Wireless Networking for Integration of Real-time Construction Metrology Systems

Lawrence E. Pfeffer
National Institute of Standards and Technology,
Gaithersburg, Maryland 20899
lawrence.pfeffer@nist.gov

Abstract

The evolution toward sensors/metrology systems with embedded processing holds great potential for construction automation, provided that effective, standard communications and data-exchange techniques can be developed. Unfortunately, most metrology systems have limited interfaces that slight the communications/data-exchange issues. Serial, hardwired links (e.g., RS-232, 422) are a common, low denominator.

The NIST virtual site simulator (VSS) is a test-bed system for exploring system architectures, interfaces, and data-visualization methods for construction automation. The architecture of the VSS has features useful for construction-site metrology systems, e.g., wireless networking and machine-independent data formats. The VSS system is described, and initial experimental results are presented, with emphasis on communications and data-exchange. Key aspects of the VSS system design include: a real-time, networked operating system; wireless Ethernet; machine independent data representations; and a data-exchange service based on a publish/subscribe model. This paper discusses how these architectural choices provide for mobility, robustness, efficiency, and extensibility, that are vital for construction automation. Finally, some conclusions and suggestions for future communications and integration work are presented.

1 Introduction

Recently, great progress has been made in metrology systems suitable for construction automation. In commercial, off-the-shelf (COTS) equipment, sensor progress has been made primarily in the areas most relevant (and easily quantified) for single-sensor systems, e.g. accuracy and bandwidth, rather than in the harder to quantify area of ease of integration. Individual, hardwired serial links (e.g., RS-232, 422) are a common, low denominator for many metrology system interfaces. Such interfaces typically have their own unique control and format idiosyncrasies, and usually require dedicated hardware and software to support them, i.e. dedicated serial I/O port, dedicated driver software. For systems with one such interface (or a small number) this style of interface is, if not elegant, certainly feasible. However, as the number of metrology devices in a system grows, this approach scales poorly. At the same time, embedded or local processing capabilities are becoming more common and inexpensive. The presence of embedded or local processing power for sensors makes it worthwhile to reconsider how sensor systems can be more easily and cleanly integrated into larger systems, such as those anticipated for construction automation.

It is unlikely that any one sensor (or sensor type) can provide all the information needed for construction automation systems. The need for multiple sensing systems is due to both the spatial expanse of typical construction sites (relative to most sensors' ranges) and due to the different sensing needs of different applications (e.g., relative vs. absolute positioning.) Thus, construction automation systems will typically need to make use of data from multiple sensors. Once multiple sensors are involved in a system, several significant integration problems become evident:

- **Communications:** How to get data from one location to another (or to many others), reliably and efficiently?
- **Format:** How to represent the data so that different suppliers and users of data can understand one another?
- **Protocol:** How to manage data in a dynamic, multi-userenvironment?
- **Sensor fusion:** How to mediate between differing observations, to maintain a coherent world view?

While these problems are not, strictly speaking, directly in the realm of metrology, they are vital to making metrology data available and useful to the rest of a

