Real Time Tools for Building Automation Applications Generator

Florin Hartescu, Claudiu Danilov, Mihaela Cosma and Lucius Nedelcu

Real Time Systems Laboratory
Research Institute for Informatics
8-10 Averescu Avenue.
71316 Bucharest
ROMANIA
e-mail: flory@roearn.ici.ro

Abstract: Building automation is realised by means of a real time information system for supervision and control of central heating, electrical lighting, access, and energy consumption. All these elements will provide security and comfort to a modern building.
The system has been developed as part of a research work carried out at Real Time Systems Laboratory of the Research Institute for Informatics in Bucharest. The system is supported by a real-time architecture RT-ARCH [1].
On implementing the system, an IBM PC 486 running under Windows NT operating system and a network of PLC-s [2] were used. This application is characterised by an object oriented approach.

Keywords: Real time system, supervision, PLC, building automation, distributed system

1. Introduction

Generally, building automation applications consist of some processes which take place in real time, such as data acquisition from the process, data acquisition simulation (during the testing phase), communication between the process and the control and monitoring system, that means graphical display of the collected data, reports on the collected data and alarm record and display. Such processes refer each room, group of rooms or the entire building.

An Object Oriented Approach

The software package for a distributed building automation system is object oriented at the conceptual level, i.e. with respect to the source code, and also at the functional level, i.e. with respect to the use and exploitation by the operator.

The Code Level

A high level object oriented language has been used for the development of a prototype system. A modularisation of the application will thus be possible, and consequently, new advantages are to be obtained: operative development, efficient team work through combining the code written by various programmers, code reuse in other projects. During the application development and the encapsulation of data and functions,. a hierarchical system resulted, which enables the realisation of intelligent objects capable of making their own decision based on the available data [9].

We have chosen to use Visual C++ v4.0, due to the fact that it meets all the requirements for the software product development.

The Application Level

The software package has been modularised in an object oriented hierarchical structure.A spatial and functional level decomposition will allow modifications in the system architecture during the application development.

The spatial decomposition uses the "document" as the object at the top of the hierarchy. It refers the whole building and contains all data about the building floors. The document also contains information on the date of the latest updating,. and all the floors in the building. The floor, which is the next level of the hierarchy, refers the spatial arrangement of the rooms in the building. The room is the lowest object - the base cell - of the application. It encapsulates both data about its configuration and specific functions such as data acquisition, alarms, data storage, etc. The database is an autonomous object, which can be accessed by every object presented in the upper rows, for both reading and writing. It contains two data types referring the building structure as described by the configuration module operator, and the data collected from sensors during the operation.

Both components run independently of each other, so that the application should not be reconfigured at the level of the collected data structure, if, say, the structure of a building is modified.

The Functional Decomposition

The object oriented modularisation of the software product provides development tools for user specific needs. The configuration module uses the same classes - objects - for the customisation of an application as the other modules do. So, the room object could, after having been attached to the document object, by means of the data acquisition module, automatically take over the data from the PLC-s and write them in the database. This renders the product functional in the sense that once the configuration phase is over, the application will run normally.

