PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 36, NOS. 3-4, PP. 281-294 (1992)

SUPERVISING MICROWAVE TELECOMMUNICATION NETWORKS WITH THE REALEX EXPERT SYSTEM SHELL[1]

István KERESE, Tamás ZSEMLYE, Károly TILLY,
Zoltán SZALAY and Balázs VADÁSZ

Department of Measurement and Instrument Engineering,
Technical University of Budapest,
H-1521 Budapest, Hungary

Received: December 11, 1992

Abstract

To effectively supervise a telecommunication network, an intelligent supervisory system is proposed consisting of a traditional process monitoring, a fault diagnostic, a communication handler and a database manager subsystem. The whole system is based on a generic expert system shell designed to operate in industrial environment. The diagnostic subsystem contains a two-level inference engine to operate on structural information and traditional *if–then* rules. To ensure easy mapping of the supervisory system to any telecommunication network a configuration environment consisting of several compilers integrated into a multi-window editor was also developed. The system was implemented on two interconnected IBM PC-s running MS WINDOWS 3.0.

Keywords: supervisory control, expert systems, monitoring, communications control applications, software tools.

Introduction

Telecommunication networks are complex, distributed systems of sophisticated components. Their reliable operation is crucial not only for TV channels or cellular telephone users, but also for air traffic controllers and ambulance cars, where a breakdown almost surely causes more trouble than mere inconvenience. A system of such complexity cannot easily be supervised, let alone diagnosed in case of malfunctions. To support these activities a distributed supervisory system was developed, which provides user-friendly tools for the efficient maintenance of the network.

The Microwave Network

Microwave telecommunication networks are systems of high complexity consisting of several (sometimes hundreds of) radio relay stations. Each of

[1]This project was sponsored by the Hungarian National Committee for Technical Development.

them receives information from other relays or primary input channels and transmits it to other relays or output channels.

A relay station consists of a receiver, a transmitter and a tree shaped multilevel multiplexer structure which is responsible for connecting the receiver and the transmitter to the input/output points which range from simple telex lines and telephone channels to radio and TV signals. Channels of different relay stations can be associated with one another, thus establishing a network of communication lines.

A network like this is so sophisticated and the service providing is it so expensive that it cannot be operated fully automatically without human operators. However, relay stations are highly reliable (i.e. the possibility of malfunctioning of a station is very low), so it is not worth hiring a distinct operator for every station. According to the current practice a central monitoring station is set up, where a few (generally one or two) operators keep track of the events in the whole telecommunication network.

The relay stations are equipped with remote control units which perform primary data acquisition and communication tasks and are responsible for connecting the relay station to the central monitoring station via a reserved global communication channel, the so called service channel, which is in essence a low speed serial bus.

Services provided by the microwave network should be as reliable as possible. If despite of all efforts a failure in signal transmission occurs, the reason should be detected and eliminated as soon as possible. The major difficulty is caused by the fact that a malfunctioning station sends alarm messages not only to the monitoring center which oversees the whole network but also to the relays is connected with it.

If a station receives an alarm message from one of its neighbours it also forwards it to the central monitoring station, therefore if an alarm message is received from a relay it does not necessarily mean that the station is faulty. Since unmanned relay stations are often located in remote places (which are) difficult to access (top of mountains, etc.) they should be monitored by systems providing powerful tools for quick and automatic error detection, localization and diagnosis to find the exact location and the cause of the network malfunction before the service personnel set out for the site.

General Architecture of the Supervisory System

To match the requirements mentioned above we developed an on-line supervisory system providing the operator with explicit, high level and intelligent information about the current state of the network.

The architecture of the system was derived from the generic tasks of a supervisory system and from the special requirements of the monitored telecommunication network. The main tasks of a supervisory system are as follows:

- Data acquisition from the supervised object/process;
- Data evaluation and preprocessing to create output can easy be to interpreted by the operator;
- Displaying preprocessed data and various system messages (e.g. current values of parameters, signal trends, special events, textual warning and error messages);
- Logging events and messages in log files;
- Receiving operator commands and transmitting them to the supervised object/process.

To avoid collisions on the service channel a flexible polling mechanism is used: although the central station is the exclusive master cyclically initiating polling of the relay stations, the polling order can vary accordingly to the current needs. To keep the amount of the transmitted information on a minimal level we opted for a variable message length protocol with minimal overhead.

