Similarities in Monitoring Civil & Computer Infrastructure

Christopher Peplin (peplin@cmu.edu)

May 29, 2011

1 Introduction

Civil engineers were some of the first to take advantage of the information revolution around the time of plugging semi-conductor prices. As early as the 1970's, major infrastructure projects were wired with real-time monitoring and control systems to lower costs and improve safety and operating efficiency. This trend continues today.

Meanwhile, computer system architecture shifted three major times in these past 40 years. Individual, personal computers gave way to mainframes and thin clients. The tide shifted back to powerful end-user clients in the 1990's. Now, the cloud and web services embody somewhat of a throwback to the mainframe era. Computer networks remained quite small in comparison to some infrastructure sensor networks, at least until until the rise of the Internet as we know it. Now, some computer systems rival the largest infrastructure projects in size (if not cost). These systems have some of the same issues with monitoring and statistics analysis.

An example of a relatively new, distributed computer application are massively multiplayer online games. The game publisher Blizzard has a vested interested in tracking the users of their online game *World of Warcraft* for billing, game balancing and to plan for future expansions. In order to scale the game world in a reasonable fashion, the developers split up the environment into thousands of shards. The gameplay statistics must make it back to a centralized location at Blizzard eventually, but the fractured architecture doesn't lend itself to a simple solution.

The data center ecosystem itself is also one massively distributed system, encompassing thousands of nodes across a diverse geography. In short, distributed systems are more common than ever and the importance of monitoring, tracking and accounting hasn't waned.

These applications are what bring the computer world more in line with the monitoring situation in civil infrastructure. Civil engineers and government organizations in charge of projects such as roads, bridges, oil & gas pipelines and waterways have been struggling with monitoring some of the earliest distributed systems. These are not distributed in the same computing sense of the word; they are often entirely offline and unpowered. For decades, their data has been gathered (often inconsistently) by hand. The engineers tasked with accounting for trillions of dollars of public assets and physical systems are dealing with what could be viewed as widespread network unreliability and wholly unreliable nodes.

2 Infrastructure Monitoring

The operators and designers of civil and computing infrastructure have a keen interest in collecting knowledge of the behavior of the systems. Infrastructure management is increasingly data-driven, requiring ever more monitoring. It is also useful for providing a global-level view of the condition of a system. This can often be a good indicator of when and where a failure occurred, giving operators prime candidates for further investigation. The three primary motivations for infrastructure monitoring are asset management, safety and operations.

Asset Management Asset management is an increasingly popular (and in some cases required) strategy for managing an organization's physical assets. It includes performing continuous inventory, risk modeling, and life-cycle and condition assessment. All of these rely heavily on computing for data collection and analysis. A monitoring system more tightly integrated with asset management tools can give more accurate predictions — e.g. automatically collected gas measurements in power transformers can be taken nearly continuously, compared to quarterly or yearly manual inspections. On the computing side, data centers are becoming increasingly heterogeneous and asset management is important, albeit to a lesser degree.

Safety Many infrastructure components require constant inspection to ensure safe operation. Components have different safe operating ranges for various statistics (temperature, pressure, cycles, etc.) and the more accurate and up-to-date this information is when it reaches the control center, the better. There are fewer safety considerations for computer applications, but a parallel metric is the reliability and robustness of users' data storage.

Operations Normal day-to-day operations of some types of infrastructure previously required many employees on-site to monitor and operate different components. With a two-way communication system between infrastructure and control room (i.e. one that sends commands and receives metrics), these jobs can be done more efficiently with fewer people and with a better sense of the status of the system as a whole. For example, operators on two ends of a pipeline many have a difficult time determining the status of an issue occurring in the middle. With remote monitoring, these parties can deal with the situation all from the same room. The same goals hold true for data center operations, and especially for geographically distributed software systems.

2.1 Sensing

Modern research prefers wireless sensors over wired [12]. These systems are harder to disrupt, which is of greater concern when the system is out in the open and many network hops away from the central office. A widely distributed wired network involves a significant amount of extra infrastructure, and damage to the infrastructure being monitoring often implies damage to the monitoring system itself. Wireless systems have the advantage of being easier to deploy, as they can self-organize into ad-hoc networks (see Figure 1) as long as they are within range of another sensor node. Connectivity problems also tend to be isolated to individual units, and are easier to troubleshoot [12].

