RINGrid contract no. 031891

Evaluation and Requirements for Infrastructures

Deliverable 3.3

Summary of Requirements and Needs to be Currently Fulfilled to Efficiently Introduce the Remote Instrumentation Idea into Practice

Document Filename:	RINGRID-WP3-D3_3-JKU-Summary.odt
Work package:	WP3
Partner(s):	JKU, PSNC, GRNET, TUI, REUNA, RNP, CNIT, CNR/ISTI, UNIS
Lead Partner:	JKU
Document classification:	Public

Abstract

This deliverable marks the end of a series of three deliverables. The three deliverables showed the reader what the state of the art in remote instrumentation middleware and networks currently is. While the second deliverable identified applications which benefit from remote instrumentation, this deliverable points out what prerequisites must be met in order to successfully convert these applications into remote instrumentation applications.

Therefore, this deliverable serves as a basis for ongoing work regarding drafting a remote instrumentation architecture (WP6) as well for standardization efforts carried out by the RISGE (Remote Instrumentation Services in Grid Environments) research group which is currently established under the patronage of the OGF (Open Grid Forum).

Delivery Slip

	Name	Partner	Date	Signature
From	Thomas Prokosch	GUP		
Verified by				
Approved by				

Document Log

Version	Date	Summary of changes	Authors
1	21-Jun-2007	Initial draft.	Thomas Prokosch
2	5-Jul-2007	Added two chapter drafts.	Damian Kaliszan, Costas Kotsokalis, Thomas Prokosch, Tasos Zafeiropoulos
3	30-Jul-2007	Most chapters finished.	Davide Adami, Damian Kaliszan, Costas Kotsokalis, Ana Lucia de Moura, Thomas Prokosch, Michael Stanton, Tasos Zafeiropoulos
4	5-Aug-2007	Final version.	Davide Adami, Antonio-Blasco Bonito, Marius Branzila, Luca Caviglione, Romeo Ciobanu, Davide Dardari, Franco Davoli, Marcio Faerman, Alberto Gotta, Damian Kaliszan, Costas Kotsokalis, Angelos Lenis, Fabio Marchesi, Thomas Prokosch, Tomasz Rajtar, Michael Stanton, Zhili Sun, Jin Wu, Tasos Zafeiropoulos, Yanbo Zhou

Table of Contents

Illustration Index

1. Introduction

This deliverable is rather short — this is on purpose. The idea of this deliverable is to give the reader an idea what criteria must be met if an instrument needs to be plugged into the grid in order to use it remotely.

In the second of three deliverables we have seen that instruments can be divided into several groups:

- measurement, control and automation,
- large-scale physics and astronomy,
- sensor networks, as well as instruments similar to
- NMR spectrometers.

As it was layed out, instruments can be divided into these groups because these application domains have certain characteristics which are unique no matter what specific instrument is used. This can be seen in the following table:

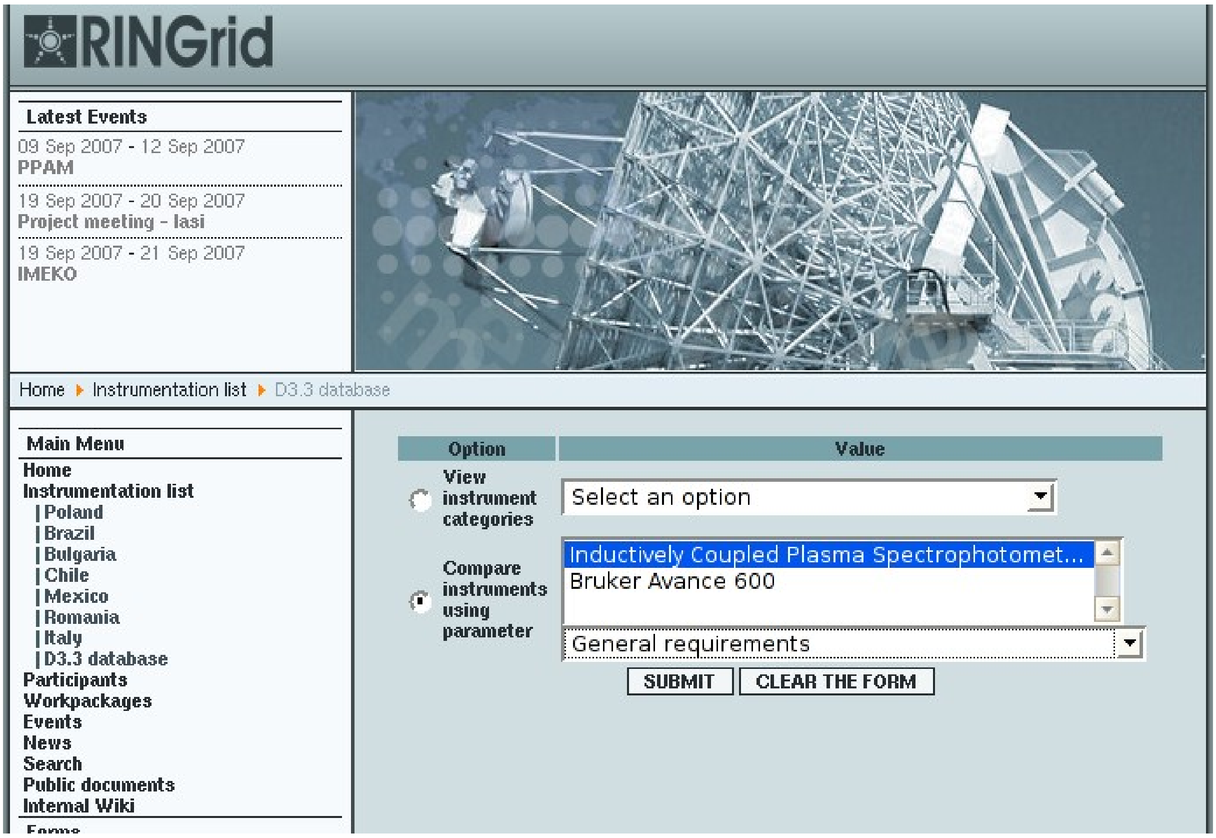
	measurement, control and automation	**large-scale physics and astronomy**	**sensor networks**	**NMR spectroscopy**
processing power	some ("smart")	high	moderate (RISC microprocessors)	high
operated manually/ automatically	manually	manually	automatically (ad-hoc multihop network)	manually
cost per element	small-medium	expensive	low-moderate	expensive
number of data sources	relatively high	singular	large	singular
operation mode	near real-time to real-time	real-time	normal (not critical)	normal (not critical)
data rate	low-moderate	high	low-moderate	moderate
data value	relatively low	very high	low	high

Table 1.1: Chosen applications and selection criteria.

The characteristics, which have been used for the definition of the application domains, can be seen on the leftmost column.

Concluding from the instruments' capabilities and requirements, this deliverable tries to give concrete numbers for each of the application domains in terms of required storage space, bandwith, interactivity and the expected collaboration between scientists, who want to cooperate on a single experiment. The facts gathered can be also seen on web by navigating through a database [1].

Figure shows how to access the database:



Latest Events

09 Sep 2007 - 12 Sep 2007
PPAM

19 Sep 2007 - 20 Sep 2007
Project meeting - Iasi

19 Sep 2007 - 21 Sep 2007
IMEKO

Option	Value
○ View instrument categories	Select an option
● Compare instruments using parameter	Inductively Coupled Plasma Spectrophotomet... Bruker Avance 600 General requirements

Figure 1.2: Accessing the database.

However, providing concrete numbers is not the only goal of this deliverable. To ease deployment of remote instrumentation infrastructures, this deliverable is a cookbook, a collection of recipes (ie. good practices), one should follow during the deployment phase. We have discovered those good practices while working with the instruments.

Now, the application domains are presented in the same order as previously outlined.

1.1. References

[1] RINGrid Remote Instrumentation Database, http://www.ringrid.eu/index.php?option=com_content&task=view&id=122&Itemid=154

2. Measurement, Control & Automation

Measurement, control and automation (MC&A) products and services are essential to the proper operation and profitability of the primary materials and energy producing industries. Specific benefits include improved plant throughput and productivity, enhanced worker safety, increased energy efficiency, higher process yields and waste product minimization, improved product consistency and uniformity. MC&A products and services play a key role in elevating worldwide living standards by increasing the availability and reducing the cost of basic materials and energy while insuring adherence to sound environmental practices.

MC&A was traditionally a product-focused business but, since the emergence of microcomputer based distributed controls, this industry has experienced the growth of a vibrant services sector. Some different kinds of MC&A instruments are referred below:

- measuring instruments (temperature, pressure, flow, level and other parameters)
- chemical analysis instruments
- recording and display instruments
- controllers and control systems
- supervisory and communications systems
- process control software products
- testing and maintenance instruments
- spare parts and supplies

2.1. Example Instrument

In this section an instrument from the “chemical analysis” category is selected to be analyzed regarding its infrastructure and service requirements. This instrument is the Inductively Coupled Plasma Spectrophotometer (ICP). The Inductively Coupled Plasma Spectrophotometer (figure 2.1) analyzes and obtains the concentration of 73 chemical elements in all types of sample.

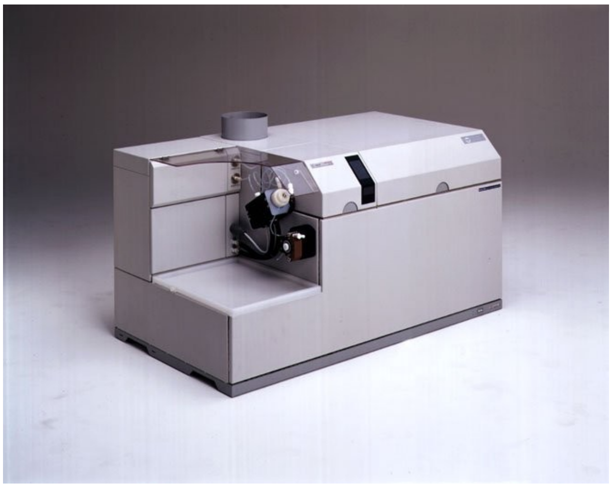


Figure 2.1: Inductively Coupled Plasma Spectrophotometer.

