Toward Alternative Decentralized Infrastructures

Bill Tomlinson
Department of Informatics
University of California, Irvine
wmt@uci.edu

Bonnie Nardi
Department of Informatics
University of California, Irvine
nardi@ics.uci.edu

Donald J. Patterson
Department of Computer Science
Westmont College
dpatterson@westmont.edu

Ankita Raturi
Department of Informatics
University of California, Irvine
araturi@uci.edu

Debra Richardson
Department of Informatics
University of California, Irvine
djr@uci.edu

Jean-Daniel Saphores
Henry Samueli School of Engineering
University of California, Irvine
saphores@uci.edu

Dan Stokols
School of Social Ecology
University of California, Irvine
dstokols@uci.edu

ABSTRACT
New forms of infrastructure are needed in a world characterized by the burdens of global climate change, a growing population, increasing socio-technical complexity, and natural and human stressors to our human systems. Enabling communities to transition to a more resilient configuration of infrastructures is crucial for establishing a distributed portfolio of processes and systems by which human needs may be met. This paper proposes a potential way to increase infrastructure resilience by supporting the creation of alternative, decentralized infrastructures (ADIs) composed of small-scale, heterogeneous systems and processes. We see two possible roles for these ADIs: first, they could be integrated with existing infrastructures in the industrialized world, thereby providing some redundancy during times of strain on larger centralized systems; and second, they could help developing communities leapfrog centralized and more capital intensive conventional infrastructure. We present a model for how ADI systems may be built, based on principles from software engineering. Finally, we identify some challenges that go beyond technical implementation details in the instantiation of ADIs, and offer some thoughts on how to address them.

Categories and Subject Descriptors
H.5.m [**Miscellaneous**]; J.7 [**Computers in Other Systems**]

ACM DEV 2015, December 01 - 02, 2015, London, United Kingdom

ACM 978-1-4503-3490-7/15/11 $15.00

DOI: http://dx.doi.org/10.1145/2830629.2830648

General Terms
Human Factors

Keywords
Infrastructure; Software Engineering; ICT4D; Sustainability

1. INTRODUCTION
This paper seeks to increase the resilience of infrastructures that support life around the world. We use Holling's definition of resilience as "a measure of the persistence of systems and of their ability to absorb change and disturbance" [20]; see also [43]. Resilient systems are not static, and may fluctuate, but still persist in recognizable form [20]. An infrastructure, as defined by the US National Science Foundation, is "a network of man-made systems and processes that function cooperatively and synergistically to produce and distribute a continuous flow of essential goods and services." [12] Persistence is achieved when the infrastructure's basic services continue in spite of change and disturbance. Because forces of disturbance such as climate change, resource depletion, pollution, and growing income disparity [24] point to a future where infrastructures must be adapted to absorb such stresses, our work examines how to transition to a world that gracefully integrates decentralized infrastructures, and potentially couples them with centralized infrastructures.

Throughout this paper, we refer to centralized interdependent critical infrastructures as ICIs. The US Department of Homeland Security lists sixteen main categories of ICIs: chemical; commercial; communications; critical manufacturing; dams; defense industrial base; emergency services; energy; financial services; food and agriculture; government facilities; healthcare and public health; information technology; nuclear reactors, materials, and waste; transportation systems; and waste and wastewater systems [11]. In many developed regions, critical goods and services in these sectors are provided by centralized, government- or corporate-controlled institutions.

In addition to ICIs, though, many small-scale alternative, decentralized infrastructures have been developed around the world to serve human needs and lie outside the purview of ICIs, either due to a lack of ICIs (e.g., in developing contexts) or due to residents' dissatisfaction with ICIs (e.g., their quality, sustainability, etc.). Examples include urban gardens, biofiltration systems, home solar panels, DIY activities, and numerous others (cf., [15, 21, 30, 35]). Although these decentralized infrastructures do not enjoy the economies of scale of conventional/centralized infrastructures (e.g., rooftop gardening vs. agribusiness), they are typically a lot more environmentally benign per unit produced (e.g., rooftop gardening typically uses a lot less pesticides and fertilizers than industrial agriculture and solar panels do not emit GHG). These decentralized infrastructures currently tend to be isolated, inefficient activities that do not address regional, community-based needs. We envision that they could be integrated together, serving as elements in a greater whole. Adapting the NSF definition of infrastructure above, we define an ADI as a "network of small-scale, heterogeneous, human-made systems and processes that dynamically integrate with each other and function cooperatively and synergistically to produce and distribute a continuous flow of essential goods and services" (adapted from [12]). Throughout this paper we refer to the specific systems that could be brought together to constitute an ADI as *elements* of that ADI. For example, an urban garden would be an element of a food ADI.

The core question addressed in this paper is this: What if ADIs could be scaled up through intelligent, computer-based management, adhering to the principles of good software design, to produce a transition to more integrated, regional structures?

