Knowledge-Based Control Systems via Internet Part I. Applications in Biotechnology

Svetla Vassileva*[1], Georgi Georgiev[1], Silvia Mileva[2]

[1]*Institute of Control and System Research - Bulgarian Academy of Sciences*
Bl. 2, Akad. G. Bontchev Str., 1113 Sofia, Bulgaria

[2]*Institute of Criobiology and Food Technologies, 53 Tcherni vrah blvd., 1407 Sofia, Bulgaria*
E-mail: vasileva@icsr.bas.bg g.georgiev@icsr.bas.bg silviadimitrova@abv.bg
* Corresponding author

Received: August 4, 2004 **Accepted: September 20, 2004**

Published: April 28, 2005

Abstract: *An extensive approach towards the dissemination of expert knowledge and coordination efforts to distributed points and seamless integration of control strategies applied to distributed yet identical systems is crucial to enhance overall efficiency and operational costs. Application of Knowledge-Based Control System via Internet will be very efficient especially in biotechnology, because many industrial bioprocesses, based on the same technological principles, are distributed in the whole world. Brewing industry oriented practical solutions illustrate this approach.*

Keywords: *Knowledge-based control system via Internet, Remote control, Biotechnology, Brewing industry*

Introduction

Industrial plant is a complex system comprised of subsystems, objects and connections, which are divided in different groups according certain criteria. The mostly used criterion is main subject of plant manufacturing. In this connection manufacturing processes are considered as core and support processes. Whenever core processes play a key role, its realization requires a successful support as well as external services often named outsourcing processes.

The entire plant layout can be rather extensive, spreading across continents in certain cases. The same set of processes used to manufacture an identical product or to monitor the same process can be branched over different plants. Close coordination and synchronization of the distributed operations is required, as well as having an efficient remote monitoring and control facility in place.

An extensive approach towards the dissemination of expert knowledge and coordination efforts to distributed points and seamless integration of control strategies applied to distributed yet identical systems is crucial to enhance overall efficiency and operational costs.
Application of Knowledge-Based Control System via Internet will be very efficient especially in biotechnology, because many industrial bioprocesses, based on the same technology, are distributed in the whole world.
The work in this paper is motivated by this observation for Knowledge-Based Control System design via Internet.

Hierarchical structure of KBCS via Internet

In general, Knowledge-Based Control Systems (KBCS) via Internet coordinate function of four levels of a plant:

1. Control of a low level technological unit – device, machinery
2. Control of a technological process
3. Control of some technological processes with a final industrial product
4. Plant control including economic criteria fulfillment.

A corresponding hierarchical structure of the control system is required for this reason.
Natural categorization of the plant-processes is carried out according its content or main goal as follows:

1. Manufacturing activities, which involve processes supporting main production as well as sources planning, computer-based design support and production preparation, PC-integrated manufacturing, etc.
2. Plant management, which involves plant control, quality control, economic analyses, strategic planing, portfolio evaluation of products and services, decision support systems, etc.
3. Administrative services which include processes of the automation of the administrative functions, archives, document digitalization and visualization, text processing and recognition, teleconferences and planing support, etc.
4. Economics, finances and book-keeping
5. External connections as well as marketing, clients, distributors and production sources provider - processes connected with the market investigations, advertising campaign organization and evaluation, analytical investigations of the operative contact with the clients, care for the clients, etc.
6. Staff policy
7. Infrastructure – processes which ensure technical equipment, servers, front-end client computers, net infrastructure, help-desk, etc.
8. Development of plant culture and know-how – processes, oriented to the plant-internal-culture improvement, library of the type solutions, “best-practices”, etc.

The different speed of data processing, time-response and quantity of data processed characterize separate levels. Low levels data processing asks for fast processing of smallest data quantity (some bytes or several tens of bytes in some millisecond’s) in comparison with the highest levels, where the data quantity is of kilo- or megabytes and time response is expected in minutes, hours or days. These particularities define the technological parameters of the KBCS via Internet as well as computing power, capacity of the memory, etc.