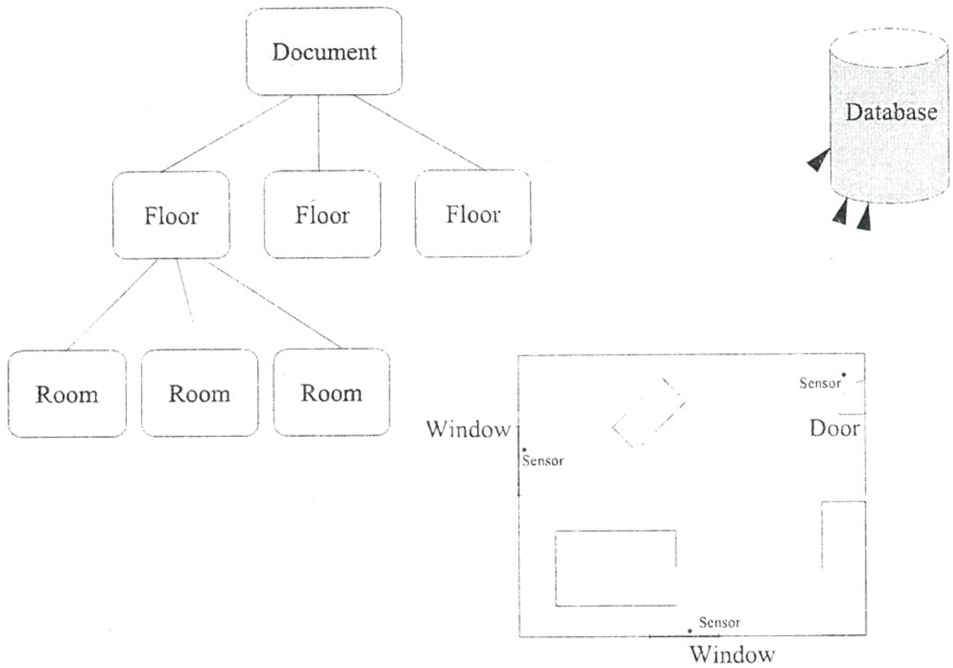

Figure 1

2. The System Structure

The system hardware is the following:

a) an IBM PC 486 computer running under Windows NT operating system [6].This computer collects (reads from the serial port coupled to the "train" of PLC-s) the data sent and puts them in a waiting queue (buffer) for as long as takes another process to identify and extract them from processing;

b) modern PLC-s from Telemecanique - TSX 17- with extensions for analogical inputs and asynchronous interface TSX SCG 113 which have the role to acquire the information provided by transducers, and send it on the serial communication line;

c) data acquisition equipment (sensors and transducers);

- the transducers are used for the data on which a permanent checking is not required: temperature, power, frequency;

- sensors for the case when the data observed are only relevant to a limit value overfulfilment or to the case in which there are actions of type "all" or "nothing" as with the sensors for presence, fire, injury.

Figure 2 presents the scheme of a communication architecture module. PLC-s can be interconnected and connected to a PC, on a serial asynchronous line RS 232 by means of TSX SCG 113 [4].This is an example of implementation at the level of PLC, which is the data acquisition equipment.

3. The System Architecture

The software system is notified to the Windows NT operating system. It has a modular architecture, being realised from many tasks, which communicate among them. The components of the software system are the following:

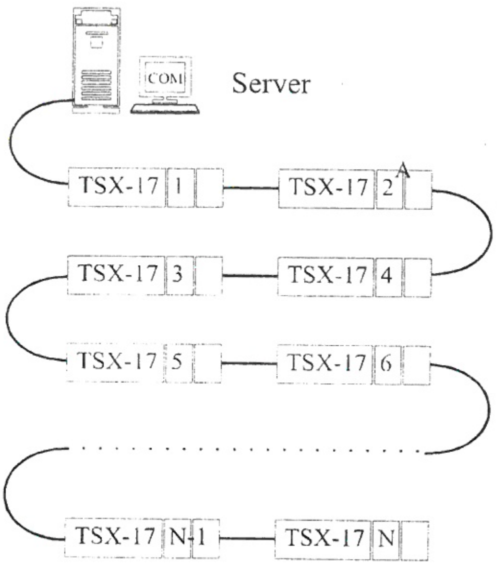

Figure 2

- configuration module
- database
- data displaying module
- communication module
- configuration display module
- data acquisition module
- data simulation module
- control module
- alarm module
- report module

4. A General Presentation

This software product is an implementation of the "building automation" concept, which assumes the carrying out of the following tasks:

- develop a model of the building by means of a graphical interface allowing that the structure at each level of the application is drawn;

- define the structure of the supervision data to be collected for each room ;

- describe the events and alarms which will be signalled by the values of the data collected.

The software product offers an MDI (Multiple Document Interface) implementation. Due to this, a many-storey building can be controlled.

Each level is represented by a document a window in the application is associated with. The user will thus be able to switch among the levels of the building in order to have data about these levels displayed (for example the collected data structure for a certain room) or alarms displayed just the moment of their occurrence.

If the software product is to apply to several buildings, it is highly recommended to group these buildings in sub-directories (remember that each level of a building is associated with a document, which is stored on a disk file).

Details about the relations in the database and the way the actual data structure has been chosen, will be transparent to the user.

