Introduction to Theory of Real-Time Micro-Operating Systems for Embedded Control Systems

Astapkovitch A.M.

Head of Student Design Center, State University of Aerospace Instrumentation
Bolshaij Morskaij, 67, 190000, Saint-Petersburg,Russia
Email: skb@aanet.ru
http://www.guap.ru

Abstract

The review of the actual state of Real-time micro-Operating Systems theory is presented. This applied theory is focused on subclass of embedded multi-channel control systems for hard real-time applications and oriented to be a theoretical base for next-generation software integrated development environments (IDE). Article concerns hierarchy of programming methods, probabilistic behavior of real-time control systems and new approaches for describing algorithms of multi-channel real-time control system functioning with AT-map formalism. Initial objects of theory are introduced. It's shown that response time for multi-channel real-time systems has to be interpreted as probabilistic value. A new formalism provides possibilities for two-dimensioned algorithm description with address-time maps (AT-map). Classification of conflict in concurrent space was introduced: AT-conflict, A-conflict, T-conflict. Formal criterion was introduced to provide possibilities for automatic solution of critical section analysis problem. Basic definitions for AT-formalism operations were introduced. Possible structure of next generation IDE is discussed.

I. INTRODUCTION

Real-time micro-operating systems (RTmOS) theory was first time presented in [1]. This theory is focused on subclass of embedded multi-channel control systems for hard real-time applications. The structure of theory is illustrated on fig. I.1.

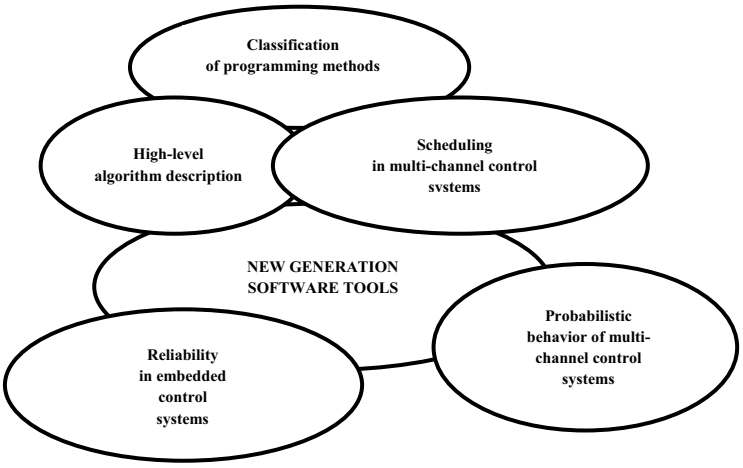


Figure 1.1 RTmOS theory structure

Typical structure of multi-channel control system is represented on Fig. 1.2.

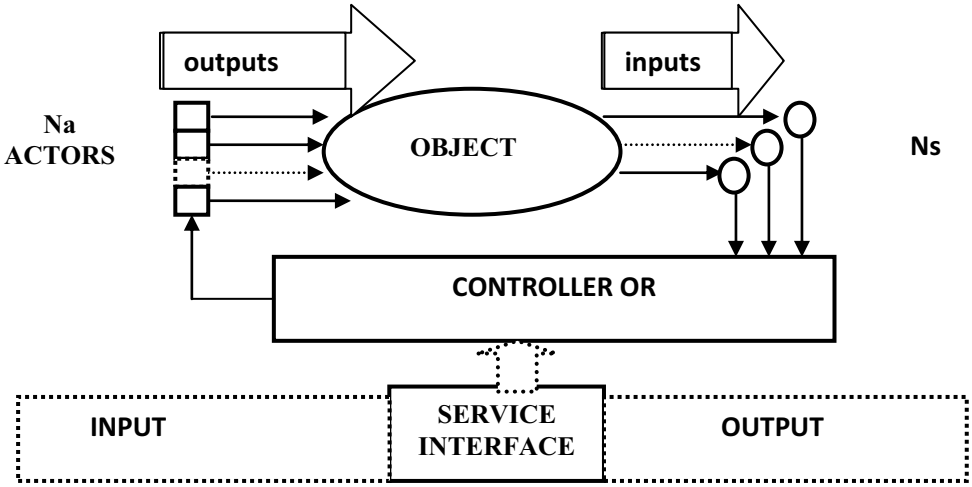


Figure 1.2 Typical structure of a control system

Classical theory defines Real Time Control System (**RTCS**) as a system that provides response on object state changing during time $\mathbf{Tr_i}$ that not exceeds specified time $\mathbf{Tsp_i}$ for every controlled channel.

$$Tr_i < Tsp_i \qquad i=1\ldots Nch \tag{1.1}$$

For specified controller's performance number of controlled channels is a complex function of different factors

$$Nch = f(Ns, Na, Tr, \ldots) \tag{1.2}$$

RTCS class can be divided into two subclasses: a subclass of control systems for Soft Real Time Application (SRTA) and subclass of control systems for Hard Real Time Application (HRTA). HRTA subclass consists of very reliable control systems, because it is supposed that every fault of control system of this subclass leads to loose of control object.

As applied to control software it has to be outlined that for HRTA control systems fault can occur if mistakes are exist in control algorithm for some channel or if specified response time is exceeded due to interference between channels.

To reduce probability of mistakes for real time applications a special programming technology has to be used. Hierarchy of software developing methods for embedded control systems was proposed in [1]. It is represented on fig. 1.3.

It can be stated that RTmOS is a key element of modern software development technology. It can be concluded, also, that today the most popular technology is down-up approach to software developing.

The main goal of the proposed theory is attempt to create a theory base for using up-down approach to programming in next generation IDE or at least to established understanding what has to be done in this directions.