Recent projects have taken a page (knowingly or not) from tools like the computer network monitoring application Ganglia (see Figured 2) and now use a unified data format, regardless of sensor or data type [12]. For example, a single update could consist of:

- Type of data (1 byte)
- Geographic coordinates, determined via GPS or inferred via signal strength of other nodes
- Network address
- Actual data (4 bytes)

These monitoring networks self-organize into a dynamic hierarchy of nodes based on their placement and capabilities. There are generally three types of nodes deployed:

- Basic sensor node
- Communication relay node — collects data in its 1- or 2-hop neighborhood
- Data Discharge Node — forward results to the Network Control Center, i.e. the one with a connection to the Internet

Whereas in computer system monitoring, each node generally has similar capabilities for communication and sensing, the role of these nodes are bound by their physical capabilities. Hierarchical organization becomes a simpler problem of guaranteeing a wide enough dispersal of communication relay and data discharge nodes to reach all of the levels of the hierarchy, compared to the somewhat arbitrary trees made in computer networks. See Figure 3 for an illustration of a common sensor node hierarchy.

A critical feature of these networks is the scalability of their performance: "[s]uch a dense array must be designed to be scalable, which means that the system performance does not degrade substantially, or at all, as the number of components increases" [17]. The fact that each node only communicates with its closest neighbors is a strong indication that such a design is scalable to tens of thousands of nodes.

2.2 Collection & Processing

A significant impediment to a monitoring system's success is the amount of processing required to put the data into a useful form. The amount of data recorded for infrastructure components can be extensive, reaching multiple terabytes of data per day, but "[m]uch of such data [is] being collected but not used because processing is too costly" [4]. Even with the recent advances in "massively distributed smart sensors" [4], infrastructure managers still lack a general computation framework to explore, experiment with and analyze the data. Controls software vendors are driven by the requirements set forth by operators, but these operators' demands are often only responding to what vendors have given them in the past.

Data collection, storage, processing and querying is a general enough task that it could and should be standardized across all areas of infrastructure. Computer systems have converged around a few simple data formats and protocols to ensure interoperability and save time by avoiding re-implementing existing features. A few of these include:

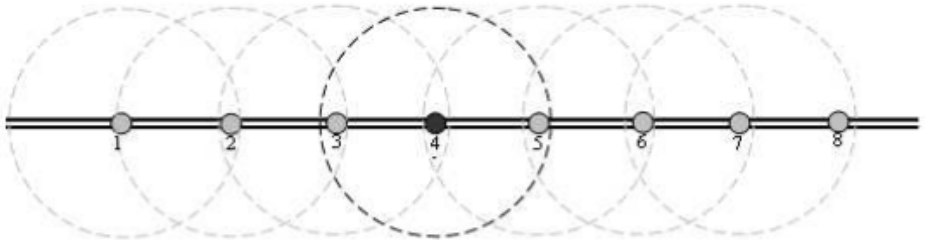

Figure 1: Wireless sensors along a pipeline form an ad-hoc wireless network and communicate only with nearby neighbors [12].

Figure 2: A screenshot of the monitoring interface for Ganglia, a computer network monitoring tool.

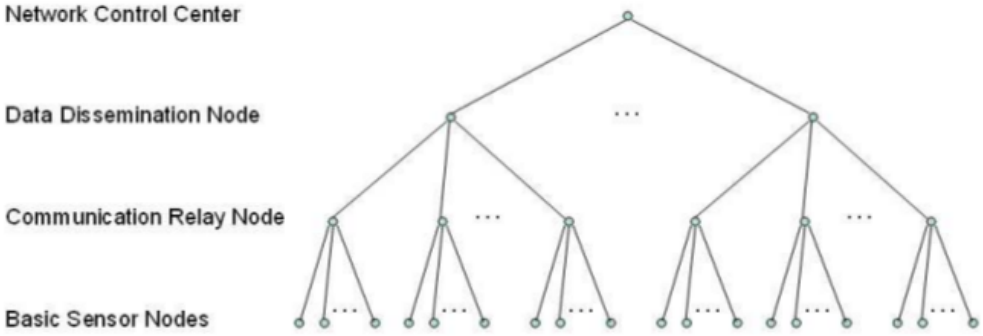

Figure 3: An example hierarchy for a network of sensor nodes [12].

- The Hypertext Transfer Protocol (HTTP) for exchanging data between remote systems, which contributed to the success of the World Wide Web.
- XML and JSON for encoding data into a predictable and quickly parseable format, which contributed to the enhanced interactivity of Web 2.0.
- SQL for querying relational databases, which made user customization possible on the web.
- HTML, Cascading Style Sheets (CSS) and JavaScript for user interfaces, which led the way for browser-based applications like Google Docs.

An example monitoring implementation using these technologies is Nagios, a tool to monitor computer grid infrastructure.

> "The Nagios distribution provides only the basic set of sensors, but custom sensors can be developed by using any existing programming language. This means that Nagios can be used for monitoring virtually anything as long as appropriate sensor can be developed" [11].

