An Architectural Framework for Distributed Monitoring Systems

Mihai Dinsoreanu
iQuest Technologies
Motilor st. 6-8, 400001 Cluj-Napoca
Romania
mihai.dinsoreanu@iquestint.com

Iosif Ignat
Computer Science Department
Technical University of Cluj-Napoca
Baritiu st. 26, 400001 Cluj-Napoca
Romania
Iosif.Ignat@cs.utcluj.ro

Abstract – **Modern data acquisition, command and control systems need to operate under a variety of conditions and have to meet specific constraints like requirements for performance, availability, scalability, security, maintainability or interoperability. Therefore, systems built on well-designed distributed architectures are more likely to meet these needs. We present in this paper DMF (Distributed Monitoring Framework): a framework that allows for rapid development of SCADA systems able to support interaction with different hardware and sensors, interoperability in heterogeneous networks and extensibility with new functionality. A specific application based on the DMF framework is presented in order to prove usability and efficiency.**

I. INTRODUCTION

The number of remote monitoring applications will increase rapidly in the near future, based on an increasing need of data elicitation from different measuring equipments, data storage and rapid data processing which also leeds to the need of monitorizing and controlling the measuring equipment on a permanent basis and location independent.

Currently, most of the SCADA systems (Supervisor Control and Data Aquisition) do not exploit the advantages of an distributed architecture mostly because they rely on legacy systems that do not meet the requirements of a modern command and control system. In our work, we considered the following main requirements of a modern SCADA system:

1. *Performance*: performance constraints are related to data propagation from sensors. Data reading and data synchronisation intervals are configurable and depend on the colected data size and communication costs.
2. *Availability*: to meet the availability constraint (99.99%) a redundant, both hardware and software, architecture is considered. Several communication methods with different priorities are suported.
3. *Scalability*: the business model allows the execution of several DMF instances therefore concurrent systems are to be managed. Data is stored in one or many databases, each database containing stored procedures to provide a unique interface for retrieving each unit's state and position.
4. *Security*: security is considered at both integrity and confidentiality levels. Radio communication in 3DES encription and physical security mechanisms like smart –card authentication can be added.
5. *Maintenance*: the large number of distributed components on a wide area makes hardware maintenance a challenge. Hardware power consumtion is kept very low and solar cells can be added while software is updated and maintained in a centralized manner.
6. *Extensibility*: DMF systems should be able to interact with already existing SCADA systems, and also to support dynamic adding of new units using different communication means.

Considering the requirements mentioned above, we developed the Distributed Monitoring Framework (DMF), a framework for rapid development of SCADA systems able to support interaction with different hardware and sensors, interoperability in heterogene networks, extensibility and standard interfaces.

Systems based on DMF are able to collect information from different environments, to react on entry data variations either by generating messages or by performing some preconfigured actions.

Elicited information can be accessed in different ways: via desktop applications, Web applications or mobile devices, PDA's or PocketPC's.

The DMF's hardware layered architecture contains two types of field equipment: DN (Data Node) and AN (Access Node).

DN are placed close to the sensors, they collect measured data and send it to AN over a radio a connection. AN can be seen as data collectors from DN and gateways to the DCS (Data and Control Services). DN and AN support different communication media depending on the specific usage conditions.

Data is collected by the AN and sent to the DCS. On the DCS level data is processed and stored.

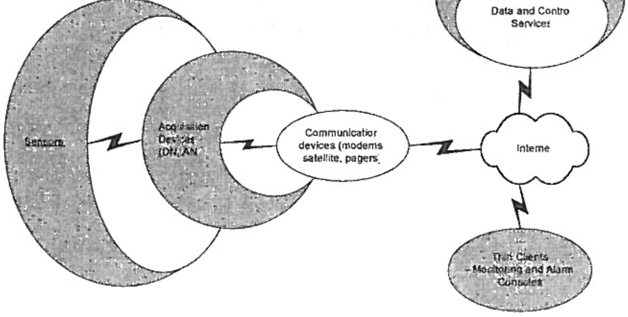

Fig. 1. DMF Overview

II. DISTRIBUTED MONITORING FRAMEWORK

DMF is a distributed system designed for developing specific wireless or cable-based data acquisition applications and Web-based monitoring and control applications.

A. Hardware Architecture

The hardware architecture is designed on one or more DN and AN layers and a central DCS as displayed in the DMF structure overview in Figure 1.

1. *Data Node* – the main task of the DN is to collect data from digital and analog sensors at configurable intervals, to store them and to periodically transmit them to the AN at the superior level. DN can be configured to store also the minimum and maximum threshold values allowed for various parameters, and is able to raise the connection to AN immediately if certain read values are exceeding the limits.