An ICP typically includes the following components:

- sample introduction system (nebulizer and spray chamber)
- plasma torch
- radio frequency generator
- digital interface

Sample introduction system. The sample-introduction area has been called the Achilles heel of ICP-MS because it is considered the weakest component of the instrument, with only 1–2% of the sample finding its way into the plasma. The mechanism of introducing a liquid sample into analytical plasma can be considered as two separate events — aerosol generation using a nebulizer and droplet selection by way of a spray chamber. The main function of the sample introduction system is to generate a fine aerosol of the sample. It achieves this purpose with a nebulizer and a spray chamber. Because the plasma discharge is inefficient at dissociating large droplets, the spray chamber’s function is primarily to allow only the small droplets to enter the plasma. Its secondary purpose is to smooth out pulses that occur during the nebulization process, due mainly to the peristaltic pump.

Plasma torch. One of the basic components that are used to generate the source (along with the radio frequency (RF) coil, and RF power supply). The plasma torch consists of three concentric tubes, which are usually made from quartz.

Radio frequency (RF) generator. Two frequencies have typically been used for ICP RF generators: 27 and 40 MHz. These frequencies have been set aside specifically for RF applications of this kind, so they will not interfere with other communication-based frequencies. The early RF generators used 27 MHz, while the more recent designs favor 40 MHz. The 40 MHz design typically runs at lower power levels, which produces lower signal intensity and reduced background levels. Free-running RF generators, in which the matching network was based on electronic tuning of small changes in frequency brought about by the sample solvent or matrix components. The major benefit of this approach was that compensation for impedance changes was virtually instantaneous because there were no moving parts.

Digital interface. The interface region is probably the most critical area of the whole inductively coupled plasma mass spectrometry (ICP-MS) system. Although we take all the benefits of ICP-MS for granted, the process of taking a liquid sample, generating an aerosol that is suitable for ionization in the plasma, and then sampling a representative number of analyte ions, transporting them through the interface, focusing them via the ion optics into the mass spectrometer, finally ending up with detection and conversion to an electronic signal, are not trivial tasks. Each part of the journey has its own unique problems to overcome but probably the most challenging is the movement of the ions from the plasma to the mass spectrometer.

Elemental concentrations can be measured to ppb (parts per billion) levels using the Inductively Coupled Plasma Spectrophotometer (ICP). The ICP spectrophotometer utilizes plasma to excite elemental electrons which produce photons that are unique to each element. An example of this procedure is the one used for a whole rock analysis. In this procedure a lithium meta-borate flux is generally used to digest the specimen. Once digested, the solution is introduced to the plasma allowing elemental concentration comparisons to known concentration curves. Using stoichiometric techniques elemental concentrations can be converted into molecular weight percentages. An inductively coupled plasma source atomizes and excites even the most refractory elements with high efficiency. With this ICP, several elements can be determined simultaneously without the need for repeated aspirations, adjustment of instrument parameters and tracking of the samples.

The ICP allows determination of elements with atomic mass ranges from 7 to 250. This encompasses from Li to U. Some masses are prohibited such as 40 due to the abundance of argon in the sample. A typical ICP will be able to detect in the region of nanograms per liter to 10 or 100 milligrams per liter or around 8 orders of magnitude of concentration units. Unlike atomic absorption spectroscopy, which can only measure a single element at a time ICP has the capability to scan for all elements simultaneously. This allows rapid sample processing.

2.2. Use Case Summary and Background

ICP-MS can be used for analysis of environmental samples such as water and various other non-particulate samples. The instrument can also determine metals in urine to check for exposure to toxic metals. The instrument is very sensitive to particulate matter and high concentrations of organics will cause the instrument to cease function, requiring cleaning.

In teaching, the ICP can be used for 1) the analysis of inorganic mixtures, 2) the characterization and analysis of solid samples and 3) the analysis of samples of ground, water and air. In the industry, ICP can be used for 1) environmental heavy suspended particle solid analyses, 2) alloy characterization and analysis, 3) kinetic of solid reaction and 4) determination of heavy water and air metal concentration.

The current groups of users of this instrument are investigators, professors, academics and the chemical industry. In the next paragraph we refer to some restrictions that are posed from this instrument.

As far as the instrument operation is concerned, access to the equipment requires the user to have knowledge of the efficient use and the risks that its erroneous use implies. In addition, there are some restrictions regarding the access to the observed data. For the data generated previously by the owner of the equipment or the involved workgroup, the access will be defined from the responsible person from the project that the data were generated. There is also a limit, established by the grid infrastructure, for the data generated by experiments of the user. The minimum period before placing observed data in public domain is three working days.

Before running the experiments, some parameter setting procedures must be followed. The equipment has a previous calibration with which the user does not need to interact. Nevertheless, if the experiment requires a reference parameter, it must be executed previously to superpose the resulting graphs and to obtain the required data. The samples need a previous modification before being inserted to the instrument. Furthermore, the materials cannot be placed directly. The material can be prepared from its origin in the remote user or at the immediate moment previous to the experiment. This depends on the lapsing of preparation means.

2.3. Infrastructure and Service Requirements

2.3.1. Storage Space

The instrument needs 512Mbytes of memory, 100 Flops for processing and 10Gbytes for storage. The results of an observation are always stored in digital format. The entire procedure is very sensitive and if some data are lost or corrupted, the experiment must be repeated.

2.3.2. Data Transmission

The equipment is connected to a computer on the network through a Fast Ethernet interface. The minimum transmission rate of the data sent to the remote users is 1Mbps. Real time access to the measurements is not permitted, but the results can be obtained and processed offline. Online information as an input to the observation process can also be provided from remote databases and digital libraries.

Standard LIMS connection. The need to accurately track large numbers of samples from sample receipt through final reporting, while maintaining strict quality control and adhering to the requirements of Good Automated Laboratory Practices (GALP) led to the integration of the ICP-MS instrument into the Laboratory Information Management System (LIMS). This integration eliminated the need for redundant sample-related data entry, with the associated possibility for error, and also maintained an electronic chain of custody for each sample, from receipt through to final report. This connection allows the client to access Laboratory Information Management System information from other departments in the lab, removing dual entry of data. The standard LIMS connection accesses information such as client sample IDs, matrices, date collected, date received, initial wt/vol, final volume, percent solids, and client name.

The ICPMS instrument computer must be capable of bi-directional communication with the LIMS computer. Most commonly, this is accomplished by networking the ICPMS computer with the LIMS computer through either a local area network (LAN) or a wide area network (WAN). Serial communication via a modem and telephone is also feasible, though typically much slower and less robust. A fully reporting package is supported, that provides results for all quality control type samples, as well as, prep and run logs. In addition, included on all result forms are reporting limits, acceptance windows, and date and time analyzed.

In order for information to flow through the ICP-MS, the software must import sample and batch information directly from the LIMS, correctly analyze the associated samples and supply the results to the LIMS in a format which is easily processed by the LIMS. A mechanism for data review is required, by the ICP-MS analyst to ensure that only data of acceptable quality is reported. In figure 2.2 is shown a typical client-based network configuration.

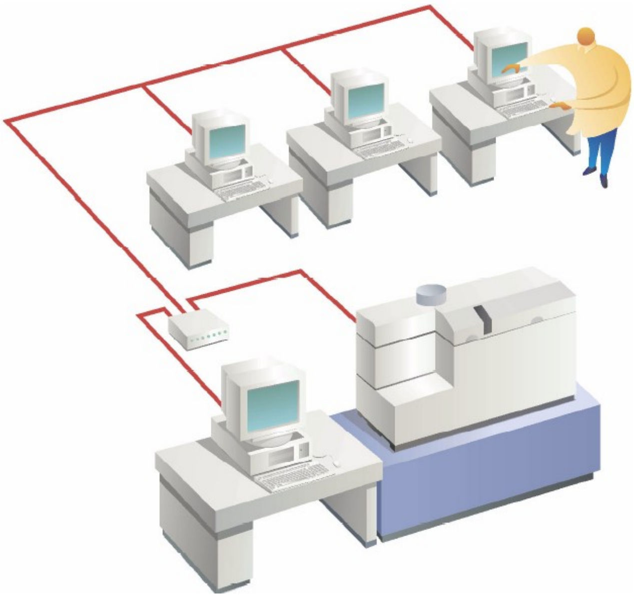


Figure 2.2: Client based network configuration using the ICP.

2.3.3. Interactivity

Interactivity via communication tools between the administrator of the system and the scientific community is possible. A service engineer or technical specialist can remotely access the instrument. This can help eliminate costly onsite service and support calls, by allowing support personnel to remotely monitor the instrument and diagnose the problem. Many times, service and technical support calls can be avoided by making adjustments via the remote connection. If a service call is necessary, it is ensured that the service engineer will arrive with the suitable parts to repair the instrument.

A remote user can also indirectly control the remote instrument (communicating by telephone, videoconference or interactive online software i.e. LabView, VNC to an operator at the instrument site, who locally operates the instrument). Visualization rendering can be done either online for interactive analysis or offline. In addition, streaming servers can transmit video and audio data from the experiments via multicast. Operational guides are also available that instruct new users on the creation of analytical methods, running samples, and reporting results.

2.3.4. Collaboration Tools

Independently available videoconference tools should be used.

2.4. Deployment Guidelines: Good Practices

The majority of inductively coupled plasma mass spectrometry (ICPMS) applications involve the analysis of liquid samples. Even though spectroscopists adapted the technique over the years to handle solids, it was developed in the early 1980s primarily to analyze solutions. There are many ways of introducing a liquid into an ICP mass spectrometer, but they all basically achieve the same result — they generate a fine aerosol of the sample so it can be efficiently ionized in the plasma discharge.