Nagios provides a uniform interface to the monitored statistics, on top of which any number of processing and analysis applications can be built. Nagios is not intended for the resiliency that something like an electricity grid would require, but many smaller infrastructure projects and those with less critical tasks could likely use this system without modification.

2.3 State of the Art

One recent project at the University of California, San Deigo set out to implement an extensible structural health monitoring platform using some of these open technologies [5]. They proposed a system with many components, including:

- Networked sensor arrays
- A high-performance centralized database
- Computer vision analysis
- Physics analysis
- Visualizations that allow comparison between experimental and numerical simulation data
- Modeling and risk analysis

This was an ambitious project, attempted to provide a standard way "to measure, acquire, process, and analyze the massive amount of data that is currently coming on-line (not to mention the terabytes of streaming data that will inundate potential users in the near future) in order to extract useful information concerning the condition assessment of the monitored structures" [5]. Unfortunately, the project's web portal is no longer online and no further information about the project could be found. A likely cause of failure is that the system tried to combine too many components into a single piece. Monitoring and data querying is one of the few tasks that can be completely generalized across industries. Considering that computer vision analysis and risk modeling will likely change substantially between a pipeline operator and telecommunications operator, these systems should be left to the various industries to implement as they see fit. As long as the data collection system provides a uniform interface to query the monitoring statistics, it can still be of great value.

2.4 SCADA

Supervisory Control and Data Acquisition (SCADA) is an all-encompassing descriptor for real-time communication systems that connect infrastructure to operators. SCADA systems can be uni- or bi-directional — that is they can both send monitoring data back to the operator and propagate control commands to infrastructure. These types of systems are popular in industrial environments as well, as a way to monitor and control heavy equipment.

There are many existing SCADA-style systems (an estimated 150-200 different protocols [10]), dating back to the late 1960's. SCADA is very popular with power utilities as a mechanism for coordinating generation capacity among power plant operators; the generators must react very quickly to changes in load, and an automated communication system is the only way to respond fast enough. In the few first decades of their existence, SCADA systems were primarily developed in-house and extremely customized for specific use cases. As computer software firms spawned in the 1980's and 1990's, more infrastructure operators switched to purchasing their controls software from existing vendors. This has both positives and negatives.

Standard Networking Protocols In the last 20 years, SCADA software vendors have migrated towards using the standard Internet Protocol (IP) for communication. This is a widely accepted protocol in computing systems, and the success and diversity of the Internet proves its flexibility. The other most popular protocols are DeviceNet and ControlNet [10].

Previously, operators would find themselves purchasing very large, completely proprietary systems

designed to monitor just a single attribute of their system - for example, monitoring electrical power phase [1]. An operator's terminal could be littered with different applications, each for a very specific part of the system. The integration between these components was poor because of the difficulties in sharing data among them.

Once concern with using IP is that now, running on the same network as consumers and potentially malicious attackers, there are additional security risks for a SCADA system, especially when ill-timed or unauthorized control messages could have disastrous consequences. The security issues are discussed further in Section 2.4.1.

Standard Interfaces Familiarizing operators with a SCADA user interface is a critical step in deploying a successful system, and one that was given short shrift in the first few decades of deployment. Along with most of the population of the 1960s, very few operators were familiar with computer interfaces and thus it "was imperative that the [SCADA] operator interface be graphical in nature to provide the dispatchers with a similar look and feel to what they were used to" [7]. User interfaces issues are discussed further in Section 4.

2.4.1 Security & Confidentiality

As mentioned, the privacy and security of a SCADA system is especially of concern when using networks shared by others. In general, "[m]onitoring and controlling these systems is an enormous undertaking, requiring constant supervision" [6]. They must be architected to avoid cascading failures so connectivity issues in one area of the Internet do not effect the stability of safety of a piece of critical infrastructure. SCADA systems are some of the most diverse and complex integrated systems, commonly containing "as many as 50,000 input/output modules for data collection" [6]. Since infrastructure systems are so interdependent, failure or damage in one operator's infrastructure may cause a larger part of the infrastructure to become unstable. Many types of infrastructure are dependent on telecommunications to keep their SCADA systems running, and if standardization and collaboration continues across industries, the integration points will only become more prevalent.

Knowing this, systems using a standard shared network with open protocols are not without fault. They allow for more efficient operation and potentially more collaboration, "but it also exposes the safety-critical industrial network to the myriad security problems of the Internet" [10].

It's important to distinguish between the control and monitoring halves of SCADA, as they have very different security requirements. Some types of infrastructure have a much greater need for control than others, who are satisfied with mostly monitoring. It would be a mistake to require both systems to have the same level of protection (although they both require robustness).