The structure of the paper is as follows. First, we describe the role that ADIs could play in a selection of sectors in developed and developing contexts. Second, we describe how an ADI system could be implemented, based on principles from software engineering, and integrated with sensors to provide with up-to-date information about its elements. Finally, we present a number of key challenges facing ADIs, and potential plans for addressing these challenges.

2. ADI SYSTEMS IN USE

Extensive computer systems exist to manage centralized infrastructures for water, energy, transportation, and many other critical infrastructure sectors discussed above. However, small-scale systems, managed locally, are frequently integrated in an ad hoc fashion, if at all. We envision a world in which human needs such as food, water, and energy are met, at levels currently enjoyed in the industrialized world, not primarily by large-scale, corporate- or government-controlled infrastructures, but rather by well coordinated, distributed collections of small-scale systems and services. This section describes the role ADI systems could play across various different sectors.

2.1 Food

In many parts of the world, food is provided by industrial agricultural systems. In others, the primary sources of nutrition are subsistence agriculture and other small-scale activities. In an ADI system, food would be produced, processed (e.g. dried, milled, deboned, and packaged), and distributed via many small-scale farms, processing systems, and transportation activities that together are able to perform at the efficiency of industrial agriculture, but with significantly greater robustness. These production, processing, and distribution systems would be dynamically coordinated by computational systems that allow for individual elements entering or leaving the system. For example, as people move from place to place, a farm may disappear in one location and arise in a new location. If a family bought a truck, they could quickly be included in the routing mechanisms that enable food to be moved from place of production to place of consumption. If climate change compromised the viability of one crop, farms growing different crops and therefore needing different distribution processes and routes could be gradually integrated into the system as a whole.

By coordinating and connecting individual elements, network economies could be realized that would lead to greater production efficiency without the large external costs of current ICIs. Information from the various elements of the ADI system would be uploaded to the computational algorithms via a combination of automatic sensors (e.g., GPS locations of vehicles) and guide human effort (e.g. number of squash harvested on a given day). This could enable optimization of the flows of resources through the food system, provide metrics to measure performance, and allow for other forms of system analysis. ADIs could integrate with tools such as KrishiEkta [2], OneFarm [1], and KrishiMantra [26] to provide farmers across a region with important information regarding opportunities within the local food system.

2.2 Water

The provision of water takes many different forms around the world, from centralized, government-controlled water infrastructures such as those found in many industrialized nations, to systems where people carry containers of water each day from hand pumps, rivers, or streams to their homes (e.g., [44]).

An ADI system could integrate numerous different forms of available water (from rivers, reservoirs, hand pumps, greywater collectors, desalinization plants, fog harvesters, biofiltration systems, etc.), and help coordinate the production of different levels of water quality to satisfy local needs. Water flow and water quality sensors could be systematically integrated into these water sources to optimize the value of their services to humans without compromising the value they provide to other organisms and the long-term viability of our ecosystems more broadly.

As an example of the potential benefits of an ADI system, rather than watering a garden with fresh water or reclaimed water, a greywater collection system from nearby homes could provide water of high enough quality for that purpose. Where such a greywater system does not exist, an ADI system could provide suggestions for potential ways to improve the overall functionality of the system by finding bottlenecks and suggesting opportunities for improvement. Water ADIs could make use of currently unclaimed water resources, such as greywater, and allow several local sources to pool their resources to benefit the community.

2.3 Energy

In the energy domain, as with food and water, there are numerous pathways by which energy is generated and distributed. From centralized fossil fuel-based power companies to solar charging kiosks [23] to energy distribution via discarded laptop batteries [7], various communities have developed a diverse array of ways to meet the need for energy. In an ADI system, these diverse pathways that energy enters and travels within human communities would be integrated, monitored, and routed algorithmically. Given the increasing policy emphasis on reducing our dependence on fossil fuels to cut our emissions of greenhouse gases, local alternative sources of energy such as wind and solar power will become increasingly important and require much better management to smooth weather related variations in order to satisfy demand for energy.

2.4 Information Technology

As with each of the above, access to information technology currently occurs through a range of channels. As well as increasing use of mobile, desktop, and internet technologies in industrialized nations, there are numerous smaller-scale IT activities throughout the world (e.g., [9, 10]). By providing a rigorous way of documenting people's IT needs/wants, the IT resources available, and dynamically connecting them, an ADI system could improve the effectiveness with which a large collection of small-scale, heterogeneous IT systems could be provided to people.

There is also significant interest in creating alternative networking systems: from work on enabling local voice communication through a mesh network [17], to providing universal Internet access to deprived regions through WiFi sharing [39]. These are all potential elements of an ADI system that would equip people with a range of networking and communication services appropriate to their geography.

2.5 Integration Across Infrastructures

An important contribution that an ADI system could make lies not just within a particular sector (e.g., food, water), but also in the process of integrating across different infrastructures. In the first example above, the linkages between food and transportation were evident. Similarly, an ADI system could be helpful in harnessing the complex linkages between IT and energy (e.g., powering devices), between water and food (e.g., watering food plants), between food and energy (e.g., biofuels), between energy and water (e.g., cooling or pumping water), and other more tenuous interactions. Allowing all of these small-scale, heterogeneous, distributed infrastructural elements to work together would hopefully better satisfy human needs than conventional industrial infrastructures, with significantly improved robustness and resilience.