The Mineral Resources Program (MRP) within the Geologic Discipline currently maintains and operates three ICP-MS facilities in Denver, Colorado to support on-going and future geologic research programs. In addition, the facilities are also available to other qualified government agencies looking for state-of-the-art ICP-MS analyses and method development capabilities.

The Solution ICP-MS laboratory contains a dedicated quadrupole ICP-MS for the analysis of a variety of sample types, including rocks, sediments, soils, sludges, biological samples and waters. Most water samples can be run directly on the ICP-MS without additional sample preparation beyond acidification and filtration. Generally, solid samples must be converted into an aqueous form through the use of an appropriate sample digestion technique. The laboratory can provide a standard digestion method or develop a customized digestion method, depending on the sample submitter's requirements.

The Laser Ablation ICP-MS laboratory contains a dedicated quadrupole ICP-MS and a state-of-the-art ultraviolet laser ablation system. The laser ablation system allows solid samples to be run directly without sample digestion. This technique has many advantages, including the ability to provide spatial information about elemental concentrations (qualitative or quantitative) within a sample. However, the detection capabilities can vary widely depending on the spot size of the laser beam, the sample matrix, and the availability of suitable solid standards for calibration.

The High Resolution ICP-MS laboratory contains a multi-collector (MC) ICP-MS and a high-resolution ICP-MS. The MC-ICP-MS is capable of providing rapid and precise isotope ratio information on samples. The single collector ICP-MS can be used to perform determinations where interferences preclude the use of quadrupole ICP-MS. Both solution sampling and laser ablation sampling are available, providing a wide range of flexibility in sample analysis.

2.5. References

[1] “Instruments and Sensors as Network Services: Making Instruments First Class Members of the Grid”, Randall Bramley, Kenneth Chiu, John C. Huffman, Kia Huffman, and Donald F. McMullen.

[2] “A beginners guide to ICP-MS”, Robert J. Thomas, Spectroscopy 2001.

[3] "New approaches for elemental speciation using plasma mass spectrometry", B'Hymer, Clayton, Judith A. Brisbin, Karen L. Sutton, and Joseph A. Caruso, 2000.

[4] ICP-MS Wikipedia, http://en.wikipedia.org/wiki/ICP-MS

[5] ICP-MS by Jenna Worley and Steve Kvech, http://www.cee.vt.edu/ewr/environmental/teach/smprimer/icpms/icpms.htm

[6] The Measurement, Control and Automation Industry, http://www.measure.org/about_industry.htm

3. Large-Scale Physics and Astronomy

3.1. Example Instrument

The selected instrument to exemplify the requirements of remote instrumentation in the Astronomy domain is the SOAR (SOuthern Observatory for Astrophysical Research) Telescope, a 4.1 m diameter optical telescope with an Altitude-over-Azimuth (Alt-Az) mounting, constructed by a consortium formed by the Brazilian Ministry of Science and Technology and the National Optical Astronomy Observatory, the University of North Carolina and Michigan State University, from the USA. The telescope, which was inaugurated in 2004, is located on Cerro Pachón, in Chile, and was designed to work from the atmospheric cut-off in the blue (320 nm) to the near infrared, to have excellent image quality (0.22 arcseconds), fast slewing and to have up to nine coupled instruments mounted.

3.2. Use Case Summary and Background

SOAR has been planned from the beginning to support remote observation, so that astronomers located arbitrarily far from the observatory may observe with and obtain real time observational data from its instruments, which include those listed below (current or planned), mostly under the control of SDSU-2 controllers, developed at San Diego State University.

Instrument	**Detector Type**	**Controller**
Infrared Side Port (ISP) Imager	1 x 2kx2k HgCdTe	SDSU-2
Optical Imager	2 x 2kx4k CCD	SDSU-2
Goodman Spectrograph	2 x 2kx4k CCD	SDSU-2
Integral Field Unit (IFU) Spectrograph	2 x 2kx4k CCD	SDSU-2
Spartan Imager	4 x 2kx2k HgCdTe	E. Loh

Table 3.1: SOAR instruments.

The SOAR Telescope Control System (TCS) software has been developed using LabView, a data acquisition and instrument control software package, running on a Real Time (RT) Linux platform. Remote interaction is supported by VNC connections and by TCS and remote instrumentation GUIs programmed in LabView. The observational data retrieved are always stored locally at the SOAR site, and can also be transferred to the astronomer's institution, replicated at other sites, and eventually ingested into a Virtual Observatory.

The remote astronomer does not control the telescope itself directly. This always always done through the intervention of a highly skilled technician located at the observatory, who is responsible for pointing the telescope and other potentially hazardous manoeuvres. The interaction between the remote observer and the technician located at the observatory is supported by H.323 video conferencing services. Backup telephone lines are also used.

At the remote site, a typical observation room will be equipped with large display units, displaying both the GUIs for the the TCS and the relevant instruments, and also the images generated at the telescope, as illustrated in the following figures. Observation rooms of this kind have been set up at the participating institutions in Brazil and the USA.

Figure 3.2: Remote observation room at the University of North Carolina.

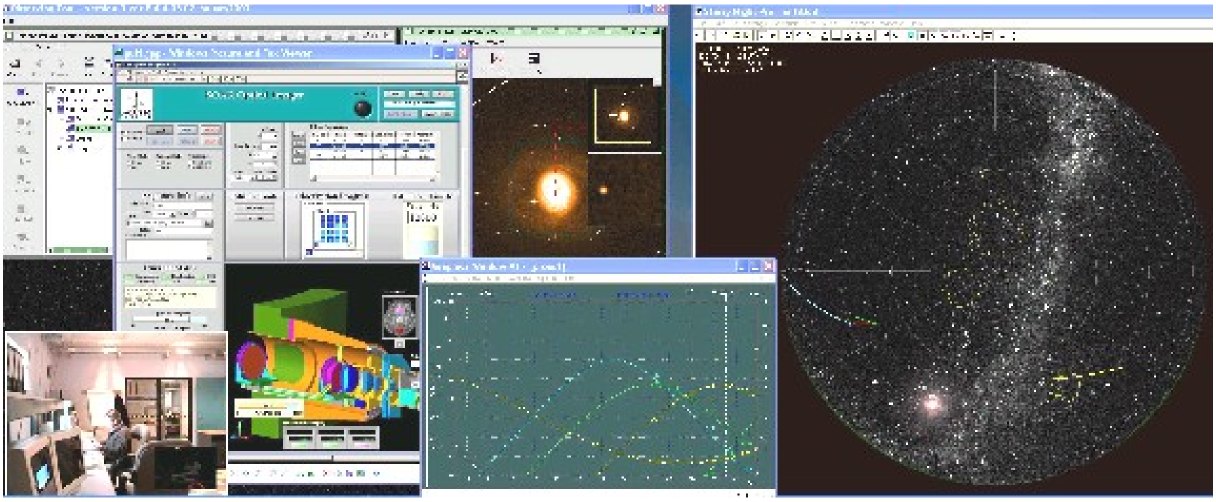


Figure 3.3: Example screen display, including GUIs and remote videoconferencing image.

Normally optical image transmission to the observer uses lossy 500:1 compression to reduce transmission time. Using this image, the observer may then designate regions of interest (ROI) within the image field, which are then transmitted using lossless compression, and patched into the observer’s window. Additionally, the full image is queued for lossless compression and transmission when possible. In 2004, a 3 Mbps link was available for SOAR remote observers, and a typical 16 Mpixel optical image (4 Mbytes) required 2 minutes for transmission.

3.3. Infrastructure and Service Requirements

3.3.1. Storage Space

SOAR optical images require 4 Mbytes for storage, and are generated about every two minutes. Thus, up to 1.2 Gbytes of uncompressed images could be generated during 10 hours of observation. During a full year, the theoretical maximum storage required is about 420 Gbytes.

3.3.2. Data Transmission

Low latency, uncongested links connecting the remote observation room to the observatory site are indispensable. Sufficient bandwidth must be provided to support transferring a potentially large amount of data - currently up to 1.2 Gbyte of uncompressed images, or half this of lossless compressed images - within a time frame limited to the allocated observation session. The need to provide an adequate response time for the remote instrumentation GUI imposes latency constraints. Furthermore, the available bandwidth must be shared with multimedia traffic generated by videoconferencing services.

3.3.3. Interactivity

The interaction between the remote observer and the telescope instruments requires VNC support and remote instrumentation GUIs using LabView. The interaction between the remote observer and the telescope operator located at the observatory requires H.323 videoconferencing services.

3.3.4. Collaboration Tools

Collaborative observation sessions with the participation of astronomers located at different remote observation rooms would require the availability of coordinated access to the interactive facilities listed in the previous section: remote instrumentation GUIs and videoconferencing services.

3.4. Deployment Guidelines: Good Practices

- In order to prevent structural damage to the telescope, remote observation sessions must be supervised by a telescope operator located at the observatory.

- The interaction between the remote observer and the telescope operator should be supported by adequate services, as described in section 3.2.
- Good quality communication channels must be provided to guarantee continuous interaction between the remote observer and the telescope operator.

3.5. References

[1] Thomas A. Sebring, Gerald N. Cecil, Gilberto Moretto, The Soar Telescope Project: A Four-Meter Telescope Focused on Image Quality, Proc SPIE, Vol 3352, p. 62-69 (1998).

[2] Michael C. Ashe, Germán Schumacher, “SOAR Telescope Control System: A Rapid Prototype and Development in LabVIEW”. Proc SPIE, Vol 4009, p. 48-60 (2000).

[3] Gerald Cecil, J. Adam Crain, Germán Schumacher, “Remote Use of the SOAR 4.25m Telescope”, Proc SPIE, Volume 4845, pp. 72-79 (2002).

[4] Michael C. Ashe, Germán Schumacher, Thomas Sebring, “SOAR Control Systems Operation: OCS & TCS”. Proc SPIE, Vol 4848, pp. 294-303 (2002).