A Knowledge-Based Control System (KBCS) is comprised of a central intelligent controller, implemented on a workstation or a high performance personal computer (PC) with a knowledge-base, instrumented via a digital bus (such as the General Purpose Instrumentation Bus) to front-end microprocessor–based controllers [4, 6]. We will refer to this central controller as the Expert Controller henceforth in the paper. The front-end controllers gather real-time information from the system and transmit it to the Expert Controller. Through the Knowledge Base the Expert Controller will execute the inference procedures to determine the best course of action. Commands will be transmitted to the front-end devices, which will carry out the control actions. These front-end devices are usually dedicated programmable logic controllers (PLCs) or controller boards that are inexpensive and not easily configurable. Clearly, the main intelligence lies in the Expert Controller, which is usually also the most costly and sophisticated component of the overall system. The maximum operating distance from the Expert Controller to the front-end devices depends on the type of bus used as well as the amount of noise interference affecting signal transmission [3]. Using a serial bus (instead of a parallel bus), this distance can be increased, albeit still within the local vicinity of the plant [5].

KBCS via Internet are supposed to work in the real-time conditions. In this connection three cases could be considered. The function under the “absolute” or astronomic time with a high accuracy (for example of 0,1 s) is assured by the synchronization through the external radio signal for the subsystems function coordination, etc. The function in enough short time (fast response) is implemented for the task solving in one discrete period of time or in one cycle of the control algorithm. Deterministic function time is an extension of the former case when it is necessary to distinguish if KBCS succeed to react in the determined time for response.
Concerning the real-time work the KBCS via Internet are divided in two main groups. Hard real time KBCS via Internet require a strict observation of time conditions, because it leads to the crucial troubles as well as function break down or failure. A non-strict observation of the time conditions in soft real time KBCS via Internet cause low accuracy or short-time failure without following problems.

Architecture of the KBCS via Internet

KBCS are usually intelligent computer-controlled systems that use knowledge and inference procedures to solve problems difficult enough to require significant human expertise [1, 2].
Hierarchical structure of the KBCS via Internet consist of a high level server, connected with the computers of the clients, distributors and production sources providers and an Expert Controller that coordinates the lowest levels (Fig. 1).

The Expert Controller resides on a workstation or a high performance PC, connected through the instrumentation bus to the front-end microprocessor-based local controllers. The Expert Controller synchronizes the front-end controllers of the technological processes, devices and machinery. The remote front-end controllers are responsible for the real-time algorithmic fuzzy, NN or PID-control.

The hardware architecture of an Expert Controller is shown in Fig. 2. The Knowledge Base (Data and Rule Base) and Inference Mechanism are located in the Expert Controller. The real-time data are collected by the front-end microprocessor-based controllers, transmitted after that by the Internet for monitoring, gathering and processing in the Expert Controller. It allows the inference mechanism to take the required decisions when the analysis of the collected data is completed. The Expert Controller decisions are sent via the Internet to update the front-end controllers.

Software architecture of the KBCS via Internet

The software of the Internet-based remote control technologies covers three main areas [7]:

1. Communication protocols (HTTPS, SSH}
2. www and hypertext pages (HTML, DHTML – dynamic HTML, ActiveX, XML or the mobile phones implemented WAP and GPS.
3. Programming languages – object-oriented languages as well as Perl, PHP, Java, JavaScript; database supporting languages as MySQL, PostgreSQL, Oracle, etc.

In connection with the server supporting software it is very important to know the type of the servers, which are: data collecting server, application server, communication server, www-server, ftp-server, proxy-server, mail-server, fax-server, etc.
Recent Internet-technologies are client-server-based platforms or some kind of the agent systems. Usually the server maintains some program (servers) for the specific tasks solving. The agent carries out client inquiry, which can not be handled by the server, by a migration in the net and checking the rest part of available servers.