2.6 ADIs in Developed and Developing Contexts

ADIs have potential application in both developed and developing contexts. If an ADI system were deployed to work jointly with conventional infrastructure, the elements that make up the ADI could provide a complement to the centralized infrastructure, thereby enabling more sustainable lifestyles, more effective utilization of resources, and resilience against potential collapse [42]. If the ADI system were deployed in a context where reliable infrastructures are lacking, the dynamic coordination could link together many existing but not-yet-well-connected mechanisms for how people's needs are met, streamlining the process and making apparent those places where new ventures and other forms of development are most needed [35]. In addition, where conventional/centralized, large-scale infrastructures are missing, such as in developing communities, ADIs could provide a mechanism to leapfrog their developed counterparts [8], as illustrated by the development of cellular communications that made obsolete the need to install landlines.

In many developing regions people have begun creating alternative decentralized infrastructures already, driven by need and implemented with materials readily available. Often though, they are interfacing with more centralized infrastructures, but in an *ad hoc* way. For example, electricity grids may be subject to parasitic loads from unsanctioned line splitting, and typically semi-centralized resource distribution (for example fuel) is augmented with individuals buying, selling and transporting small amounts on a gray market. The development of well-structured ADIs could actually facilitate these kinds of relationships, but in a way that would benefit traditional suppliers and the *ad hoc* suppliers as well. Clearly there is a need because there exist many ADIs, very poorly specified, in developing regions already. As we continue to work on these ideas, we want to be alert to the varied global socieconomic contexts in which ADIs might be deployed. Infrastructural components vary across contexts, yet we believe that the notion of interfaces between centralized and decentralized infrastructures has great generality.

3. HOW AN ADI SYSTEM WOULD WORK

We envision the core of the implementation of such a system being based on ideas arising out of software engineering. This section presents a summary of the technical aspects of a potential ADI implementation. The design of such a system would be informed by interviews with key stakeholder groups to ensure that the technical aspects provide the desired effect on quality of life. For example, interviewees could include:

- scholars and activists with deep expertise in particular ADI domains or the interactions between domains, for example, [15] on food, [38] on water, [33] on energy, and [40] on information technology;
- citizen users of alternative technologies and participants in local community activities geared to sustainability; and
- private sector consultants, regulators, government decision-makers, and politicians whose joint efforts are essential for catalyzing the adoption and expansion of alternative sustainable infrastructures.

These interviews would inform the development of computational models of ADI elements and the interfaces between them, and of the process by which sensors could be enabled to measure aspects of the elements automatically.

3.1 Interfaces Among ADI Elements

A key challenge for integrating many small-scale systems lies at the interfaces between those systems. The software engineering community has well-established principles for rigorously specifying interfaces and building systems around those interfaces. Software engineering has developed sophisticated understandings of the roles and properties of interfaces for specifying interdependencies and interactions among the components of complex software systems, to explore and characterize the interfaces between interdependent infrastructures. Software engineers are also looking at adapting existing practice to cater to infrastructural requirements such as sustainability [37] and resilience [36].

Software engineering focuses on "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software" [22]. We see significant potential in bringing this approach to infrastructure analysis. In particular, we propose that coupled decentralized/centralized infrastructures could work efficiently and effectively through intelligently managed interfaces.

A key focus of the ideas presented here is to use insights from software engineering, drawing on the analogous complexities of large-scale software systems with interdependent components that are addressed by disciplined techniques to specify well-defined interfaces that describe component interdependencies. Software interface specifications are carefully defined rules constraining the

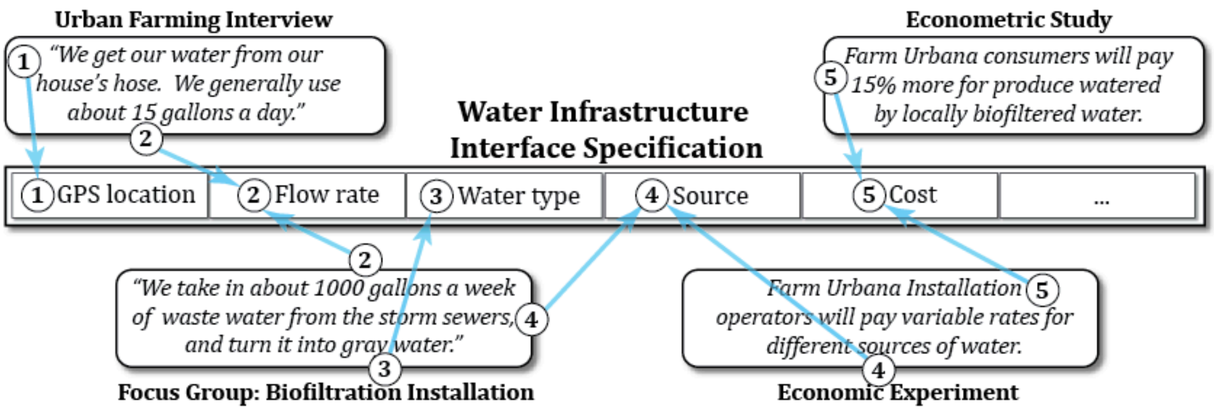

Figure 1: An illustration of how content from stakeholder interviews could inform the infrastructure interface specification model.