Some common strategies for improving the security of SCADA systems on an open network are:

- Make certainly all network traffic is protected using common Internet security measures such as Transport Layer security (TLS)
- Establish industry-wide data security practices.
- Educate operators on these best practices. Many reported security breaches were the result of operator error.

Openness Before fretting too much about security and spending additional money, the operators should also consider the true sensitivity of the data they are collecting. The surface appeal for protecting data is great, but often the data would not expose anything a watchful eye could not detect by manual inspection. The true motivation for protecting data may be different. For example, the Federal Energy Regulatory Commission removed previously accessible statistics and documents from their website in the name of security, "but a 2003 investigation strongly suggests that advancing the economic interests of favored industries or keeping executive actions from being scrutinized are the actual motivations" [14].

Operators need a standard and objective process for determining the sensitivity of information, and should err more on the side of release than protection (depending on the risk involved). The reason is that "[r]evealing data on the vulnerabilities of certain kinds of infrastructure can be a net benefit when the target would be inadequately defended absent that revelation. A series of GAO reports about weaknesses in defensive measures at commercial nuclear power plants, for example, played a key role in overcoming industry resistance to stricter security standards" [16].

Computer engineers have long appreciated the massive amount of cognitive power accessible through the Internet, people ready and willing to tackle difficult problems. The same is true for infrastructure issues, and "government officials should release organizational information whenever society is more effective than terrorists at utilizing it" [16]. This level

of transparency could both force utilities to "internalize more fully the costs of attacks" [16] and encourage more self-interested regulatory action to protect from attack in the first place.

Attacks are so rare that as a private company operating a piece of infrastructure, it doesn't often make business sense to spend money on protection. The government lacks the regulatory power to make these industries account for the full social costs of these events — and up to 90% of the nation's critical infrastructure is privately owned and operated [19]. Private industry, even when operating publicly owned infrastructure, lacks strong incentives to properly protect it.

In computing, open source cryptographic algorithms are more trusted and widely used than closed-source proprietary complements because their vulnerabilities have been exposed and patched. They are not more susceptible to attack solely because of increased knowledge of their inner workings. Security by obscurity is insufficient; non-disclosure is almost certainly an admission of the existence of vulnerabilities, but it doesn't guarantee any of them are being resolved.

Non-disclosure risks more than just the reputation of the company in question - it risks public safety and the structural integrity of critical infrastructure. Companies certainly recognize this, and is a part of the reason for protecting data - anything exposed to the government could potentially show up in civil enforcement action against the company.

Of course, there must be a balance between openness and security. Data on nuclear fuel distribution and quantity is protected because of the potentially extreme consequences from unauthorized access. However, the positives of openness and possibilities of early vulnerability discovery likely outweigh any risk in most other industries.

A monitoring system based on open standards could easily support a mechanism to filter and aggregate operating statistics for public consumption. Real-time data will likely always be considered too sensitive, but there should be a clear path from SCADA systems to a more accessible (but still digital) version of the data, one without any extra costs beyond initial setup.

3 Comparison

This section compares the culture and basic monitoring styles of civil and computer engineering.

3.1 Culture

Civil and computer engineers come from different backgrounds and don't have much shared history. The two industries have evolved with very different major players and cultures.

3.1.1 Traditional Civil Engineering Ethos

The first 20 years after monitoring systems were first widely deployed can be described as being embodied by a traditional civil engineering ethos. With the newfound availability of computers and data storage, this was a period of great change in design and planning.

Process & Information Security From the start, information security was a critical concern for operators. They also intended to keep specific operation process details within a company or industry. These were viewed as details that didn't need to be exposed, for the good of the company and the infrastructure.

Custom Software Software development was a very new field, and many operators found themselves hiring consultants to develop custom monitoring software for their specific use case. Each operator had slightly different requirements, even within the same industry, and at the start there wasn't much interoperability or data sharing.

Expert Interfaces For industries with generations of existing workers, there wasn't a clear path to designing proper user interfaces or training the operators. One approach is to make the interface as visually similar to the physical component. Others focused on more tabular displays of data, similar to what was previously a hand-written spreadsheet. In both cases, it often took an expert understanding of the system to be able to interpret the (often terse) monitoring system displays.

3.1.2 Computer Engineering Ethos

At the same time, the computer engineering world was rapidly expanding and in hindsight had a few basic principles.

Openness In part due to the early ties between computers and academia, some of the first widely used applications were open source — this means that the underlying source code for the system is provided free of charge to view and modify. This allowed

new developers to quickly see how existing systems worked, and to assist in the development and expansion of applications.

Standard, open communication protocols also played an important role early on, especially with the development of the Internet. Without a freely implementable communication standard, the World Wide Web could not have succeeded.