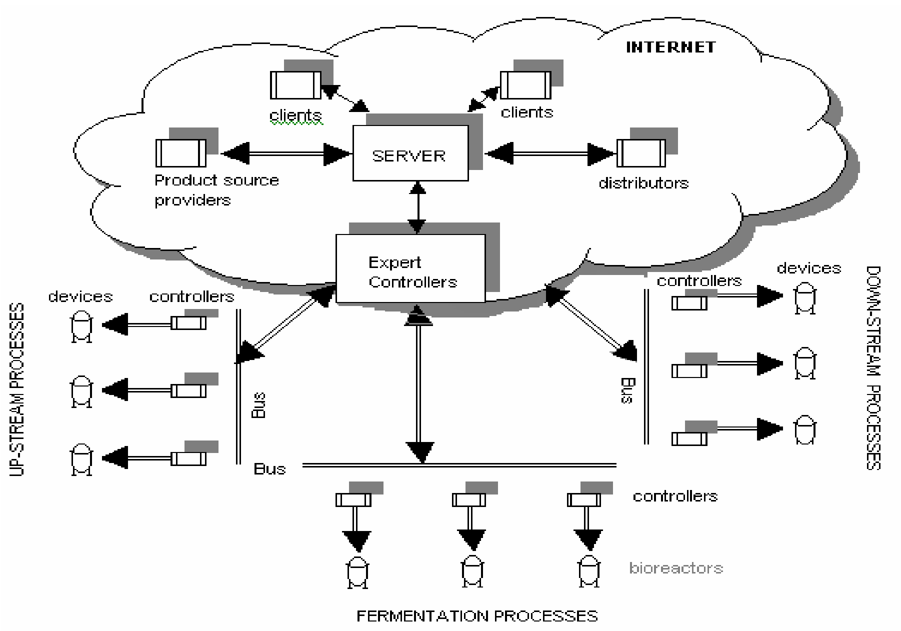

Fig. 1 Hierarchical structure of the KBCS via Internet

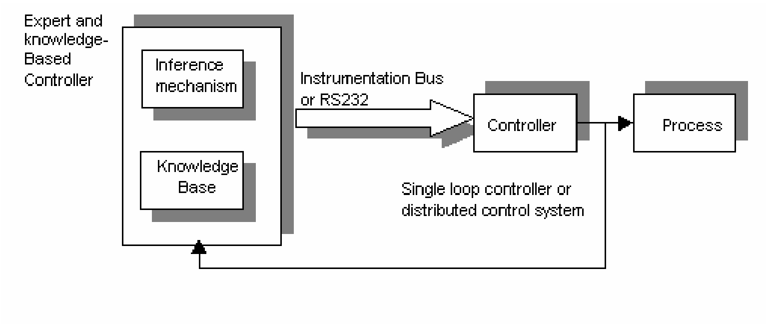

Fig. 2 Expert Controller architecture

Wide opportunities are achieved by the Microsoft platform .NET, based on the Internet standards as well as HTTP (communication protocol), XML (data exchange), SOAP (standard format for web service requirements), UDDI (standard for web-service look up). The .NET platform is language independent and now supports C++, C#, VisualBasic, JScript and COBOL.

The Expert Controller uses a web page with Datasocket reader components that connect to the Datasocket server, and which read the system information from the front-end local controllers. Datasocket is a programming tool that enables the user to read, write and share data between applications and/or different data sources and targets. Datasocket can access in local files and data on the Hypertext Transfer protocol (HTTP) and the File Transfer protocol (FTP), on the universal resource locator (URL) for connection to the data source location, on OLE for process control (OPC) and on Datasocket Transfer Protocol (DSTP) for sharing live data through the Datasocket server. All that shows a complicated and overloaded function of a single Expert Controller. To distribute the workload two or more Expert Controllers can be implemented. All expert and front-end local controllers will be connected through the Datasocket server.

Application Example: KBCS via Internet for beer manufacturing

A plant-wide "virtual brewer" KBCS – concept is proposed (Fig. 3), using a network of interacting intelligent systems for data acquisition, assessment, interpretation, decision support and control, to provide efficiency savings and to assure consistent quality of final product. These intelligent systems are based on quantitative information, rules-of-thumb and tacit knowledge, gathered in the practice. The KBCS main goal is to retain and renew intellectual capital in the beer manufacturing.

KBCS configuration

High level processes coordinates a high level server, connected with the computers of the clients, distributors and production sources providers and an Expert Controller that coordinates the lowest levels (Fig. 1).

Main functions of KBCS high-level are control of some technological processes with a final industrial product and plant control including economic criteria fulfillment.
Control of a technological process is main function of KBCS low-level (Fig. 3). Three groups of low-level technological processes are distinguished in biotechnological manufacturing, as well as in the beer manufacturing: up-stream processes, fermentation stage and down-stream processes. Each of them can be considered in connection with the process monitoring and control and with the technological equipment diagnostics. In brewing manufacturing the processes groups are as follows:

- Up-stream processes, connected with the malt processing, wort complex, starting yeast culture preparation on one hand and with the up-stream devices engineering from the other hand.
- Fermentation processes as a main stage in beer manufacturing, connected with yeast growth and some important technological variables of fermentation monitoring, biosensor data processing and storage, final product quality prediction, diagnostics of the cylindro-conical fermenting tanks and sensor, fault detection.
- Down-stream processes, which involve storage of cultural medium, additives and filtration processes, bright beer processing, beer quality rating and packaging on one hand and the down-stream technological equipment monitoring on the other hand.