[5] Germán Schumacher, Michael C. Ashe, “SOAR TCS: From Prototype to Implementation”. Proc SPIE, Vol 4848, pp. 230-238 (2002).

[6] Michael Ashe, Marco Bonati, Steve Heathcote, “ArcVIEW: a LabVIEW-based astronomical instrument control system”, Proc SPIE, Vol 4848, pp. 508-518 (2002).

[7] Germán Schumacher, Steve Heathcote, Victor Krabbendam, “SOAR TCS: from implementation to operation”, SPIE, Volume 5496, pp. 32-37 (2004).

[8] Gerald Cecil, J. Adam Crain, “Remote Observing with SOAR: Tactics and Early Results”. SPIE, Volume 5496, pp. 73-80 (2004).

[9] SOAR web page, http://www.soartelescope.org

[10] SOAR web page (in Portuguese), http://www.lna.br/soar/soar.html

4. Sensor Networks

A sensor network [SN1] is composed by a large number of singular devices (nodes) employed to produce some kind of data: for example, nodes can be equipped with physical parameter (e.g temperature, pressure, humidity, etc.) sensors to collect measurements on the surrounding environment, can be employed acquire position informations to enforce entities localization or can be used to calculate partial results in a larger distributed computation process.

Therefore, from a more abstract point of view, sensor network can be considered as a macro-entity allowing data collection and, then, be defined, in relation to this document topic, as a special kind of remote instrumentation.

Due to the wide interest expressed recently by the scientific community on the sensor network applications, future grid system, enabling remote instrumentation interaction, will certainly meet the need to support this type of resource through an extension of the common instrumentation interaction interface in order to enable the user to access it in a transparent and standard way, while supporting the specific sensor network features and satisfying the peculiar network and functional requirements introduced in this section.

4.1. Example Instrument: Wireless Sensor Network

A wireless sensor network is created with a huge number of low power, low cost sensors [BAR07] which self organize into a (multi-hop) ad hoc network. Sensors are spread in an environment (sensor field) without any predetermined infrastructure and cooperate to execute common monitoring tasks which usually consist in sensing environmental data. The sensed data are collected by an external sink node, when it is connected to the network. The sink node, which could be either static or mobile, is in turn accessed by the external operators to retrieve the information gathered by the network. An example of a wireless sensor network architecture is shown in figure 4.1.

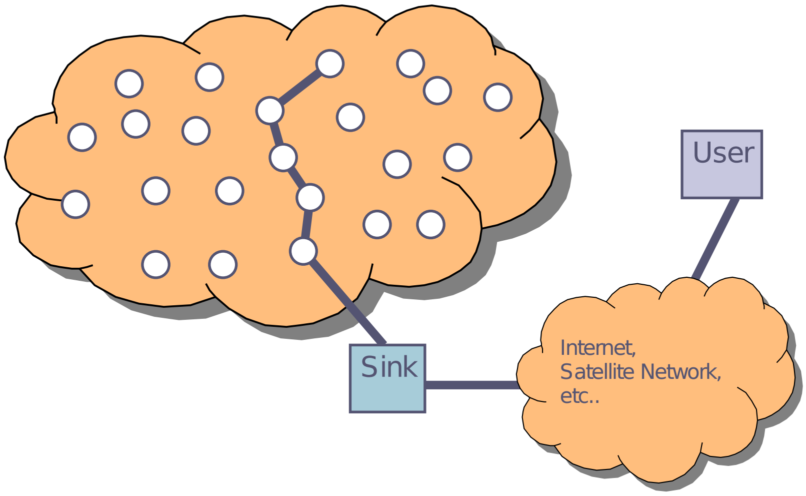


Figure 4.1: An example of a wireless sensor network architecture.

A wireless sensor is characterised by its small size, its ability to sense environmental phenomena through a set of transducers and a radio transceiver with autonomous power supply. Current low-end sensors employ low cost microcontrollers with a small program and data memory size (about 100 KB). An external flash memory with large access times may be added to provide secondary storage and to alleviate the application size constraints imposed by the on-chip memory. Common on-board I/O buses and devices include serial lines such as the Universal Asynchronous Receiver-Transmitter (UART), analog to digital converters and timers.

Two approaches have been adopted for the design of transducer equipment. The most general and expandable approach, as pioneered by Crossbow [CRO], consists in developing transducer boards that can be attached (and possibly stacked one on top of the other) to the main microcontroller board through an expansion bus. The other approach (followed by Moteiv [MOT]) is to put transducers directly on the microcontroller board.

By means of a radio transceiver circuitry a sensor unit communicates with nearby units. Most recent wireless sensors embed a radio compatible with the standard IEEE 802.15.4 [IEEE03], which operates in the 2.4 GHz band and offers a 250 Kbps bandwidth.

Sensors are powered by batteries, usually a couple of standard AA. Battery size usually determines the size of the sensor, so existing hardware is roughly a few cubic centimetres in size. Studies are currently under way to replace/integrate battery sources with some power scavenging methods such as solar cells but there are some reservations about the actual effectiveness of such methods. Solar cells, for instance, do not produce much energy indoor or when covered by tree foliage.

4.2. Use Case Summary and Background

While originally developed for military purposes to enable territory surveillance on battlefields, currently sensor network are being widely employed in ecological or ethological monitoring and environmental phenomenon observation applications exploiting the intrinsic distributed characteristic to cover large or “difficult” area or situations where traditional instrumentation based experiments would prove to be impossible.

In particular some significant deployment projects can be cited as real use case scenarios, highlighting the current trend in sensor network utilization:

Great Duck Island Project [GDI]. This project has enforced the design and implementation of a wireless sensor network architecture aiming to collect data about the habitat in a small island of Maine (USA) in order to study a predictive model of seabirds behaviour. In particular the observations focused on the petrel population measuring several parameters about the individuals and relating them, along the breeding season, with environment and atmospheric conditions. Utilization of sensor network in this context has allowed to reduce the personnel needed to perform observations and, consequently, the disturbance to the observed ecosystem (and to the measured data) unavoidably caused by human presence and by traditional instrumentations. Data acquirement process has been adjusted to follow the dynamic of different parameters ranging from 10 minutes to 4 hours timing: collected data have been organized in a centralized database with the possibility, for user, to access them through internet.

Hawaii environment monitoring [HEM]. This project has studied and realized an architecture to monitor some particular areas of the island to survey the environmental and health conditions of some rare plant species. Utilization of wireless sensor network has proved necessary to overcome the difficulties introduced by the complicated topography of involved sites and the extremely varying environmental features. In particular, the embedded device, equipped with physical parameters sensor and medium resolution camera, have been enclosed in special capsule granting an extended environmental protection, to form a “pod”: these unit, completed with Global Positioning System (GPS) capability, wireless communication gear and power source, have been consequently deployed on the territory to arrange a network of data collecting nodes capable to propagate gathered informations to the a reference base station for the specific geographical zone.

Beside environmental characteristic monitoring topic, other employment of sensor network has been investigated by the scientific community: for example a research [TMSN] has explored the possibility of vehicle traffic monitoring using wireless sensor (then lowering the deployment and maintenance costs), research has been, also, performed to realize a technologies, called LaserSPECks [LS], exploiting the capabilities of wireless sensor network in the gas sensing field with the infrared laser spectroscopy technique to obtain gas trace detection with performances comparable to traditional instrumentation but with much lower costs.

4.3. Infrastructure and Service Requirements

Some requirements should be taken in account while designing and implementing a sensor network as a grid resource to collect data: This section presents an overview of those needs along with a concise introduction of current technologies and techniques commonly adopted to address them.

From the hardware point of view, nodes of a sensor network should be characterized by low power consumption to maximize the lifetime, support for different type of sensor and a reasonable powerful programming capability to implement middleware functionality and/or particular algorithms.

In recent times the great improvement in the miniaturization has lead to the development of specialized technologies for the development of embedded devices with standard I/O interfaces integrated on board and characterized by a very low power consumption profile while still allowing enough computational power to allow the execution of specifically designed operating system (e.g. TinyOS [TOS], Contiki [CTK], etc.).

As reported in deliverable D3.2, some products are currently enough mature for sensor network implementation, like the Moteiv [MOT] or the widely employed Crossbow mica series [CRO] that allow a customizable construction of the node.

Table 4.2 shows features of some of the most used platforms for wireless sensor networks.

	Btnode 3	**mica 2**	**mica 2dot**	**mica z**	**telos A**	**tmote sky**	**EYES**
manufacturer	Art of Technology	Crossbow			Imote iv		Univ. of Twente
microcontroller	Atmel Atmega 128L				Texas Instruments MSP430		
clock frequency	7.37 MHz		4 MHz	7.37 MHz	8 MHz		5 MHz
RAM (KB)	64 + 180	4	4	4	2	10	2
ROM (KB)	128	128	128	128	60	48	60
storage (KB)	4	512	512	512	256	1024	4

	Btnode 3	**mica 2**	**mica 2dot**	**mica z**	**telos A**	**tmote sky**	**EYES**
radio	Chipcon CC1000 315/433/868/916 MHz 38.4 Kbauds			Chipcon CC2420 2.4 GHz 250Kbps IEEE 802.15.4			RFM TR1001 868 MHz 57.6 Kbps
max range (m)	150-300			75-100			
power	2 AA batteries		coin cell	2 AA batteries			
PC connector	by PC-connected programming board				USB		serial port
OS	Nut/OS	TinyOS					PEEROS
transducers	on acquisition board				on board		on acquisition board
extras	+ bluetooth radio						

Table 4.2: Comparison for various sensor architectures.

4.3.1. Storage Space

Datasets produced by sensor networks are heavily dependant on the nature of the SN application itself: The implementation process and the functional characteristics of this resource, in fact, are strictly adapted and tailored to specific activities that has to be accomplished.

