LUIGINO BENETAZZO, MATTEO BERTOCCO, CLAUDIO NARDUZZI

Università di Padova
Department of Information Engineering, Italy
{luigino.benetazzo, matteo.bertocco, claudio.narduzzi}@unipd.it

SELF-CONFIGURING MEASUREMENT NETWORKS

This discussion paper presents issues in the design of self-configuring networked measurement systems. Key requisites are summarized at first. Then, architectural features will be analyzed in order to let a designer be aware of advantages and efforts of presented choices, that allow a revolutionary shift of emphasis from a PC-centric measurement architecture, to a network-centric architecture.

Keywords: remote and distributed measurement, self-configuring networks, instrumentation

1. OVERVIEW

An interesting issue coming from the telecommunications area consists in the ability of a set of independent nodes to set-up a communication network, with the possibility to optimally route and deliver messages. The concept can be borrowed and extended to the field of measurement and instrumentation, in terms of the ability to build a self-configuring network of measurement nodes. Such vision matches two aspects: the abstraction of measurement functions from the underlying instrumentation hardware and the capability to discover and implement the configuration best suited for the assigned task.

There are several reasons beyond the prospective interest in self-configuring measurement networks. In some cases the main reason is just a *need*. For instance, in key applications in the field of telecommunications a transmitter, a receiver and a measuring instrument, located far from each other, have to be controlled so that useful information is obtained. Hence, some mechanism is required to let the test engineer accomplish the task. Less trivially, when quality of service of an entire network is under investigation, a common procedure consists in setting up a monitoring network composed of many test points at different locations. In this case, there is again the need for an architecture that allows to easily set-up a network of remote test points which, however, should collect data in a coordinated way. The end user should be able to see the whole monitoring network as a single measuring device yielding the required information and analyzing it.

Environmental monitoring is perhaps one of the most interesting fields for self-configuring distributed measurements. In this case it can be assumed that a rather large number of sensors are distributed in a geographical area. Procedures need to be set up in order to interact with the sensors, both for setting sensor parameters and for collecting data coming from the sensors themselves. At a higher degree of abstraction, such data are used to extract information having a more general meaning: for instance, one may detect alarm conditions (e.g., landslides) in a given region from the survey of geological information. However, correct interpretation is only possible if each node in the measurement network is coordinating itself in a proper way with the other devices.

The field of education and e-learning is another case where remote access to electronic instrumentation and to devices operating as a self-configuring network provides new challenges. Electronic systems are becoming complex, expensive and evolve quickly, so that a single Technical School or University many not have sufficient resources to keep pace with technology updates. Moreover, students often use these complex resources for a limited amount of time: once a sufficient skill level has been gained, a certain set of instruments may no longer be necessary as the student progresses to new and different measurement problems. Hence, costs are paid back only if a large number of people can share physical resources in a time-scheduled way. This means a variety of specialized test labs can be set-up, each offering a number of experiments with state-of-the-art instruments, via a remote access policy. Again, this view fits in the concept of a self-configuring measurement network. In fact, a set of instruments and devices (such as programmable interconnect switches and matrices), have to be continuously rearranged to suit the requirements of different teaching experiments and student skills.

Dedicated networks have been used for a long time in the control of industrial plants. Now, the emerging trend of using ethernet also for field bus operations opens a new opportunity of considering classical instrumentation and field sensors or actuators from a common point of view. Applications can benefit from such a unified view, where services can now share resources. Considerations on the cost of sophisticated measuring equipment, replacement of obsolete instruments, efficient use of available resources, motivate the quest for an environment where the test procedure is well defined and required measurement functions are provided on demand by available instruments. Furthermore, in an integrated factory environment the use of abstract and well-formalised measurement procedures allows better integration between design engineering and test engineering tasks. Wider access to testing facilities enables people to closely interact, while enabling a quicker development of test procedures. In fact, software code and devices developed for testing purposes can be more easily shared, providing greater efficiency.

The above examples show the potential advantages of a unified view, in the form of self-configuring measurement networks. In spite of strong differences among the considered applications, some important common needs can be identified and the general architectural features that give self-configuring networking capabilities to a measurement application can be highlighted.