Operational principles of a KBCS low-level

A cylindro-conical fermenting tank is considered for illustration of the KBCS low-level operational principles with KBCS software components in Fig. 3.

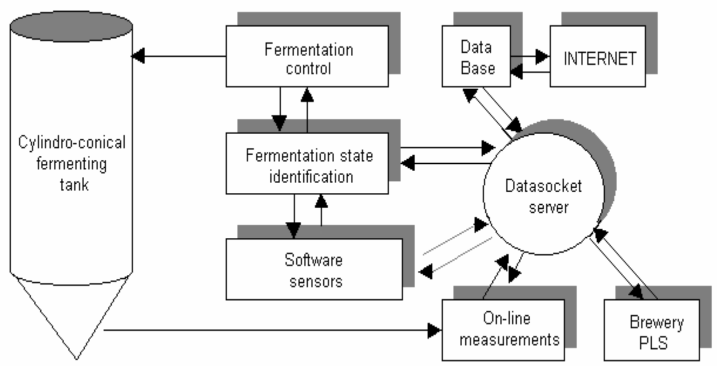

Fig. 3 Cylindro-conical fermenting tank with KBCS software components

Communications between Expert Controller and single loop front-end local controller of a distributed control system of a fermentation process, are realized by messages exchange by an instrumentation bus or RS232, as is illustrated in Fig. 2. The message structure is consistent with the type of the used microprocessor. A possible realization by implementing microprocessors and bus RS232 require the messages include data blocks with different length. The structure of information words is shown on Fig. 4.

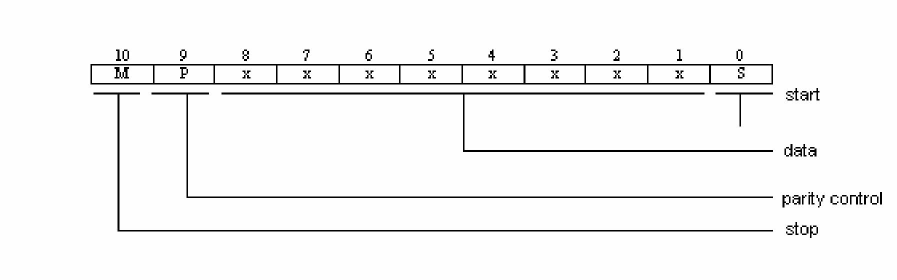

Fig. 4 Information word - structure

Kind of information words

Command words

- command-address word – UKAZx (Fig. 5)

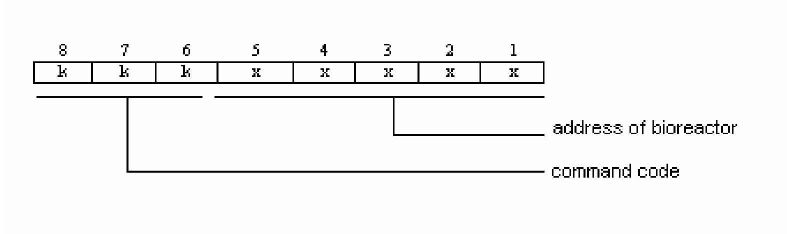

Fig. 5 Command-address word

The meanings of the bits are:

k k k
----
0 0 0 - setting of command points
0 0 1 - changing the mode of operation
0 1 0 - telecommand execution
1 0 0 - basic mode of operation
1 1 0 - initialization

- parameter choice word – IZBP (Fig. 6)

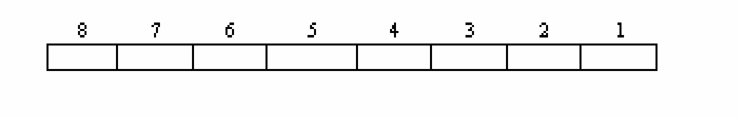

Fig. 6 Parameter choice word

The word is used for definition of the values of the parameters. It is possible to use several words depending on the parameter m, defined in the choice word (IZB).
This word is used for changing mode of operation, setting parameters and repetition mode as follows:

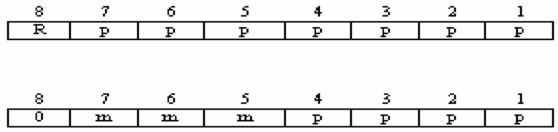

Fig. 7 Choice word

a) If it is necessary to change the mode of operation R must be set to 0, the remaining 7 bits define the regime, in which the bioreactor works;

b) Setting parameters – m specifies the number of the words for parameter choice, and p specifies the parameters;

c) Repetition mode – R=0, p specifies the block that must be send over again. If p=0, the block is not be send again, if p=1, the block must be send again.