Finally, openness has been a core tenant of security from the start of computing. Even today, many large software companies share fine-grained details of the operations of their systems in order to expand knowledge in the field and potentially reap the benefit of open source contributions to their projects.

Humanized UI User interface design is an important part of everything to do with computers - without the interface, there is very little of interest in networked applications. This is opposed to infrastructure, where the physical components exist in the world regardless of if they are monitored by a communications network.

The interfaces range from technically demanding (like the early civil infrastructure displays) to layman friendly. A core principle is flexibility, however, meaning that with time and effort, almost every system can settle on a good UI.

3.1.3 Modern Civil Engineering Ethos

As SCADA matured over the last 20 years, the ethos of engineers working on monitoring civil infrastructure has changed. Some of the ideas from computer engineering have migrated over, and the further development of a strong software industry improved standards and interoperability.

Industry-level Standardization Most infrastructure industries have some general standards and guidelines for SCADA systems. Specific software vendors have become more popular in certain areas, meaning that an operator moving from one power utility to another is more likely to find familiar nomenclature and interfaces. For example, in 1993 the American Petroleum Institute issued a set of overall SCADA guidelines to reconcile a huge diversity of interface styles for pipeline monitoring. Infrastructure management is founded on a set of core ideas and tools that can be applied to many different industries with only a few changes (new expert consultants, different Weibull curves, etc.), and the same is true for monitoring systems.

Third-party Software Most monitoring software is now developed by third-party software vendors such as Seimens and Rockwell Automation. While these systems still have some proprietary components, communication protocol standards now have much more support.

3.2 Active versus Passive

Computer network monitoring systems are typically monitoring other software system and the monitoring is inherently very tightly woven with the application itself. This allows for a passive monitoring approach that gathers its data directly from the application, or by politely eavesdropping on network activity. This isn't possible in non-computing infrastructure where the "application" is something physical, without any explicit bits associated with it [9].

Both types of monitoring accomplish the same end goal, but passive monitoring in computer networks is used primarily in an attempt to lower the performance overhead of monitoring. Civil infrastructure will rarely if ever have the same problem (since wireless sensors can be positioned as to not obstruct normal operation), but the difference could lead to another divergence in monitoring techniques.

4 User Interfaces

4.1 State of UI in SCADA

The purpose of SCADA is to "allow operators to monitor and control systems" [6], so the presentation of data is a critical component that should receive equal consideration with sensor hardware and data security. Unfortunately, this hasn't been the case for many SCADA developers. A 2005 safety study by the National Transportation Safety Board [18] on SCADA for liquid pipelines found that inconsistent and problematic user interfaces were the cause of many accidents, challenging the claims that the systems are a success. The study concluded:

> "The principle issue in the SCADA-related accidents investigated by the Safety Board was the delay in a controller's recognizing a leak and beginning efforts to reduce the effect of the leak. SCADA factors identified in these accidents include alarms, display formats, the accuracy of SCADA screens, the controller's ability to accurately evaluate SCADA data during abnormal operating

> conditions, the appropriateness of controller actions, the ability of the controller and the supervisor to make appropriate decisions, and the effectiveness of training in preparing controllers to interpret the SCADA system and react to abnormal conditions" [18].

This highlights the importance of good interface design, as most of these issues are simply the result of controllers not understanding the information presented to them by the monitoring system. These are operators who interpret a graph incorrectly, dismiss repeated warnings because of frequent false positives, and those who can't get a good sense of the true health of the system in the field based on tabular data. An example SCADA interface is pictured in Figure 4.

The study examined the role of SCADA systems in 13 pipeline accidents from 1992 to 2004, and found that a SCADA system contributed to the accident in some form in ten of them. Implemented poorly, a monitoring and control system can harm instead of help.

These systems should strive to control as much as is reasonably possible automatically. A common complaint about these systems is that false alarms are so frequent, that operators have a difficult time distinguishing the real alarms when they do occur. These annoying alerts can lead to a serious signal being ignored for a long time by an operator who has tuned them out, or a misdiagnosis of the root cause. As many as possible of these false positives should be handled by a low level of automatic control operations done by the computer, even if it means stopping system components for a brief period when not strictly necessary. Recent research on data center fingerprinting [3], which identifies new incidents based on data collected during historical events, could be applied to such automatic failure identification and repair. The human operator should be brought into the loop only when a bigger system problem is identified and automated response is not sufficiently subtle or intelligent.