Of course, the implementation of a complex test and monitoring system cannot be an occasional combination of devices, but it is perfectly reasonable to assume that a given selection of instruments can be combined, on a temporary basis, to perform a given task.

2. SELF-CONFIGURING MEASUREMENT NETWORKS

The aim of this discussion paper is to outline and discuss the challenges behind the implementation of a measurement procedure by a self-configuring networked measurement system. The general structure of a networked measurement environment is illustrated in Fig.1. Only wired connections are presented in this figure, but it is of course possible to also consider wireless connections to some, or all, of the measurement nodes. The important fact that needs to be emphasized is that no *a priori* assumption is made on how each node will interact with the rest of the network.

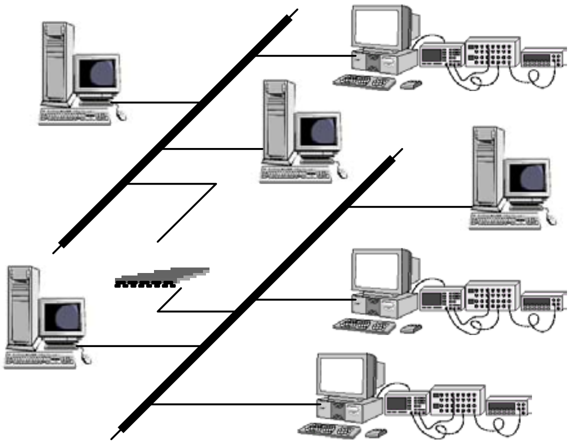

Fig. 1. A networked measurement environment.

In a self-configuration view, each measurement node will simply let the rest of the world know about its own capabilities regarding a number of features:

- the ***measurement capabilities***, represented by the measurement functions the node can carry out, together with relevant information regarding measurement ranges, calibration and any further information concerning the accurate generation of measurement data. In the field of smart sensors, this can be equated to the IEEE 1451 transducer electronic data sheet (TEDS);
- the ***connectivity functions*** supported by the node, i.e., the kinds of physical media and protocols that the node can use to communicate with other nodes in the network;
- the ***synchronization*** functions that the node can implement. Nodes could be located in a restricted space, for instance, if the self-configuring network paradigm is applied to automatic test equipment connected by an LXI interface. Conversely, in some environmental sensing applications they might be spread randomly over a large territory. Consequently, the requirements for accurate node synchronization might vary over several orders of magnitude. Different synchronization procedures might be supported, notably, IEEE 1588 Precision Timing Protocol, particularly when dedicated high-speed connections are available. For less demanding applications, even the standard Network Timing Protocol (NTP) might be considered. When extreme accuracy is necessary, the option of choice would probably be GPS-based timing;
- the ***security features*** that can be employed to reliably transfer measurement information. In the context of networking measurement systems, security does not mean just prevention against hacker attacks, but the guarantee that unauthorized uses of devices are avoided, so that at any time the consistence, accuracy and traceability of measurement information can be documented. In an open networking environment, even unintentional intrusions may have catastrophic effects on the set-up of measurement nodes and on the correct interpretation of results. Of course, it also has to be remembered that access to valuable data may need to be restricted to a limited number of authorized parties;
- the ***community services***, that represent the set of information and common procedures that a measurement node must implement and support to allow its functions and services to be offered to the rest of the world. These services may include optional functions, such as data storage, post-processing and data compression algorithms, etc.

This set of features is graphically presented in Fig. 2. It has to be emphasized that the model is totally independent from a number of underlying features, such as hardware platforms, specific communication protocols, programming languages and so forth.

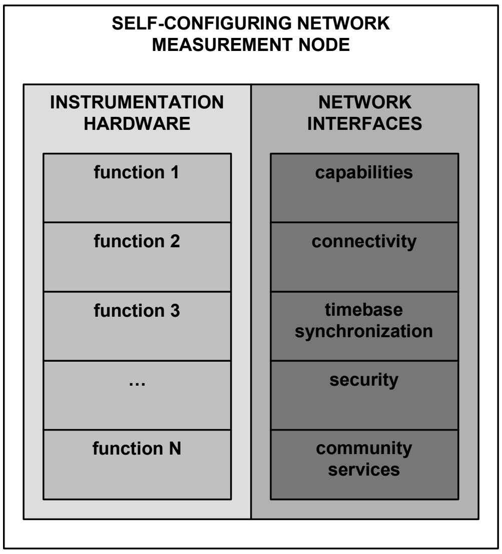

Fig. 2. Fundamental components of a measurement node in a self-configuring measurement network.