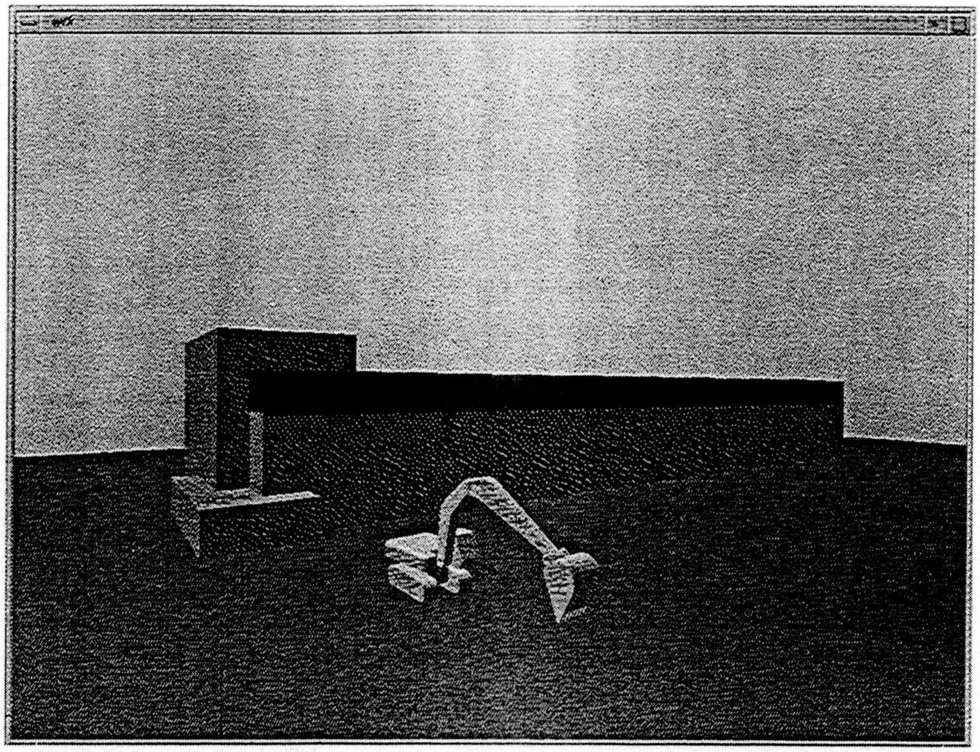

Figure 1. Graphical output from the Virtual Site Simulator: The VSS creates a 3-D, portrayal of a construction site, based on "live" sensor data from the sensors mounted on the model excavator. The sensor data are relayed to the UNIX-based graphics program over a wireless Ethernet link, using the NDDS data delivery software.

construction automation system.

The primary purpose of this paper is to discuss the communications, format, protocol, and sensor fusion issues, as they relate to gathering and using sensor data for construction automation, and show that these areas deserve careful consideration from the metrology community. To do so, this paper presents an initial approach to these problems (at the architectural level); presents some experimental results (see figures 1, 2, and 3); analyzes the VSS system design, and finally suggests areas that could benefit from further work.

2 Integration Issues

Due to the nature of typical construction environments and processes, usually more than one sensor/metrology system will be needed to provide all of the measurements required for a construction automation system. There are many reasons for this situation; among them are the physical expanse of typical construction sites, the often cluttered/complex site geometry, and the varying information needs during different tasks (e.g. absolute vs. relative measurements, different scales.)

In multi-sensor (and, almost certainly, multi-computer) systems, there are many system integration issues that must be addressed. Among the most important of these are communications, format, protocol, and sensor fusion. The balance of this section will present these issues, and will consider the aspects particular to construction automation.

One fundamental integration issue is communications. Sensor/metrology system data is useless if the data doesn't go anywhere; to be useful the data must be communicated to people or machines that can use it. In this paper, communications means moving the information (usually in a digital form) from one location to another. Communications channels are usually characterized in terms of bandwidth, error rate, and latency. Another closely related set of issues are those that relate to how the communication channel(s) are managed; e.g., how is communications started or stopped; whether a channel can be shared, or not. For use in construction automation, mobility (of both sensor systems and users of sensor data) is also very important, and thus there is an additional set of issues, namely, communications range, site coverage, and interference.

The second integration issue considered here is that of format. For the communicated data to be understood, there must be agreement between the producer(s)