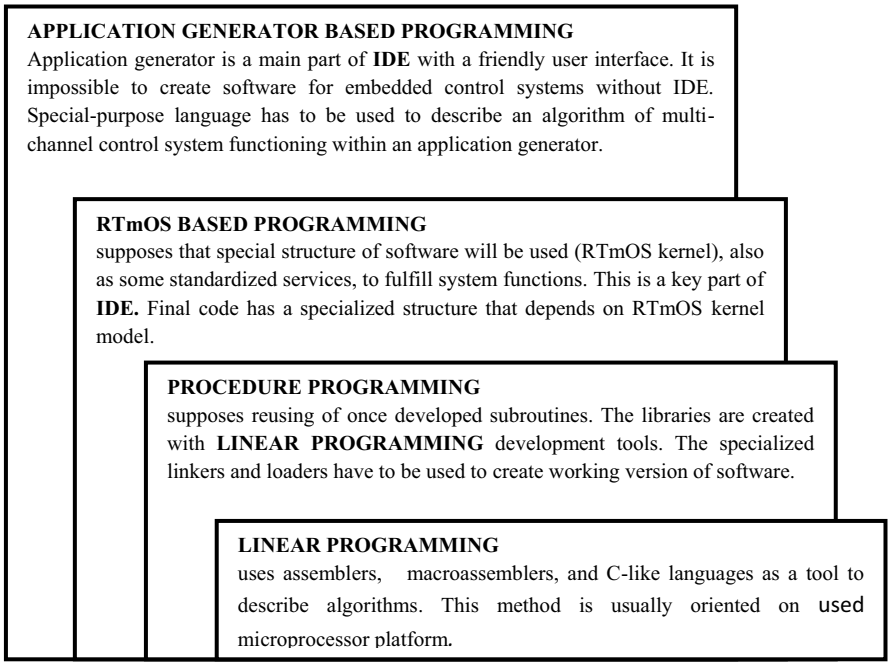


Figure 1.3 Hierarchy of modern software developing methods for embedded control systems

It was pointed out in [1] that embedded control systems has some special features as:

- known number of control channels;
- well determined control algorithms for every channel;
- this system features are fixed during system design phase;
- structure and properties of the software depend on the used RTmOS kernel features.

In accordance with definition, for multi channel real time applications, resulting software has to fulfill specified time requirements, that is expressed by (1.1). It is also as formal logic requirements to algorithm for every control channel. RTCS final software code is a set of RTmOS kernel code and user functions code. It is produced with some kind of application generator.

An application generator for multi-channel real-time control systems has to provide possibility for using of library functions: OS kernel, system-level software (drivers and system services) and user application software. Application generator based programming leads to use some high-level algorithm description. This description has to include information about necessary addresses and time resources (a 2-D algorithm description) to implement control algorithm for every channel. Next generation IDE has to be based on up-down programming automatic critical section analysis;

Technology and, at least, has to provide following possibilities for user:

- automatic analysis of possibility to fulfill prescribed time specifications;
- automatic analysis for dead lock situations.

To solve this task a theoretical base has to be established that will provide possibility to create a new generation Integrated Development Environments (IDE), oriented on multi channel RTCS. Theory of RTmOS is very young, so this report has a goal to give a brief review of a present status and concentrated on some fresh ideas.

II. INITIAL OBJECTS OF RTmOS THEORY

RTmOS theory uses some initial objects described below.

Task and Composed Task. Task is functionally complete code that responsible for some function and implemented as a library text macro or as a subroutine. In scheduling theory the initial object "job" is used as an analog of "task". Some tasks can be joined to sequence to perform some complex function. Resulting object is defined as a composed task. In this case an inter-task information exchange will exist.

Process. Process is a run-time realization of algorithms described with tasks or composed tasks. Process can be cyclic or asynchronous. Cyclic process is formed when executing the same task or composed task sequence in infinite cycle. Asynchronous process is formed when executing task sequence is not repeated exactly.

Procesogramma and Cyclogramma. Processogramma and cyclogramma are graphical representations of processes' time-base. As it can be seen from fig. 2.1, processogramma shows time-base of process on limited duration. It is useful to describe fragments of a process. Cyclogramma describes a cyclic process.

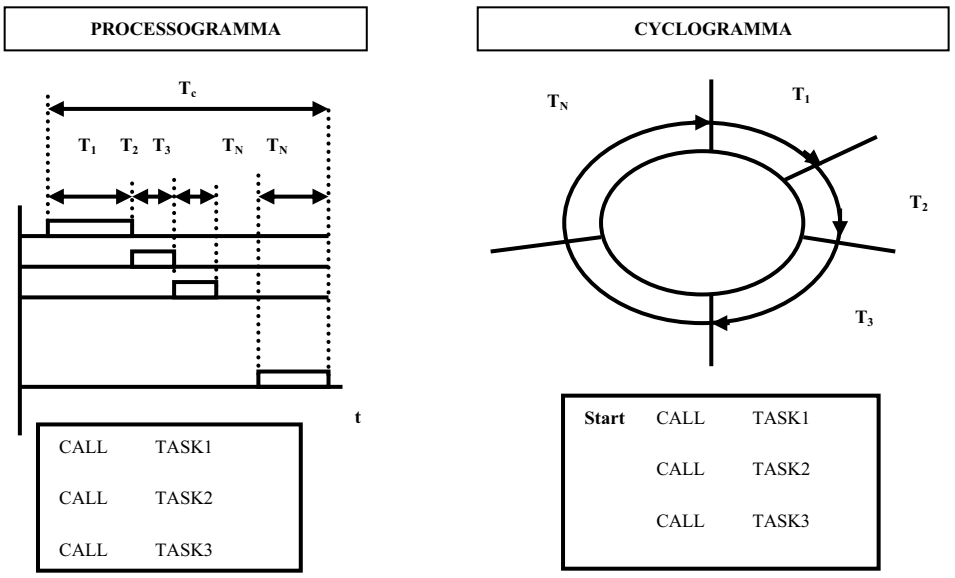


Figure 2.1. "Processogramma" and "Cyclogramma" illustration

Pseudo parallel processing. In typical multi channel RT-control system one microcontroller has to handle a set of input signals and actors. To perform control algorithm for every pseudo-parallel processing channel of RT-control system it is necessary to distribute microcontroller resources between processes responsible for each channel. It means that processes compete for microcontroller resources always, because at least microcontroller kernel is used in pseudo parallel processing mode. All task need some memory addresses and

improper memory usage is a source of mistakes. This problem is known as critical section analysis. Up today programmer solves this problem by hand (and brain of course).

AT-map. In accordance with AT-model [1, 2, 3] processes compete for resources in global concurrent region, that for RTCS has, at least, two dimensions (address and time). AT-map is a formal description of necessary resources (memory addresses and time slots) to perform task execution (i.e. processes' flow), represented as a rectangular matrix. For AT-model address axis is formed from all microcontroller resources (memory cells, pin-outs, OS objects, etc.), that are numbered in some way. Time axis is divided into intervals (seconds or command cycles) that are numbered also.