The standardization of colors and symbols within (and even across) industries is also required to minimize re-training and misinterpretation. This hasn't always happened, primarily because a lot of SCADA system development "occurred company-by-company due to the unique characteristics of each company's operating practices and other computer systems. For example, one company may use red to show an operating pump while another may use green" [18]. Consistent, appropriate use of color is one of the major tenants of good user interface design and all existing displays should be evaluated to make sure they meet this standard. Other UI improvements that the NTSB suggest are:

- Real-time comparison with historical data to recognize abnormalities, especially in areas the operator many not be explicitly trained.
- Integrate company safety procedures into control systems to guide operators down the correct resolution path.
- Extended operator training — operators should be experts in the system.
- Increase contrast between foreground and background colors.
- Minimize the quantity of colors and make sure their meaning is consistent between screens.
- Target 40% blank space on screens to minimize clutter.

4.2 User Interfaces & User Experience

Beyond incremental improvements, civil infrastructure interfaces have a unique opportunity to push the envelope of interfaces. These systems have very wide market penetration, and whatever interface they provide (good or bad) has a tendency to become the industry standard out of familiarity. A potentially more revolutionary interface change could take advantage of an immersive virtual world instead of buttons and charts. Research regarding different physical interfaces for computers, and the effect of integrating virtual elements with the real world found that they can "create a system that leverages the human mind's pattern recognition skills to detect anomalies on a live running network" [8].

Naturally, due to the popularity of computer and video games among computer network operators and researchers, a few attempts have been made to create such an immersive visualization for network monitoring. The primary challenge is to carefully select "input and output metaphors within [the] real time 3D virtual environment," [8] and it proved especially challenging because there is no obvious analogous physical element for network constructs — the activity is all bits flowing on wires. Physical infrastructure monitoring is not bothered as much by this issue, since at the root of most infrastructure monitoring task is some physical object. Metaphors are not required.

An immersive, humanized environment offers more opportunities for pattern recognition of data that would be shown in an unsuitable format on a

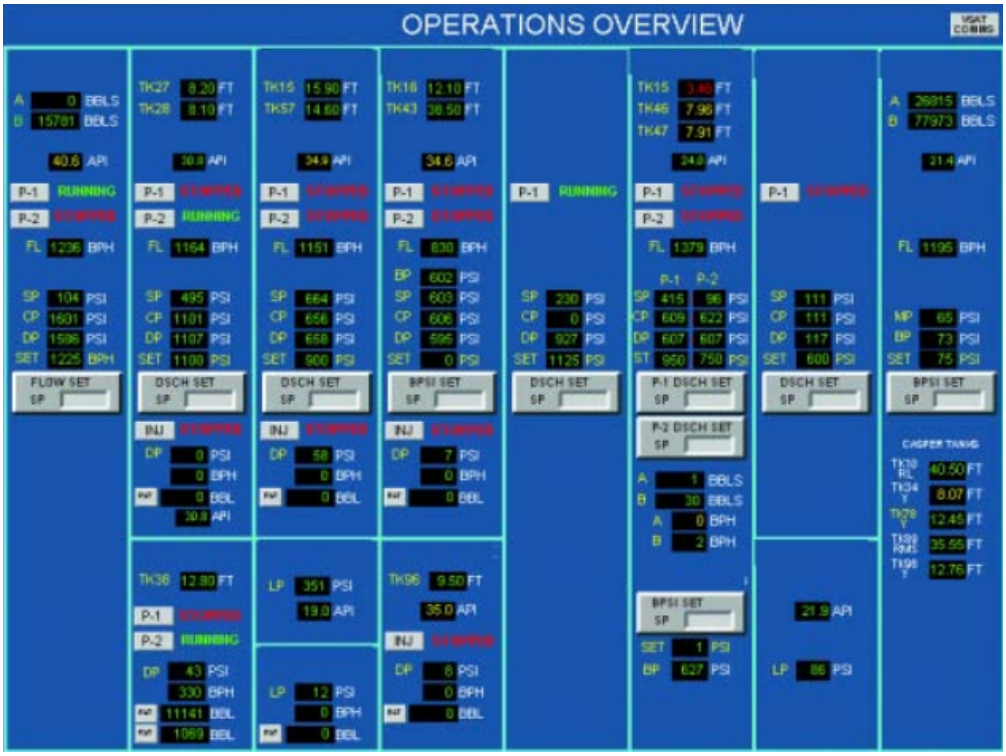

Figure 4: A screenshot from a SCADA system for monitoring a pipeline [18].

traditional display. Immersive virtual environments can also improve collaboration between operators. Currently, operators working together on a problem must communicate through voice or video. This is because it's difficult to tell at a glance what an operator is doing at a computer terminal. Inside a virtual world, the operator's current activity and status can be made obvious by the actions of their character. If the operator is opening a valve on the pipeline, they will appear doing just that inside the environment. In the control center, they're sitting in an office chair as always.

4.3 Serious Games

The idea of serious games is a good example of a daring use of existing interfaces in a novel way.