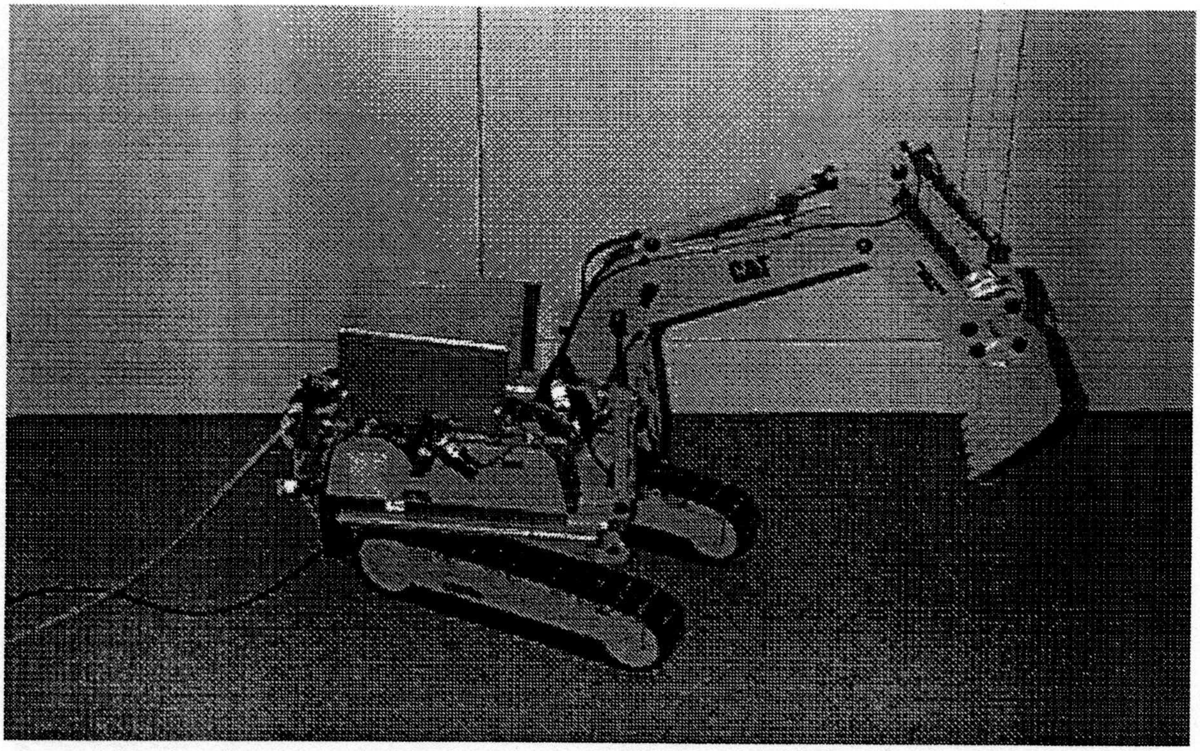

Figure 2. The scale model excavator used in the virtual site simulator. The Aluminum box holds the flux-gate compass and the pitch/roll inclinometers. Since the computer cannot fit onboard the model, the sensor signals connect via cables to the real-time computer. On a full-sized vehicle, the real-time CPU and the wireless Ethernet link would be onboard.

and the users/consumers of that data on how that data are represented, namely choices numerical types/precision, representastion of character data, choice of units, etc. In simple systems, with only one data producer and one consumer, format can be handled in an ad-hoc manner. However, once a system has more that one producer of information (i.e., sensors that measure the same thing, perhaps in different ways, or under different conditions), then the use of a common format to represent the data is needed for clean interoperability. Issues of format often conflict with those of communications. The communications issues generally favor concise representations, whereas the formats that best suit ease of use/interpretation are those that include self documentation information (e.g. units, type of numerical representation), and are thus considerably more verbose.

As used here, protocol refers to the rules for interaction among data producers, data consumers, and the underlying computer infrastructure (operating systems, communications system, etc.) For example, protocol encompasses the rules/procedures to request data, send data, or to query regarding the availability of data. Since construction automation tasks will frequently require many different pieces of equipment (from different vendors) to work together, protocols must be well defined (so that they really work), yet have some flexibility to accommodate system evolution.

Sensor fusion (or alternatively, world modeling) is the process of combining multiple, perhaps conflicting, measurements to produce a self-consistent world state that is optimal for some intended use. Since different uses of data imply different ideas of optimality, sensor fusion is, by necessity, application specific. For example, a servo-control system may require data that is recent, even if it is more noisy, in order to guarantee stable operation. Some other task, one that has a longer time scale, may be able to make use of data from some alternative source that is more accurate, but of lower bandwidth and/or longer time latency.

3 Experimental System

The approaches described in this paper were developed as part of NIST's research on construction automation. An overview of a representative system will provide a basis for further discussion of system integration issues. The Virtual Site Simulator is an experimental test-bed that brings together sensors, real-time computers, wireless communications, site/equipment models, simulation and graphics software into an integrated system. The major components of the VSS are described below, starting with the hardware.