Because the distributed system is an asynchronous one, the base reaction logic was implemented at DN level. The command abilities of digital outputs allow DN to act independently and immediately by setting the value of a digital output line and thus controlling the collection devices without any specific instructions from the DMF superior levels. The reaction speed is therefore significantly increased, providing a quick local response to critical events in case of communication errors.

Several transport level types are supported for the communication between AN and DN: serial transmission using direct connection, radio modem, GPRS or CDMA modems as well as low-altitude satellite communication.

Considering DN scarce resources, for the network communication level we chose the UDP protocol which suits also asynchronous connections.

DN's hardware architecture consists of:

- C8051 Processor family with combined frequency signal, Flash memory, RAM, 12-bit Analog – Digital Convertor, UART Interface.
- Analog Inputs
- Digital Input/Output
- JTAG Connector
- Configurable EEPROM
- Radio Interface, RS485 or RS232 Interface
- Optional LCD Panel with keyboard.

The components of the Data Node structure are shown in Figure 2.

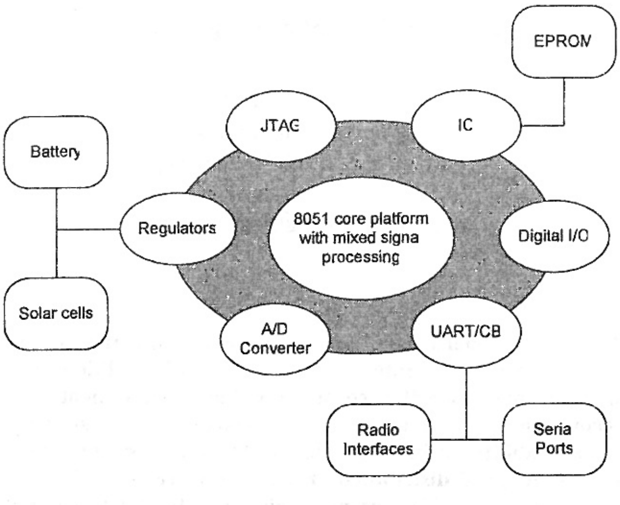

Fig. 2. DN Structure

2. *Access Node* - AN main function is to periodically collect data from one or more attached DNs, archive the received data and periodically send the archives to the Data and Control Services (DCS). More than that, AN also receives alarm messages if input values are exceeding limits and sends the alarm messages immediately to the DCS. An AN device can be either directly connected to the DCS or via Internet using a local Ethernet network, a phone modem, a bidirectional pager, a cell modem or a satellite modem.

AN's hardware architecture consists of:

- ARM 7 Family Processor
- Configurable Flash Memory
- Configurable RAM
- Ethernet 10baseT Port
- Radio or modem Interface for DN's
- Internet Interface via phone lines, bidirectional pager, cell modem or sattelite modem
- Programmable Digital Input/Output
- Serial Connectors
- Optional LCD Panel with keyboard.

The components of the Access Node structure are shown in Figure 3.