Elements of AT-map can be bit (for binary AT-map) or integer (for byte AT-map) values. Binary AT-map is formed according to the next rule: **if task uses I-th address during J-th time slot, element (I, J) of matrix is equal to 1, otherwise it equals to 0.**

Examples of AT-maps are represented on fig. 2.2. AT-map with integer and real elements can be used for more sophisticated purposes. ***Pseudo parallel processing***. In typical multi channel RT-control system one microcontroller has to handle a set of input signals and actors. To perform control algorithm for every pseudo-parallel processing channel of RT-control system it is necessary to distribute microcontroller resources between processes responsible for each channel. It means that processes compete for microcontroller resources always, because at least microcontroller kernel is used in pseudo parallel processing mode. All task need some memory addresses and improper memory usage is a source of mistakes. This problem is known as critical section analysis. Up today programmer solves this problem by hand (and brain of course).

AT-map. In accordance with AT-model [1, 2, 3] processes compete for resources in global concurrent region, that for RTCS has, at least, two dimensions (address and time). AT-map is a formal description of necessary resources (memory addresses and time slots) to perform task execution (i.e. processes' flow), represented as a rectangular matrix. For AT-model address axis is formed from all microcontroller resources (memory cells, pin-outs, OS objects, etc.), that are numbered in some way. Time axis is divided into intervals (seconds or command cycles) that are numbered also.

Elements of AT-map can be bit (for binary AT-map) or integer (for byte AT-map) values. Binary AT-map is formed according to the next rule: **if task uses I-th address during J-th time slot, element (I, J) of matrix is equal to 1, otherwise it equals to 0.**

Examples of AT-maps are represented on fig. 2.2. AT-map with integer and real elements can be used for more sophisticated purposes.

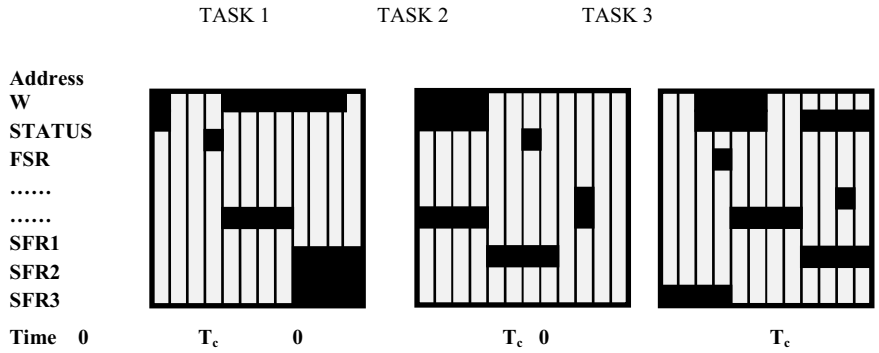


Figure 2.2 Examples of AT-maps W, STATUS, FSR, SFRx registers has unique numbers.

III. CRITICAL SECTION ANALYSIS PROBLEM

Automatic solution of critical section analysis problem is a key element of up-down software developing technology. AT-map was proposed as a tool for description of resource intersection between processes. The following definitions can be used as a formal criterion of resource intersection presence.

Definition 1. AT-conflict between processes exists, if the different processes have to use the same address during the same time slot.

Formal criterion of AT-conflict's absence:

$$\max_{i,j} \sum_{k=1}^{N_{PR}} (AT_k) = 1 \qquad (3.1)$$

where N_T, N_A – maximal AT-map matrix dimensions, N_{PR} – number of processes.

Critical section analysis problem can be stated as integer linear programming task if value $F(p_i,\ldots p_m)$ (see exp. 3.2) is used as a measure of conflict.

$$F(p_i \ldots p_m) = \max_{i,j} \sum_{k=1}^{N_{PR}} (AT_k) \qquad (3.2)$$

Set of optimization parameters $\{p_i\}$ depends on application. This approach can be used as a very first step. It is clear that specialized algorithms have to be developed.

The same approach can be useful for two other conflict types.

Definition 2. A-conflict between two processes exists, when different processes use the same address.

Definition 3. T-conflict between two processes exists, when different processes use the same time slot.

It has to be pointed out, that presence or absence of conflicts is not bad and not good - it depends on application. As example, global variable in any case leads to A-conflict between program units, but this conflict is introduced intentionally to send information from one program unit to another.

IV. PROBABILISTIC BEHAVIOR OF MULTI CHANNEL REAL-TIME CONTROL SYSTEMS

Multi channel control system demonstrates a probabilistic behavior in some conditions even in normal operation modes. It means that response time has to be specified as probabilistic value. Software development technology for hard real time systems should be modified to take into account this phenomenon. Some basic ideas are introduced below.

Embedded control system reacts on input signals that come from sensors. There are some possibilities to organize processing of input signals. Let us start from simple situation when interrupt system is switched off and FIFO scheduling discipline is used to process digital input signals. This sample system can handle only signals that are active while processing task is running. Processogram for model system with $k=N_{pr}$ channels is shown at fig. 4.1.

Let us suppose that every task processes one input signal to produce a reaction to controlled object with actors. In this situation whole task set can be executed during $\mathbf{T_c}$ period. Response time $\mathbf{T_r}$ equals $\mathbf{T_c}$. It is stated with the assumption that $\mathbf{T_{serv} << T_i}$, where $\mathbf{T_{serv}}$ is a signal service duration. All other time task waits for a signal. $\mathbf{T_c}$ depends on number of channels $\mathbf{N_{ch}}$ and duration of every task $\mathbf{T_i}$

$$T_r = T_c = \sum_{i=1}^{N_{ch}} T_i \qquad (4.1)$$

Task can have different duration values for different branches of an algorithm. Duration varies from $T_{i\,min}$ to $T_{i\,max}$, so

$$(\sum_{i=1}^{N_{pr}} T_{i\,min}) < T_c < (\sum_{i=1}^{N_{pr}} T_{i\,max}) = T_{c\,max} \qquad (4.2)$$