interactions of interdependent components, specified using an interface description language (IDL) [27]. Software engineering researchers have evolved IDLs, making it possible to specify not just input/output between components, but a range of additional properties such as dependencies [45], measurements [14], and behavior [28]. A related development in software engineering is Application Programming Interface (API) design [5, 41]. APIs are crafted explicitly to expose only chosen functionality and/or data of a software application (components) while safeguarding other parts of the application. A properly crafted API enables the underlying implementation to change without affecting a client system using the application – in fact, the application can be completely "swapped" with another, so long as the replacing application adheres to the API [29].

We envision that the concept of component interface specification could be fruitfully applied to connecting ICIs and ADIs. The interface specification of an infrastructure would specify the resources provided as well as certain attributes of the resources. We suggest that qualities adapted from software, such as efficiency, reliability, robustness, reusability, visibility, and others should be characterized by the interface specification as well, as such qualities are especially relevant to infrastructures. Interface specifications could enable ADIs to be explored more easily in simulations, and eventually scaled up from local demonstration projects to regional scale innovations that may complement existing, dominant ICIs.

The interface specification model would include several aspects of infrastructure interdependencies, including: (a) resources and other inputs/outputs required and provided along with the attributes of these objects, (b) object and attribute visibility among infrastructures, and (c) other desired/delivered qualities of the infrastructure. The value of such carefully defined interfaces lies both in defining the scope of particular ADIs and the potential to replace one infrastructure (ICI) with an alternative (ADI) satisfying the same API. This ability to "swap" in an alternative infrastructure leads to resiliency in a manner similar to design diversity for software fault tolerance [4] and is similar to dynamic software component "swapping" by RAIC—Redundant Arrays of Independent Components [29].

The goal is to arrive at the minimal interface necessary to capture the characteristics of most interactions that ADIs may have with ICIs or with each other. While such a technical approach may seem unnecessary at first glance (e.g., because an existing ADI already realizes where its inputs are coming from and where its outputs are going), ensuring consistency across ADIs and ICIs through rigorously defined interface specifications makes it feasible to place the interactions across those systems under computational control.

3.1.1 Inputs/Outputs

An infrastructure interface specification (as with a software interface) should define the input/output objects that are required/provided in its interaction with other infrastructure systems, along with the attributes of those objects. Primary I/O objects are resources, but there may also be parameterization, for instance, to control the magnitude of resources or other inputs controlling infrastructure functionality as there may be other outputs denoting infrastructure state.

The interface must cover not only the commonalities but also the unique aspects of interactions. For example, in interfacing with the water sector (see Figure 1), the attributes "flow rate," "water type," and "location" may be shared across most water sources discussed in stakeholder interviews, but less ubiquitous issues such as a "bacterial content" attribute could surface in rarer instances. Each attribute must also be defined by the appropriate measurement or units (e.g., gallons per day vs. cubic feet per second, or cold | cool | warm | hot | boiling vs. degrees Fahrenheit/Celcius), ways of converting from one unit to another ("cool equals 55-85°F"), and sensible default values.

3.1.2 Object Visibility

A second key aspect of infrastructure interface specification relates to object and attribute visibility. Continuing with water as an example, an organization may be comfortable sharing the flow rate out of a bathroom sink drain, but not the water's bacterial content (which could be used to assess the personal health of people who live/work there). Based on stakeholder interviews, we would seek to establish appropriate categories of visibility—e.g., private, friend organizations only, same sector only, other sectors only, public – as well as specific visibility between infrastructures (ADIs or ICIs). We would also seek to establish sensible default visibility values for each object in an interface.

3.1.3 Infrastructure Qualities

A third aspect of infrastructure interface specification focuses on the qualities of the infrastructure system represented by an interface. One key aspect of a critical infrastructure, for instance, is "up-time", which is related to but not equivalent to the software quality of reliability. As an example, regional energy companies in the US have very high levels of up-time, while a local solar