Because of the high complexity of the solved tasks and the rather strict timing requirements the supervisory system is implemented on two computers interconnected with an RS–232 serial line and consists of a monitoring, a diagnostic, a communication and a database management subsystem. One of them is charged with traditional monitoring functions while the other runs a fault diagnostic system. The operation of both computers is based on the network database storing information about the topology, actual parameter values and other characteristics of the supervised network.

Since our goal was to develop a supervisory system for a generic network and not for a particular one we had to assure the easy mapping of the supervisory system to any network topology and station structure.

In order to achieve this goal we also created a configuration environment processing declarative descriptions of the targeted network, which makes it possible to define any network dependent feature in the supervisory system.

The Monitoring Subsystem

The monitoring subsystem — essentially a traditional supervisory station — performs communication with the network, generates data and trend displays, alarm and warning messages, and provides possibility for remote

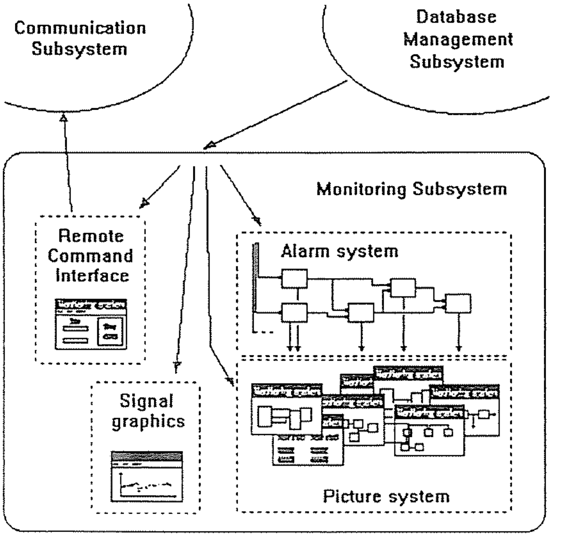

Fig. 1. Architecture of the monitoring subsystem

control. Besides these it is also responsible for providing data to the diagnostic subsystem.

The monitoring subsystem displays the supervised network or a part of it from different aspects. The main duties of this subsystem are as follows:

- Managing continuous updating of various pictures describing the microwave network;
- Alarming the operator when a relay station fails;
- Providing operator interface to issue remote commands.

The technological picture system helps to review the state of the microwave network. These pictures are organized in tree structure. The lower a picture in this hierarchy is placed the finer details of the system are shown on it.

The picture at the root of the tree describes a rough view of the supervised network representing the spatial and logical location of relay stations and supervisory systems. Pictures at the second level contain detailed representation of distinct parts of the network, usually comprising only a few relay stations.

Pictures at the next level of the hierarchy show the internal structure of relay stations. These pictures describe the building blocks of the relays (i.e. the functional elements) along with their error state and connections to each other.

The most detailed information about a relay station is described on the fourth level pictures. These pictures show the values of analog and

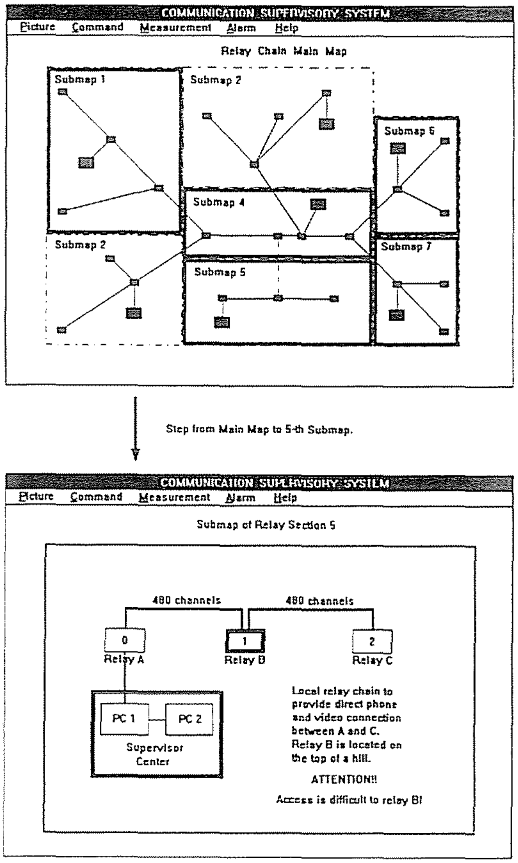

Fig. 2. The main map and a submap of the microwave network

digital signals of the selected functional element and their counterparts on the connecting relay station. The point in displaying signal values of two functional elements placed on two distinct relays is that these two elements of the same type form a logical communication channel.

