

Experiences Integrating Building Data with sMAP

*Stephen Dawson-Haggerty
Andrew Krioukov
David E. Culler*



Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-21

<http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-21.html>

February 6, 2012

Copyright © 2012, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Acknowledgement

Special thanks to Albert Goto for much helpful assistance. This work is supported in part by the National Science Foundation under grants #CNS-0932209, an NSF Graduate Fellowship, the FCRP MySyC Center, Intel Corp, and Samsung Corp.

Experiences Integrating Building Data with sMAP

Stephen Dawson-Haggerty, Andrew Krioukov, and David Culler

Abstract

Originally designed as a RESTful protocol to expose reading data from simple instruments, we discuss the shifting role of sMAP as a common interchange format for timeseries data within a number of building projects. Due to the change in role and resulting widened set of use cases, we discuss where sMAP has fallen short and design modifications necessary to address them. Key contributions are identifying a set of features required for the broad applicability of a protocol for publishing time-series data, and a concrete proposal and implementation of our design grounded in a diverse set of use cases and real deployments.

1 Introduction

Buildings are sources of diverse physical information due to their heterogeneous set of systems and long lifetimes. Furthermore, they are currently the subject of extensive research efforts towards reducing their energy consumption in order to meet efficiency and carbon targets; for instance, the California Energy 2050 study *assumes a 40% reduction* in energy consumption across the entire building stock by that time. In order to affect these reductions and make them stick, one key enabler will be integrating substantially more information into building controls, deeper analysis, and better modeling; there is good evidence that considering weather forecasts, occupancy, user preference, and other inputs not frequently incorporated into building control loops can reduce the energy spend through techniques like improved scheduling, fault detection, and superior control logic. All of these techniques depend on obtaining ac-

cess to and integrating with building data.

Since 2009, we have been engaged in an extensive effort to break open access to this information, which is frequently locked behind obscure instrument interfaces or inside of proprietary stacks, and make it available easily and simply using the Simple Measurement and Actuation Profile (sMAP). The sMAP profile is designed to present **physical information** over HTTP using a simple JSON schema; it allows information consumers like databases and controllers to access time series data from instruments without needing access to the underlying measurement infrastructure, and allows multiple consumers to coexist on top of the same data generators; this has been a key enabler in our research. Since we set out on this project, we have exposed tens of thousands of channels of data from hundreds of instruments and tens of buildings using sMAP.

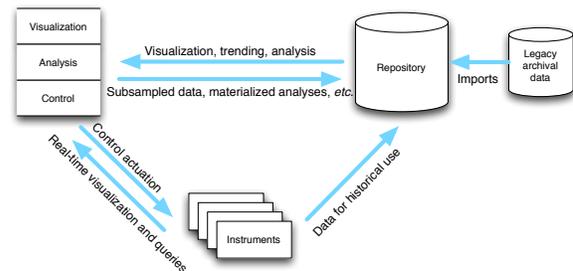


Figure 1. The arrows represent places where sMAP is used. sMAP usage has expanded from its original mission of uniformly presenting access to data from instruments, to a more general-purpose protocol for describing and transmitting time series data from instruments, analysis engines, and repositories. We have redesigned the protocol and implementation to both better support this usage as well as making it simpler and more uniform.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

In the course of applying sMAP to many different instruments and systems, we have gained significant experience with using it to publish many types of physical data; we also use it to make actuation available through

a consistent interface. Here we codify these lessons and show how they have guided the redesign of sMAP. In this paper we make three primary contributions. First, we present a survey of data which is available inside buildings, or might be added during a retrofit. Secondly, we examine the sources and sinks of this data inside a building research ecosystem, and examine the requirements for publishing it via sMAP or other protocols. Finally, we introduce a revision of the sMAP profile, sMAP 2 which more completely addresses the needs of building information data acquisition.

2 Building Information Sources

Researchers working in buildings encounter a wide variety of information. We have found that the sources of this information can be broken roughly into three categories:

Building management systems: many modern large buildings have an existing management system (BMS) responsible for supervisory control and analysis; it typically has a front-end GUI which allows a manager to “trend” readings from an extensive albeit primitive network of sense and control points, and compare them to a historical data.