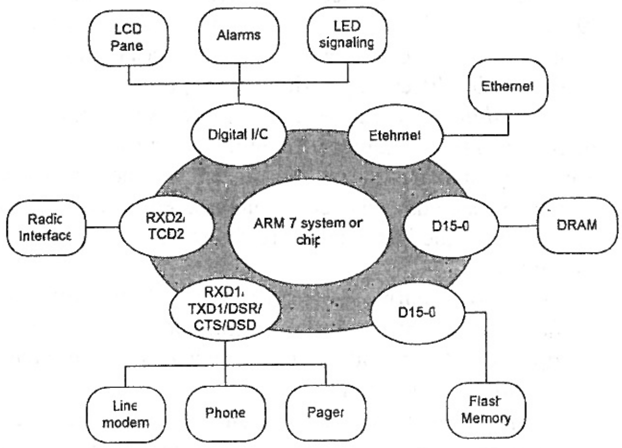

Fig. 3. AN Structure

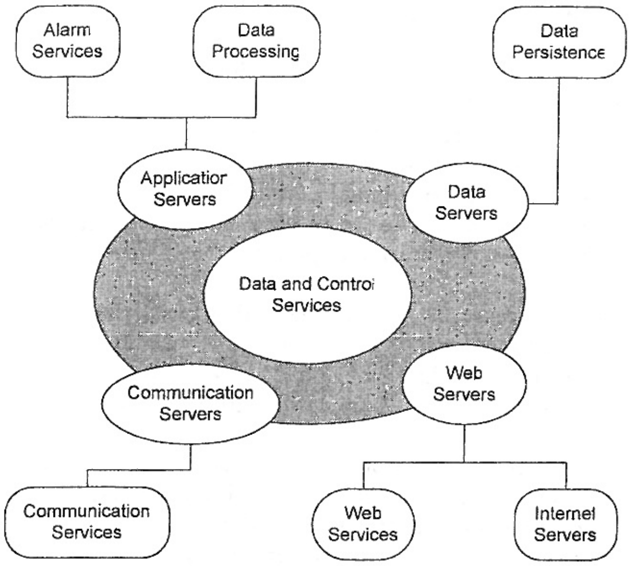

Fig. 4. DCS Structure

3. *Data and Control Services* - Considering the framework constraints (connectivity, scalability, maintenance, security and recovery), the DCS's Architecture is designed as a set of peer services in order to allow efficient functionality distribution. DCS is therefore deployed (distributed) on several servers, each of them being assigned specific responsibilities:

- The Communication Servers are responsible with the ANs communication. They receive regular data or alarm messages and sends commands and function parameters back to AN. Received data is sent to the Data Servers while alarm messages are sent to both the Data Server and the Alarm Services. The Communication Server components are deployed on the Communication Servers or Application Servers.
- The Alarm Services send alarm messages to pre-assigned IP Addresses of alarm consoles using UDP datagrams. The alarm consoles can be in- or outside the Data Center. The Alarm Services components are installed on the Application Servers.
- The Data Persistence Services are responsible for holding all the messages for the DMF devices allowing for reuse. The same structure can be easily adapted to different Web applications and persistence services are shared between them.

The components of the Data and Control Services structure are shown in Figure 4.

4. *Meshing* – Meshing wireless topologies [2], [4] allow for "point-to-point" and "peer-to-peer" connections, creating thus ad-hoc, multi-hop networks.

Self-Configuration

The Mesh network is able to self-configure and doesn't need human action to send messages to destination.

Self-Repair

If one or more nodes fail, the network is able to redirect messages so the functionality is not necessary affected.

Redundancy

Redundancy is mainly determined by the node density in mesh networks. The network can be on purpose oversized in order to create one or more transmission routes for each device.

Scalability

A scalable network can be extended to hundreds or thousands nodes. Since there is no central control point, it is recommended to add several collection points.

Systems developed as instances of the framework are able to work on heterogeneous transport media according the user's specific needs. For this purpose a communication protocol was developed that is reliable and resistant to infrastructure and network changes

The protocol is defined as a collection of interfaces and for every specific communication layer the corresponding classes are instantiated. Class responsibility for the instantiation of specific classes is delegated by using the factory pattern.

Another protocol feature is the ability to automatically commute between different communications media if the current one becomes unavailable or if a more performant becomes available. Commutation is possible on both AN and DCS levels and DMF may function concurrently with different configurations.

B. *Software Architecture*

1. Data Node – Due to limited resources at the DN level the software is low-level using 8051 processor assembly language. For this reason architecture efforts have been targeted towards optimization with the reuse possibilities rather low, having some shared routines or just copy-paste in some cases.

The DN software contains a set of routines that implement both data elicitation and control functions and the communication protocol. Several communication protocols are available corresponding to cable or wireless connections.

The behavior of the device is controlled by a configuration file, editable via a dedicated application and remotely uploadable. The config file defines parameters like: device unique ID, data reading period, report frequency, alarm thresholds and corresponding actions etc. By changing the config file, a device can be reused in different system area or even in a different application without modifying any of its software.

DMF instances are actually reusing hardware and knowledge at the DN level.

2. Access Node – At the AN level, the software is an independent application having the following functionalities: controlls communication with supervised DNs, controlls digital inputs/ouputs, controlls communication with the DCS and also the optional LCD display. All relevant parameters like the devices ID, report frequency, list of supervised DNs, communication parameters are stored in a config file that can be edited and remotely uploaded.

Due to bidirectional communication, AN does not only send data to DCS but also accepts commands from DCS and the attached DNs. A locally stored certificate and a certificate stored on the DCS allows the authentication of the parties involved in the communication. A special functionality allows for remote firmware update on both AN and DN leading to a significant decrease of maintenance costs. For problem diagnosis the dump of the AN's state on the DCS is analyzed by the technical staff.

C. *Scaling the Distributed Monitoring Framework*

There are three main architectures that were thoroughly tested but different combinations are also possible.

The three architectures are:

1. Complete Architecture – the complete architecture (classic) contains all the platform components, both hardware and software. It also implements all the methods and the entire communication protocol. It is star-structured containing several DCSs with centralized control, each monitoring several ANs. Each AN manages up to eight DNs (this limitation can be surpassed but it is recommended from the data storage point of view).

2. DN Based Architecture – this version consists of a limited architecture containing just DNs and the DCS.

The communication between them is realized by implementing a bridge component between the DN and DCS, the bridge component implements the AN interfaces on both communication ends.