Once these interfaces have been formalised, attention can turn to the general architecture of an self-configuring measurement network; Fig. 3 proposes one possible structure. Although in the figure measurement nodes and smart sensors are presented as different units, this is not a conceptual distinction. In fact, smart sensors designed according to the IEEE 1451 standard are very close to the abstract model proposed in Fig. 2.

The self-configuring network architecture includes some special functions that need to be discussed, namely, the lookup service, the sensor broker and the application broker. It has to be noticed that these functions need not be centralised within dedicated nodes, but can be spread throughout the network.

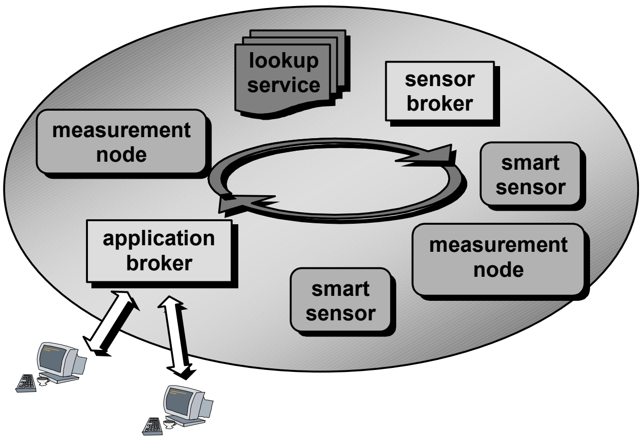

Fig. 3. Functional elements of a self-configuring measurement network.

More specifically, these functions implement the following main features:

- the ***lookup service*** has the task of receiving from all network nodes the notifications about their functions and capabilities, as contained in their network interfaces; it deals as well with the enquires about the services needed to perform a given measurement task;
- the ***sensor broker*** is responsible for determining the availability, within a set of nodes, of the basic measurement functions required to fulfil a specific task;
- the ***application broker*** provides directions to access algorithms that a client can use to accomplish a given measurement procedure.

To better explain the role of the above elements two examples are considered.

We consider first sensor networks for distributed monitoring of some environmental parameter of interest. The measurement nodes are, of course, sensors that could be provided with a wireless link. The nodes can be grouped together according to different schemes, basically in a hierarchy of clusters. The exact configuration depends on a number of factors, for instance power consumption, or the distance between nodes. The application broker provides algorithms that enable the end user to observe the data which are useful to his measurement needs. Depending on the actual algorithm selected by the application broker, the sensor broker chooses the better configuration of the subset of available sensors actually needed for the acquisition of the required data. Moreover, the sensor broker may implement checks, to ensure that data retrieval is carried out by an application that has adequate security credentials, while the application broker may implement services that associate to each end-user the correct rights.

As a second example, suppose that in a learning context a number of experiences have to be made available to several students within the framework of some shared laboratories. Devices Under Test (DUT's) would be connected to instruments via matrix switches, in such a way that the same instruments can be used for different experiments in a time-sharing scheme. In this environment, the lookup service is tasked with registering references that allow to discover whether the physical resources, as well as procedures, that the user may require at local workstations can be retrieved from some node in the measurement network. The sensor broker then would provide a specific measurement application with ways to access physical resources, and is eventually responsible for the time-sharing of hardware in a multiple user scenario. Finally, the application broker would deliver to the end-user the software procedures he needs to perform a specific experiment.

3. SYSTEM COMPONENTS

In the previous section, an ideal architecture from a self-configuring measurement network has been outlined. It is now appropriate to carry out a brief analysis of the main requirements and off-the-shelf solutions that a designer might consider. Then, approaches for networking measurement applications, from the instrument up to the whole system, are discussed. The aim is, on the one hand, to assess what tools are already available for building a complex measurement system, on a self-configuring basis, from the combination of different instruments; on the other hand, to point to open issues where further research and new developments are needed.