Retrofit systems: buildings used as research systems often also contain many other pieces of instrumentation which are measuring the same system but not integrated with the BMS for one reason or another.

Transient instrumentation: researchers also often perform audits or point measurements of building performance using portable instruments which is left in place for anywhere between a few minutes and a few months, then removed and reused elsewhere.

2.1 Case Studies

In order to draw out the requirements of a data collection protocol suited to this environment, we briefly present four case studies of building data environments which include these elements.

The first is a demand-response and load shedding project occurring with Sutarja-Dai Hall as the building testbed. The goal of this project is to be able to achieve a 20-30% reduction in energy use during periods of peak demand. The building is new construction, commissioned in 2009 and contains a state-of-the-art Siemens Apogee BMS, along with around 6000 sense and control points monitoring many points in the wet and dry HVAC loops, indoor air quality and light levels, light switch positions, and per-floor electrical plug load meters exposed over BACnet. There is also a separate WattStopper system with a separate BACnet/IP implementation with control of the per-zone light and thermostat set-points. Due to peculiarities of their BACnet implementation, it is not possible to sample all 6000 points at high rate, but we are able to sample around 1000 of them

at sub-minute resolution; still a considerable amount of data. BACnet exposes and organizes the points using a simple scheme which corresponds to the topology of the underlying serial bus (rather than a more meaningful systems or physical view).

Our second case-study is the California Energy Commission (CEC) building-to-grid testbed, located in the neighboring Cory Hall. Legacy in many senses of the word, after the latest retrofitting contains roughly 2000 channels of data from 50 discrete three-phase electric meters. These meters are not integrated by a BMS, but instead are Modbus meters accessed over a vlan (using Modbus/Ethernet adaptors). Additionally, we have integrated data from three existing IP-based electric meters, steam gauges, condensate meters, and 20 ACme plug-load meters and temperature sensors. Historical data for some systems is available from a campus contractor. Each system as provided by the contractor has different interfaces.

The third deployment was conducted in collaboration with the LBNL Building Technologies Group in order to better understand how plug-loads are used by building occupants. Accounting for around 40% of building energy use, it is the most poorly understood piece. In order to improve our understanding, we inventoried all plug loads in LBNL Building 90, a large commercial building; we then performed a stratified sampling of them and installed 450 ACme plug-load meters, forming a large 6loWPAN/IPv6 mesh network. The meters report 10-second power and energy data through seven Load-Balancing Routers back to a central repository.

Finally, also in collaboration with LBNL we also conducted a number of residential studies with approximately 80 ACme meters deployed in each residence. The goal there was also to better understand the usage patterns of electrical plug-loads.

3 Experiences with sMAP 1.0

The sMAP profile, as described in previous work [5] specifies an organization of HTTP resources, and the contents of these resource as JSON schema. The four top-level resources in the sMAP profile are:

`/data` contains resources for reading and controlling meters, sensors, and actuators.

`/reporting` allows control of periodic reports for syndication.

`/status` contains a single universal universal field specifying if the device data is valid, as well as instrument-specific codes.

`/context` contains any information about the device’s relationship to other devices. This includes the device’s Global Unique Identifier.

The most-commonly used resources are `data` and `reporting`. Data is organized as three-level deep set of resources, of the form `<point>/<type>/<channel>`.

