Open Standard Development Platforms for Distributed Sensor Networks

William Merrill, Katayoun Sohrabi, Lewis Girod, Jeremy Elson, Fredric Newberg, and William Kaiser
Sensoria Corporation
200 Corporate Point Suite 100
Culver City, CA 90230, USA
williamm@sensoria.com

ABSTRACT

In the development of distributed security sensor networks a large variety of prototype systems have been implemented and tested. However these systems tend to be developer specific and require substantial overhead in demonstrating more than one application. To bridge the gap between embedded, networked systems and desktop simulation environments, systems are necessary which are easily deployable and allow extended operation of distributed sensor networks, while allowing the flexibility to quickly test and evaluate a variety of operational algorithms. To enable fast optimization by leveraging the widest development community, open standards for such a portable development system are desired. An open development system allows individual developers and small groups to focus on and optimize specific aspects of a distributed sensor network within realistic deployment constraints, prior to complete integration and deployment of a system within a specific application. By providing an embedded sensor and processing platform with integrated wired and wireless networking, a modular software suite separating access and control of individual processes, and open APIs, algorithm development and software optimization can be greatly accelerated and more robustly tested. To meet the unique needs of distributed sensor network applications, additional separation must be provided between the access to various subsystems, for example real-time embedded control versus tasks with less stringent timing requirements. An open platform that separates these requirements allows developers to accelerate testing and development of applications by focusing on individual components of the distributed sensor system, such as target tracking or low power networking. The WINS NG 2.0 developer's platform, provided by Sensoria Corporation for the DARPA/ITO Sensor Information Technology (SensIT) program, provides one example of such a system. This systems bridge the gap between dedicated desktop development environments and embedded application-specific unattended sensor systems. This system provides open access control to high data rate sensing, local multi-hop wireless and wired networking, node geolocation, the Linux operating system, additional software process separation and control, and a size and power constrained system with access to both high and low level system control. This paper describes the benefits in providing open standards to develop and compare distributed sensor applications while using the WINS NG 2.0 development system as an example of the flexibility and development speed an open system approach facilitates.

1. REQUIREMENTS OF UNATTENDED DISTRIBUTED SENSOR SYSTEMS

As the size and cost of embedded electronics systems shrinks while their complexity increases new avenues are opened up for technological applications beneficial for defense, security, and law enforcement [1,2,3]. Particularly with the advent of embedded computing platforms merging substantial sensing capability with complex processing, command, and control in a limited volume and at acceptable power levels, the application and impact of distributed sensing systems is expected to substantially increase. In order to provide the best overall performance, embedded sensor systems must be distributed to cover a wide area, operate in a peer-to-peer fashion to provide redundancy, communicate wirelessly over an autonomously assembled, self contained network, provide long lifetime on limited batteries, provide complex sensing and processing capability to ensure adaptability, and intelligently limit the information and decisions passed up to the end user. In addition to facilitate development, testing, and the interaction of disparate applications and algorithms each system or algorithm component being tested should provide common attributes and interfaces. To determine what those requirements for a common unattended distributed sensor system are, a general summary of the restrictions of such systems through the fundamental physical bounds within which the system must operate, are discussed below.

1.1. Energy Issues

The majority of unattended ground sensor systems must operate from batteries or limited energy renewal means like solar cells for extended periods to be effective. Thus the energy constraints, in combination with the specific application, generally provide the fundamental limitation on the system performance. Figure 1 provides a relative ranking based on peak power of orders of magnitude capabilities for embedded systems, given a constant energy budget. Provided with the processing capability required to support an application this figure gives an idea of the system size and performance realizable in terms of the energy supply needed to sustain the application. This figure provides an indication of processing energy, however generally the largest single power draw in a wireless distributed sensor system is the radio, such that substantial processing may often be used before the processor utilizes equivalent levels of energy as the communication systems.