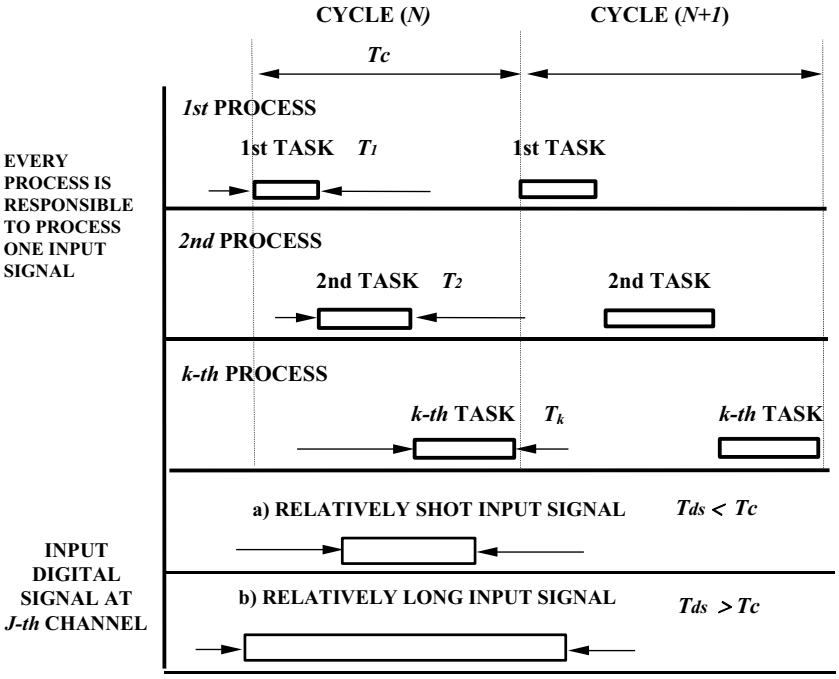


Figure 4.1 FIFO dispatcher processogramma

Using standard approach, the response time T_r for such system in normal working conditions can be estimated as

$$T_r \approx T_{cmax} \qquad (4.3)$$

Whole set of digital signals can be divided in accordance with the rule:

Tds < T_c, subset of relatively short signals in comparison with **Tc**,

Tds > Tc, subset of relatively long signals in comparison with **Tc**.

As it can be seen from fig. 4.1, the relatively short signal can be missed, if it comes when system executes processes that are responsible to other signals. Probability to process digital signal at input channel j could be estimated as

$$\begin{aligned} Pds_j &= \frac{T_j + Tds}{T_c} \quad \text{for } 0 < Tds < T_c - T_j \\ Pds_j &= 1 \qquad\quad \text{for } Tds > T_c - T_j \end{aligned} \qquad (4.4)$$

where T_j – time to process signal at channel j, Tds – duration of digital signal.

Probability to miss digital signal at input channel j can be estimated as

$$Pf_j = 1 - Pds_j \qquad (4.5)$$

The same approach can be used for RR dispatcher. Proccessogram of such system is shown on fig. 4.2.

For the system with N_{ch} channels and T_k time slot duration, T_c can be estimated as

$$T_c = N_{ch}(T_k+T_{isr}+T_{disp}). \quad (4.6)$$

Let us assume that

$$T_j < T_k \qquad j=1\ldots N_{ch} \quad (4.7)$$

where T_j – time to process signal at channel j.

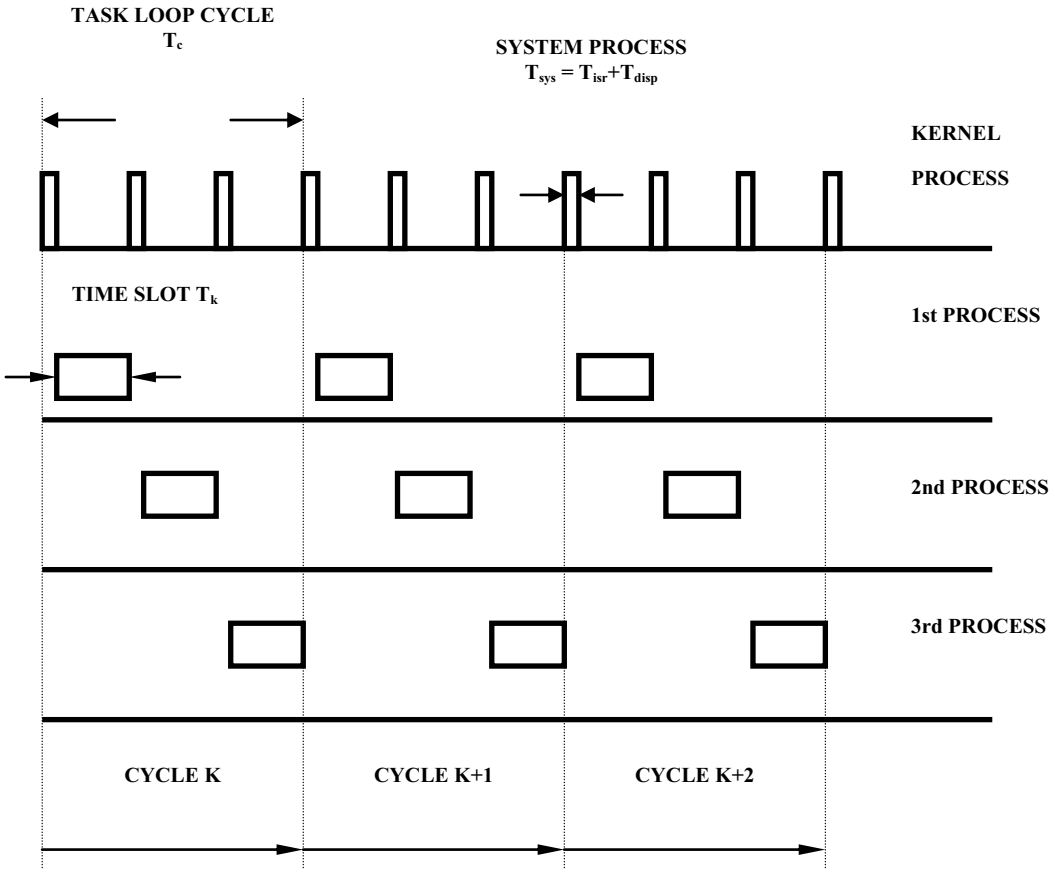


Figure. 4.2 RR dispatcher processogramma

Response time in normal working conditions can be estimated as

$$\mathbf{Tr_j \approx T_c} \quad (4.8)$$

Probability to process input signal could be estimated as

$$\begin{aligned} Pds_j &= \frac{T_k + Tds}{T_c} \quad \text{for } 0 < Tds < T_c - T_k \\ Pds_j &= 1 \qquad\quad \text{for } Tds > T_c - T_k \end{aligned} \quad (4.9)$$

Probability to miss signal for these conditions can be estimated as

$$Pf_{ds} = 1 - \frac{T_k + Tds}{T_c} = 1 - \frac{T_k + Tds}{N_{ch}(T_k + T_s)} \quad (4.10)$$

It can be pointed out, that **RR** scheduling discipline is more preferable for HRTA. For control system with RR dispatcher response time has more deterministic character due to absence of interference between processes in time domain. But for relatively short signals in comparison with Tc response time is a probabilistic value still:

$$Tr_j = Tr_j\,(Pds_j) \tag{4.11}$$

The next point in this road is an interrupt system. In practice RT control systems use interrupt-based scheduling to decrease response time while processing asynchronous input signals. It has to be pointed out that RR dispatcher can't be implemented without system timer, that supposes using the interrupt system.