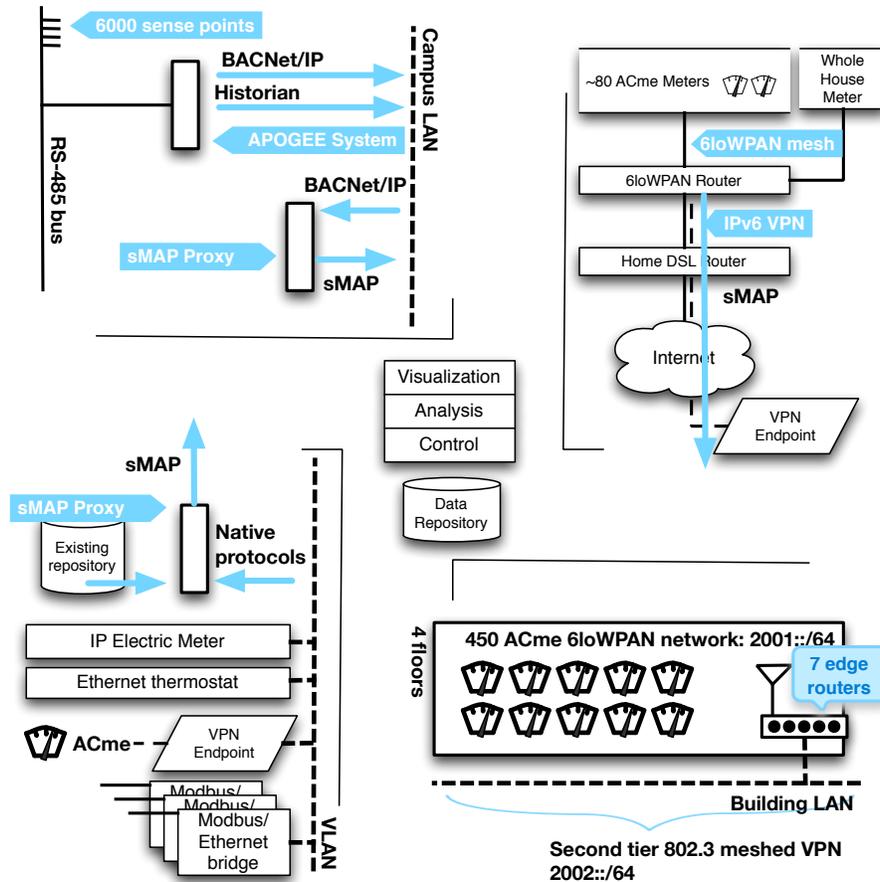


Figure 2. We present four case studies of sMAP implementations. Clockwise from top-left, the first provides a sMAP gateway to a BACnet-based Building Management System which contains several thousand points. The second manages residential deployments around 80 6loWPAN/IPv6 plug-load meters, and whole house data from a separate meter. The third is used in a deployment of 450 discrete 6loWPAN plug load meters in a commercial building environment; and the final deployment is a very heterogeneous building electric testbed environment with many types of instruments.

The **point** corresponds to a physical point of instrumentation, such as a particular sensor or weather station. It is common for a single transducer to produce multiple time-series; each of these are mapped to channels. The type divides channels into meters, sensors, and actuators. Finally a **channel** is used to represent a particular stream of readings; it is a resource containing reading, profile, formatting, and parameter JSON objects.

The reporting resource allows data consumers to install a listener for changes on part of the resource tree. When new readings are received, a sMAP implementation delivers it to subscribers via an HTTP POST. Subscribers specify the set of resources they are interested in via a URL; sMAP sources may deliver either the entire resource or a subset when new readings arrive.

sMAP is sometimes accessed using HTTPS; when that is the case, we use client certificates for authenti-

cation.

3.1 High-level Feedback

In the course of two years of conducting studies and integrating building data using sMAP, we have identified both a number of areas where our ideas about what was important were right, as well as where they fell short. As a result, we have fed back many of our experiences into the design of the next version of the protocol.

Overall, the main positive takeaway of our work has been that **ease of access does spur innovation**. Researchers inside and outside of our group have built a diverse and interesting set of applications on top of sMAP, including improved control loops for HVAC systems resulting in a 40% reduction in energy use [3], a personalized lighting control system which can be retrofitted onto existing construction and reduced energy consumption by 50%, a compute cluster which is grid-aware and

uses information on the electrical supply to schedule jobs to increase the amount of “green” energy used [8], as well as several mobile applications which mash up physical space with the data available about it [6]. Active projects use sMAP data and actuation to enable deep demand response. In all of these cases sMAP provided the layer of architectural separation which allowed developers to make progress without knowing about the underlying instrumentation protocols.

The provisions made for using sMAP in an embedded context have not been heavily used, but supporting this from the ground up is still an important design consideration given that we believe this will become more and more likely given recent developments.