Address words

- word for address of information point –ADRT (Fig. 8)

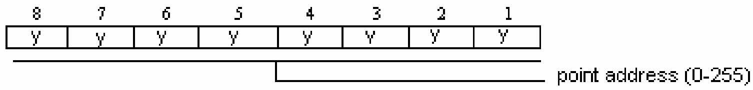

Fig. 8 Word for address of information point

Specifies the output must be turn-on/turn-off for telecommand execution. The controller checks the correctness of the selected output.

- word for address of data – ADRI (Fig. 9)

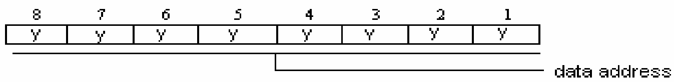

Fig. 9 Word for address of data

It contents the relative data address in the data blocks. One can address up to 256 bytes data of various types (2048 one-bit signalizations.).

Data words – S

- single-bit signalizations – S1 (Fig. 10)

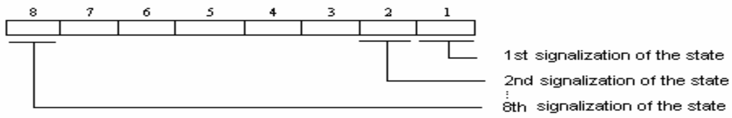

Fig. 10 Single-bit signalizations

Single-bit signalizations are grouped by 8 and are transferred as a word. Each one of the bits gives the state of the appropriate input.

- double-bit signalizations – S2 (Fig. 11)

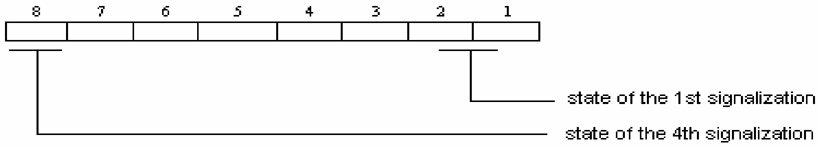

Fig. 11 Double-bit signalizations

Double-bit signalizations are grouped by 4 and are transferred like a word.

- changes of double/single bit signalizations - ΔS1,ΔS2 (Fig. 12)

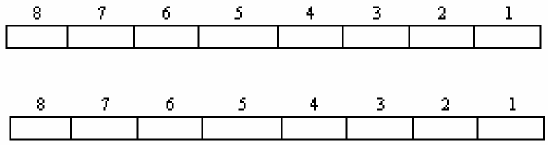

Fig. 12 Changes of double/single bit signalizations

The local controller detects the changes of the signalizations transferred at the Expert Controller. The local controller sends relative addresses of the single/double bit signalization and their values.

Word for block description OPIS

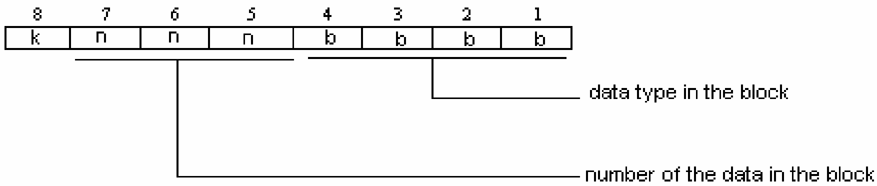

Fig. 13 Block description by a word OPIS

It is used in the local controller's response for data definition and the block's length. A block can contain only one type of data.

- b - type of data. the following types are possible (up to 16)
  single-bit signalizations - S1;
  double-bit signalizations - S2;
  single-bit signalizations change - ΔS1;
  double-bit signalizations change - ΔS2;
  test-diagnostic data;
  error identification etc.
- n - number of data bytes in a block - from 1 to 8 bytes
- k=1 - last block in the response; k=0 - the block is not last one

Word for parity check VAR

Each block ends with a parity check word – VAR.