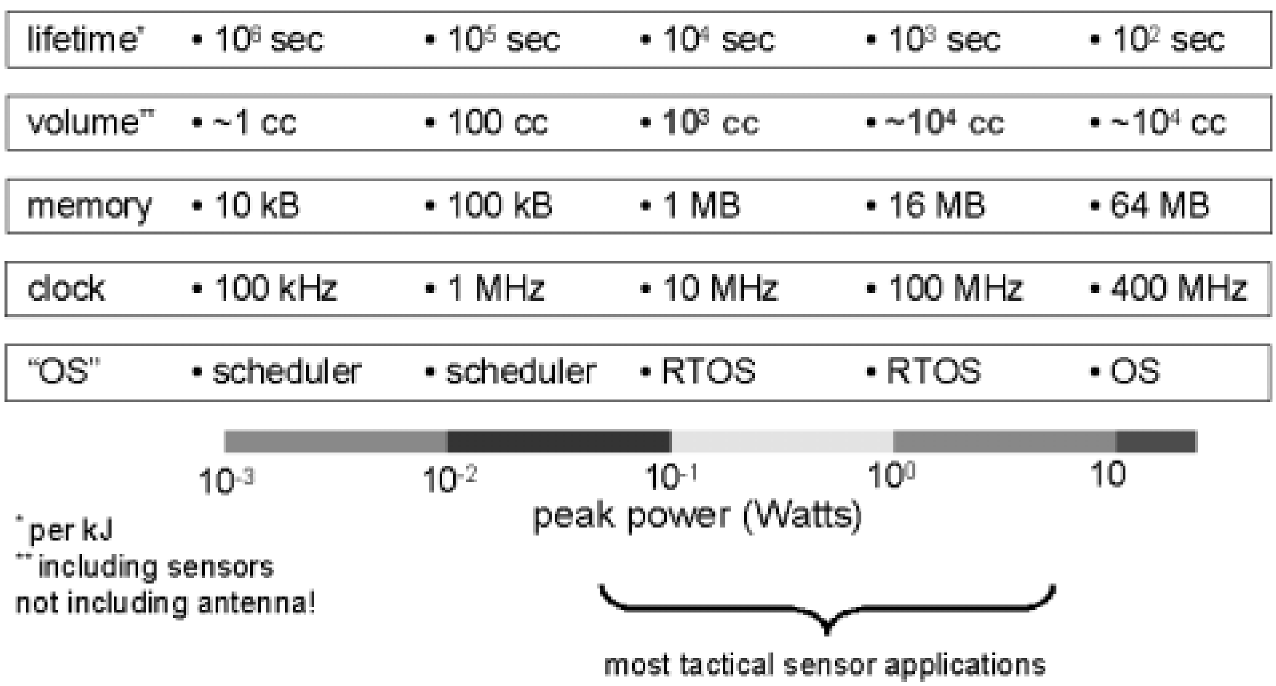


Figure 1: Ranking of orders of magnitude for distributed sensor platforms according to their peak power usage.

An illustration of the energy cost of information is shown in figure 2. The energy cost associated with usage of different platform subsystems, such as sensing systems, signal processors, radios, is illustrated with analog sensing subsystems generally having the lowest energy cost. Operating the processing subsystem costs more in terms of energy than utilizing the analog sensor interface, but is still less costly in terms of energy consumption than, wireless communication. Due to this, application specific processing can trade off energy efficiency for processing flexibility (figure 2) and intelligently processing information at the sensor node can result in significant overall system energy savings, translating into increased system life or increased operational utility. This energy requirement guideline is supportive of deploying increased processing capability closer to the sensor, relieving the data transmission burden on the wireless network. To support this type of configuration with the most flexibility for a wide variety of applications, open development standards for distributed algorithms operating near the sensors are needed. Open systems allow standardization of the interface layers in a distributed sensor system, where each layer may represent a distinct phase in the life of data, as illustrated in figure 2. Thus optimization of each layer can be independent of the other layers, providing the most flexibility in the system, for development and comparison of various systems.

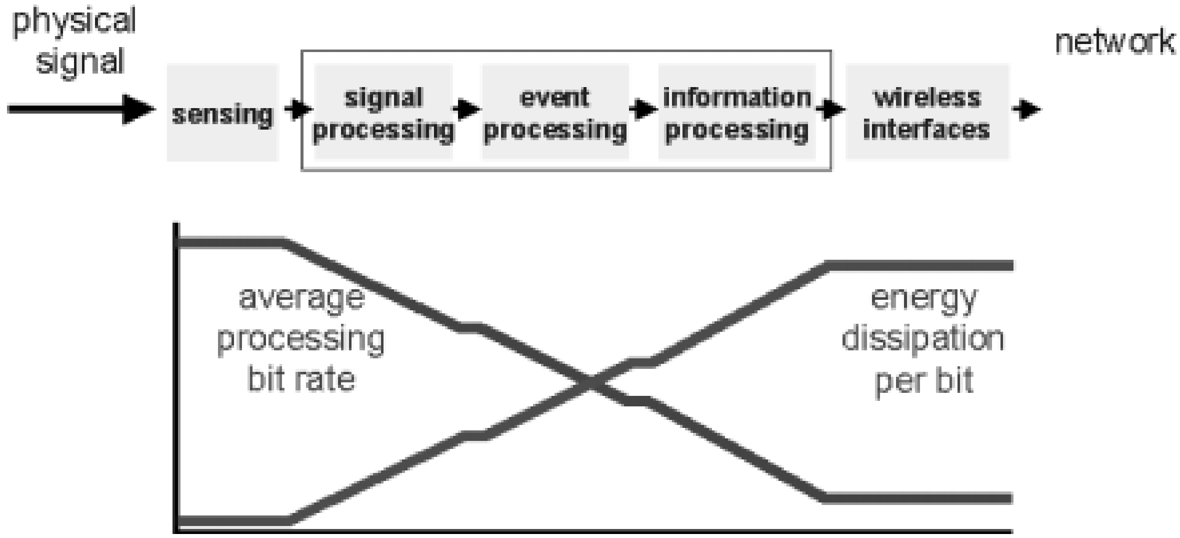


Figure 2: Qualitative illustration of the energy cost of incoming information for a distributed sensor system.

1.2. Constraints of the Sensing Modalities

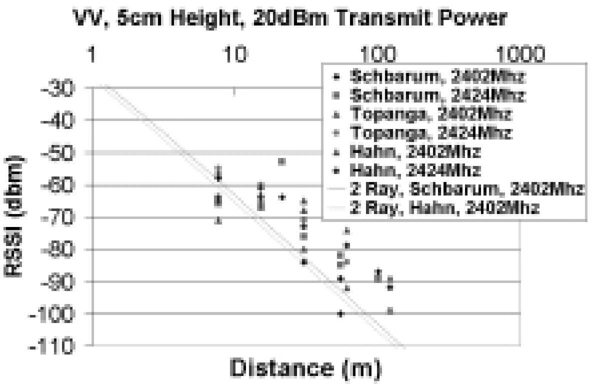
The utility of a sensor system is constrained by its sensing, while the effectiveness of a sensor is dependent on the intended target. For example to detect personnel, most sensor systems are only effective at short ranges (less than a few hundred meters). As a result, in combination with the communication range sensing range for the targets envisioned defines the deployment densities of a distributed sensor system.