A model processogram for system with one high priority input channel is shown at fig. 4.3. Interrupt signal servicing time $\mathbf{T_{rhp}}$ consists of context switching time $\mathbf{T_{isr}}$ and signal processing time $\mathbf{T_{hp}}$. Response time for background signals in that case can be estimated as

$$Tr_j \approx T_c = (T_{cmax} + T_{isr} + T_{hp}) = (T_{cmax} + T_{rhp}) \tag{4.12}$$

The probability to process background signal can be estimated as

$$\begin{aligned} Pds_j &= \frac{T_j + Tds}{T_{ci}} \quad \text{for } 0 < Tds < T_{ci} - T_j \\ Pds_j &= 1 \qquad\quad \text{for } Tds > T_{ci} - T_j \end{aligned} \tag{4.13}$$

where Tci = (Tc + K_{int} Trhp). Factor $\mathbf{K_{int}}$ depends on number of interrupts per one FIFO cycle $\mathbf{T_c}$.

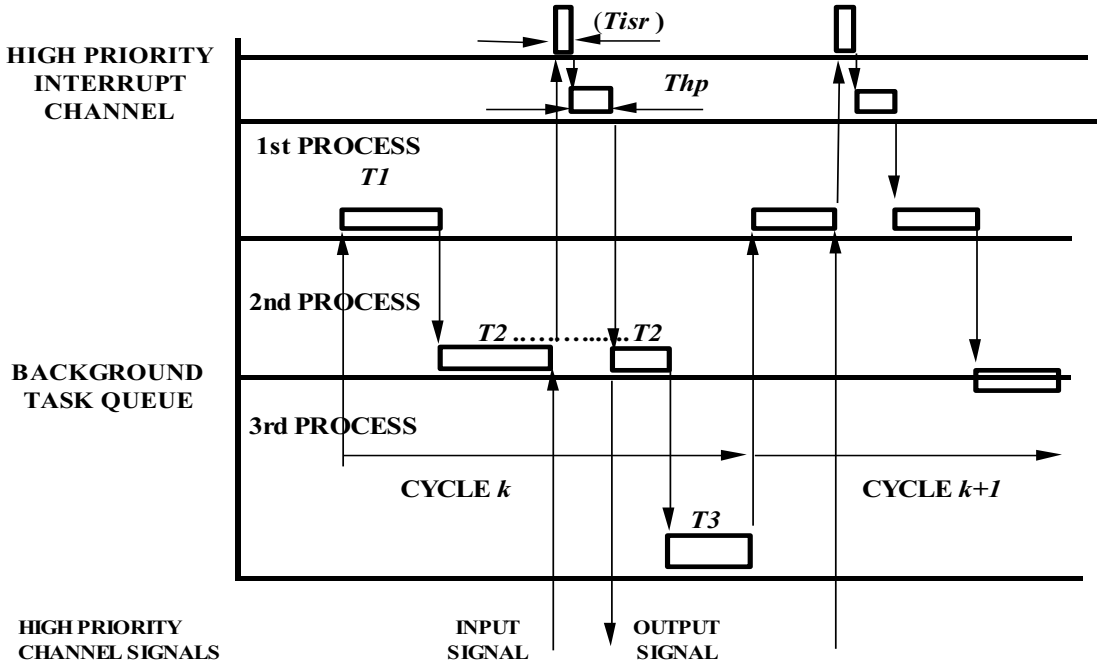


Figure 4.3 Proccesogramma for interrupt-based system

V. OCS AND MCLET SYSTEM MODELS

Two simple models based on Poisson flow model will help to establish better understanding of the probabilistic behavior of RTCS that use the interrupt system to process high-priority signals.

Interrupt system usage looks like this. After the interrupt system has got signals some action is to be made. From the very beginning the interrupt system has to be disabled to prevent interruption of current signal processing by other interrupt sources. After that the context of interrupted task has to be saved, then an interrupt service routine can be started.

In accordance with fig. 4.3 response time for high-priority signal can be estimated as

$$T_{rhp} = T_{isr} + T_{hp} \tag{5.1}$$

OCS (One Channel System) model supposes that the system has just one high-priority input channel that is describable as a Poisson flow with intensity $\lambda = 1 / Tas$, where Tas is the mean wait of input signal.

For Poisson flow a probability of that exactly *k* signals will come during period T is equal to

$$P(k, T) = \frac{(\lambda T)^k e^{-\lambda T}}{k!} \tag{5.2}$$

For input signals, that are relatively short in comparison with T_{rhp}, the probability to process all incoming signals is equal to probability of at most one signal incoming during period T_{rhp}

$$P_{php} = P(0, T_{rhp}) + P(1, T_{rhp}), \tag{5.3}$$

where

$$P(0, T_{rhp}) = e^{\frac{-T_{rhp}}{Tas}}, \; P(1, T_{rhp}) = \frac{T_{rhp}}{Tas} e^{\frac{-T_{rhp}}{Tas}}.$$

For conditions described above there is a probability to miss a high priority signal due to self-blocking effect

$$P_{fhp} = 1 - P_{php} = 1 - (P(0, T) + P(1, T)) \tag{5.4}$$

A stability condition for OCS can be formulated as

$$Tas = Ks \, T_{rhp}, \tag{5.5}$$

where safety factor $Ks > 1$ is used to decrease a probability of high-priority signal missing.

Probability of high-priority signal missing due to self blocking effect for $Ks >> 1$ can be estimated as

$$P_{fhp} = 1 - P_{php} = 1 - (P(0, T) + P(1, T)) \tag{5.4}$$

$$P_{fhp}(OCS) = 1 - (P(0,T_{rhp}) + P(1,T_{rhp})) = 1 - (\frac{1}{Ks} + 1) e^{\frac{-1}{Ks}} \approx \frac{1}{Ks^2} \tag{5.6}$$

This approach allows selecting control system parameters with specified probability of system's failure.

Next step along this road supposes examination of multi-channel system specific features. MCLET (Multi Channel Low Equal Traffic) model supposes that system has Nch equivalent signal channels. In accordance with the model, system has to process digital signals with a Poisson nature that have the same intensity: $\lambda = 1 / Tas$.