> "The term serious games has developed as a rebuttal to the idea that games are purely for leisure purposes and its use goes back to Plato's work on the importance of play as a teaching method. Recently, the serious games movement has emerged from academic communities identifying the power of play for supporting non-leisure activities such as education and training" [13].

Beyond training, serious games can also be used for monitoring and interacting with live environments in real-time. Using a first- or third-person viewpoint into a world that looks very similar to the actual physical components is a more natural way to get the sense of a system's status.

A core requirement of an infrastructure monitoring version of a serious game is to avoid simplifying critical components to the degree that accuracy is lost. The interface must certainly contain abstractions, lest it become a micromanagement simulator, but it cannot ignore real-world issues that other training simulations tend to ignore.

4.4 Modern Video Games

Thankfully, a lot of work has already been done in developing immersive 3D virtual environments in modern computer and video games. For the purposes of entertainment and creative expression, game developers are creating increasingly detailed worlds. Some of the important features that could be applied to infrastructure management are highlighted by a few recent titles:

- Valve's Half-Life 2 (2004) — provides advanced real-time physics simulation that could be used

Figure 5: A screenshot from Mirror's Edge, a video game with expansive urban environments.

for combining real-time sensor data and prediction models.

- Electronic Arts' Mirror's Edge (2007) — includes expansive, detailed urban environments that could be used for building and bridge inspection. See Figure 5 for a screenshot from the game.
- Gas Powered Games' Supreme Commander 2 (2010) — gives top-down strategic control of thousands of military units, which could be used for routing traffic on busy streets, waterways and in the air.
- Rail Simulator Development's RailWorks 2 (2010) — gives complete control of accurately modeled passenger and cargo trains. With a connection to a SCADA system, this game could be useful almost out of the box.

Most state-of-the-art game engines (the software that renders the world and controls the behavior of objects in it) are not available to the public directly, but many expose a flexible interface that allows anyone to modify the gameplay. Computer games are often popular well beyond their initial release because of large communities of gamers and developers creating "mods" that twist the game engine into completely new forms. Some examples that suggest the possibilities for infrastructure management mods include:

- Gary's Mod (for Half-Life 2) — strips down the game to its core physics engine to allow creating devices as varied as operational submarines, construction cranes and rocket ships.
- Empires (for Half-Life 2) — extends the formerly single-play, first-person only game to include a commander with a top-down view of the battlefield who can give orders to players exploring the world. This style of game closely matches the management hierarchy of many infrastructure operators.
- BusMod (for Grand Theft Auto 4) — in a game built for causing violent mayhem in an urban environment, this mod highlights a more mundane feature of the city and lets the player drive a bus route.

A few open-source game engines are available, if the modding interface isn't sufficient for any specific monitoring task. Some research has been done on using one of these engines for a computer network monitoring task [8].

5 Conclusion

Civil infrastructure is a notoriously underfunded part of our society. The ASCE's yearly infrastructure report cards [15] have been giving failing grades for over a decade, yet overall investment in infrastructure hasn't risen. The improvements proposed in this paper clearly cost money, something which few infrastructure managers have available for non-critical upgrades.

However, remember that the promise of real-time monitoring and SCADA was a decrease in total lifetime cost of infrastructure. This promise was largely fulfilled. With some up-front investment, more efficient operation and monitoring can potentially save a lot of money. After 40 years of SCADA systems, it's time to move into the next major phase of technological advancement, which holds the same cost saving promise. Monitoring systems succeeded, but they risk falling behind like the rest of America's infrastructure without proper consideration.

The next evolution in monitoring must consider the true target of the system: the users.

> "Engineering makes things for end-users. Accounting makes things for markets, demographics and consumers. Design makes things for people" [2].

Engaging interfaces are not reserved exclusively for entertainment and pleasure, and neither is good design.

References

[1] J. Bertsch et al. "Experiences with and perspectives of the system for wide area monitoring of power systems." In: *Quality and Security of Electric Power Delivery Systems, 2003. CIGRE/PES 2003. CIGRE/IEEE PES International Symposium*. Oct. 2003, pp. 5 –9. DOI: 10.1109/QSEPDS.2003.1259313.

[2] Julian Bleecker. *Design Fiction: A Short Essay on Design, Science, Fact and Fiction*. Near Future Laboratory. Mar. 17, 2009. URL: http://www.nearfuturelaboratory.com/2009/03/17/design-fiction-a-short-essay-on-design-science-fact-and-fiction/.

[3] Peter Bodk et al. "Fingerprinting the datacenter: automated classification of performance crises." In: *EuroSys*. Ed. by Christine Morin and Gilles Muller. ACM, July 10, 2010, pp. 111–124. ISBN: 978-1-60558-577-2. URL: http://dblp.uni-trier.de/db/conf/eurosys/eurosys2010.html#BodikGFWA10.