For example considering acoustic sensing, in [4] a thorough analysis of the range limitations for a complex acoustic sensing system are described. The analysis in [4] of the ADAS sensor system targeted at vehicles detections gives a number of conclusions, key among them being: the substantial environmental dependence that limits viable target range for ground vehicle detection to a few kilometers and the difficulty of resolving multiple vehicles along the same baseline. Some of these problems can be resolved as suggesting in [4] via the use of other sensing modalities, such as a combination of acoustics and imaging. However the use of multiple sensing modalities does not increase the range of the system beyond that of the longest-range modality used. Thus in most cases systems operating with sensor spacing of at most a few kilometers (for vehicle sensing) and a few hundred meters (for personnel) are expected. Thus the sensing range in conjunction with the expected communication range determines the sensor deployment density, at least for applications utilizing multiple sensors for continuous coverage, or utilizing concurrent views of the same target.

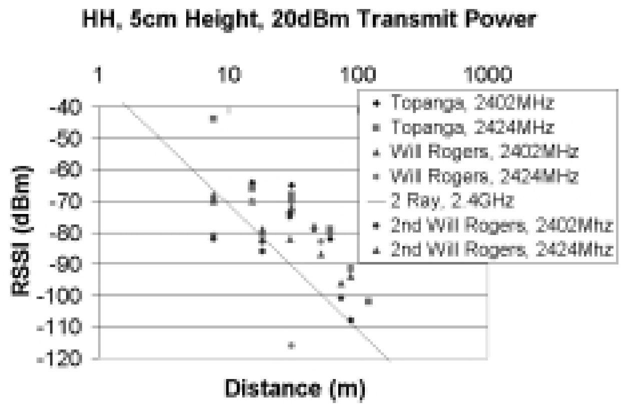
Different sensing modalities have different environmental and target-specific limitations, thus the most robust systems should combine multiple sensing modalities for the best target resolution and to reduce the occurrence of false alarms. An open interface to a variety of sensors provides the most flexibility in supporting the fast development of multiple algorithms leveraging multiple sensor modalities, while providing the flexibility and local processing capability to support low energy sensing front-ends triggering the use of higher energy classification signal processing, and/or higher energy alternate sensing modalities.

1.3. Limits Imposed by Wireless Communication

Distributed sensor systems can be connected both with wires and wirelessly, however the logistics burden for deploying wireline systems is large enough that most current distributed sensing systems in development utilize wireless communication. However wireless peer-to-peer outdoor systems introduce other complexities. Particularly for ground based sensor systems the propagation environment can be very harsh. At the UHF and higher frequencies used to support most networked sensors, the dominant propagation effects can be modeled well locally with ray tracing analogies. Near a finite conductivity ground, a fourth power propagation fall off with distance is observed for both horizontal and vertical polarization within these analogies [5]. This fall off is appropriate as a lower bound even for complex environments at short ranges, as demonstrated in the two plots of figure 3, which show the path loss in the 2.4GHz bands for antenna heights of 5cm in three relatively open rolling fields (3a), and in three more cluttered hiking trails (3b) [6]. While the actual path loss varies substantially from environment to environment, the worst-case likely path loss of the flat open area (i.e. a fourth power fall off within a few hundred meters outdoors) provides an operating limit for most applications. These communication restrictions, in the context of limited available power emphasize the need for short-range links to save energy.



(a)



(b)

Figure 3: (a) Measured path loss in three relatively open areas for vertical-to-vertical polarized signals. (b) Measured path loss along three hiking trails for vertical-to-vertical polarized signals.

Figure 4 illustrates the idea of the power savings achievable by using multiple short individual wireless links rather than one long link. In the figure the bars represent the energy required to communicate energy over a distance R along the surface of the ground at each node. Since this energy increases as R^{-4} with range (R), by reducing the distance by a factor of N-1 for each hop (with N total nodes to relay the signal), the total energy required for the radio to support the link over R scales, with worst-case propagation, with $N/(N-1)^4$. This scaling assumes the power of each radio is optimized for that link distance.

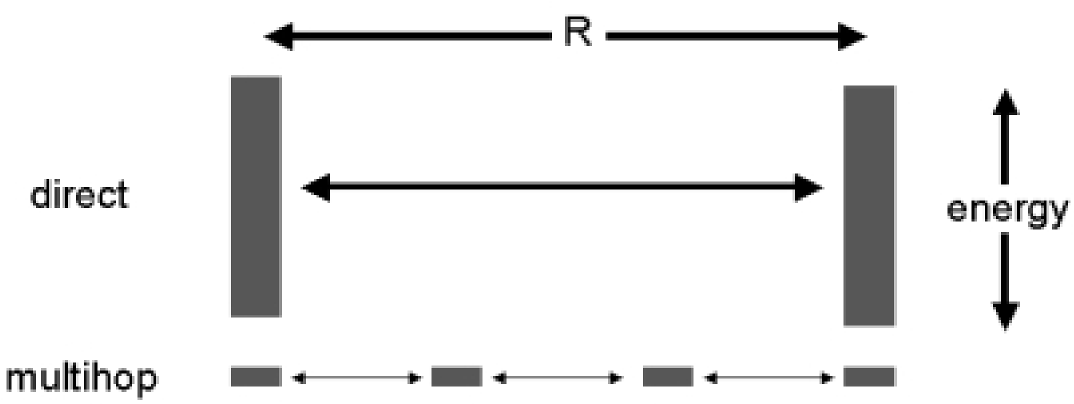


Figure 4: Conceptual comparison of the energy required to send a constant sized message over a distance R, using a single optimized link, or multiple hops along the ground.