While several among the measurement node features discussed above are quite general, and can be implemented by relying on the most recent achievements in information and communication technology, some aspects are specific to the field of measurement. Therefore, we shall discuss them briefly.

Several significant achievements have occurred recently, from the continued development of IEEE Standard 1451 on distributed instrumentation, to the recent introduction of LXI (*LAN eXtensions for Instrumentation*). The latter represents a revolutionary shift of emphasis from a PC-centric measurement architecture to a network-centric architecture. The important implication of these approaches is that each instrument is allowed to operate as an independent entity endowed with considerable intelligence and computing power. At the same time, intelligent instruments can coordinate their operation through the network. Hence, mechanisms for the setting-up of self-configuring networks, such as those outlined above, acquire considerable interest in the field of measurement.

Synchronization in a networked environment can be obtained by network protocols, as already mentioned. In particular, IEEE 1588 is, arguably, the standard of choice (for instance, in LXI) to achieve accurate time alignment among different instrument timebases. It is thus possible to exactly schedule the steps of a complex measurement procedure. However, several open issues remain where the accurate synchronization of geographically distributed networks is concerned, or when power-constrained wireless nodes are considered. In these cases efficient algorithms, that can operate correctly with network architectures including slow links or time variable configurations, still need to be developed.

It also has to be emphasized that calibration of a self-configuring measurement network is an issue of considerable interest. Some significant work has already been carried out on the calibration of distributed sensor networks by remotely controlled travelling standards, and relevant references are included in this paper.

Device-independence is another important requisite for self-configuring networked instruments. Since a system is set-up on a temporary basis, device-independent interfaces are essential to allow discovery mechanisms, whereby instruments within the system recognize each other. In turn, this makes system set-up and configuration simpler, although this is obtained at the expense of higher computing power requirements. In fact, device independence can be ensured by middleware software that interacts with hardware on one side, while offering generalized application programming interfaces. Examples of off-the-shelf solutions for measurement systems are represented by IVI Drivers or, at a lower level, by VISA drivers. It is however noticed that these proposals cover only a limited number of interesting cases, while many other practical devices are not yet supported.

Assuming that measurement nodes having the features outline above are available, the implementation of a general architecture, like the one proposed in Fig.3, places further requirements. One should note that the literature presents several examples of multi-level client-server architectures which, however, are hardly suitable to realize a highly reconfigurable system. In particular, the implementation of system access policies from non-centralized points, the features associated to dynamic application delivery of the application broker and, finally, the intelligent sensor management offered by the sensor broker are all functions for which the client-server paradigm is not well suited.

Although, in principle, the architecture of Fig. 3 could be built up from scratch, some environments already successfully used for other purposes can be exploited. It has to be remembered that the tools offered by these environments are by no means a ready-made solution. They rather represent a very helpful starting point which can make work easier. In the following, we briefly present some relevant examples of such environments.

A first example is constituted by *Remote Method Invocation* (RMI), that enables a generic application to run procedures on a foreign host. Although interesting, RMI provides in practice a basic support for the development of measurement networks, and one could prefer an environment providing some additional feature so that coding could be made easier. For instance, the *Enterprise Java Beans* mechanism is an example of software code delivery and remote resource invocation that adds some useful features at the price of some constraints. An interesting extension is constituted by the JINI framework, whose successful applications in information technology can be almost directly related to the conceptual view presented in Fig. 3. For instance, look-up services are already implemented in JINI, as well as the integration with web services.

A promising development would be the adoption of the GRID architecture. This was initially developed in the context of distributed computing, but several extensions to other application fields are currently explored by the research community. For instance, a GRID infrastructure could be used for exporting the data collected by measurement nodes: in this case, a smart sensor would be seen as a GRID resource. It might be argued that GRID nodes are usually assumed to have a considerable amount of computing power, which might not be the case for a single sensor. This raises a number of issues about how a measurement network could be best mapped into a GRID structure, suggesting, for instance, the possibility that GRID nodes are sensor clusters, rather than single units. The attractive aspect is that activities like the post-processing of data from a geographical network, the archiving of historic data and the cross-analysis of information from different measurement networks are well suited to features of a GRID. This motivates the interest in GRID applications in the field of measurement, which has already brought some results.