Consequently, the process of data management both for collecting data within the sensor network as well as to present it to the user (by means of the grid system), is subject to the meaning of the data: Some deployments are interested only in an aggregated information produced from the various sensors measurements while other applications are focused on collecting every single value to create, for example, a history of the environment according to several physical parameters perspectives.

The Great Duck Island Project project, for example, can be considered as an example of the latter scenario: In this application, the deployed sensor network was composed by 32 nodes each equipped with hardware to produce several different measurements at different time intervals. Then, the results were stored in a database to let scientists create reports highlighting different physical dimensions like temperature, pressure etc., of the involved environment during a certain scope of time.

In a period of three-month period during summer 2002, the sensor network delivered 1.8 million measurement packets to the data storing facility, with individual motes delivering up to 50,000 packets with an approximate size of 25 bytes.

In conclusion, dimensioning of storage space for sensor network should take into account several points related to the nature of the application itself that can be expressed as peculiar properties of informations collected by the SN:

- Level of aggregation,
- meaning of the data collected,
- throughput frequency of measurements, as well as
- level of compression (if sensors are capable of executing compression algorithms before delivering data).

4.3.2. Data Transmission

Sensor network usually does not demand extreme performances in terms of bandwidth: with a balanced timing of the communication activity, nodes would require a reasonably low band capacity for exchanging small sized data like sensing informations and middleware related messages.

In common scenarios, a weak reliability of communication medium could be obviate by the great number of nodes and by an accurate deployment realizing a redundancy of connection through different paths.

Several networking technologies, both wired and wireless, has been introduced for sensor network: in particular wireless technologies (e.g. ZigBee [ZIG05], Bluetooth [BLUE], etc.) have been utilized by recent researches because in most project the deployment has to be accomplished in geographically sparse and often “difficult” outdoor areas. The wireless technologies, presents some open issues regarding support for security of data transmission: typical environment where sensor network are usually employed and the over-the-air nature of connections makes virtually impossible to restrict network access and, therefore, to avoid unauthorized interception of sensible data. For this reason several efforts have been done to integrate strong security mechanisms in new versions of networking technologies to enforce authentication and cryptography directly in protocols (e.g. version 2.1 of Bluetooth technology).

Time coherence. Being data collection/production resources, sensor networks need to enforce both semantic and temporal correctness properties for the exchanged informations in order to provide meaningful results. While considering the semantic correctness property already granted by the precision of sensing equipments integrated on the node and by the transmission control techniques provided by communication components, the temporal correctness property can become a serious design problem due to the need to maintain a coherent time-base along all distributed devices: data acquired by each nodes should be timestamped in a coherent way in order to allow, for example, aggregation of informations produced in particular time intervals, or temporal ordering of data to produce a “history” of measurements.

Ideally a start-up clocks synchronization step would be sufficient, unfortunately, in real world application, oscillators used to implement clocks on node suffer from a variable precision degradation due to frequency drifting phenomenon: sensor networks, then, require nodes to support specialized algorithms to address this issue and grant a reasonable precision in time-marking collected data.

Middleware support. Basing on the planned sensor network application, different types of middleware should be supported in the design and implementation process. Currently, several middlewares are employed in distributed environment project, each one focusing on particular needs of the specific application: for example middlewares provided by SINA [SRIS] or COUGAR [BON00] define a complete query system to access to data collected by nodes, others like MILAN [HEI04], AUTOSEC [AUTS] and DSWARE [EVTD] focus on quality of service (QoS) enforcing policies while still considering energy efficiency, others like IMPALA [IMP] allow to maintain and update application code on single node through a code migration mechanism optimized for granting an efficient energy usage. Therefore in the deployment of the sensor network, analysis of the application dynamics should constrain the decision about the middleware to support.

4.3.3. Interactivity

Most sensor networks include in their architecture a particular node, usually referred to as base station, that, being equipped with more capable hardware, connectivity and power source, can act as a gateway for the sensor network: This component, in addition to enabling the grid system interfacing, could be designed to offer means to allow the monitoring and configuration of the sensor network, both with the use of special designed GUI or with traditional interaction mechanisms (e.g., web-based interfaces, RPC, Telnet or SSH based sessions, etc.).

4.3.4. Collaboration Tools

Usually, sensor networks do not integrate tools to support cooperative real-time monitoring of the activity: Support for on-line collaboration is more likely to be demanded from external tools or to services integrated into the grid infrastructure. Data, instead, can be managed to enable multi-user accesses without interference in the sensor network specific working database: for example in the cited Great Duck Island Project, a database replication mechanism has been included to produce remote copies of the central database, updated every few minutes, then offering to users the capability to query the data, working on periodical snapshots of the dataset and, then, avoiding probable synchronization problems.

4.4. Deployment Guidelines: Good Practices

In this section various guidelines and approaches for the realization of sensor networks are described which should be used in future remote instrumentation grids.

Approach for deploying sensor network applications. Two classes of applications can be distinguished. One is involved in event detection whereby each sensor periodically checks if some environmental conditions are locally satisfied or match a predefined pattern (e.g., animal sightings). In such applications neighbouring nodes may cooperate to achieve a higher confidence on the event characteristics and pattern matching degree but the event data is stored in the network (for later retrieval) or directly sent to the sink.

The other class is engaged in long running environmental observations that continuously perform sampling and result in data streams. This extremely large amount of data cannot be stored in the network, given the limited memory resources of nodes and must ultimately flow to the sink or be discarded. The need to collect data from many highly distributed nodes must be balanced with the high cost of communication. A simple way to reduce messages is to act at the network layer and combine several messages bound for the sink into one big message. This solution only alleviates problems since messages can only grow up to a maximum (usually small) size in a sensor network. Data aggregation and in-network data processing is a more promising approach that consists in moving computing activities from the PC into the network [MAD02a], [MAD02b]. Instead of just forwarding data toward the sink, nodes perform computation and data-management tasks so that user requested data is not extracted from raw data on the PC but is directly furnished by the network. Nodes can do some processing on a data stream (like taking temporal averages or computing functions) or combine it with other data streams (like joining or taking spatial averages) and ultimately produce another data stream which they forward to another node.

The above described approaches deal essentially with the issues related to the programming of the sensor network and with the data gathering tasks. However there are also issues in the management of the sensor network which are in some sense orthogonal and which are related to the mechanisms necessary to dynamically discover the presence of sensors or of the network and discover the services offered by the network. These mechanisms are generally accompanied by profiling mechanisms, that is mechanisms which enable the user to learn how to use the sensor network. In other words the sensors or the sensor network advertise its presence and the offered services using codes which corresponds to profiles. The user may use these codes to access a profile database containing the specifications (possibily written in formal language) of the services.

To distinguish between these two types of interaction we classify as low-level interaction models the models which deal with dynamic network programming and data gathering, and we classify as high-level interaction models the models dealing with device and service discovery. The low-level interaction models can be divided into

- directed diffusion, and
- the database approach.

There are a number of approaches dealing with high-level interaction. We mention among others the Mires [SOU04], TinyLime [GIA05], and ZigBee [ZIG05]. It should be observed that ZigBee deserves particular attention since it is a recently introduced standard for wireless sensor network. For this reason in the rest of the section we briefly summarize its main features as a matter of example of this whole class of approaches.

Directed Diffusion [INT00] is an early attempt to define a data management paradigm in sensor networks. A user request for specific data is translated into an interest for some kind of data with a certain data rate. Interest dissemination begins with the sink broadcasting the interest message to its neighbours. Before forwarding the message each node records the interest and data rate in its cache and sets up a gradient toward the source of the message. This way the interest propagates throughout the network.

Nodes that detect or receive data matching one of their cached interests forward such data along gradients with the associated data rates. Via neighbour-to-neighbour propagation, data finally reaches the sink. The sink can reinforce paths by sending a new interest message with a higher data rate through selected paths. Nodes on the path that are not reinforced ultimately clear their cached interest upon timer expiration. Nodes choose to reinforce a neighbour on the basis of higher quality/rate of received data. Reinforcement can also be triggered by non sink nodes when they detect reduced quality data from existing paths.

Chief advantage of Directed Diffusion is that data exchange is exclusively based on locally exchanged interests. There are no explicit end-to-end multihop paths and no need for routing and network-wide addresses. Multipath data delivery (via reinforcing multiple paths) and local data path repair (via node-triggered reinforcing) are also available. A disadvantage is load unbalance since nodes close to the sink have to manage a large part of control and data traffic. Another problem is limited possibility for in-network data processing and aggregation since different data can be combined only if they are routed through a common node.

The Database Approach. An interesting approach that recently gained in popularity and offers powerful, application-independent, data abstraction and manipulation functionalities is to view the sensor network as a distributed database system. The user formulates data requests via an SQL-like query language that includes syntax to specify sampling rates as well as query duration [MAD02a], [MAD02b]. The high level query is translated into a set of data acquisition (sampling), data processing and data transfer operations that must be carried out by the nodes in the network. Query optimization then evaluates several task allocation alternatives (query execution plans) and chooses one that minimizes energy consumption. Figure 4.3 shows how an SQL like query can be translated into a relational algebra expression that is later optimized to produce a query execution plan which is finally converted into commands to inject in the sensor network.

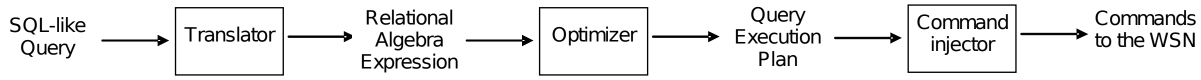


Figure 4.3: Translation of an SQL like query into WSN commands.

The selected query execution plan is then injected into the network as a series of commands. A node can be instructed to join two data streams, implement filtering operations selecting records on the basis of some predicate or compute functions depending on record contents. Other forms of in-network aggregation include taking temporal and spatial averages of transducer readings. While the former can take place on the sensing node, the latter requires collecting readings from several nodes using a tree built over the area where the average must be taken and can be done on-the-fly as data moves along the tree edges. A similar technique can be applied to other aggregate operators like Min, Max, Count and Sum. Reducing message exchange also demands that data aggregation be applied as close as possible to data sources (transducers). As a result of distributed in network query execution only the query outcome reaches the sink.