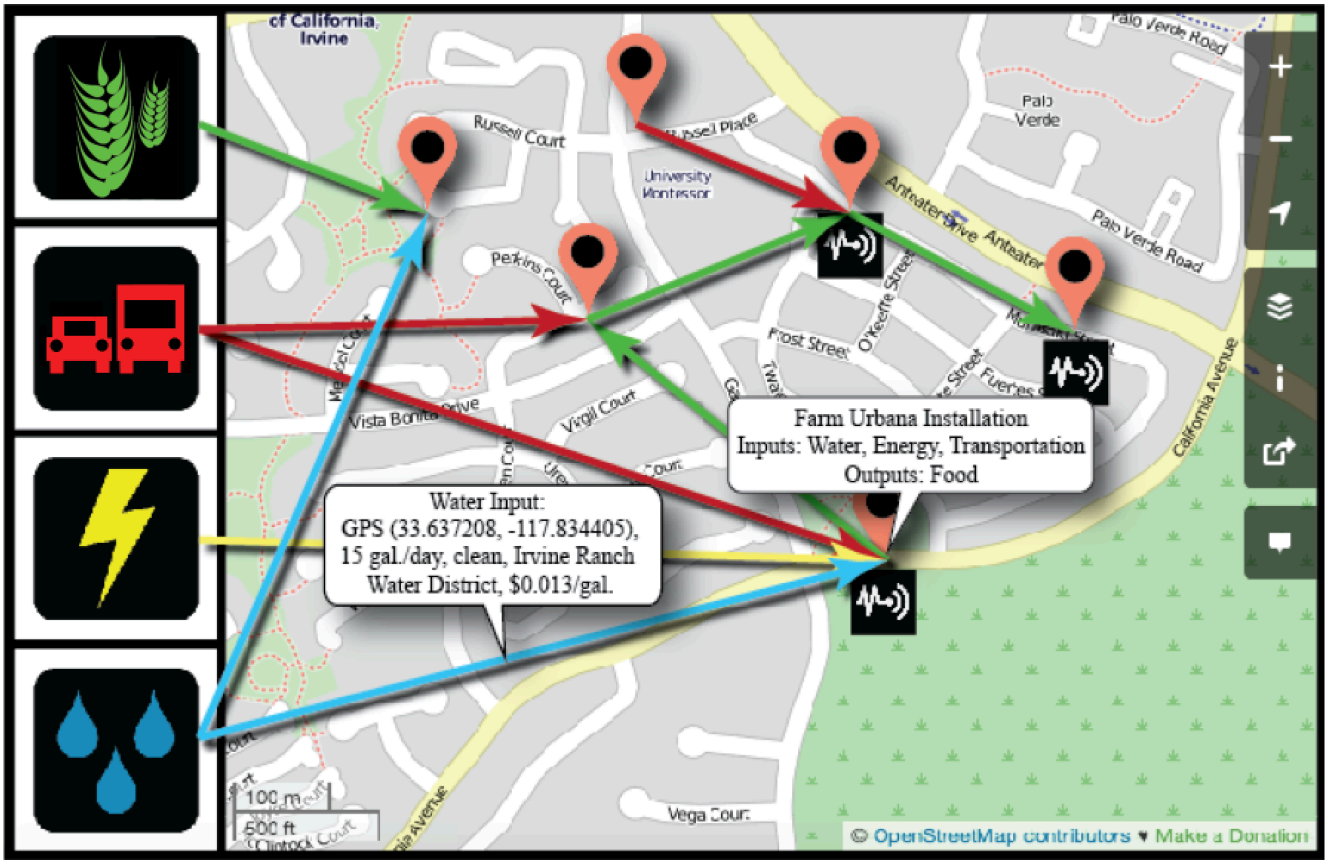

Figure 2: A mockup of a visualization of an ADI system. © OpenStreetMap contributors and the authors of this paper.

installation is inherently intermittent based on availability of sunshine. Critical infrastructures are also highly interconnected. Seemingly separate infrastructures such as water and communication can affect the performance of elements in each infrastructure [35], requiring not only the development of resilient networks of elements and infrastructures (e.g., [6]), but also ADIs that are flexible and adaptable. Stakeholder interviews would help determine what software qualities and variants apply to which infrastructures, and allow the interface specification model to be augmented to specify particular qualities.

3.2 ADI Monitor System

Advances in manufacturing efficiencies and the subsequent reduction in prices of sensors have caused their deployment to rapidly increase and their role in infrastructures to grow. From barometers on cell-phones and motion sensors on light switches in offices, to seismographs in civil infrastructure and smart meters in homes, physical sensors are proliferating. In order to facilitate system maintenance and the analysis of data, many of these sensors are connected to applications and to Internet services. Collectively this "Internet of Things" is becoming an increasingly important and pervasive facet of infrastructure interface design (cf., [3]).

At the same time, software, both simple and sophisticated, is being used to create virtual sensors; for example, geographic crowd sentiment analysis derived from Twitter feeds that capture the mood of a city [25, 32], and search engine aggregations that identify disease outbreaks [18]. While the spread of sensors has created increased opportunities for context-aware applications and is tightly coupled with infrastructure [3], it is difficult to manage the opportunities afforded by the scale of this trend—there are difficulties inherent in discovering, collecting, fusing, and reasoning with data from the heterogeneous set of distributed sensors.