3.2 Detailed Takeaways

3.2.1 Instrument Modeling

Originally, the intention was that *a sMAP source represents an instrument*; that is, each instrument would be present on the network as a single sMAP server. This intuition was correct in many cases; for instance, sMAP sources make data from Dent, PQube, and Veris electric meters, Vaisala weather stations, HeatX steam gauges, Omega iSeries condensate meters, and many others using this model of “one sMAP source per instrument.” However, most of the problems we have encountered using sMAP in these diverse settings have been cases when this assumption does not hold. There are numerous cases where sMAP is now used to represent a collection of either homogeneous or inhomogeneous instruments.

A first case are examples like networks of ACme and Hydrowatch nodes [7, 11]. These are both embedded devices running the blip TinyOS IPv6 stack, and periodically report readings to a network entity. sMAP is deployed as a single application-layer gateway which reports readings from an entire collection of embedded devices. One reason for this is that since sMAP does not provide a discovery mechanism, it is easier for new streams or channels to appear within an existing well-known sMAP server, than to inform all consumers that a new sMAP source representing a single ACme.

The second case of a collection of inhomogeneous instruments occurs when providing a sMAP interface to a much more complicated system like a BMS. There is typically little organization of the data provided by the legacy BMS; however what organization is present is not always easy to shoehorn into the simplistic three-level hierarchy sMAP affords. An iterated version of sMAP would allow the more flexible organization of timeseries.

3.2.2 Archival use

Typically the first consumer of a new sMAP source is an archival database, such as OpenTSDB or StreamFS [1, 9]. The current sMAP design has presented several problems for data consumers. Since data the data transmitted is identified by the resource path on the sMAP

server, the consumer must use the string components of resource names to identify which timeseries the data belong to; the fact that individual channels cannot be renamed has been a problem in some cases.

Furthermore the design of sMAP causes archivers to loose data when it might not be strictly necessary to do so. As part of the residential pilot studies performed with LBNL, we frequently encountered very unreliable home Internet connections. In one deployment, external connectivity was available for only a few hours a day, on average; limited connectivity has been a common use case. As a result of the sMAP design, data consumers may loose data whenever Internet connectivity is interrupted, or the consumer restarts even for an instant; a new version of sMAP should gracefully handle intermittent connectivity and consumer crashes.

3.2.3 Database-backed Deployments

Another common use case has been integrating and importing pre-existing databases. Once users are accustomed to using sMAP for uploading data to a database, they naturally wish to migrate old data into a new system using the same interface, as well many forms of streaming data available on the web but not normally envisioned as instruments; the California ISO wholesale market prices, PV availability, and weather data have all been swept up as sMAP sources.

The *obvius.com* site contains whole-building energy data from much of the UC campus from the past several years, and was used to back-load data for the CEC Building Testbed project. Real-time data is also available in some cases from the front-end data acquisition modules. This case is very typical since implementers first acquired real-time access to the data, and then obtained access to a database which contains historical readings. The current sMAP protocol is an inefficient way to upload data since it essentially requires one HTTP POST per data point.

3.2.4 Discovery, Organization, and Management

The data sources we deal with are typically of moderate complexity. We have not encountered problems where the originating data source’s information model was so sophisticated that we could not map it onto a sMAP resource hierarchy. However, sMAP sources can only be discovered by manually entering their IP address into a consumer. The HTTP hierarchy and SSL certificates for a particular sMAP source are also manually configured (by the sMAP source author).

4 Profile Redesign

As we embarked on a redesign of sMAP, it was relatively clear that although many protocols contain elements which we liked and wished we had, no single protocol unified all of them. There have been some new developments, and we had gained more operational experience interacting with installed systems, which in our experience used either a standard protocol like BACnet

or a proprietary interface. None of these protocols were designed with all of our use-cases in mind; they were not intended to reliably deliver all data when connectivity is intermittent, when the endpoint may go down, or to permit bulk-loading. Overall, these legacy protocols were mostly not designed to operate over the Internet; as we discovered, it is not a good idea to expose your BACnet/IP control system on a public network.