This model system can process all high-priority signals with the condition: when one channel has signal, no other (Nch-1) channel has one and that channel has at most one active signal.

$$P_{fhp}(MCLET) = 1 - P_{php}(OSC)\, P_{php}(MCLET_{others}) \quad (5.7)$$

$$P_{php}(MCLET_{others}) = P(0,T_{rhp})^{(Nch-1)} = e^{\frac{1-N_{ch}}{Ks}} \quad (5.8)$$

Where

$$P_{php}(OCS) = (\frac{1}{Ks}+1)e^{\frac{-1}{Ks}},$$

$$P_{fhp}(MCLET) = 1-(\frac{1}{Ks}+1)e^{\frac{-N_{ch}}{Ks}}$$

Table 5.1 represents probability of missing relatively short signals calculated with (5.6)-(5.8) for different safety factors.

Some new approaches were proposed during last decade. These ideas root from Allen's interval algebra [3]. A number of algebras (interval, event, even-temporal and so on) have been proposed for different applications [4, 5, 6] after his work.

Table 5.1.
Probability to miss signal for Poisson input signal stream

Ks	Pfhp(OCS)	Pfhp(MCLET)					
		Nch	10	30	100	300	1000
1	0,2642		0,9999	1	1	1	1
2	0,0902		0,9899	1	1	1	1
5	0,0175		0,8376	0,9970	1	1	1
10	0,0047		0,5953	0,9452	1	1	1
100	10^{-5}		0,0861	0,2518	0,6284	0,9497	1
1000	10^{-7}		0,0090	0,0286	0,0943	0,2584	0,6318
10000	10^{-9}		0,0009	0,0029	0,0099	0,0295	0,0951

Reliability of embedded control system is a function of used scheduling discipline, safety factors and behavior of input signal stream. In accordance with consideration presented above, for embedded control system it is necessary to specify mean response time and it's variance, also as a probability of signal missing.

So, the following statement can be obtained from considerations presented above.

STATEMENT. The response time of multi-channel RTCS has to be interpreted as a probabilistic value:

$$\mathbf{Tr_j = Tr_j\,(Pds_j)} \quad \mathbf{(5.9)}$$

This statement means that the whole approach of software designing for real-time applications has to be changed thoroughly. It has to be pointed out that in real engineering practice it is necessary to take into account duration of initialization phase, abnormal conditions and so on.

It means, that a question "What is the probability of that the software developed for multi-channel RTSC will work properly?" has sense and great practical meaning.

Here some paradox exists that can be illustrated with (5.6). Reliable system has to use a large safety factor, so probability of failure is low. It means, that testing procedure for really reliable system has a long duration and has to be planned with probabilistic nature of control system. In work [2] some special procedures were proposed to overcome this situation.

VI. AT-FORMALISM

It can be pointed out, that probability analysis presented above uses high-level one-dimensioned description of algorithm.

These ideas can be useful for analysis of time structure of the automatically generated code and for the experimental researching of multi channel system in time domain. As example, the Interval Calculus from [4] and it application for data management [7] can be pointed.

All this proposals have deal with the one dimensional objects. As it was concluded in [1] an automatic solution of critical section analyses problem will needs more complex instrumentation, then some variation of the interval algebra.

In accordance with AT-model one needs a two-dimensioned description to use formal criterion (3.1), supposing that a set of full rectangular matrixes has to be formed. Every such matrix describes resources that required for every process during the interval of interest. As a rule, embedded control system dispatcher is some sort of an endless loop. That means, that description, based on standard matrix representation will be difficult to use for practical purposes because it will require endless memory.

AT-formalism was proposed in [1] and developed in [2],[8],[9] as a tool for symbolical descriptions of AT-map for composed task and AT-map for processogramma and cyclogramma. AT-map of composed task is described as a formula that contains a sequence of AT-formalism operations applied to tasks' AT-maps. Summary description of a process, produced by complex task, can be represented in the same way.

This approach supposes that a software library has a special description for every subroutine. This description is based on AT-model and it concerns resources, needed to perform the function. It is reasonable to call it as AT description. Automatic analysis of critical section and some other problems can be solved with AT description having used.

A classification and a set of operations are described with :

Definition 1. Operation that affects just one dimension of AT-map will be called as one-dimensioned operation, namely, A-operation or T-operation. Operation that affects two dimensions of AT-map will be called as two-dimensioned operation or AT-operation.

Definition 2. Operation that is performed at one AT-map will be called as monadic operation. Operation that is performed at two AT-maps will be called as dyadic operation. Operation that is performed at more then two AT-maps will be called as polyadic operation.

Definition 3. Changing of the result object dimension will be described with integer number, called operation rank.

Definition 4. Operation that uses one parameter will be called as scalar operation. Operation that uses more than one parameter will be called as vector operation.

AT-formalism is a very young branch of computer science and still open for further development. To illustrate main ideas of AT-formalism, the so-called union operation was used as example shown below. It has to be pointed out that used notations don't correspond to works [1], [2], [3]. It is a result of further development of this theory [10], and proposed notation that is more convenient from practical point of view.

Fig. 6.1 illustrates basic ideas of AT-formalism with union operation applied to two tasks' AT-map. This operational is 1-D (at T-axis). It has to be outlined that this operation does not influence on value of origin matrix elements, but it changes their positions. So, it can be stated that it is an operation over the elements' indexes.

Let us use notation **[∨+D]** for union operation of two AT-maps. Parameter D is a signed integer number that describes a displacement of the second operand in regard to maximal time slot number of the first operand. For the following example D = 0. Notation:

$$\sum_{i=1}^{N_{pr}}{}_{[\vee+D]}(AT_i) \tag{6.1}$$

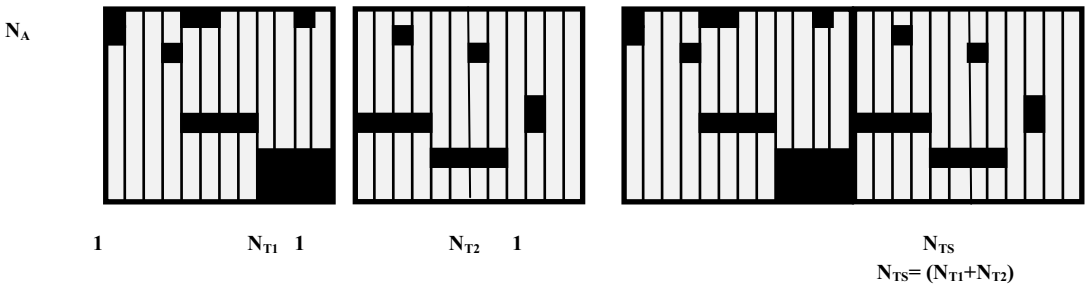
is used to describe polyadic 1-D T-axis union operation. Using this notation, system process shown with processogramma presented with fig. 4.2, can be described as:

$$\mathrm{AT_s} = \sum_{i=1}^{\infty}{}_{[\vee+D]}(\mathrm{AT_{sys}}) \tag{6.2}$$