The actual energy savings achieved via a multihop system must be balanced against the power control on the radio (to use only the energy required to transmit the message at each link the distance R/(N-1)), the linearity of the radio power draw as a function of transmit power variation, the energy cost of wireless transmission in comparison with other system components, and the latency required to multi-hop messages at each point versus the latency supported by the application. However, as is illustrated in figure 3, for a 100mW transmit power system, near the ground, only short hops are possible with significant transmit power increases necessary to support longer ranges, unless multiple hops are used to extend the total communication range.

Successful distributed sensor architecture therefore should provide means to support multihop networking. This requires substantial built-in flexibility at all layers of the system design, including the networking stack. This flexibility results from the provisioning of open interfaces at the radio physical layer, at the network assembly and routing layers, as well as at the upper level networking protocols that support user applications.

1.4. Widely Disparate Requirements

A robust, distributed, widely applicable sensor system must meet a wide range of requirement including communication and operation security, network and application scalability, ease of manufacturing, a variety of deployment options, package size and survivability restrictions, requirements for end-to-end integration with other systems, extended operation lifetime, and operation in substantially varying environments. As a result substantial complexity must be built into most applications. For example the utility of a closed, single threaded system is generally limited to the small set of application scenarios for which it was designed. It is difficult to upgrade this system in the field, to design it to intelligently adapt to changing conditions in the field, or for the system to make the most effective use of its available energy in a dynamic application. By contrast, a platform based on a more capable processor with an open operating environment, is able to provide much higher levels of flexibility through the use open and standard physical and software interfaces, to support a wider suite of software and hardware modules, and hence a wider set of applications.

The aim of this paper is to describe the benefit of an open development platform for unattended sensor applications in meeting the disparate requirements of those applications as opposed to the restrictions presented by a closed architecture. As illustrated in figure 5, unattended sensor applications are seen as requiring a blend of expertise from general purpose computing, embedded computing, and networking, to create an optimal platform supporting the signal processing and algorithmic development necessary for diverse applications.

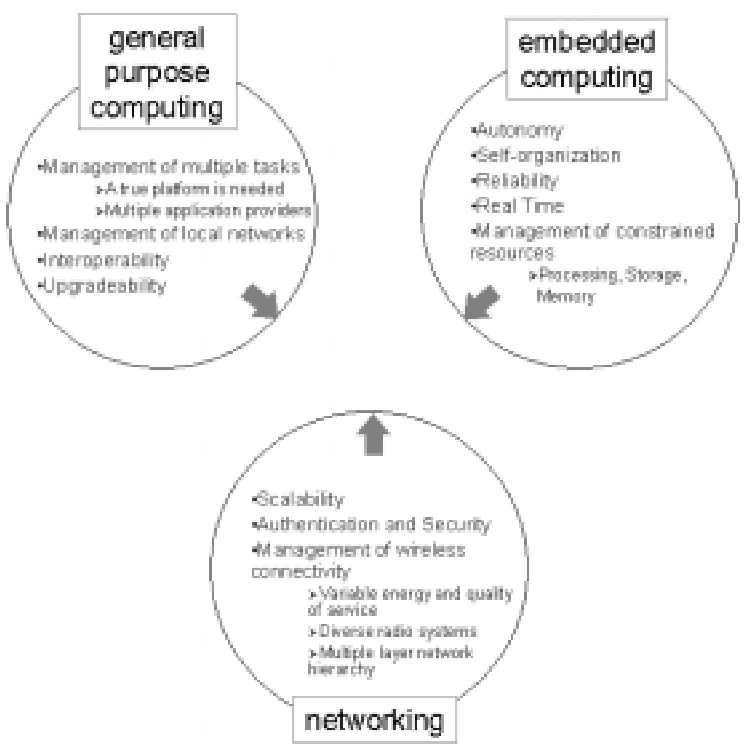


Figure 5: Various system requirements that can influence the capability of a wireless integrated networked sensor system.

2.OPEN DEVELOPMENT SYSTEM ARCHITECTURE

As discussed above, widely disparate requirements exist for unattended, networked sensor systems. An open development system provides the most flexibility in meeting these requirements. For example supporting a wide development community and providing sufficient flexibility for system optimization requires a system designed for general distributed sensor applications based on the constraints of that application space. Such a system should open up each layer to allow maximum optimization and design flexibility for further development. For maximum flexibility a system should provide open interfaces through: a modular hardware reference design that can be modified to meet size and packaging requirements, an open software architecture that supports interfacing to and replacing modular software components, and a modular subsystem approach allowing easy interchange of sensors, radios, or other subsystems to meet specific application requirements with a minimum of modification. Sensoria Corporation has developed various hardware reference designs based on input from the defense community that reflect these design practices. One of these, the WINS NG 2.0 system, is described in the following sections. In addition we have developed an open software architecture for these hierarchical hardware systems. Some features of this open software architecture are described below.