In the face of these trends and challenges, we propose to develop a monitoring network (the "ADI Monitor") that is particularly well suited to the decentralized nature of ADIs (see Figure 2 for a mockup of a visualization of such a system). An initial prototype [6] has been developed of this system, but significant work is still needed in order to allow it to monitor real-world infrastructure interface connections. In our current design the ADI Monitor has computational processes or "nodes" that provide value-added information services on top of networked physical and virtual infrastructure sensors. Conceptually, the "owner" of a sensor pairs one ADI Monitor "node" with one sensor that is already exposed on the internet. Nodes communicate with each other using peer-to-peer networks. Here we describe future work that could extend this monitoring system so that real-world deployments are robust and easy for end-users to operate. The system would enable a range of capabilities, detailed below.

Organization of Existing Sensors: The ADI Monitor would not seek to deploy new sensors, but to integrate existing sensors into our network. In the pairing process of an ADI Monitor node with the underlying sensor, the sensor is exposed to the rest of the ADI Monitor system with a consistent data interface. The node/sensor pairs become searchable through an automated directory and the complexity of gathering data from now-exposed sensors is greatly reduced. Examples of sensors that we are integrating include smart-meters on homes, electric current sensors on solar panels, irrigation system operational status, micro-weather stations, and smart-phone/wearable sensors. We intend to broaden our targets based on feedback from our stakeholder interviews.

Intermittency Tolerance: The ADI Monitor nodes can communicate using a peer-to-peer networking protocol [34] that is itself resilient to intermittency, but also, as a result, supports continued monitoring in the face of partial infrastructure outages.

Scaling: By using a decentralized peer-to-peer network, the ADI Monitor supports extremely large networks of sensors. We anticipate incorporating approximately 10,000 sensor values in our next stage of deployment, many focused on geographic specializations.

Analytics: By incorporating statistical modeling of the sensor values, ADI nodes can be given predictive analytics that enable them to function on par with utility-grade infrastructures.

Operator Reflection: Consistent, reliable access to decentralized sensor systems coupled with high quality user-interfaces would enable ADI owner/operators insight into how to manage their ADIs more effectively.

Research Data: The construction of the ADI Monitor would utilize cloud resources such as Google App Engine and Amazon EC2. By consolidating and exposing sensor systems we potentially introduce new vulnerabilities related to privacy and data exploitation. It is important to have mechanisms that mitigate these concerns by using high-grade encryption and access parameters controlled by sensor owners and designed from stakeholder interviews.

Typical sensor studies focus on a limited sensor portfolio, i.e., researchers build event recognition models by applying machine learning methods to the data collected from one sensor or from a closely coupled set of sensors. Though this demonstrates how powerful a small federation of sensor data can be if interpreted correctly, it ignores the fact that our world is a much more broadly instrumented multisensory environment now. Such an environment should enable new events to be recognized for the first time. We hypothesize that the state of infrastructures is just such a new category of context. Each of the many sensors provides a different perspective and complements information from other sensors to construct a comprehensive picture. As well as enabling many more sensors to be aggregated by a single application, the ADI Monitor also enables a single sensor to be simultaneously utilized by more than one application.

Applications that are built on top of the ADI Monitor would be able to harvest information from a network of sensors, including physical stationary sensors, mobile and worn sensors, and virtual or physical social sensors, and would provide richer insights and potentially support novel use cases.

The final component of the ADI Monitor would be a mobile application that enables users to connect sensors to the ADI Monitor network and to visualize related data that is already incorporated. For example, an urban farmer may stand next to the location where her outdoor spigot connects to her automated sprinkler system, and use her mobile phone to launch an ADI Monitor node to monitor the sprinkler's operation. In the process, it could reveal assets in her immediate vicinity (e.g., water output from a nearby biofiltration system) that may be useful to her farm and already implemented as ADI Monitor nodes.

4. OUTSTANDING CHALLENGES

The possibility of ADI systems coming into existence brings to the fore a number of significant challenges beyond the technical details of implementation discussed above.

First, an ADI system would rely heavily on a computing infrastructure in order to work well. While such computing infrastructures are available in many parts of the world (more than half the world's population are mobile phone subscribers, and there are more mobile connections (including M2M) than there are people [13, 19], not everyone has equal access to the computing infrastructures that would make ADIs feasible. Therefore, it is possible that, from a quality-of-life perspective, ADIs would preferentially benefit those who already have more than enough. Nevertheless, from a sustainability perspective, those who enjoy such abundance may be exactly the people who should migrate to a more sustainable model of infrastructure. Either way, to address this issue at least in part, ADI implementations should be developed with the goal of utilizing the lowest-tech and most available technologies on which they would be viable.

A second problem involves the question of justice, from an algorithmic perspective. If a computational system is routing resources, who gets to decide what the optimal utilization of resources is? Does the algorithm seek the highest average quality of life, the highest minimum quality of life, the greatest sustainability of the system, or some other goal? There have been discussions about how Google's search algorithm may be a powerful influencer of elections [16]; similarly, the algorithm underlying an ADI would by its nature bias the distribution of life sustaining resources. Therefore, it would be critical to have a wide range of key stakeholders involved in decisions about how the algorithms should function, and transparency in the implementation of those algorithms must be ensured.