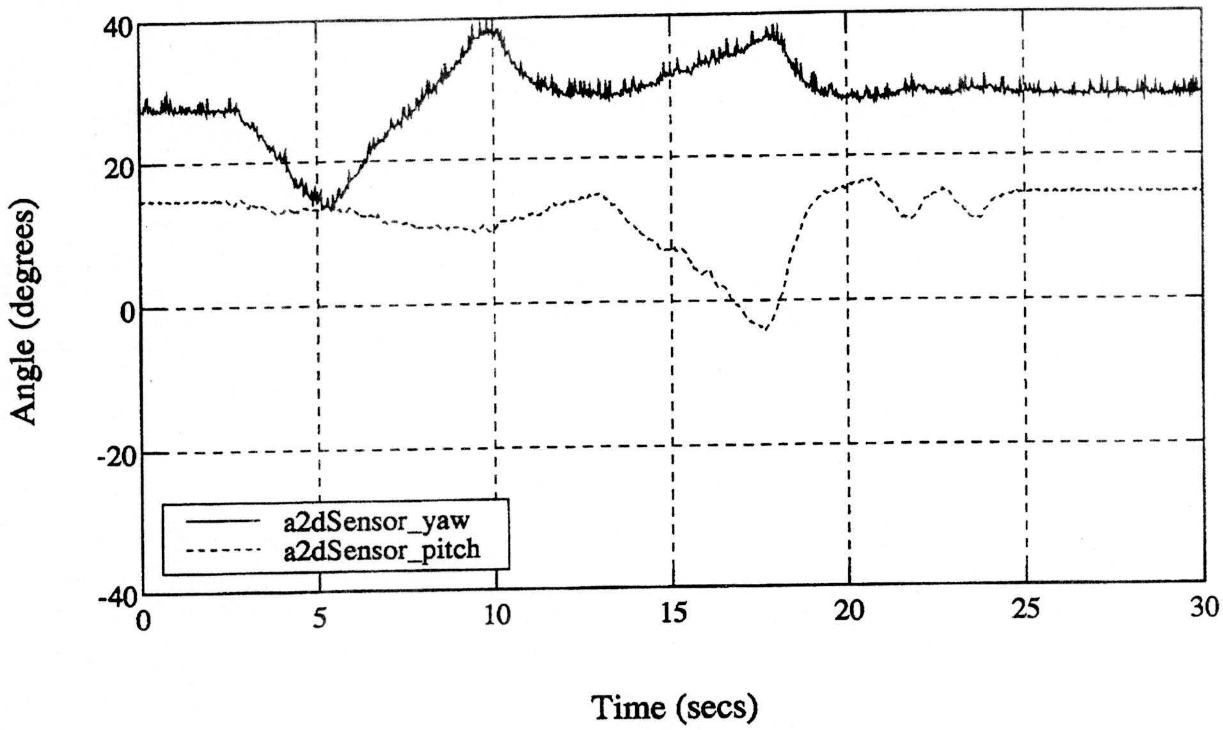

Figure 3 Sensor data from two of the vehicle-mounted sensors (flux-gate compass and pitch inclinometer.) The sensor data is relayed in real time to a UNIX workstation, via a wireless Ethernet link. The "live" data is plotted using the program StethoScope.

The initial implementation uses a small scale model of an excavator, rather than a full-sized piece of construction equipment, as a sensor platform for the VSS. Working with a scale model makes several aspects of the test-bed easier (cost, safety, indoor operation.) However, a key goal is to build another version of the VSS that works with a full-sized vehicles. The VSS vehicle currently carries three types of sensors: a compass, inclinometers, and joint-position sensors (potentiometers.) Unfortunately, this sensor suite does not provide the full kinematic state of the excavator model; for example, the VSS does not have sensors for the translational coordinates. These quantities are currently estimated, in an open-loop fashion. Future implementations will incorporate position sensing.

The sensors are connected to a real-time computer. The computer is a PC clone (Compaq 386) running a POSIX-conformant [1], real-time operating system, VxWorks [2]. This processor is connected to the rest of the VSS system with a wireless Ethernet link. Although the intent is to have the real-time computer mounted on the vehicle eventually, this is not feasible with the scale model, so the model and sensors is connected to the real-time computer via a tether cable.

Although the wireless Ethernet link makes it possible for the VSS real-time computer to interact with any system on the local network, most of the interactions are with two UNIX systems, a SUN-5 (which acts primarily as a file server and code development for the VSS real-time processor) and an SGI Onyx, which is used to run the graphical simulation of the "construction site." The graphical simulation is based on WorldToolKit [3], a graphics/virtual-reality package. The graphical simulation is based on real-time sensor data from the excavator-mounted sensors. The sensor data are relayed from the real-time computer to the UNIX-hosted graphical simulation via wireless Ethernet and the NDDS communications software [4], [5].

4 Architecture

Now that the hardware of the VSS has been outlined, a description of the system and software architecture will be more clear. In a matter similar to the hardware description, the software architecture will be described beginning at the sensors and moving toward the rest of the system.

All of the sensors are interfaced to a real-time computer. At present, all of the sensors in the VSS are read by sampling voltages via a multiplexed analog-to-digital converter. (The digital compass can also communicate through an RS-232 serial line, but this would merely slow down the current application, and tie up a serial port.) An A2D driver component in the real-time software reads voltages and converts the signals into application-friendly units. The resulting sensor signals are stored in a matrix, along with signal names and units. The A2D driver component is one small part of the software running on the real-time computer. A key purpose of this real-time computer, so far as sensing is concerned, is to encapsulate the different sensors data within a uniform data structure (e.g. matrices for VSS) and provide uniform ways to manipulate that data (e.g., transmit it over the network) independent of the original type of sensor or its native interface method (analog voltage, current loop, or serial bit stream.)

4.1 Real-Time Computer Software:

The software architecture of the real-time computer is far more significant that the details of the sensors, especially given that one of the purposes of this computer is to hide the sensor details. The key aspects of the real-time computer are the following:

- Real-time operating system (RTOS)
- Standardized operating system (POSIX)
- Standardized networking (TCP/IP, sockets)
- Support for real-time software development tools

In order to provide the quick response to events needed for construction automation, the operating system must be truly real-time in nature. That requirement effectively translates to the need for preemptive multitasking, quick context switching, and inter-task communication methods. For stability in a rapidly evolving technical field, a platform-independent, standardized operating system is also indicated. Thus, we selected a POSIX-conformant RTOS [1] that is available on a wide range of processors, VxWorks [2].