On the other hand Internet protocols like XMPP and SNMP support to some extent the kinds of discovery, security, and manageability features which we are interested in but do not meet the range of our needs better than sMAP 1 [10]. WattMiser and Pachube are RESTful systems being developed along the same lines as sMAP; sMAP is distinguished by a limited architectural focus with careful attention to the protocol itself, rather than embedding it within a larger system. [4, 2]

Therefore, we iterated on the sMAP design, keeping or improving the ease-of-use for both the sMAP source implementer as well as as the data consumer, while extending it in some ways to better support our new use cases. We are currently migrating our existing feeds over to the new design, and have completely implemented and released the new specification along with a small set of base drivers at <http://eecs.berkeley.edu/~stevedh/smap2/>.

4.1 Data Representation

Several changes to the objects being represented allow us to efficiently transfer bulk readings and organize data with associated metadata.

1. The primitive streams we are representing are Timeseries. Time series consist of sequences of readings from a single channel of an instrument and associated metadata. Time series may be organized by the sMAP implementer into Collections, representing logical organizations of the instrumentation; this organization is reflected in the resource hierarchy exposed by the sMAP HTTP server. This allows us to preserve existing, external namespaces when integrating existing sources.
2. sMAP 2 supports adding Metadata to either Timeseries or Collections to better support integrating existing data sources where the Metadata should be transferred along with the data. This was not available in sMAP 1.
3. Timeseries are durably identified by `uids`; thus it is possible to change any other attribute of the series. This results in an overall representation which is simpler and more uniform overall.
4. Schemas are expressed as Avro schema. If clients indicate that they can accept this schema via the `HTTP Accepts:` header, servers may return Avro-packed results with the appropriate Content-encoding set. Clients should be made available to

take advantage of this. This permits the efficient transfer of large numbers of objects.

4.2 Metadata

sMAP v1 supported very limited metadata. The largest conceptual change made in version 2 is extend the metadata model to better support integrating existing systems, as well as creating systems where the implementers have significant metadata they wish to tag their data with. We make several simplifying assumptions:

1. Metadata only has meaning when it is attached to data, where “data” is one or more time-value pairs (data points). We logically consider each data point to be attached to a full set of metadata; previously any metadata was attached to a channel, and so could not be changed without effecting past readings.
2. Metadata is inherited within a sMAP source: the metadata for any data point consists of the Metadata for that Timeseries, along with the Metadata all Collections in the recursive parent set in the resource hierarchy. If the same piece of metadata is present at multiple places, the metadata “closest” to the Timeseries (*i.e.*, deepest in the tree) is used.

With this system, we can both efficiently compress the metadata when transmitted over the network by only transmitting changes, and also resolve many issues dealing with what happens when resources move within the resource hierarchy: after a move, all Timeseries which were moved simply receive new metadata. Since the Metadata can only apply to actual data, there is never an ambiguity about which piece of metadata applies to a point, and it is always safe to re-send all metadata. The consumer is responsible for detecting whether the re-sent metadata is a change or merely an unnecessary retransmission.

4.3 Reporting

Allowing sMAP to be used to import data and to improve reliability requires only modest protocol changes.

1. sMAP 2 allows the provisioning of a list of delivery endpoints (URIs), which will be attempted in order; this allows consumers to implement fail-over with support from the sMAP server.
2. Since a Timeseries references all metadata applying to it, sMAP can be used to upload archival data by uploading a document containing both data and metadata.
3. The implementation now supports “reliable reporting,” when given sufficient local storage. sMAP servers treat the stream of outgoing readings as a log, and only truncate the log when it has been successfully delivered to the consumer. If desired, the log may be buffered to backing store so it can survive crashes and restarts.

4.4 Proxies

sMAP 2 servers may have the ability to reverse-proxy a number of other sources, and place them inside of a Collection. This is beneficial for a variety of reasons; the set of proxied source may represent some logical set of sensors such as all the sensors in a car, and allows consumers to subscribe to a single source rather than a set. Furthermore, it enabled the embedded use-case since it provides rate-limiting, access control, and buffering at a gateway rather than at the instrument.