One method of providing an open software architecture is to ensure that code is modular with open interfaces at each layer, allowing specific applications to use the layers necessary for most efficient operation, while protecting the overall system from the failure or malicious operation of each software component. An example of the software layers provided by Sensoria Corporation in their current wireless integrated sensors systems is shown in figure 6. This Sensoria open standard software architecture has been developed to enable rapid implementation of the many distributed applications and processes for networked sensor systems. Sensoria's latest open development platforms exploits the Framework for User Space Device Drivers (FUSD) that enables POSIX-standard device file interfaces to both hardware and software modules, for example to applications and devices within any of the layers of figure 6.

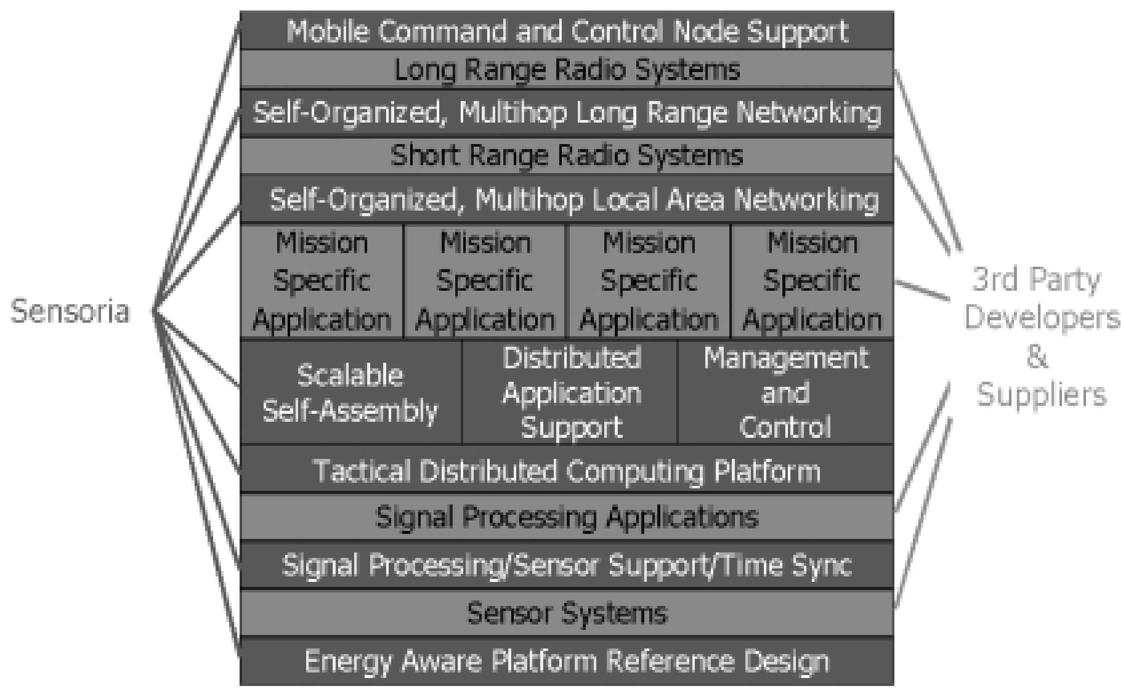


Figure 6: Illustration of the software layers built into Sensoria Corporation's modular software approach. Each layer presented provides an open API allowing individual developers to build their own applications quickly, compare performance of similar applications, or rewrite components as desired.

Conventional platform architectures require that the device drivers be hosted in kernel-space, where drivers are not provided with mutual memory protection and have limited access to system resources including storage and other drivers and services that can typically only be used from user-space. In contrast, the FUSD allows drivers or services to be implemented in user-space, exploiting the full advantages of user space services including memory protection, process management, and properly controlled access to system resources, while still remaining accessible via the open POSIX-standard device file interface. This architecture now extends the operating system's basic system call interface to include access to management systems, and logical services, as well as traditional access to physical devices. FUSD drivers and services may reference other FUSD drivers. Such composition promotes rapid development of complex distributed drivers and services, robust operation, and fine-grained access control of both logical services and physical hardware.

For the hardware subsystem device drivers, FUSD allows the driver interface to remain independent of the specific radio, sensor, or other hardware interfaces so that applications access a common open API specification. The flexibility enabled by the FUSD architecture allows the driver to support many complex features required for energy and bandwidth efficient operations, in the same consistent manner. FUSD allows drivers to be written as user space processes, and provides a consistent open interface for multiple hardware drivers, and software modules. While the API interface remains constant, the API implementation, hosted by the FUSD system, is specifically adapted to each radio, sensor, or subsystem choice.

In addition to the open interface control between processes and components provided by FUSD, and to the features of the open Linux OS, Sensoria Corporation has also developed the Process Initialization technology (PInit) to provide additional robustness and resilience for support of many simultaneous distributed computing tasks, all operating on a low-power, resource constrained platform. PInit provides a mechanism to start, stop, and monitor every process running on the platform, enables watchdog functionality, and provides a mechanism for response to individual module failure or unexpected operation. PInit, in addition to providing local autonomous and remote management, also provides an application messaging and control bus for all platform applications, to support a system operating a complex variety of applications, such as illustrated in figure 7. For example considering figure 7, PInit, provides a mechanism to monitor and control each of the general purpose applications, and each software module on the system.