Standardized networking is a key to our systems integration approach. Thus it was vital to select an RTOS that supports standardized networking. For our purposes, this meant support for the TCP/IP layered protocol model [6], a BSD-style sockets network interface library, and device drivers for Ethernet (IEEE 802.xxx [7]) family hardware.

A final, but no less important aspect of the real-time computer was that of availability and support for software development tools relevant to real-time systems. This need goes beyond that for conventional compilers, linkers and debuggers, since real-time systems are not static systems. Time is, quite literally, of the essence in such systems; thus tools to examine signals during operation (and that change timing minimally, while doing so) are extremely helpful. Tools that support the block-diagram/data-flow paradigm common to many control systems are quite useful. The VSS was developed using ControlShell [8], StethoScope [9], and NDDS [5]; some other block-diagram/data-flow tools are discussed in [10].

4.2 Networked Communications

Networked communications, in this case Ethernet, provides the VSS with a method to move data from one location to another. Using networking standard based on the TCP/IP model [6] and the IEEE 802.xxx family of local area network [7] provides many other important benefits. The TCP/IP model is a layered one, and the layering provides an attractive level of abstraction, when one interfaces at or near the top of the protocol "stack." At the same time, the layering provides significant flexibility, as alternative implementations for laayers can be substituted, provided they conform to the specified interfaces. A prime example of the benefit of this flexibility in the VSS is the use of a wireless Ethernet link for part of the network. Although the physical transmission is completely different than the balance of the network, the rest of the hardware and software in the VSS is well isolated from that detail, and inter-operates transparently to it. More details on the wireless Ethernet are provided in section 4.3. The TCP/IP/Ethernet system of communication has another important benefit; it is well understood in the computer and Electrical Engineering communities, and it is broadly supported by industry. This situation facilitates rapid, economical development of networked systems.

Two more benefits that make Ethernet-based networking extremely attractive for metrology, and particularly for construction-site metrology are high data rates and shared use. First, the transmission rates are generally high, especially compared to typical RS-232/422 serial communications. Although the transmission rate is widely variable, depending on the physical media standard chosen (e.g. approximately 1.6 Mega-bits per second for the wireless Ethernet used in the VSS, to 100 Mega-bits per second for 100BaseT Ethernet), it is sufficient for most metrology applications, even when one considers the extra traffic of a large multi-user, distributed system. The key benefit of TCP-IP/Ethernet networking is that one connection can carry a wide variety of traffic, rather than requiring a dedicated hardware channel for each data stream. Further, the protocols make it easy to send different data, different amounts of data, and different types of data on demand, provided that the demands stay within the data-rate limitations of the media being used. The high data rates and the flexibility to use the network to send different data at different times, as needed, makes it possible to use such a network for all the communications in a distributed system. For example, the VSS uses its wireless Ethernet connection not only to send sensor updates out, but also to download its software, to receive commands, and to send/receive debugging information during program development. In fact, these operations can occur simultaneously, provided that the specific demands do not exceed either the network or the processor's capacity. In fact, by properly arranging priorities, etc., occasional network overloads can be handled gracefully.

4.3 Wireless Networking

Construction sites are generally large; typically hundreds of meters in expanse, and longer for bridges, dams, or airports, etc. In addition, construction sites have men, machines and material continually moving around. Because of the expanse and the continual movement, two things are evident about metrology systems used for construction automation. First, such metrology systems will generally have to be movable too, in order to make the measurements needed during different phases of the work. Second, such metrology systems will not be able to use wires to transmit information for any great distances; such wires would rapidly snarl or be destroyed by the movements of people or equipment on the site (or take inordinate effort to prevent wire tangling or breakage.) Thus, wireless communications is invaluable for construction automation, and wireless Ethernet is an attractive means to achieve that end.

A wide variety of wireless Ethernet equipment has been developed, originally using different transmission media (Infrared light, various bands of the R-F spectrum) and different modulation methods. Generally, these systems were not inter-operable. The inter-operability issue has recently been addressed by the IEEE P802.11 wireless LAN draft standard [11], although there are some issues, such as management of routing to mobile nodes ("roaming"), that are outside the scope of the standard. Nevertheless, the adoption of IEEE P802.11 should help inter-operability considerably, within any of the three IEEE P802.11-specified physical media.

The current target markets for wireless networking products does not include construction automation. More typical applications for wireless networking include building-to-building links; elimination of wiring inside buildings; or integrating portable computers onto a network. Construction automation is most analogous to the last of these, since the two share the characteristic of constant mobility. Fortunately, the characteristics of some of the existing products meet the needs of the VSS, in terms of range, data rates, and reliability. Many of the systems surveyed claimed ranges of hundreds of meters (in typical outdoor conditions, generally less inside building), and typical data rates are in the close order of 100 kilobits/sec. While this data rate is low compared to any of the wired Ethernet media (e.g. 10BaseT or 100BaseT), it appears usable for most metrology applications relevant to construction. Although the VSS has not been used to make any definitive tests of reliability, the equipment used in the VSS has performed acceptably.