One word of caution concerns the *learning curve* for developers of a self-configuring remote measurement system, which should be kept as short as possible. One should note that networking architectures imply a number of features and configuration parameters previously unknown by technicians in the measurement world, which cannot be dealt with in a simple fashion. Development of applications for a dedicated measurement system can be made comparatively simple by the use of commercial development environments, such as LabVIEW from National Instruments and VEE from Agilent Technologies. The simplicity of use of these environments may have to be paid for in terms of lack of integration with higher conceptual levels, such as GRID. Hence, if self-configuration, as may be required in measurement networks, is a critical issue, different programming languages could be considered.

From the application developer viewpoint, tools should be provided that enable a designer to see the whole system in a generalized form, so that attention is concentrated on the sequence of input stimuli to be provided and the processing of corresponding answers received from the device under test. Formal approaches based on XML and UML as specification languages are being considered by research groups and should be brought to the readers' attention. For instance, XML has been already advantageously used to describe virtual instruments both in terms of user interface and functionalities, in such a way that a general purpose tool can build up a complete measurement application starting from a formal description written in an XML file. Similar work has been already done in the telecommunications field for the description of complex systems. Hence, extension of such applications to the more general context of self-configuring measurement networks would be an interesting research topic.

4. FINAL REMARKS

Many issues are still open, from the points of view of both research and pure deployment of a well-defined distributed measurement application.

A first aspect is interoperability, that is the ability that a self-configuring distributed measurement network could exhibit, of joining another net and sharing data as well as resources. Since many conceptual components are involved, such as protocols, supported hardware, desired features, software tools etc., a number of constraints arise. Consequently, different models may be developed, that in turn may be practically compared.

A second aspect is the robustness of a measurement network to different impairments such as noise, hardware failures, or communication failures. How such detrimental effects impact on the architectures one may adopt, and how each self-configuration scheme responds to each failure is a matter of investigation. By the way, impairments themselves are not well-defined, hence a comparison between different solutions now lacks proper performance indices.

Moreover, calibration of a distributed measurement system is still a research topic. In particular, there are no generally agreed calibration techniques when the measurement devices are far from each other or the measured quantity is by definition distributed. Once more, calibration of a measurement network that is not statically defined, is an even more complicated theoretical problem. To look one step further, even assuming that some calibration methods are well-known, one may require self-calibration or self-check techniques, for instance in order to enable a self-configuring net to discover one out-of-calibration device, and consequently to apply adequate recovery procedures.

Indeed, the previously discussed aspects show that a satisfactory model for a self-configuring measurement network is still a subject for scientific work.

In this context, just one possible architecture has been highlighted, and some off-the-shelf solutions have been considered as possible starting points for the implementation of the corresponding components.

The aim of this discussion paper is to help the pratictioner or a researcher understand the challenges beyond a self-configuring network

ed system. It is hoped that useful guidance has been provided for an efficient design of the final application or in clever selection of already-available application tools. For this reason, an extended list of references is reported.