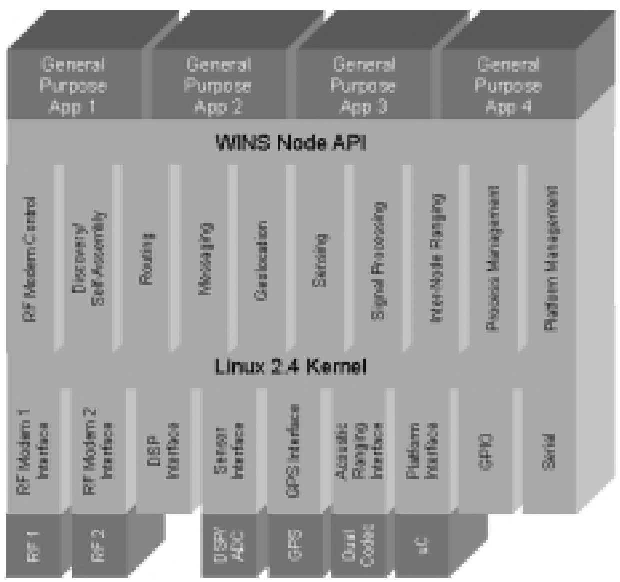


Figure 7: Illustration of the open software interfaces developed on Sensoria Corporations Wireless Integrated Network Sensor platforms.

An additional benefit of the open flexible architecture is the capability of the distributed sensor platform to be easily integrated into the development process. While cross-compilation and developer tools enable initial development of embedded system algorithms to occur on a dedicated desktop system, the open interface's to each system component provide substantial flexibility for debugging and logging performance of applications once they are ported on to the embedded system. By enabling easy developer access to the embedded system directly (with telnet, ftp, and other common services, for example over a detachable Ethernet module, or development serial port) the development and test procedure can be significantly streamlined. For example in the systems developed it is possible to log into an embedded system, monitor application progress, and issue commands while operating the application under development. This avenue is of substantial use for development when trying to understand system failures or unexpected behavior, particularly when compared with the opaque nature of a closed system.

3.WINS NG 2.0: AN EXAMPLE OF AN OPEN SYSTEM DEVELOPMENT PLATFORM

One example of an open, distributed sensor development system is Sensoria Corporation's WINS NG 2.0 (Wireless Integrated Network Sensor – Next Generation) platform developed for DARPA/ITO's (recently moved to DARPA/IXO) SensIT (Sensor Information Technology) program. This platform represents a trade-off between the power constrained, embedded requirements of a fieldable sensor system, the usability of a desktop development environment, and the flexibility required to support various networking options consistent with figure 5. The WINS NG 2.0 nodes provide a baseline sensor node operation, with API interfaces to develop software applications for areas including: collaborative signal processing, intelligent networking, data storage, data dissemination, and sensor technology application demonstration. Within SensIT more than ten separate groups develop with this platform.

Figure 8: WINS NG 2.0 development platform.

Two pictures of the WINS NG 2.0 development system are shown in figures 8 and 9. The picture in figure 9 was taken at the third SensIT situational experiment (SITEX02) where seventy-five WINS NG 2.0 nodes were deployed at the Marine Core Air Ground Combat Center in Twenty-nine Palms, CA in the configuration shown in figure 11a. The WINS NG 2.0 nodes provide a system on which SensIT users can build and test their distributed sensor algorithms. The nodes were designed with feedback from the SensIT community, to provide a developer friendly system (based on the Linux 2.4 kernel) with substantial sensing and networking capability. A summary of the hardware specifications is presented in table 1. As illustrated in the architecture diagram of figure 10, the WINS NG 2.0 platform is designed for modular operation, so that each block of the system shown in figure 10 can be operated independently in order to maximize utility and energy efficiency. In addition, flexibility is built into the nodes to support different sensing options with four channels of analog inputs, multiple digital input lines, and bus interface access that allow additional sub-systems such as imaging modules (figure 10), or 802.11b PCMCIA cards to be used with the platform.