4.4 Data Formats:

Although communications is necessary for systems scattered about a construction site, it does not, by itself, guarantee that the different systems involved will be able to understand one another. Mutual understanding also requires a common set of formats to represent basic quantities of interest (such as time, distance, angle, and mass), as well as other more complex quantities, such as names, coordinate systems, and tolerances. There are several important aspects to data formats, among these are:

- Machine independent representation,
- Extensability,
- Units,
- Efficiency.

Different computer systems use different internal representation. Even data as simple (supposedly) as integers may be of different sizes (e.g., 2, 4, or 8 bytes), may be encoded differently (2's-complement vs. sign, magnitude), and may be stored in different order (big vs. little-endian) on different machines. (Surprisingly, a byte, has not always been eight bits, although that has become the common practice.) Similarly, character data is not universally represented using the ASCII character set; there are others, such as the ISO 8-bit character set. In order for a set of computer systems to begin understanding one another's data, there must be agreement as to the fundamental types of information to be exchanged, and there must be agreement on the sizes, byte-order, and encoding of those basic types.

Since some of the data relevant to metrology is not representable using a single, simple type (Cartesian coordinates, orientation, etc are not scalar quantities) there is also the need for agreement on the representation of compound types to represent these complex data objects. While many of these may be represented by vectors or matrices of simple types, there is still the need to represent general compound structures. Similarly, there is a need to be able to extend the vocabulary with new data types and structures, as the system evolves. Agreement on units of measure (or, the ability to recognize and convert among dimensionally-compatible units) is also vital. Finally, there is the issue of efficiency vs. that of self-documentation. Given an agreed-upon meta-format, one could include all of the representation information (size, byte order, encoding, units, etc.) with each datum so that it could be interpreted as a self-contained unit. However, such a representation is inefficient in both storage space, time to communicate, and time to interpret on the receiving end. On the other hand, completely "naked" bits, without the context to interpret their intended meaning, is an approach that could invite costly system errors. Neither extreme is attractive, and either a compromise, or a flexible approach facilitating exchange of context information when needed, seems better than either extreme.

The current approach to data representation used in the VSS experiments is based on the eXternal Data Representation (XDR) standard originally developed by SUN Microsystems, Inc. The use of XDR for the VSS experiments is dictated by the use of the NDDS package [5] to handle the format and protocol aspects of system integration. The XDR standard, developed as part of SUN's approach to network computing, has been widely adopted by UNIX systems, and has been ported to other platforms. XDR provides a machine-independent representation of many simple data types; provides methods to represent compound types; as well as means to extend the vocabulary to user-defined types. By itself, XDR is not a complete panacea for data exchange. One limitation is that XDR depends on compile-time consistency of definitions among all the systems involved. Units of measure are not addressed within XDR, nor does it provide much capability for self-documentation; those need to be addressed separately. In the VSS experiments, units and a limited amount of self-description is provided by the use of a matrix-type data structure (the CSMat, [8]) that has some provisions for both units and element names.

4.5 Protocols for Data Exchange:

As used here, data-exchange protocol means the rules and mechanisms for providing and acquiring data. Implicit in such protocols are the semantics that it implements, i.e. the underlying meaning or model that the protocol supports. For the VSS system, we have chosen to make use of a commercial package, NDDS [5] that is available for VxWorks, several versions of UNIX (and apparently, for Microsoft Windows '95, although we haven't used that, yet.) There are many alternatives for data exchange protocols and implementations (e.g. CORBA, TCA/TCX); some of these are discussed in [4] (see sect. 4.2.)

The underlying model for NDDS is one of publication and subscription to data instances, independent of machine identities, rather than one of sending and receiving to specific machines. Thus, rather than querying a specific system for a given piece of data, one requests a subscription to that data, without having to know the name of the source. There are two aspects of this type of arrangement that are attractive in distributed systems, particularly those involved in real-time monitoring and control. First, the system is decentralized, in that one can obtain data by the data's name, without knowing the name of specific data source. Secondly, the subscription provides continual updates (until you cancel the subscription) from a single subscription request. The protocol provides a means to be notified if data become unavailable, as well as a means to protect against being flooded by data updates at a rate greater than the subscriber can accept. In a conventional client/server architecture, one would have to make a request for every update. In a distributed system, the extra overhead of the continual requests can clog the network, especially if there are many high-update-rate activities going on at once, and the network data rate is limited. NDDS provides one-time queries as well, but its main attractiveness for applications like construction automation, is that it provides a protocol that closely matches the needs of the application -- to receive timely information with minimal overhead.