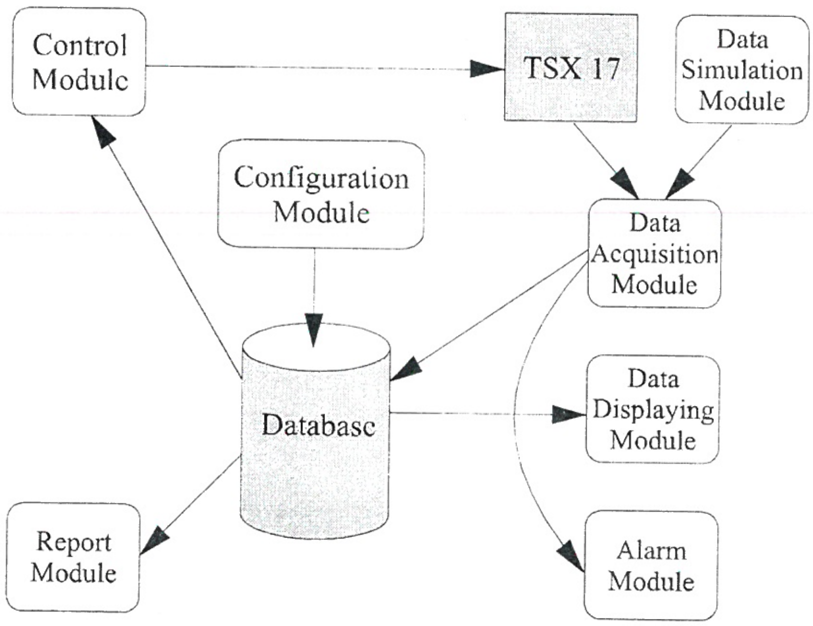

Figure 3

Based on the values of the collected data, events are generated. These can be alarms, which will be stored in the database. All of these are useful for generating reports on the detected events, and creating the data source for an after-damage main analysis.

From the point of view of event/alarms generation, one should mention that all the produced events are automatically recorded in the database. Besides, on the screen there will be details about alarms for the document (level) which is in the active window. In this way watching only one window is enough for the supervision of one level (floor) of the building.

A user help is in the fact that as long as the program runs, at the bottom of the window, there will be displayed information about the action the user is going to take by selecting a certain option from a menu or by clicking one button of the toolbar.

The prototype has also implemented an on-line "help".

5. Conclusions

The development of object oriented tools for building automation extends, diversifies and capitalises on the so far acquired experience of the "Complex Real Time Systems" team. The object oriented methods used in the RT-ARCH architecture [1], are advanced methods for automatically controlled systems.

Real time applications for building automation need to run on high performant hardware :

-PC 486 DX4/100 MHz Computer (at least 16M RAM);

-Video card with graphic accelerator on 32 or 64 bits, very reliable hard disks: EIDE or SCSI;

-PLC-s with extensions for analogical and digital inputs;

One of the most important characteristics is the software product reliability. This is meant to operate under Windows NT operating system, which ensures high portability, and to use an object oriented architecture.

REFERENCES

1. HARTESCU, F., **RT-ARCH – A New Approach in Real Time Application Design**, IASTED Mini and Microcomputers and Their Applications, Lugano, Switzerland, June 1990.
2. **Télémécanique, PLC TSX - Documentation TSX**, 1993 .
3. **Direct Members Newsletter**. BatiBUS International Club.
4. **Télémécanique TSX 17 Micro-automates,** 1992.
5. **ALAN x33 Building Automation System**.
6. MURRAY III, W.H. and PAPPAS, CH.H., **Application Programming for Windows NT**, MCGRAW -HILL, 1993.
7. PETZOLD, C., **Programming in Windows**, 1990.
8. **Wind River Systems, VxWorks**, Reference Manual, 1.0 Beta, 1993.
9. NEGRESCU, L., **C++ Language,** MICROINFORMATICA, 1995 (in Romanian).
10. CRISTEA, V. et al, **Programming Techniques**, " Teora" Publishing House, 1993 (in Romanian).
11. **dSPACE Product Overview**.
12. **Parallel and Distributed Computing and Systems**, Proceedings of the 7th IASTED/ISMM International Conference, Washington, DC, ACTA PRESS, 1995.