Symbolical representation

AT_1 [union operation] AT_2 = AT_c

Graphical representation



AT-map representation using sparse matrix technique

AT_1 **[union operation]** **AT_2**

[NA, NT1, N1] $\{i, j\}_1$ **[NA, NT2, N2] $\{i, j\}_2$**

Operation result description

AT_s [NA, NT1+NT2, N1+N2] $\{i, j\}_s$

where

$\{i,j\}_s$ = $\{i,j\}_1$, if k = 1 .. NT1

$\{i,j+NT1\}_2$, if k = (NT1+1)…(NT1+NT2)

Figure 6.1 Example of the union operation that forms composed task's AT-map

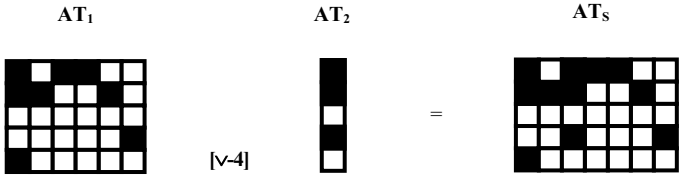


Figure 6.2 Union operation using negative D parameter

Whole processogram for RR-scheduling can be described with formula

$$AT_s = \sum_{k=1}^{\infty}{}_{[\vee+D]}\left(\sum_{i=1}^{N_{pr}}{}_{[\vee+D]}\left(AT_{sys[\vee+D]}AT_i\right)\right) \qquad (6.3)$$

Union operation with D < 0 can be used for description of process that includes interrupts as it illustrated at fig. 6.2.

One-dimensioned A-union operation can be used to describe code allocation in memory, as it illustrated at fig. 6.3.

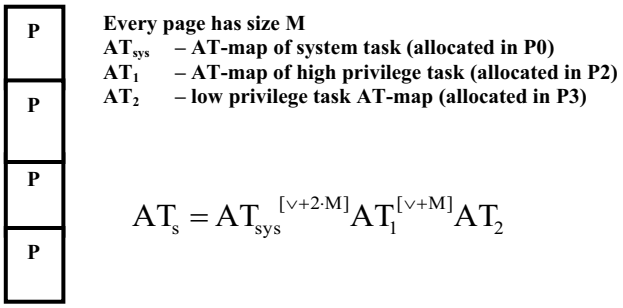


Figure 6.4 Special features of one- and two-dimensioned operations:
a) 1-D A-union operation b) 1-D T-union operation c) 2-D AT-union operation

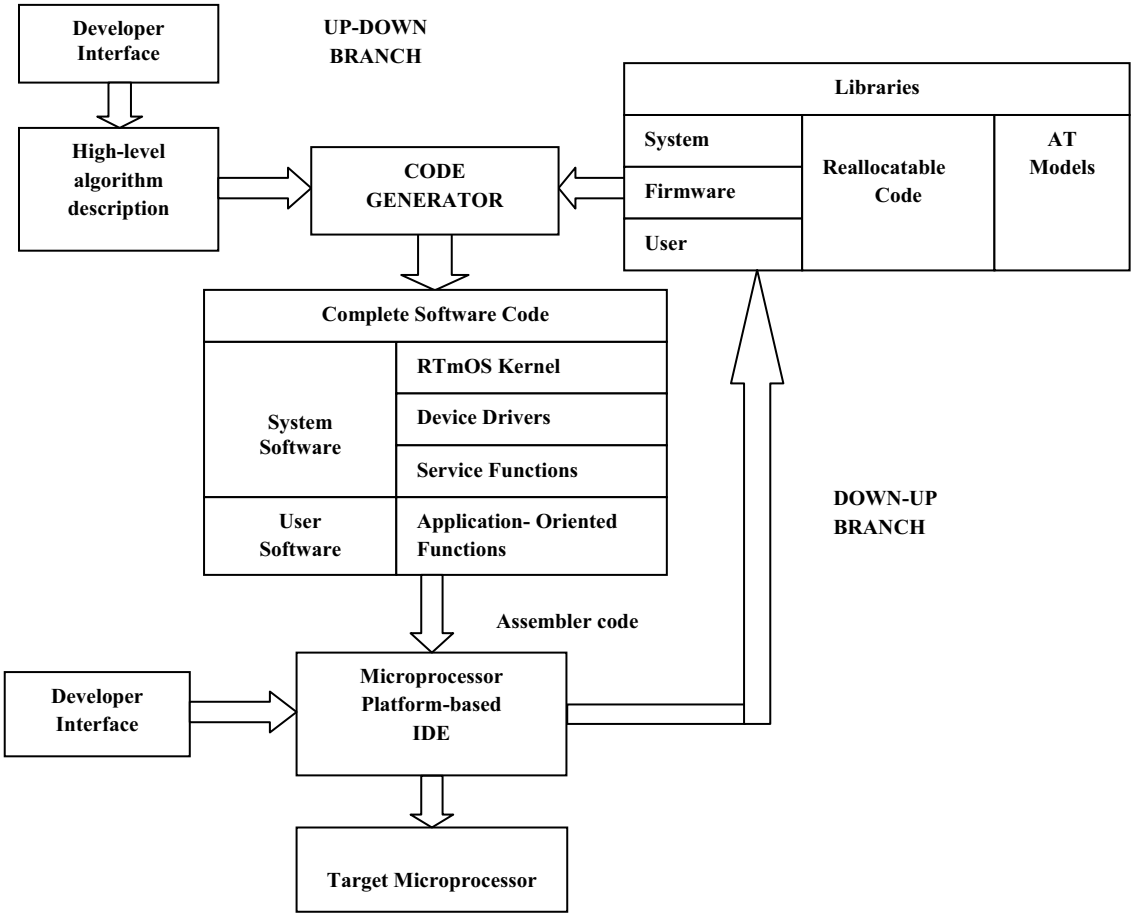


Figure 7.1 Structure of next-generation IDE

Extraction operation [∧+D] can be introduced as an opposite to union operation:

$$(AT_{1[\vee+D1]}AT_2)_{[\wedge+D2]}AT_2 = AT_1 \qquad (6.4)$$

Some rules are to be introduced to provide such conversion with D1=D2.

