PACEM into appliances.

Unresolved Questions These questions, while important, are beyond the scope of this first technological feasibility proposal:

- The initial vision of PACEM is for homes and small businesses. What is needed to allow it to apply to large apartment or office buildings?
- What incentive design will best cause both utilities and customers to adopt PACEM?
- Would it be better to communicate using the electrical wires rather than wireless?
- What definition of "fair distribution" is best?
- What is the break-even point between decreased demand and the energy used to run PACEM, and how minimal can the power demands of PACEM be?
- What is the best strategy for mitigating the security risks introduced by intelligent control?

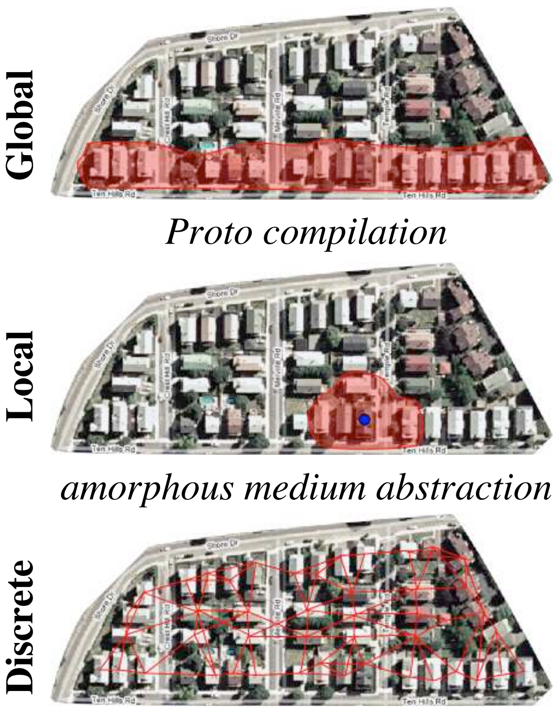
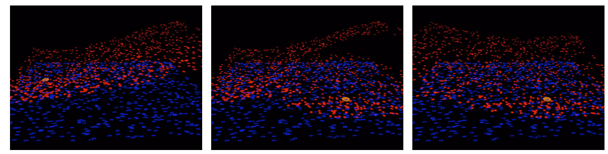


Figure 4: Proto separates spatial computing tasks into three layers: global, local, and discrete. Programs at the global level describe regions of space (e.g. “houses on Puritan Road”). They execute locally on continuous neighborhoods of an amorphous medium, which is approximated on the actual discrete network.

Related Projects PACEM draws inspiration from other, smaller scale energy demand regulation projects. One such example is Hewlett-Packard’s “Smart Cooling” project, which includes temperature control through the spatial distribution of processes[17]. Another is the market-based time-shifting of refrigerator cooling decisions envisioned by Ogston et al.[15] We have avoided their market-based approach, however, due to the tendency of markets to produce unexpected emergent behaviors.

1.2 Enabling Technology

The PACEM vision is enabled by our work in spatial computing and amorphous computing[1], particularly the language Proto. A spatial computer is a network of devices distributed through space such that the ability of devices to communicate depends strongly on their proximity; examples include sensor networks, robotic swarms, and cells during morphogenesis.



(a) T=0 (b) T=11 (c) T=74

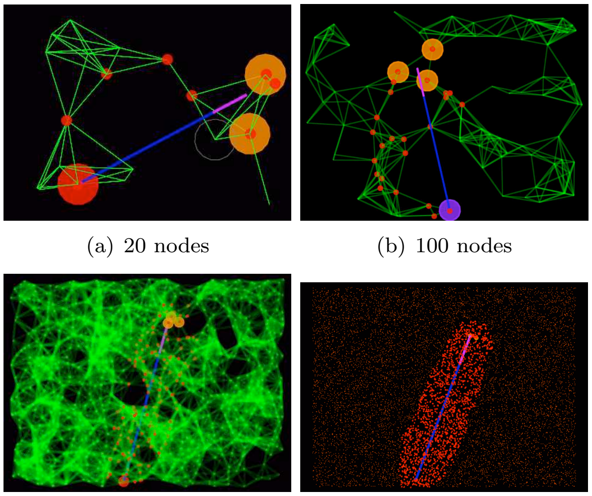
Figure 5: A self-healing gradient reconfigures in response to a change of source, running in simulation on a network of 1000 devices. The network is viewed at an angle, with the value shown as the height of the red dot above the device (blue).

Proto is a high level language in which programs are described in terms of continuous regions of space and time, rather than individual devices. Proto depends on the *amorphous medium* abstraction, which views a network of devices as an approximation of a computational material with a processor at every point. Together, Proto and the amorphous medium abstraction split the programming task for a spatial computer into three largely independent layers (Figure 4): description of the application at the global level, compilation to local actions that create the specified global behavior, and approximation of an amorphous medium on the real network.

Programs written in Proto are succinct: we have observed two orders of magnitude decrease compared to other approaches[3]. Scoping appropriate to space/time operations makes them easy to reuse without interference[5]. Finally, the continuous abstraction makes them highly scalable: if a program works for a single neighborhood, it is almost certain to work for an entire metropolitan area.

Using Proto, we have constructed a library of spatial computing building blocks and applications in several areas, including sensor networks and swarm robotics. These include robust, self-healing constructs such as an active gradient that estimates minimum distance to a designated region. Active gradients correct their values for changes and failures (Figure 5) at a provably fast rate[6]. They can be used to build robust higher level modules that inherit these scaling and self-repair properties (Figure 6).

