White Paper

sentilla™

A New Vision for Pervasive Computing

Moving Beyond Sense and Send

Contents

A New Vision for Pervasive Computing
Moving Beyond Sense and Send

by Joe Polastre, Chief Technology Officer & Co-founder

Computers have become a pivotal component of our daily lives. Whether we use them to gather information on the web, play games, or run our businesses, computers are noticably more widespread, smaller, and mobile. What we often overlook are the billions of computers around us that we never see. Over 10 billion microcontrollers ship each year, and they exist in unexpected objects. Even my skis have a small microcontroller in them. But we've only scratched the surface of the intelligence that all these computers integrated into the world can provide — like adjusting the tension in my boots when ready to land, knowing far I traveled on my skis, calculating long I'm in the air after each ski jump, or alerting me when my skis need to be sharpened. These small computers, with the ability to inter-network with the rest of the world and our existing IT environments, are revolutionizing how businesses operate and how people live, work, and play.

Pervasive computers can solve some very big problems. Attached to shipping containers, pervasive computers directly communicate with each other to orient themselves according to their contents, manifests, and destinations. Attached to power transmission structures, pervasive computers ensure that our critical infrastructure is always available and remains safe and secure. Pervasive computers are useful in a wide range of markets, like agriculture, health care, energy conservation, and manufacturing. But don't limit pervasive computers to just these markets; toys can communicate amongst themselves to create new entertainment scenarios like distributed tamagotchis. And pervasive computers can even be used to track social interactions in the physical world—the next frontier for social networking.

Pervasive Computing Applications
Speeding the flow of goods through the supply chain

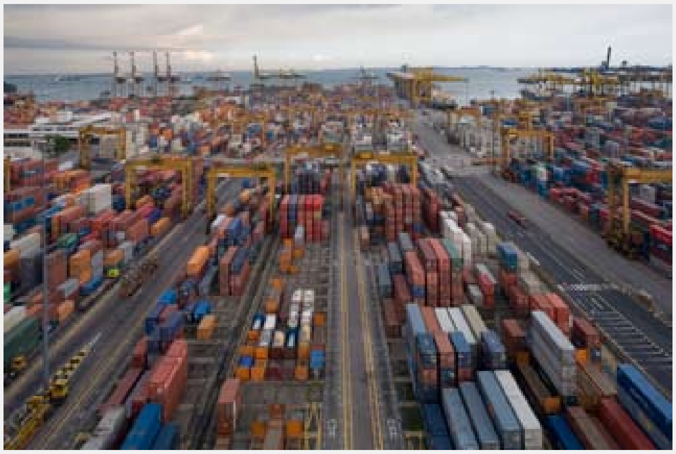

Figure 1
The logistics relies on paper manifests and manual labor to verify the assembly and contents of containers. Pervasive computers attached to each container can automate this process -- saving a tremendous amount of time and improving the flow of goods in supply chains.

In order to achieve the vision of pervasive computing, a new approach to the software, infrastructure, and applications is necessary. Conventional approaches attach sensors to a wireless transceiver and then process the millions of real-world data points at a central server, often located long distances away from the phenomenon of interest. This approach is central to the "wireless sensor networks" (or WSNs) industry. With WSNs, applications are limited to a single function called "sense and send." In this article, we describe the inherent problems with the sense and send paradigm, its inability to solve pervasive computing applications, and how businesses can move beyond sense and send.

Sense and Send Applied

Among the most obvious sense and send applications are those that deal directly with wire replacement. Foremost among these is monitoring the condition of the environment — its moisture, humidity, temperature, and chemical composition — for agriculture, seismology, pollution monitoring, and other fields. Nodes gather data locally and forward them to a server for analysis. In essence, monitoring environmental conditions entails robust data logging with a simplified, wireless information retrieval system.

High-latency, low-frequency condition monitoring is sense and send at its most basic. Other monitoring, such as machine or structural monitoring, can place stringent demands on the system. For instance, condition-based maintenance (CBM) is a class of pervasive applications where engines and similar heavy machinery or structures are monitored to create vibration signatures. When these signatures fall out of specification, technicians can be dispatched to address potential issues before they become problems. When large numbers of nodes are collecting large amounts of data, the wireless connections become saturated; moreover, these applications use significant amounts of power since they are constantly performing a "sensing and sending" function with little processing or data aggregation taking place at the local node level.

Pervasive applications rarely end once data is captured; further action is often still required. An application may need to swivel a camera, close a solenoid, or sound an alarm. Yet in sense and send, the node itself can't make the decision to actuate; it can only forward the relevant data to a server that must process the information to make a decision. When servers are numerous network hops away, latency prevents actuation in real time.

The situation becomes even more precarious when applications require the nodes to be mobile. Nodes may be attached to cargo containers, pallets, crates, or people moving into and out of an environment, e.g., firefighters in a building or containers moving in and out of shipyards. These mobile nodes are not connected to the same server at all times — in fact they may not be within reach of a server at all. However, they are still required to perform their functions, such as halting a train that is dangerously close to derailing. Pervasive applications such as this one require logic on the node.