REFERENCES

1. Grimaldi D., Nigro L., Pupo F.: *Java-based distributed measurement systems,* IEEE Trans. Instrum. Meas., pp. 100-103, 1998.
2. Bertocco M., Ferraris F., Offelli C., Parvis M.: *A client-server architecture for distributed measurement systems*, IEEE Trans. Instrum. Meas. vol. 47, no. 5, pp. 1143-1148, 1998.
3. Spoelder H.: *Virtual instrumentation and virtual environments,* IEEE Instrum. Meas. Mag, vol. 2, pp. 14-19, 1999.
4. Lee K. B., Schneeman R. D.: *Distributed measurement and control based on the IEEE 1451 smart transducer interface standards*, IEEE Trans. Instrum. Meas., vol. 49, no. 3, pp. 621-627, 2000.
5. Dricot J. M., De Doncher P., Diericks M., Grenez F., Bersini F.: *Development of distributed Self-Adaptive Instrumentations Networks using JINI technology*, Proc. IEEE Internat Workshop VIMS/2001, (Budapest, Hungary), pp. 22-27, 2001.
6. Gui Yun Tian: *Design and implementation of distributed measurement systems using fieldbus-based intelligent sensors*, IEEE Trans. Instrum. Meas., vol. 50, no. 5, pp. 1197-1202, 2001.
7. Edwards W. K.: *Core Jini.* 2nd ed. Prentice-Hall, 2001.
8. Potter D.: *Smart plug & play sensors*, IEEE Instrum. Meas. Mag., vol.5, pp. 28 - 30, 2002.
9. Moczar G., Csubak T., Varady P.: *Distributed measurement system for heat metering and control*, IEEE Trans. Instrum. Meas., vol. 51, no. 4, pp. 691-694, 2002.
10. Winiecki W., Karkowski M.: *A New Java-Based Software Environment for Distributed Measuring Systems Design*, IEEE Trans. Instrum. Meas., pp. 1340-1346, 2002.
11. Bertocco M., Cappellazzo S., Flammini A., Parvis M.: *A multi-layer architecture for distributed data acquisition*, Proc. 19th IEEE Instrum. Meas. Technol. Conf., pp. 1261-1264, 2002.
12. Bertocco M., Cappellazzo S., Narduzzi C., Parvis M.: *A distributed sensor network based on JINI Technology*, Proc. IEEE Internat Workshop VIMS/2002, pp. 68-71.
13. Bucci G., Fiorucci E., Landi C.: *Digital measurement station for power quality analysis in distributed environments*, IEEE Trans. Instrum. Meas., vol. 52, no. 1, pp. 75-84, 2003.
14. Wheelwright L.: *The network centric system*, Proc. IEEE Internat. Conf. AUTOTESTCON/2003, pp. 44-47.
15. Benetazzo L., Bertocco M., Cappellazzo S., Pegoraro P.: *Platform independent architectures for cooperative remote control of automated test laboratories*, Proc. IEEE Internat. Conf. AUTOTESTCON/2003, pp. 52-57.
16. Bertocco M., Cappellazzo S., Carullo A., Parvis M., Vallan A.: *Virtual Environment for Fast Development of Distributed Measurement Applications*, IEEE Trans. Instrum. Meas., pp. 681-685, 2003.
17. Ferrero A., Salicone S., Bonora C., Parmigiani M.: *ReMLab: a Java-based remote, measurement laboratory*, IEEE Trans. Instrum. Meas., pp. 710-715, 2003.
18. Trullols E., Sorribas J., del Rio J., Samitier C., Manuel A., Palomera R.: *LabVir: A virtual Distributed Measurement System*, Proc. IEEE IMTC/2003, pp. 776-779.
19. Wienicki W., Mielcarz T.: *Internet-based methodology for distributed virtual instrument designing*, Proc. IEEE IMTC/2003 Conf., pp. 760-765, 2003.
20. Panegiani F., Macii D., Carbone P.: *Open Distributed Control and Measurement System Based on an Abstract Client-Server Architecture*, IEEE Trans. Instrum. Meas., pp. 686-692, 2003.
21. Ewald H., Page G. F.: *Client-server and Gateway-Systems for Remote Control,* Proc. IEEE IMTC/2003, pp. 1427-1430.