The displaying of the pictures can be controlled with the help of menus or by clicking the mouse at 'hot spots' on the screen.

The alarm system activates the display driver to indicate the error state of those faulty elements which are currently displayed. Depending on the seriousness of the detected failure two different error levels can be displayed: one for simple errors and one for emergency situations.
The alarm system can provide three different outputs:

- The computer beeps in case of an error;
- The symbol of the faulty element or relay station is colored differently from those operating normally. This feature is assured by the

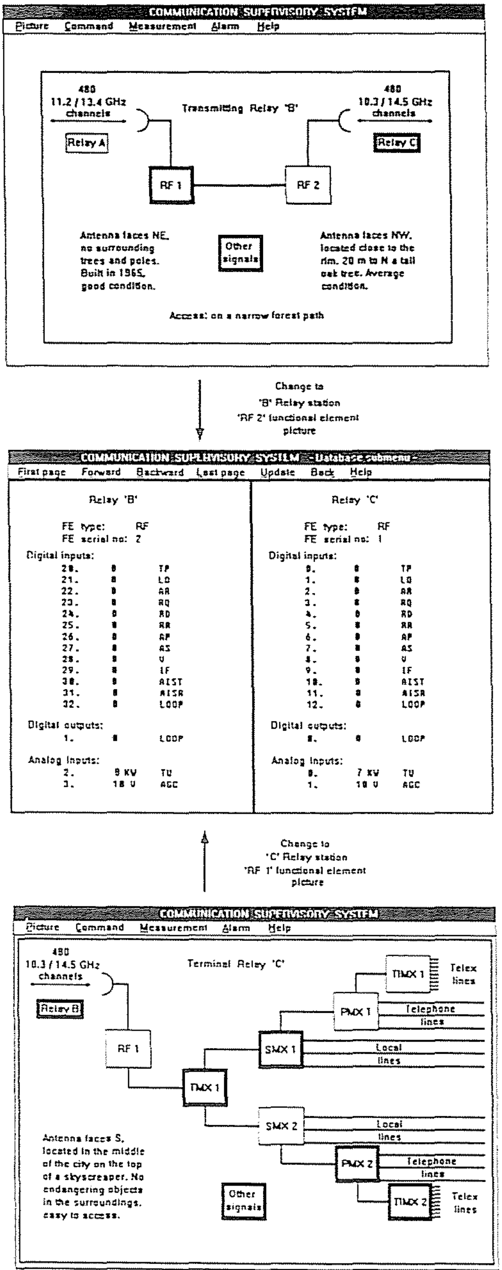

Fig. 3. Pictures describing the internal structure and signal values of relays

alarm propagation network, which is triggered by the changes of input signals;
- If the place of the error cannot be shown on the technological picture a message is sent to the operator.

The monitoring subsystem provides a menu driven interface for the operator to issue remote commands which can be used to change the state and to perform remote measurements on the relay stations. Results of measurements can be displayed graphically or in tables.

The Diagnostic Subsystem

The diagnostic subsystem — being an intelligent supervisory station — supplies the operator with high level information about the network, such as textual data logs, operational statistics and intelligent diagnostic messages. Messages from operators of other monitoring stations are also displayed by this subsystem.

In alarm situations (e.g. when a station breaks down) it is very difficult for an operator to make fast and correct decisions, because he may receive too many and sometimes apparently contradictory alarm signals. A good method of decision support is to incorporate information about the structure and behaviour of the network (DAVIS, 1985) and of individual stations into the supervisory system. This method can assure accurate decisions within reasonable time, although enables the operator to override automatically generated diagnostic advices.

The operation of the diagnostic subsystem is based on the same communication and database management subsystems which are used by the monitoring subsystem, consequently the basic means of data exchange between the monitoring and the diagnostic subsystem involve the database network. However, the diagnostic subsystem only reads the database while the monitoring subsystem is responsible for maintaining the network database consistency.

The diagnostic subsystem performs the following activities:
- Traditional textual data logging of significant events (e.g. remote commands, failures, etc.) with a short description of the event; data logs can be watched on the display screen or hardcopied to a printer;
- Computing failure statistics on the operator's request for specified stations, communication lines and printed circuit board types.

From the failure statistics the operator can acquire information about the relative reliability of different system components. After a longer period of operation the weak spots of the system (e.g. unreliable board types, etc.) can be pointed out, supporting later improvement of the network.