Query execution should also be tolerant to node failures: task assignment should not be rigid and immutable but mechanisms should guarantee automatic recovery. [YAO03] suggests that constructing a query execution plan should amount to linking together several flow blocks. Each flow block has a certain data collection task involving a set of geographically close nodes (e.g., taking a spatial average). A leader is elected among these nodes, and data is collected and routed towards the leader with aggregation and computation performed along the path and possibly at the leader itself. The leader periodically notifies the other nodes that it is still alive to prevent automatic reconfiguration of the flow block internal organization. Query optimization should consider flow blocks as basic, locally autonomous building blocks.

TinyDB [MAD03] is a sensor network database implementation developed at UC Berkeley. An SQL-like language with extensions for query duration and sample rates is used to express queries over a single sensors table that represents all sampled data in the network (with one row for each sensor being continuously updated). TinyDB supports spatial aggregation operators as described in [MAD02b], filtering based on predicates and special joins taken over the sensors relation and a storage point or two storage points (a storage point is a bounded subset of a stream i.e., a limited number of records).

Power-aware optimization and query execution plan generation is performed on the basis of meta data concerning transducers and operator parameters and it results in a suitable ordering of sampling activities and predicate-based selection. Query dissemination is achieved via Semantic Routing Trees (SRTs): routing trees built from the sink. During the tree construction process each node gathers range information regarding the values of some attribute covered by each of its subtrees. A query later propagates down the various paths in the SRT as long as there are interested nodes.A major limitation of TinyDB is that data streams flow towards the sink along the edges of the routing tree: queries involving more complex data communication patterns are not allowed.

Cougar [BON00, BON01, YAO02] is a sensor network database developed at Cornell University and shares many similarities with TinyDB. The user expresses a query in a high level declarative language that extends SQL. Nodes are modelled as Abstract Data Types (ADTs) with interface functions providing access to encapsulated data. The FROM clause of a Cougar query may refer to a sensor network relation, say R, including attributes identifying a node position as well as the node ADT, say s, while SELECT and WHERE clauses may refer to actual node specific data invoking access methods on node ADTs like R:s:getTemp(). A query optimizer running on a PC generates a query execution plan that specifies data flow and computation activities to carry out at each node, including organization of aggregation trees. From an implementation point of view a virtual relation is associated with each method available for the node ADT. The virtual relation for a method includes attributes for the node id, input arguments, output value(s) and timestamp. A virtual relation is fragmented over all nodes that produce records for it (i.e., implement the associated method) and is stored distributively in the network.

Mad-WiSe [AMA05b, AMA06a, AMA06b] implements a distributed database system that supports in-network query processing. Similarly to the previous approaches, it parses an SQL-like query and selects one of several query plans for execution. Query optimization is carried out by applying several transformation rules based on heuristics to considered query execution plans. These rules take into account transducer sampling costs, predicate selectivity and transmission costs. Query processing is based on streams that abstract data channels between operators of a query algebra and drive their pipelined behaviour (computation and aggregation is carried out on flowing records with almost no need of storage). Operators include selections, projections, spatial and temporal aggregates as well as unions and joins. The ability to perform joins between streams is unique to MaD-WiSe and permits comparison of data from different sources to be carried out in the network.

The ZigBee Approach. The ZigBee standard [ZIG05] builds upon the IEEE 802.15.4 standard. It specifies the network and the application layers of a wireless sensor network. The network layer (NWK) is in charge of organizing and providing routing over a multihop network (built on top of the IEEE 802.15.4 functionalities), while the Application Layer

(APL) intends to provide a framework for distributed application development and communication. The APL comprises the Application Framework, the ZigBee Device Objects (ZDO), and the Application Sub Layer (APS). The Application Framework can have up to 240 Application Objects, that is, user defined application modules which are part of a Zigbee application. The ZDO provides services that allow the APOs to discover each other and to organize into a distributed application. The APS offers an interface to data and security services to the APOs and ZDO. An overview of the ZigBee protocol stack is shown in figure 4.4.

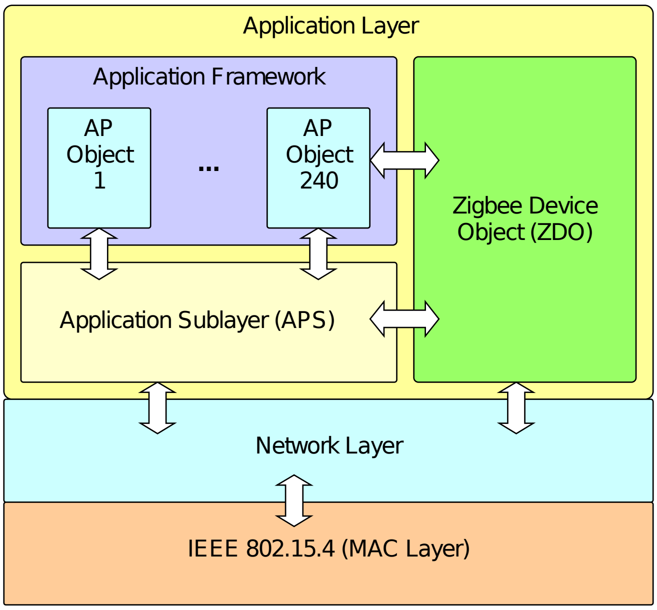


Figure 4.4: ZigBee functional layer architecture.

A ZigBee application consists of a set of Application Objects (APOs) spread over several nodes in the network. An APO is a piece of software (from an application developer) that controls a hardware unit (transducer, switch, lamp) available on the device. Each APO is assigned a locally unique endpoint number that other APOs can use as an extension to the network device address to interact with it. The ZigBee Device Object (ZDO) is a special object which offers services to the APOs: it allows them to discover devices in the network and the service they implement. It also provides communication, network and security management services. The Application Sublayer (APS) provides data transfer services for the APOs and the ZDO.

A ZigBee application must conform to an existing (ZigBee Alliance-accepted) application profile. An application profile defines message formats and protocols for interactions between APOs that collectively form a distributed application. The application profile framework allows different developers to independently build and sell ZigBee devices that can interoperate with each other in a given application profile. Each APO encapsulates a set of attributes (data entities representing internal state, etc.) and provides functionalities (services) for setting/retrieving values of these attributes or being notified when an attribute value changes. In the context of a profile, a group of related attributes is termed a "cluster" and identified with a numeric id. Typically a cluster represents a sort of interface (or part of it) of the APO to the other APOs.

The application profile must specify one of two possible communication service types. For the Key Value Pair (KVP) service type the ZigBee standard has predefined message layouts which must be suitably filled by APOs to request a given operation on attributes residing on a remote APO. The interactions between APOs is limited by the operations supported on attributes. The Generic Message service type is suitable for applications that do not fit in the KVP service type and leaves responsibility to the application profile for specifying message types and their contents.

A special application profile, named the Device Profile, must be implemented by all nodes in a ZigBee network. The object responsible for this profile is the ZDO. The Device Profile requires its implementing objects (ZDOs) to support device/service discovery procedures wherein a node attempts to discover existing nodes in the network, active endpoints on some node and/or the services they implement (available cluster ids). In practice service discovery exploits cluster descriptors ad cluster identifiers to determine the services offered by a given APO. It can be accomplished by issuing a query for each endpoint on a given device or by using a match service feature.

Note also that the device and service discovery requests can be conducted based on different input parameters. Typically the device discovery takes in input a 64 bit extended IEEE address of a device and returns its network address and/or the list of the network addresses of its associated devices. Service discovery is more complex and takes in input a network address and optionally a endpoint number, a cluster identifier, a profile identifier, or a device descriptor. The queried device returns the set of endpoints matching the query (for instance the endpoints which implement a given cluster).

Power efficiency management techniques. In case of sensor network with limited not renewable power source, the management of power consumption due to the various activities to be performed, gain crucial importance. This section reports on research on techniques focused on efficient usage of energy available to sensor network node.

Energy efficient distributed acquisition and pre-processing through WSNs. A scenario where a self-organizing wireless sensor network (WSN) with energy constraints employed for environmental data measurement, is considered in this technique [DAR1].

The sensor devices (denoted as nodes) are randomly distributed; the overall goal of the sensor network is to sample measurements and digitally process them to obtain significant informations on the environment: nodes transmit samples to a supervisor, without energy constraints, by using a clustered network and the supervisor task is to manage communication by waking up a certain number of nodes at times with triggering packets in order to collect samples.

Typical environmental phenomena (e.g., temperature or pressure measurements) are slow time-varying if compared with the packet delivery time in WSNs. For this reason, it can be considered a quasi-static scenario, which means that the round time is considered to be much smaller than the change rate of the observed field. In this scenario no stringent time synchronization constraints among nodes are present. The signal to be sampled is described here through the (target) l-dimensional spatial random process $Z(\mathbf{s})$ ($\mathbf{s}$ being the spatial variable) with realizations $z(\mathbf{s})$. We consider the sample space as a finite region A where the process is observed, centered on the supervisor. Without loss of generality, we consider A a circular area with radius R. Hence, the actual (truncated) signal of interest is $x(\mathbf{s}) = z(\mathbf{s}) \times r_A(\mathbf{s})$.

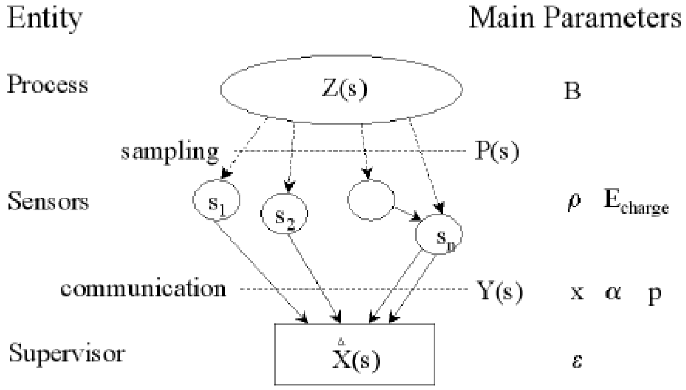


Figure 4.5: Main quantities in the process estimation.