3. AN Based Architecture – this compact architecture contains no DNs, the acquisition functionality beeing done directly by ANs.

This architecture is mainly used in case of intelligent sensors or sensor clusters placed in accessible areas.

III. CASE STUDY

To demonstrate the usability and efficient development based on DMF, a proof of concept is presented next. We developed a simple application to monitor the parameters of the river water. The application has the AN based, compact, architecture as defined above.

A sensor cluster is directly connected to the AN device using its RS232 serial interface. The communication between AN and DCS is GPRS-based using UDP.

Due to the fact that the communication protocol supports several network layers, TCP can also be used but this choice would increase traffic and increasing therefore cost.

In addition to the basic architecture, the application integrates a GPS module that allows geographical visualization of sensors in field. The integration of the module was easy, being treated as a standard data acquisition device. The GPS device interface was adapted to the DMF acquisition interface and the data was collected and processed by the system in a standard way. The presentation layer used this data for geographical information.

A. *Application Functionality*

This simple application provides the following functionalities:

- visualization of parameter values elicited on field
- visualization of alarms
- configuration of active sensors

Data visualization is structured on several detail levels according to the structure of the monitoring devices.

The GPRS modems connected to AN provide also positioning information through the embedded GPS devices. Actually, AN is dealing with modems in a dual manner: as modem and as GPS sensors.

AN is sending the received information to DCS for storage and processing.

Due to the lack of support of the GPRS network for real IP addresses and the usage of UDP for reducing network traffic, the communication protocol was only partially instantiated.

When initializing AN the data reading frequency and the data transmission frequency is configured. The two frequencies are choosen such as AN's buffer will not overflow even if communication with DCS fails several times.

On the first connection AN sends its own configuration too so DCS will know the established frequencies. In case of reception failure at the configured frequencies, an alarm messag ewill be sent to the supervisor to check the exact causes of the interruption. For reliability reasons, a second communication system can be considered (for example on satelite modems).

B. *Application Presentation Layer*

All the data is saved at the DCS level and can be processed in order to provide usefull information to the user. The provided information is related both to the last readings and also to the parameter history in graphical or table mode.

The structure of the Web application is presented in Fig. 5.

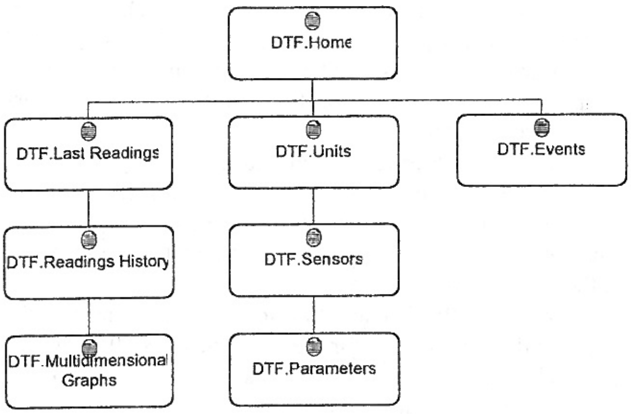

Fig. 5. DMF Application Instance Structure

The application uses two Web Services that send data in both formats: XML for tables and graphical images. Grapg rendering is performed on the server therefore the entire application up to this point is browser-independent.

The entire DMF infrastructure is completely transparent for the end-user that can access only the monitoring devices and associated values read on the field.

IV. CONCLUSIONS AND FUTURE WORK

As shown in this example, DMF provides the developers with all the necessary interfaces in order to rapidely instantiate it and build specific applications.

The application tier is completely separated from the presentation tier, therefore different types of graphical interfaces can be implemented. The entire infrastructure, monitoring components and communication means, is seen by the developer as a unitary system.

V. REFERENCES

[1] George Coulouris, Jean Dollimore, Tim Kindberg, *Distributed Systems Concepts and Design*, Third Edition, Addison-Wesley, 2001

[2] Yih-jia Tsai, Philip K. McKinley, "An Extended Dominating Node Approach to Broadcast and Global Combine in Multiport Wormhole-Routed Mesh Networks", *IEEE Trans. on Parallel and Distributed Systems,* vol. 8, no. 1, Jan 1997

[3] L. Lamport, "Time, clocks and the ordering of events in a distributed system", *ACM Communications*, vol. 21 No. 7

[4] Robert Poor and Brent Hodges: "Reliable Wireless Networks for Industrial Systems", Ember Corporation, Boston, Massachusetts

[5] K. Wreder, Yi Deng, "Architecture-Centered Enterprise System Development and Integration Based on Distributed Object Technology Standard," in *Proceedings of the Twenty-Third Annual International Computer Software and Application Conference*, 25-26 October, 1999, Phoenix, Arizona