[4] Peter C. Chang, Alison Flatau, and S. C. Liu. "Health Monitoring of Civil Infrastructure." In: *Structural Health Monitoring* 2 (2003), p. 257.

[5] A. Elgamal et al. "Health Monitoring Framework for Bridges and Civil Infrastructure." In: *Structural Health Monitoring 2003. from diagnostics & prognostics to structural health management: proceedings of the 4th International Workshop on Structural Health Monitoring*. Stanford, CA, Sept. 2003, pp. 123–130.

[6] John D. Fernandez and Andres E. Fernandez. "SCADA systems: vulnerabilities and remediation." In: *J. Comput. Small Coll.* 20 (4 2005), pp. 160–168. ISSN: 1937-4771. URL: http://portal.acm.org/citation.cfm?id=1047846.1047872.

[7] K. Ghoshal. "Distribution automation: SCADA integration is key." In: *Computer Applications in Power, IEEE* 10.1 (Jan. 1997), pp. 31 –35. ISSN: 0895-0156. DOI: 10.1109/67.560831.

[8] Warren Harrop and Grenville J. Armitage. "Modifying first person shooter games to perform real time network monitoring and control tasks." In: *NETGAMES*. Ed. by Adrian David Cheok and Yutaka Ishibashi. ACM, May 24, 2007, p. 10. ISBN: 1-59593-589-4. URL: http://dblp.uni-trier.de/db/conf/netgames/netgames2006.html#HarropA06.

[9] Petr Holub et al. "Grid Infrastructure Monitoring as Reliable Information Service." In: *European Across Grids Conference*. Ed. by Marios D. Dikaiakos. Vol. 3165. Lecture Notes in Computer Science. Springer, Jan. 5, 2005, pp. 220–229. ISBN: 3-540-22888-8. URL: http://dblp.uni-trier.de/db/conf/eagc/eagc2004.html#HolubKMR04.

[10] Vinay M. Igure, Sean A. Laughter, and Ronald D. Williams. "Security issues in SCADA networks." In: *Computers Security* 25.7 (2006), pp. 498 –506. ISSN: 0167-4048. DOI: DOI:10.1016/j.cose.2006.03.001. URL: http://www.sciencedirect.com/science/article/B6V8G-4JXRWXY-1/2/b9d08f2cdd9717ffab6c60e9b7d658f1.

[11] Emir Imamagic and Dobrisa Dobrenic. "Grid infrastructure monitoring system based on Nagios." In: *Proceedings of the 2007 workshop on Grid monitoring*. GMW '07. Monterey, California, USA: ACM, 2007, pp. 23–28. ISBN: 978-1-59593-716-2.

[12] I. Jawhar, N. Mohamed, and K. Shuaib. "A framework for pipeline infrastructure monitoring using wireless sensor networks." In: *Wireless Telecommunications Symposium, 2007.*

WTS 2007. Apr. 2007, pp. 1–7. DOI: 10.1109/WTS.2007.4563333.

[13] Fotis Liarokapis and Sara de Freitas. "A Case Study of Augmented Reality Serious Games." In: *Looking Toward the Future of Technology-Enhanced Education: Ubiquitous Learning and the Digital Native*. IGI Global, 2010, pp. 178–191.

[14] Susan N. Mart. "Let the People Know the Facts: Can Government Information Removed from the Internet Be Reclaimed?" English. In: *Law Library Journal, Vol. 98, p. 7, 2006* (). DOI: 10.2139/ssrn.741406.

[15] *Report Card for America's Infrastructure*. American Society of Civil Engineers. Apr. 22, 2010. URL: http://www.infrastructurereportcard.org/.

[16] Jacob N. Shapiro and David A. Siegel. "Is this Paper Dangerous? Balancing Secrecy and Openness in Counterterrorism." In: *Security Studies* 19.1 (2010), pp. 66 –98.

[17] B. F. Spencer, Manuel E. Ruiz-Sandoval, and Narito Kurata. "Smart sensing technology: opportunities and challenges." In: *Structural Control and Health Monitoring* 11.4 (2004), pp. 349–368. ISSN: 1545-2263. DOI: 10.1002/stc.48. URL: http://dx.doi.org/10.1002/stc.48.

[18] *Supervisory Control and Data Acquisition (SCADA) Systems in Liquid Pipelines*. Safety Study NTSB/SS 05 02. National Transportation Safety Board, Nov. 2005.

[19] Kristen E. Uhl. "The Freedom of Information Act Post-9/11: Balancing the Public's Right to Know, Critical Infrastructure Protection, and Homeland Security." In: *American University Law Review* 53.1 (Oct. 2003), pp. 261–311.