The signal $x(\mathbf{s})$ has finite energy E_0 and belongs to the random process $X(\mathbf{s})$. The goal is to create an estimate of $x(\mathbf{s})$, which we denote by $\hat{X}(\mathbf{s})$. In Figure 3.2.1 a scheme of the whole estimation process is shown.

A good indicator of the estimate quality is the average normalized estimation error defined as the normalized mean square error (MSE):

$$\varepsilon = \frac{1}{E_0} E\left\{ \int_{\Re^l} \left(\hat{X}(\mathbf{s}) - x(\mathbf{s}) \right)^2 d\mathbf{s} \right\}.$$

When a specialized distributed digital signal processing (DDSP) is not adopted, all samples successfully received are processed by the supervisor, in order to determine the estimate $\hat{X}(\mathbf{s})$. Each sample has a probability equal to p to be missing because of an unconnected node, or owing to MAC failures.

In order to reduce the overall energy consumption due to the transmission of samples, then, the signal processing task necessary to have an estimate of the target process, should be partially decentralized. In particular, considering a clustered network architecture, samples coming from nodes are not directly collected by the supervisor but they reach the final destination through intermediate nodes (the cluster heads, CH), which perform partial signal processing. At each CH, loss-less data compression techniques can be adopted thus reducing the amount of data transmitted. Typical compression techniques take advantage of the correlation among adjacent samples.

The efficient trade-off between energy conservation (i.e., network lifetime) and estimation error is obtainable by adopting this technique and numerical results show that both the DDSP technique and the MAC protocol choice have a relevant impact on the performance of a WSN. In particular, the DDSP technique proposed provides relevant advantages in terms of energy efficiency at the expense of an increased estimation error; the role of MAC protocol can be very significant and its choice affects overall performance; a saturation effect on the performance when node density is increased is present due to the clustering architecture and MAC.

Energy efficient distributed detection through WSNs. In most applications, the intelligent fusion of information from geographically dispersed sensor nodes, commonly known as distributed data fusion, is an important issue. A related problem is the decentralized (or distributed) detection problem [DAR2], where a network of sensors, together with a global detector (or fusion center), cooperatively undertake the task of identifying the presence or absence of a phenomenon of interest (PoI).

The classical binary decentralized detection system of a WSN with a large number of identical sensor nodes deployed 00randomly over a wide region communicating directly with the fusion center has been considered. The goal is to detect or monitor a PoI in the sensor field using these geographically dispersed nodes. Specifically, each node takes a local decision about the presence of absence of the PoI and sends its decision to the fusion center which is responsible for the final decision based on the information gathered from local sensors. Two problems have to be considered: the design of the decision rule at the fusion center and the design of the local sensor signal processing strategies. In case of perfect knowledge of system parameters the design of the decision rule at the fusion center is a well established task. The design of the local sensor decision rule, i.e., the likelihood ratio test (LRT) threshold in binary detection, instead is more challenging due to the distributed nature of the system. In fact, the optimal choice of each sensor LRT threshold is coupled to each other node threshold whereas nodes are not in general fully connected or communication among nodes could be too energy demanding. Recently it has been demonstrated that under the asymptotic regime (i.e., large number of nodes), the identical LRT threshold rule at the sensors provides the optimal error exponent if local sensor observations are independent and identically distributed.

Unlike in classical decentralized detection problems, greater challenges exist in a WSN setting. There are stringent power constraints for each node, and communication channels from nodes to the fusion center are severely bandwidth-constrained. In addition, the communication channels are no longer lossless (e.g, fading, noise and, possibly, interference are present), and the observation at each sensor node is spatially varying.

Recently, there has been great interest in cooperative communication. One may also exploit diversity associated with spatially distributed users, or simply cooperative diversity, in WSN. In these networks, multiple sensor nodes pool their resources in a distributed manner to enhance the reliability of the transmission link. Specifically, in the context of decentralized detection, cooperation allows sensor nodes to exchange information and to continuously update their local decisions until consensus is reached across the nodes. For example, cooperation in decentralized detection can be accomplished via the use of Parley algorithm. This algorithm has been shown to converge to a global decision after sufficient number of iterations when certain conditions are met. However, without a fully-connected network and given that the sensor observations are spatially varying, Parley algorithm may result in convergence to a wrong decision at most of the nodes.

Considering a scenario where the signal generated by the Poi, the Parley algorithm will likely lead to consensus in the wrong decision since majority of the nodes in the WSN have rejected hypotheses that the Poi is present (defined as H1). To overcome this problem without generating algorithm variants based on a huge number of message exchange, a new consensus algorithm has been introduced that, basing on flooding protocol, adopts a voting scheme to enable agreement in decisions and to control false-alarm flooding. Moreover, only nodes that declare H1 are allowed to broadcast their decisions and, when time constraint is not stringent, parameters like the number of protocol iterations can be chosen large enough to allow the consensus flooding protocol to terminate correctly: at this point active nodes can report their decision on Poi to the fusion center. Results emerging from accomplished test, show that traditional technique using a parallel delivery of measurements from sensors to fusion center maintain better performance only with clearly detectable Poi: in cases where the signal to acquire is weak, the cooperative approach obtain better results with more efficient node energy usage. So in scenarios where the deployment of the sensor network have to deal with uncertainty about sensors or observed phenomenon locations, the proposed approach offers a new and efficient solution.

4.5. References

[IMP] T.Liu, M.Martonosi, “Impala: A middleware system for managing autonomic, parallel sensor systems”, 2003.

[EVTD] S.Li, S.Son, J.Stankovic, “Event detection services using data service middleware in distributed sensor networks”, 2003.

[SRIS] Srisathapornphat, C.Jaikaeo and C.Chien-Chung Shen, “Sensor Information Networking Architecture,” International Workshops on Parallel Processing, Pages 23-30, 2000

[SN1] Archana Bharathidasan, Vijay Anand Sai Ponduru, “Sensor Networks: An Overview”.

[GDI] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, John Anderson , “Wireless Sensor Networks for Habitat Monitoring”, 2002.

[HEM] Edoardo S. Biagioni, K. W. Bridges, Brian J. S. Chee, “A Remote Ecological Micro-Sensor Network”, 2000.

[AUTS] Q.Han N.Venkatasubramanian, “Autosec:An integrated middleware framework for dynamic service brokering”, 2001.

[TMSN] SinemColeri, Sing Yiu Cheung, Pravin Varaiya, “Sensor Networks for Monitoring Traffic”, 2004.

[LS] Stephen So, Farinaz Koushanfar, Anatoliy Kosterev, Frank Tittel, “LaserSPECks, laser SPECtroscopic trace-gas sensor networks”, 2007.

[TOS] TinyOS web site http://www.tinyos.net/

[CTK] Contiki Operative System web site http://www.sics.se/contiki/

[BAR07] Paolo Baronti, Prashant Pillai, Vince Chook, Stefano Chessa, Alberto Gotta, and Y. Fun Hu, “Wireless Sensor Networks: a Survey on the State of the Art and the 802.15.4 and ZigBee Standards”, Computer Communications, 30, 2007, pp. 1655-1695.

[IEEE03] Institute of Electrical and Electronics Engineers, Inc., IEEE Std. 802.15.4-2003 ‘‘Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs), New York, IEEE Press. October 1, 2003.

[HEI04] W.B. Heinzelman, A.L. Murphy, H.S. Carvalho, M.A. Perillo. Middleware to Support Sensor Network Applications. IEEE Network, 18(1), 6– 114, 2004

[SOU04] E. Souto, G. Guimaraes, G. Vasconcelos, M. Vieira, N. Rosa, C. Ferraz. A Message-Oriented Middleware for Sensor Networks. In 2nd International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC 2004), pages 127–134, Toronto, Canada, October 2004

[GIA05] C., Giani M., Giorgetta M., Giusti A., Murphy A.L., and Picco G.P., “data collection in Sensor Networks: the TinyLime Middleware”, Journal of Pervasive and Mobile Computing, vol 4, n.1, pp. 449-469,Elsevier, December 2005.

[CRO] Crossbow Technology Inc. http://www.xbow.com.

[MOT] Moteiv Corporation. http://www.moteiv.com.

[MAD02a] Samuel Madden, Robert Szewczyk, Michael J. Franklin, and David Culler, “Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks”, Proc. 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002), pp. 49-58, Callicoon, NY, USA, June 2002.

[MAD02b] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong, „TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks”, Proc. 5th Symposium on Operating Systems Design and Implementation (OSDI 2002), Boston, MA, USA, December 2002.

[INT00] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin, “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks”, Proc. 6th International Conference on Mobile Computing and Networking (MobiCom 2000), pp. 56- 67, Boston, MA, USA, August 2000.

[YAO02] Yong Yao and Johannes Gehrke, “The Cougar Approach to In-Network Query Processing in Sensor Networks”, SIGMOD Record, 31(3):9-18, September 2002.

[YAO03] Yong Yao and Johannes Gehrke, “Query Processing for Sensor Networks”, Proc. 1st Conference on Innovative Data Systems Research (CIDR 2003), Asilomar, CA, USA, January 2003.

[MAD03] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong, “The Design of an Acquisitional Query Processor For Sensor Networks”, Proc. SIGMOD 2003, pp. 491-502, San Diego, CA, USA, June 2003.

[BON00] Philippe Bonnet, Johannes Gehrke, and Praveen Seshadri, “Querying the Physical World”, IEEE Personal Communications, 7(5):10-15, October 2000.