Fermenting tank state identification

Communication "Expert controller-local controller" for cylindro-conical fermenting tank state identification (according Fig. 1 and. previous part 5.3) is realized by the data acquisition procedure. This procedure involves data gathering in basic mode, and initialization or repetition mode (when local controllers change their mode of operation). For each mode of operation the blocks number in reply are set. In initialization mode all necessary messages for cylindro-conical fermenting tank state update are sent. A block may consist up to 8 bytes. In local controller configuration the priority of data is also set.

Basic mode command

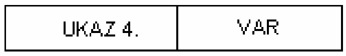

Fig. 14 Basic mode command

The local controller can respond by one of the following three ways:

a) If the local controller does not detect changes in the states between two control commands, it sends telegram with updated data, consisting only of a block (Fig. 15).

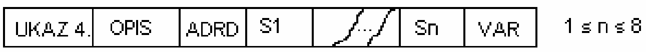

Fig. 15 Telegram with updated data

b) If the local controller has detected changes in the states, the telegram has been sent is composed of blocks, containing changes (up to 6 data blocks and a block with updated data) (Fig. 16);

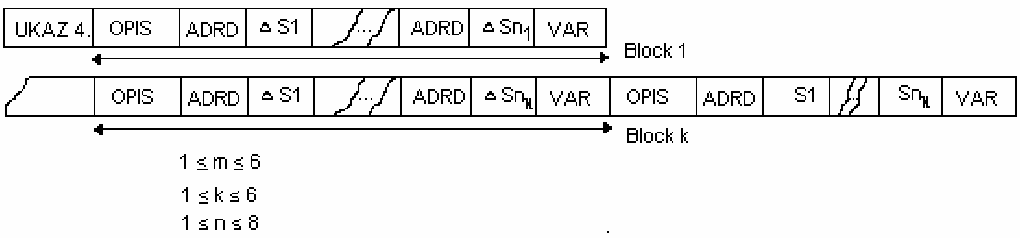

Fig. 16 Telegram containing block changes

c) If the changes are more than 6 blocks or because of fault the local controller has not worked between two commands, the local controller automatically goes in initialization mode and sends the telegram with the following structure (Fig. 17).

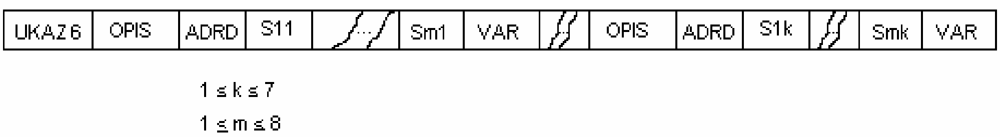

Fig. 17 Initialization mode telegram

Local controller mode change

The local controller mode change command is composed of three blocks as is shown in Fig. 18:

Fig. 18 Control command for operation on the local controller mode change

The local controller replies by two ways:
a) If the specified mode (word IZB) has a higher priority in comparison with a current mode, the local controller goes in the specified mode and the telegram is composed of blocks (from 1 to 7) with data for the specified mode (Fig. 19).

Fig. 19 Replay telegram, containing data of the specified mode

If a mode of repetition is specified, the telegram contents incorrect received blocks.
After all data of specified mode are sent, the local controller automatically goes in basic mode.

b) If between two control commands the local controller has not been worked, the telegram contains reply for initialization.

Expert Controller sends a control command for local controller's initialization at the system startup or if the local controller has not been worked because of damage. The reply consists of two telegrams, sending one after one. The first one consists of blocks that gives the state of local controllers (Fig. 20).

Fig. 20 Telegram of front-end local controllers state

The second one consists of the information about the local controller efficiency – test mode (Fig. 21).

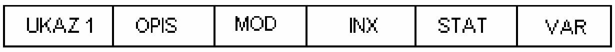

Fig. 21 Telegram of front-end local controller efficiency – test mode

MOD – Type of the defected local controller
INX – Index of the defected local controller
STAT – Information about the detected fault.

If there are no faults, MOD, INX and STAT are zero.