A significant set of application classes have requirements that go beyond sense and send. Some examples include:

- Responding to events or state locally to allow pervasive applications to execute autonomously, thereby taking context-aware actions determined by business rules. Furthermore, applications that require low-latency actions can be executed by the nodes in real time.
- Collaboration between nodes enables new functionality, such as voting, weighting, and summarization. Nodes can determine their neighbors and work with them to determine the orientation of a container, track a firefighter in a building, or collectively analyze heat, motion, and sound signatures to classify that a human is present.
- Local filtering of data, data analysis and classification, and removal of false positives to ensure that pervasive applications only use communication and power resources as required, reducing costly battery replacement maintenance operations and removing reliance on potential single points of failure (a network gateway or backend server).

A New Approach

To move beyond sense and send, a different paradigm that provides a new level of flexibility and programmability is required. Since data processing, analysis, and inter-node collaboration are primary functions required to address pervasive application needs, the focus must be on the computer itself rather than the sensor or network interface. Instead of making RFIDs smarter, approach the problem from the other side by making computers smaller. To illustrate this approach, let's look at the components of a pervasive computer and compare pervasive computers to WSNs and RFIDs.

A pervasive computer consists of the typical components you'd find in the computer sitting on your desk—a central processing unit (CPU), input and output devices, and a communications interface. The primary difference is that pervasive computers are small (about the size of a quarter), run on batteries, communicate wirelessly, and have very different input and output devices (such as sensors and actuators instead of keyboards and mice). Pervasive computers monitor and control the physical world, through the use of sensors and actuators.

If one were to remove the CPU from a pervasive computer, the system is no longer programmable. It cannot process, filter, or analyze data. It is no longer capable of making decisions, coordinating with other pervasive computers, or autonomously controlling the environment. These devices, consisting of peripherals (sensors) connected to a wireless link or network, are called wireless sensor networks. Similarly, if the networking component is removed from a WSN system, the system can only communicate with a master within range in a point-to-point manner. These systems are called Active RFID. And finally, if the power source is removed, the resulting system is Passive RFID.

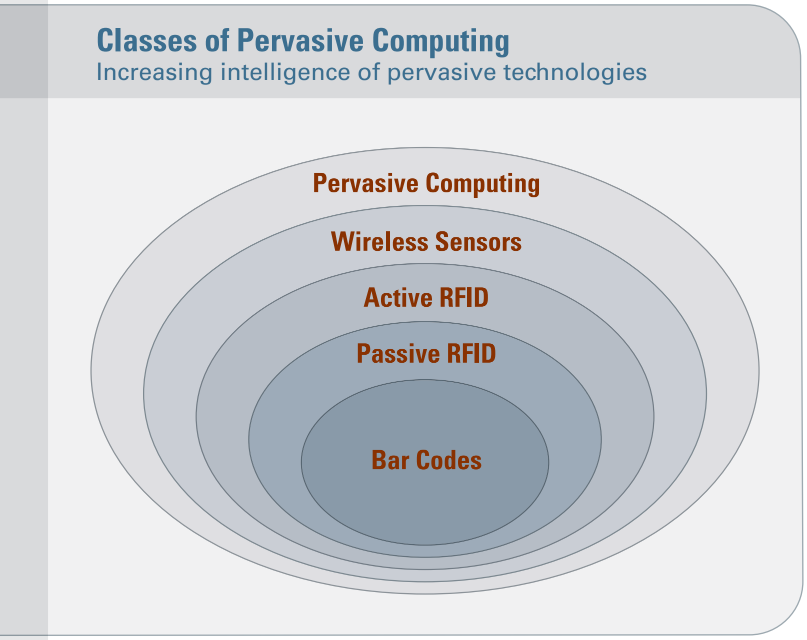

Figure 2
A visual representation of the capabilities of various real-world technologies. Pervasive computers solve a wide range of large real world problems not addressed by previous approaches.

Challenges

To satisfy the demands of these new application classes, the developer can either focus on low-level details or sacrifice application flexibility. In the first case, building a system from the ground up, the developer has to worry about radio frequencies, frequency hopping, latency, power tradeoffs, interference, distance, network routing strategies, security, and countless other physical and link-layer characteristics. If the developer manages to get a handle on these issues, he or she is then faced with programming nodes in a low-level language such as embedded C.

All software applications require four main components — the ability to develop an application, deploy the application in a product, integrate the product with existing infrastructure, and manage the product remotely. Although C is efficient, it lacks abstraction support and is poorly suited for applications distributed among millions of computers. Moreover, lifecycle support does not exist for programs running on the small computers themselves. Tools such as debuggers and simulators are missing or incomplete, as are other post-development tools used for deploying and managing the networks. Updating software on WSN nodes is nearly impossible in most cases and often requires an update of the entire firmware on the node to make a single parameter change.