In a highly complicated network there may be more than one supervisory station; so called mobile supervisory stations can also be temporarily set up by service personnel during reparation of a faulty station. It is essential to provide the operator with the possibility to communicate with his colleagues in the form of textual messages. The main advantage of this method compared to e.g. telephone conversation is that this way the transmitted messages can be handled like events and can be automatically logged.

The diagnostic subsystem gives explicit textual advices to the operator regarding the current fault. Since it is the most sophisticated feature of the diagnostic subsystem we will discuss it in the rest of this section.

The advisory system is built on the knowledge-based approach with a multi-layered, modular knowledge representation scheme and a two level inference engine. The most important requirements are as follows:

- The results of the advisory system have to be automatically generated from the actual network database within the time limit, this requires some kind of event driven real-time operation;
- Since the state of the network is changing continuously the system must be prepared for time variant operation, i.e. some of the drawn conclusions must be withdrawn after a time; this feature requires nonmonotonic inference engine (WINOGRAD, 1980);
- The supervisory system must be able to handle efficiently different network configurations. Consequently, only methods storing information of the network in a very compact form are satisfactory, though this feature contradicts the high complexity of the network.

Considering the requirements above we chose a multi-layered solution: the two-level inference engine consists of a meta-forward level for handling structural information of the network and a forward level to perform rule-based inference in order to determine the cause of the occurred fault.

The forward inference engine (PAPP et al., 1989) performs event driven real-time operation. The incorporated expert rules are described in the traditional *if* – *then* form and make automatic search of the fault tree possible: current parameter values are matched against the conditional (*if*) parts of the rules and if the conditional part of a rule is satisfied the activities in the conclusion part are executed.

Since relay stations may have different configuration it is possible to associate a rule-base to every station. However, if we take a deeper look at the problem it turns out that even a single relay station is too complicated to create efficient and relatively small rule-bases with acceptable efforts, not to mention the possible number of different rule-bases which can be several hundred (BAGÓ et al, 1986).

To overcome this problem we associated the lowest (rule-based) system level to the lowest (functional element) level of the network. Any station can be built using a very limited number of (only six) functional elements: receiver, transmitter and four types of multiplexers. Every functional element has its own, rather simple rule-base each of which holds less than 50 rules.

Rules can be categorized into two semantic classes:

- Advice generation rules are used to generate textual output for the operator about the actual failure causes;
- Fault propagation rules are the connections between individual rule-bases (i.e. functional elements). These provide information for the higher level inference engine to continue investigation on other functional elements by switching to their own rule-base.

When a fault occurs the meta-forward engine is started first. It automatically checks stations which have reported faults. First of all it determines the faulty station with a very simple algorithm: since the stations signal faults to each other, the station whose every neighbour reports fault is the cause of the network malfunction.

Every station has a tree shaped internal structure. After finding the faulty station the meta-forward engine determines the top level functional element of the station, activates the rule-base of it with the current set of fact-values and starts the forward inference engine.

If an advice generation rule is fired (i.e. the cause of the error is found), the inference procedure stops (we assume single faults).

If a fault propagation rule is fired, the meta-forward engine gets the control back and activates the rule-base of the next functional element which is determined using the so called meta-agenda. The meta-agenda is the main link between the meta-forward and the forward inference engine. Fault propagation rules put the identifier of the functional elements to be tested into the meta-agenda. The meta-forward engine takes the first item of it, switches to the actual rule-base and control is passed to the forward inference engine.

The Communication Subsystem

The communication subsystem ensures information flow between the microwave network and the supervisory system. The communication subsystem periodically polls the relay stations requesting the values of various signals characterising the state of the relays. In order to reduce the polling period messages contain only the change in the state of the interrogated relay station.

If a relay station fails it is taken out of the regular polling order and is requested to send information about its current state in the form of various parameters. In order to collect these parameters the communication subsystem sends remote commands to the relay stations. The communication subsystem passes data arriving from the microwave network to the database management subsystem.

The Database Management Subsystem

The database management subsystem assures the database consistency of the supervisory system. The database is stored on both computers of the supervisory system, providing partial operation of the system even if one of them breaks down.

In order to maintain consistent databases their items are tagged with a time stamp which is refreshed when the item is updated.

The architecture of this subsystem is based on the object oriented approach. Every relay station is represented by an object which stores information about the inner structure of the relay and its connections to other stations. These objects are built up of local objects, which represent the functional elements of the relay, e.g. multiplexers and radio frequency transmitters. Digital and analog signals of the functional elements are stored in the database in groups.