22. Pyszlak P., Remigiusz R., Rak J., Majkowski A.: *The next approach to the design of a web-based virtual laboratory*, Proc. IEEE IMTC/2003, pp. 1083-1086.
23. Cristaldi L., Deshmukh A., Lazzaroni M., Monti A., Ponci F., Ottoboni R.: *A Monitoring System Based on a Multi-Agent Platform*, Proc. IEEE conf. IMTC/2004, pp. 907-911.
24. Cicirelli F., Furfaro A., Grimaldi D., Nigro L., Pupo F.: *Management Architecture for Distributed Measurement Services*, Proc. IEEE IMTC/2004, pp. 974-979.
25. Andò B., Giannone P., Pecora E., Pitrone N.: *Development in Web-based Laboratory Sessions*, Proc. IEEE IMTC/2004, pp. 1155-1158.
26. Chen S.: *Open Design of Networked Power Quality Monitoring Systems*, IEEE Trans. Instrum. Meas., pp. 597-601, 2004.
27. Paolone M., Peretto L., Sasdelli R., Tinarelli R., Bernardi M., Nucci C.A.: *On the use of data from distributed measurement systems for correlating voltage transients to lightning*, IEEE Trans. Instrum. Meas., vol. 53, no. 4, pp. 1202 - 1208, 2004.
28. Yingqin Y., Kam M.: *Distributed decision fusion with a random-access channel for sensor network applications*, IEEE Trans. Instrum. Meas., vol. 53, no. 4, pp. 1339-1344, 2004.
29. Cristaldi L., Ferrero A., Monti A., Ponci F., McKay W., Dougal R.: *A virtual environment for remote testing of complex systems*, IEEE Trans. Instrum. Meas., vol. 54, no. 1, pp. 123-133, 2005.
30. Benetazzo L., Bertocco M., Narduzzi C.: *Networking Automatic Test Equipment Environments*, IEEE Instrum and Meas Magazine, vol.8, no.1, pp. 16-21, 2005.
31. Schmalzel J., Figueroa F., Morris J., Mandayam S., Polikar R.: *An architecture for intelligent systems based on smart sensors*, IEEE Trans. Instrum. Meas., vol. 54, no. 4, pp. 1612-1616, 2005.
32. Judd R.: *Expanding LXI class C implementations into low cost applications beyond the "standard" test instrumentation functions*, Proc. IEEE Autotestcon/2005, pp. 68-71.
33. Franklin P., Creque A., Reddy R.: *Exploring LXI's advanced capabilities*, Proc. IEEE Autotestcon/2005, pp. 61-67.
34. Wheelwright L.: *The LXI IVI programming model for synchronization and triggering*. Proc. IEEE Autotestcon/2005, pp. 49-55.
35. Pleasant D.: *Lan-based measurement triggering using LXI instrumentation*, Proc. IEEE Autotestcon 2005, pp. 44-48.
36. Caesar G.: *Integrating PXI with VXI, USB, and LXI instrumentation*, Proc. IEEE Autotestcon 2005, pp. 856-860.
37. Ryland J.: *Can LXI replace GPIB?*, Proc. IEEE Autotestcon 2005, pp. 738-742.
38. Carullo A.: *Metrological Management of Large-Scale Measuring Systems*, IEEE Trans. Instrum. Meas., vol. 55, no. 2, 2006, pp. 471-476.
39. Angrisani L., Battaglia L., Schiano R., Moriello L.: *Design and Implementation of a GRID Service for Power Measurement in Digital Wireless Communication Systems*, Proc. IEEE Instr. Meas. Tech. Conf., pp. 2300-2305, 2006.

websites

40. *IEEE-1451 website at NIST*, [Online] Available: http://ieee1451.nist.gov
41. *IEEE Open1451 project*, [Online]. Available: http://open1451.sourceforge.net/
42. *Standard for a smart transducer interface for sensors and actuators - a high-speed CANopen-based transducer network interface*, IEEE 1451.6, proposed standard, [Online] Available: http://grouper.ieee.org/groups/1451/6
43. *Interchangeable Virtual Instrumentation (IVI) consortium documentation*, [Online] Available: http://www.ivifoundation.org.
44. *Standard commands for programmable instruments*, SCPI Consortium, 1997. [Online] Available: http://www.scpi.org.
45. *LAN eXtentions for Instrumentation website*, [Online] Available: http://www.lxistandard.org.
46. *LabVIEW website at National Instruments*, [Online] Available: http://www.ni.com/labview/.
47. *VEE documentation at Agilent Technologies website*, [Online] Available: http://www.agilent.com. (note: use the search box with the keyword "VEE").
48. *VXI Plug & play alliance website*, [Online] Available: http://www.vxipnp.org.
49. *The Common Object Request Broker: Architecture and Specification*, (1998). [Online] Available: http://www.omg.org.
50. *Jini community website*, [Online] Available: http://www.jini.org.
51. *OpenGIS, "Observations and measurements recommendation"*, [Online] Available: http://www.opengeospatial.org/specs/?page=recommendation.
52. *The Open Geospatial Consortium*, [Online] Available: http://www.opengeospatial.org
53. *The globus alliance website*, [Online] Available: http://www.globus.org