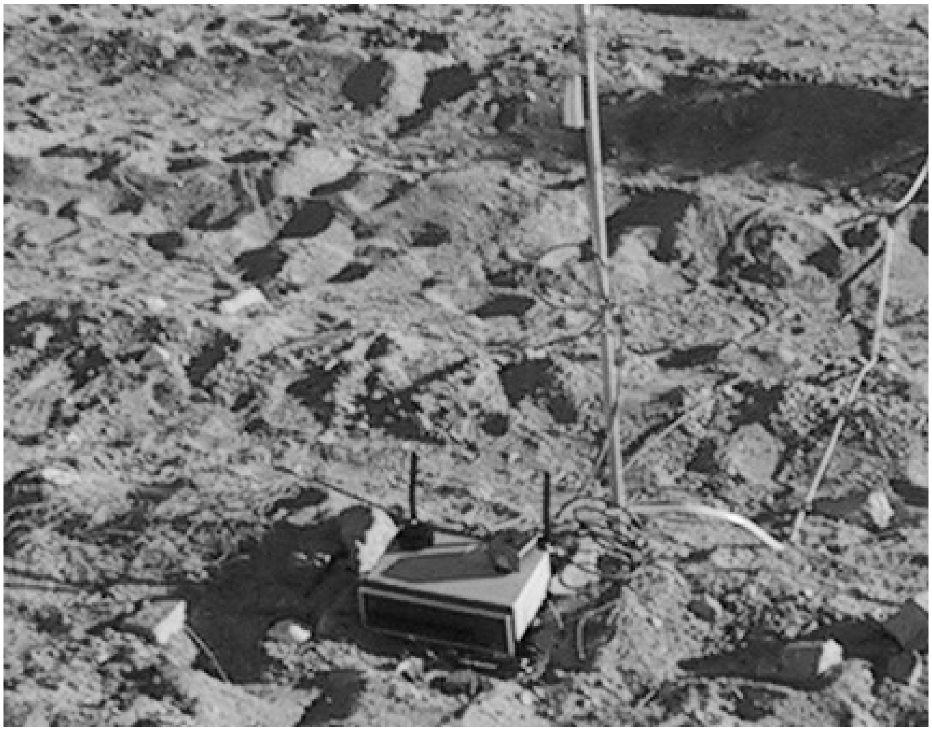


Figure 9: WINS NG 2.0 node deployed at the Marine Corp Air Ground Combat Center, Twenty-Nine Palms, CA, November 2001.

Platform Processor Core	Hitachi SH-4, 32-bit RISC
Platform Processor Performance	300 MIPS CPU, 1.1 GFLOP FPU
Platform Memory	16 to 64MB RAM, 16 to 32MB Flash
Analog Input Channels	4
Sampling Frequency (samples/sec)	20 KHz per channel
A/D Resolution	16-bit
RF Modem	Integrated dual channel, 2.4GHz modem
FCC Certification	FCC Part 15.247 and ETS300-328 rules, license free
GPS Receiver	L1 frequency, C/A code
GPS Protocol	NMEA v2.2
GPS Antenna Cable Length	6’
Digital I/O	15 configurable GPIO lines
Expansion Ports	Two PCMCIA Type II slots
Sensor Connectors	Polarized, 6-pin, differential input
Wired Interfaces	10 Mbps Ethernet, RS-232 Serial

Table 1: Hardware specifications of the WINS NG 2.0 development system.

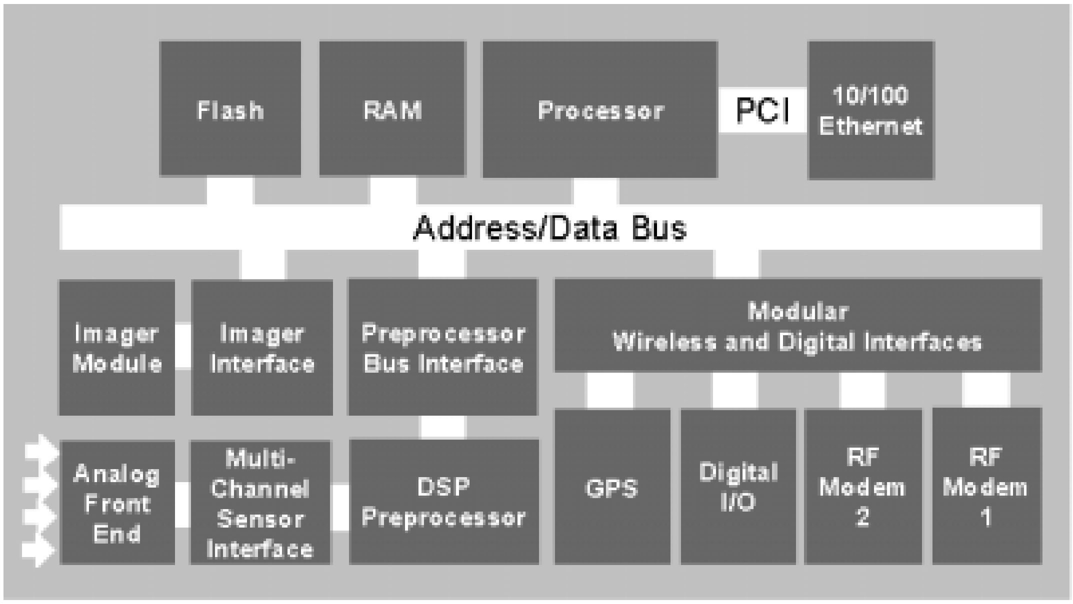


Figure 10: Hardware architecture for the WINS NG 2.0 and WINS Imager 2.0 development nodes.