4.6 Sensor Fusion

The NDDS protocol provides a means for mediating between multiple producers of the same data. Similarly, it provides means for addressing the temporal issues critical to real-time systems. Multiple systems can each publish the "same" data signal' each assigns a strength to its publication, which can be though of as a truthfulness or signal quality. (However, it is left up to data publishers to assign sensible values to these strengths; if no convention is followed, the strengths become meaningless.) Similarly, publishers specify a persistence for the data, analogous to the spoilage dates on dairy products. Similarly, publishers must specify persistence times for data that are reasonable, based on the nature of the sensed quantity and the metrology implementation used to measure it. Although this is a very simple type of sensor fusion, it does address some of the major needs of distributed monitoring/control systems, particularly the temporal issues.

5 Analysis and Conclusions

5.1 Sensor Interfaces

At the internal sensor-to-computer interface level, designers of metrology systems will continue to be driven by performance issues (accuracy, bandwidth) rather than by ease of integration into larger systems. However, for the instrument's interface to the outside world, ease of integration is a factor worthy of more weight than it has generally received. Since interfaces that require a dedicated interface (such as RS-232) complicate multi-sensor systems, alternatives that share communications resources (e.g., networks like Ethernet) are particularly attractive for integrating metrology systems into construction automation. However, since many metrology users in other fields do not have multi-sensor requirements, dedicated interfaces will still be desirable, and networked interfaces to general-purpose metrology systems might be better incorporated as options.

5.2 Real-time Operating Systems

Metrology systems with embedded processing need real-time functionality. While this can be developed from scratch, the increasing complexity of systems biases the decision toward using an existing RTOS. There are a wide variety of both commercially-supported RTOS [12], as well as some freely -available ones [13]. POSIX conformance is helpful, in that it provides standardization, and thus considerable portability with respect to hardware and software suppliers. However, POSIX conformance is not necessarily the biggest issue in RTOS; instead, much attention is currently being paid to Microsoft's Windows NT operating system. So far, it is unclear how suitable Windows NT is (or will become) for various real-time applications. This is currently the subject of active discussion in the real-time community. One source of such discussion is the newsgroup comp.realtime.

5.3 Networking

The TCP/IP network protocol family is widely supported and well tested. It forms the protocol backbone of the National Information Infrastructure (the US portion of the global Internet), and has been adopted by much of the rest of the world. For these reasons, it is reasonable to use TCP/IP as the basis for networking in construction automation systems. The Ethernet local area network (IEEE 802.xxx family) is one of best understood and best supported choices for the physical and lower levels of a network. A recent version of Ethernet, IEEE 802.11, is being developed specifically for wireless LANs. Equipment vendors are beginning to advertise and ship equipment they claim is 802.11-compatible (although, how compatible and inter-operable remains to be seen.) For these reasons, 802.11-based equipment is a promising technology for construction automation systems that require mobility.

Ethernet may remain too expensive to include in some instruments. Even for those cases, some form of networked communications (perhaps as an add-on option) may facilitate integration into complex systems, rather than dependence on dedicated, one-to-one communications hardware. Other local area networks (e.g. CAN, fieldbus, PROFIBUS, etc.) may provide lower-cost alternatives to wired Ethernet. The electronics markets are very dynamic and bear careful watching, to see which network standards will dominate and persist. On a multi-sensor platform (e.g. a construction machine) the use of a single wireless Ethernet link for off-board communications, coupled with some lower-cost network to connect the on-board sub-systems, may be an attractive design.

5.4 Data Formats

XDR is one approach to some of the format/data exchange issues confronting construction automation. XDR imposes certain overhead requirements, and it does not address all of the issues. However, it is widely supported, extensible, and covers a substantial part of the data exchange problem, namely the machine independence issue. However, it does not, by itself, define all of the metrology/CA data types needed, nor does it address units. The metrology and construction automation communities have to develop solutions and (hopefully) reach consensus on these issues. Our intent is to use the VSS as a test-bed for developing a set of data types for use in construction automation, starting with metrology, and to make these available for others to evaluate and use.

5.5 Data-Exchange Protocols

Similarly, NDDS is one of several different approaches to application-level communications protocol (and implementation) in distributed systems. NDDS is not necessarily the best answer, nor is it a complete solution. However, it has proved quite useful in experiments using the VSS hardware. It does address several issues: host-name independence; robustness to configuration changes; efficient use of network; temporal issues. NDDS is proprietary; however, we note that other vendor-developed standards (e.g. XDR) have sometimes evolved into industry standards. NDDS does not presently address security issues, and security may become very important to reliable construction automation, as it has to other distributed systems, such as electronic commerce. We will continue to evaluate alternatives, and will consider switching, if a better alternative becomes available.