Discovery can be accomplished either by manual configuration, or by an automatic mechanism; the sMAP standard library will soon support the discovery of other source using Zeroconf and buffering their readings within the proxy. For example, using this specification is possible to create an aggregate sMAP source which locates and republishes all of the instrumentation on a subnet automatically; alternatively, an administrator can manually configure a proxy to collect and re-expose all instrumentation from a particular building. This allows the architectural flexibility to place services which are best centralized such as authentication and access control in a proxy, and mediating access to the back-end sMAP sources.

5 Conclusions

As a result of our experiences, we are able to use sMAP much more cleanly and broadly than before; for importing large quantities of data, transferring metadata between systems, and in deployments with intermittent connectivity. We can create virtual sMAP proxies which take a large number sMAP sources and represent them as a single one while continuing to meet our original design goals for the system, integrating diverse existing instrumentation into a single unified view. Using this system, a personalized lighting control application integrates data from two BACnet systems and saves 50-75% of lighting energy relative to a baseline, while other thermostat research uses data from wireless sensor nodes, the building management system, weather forecast data, historical data, and a programmable thermostat to build implement their algorithm in a real room and reduce energy consumption by 40%.

sMAP has been updated, based on significant real-world deployment experience into a protocol for more generalized publishing of time-series data while continuing to meet the original design goals of being simple, easy-to-consume, and implementable on a wide range of devices. Since a great deal of work is currently being developed centered on optimizing buildings as a system, we continue to believe that a simple, well-considered system with the proper tools and abstractions will be a key enabler for the broad deployment of advanced technology and research in buildings.

Acknowledgments

Special thanks to Albert Goto for much helpful assistance. This work is supported in part by the National Science Foundation under grants #CNS-0932209, an NSF Graduate Fellowship, the FCRP MySyC Center, Intel Corp, and Samsung Corp.

6 References

- [1] Opentsdb. <http://www.opentsdb.net>.
- [2] pachube. <http://www.pachube.com/>.
- [3] A. Aswani, N. Master, J. Taneja, D. Culler, and C. Tomlin. Reducing transient and steady state electricity consumption in hvac using learning-based model predictive control. IEEE Computer Society, 2011. To appear.
- [4] R. S. Brewer and P. M. Johnson. WattDepot: Enterprise-scale, sensor-based energy data collection and analysis. Technical Report CSDL-10-06, Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822, May 2010.
- [5] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler. smap: a simple measurement and actuation profile for physical information. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, SenSys '10*, pages 197–210, New York, NY, USA, 2010. ACM.
- [6] J. Hsu, P. Mohan, X. Jiang, J. Ortiz, S. Shankar, S. Dawson-Haggerty, and D. Culler. Hbci: human-building-computer interaction. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building, BuildSys '10*, pages 55–60, New York, NY, USA, 2010. ACM.
- [7] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler. Design and implementation of a high-fidelity ac metering network. In *IPSN '09: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 253–264, Washington, DC, USA, 2009. IEEE Computer Society.
- [8] A. Krioukov, P. Mohan, S. Alspaugh, L. Keys, D. Culler, and R. Katz. Napsac: The design and implementation of a power proportional web cluster. In *Proceedings of the 1st ACM SIGCOMM Workshop on Green Networking*, New York, NY, USA, 2010. ACM.
- [9] J. Ortiz. A system for managing physical data in buildings. Technical Report UCB/EICS-2010-128, EECS Department, University of California, Berkeley, Sep 2010.
- [10] A. Rowe, M. Berges, G. Bhatia, E. Goldman, R. Rajkumar, and L. Soibelman. Demo abstract: The sensor andrew infrastructure for large-scale campus-wide sensing and actuation. In *IPSN*, pages 415–416. ACM, 2009.
- [11] J. Taneja, J. Jeong, and D. Culler. Design, modeling, and capacity planning for micro-solar power sensor networks. In *IPSN '08: Proceedings of the 7th international conference on Information processing in sensor networks*, pages 407–418, Washington, DC, USA, 2008. IEEE Computer Society.