Since this task is so daunting, wireless sensor network vendors have resorted to the simplified sense and send approach, placing conditions on what can and cannot be accomplished. Moreover, customization is limited to basic configuration changes. Parameters can be tweaked, but not exceeded. This means that innovation is limited to the vendor's vision, as opposed to the developer's. As a developer myself, I am most frustrated when told that what I want to accomplish is not permitted. Innovators in organizations large and small are prevented from turning their visions into reality by both approaches — low-level full-system design and sense and send products. As a consequence, the overall wireless sensor network industry remains stuck in a "sense and send" world.

Solution

We must move beyond the "dumb" devices of wireless sensor networks and RFID to value-rich applications with pervasive computers. By treating devices as computers, developers can build their ideas into solutions. They can leverage tools, paradigms, and interfaces made familiar from their experience with laptops and servers. Low-level details of embedded computers are abstracted by standard APIs while retaining flexibility.

To ease and expedite development, solutions must rely on familiar, widely-used, and accessible tools used by developers coupled with standard computing paradigms so that developers and IT managers are not required to learn new languages. Responding to customer feedback, Sentilla's pervasive computing software platform (Figure 3) lets users interface with pervasive computers in the same way that they use other computers in their enterprise

The software platform enables the full application life cycle of development, deployment, integration, and management. Sentilla developers use Java technology to build applications and embed business rules into pervasive computers using familiar Java software tools, APIs, and interfaces. For the first time, Java technology on low-cost, low-power hardware (such as wireless microcontrollers used in sense and send systems) turns billions of microcontrollers into fully functional computers without requiring new, or more expensive hardware.

Beyond development, the pervasive computers — with their data acquisition (DA), processing, and autonomous response — are miniature data centers providing services to enterprises through service-oriented architecture (SOA) interfaces. For IT managers, Sentilla's approach doesn't require them to learn new management tools and dashboards; the pervasive computers are remotely managed — displayed and controlled as computers — by tools such as OpenView and Tivoli. Using these familiar interfaces and computer tools, such as Java and SOA, pervasive applications can be developed in days, rather than months.

Sentilla Software Suite

A complete life-cycle solution for pervasive computing

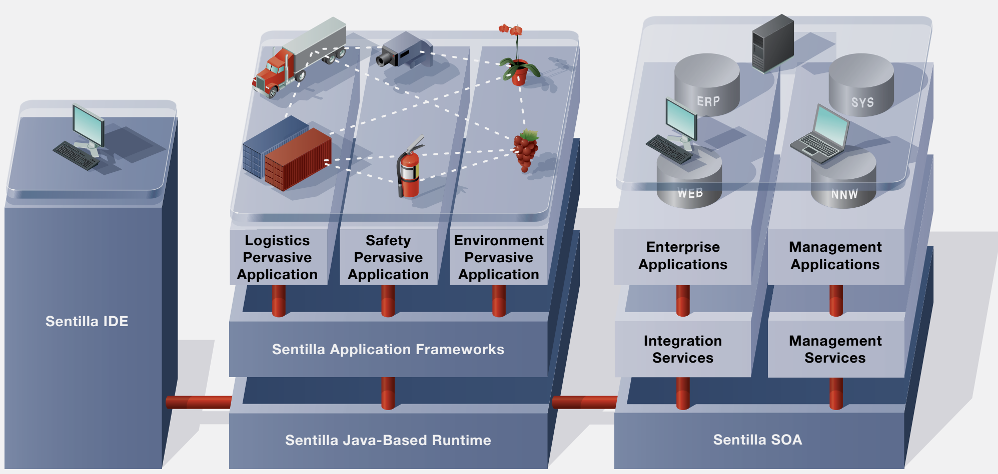

Figure 3

The Sentilla software suite addresses the full software life cycle of an application, from development to deployment to integration to management.

The concept of wireless sensor networking originally burst onto the scene with an article in 1999 about "smart dust" — computers that can be sprayed on the wall, deployed anywhere throughout the environment, and collaborate to solve big problems. The industry has strayed from the concept of smart dust, reduced to sense and send. The potential for pervasive computers remains; all that is required is for the six million Java developers to be given familiar tools to build their vision. Sentilla's software accomplishes our new vision of pervasive computing, making innovative, killer applications possible — what idea do you want to build?

About the Author

Joe Polastre is Co-founder and Chief Technology Officer of Sentilla Corporation. Dr. Polastre is responsible for defining and implementing the global technology strategy and overseeing the product roadmap. His experience with pervasive computing stems from his deep background with wireless sensor networks, having deployed the first-ever autonomous network on Great Duck Island, Maine, in 2002. In addition to his practical experience, Dr. Polastre sits on numerous technical boards and commissions and is a tireless evangelist for pervasive computing. His ability to explain the technology in a straightforward and engaging manner puts him in constant demand for speaking engagements worldwide. Dr. Polastre holds Ph.D. and M.S. degrees in Computer Science from the University of California, Berkeley, and a B.S in Computer Science from Cornell University.

Sentilla Corporation
201 Marshall Street
Redwood City, CA 94063
tel. 650.241.0220
www.sentilla.com