The database management subsystem stores actual values of the input signals in the computer memory, while old values are stored on the hard disk. The values stored on the disk represent the time records of the signals. A data-driven object oriented processing chain prepares input data in order to generate statistics from the incoming data, and activates the alarm system if needed (BAGÓ et al, 1986).

The Configuration Environment

The configuration environment of the supervisory system ensures easy mapping of the generic supervisory system to a dedicated communication network. It consists of a set of compilers integrated into a uniform multiwindow editor which makes definition editing and compilation interactive, fast and simple. The output of the compilers is a set of pictures to be displayed by the monitoring subsystem, the network database and the preprocessed rule-base which can directly be used by the advisory system.

To increase reliable operation of the compilers they were developed using the LEX lexical analyzer generator and YACC parser generator programs generally used in UNIX systems. However, to avoid symbol collisions

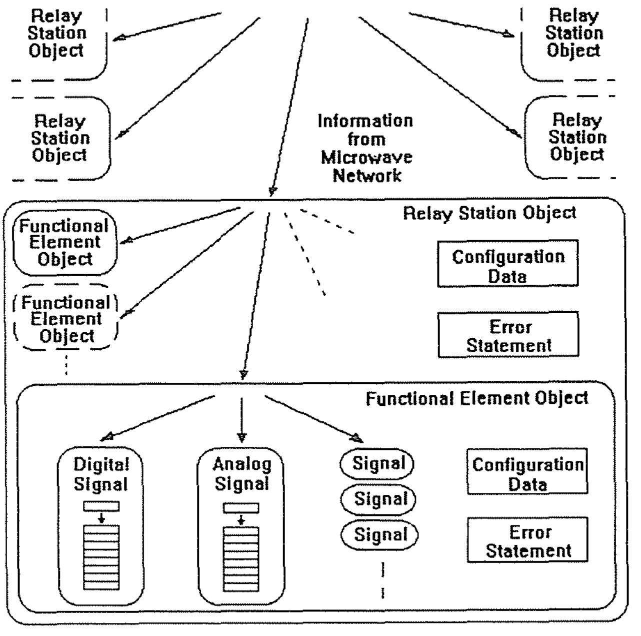

Fig. 4. The object oriented database

and the remarkable overhead caused by multiple inclusion in the final code of the same scanner and parser algorithms operating on different data certain modifications had to be introduced in the output of the generator programs. These changes made possible to include the scanner and parser driver routines only once; switching between different compilers was carried out as a switch from one set of syntax descriptor tables to another. Other major components of the compiler set are the unified code generator and symbol table handler modules.

The environment contains four compilers: a main-map, a submap, a relay description and a rule compiler. The first three generate pictures and data for the network database, while the last one translates the high level rules of the advisory system to a 'machine friendly' format.

To provide enough flexibility for future developments of the relays a configuration file containing the main features of possible functional elements is processed when the configuration environment is started.

Though the compilers stop translation when the first error is encountered, due to the usually small size of input files and the fast compilation it does not cause real inconvenience. On the other hand the compilers detect every possible error and provide detailed messages when necessary.

As an illustration of the pictures generated by the environment see *Figs 2* and *3*.

The Software Background

As a base for the supervisory system a generic expert system shell called REALEX was chosen (PAPP and colleagues, 1989; PAPP, 1990), which is a high level programming environment designed to be a versatile toolkit of system designers to build industrial process supervisory systems. This toolkit provides support to perform common supervisory tasks such as event logging, data acquisition, alarming, etc.

The REALEX shell gives only a skeleton to create process supervisory programs. From the system designer's point of view it provides slots to be filled with descriptions of particular processes. These slots reflect the different aspects of the process to be supervised, e.g. events to log, transformations to be performed on measured data, etc. However, the shell also provides some AI originated tools: the supervisory systems generated from it are able to perform post-mortem diagnosis of the supervised system in case of a crash. *Fig. 5* shows the simplified architecture of the REALEX shell.