More or less full set of operation is described with table 6.1. Some remarks about can be made.

Projection operations can be used as a bridge to some sort of interval algebra, if one will use start and last number of time slot to describe an interval.

Multi union operation has no analog in classical theory and some additional work have to be done to develop a practical tool.

AT-map formalism was proposed as a tool for theory of RTmOS but was used in practice in some projects.

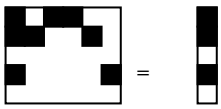
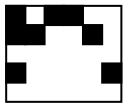
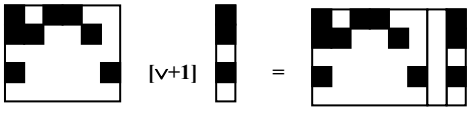
As example, projection and cutting operations were used to describe the user interface in multichannel control systems, AT-model were used for system analyses of multichannel digital video systems.

VII. CONCLUSION

It is clear that for modern electronic devices the software development is impossible without using IDE.

Table 6.1

Basic Symbolic At-Card Operations

NAME AND SYNTAXIS	EXAMPLE	GENERALIZATION AND REMARKS
PROJECTIONS: [⊥A] AT = {a_i} [⊥T] AT = {t_i} ONE OPERAND OPERATION RANK (-1) MATRIX PROJECTION ON ONE AXE : A or T	**[⊥A]**  **=**	**PROJECTION ON PREDEFINED SET {m_i} [⊥(A def{m_i}] AT = { a_i}. [⊥(T def{m_i}] AT = { t_i}.**
CARD DIMENSIONS [DT] AT , [DA] AT ONE OPERAND OPERATION RANK (-1)	**[DT] AT = $i_{max} - i_{min}$ [DA] AT = $j_{max} - j_{min}$**	**REALIZATIONDEPENDS ON CARD DESCRIPTION**
d2 CUTTING : [d1 □ d3] AT=AT1 d4. TWO OPERANDS OPERATION RANK(0), THAT FORM A NEW AT-CARD. OPERATION HAS PARAMETERS d1,d2,d3,d4	**0 [4□ 0] 4**  **=** 	**PARAMETERS d1,d2,d3,d4 ARE INTEGER POSITIVE NUMBERS**
UNION : AT2 [∨+d] AT1 = AT3 TWO OPERANDS SCALAR OPERATION RANK (0) THAT FORMS ONE AT-CARD FROM TWO. RESULT DEPENDS ON AT1,AT2 AND PARAMETER d	 **[∨+1] =**	**MULTI OPERANDS UNION OPERATION APPLIED~TO N AT-CARD ∑ [∨+d] AT_i IT IS POSSIBLE TO INTRODUCE A VECTOR UNIONOPERATION ∑ [∨+D_i] AT_i .**
MULTI UNION OPERATION RANK(+1) [W+[d1-d2]] THAT IS NUMBER OF INSERTION OPERATION WITH PARAMETERS d1, d1+1, ….d2. RESULT ARE (d2-d1+1) AT CARDS.	**This operation is proposed as a tool to describe the AT-map for code with an interruptions. From point of the view classical theory of sets this operation introduces a new type of set with insertion rule ” One From Plenty” .**	

It can be stated, that modern IDE use the RTmOS and down-up technology of programming and it is the common practice. As example, the latest microprocessor platform BlackFin (Analog Devices, RTmOS VDK), OMAP (Texas Instruments, RTmOS Windows CE 6.0) can be pointed.

For this IDE the algorithm is presented as set of OS threads, that are described with ANSI C/C++ languages. It is possible to use an assembler inserts also. Third party companies have developed a set of RTOS kernels for this platform: ThreadX (Express Logic, USA), ucLinux (open project), NUCLEOUS (Accelerator Technology, USA), MicroC/OC-II (Micrium, USA).

From the very beginning RTmOS was developed as applied theory, oriented to be a theoretical base for developing the next generation IDE for real-time applications.

Next-generation IDE has to support up-down technology in addition to down-up one.

As a very first proposal, the structure that is represented on fig. 7.1 can be used. Down-up branch is used as a basic tool to create a library functions. The librarian is to have possibility to create AT-descriptions of stored functions. Whole control system algorithm has to be described with the high level language. This description has to be used for computer added analyses of code. AT-description creates the base for such analyses.

Now is better understanding of the existing problems has been established, but a lot of work has to be done after this first step.

REFERENCES

[1] Астапкович А. М. Микрооперационные системы (mOCPB) для встроенных систем управления.// В сб. Информационно-управляющие системы и сети. Структуры. Моделирование. Алгоритмы. Политехника, СПб, 1999, с.196-224.

[2] Астапкович А. М. Микрооперационные системы реального времени. “Политехника ”, СПб., с.246.

[3] Allen J. F. Maintaining Knowledge about Temporal Intervals, Communications of the ACM, vo.26(11),pp.832-843,1983

[4] Allen J. F. and Ferguson G. Actions and events in interval temporal logic. Journal of Logic and Computation, 4(5) pp.531-579, October, 1994

[5] Chakravarthy S., Krishnaprasad V., Anwar E., Kim S-K. Composite events for active databases: Semantic, context and detection. 20-th International Conference on Very Large Data Bases, September 12-15,1994, Santiago, pp.606-617, Los Altos, CA 94022,USA,1994, Morgan Kaufmann Publishers

[6] Yoneki E., Jean Bacon. Determination of Time and Order for Event-Based Middleware in Mobile Peer-to-Peer Environment. Proceedings of the 3rd Int Conf on Pervasive

[7] Kowalski R., Sergot. A logic-based calculus of events. Foudations of Knowledge Base Management: Contributions from Logic, Databases and Artificial Intelligence, p. 23-55, Springer, Berlin, Heidelberg, 1989.

[8] Астапкович А. М. Формализм адресно-временных карт для описания алгоритмов функционирования многоканальных систем управления. Введение в формализм адресно-временных карт. Информационно-управляющие системы 4/2003, с. 6-14,

[9] Астапкович А. М. Формализм адресно-временных карт для описания алгоритмов функционирования многоканальных систем управления. Базовые объекты и операции с АТ-картами Информационно-управляющие системы 2/2004, с. 26-36.

[10] A.Astapkovitch, A.Kasatkin Review of Real-Time micro-Operating System Theory Proc. International seminar “Challenges of the information age in the field of education”, p.7-18, SUAI, June 24-28, Saint-Petersburg, 2006 http://www.guap.ru>student design center>publications .