Telecommand execution

Communication between Expert Controller and local controllers for a telecommand execution is realized by telegram replies of local controller. Each telecommand is executing in two steps:

a) The Expert Controller sends control command for setting of command points (Fig. 22)

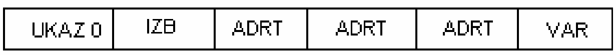

Fig. 22 Command for command points setting

In the word IZB the number of command points is set, depending of the type of the telecommand. Local controller replies (Fig. 23):

UKAZ 0	ADRT	ADRT	ADRT

Fig. 23 Local controller replay on a command for command points setting

The words IZB and VAR are not sending, because the expert controller checks the correctness of every point.

b) After the Expert Controller determines, that chosen outputs are equal with the received addresses of the points, it sends (Fig. 24):

UKAZ 2	VAR

Fig. 24 Telegram after detecting equality state between chosen outputs and the received addresses of the points

The local controller checks if there are activated outputs and carries out the appropriate tests. If the tests are successful, the appropriate outputs are activated and command UKAZ2 be sent. If the tests are not successful, the local controller sends telegram with UKAZ2, VAR in which the address of the local controller is null. That means, that there is a fault and the command could not be executed. After sending a telegram-response on UKAZ2, local controllers turn on basic mode of operation.

Fault detection

At each operation mode the local controller performs a wide variety of tests concerning the controller correct functioning. If a fault occurs, the local controller sends the following telegram (after receiving the next control command) (Fig. 25):

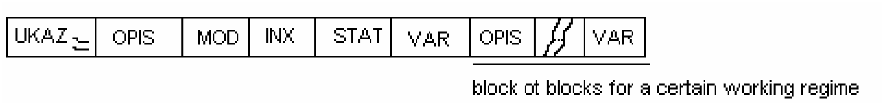

Fig. 25 Local controller fault detection telegram

If the faults are more (the maximal number in a telegram is 4), the words MOD, INX and STAT are repeated so many times as the number of the faults.

The communications between expert controller and local controllers is time dependent. The maximum time between sending of two-control command is setting. If this time expires and expert controller did not receive telegram-response from local controller, expert controller repeats sending of the tele-command.

At local controller configuration the maximum time between sending of two telegrams and maximum time between two consecutive words in a telegram are setting. If any of the times expires, local controller automatically goes in a test mode.

Conclusions

In practice, the fermentation and maturation of beer is mainly monitored only by periodical manual analysis of gravity before and of diacetyl after degradation of sugars. The necessary evaluation and beer quality prediction requires personnel and laboratory capacities and causes delay of some hours between measurements and appropriate reaction. On-line measurement systems are not installed in the fermentation cellars in a large scale. Thus, only a few measurement points predominantly at the end the main fermentation are accomplished, using standard off-line methods.

For an active control the observability of the beer fermentation process, i.e. the on-line measurement of the relevant quantities, is a prerequisite. Regarding the lack of existing technically applicable solutions for the observation and event driven control of the beer fermentation, simultaneously considering the large variability of the process trajectories, depending on yeast strain or technological variations, implementation of KBCS is a recent engineering solution, which uses efficiently the expert knowledge in brewing. The proposed approach allows predicting the final product quality depending on the quality of each stage of the manufacturing process. The simulation results will be shown in the Part II of this paper.

The expansion of KBCS-applications through the Internet allows a wide implementation and exchange of entire expert knowledge in the globalizing world. The principles of such a system could be implement in different fields of the human activity.

Acknowledgment

*This work was partially supported by Grant № I-1406/2004 from the National Fund for Scientific Investigations, Bulgarian Ministry of Education and Scienc*e.

References

1. Barr A. and Feigenbaum (1981). The Handbook of artificial Intelligence, v.1, Pittman, London, E.A.Eds.
2. Barr A. and Feigenbaum (1982). The Handbook of artificial Intelligence, v.2, Pittman, London, E.A.Eds.
3. Berge J. (2002). Fieldbusses for Process Control, the Instrumentation, Systems and Automation Society.
4. De Silva, C.W. and A.G.J. Mac Farlane (1989). Knowledge-Based Control Approach for Robotic Manipulators, Intern.Journal of Control, 50, 1, 249-273.
5. Jordan J.R. (1995). Serial Networked Field Instrumentation, Chichester, W.Sussex, Eng. New York: Wiley.
6. Samad T. (2001). Perspectives in Control Engineering: Technologies, Applications and New Directions. New York: IEEE Press.
7. Van Eck P., J.Engelfriet, D.Fensel, F.Van Harmelen, Yde Venema and M.Willems (2001). A Survey of Languages for Specifying Dynamics: A Knowledge-Engineering perspective. IEEE Trans. On Knowledge and Data Eng., 13, 3, 462-496.