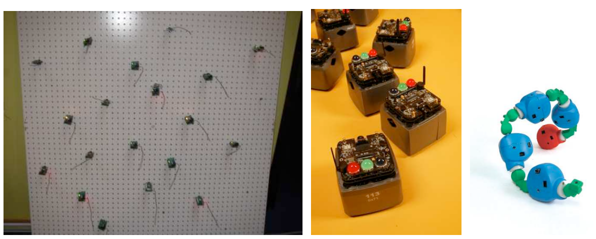
At present, Proto runs in simulation and on three platforms (Figure 7), Mica2 motes (a popular sensor network component), SwarmBots[12], and the Topobo modular robotics platform[18]. The simulator provides powerful 3D visualization and debugging capabilities, including an augmented reality capability that greatly speeds debugging of real networks.



(a) 20 nodes (b) 100 nodes

(c) 1K nodes (d) 10K nodes

Figure 6: Target tracking, built from gradients and simulated to show scaling from 20 to 10,000 nodes.



(a) Mica2 Motes (b) SwarmBots (c) Topobo

Figure 7: Proto currently supports platforms for sensor networks (a), swarm robotics (b), and modular robotics (c). (Photo credit: (b) James McLurkin and Swaine Photography, (c) Hayes Raffle and Amanda Parkes)

Related Work Proto uses Bachrach's previous work on Gooze[2] and Beal's work on the amorphous medium abstraction[4]. Among other amorphous computing work, Butera's paintable computing[7] lacks a global abstraction; other languages focus on shape formation (e.g. [13]). Other spatial computing languages are found primarily in sensor networks. These languages, such as TinyDB[11], Regiment[14], and Kairos[9], are focused on the task of gathering data to a collection point, however, and are inappropriate for a distributed control task like PACEM.

1.3 Objectives

We plan to test the feasibility of implementing PACEM using cheap, decentralized communication hardware. To this end, we will:

- Stabilize and document Proto, transforming it from a research language to a working language.
- Develop cooperative demand regulation algorithms based on our library of existing robust spatial computing algorithms.
- Demonstrate the feasibility of this approach with simplified scenarios, testing on thousands of nodes in simulation and on a network of 50-100 wireless devices (likely based on Plug[10] and/or Meraki RoofNet nodes). We will show additive benefit (increasing the percentage of customers using PACEM does not degrade overall behavior) and smooth control (regulation of demand does not lead to overcorrection and oscillation).

At the end of the year, there should be no fundamental technological obstacles preventing development and deployment of PACEM. The next steps for PACEM will be an economic study, product design, and recruitment of utility and governmental partners.

References

[1] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. Knight, R. Nagpal, E. Rauch, G. Sussman, and R. Weiss. Amorphous computing. Technical Report AIM-1665, MIT, 1999.

[2] Jonathan Bachrach. Gooze: a stream processing language. In *Lightweight Languages 2004*, November 2004.

[3] Jonathan Bachrach and Jacob Beal. Programming a sensor network as an amorphous medium. In *Distributed Computing in Sensor Systems (DCOSS) 2006 Poster*, June 2006.

[4] Jacob Beal. Programming an amorphous computational medium. In *Unconventional Programming Paradigms International Workshop*, September 2004.

[5] Jacob Beal and Jonathan Bachrach. Infrastructure for engineered emergence in sensor/actuator networks. *IEEE Intelligent Systems*, pages 10–19, March/April 2006.

[6] Jacob Beal, Jonathan Bachrach, Dan Vickery, and Mark Tobenkin. Fast self-healing gradients. In *(to appear) ACM Symposium on Applied Computing 2008*, March 2008.

[7] William Butera. *Programming a Paintable Computer*. PhD thesis, MIT, 2002.

[8] S. Darby. The effectiveness of feedback on energy consumption. a review for defra of the literature on metering, billing and direct displays. Technical report, Environmental Change Institute, University of Oxford, 2006.

[9] Ramakrishna Gummadi, Omprakash Gnawali, and Ramesh Govindan. Macro-programming wireless sensor networks using *airos*. In *DCOSS*, pages 126–140, 2005.

[10] J. Lifton, M. Feldmeier, Y. Ono, C. Lewis, and J.A. Paradiso. A platform for ubiquitous sensor deployment in occupational and domestic environments. In *International Conference on Information Processing in Sensor Networks (IPSN)*, April 2007.

[11] Samuel R. Madden, Robert Szewczyk, Michael J. Franklin, and David Culler. Supporting aggregate queries over ad-hoc wireless sensor networks. In *Workshop on Mobile Computing and Systems Applications*, 2002.

[12] James McLurkin. Stupid robot tricks: A behavior-based distributed algorithm library for programming swarms of robots. Master's thesis, MIT, 2004.

[13] Radhika Nagpal. *Programmable Self-Assembly: Constructing Global Shape using Biologically-inspired Local Interactions and Origami Mathematics*. PhD thesis, MIT, 2001.

[14] Ryan Newton and Matt Welsh. Region streams: Functional macroprogramming for sensor networks. In *First International Workshop on Data Management for Sensor Networks (DMSN)*, August 2004.

[15] Elth Ogston, Astrid Zeman, Mikhail Prokopenko, and Geoff James. Clustering distributed energy resources for large-scale demand management. In *IEEE SASO 2007*, 2007.

[16] Elinor Ostrom. *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, 1990.

[17] C.D. Patel. A vision of energy aware computing from chips to data centers. In *The International Symposium on Micro-Mechanical Engineering*, 2003.

[18] H. Raffle, A. Parkes, and H. Ishii. Topobo: A constructive assembly system with kinetic memory. *CHI*, 2004.