A third problem lies in the possibility that, by centralizing the decision-making process controlling the distribution of resources (even if the production of those resources is via small-scale decentralized infrastructure elements), ADIs might create a single point of exploitation for those elements. The elements that would be brought together into an ADI presumably already have some mechanism for distributing their products and services to relevant end-points, even if those mechanisms may not be particularly efficient or robust. Nevertheless, efforts exist to allow digital systems to be as robust as the offline equivalents that they replace (e.g., [31]); the ADI routing systems could hopefully be made similarly robust.

Overall, while challenges are substantial, they are comparable (if not identical) to those found in conventional industrial infrastructures. We hope that the transparency that could be enabled by an ADI system would help address them.

5. CONCLUSIONS

In this paper, we have presented the concept of alternative, decentralized infrastructures (ADIs), and their potential role in providing for human needs around the world. We have laid out a plan by which ideas from software engineering could be brought to bear on the process of implementing ADI systems. By doing so, we seek to elevate the role of small-scale, decentralized aspects of infrastructure in the larger landscape of infrastructures. We hope that the conceptualization of ADIs presented here will help inform policy decisions that facilitate smoother transition to more resilient infrastructure, and societal shifts toward more sustainable ways of life.

6. ACKNOWLEDGMENTS

The authors thank Richard Donovan, Neil Young and the ACM DEV reviewers for their suggestions and contributions to this work.

7. REFERENCES

[1] Aditya, V. and Sasikumar, K. 2013. Nutrient management decision support system for livelihood security of farmers. *Proceedings of the 3rd ACM Symposium on Computing for Development (ACM DEV '13)* (New York, New York, USA, 2013).

[2] Agrawal, R. and Sundari, S.K. 2012. KrishiEkta : Integrated Knowledge and Information Distribution System for Indian Agriculture. *Proceedings of the 2nd ACM Symposium on Computing for Development (ACM DEV '12)* (New York, New York, USA, 2012).

[3] Atzori, L., Iera, A. and Morabito, G. 2010. The internet of things: A survey. *Computer networks*. 54, 15 (2010).

[4] Avizienis, A. and Kelly, J. 1984. Fault tolerance by design diversity: Concepts and experiments. *Computer*. 17, 8 (1984).

[5] Bloch, J. 2006. How to design a good API and why it matters. *In Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications (OOPSLA '06)* (New York, New York, USA, 2006).

[6] Brock, J. and Patterson, D.J. 2015. Cacophony: Building a Resilient Internet of Things. In First Workshop on Computing within Limits. *First Workshop on Computing within Limits (LIMITS 2015).* (Irvine, CA, 2015).

[7] Chandan, V., Jain, M. and Khadilkar, H. 2014. UrJar: A Lighting Solution using Discarded Laptop Batteries. *Proceedings of the Fifth ACM Symposium on Computing for Development (ACM DEV-5 '14)* (New York, New York, USA, 2014).

[8] Chen, J. 2015. Computing within Limits and ICTD. *First Workshop on Computing within Limits (LIMITS 2015)*. (Irvine, CA, 2015).

[9] Chen, J. and Subramanian, L. 2013. Interactive web caching for slow or intermittent networks. *Proceedings of the 4th Annual Symposium on …* (New York, New York, USA, 2013).

[10] Corrigan-Gibbs, H. and Chen, J. 2014. FlashPatch: spreading software updates over flash drives in under-connected regions. *Proceedings of the Fifth ACM Symposium on Computing for Development (ACM DEV-5 '14)* (New York, New York, USA, 2014).

[11] Critical Infrastructure Sectors: 2015. *http://www.dhs.gov/critical-infrastructure-sectors*. Accessed: 2015-09-06.

[12] Critical Resilient Interdependent Infrastructure Systems and Processes (CRISP): *http://www.nsf.gov/pubs/2015/nsf15531/nsf15531.htm*.

[13] Current World Population: 2015. *http://www.worldometers.info/world-population/*. Accessed: 2015-09-06.

[14] Damevski, K. 2009. Expressing measurement units in interfaces for scientific component software. *Proceedings of the 2009 Workshop on Component-Based High Performance Computing (CBHPC '09)* (New York, New York, USA, 2009).

[15] Despommier, D. 2010. *The vertical farm: feeding the world in the 21st century*. Thomas Dunne Books, St. Martin's Press.

[16] Epstein, R. 2015. How Google Could Rig the 2015 Election. *Politico Magazine*.

[17] Gabale, V., Raman, B., Chebrolu, K. and Kulkarni, P. 2010. LiT MAC: addressing the challenges of effective voice communication in a low cost, low power wireless mesh network. *Proceedings of the First ACM Symposium on Computing for Development (ACM DEV '10)* (New York, New York, USA, 2010).