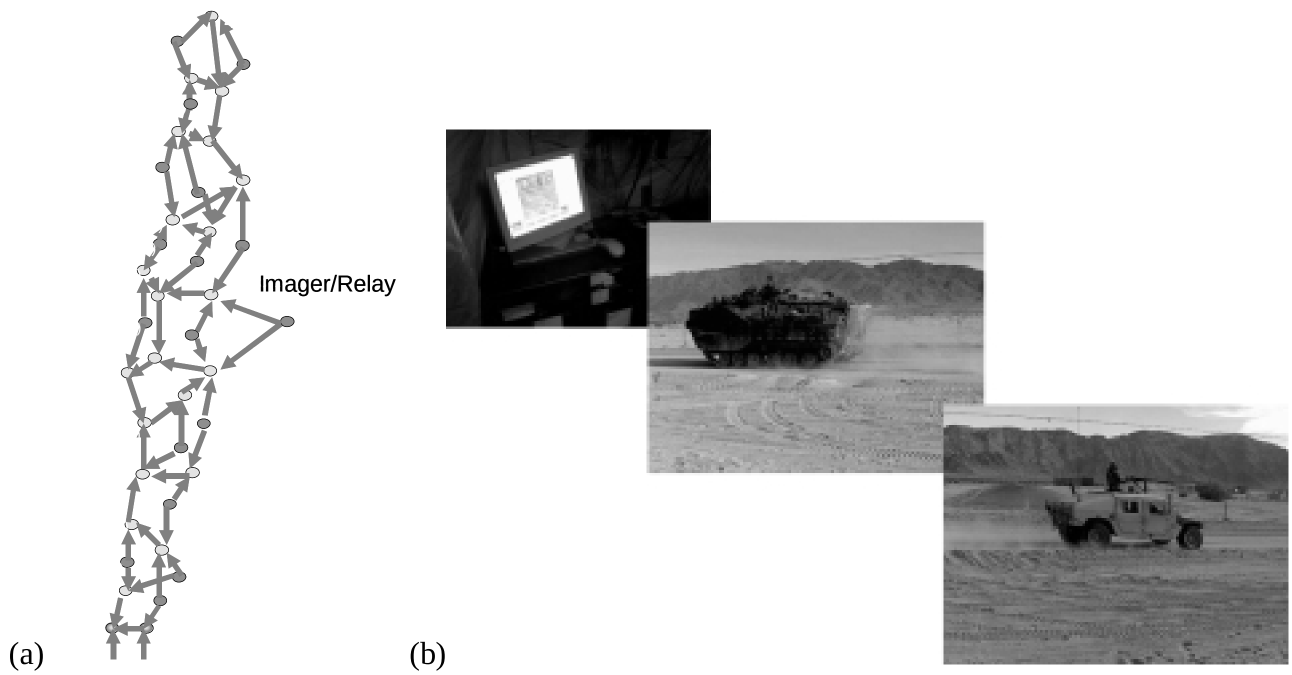


Figure 11: (a) WINS NG 2.0 deployment scenario for SITEX02, each circle represents a WINS NG 2.0 node, with the arrows illustrating radio links. The span of this deployment from top to bottom was approximately 500m. (b) Pictures taken by a modified WINS NG 2.0 system including a visible CCD. Locally triggered pictures resulting from vehicle passage, and the associated display, at Steel Knight.

The WINS NG 2.0 development system provides open access to a number of layers of the system stack shown in figure 6. Software APIs provide an interface with the DSP for sampling, interfaces to each of the radios to allow SensIT to develop routing and networking procedures, an interface to the integrated GPS module, interfaces to additional peripherals (Digital GPIO and LEDs), as well as an interface to the imaging subsystem on those units with the integrated Imager module. The WINS NG 2.0 systems as supplied by Sensoria Corporation to meet the SensIT requirements do not provide an end-to-end operation for a specific application. Rather they are dependent on software provided by the SensIT community to develop complete systems for various applications.

An example of the results from the use of the WINS NG 2.0 system, specifically the version with the imager module, is provided in figure 11b. The two vehicle pictures in this figure were taken with a WINS NG 2.0 system with an integrated imager. Based on local detections of the vehicles with passive infrared sensors, the images were taken and relayed back to a remote server over an integrated network and displayed as shown in the top picture of figure 11b. This application was developed in three weeks, with a resulting level of robustness that allows it to operate continuously for over one week during the Steel Knight Exercise at the Marine Corp Air Ground Combat Center in Twenty-Nine Palms, CA during December of 2001.

4.CONCLUSIONS

The complexity of distributed sensor applications and the wide areas for which these systems are applicable both increase the benefit of open standard development platforms. This paper has described a number of the limitations that must be considered in development of systems for distributed sensor applications, a discussion of open software and hardware architecture accounting for these considerations, and one example of an open development system used to develop and refine sensor networks for various applications. Specifically a brief descriptions of Sensoria Corporation's work in providing open development systems for the Sensor Information Technology (SensIT) DARPA/IXO program has been provided.

ACKNOWLEDGEMENTS

This work was partially supported by DARPA/ITO within the SensIT contract #F30602-99-C-0171 and by DARPA/ATO within the Tactical Tags Feasibility Study contract #DAAD16-99-C-1024.

REFERENCES

[1] G. Asada, M. Dong, T.S. Lin, F. Newberg, G. Pottie, W.J. Kaiser, and H.O. Marcy, "Wireless Integrated Network Sensors: Low Power Systems on a Chip", *Proceedings of the European Solid State Conference,* 1998.

[2] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks", *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking",* 1999, pp. 263-270.

[3] J. M. Kahn, R.H. Katz, and K.S.J. Pister, "Next century challenges: mobile networking for Smart Dust", *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking*, pp. 271-278, 1999.

[4] John Matsumara, Randall Steeb, Randall G. Bowdish, Scott Eisenhard, Gail Halverson, Thomas Herbert, Mark Lees, and John D. Pinder, "Rapid Force Protection Technologies: Assessing the Performance of Advanced Ground Sensors", RAND Technical Report, Arroyo Center/NDRI, June 2000, available at: http://www.rand.org/publications/DB/DB262/

[5] Simon R. Saunders, Antennas and Propagation for Wireless Communication Systems, John Wiley & Sons, LTD, New York, 1999, pp. 94-97.

[6] W. M. Merrill, K. Sohrabi, G. J. Pottie, W. Kaiser, H. L. N. Liu, and J. Leong, "PicoWINS: Investigating the Feasibility of Wireless Tactical Tags", Final report of contract DAAD16-99-R-1006, US Army Soldier Systems Command, October 15, 2001.