5.6 Data-Fusion

NDDS provides a simple mechanism for resolving between multiple producers of the same piece of data, and NDDS does address temporal issues in a way more suitable to real-time systems than many of the alternatives. However, although the resolution mechanism takes some adjustable parameters, it is limited in its flexibility. In particular, the publishers of information have virtually all the control over the fusion process. Thus, information consumers that have specific data-fusion needs are not presently given a means to decide for themselves how they want to choose the best data for their application; NDDS makes its decision solely based on the strengths and persistence of the published data. While there are work-arounds, these require a high degree of cooperation among data producers (e.g. publishing data under multiple names), and generally work against the spirit of NDDS's current data-fusion model. Development of a more flexible data-fusion system, that would work efficiently in a dynamic environment is a worthy topic for further research.

5.7 Overall Recommendations

The use of communications interfaces that can share hardware resources, such as LANs, can greatly facilitate the integration of multi-sensor systems, and thus will benefit many construction automation systems. Due to their standardization and broad acceptance, the TCP-IP protocols and Ethernet LAN hardware are a good foundation. Incorporation of IEEE 802.11-based wireless Ethernet is particularly attractive for construction-site applications that require mobility.

No set of formats or data-exchange protocols is going to perfectly satisfy all metrology (or other data exchange) needs in construction automation. The use of common formats will generally require users to either adopt the data-structures of the format standard for their internal use, or accept the burden of translation back and forth. Similarly, any protocol will have its limitations, and will inevitably favor some system choices over others. However, without some agreement on data formats and exchange protocols, the "Tower of Babel" situation will persist, and that scenario may jeopardize the future of construction automation. Thus, development of a set of useful, standard formats and data-exchange protocols (with attention to the related integration issues) is vital to the success of construction automation.

6 Acknowledgments and Notices

The author would like to thank Dr. Kent Reed and Dr. William Stone for their generous assistance and their technical contributions to the VSS project.

Certain trade names and company names are mentioned in this paper; in no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purposes discussed. All trademarks, servicemarks, etc. are property of their respective holders

7 References

[1] IEEE Std. 1003.1 (ISO/IEC 9945-1). Portable Operating System Interface (POSIX). 1003.1: System Application Program Interface; 1003.2: Shell and utilities; 1003.3: Test methods for measuring conformance; 1003.4: Draft real-time extensions. http://stdbbs.ieee.org/

[2] VxWorks 5.2 Programmer's Guide, Wind River Systems, 1010 Atlantic Ave., Alameda, CA 94051. (510)748-4100. http://www.wrs.com/

[3] WorldToolKit Reference Manual, Release 6, SENSE8 Corp., 100 Shorline Highway, Suite 282, Mill Valley, CA 994941, http://www.sens8.com

[4] Pardo-Castellote, G., Experiments in the Integration and Control of an Intelligent Manufacturing Workcell, Ph.D. Dissertation, Stanford University, Dept. of Electrical Engineering, June 1995. Available from Universty Microfilms, also available as Stanford University SUDAAR 675.

[5] Network Data Delivery Service User's Manual, version 1, Real-Time Innovations, Inc., 155A Moffett Park Dr., Ste 111, Sunnyvale, CA 94089. http://www.rti.com

[6] Comer, D. Internetworking with TCP/IP, Prentice Hall, Englewood Cliffs, NJ 07632, 1988

[7] IEEE Std.802.xxx family of standards (see also ISO/IEC8802-xxx), IEEE Customer Service, Piscataway, NJ 08855, (800)678-IEEE, http://stdbbs.ieee.org/

[8] ControlShell Programmer's Reference Manual, version 5, Real-Time Innovations, Inc., 155A Moffett Park Dr., Ste 111, Sunnyvale, CA 94089. http://www.rti.com

[9] StethoScope User's Manual, version 4.2, Real-Time Innovations, Inc., 155A Moffett Park Dr., Ste 111, Sunnyvale, CA 94089. http://www.rti.com

[10] Pfeffer, L. and Tran, H., Key Characteristics for Software

for Open Architecture Controllers, in Open Architecture Control Systems and Standards, Frederick E. Proctor, Editor, Proc. SPIE pp. 36--44 (1997)

[11] IEEE P802.11 Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, draft, IEEE Customer Service, Piscataway, NJ 08855, (800)678-IEEE, http://stdbbs.ieee.org/

[12] "Real Time Operating Systems and Kernels," in Real-Time Engineering Vol. 3 No. 2, Summer 1996.

[13] RTEMS Real-Time Operating System (available under terms similar to the Gnu "copyleft" license). http://lancelot.gcs.redstone.army.mil/rtems.htm