[BON01] Philippe Bonnet, Johannes Gehrke, and Praveen Seshadri, “Towards Sensor Database Systems”, Proc. 2nd International Conference on Mobile Data Management (MDM 2001), pp. 3-14, Hong Kong, China, January 2001.

[AMA05b] Giuseppe Amato, Paolo Baronti and Stefano Chessa S, “MaD-WiSe: Programming and Accessing Data in a Wireless Sensor Network”, International Conference on Computer as a Tool (EUROCON 2005), pp. 1846-1849, Novembre 2005.

[AMA06a] Giuseppe Amato, Paolo Baronti, Stefano Chessa and Valentina Masi, "The Stream System: a Data Collection and Communication Abstraction for Sensor Networks", 2006 IEEE International Conference on Systems, Man, and Cybernetics, October 2006.

[AMA06b] Giuseppe Amato, Paolo Baronti and Stefano Chessa S, "MaD-WiSe: A Distributed Query Processor for Wireless Sensor Networks", Technical Report ISTI-2006-TR-39.

[ZIG05] ZigBee Alliance, "ZigBee Specifications", version 1.0, April 2005.

[BLUE] Bluetooth Official Site, http://www.bluetooth.com/bluetooth/

[DAR01] DavideDardari, AndreaConti, ChiaraBuratti, RobertoVerdone, "Mathematical Evaluation of Environmental Monitoring Estimation Error through Energy-Efficient Wireless Sensor Networks", IEEE Transactions on Mobile Computing, vol. 6, no. 7, July 2007.

[DAR2] Tony Q.S. Quek, Davide Dardari, Moe Z. Win, "Energy Efficiency of Dense Wireless Sensor Networks: To Cooperate or Not to Cooperate", IEEE Journal on Selected Areas in Communications, vol. 25, no 2, February 2007.

5. NMR Spectroscopy

5.1. Example Instrument

As an example of remote instrumentation usage in NMR spectroscopy the Bruker Avance 600 spectrometer has been tested. It has been manufactured by Bruker BioSpin GmbH, Rheinstetten, Germany is owned by Faculty of Chemistry at Adam Mickiewicz University in Poznań, Poland.

High Performance Digital NMR Spectrometer is equipped as follows: UltraShield Superconducting magnet (14,095 Tesla), orthogonal shim system (BOSS II) with 28 shim gradients, smart magnet system (BSMS) for shim and lock control, digital lock control unit, three RF channels, digital temperature controller, shaped pulse generator, GRASP II accessory and the following NMR Probes:

- TBO: inverse triple 1H/{13C}/BB, 2H lock; f=5mm, rang of BB=31P-109Ag, Z-gradient,
- TBI: inverse triple 1H/{31P}/BB, 2H lock; f=5mm, range of BB=31P-103Rh, Z-gradient,
- BB: broadband, BB/{1H}, 2H lock; f=5mm, range of BB=31P-109Ag, Z-gradient.

The spectrometer has been used by scientists from Faculty of Chemistry working for Adam Mickiewicz University and Institute of Bioorganic Chemistry and Polish Academy of Sciences. The main fields of application are organic chemistry, physics and biochemistry. During the project some potential new groups of users have been identified: pharmacists, scientists from chemical industry.

5.2. Use Case Summary and Background

The following summary presents spectrometer’s use case and its most important information:

- this instrument is not used in configuration with other equipment
- situated in different location
- data is numerically represented
- if a data file is lost or corrupted, how feasible is it to reproduce an observation result: repetition of measurement
- the instrument can be directly operated by a remote user (by using VNC for example)
- instrument does not provide an Application Program Interface (API) for integration with custom-designed user or project software application
- measurements (data streaming) can be accessed from a remote location in both real time and offline

5.3. Infrastructure and Service Requirements

There are several types of requirements which have to be considered while performing NMR experiments:

The sample cannot be manually manipulated during experiment (the spectra are acquired according to the defined parameter setting). Only one sample per one experiment is allowed. Similarly one user is permitted to enter the site at the time and have access to the streaming media produced by observations. The following set of operations is available: locking, shimming, tuning, pulse calibration and pulse program parameter setting.

Time constraints might also be important for conducting NMR experiment. They have to be considered for experiments scheduling and assigning users to the instrument. The average time of a simple experiment is one hour although there are experiments that last few minutes and few days depending on problem complexity.

Measurement conditions depend on the solvent and probe used. They are performed in temperature range (-150–180∞C).

5.3.1. Storage Space

The described instrument storage requirements strongly depends on experiment’s parameters such as experiment resolution etc. The simplest parameters settings the lowest requirements are. The instrument needs maximum about 10-20 Gbytes for storage. Experiments results can be digitally stored on a local computer that controls spectrometer or in Data Management System. The offline processing may require additional space. Similarly to one of the previous described instruments if the file is lost the experiment has to be repeated which of course increases the cost of whole venture.

5.3.2. Data Transmission

PC with PCI Ethernet 10/100 Mbps LAN interface card (10 Base T) with RJ-45 connector for cat. 5/cat.5e cable specification is used to access the spectrometer. The determined data transmission rate is about 10Mbit/sec. It is strongly correlated to visualization served by VNC connection in which data are transmitted in either raw packed format. Additionally, if the data is sent to the other workstation for postprocessing the transmission rate might grow depending on data files size. In this case the faster network connection the faster data are sent from one workstation to another and thus the offline computations can be completed sooner.

5.3.3. Interactivity

Interactive issues are provided by VNC connection to the instrument.

5.3.4. Collaboration Tools

In some cases external applications (e.g. Cyana or Xplor software) have to be used for additional calculations. Offline data processing can be guaranteed by Bruker’s specialized software XwinNMR/Topspin. It is used to filter and/or extract a valuable information from the data provided at the end of the experiment.

5.4. Deployment Guidelines: Good Practices

There are few issues about which spectrometer users have to remember:

- Workstation serving spectrometer has to have a secure connection to the instrument and has to be separated from the other computers working in the local network for the safety reasons. Bruker Avance 600 is a very expensive equipment and access has to be limited to the trusted persons.
- Samples that is inserted to the instrument must be prepared in deuterated solvent.
- For successful sample deployment locking, shimming, tuning, pulse calibration and pulse program parameter setting must be applied.

5.5. References

[1] “Combinatorial analysis of 2D-NOESY spectra in Nuclear Magnetic Resonance spectroscopy of RNA molecules”, Ph.D. thesis, Marta Szachniuk, 2005, Institute of Computing Science, Poznań University of Technology

[2] Lawenda, M., Meyer, N., Rajtar, T., Okon, M., Stoklosa, D., Kaliszan, D., Stroinski, M.: Generalization aspects in the virtual laboratory system. Remote Engineering and Virtual Instrumentation 2004, ISBN 3-89958-090-7, Villach, Austria, 28-29 September, 2004

[3] Szychowski, P., Kopec, M., Lawenda, M., Wolski, M., Meyer, N., Mazurek, C., Stroinski, M.: Secure access to GRID resources. The 4th Cracow Grid Workshop, ISBN 83-915141-4-5, pp. 347-354, Cracow, Poland, December 12-15, 2004

[4] http://195.96.251.250/ringrid/db/view_iif.jsp?id=169.

[5] http://vlab.psnc.pl

6. Summary

As this deliverable shows, most instruments are currently not so demanding as initially thought. However, right now the instruments are poorly integrated into grid enviroments, and one can expect that the demands on the instruments will increase once the remote instrumentation idea has cought fire and remote instruments are used by researchers worldwide.

Especially, data storage as well as collaboration tools leave much to be desired, and are poorly integrated into the middleware architecture. Nowadays, collaboration is accomplished by low-level and therefore inadequate tools like Skype or Gadu-Gadu. The same is true for visualization components or the components which have the task to control the instruments over the grid: Too often we have heard that these tasks are currently fulfilled by tools like VNC. So, there are some gaps to be filled in subsequent projects.

Additionally, communication between grid nodes (for example the remote instrument's controlling host and the scientists' workstations) is not coherent. Workflow is rudimentary and differs from instrument to instrument. One needs to define communication protocols which take into account all the different data formats (video, analysis results, spreadsheets, real-time values, ...) which are encountered in a remote instrumentation scenario and define a workflow-based programming interface (API) which can be used by untrained scientists. This way, the remote instrumentation idea can be brought to researchers who are not computer scientists.

It is now up to WP4 to identify trends in the remote instrumentation research and see whether some of these gaps are about to being closed.

References

The references are contained at the end of each individual section for better text comprehension.

Contact Information

All authors affiliation:
Poznań Supercomputing and Networking Center
ul. Noskowskiego 10
61-704 Poznań, Poland

URL: http://www.man.poznan.pl
Tel. (+48 61) 858-20-00
Fax (+48 61) 852-59-54

Davide Adami	davide.adami @ cnit.it
Antonio-Blasco Bonito	blasco.bonito @ isti.cnr.it
Marius Branzila	branzila @ ee.tuiasi.ro
Luca Caviglione	luca.caviglione @ cnit.it
Romeo Ciobanu	rciobanu @ ee.tuiasi.ro
Davide Dardari	davide.dardari @ deis.unibo.it
Franco Davoli	franco.davoli @ cnit.it
Marcio Faerman	marcio @ rnp.br
Alberto Gotta	alberto.gotta @ isti.cnr.it
Damian Kaliszan	damian @ man.poznan.pl
Constantinos (Costas) Kotsokalis	ckotso @ admin.grnet.gr
Angelos Lenis	anglen @ netmode.ece.ntua.gr
Fabio Marchesi	fabio-marchesi @ libero.it
Thomas Prokosch	tprokos @ gup.uni-linz.ac.at
Tomasz Rajtar	ritter @ man.poznan.pl
Michael Stanton	michael @ rnp.br
Zhili Sun	z.sun @ surrey.ac.uk
Jin Wu	jin.wu @ surrey.ac.uk
Tasos Zafeiropoulos	tzafeir @ grnet.gr
Yanbo Zhou	y.zhou @ surrey.ac.uk