[18] Ginsberg, J., Mohebbi, M. and Patel, R. 2009. Detecting influenza epidemics using search engine query data. *Nature*. 457, 7232 (2009).

[19] Global Data: 2015. *https://gsmaintelligence.com/*. Accessed: 2015-09-06.

[20] Holling, C. 1973. Resilience and stability of ecological systems. *Annual review of ecology and systematics*. 4, (1973).

[21] Hopkins, R. 2008. *The transition handbook: from oil dependency to local resilience*. Chelsea Green Publishing.

[22] IEEE 2010. Systems and software engineering -- Vocabulary. ISO/IEC/IEEE 24765:2010(E). 2010.

[23] Iland, D. and Belding, E. 2014. Open Charging Kiosk: A Business in a Box. *InProceedings of the Fifth ACM Symposium on Computing for Development (ACM DEV-5 '14)*. (New York, New York, USA, 2014).

[24] IPCC 2014. *Part A: Global and Sectoral Aspects. Contribution of Working Group II to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*.

[25] Kouloumpis, E., Wilson, T. and Moore, J. 2011. Twitter sentiment analysis: The good the bad and the omg! *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM 11)* (2011).

[26] Kumar, V. and Dave, V. 2013. Krishimantra: agricultural recommendation system. *Proceedings of the 3rd ACM Symposium on Computing for Development (ACM DEV '13)* (New York, New York, USA, 2013).

[27] Lamb, D. 1987. IDL: Sharing intermediate representations. *ACM Transactions on Programming Languages and Systems*. 9, 3 (1987).

[28] Leavens, G., Baker, A. and Ruby, C. 2006. Preliminary design of JML: A behavioral interface specification language for Java. *ACM SIGSOFT Software Engineering Notes*. 31, 3 (2006).

[29] Liu, C. and Richardson, D. 2002. RAIC: Architecting dependable systems through redundancy and just-in-time testing. *ICSE 2002 Workshop on Architecting Dependable Systems (WADS)* (Orlando, FL, USA, 2002).

[30] Lydon, M. and Garcia, A. 2015. *Tactical urbanism: Short-term action for long-term change.* Island Press.

[31] Maniatis, P. and Roussopoulos, M. 2005. The LOCKSS peer-to-peer digital preservation system. *ACM Transactions on Computer Systems*. 23, 1 (2005).

[32] Mitchell, L., Frank, M. and Harris, K. 2013. The geography of happiness: Connecting twitter sentiment and expression, demographics, and objective characteristics of place. *PLoS ONE*. 8, 5:e64417 (2013).

[33] Modi, V., McDade, S., Lallement, D. and Saghir, J. 2005. *Energy and the Millenium Development Goals*.

[34] p2p4java: *https://github.com/djp3/p2p4java*. Accessed: 2015-09-06.

[35] Patterson, D.J. 2015. Haitian Resiliency: A Case Study in Intermittent Infrastructure. *First Workshop on Computing within Limits (LIMITS 2015)*. (Irvine, CA, 2015).

[36] Penzenstadler, B., Raturi, A., Richardson, D., Silberman, S. and Tomlinson, B. 2015. Collapse (& Other Futures) Software Engineering. *First Workshop on Computing within Limits (LIMITS 2015)*. (Irvine, CA, 2015).

[37] Penzenstadler, B., Raturi, A., Richardson, D. and Tomlinson, B. 2014. Safety, security, now sustainability: The nonfunctional requirement for the 21st century. *IEEE Software*. 31, 3 (2014).

[38] Radcliffe, J. 2004. *Water recycling in Australia: a review undertaken by the Australian academy of technological sciences and engineering*.

[39] Sathiaseelan, A. and Mortier, R. 2014. A Feasibility Study of an In-the-Wild Experimental Public Access WiFi Network. *Proceedings of the Fifth ACM Symposium on Computing for Development (ACM DEV-5 '14)* (New York, New York, USA, 2014).

[40] Smarr, L. 2010. The Growing Interdependence of the Internet and Climate Change. *IEEE Internet Computing Magazine*. Jan/Feb, (2010).

[41] Stylos, J. and Myers, B. 2007. Mapping the space of API design decisions. *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VLHCC '07)*. (Washington DC, USA, 2007).

[42] Tomlinson, B., Silberman, M., Patterson, D., Pan, Y. and Blevis, E. 2012. Collapse informatics: augmenting the sustainability & ICT4D discourse in HCI. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)* (New York, New York, USA, 2012).

[43] Walker, B., Holling, C., Carpenter, S. and Kinzig, A. 2004. Resilience, adaptability and transformability in social--ecological systems. *Ecology and society*. 9, 2:5 (2004).

[44] Water & Desertification: 2010. *http://archive.unu.edu/africa/activities/water.html*. Accessed: 2015-09-06.

[45] Wolf, A., Clarke, L. and Wileden, J. 1989. The AdaPIC tool set: Supporting interface control and analysis throughout the software development process. *IEEE Transactions on Software Engineering*. 15, 3 (1989).