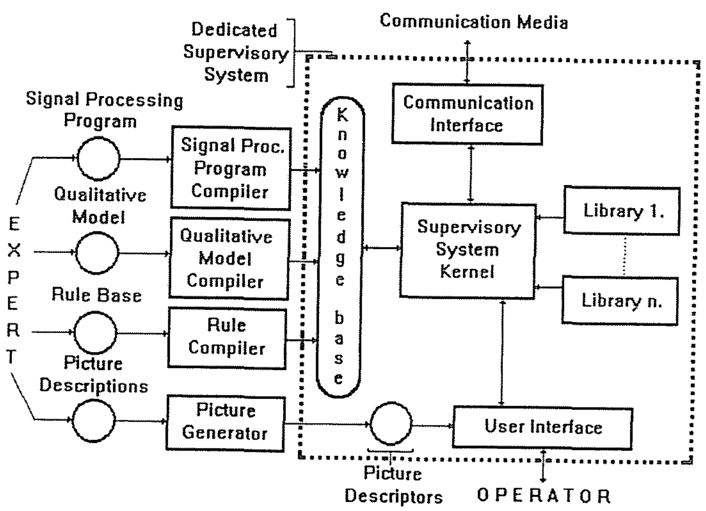

Fig. 5. Architecture of the REALEX shell

To provide adequate tools for system description the REALEX shell supports three different models of the supervised system.

The mathematical model (which in essence is a signal processing program) is based on the data-flow concept and is implemented on object oriented basis: the model of the supervised system can be constructed like chains of atomic transformations. The REALEX shell supplies a rich library of transformations, which can easily be extended to meet particular needs.

The heuristic (rule-based) model uses the well known rule-based paradigm. However, to ensure that the speed of inference does not depend on the number of rules a sophisticated preprocessing of them must be performed before the system starts operating (VADÁSZ, 1992). This preprocessing is performed by the rule compiler. Rules are interpreted in data-driven (forward chaining) manner. This tool was used to implement the fault diagnosis part of the telecommunications network supervisory system.

The third (qualitative) modelling feature of the REALEX shell was not used in the supervisory system for telecommunication network.

The REALEX shell also provides additional facilities which were widely exploited. These are the multiprocessing (including remote processes) and interprocess communication (IPC) features. Relays of the microwave network can be accessed as remote processes via the standard IPC mechanism of the REALEX shell. This IPC mechanism was designed to be capable to integrate various communication media to meet different application dependent needs.

The REALEX shell was designed and implemented at the Department of Measurement and Instrumentation Engineering of the Technical University of Budapest. It was written in C language, although the shell also provides a LISP interface for the users. This software architecture was chosen for and proved to be very efficient in integrating symbolic and numerical computation. The shell runs on IBM PC under MS–DOS, OS/2 and MS WINDOWS 3.0. operating system.

The dedicated software of the telecommunications network supervisory system was mostly written in LISP, except for some time-critical parts. In order to meet some special requirement of this application a communication loop as a new communication media was added to the module managing the inter-process communication. For the same reason the user interface was re-designed, although the original picture handling concept of REALEX was fully adopted. The dedicated microwave supervisory system and its configuration environment runs on two IBM PCs under MS WINDOWS 3.0.

Conclusions

Supervision of telecommunication networks needs supervisory systems of high reliability and flexibility. A simple dedicated system is not acceptable because of the features of the application domain. The development work described above in detail resulted in a versatile supervisory system for telecommunication networks. It is easily configurable for different network structures and relay types. Exploiting possibilities of the underlying REALEX shell the system successfully combines the traditional data acquisition with the knowledge based approach. This solution supports the operator with high level information preventing him or her being overloaded by the data in case of a fault in the supervised network. Moreover, it reduces drastically the overall cost of repairing.

This work has shown that the rule-based paradigm can successfully be used in the field of supervision and control of complex processes.

References

BAGÓ, B. et al. (1986): A Multi-level Signal Processing System. *Proceedings of the 8th Annual Conference of the IEEE-EMBS*, Vol. 2, pp. 825–828.

DAVIS, R. (1985): Diagnostic Reasoning Based on Structure and Behaviour. Qualitative Reasoning About Physical Systems, MIT Press, Cambridge, Mass. pp. 347–410.

PAPP, Z. et al. (1989): Expert System Architecture for Real-time Process Supervisor Applications. In G. Rzevski (Ed.), Artificial Intelligence in Manufacturing, CMP & Springer-Verlag, Berlin. pp. 223–240.

PAPP, Z. (1990): Programming Tool for Integrating Numerical and Knowledge Based Signal Processing Techniques. *International Symposium on Knowledge Based Measurement — Application, Research and Education*. 1990, Karlsruhe, pp. 181–188.

WINOGRAD, T. (1980): Extended Inference Modes in Reasoning by Computer Systems. *Artificial Intelligence*, Vol. 13, No. 1–2, pp. 5–26.