Scientific Journal of Riga Technical University
Computer Science. Technologies of Computer Control
2011
Volume 48

Self-Organizational Paradigm in the Time Critical Systems

Valery Zagursky[1], Dmitry Bliznjuk[2], Roman Taranov[3], [1-3]*Riga Technical University*

Abstract. **The present paper examines the problems of time critical systems. The core of the problem lies in dynamic resource allocation and time dependent functional changes of the systems. The extensive research is of crucial importance for system planning, performance evaluation, time synchronization, resource access control and system infrastructure. The formalism in a group of stochastic automata is also examined. The definitions of "time critical system" and "real time system" are given in a context of dynamic and stochastic systems. Such a system functioning is determined by a functional, logical and morphological description. The procedures of functioning can be specified in different ways: as macro models [2], and directed graph of procedures [4].**

The mathematical approach to generalization of the above mentioned definitions has been proposed. This approach might be used for self organization of system infrastructure or making final decision. Finally, the paper presents the approach to optimization and design of critical time functioning systems.

Key words: self-organization, critical time, real time, timing automata, stochastic systems.

I. Introduction

New systems research and development are consisted with functional and morphologic descriptions. The main demand for successful self-organization is some general start program. This program is capable to improve system behaviours, by increasing quantity of contenting in it information. This features can be performed by media account or /and in cooperation with another systems. The functional description of self-organizing features has been proposed in the paper [1,6] as a sample of homeostat and the same different extensions in the paper [4,7-9]. The multiform morphologic description has been proposed in [2,3,5,7-10] and can be used for design and simulation of diversity dynamic behaviours at different systems levels description. The innovative approach has been proposed in [11-13] where real-time systems have been described by using priced timed automata. Timed automaton and their game extension provide limited quantitative aspects of the system description. There are limited possibilities for flexible and simultaneous implementing of description at different system levels.

II. Problem statement

In this paper we shall assume that automaton [5] is described by this formula:

$$\varphi(t+1) = \phi\{\varphi(t), s(t+1)\}$$
$$f(t) = F[\varphi(t)]$$

, where $t = 1, 2, 3 \ldots$ - discrete time;

$s(t) = (0,1)$ – integral input variable;

$s = 0$ – single winning of positive situation in a system by successful procedure,

$s = 1$ – the loss;

$f(t) = \{f_1, f_2, \ldots f_a, \ldots f_j\}$ – function of operation of automaton

if $f(l) = f_a$, $a = 1, j$ – automaton made a action,

$\varphi(t) = \{\varphi 1, \varphi 2, \ldots, \varphi j, \ldots \varphi n\}$ – state of automaton;

n – capacity of its' memory.

If $F(\varphi_i) = f_a$, where f_a corresponds to φ_j. The transition of automaton from one state to another can be described by this transition matrix:

$$a = \| a_{ij}(s) \|, \text{ i, j = 1, n}$$

and for determined automate each row of matrix a contains one unit and other zeros. For stochastic automaton $a_{ij}(s) \leq 1$ the possibility to translate from i to j will take place with known S.

For the static random environment $a = a\ (a1, a2, a3, .. aj)$ action f_a in a moment t, will result in a loss in a moment $t+1$ $(s=1)$ with probability of $p_a = (1-a_d)/2$ and winning $(s=0)$ with probability of $q_a = (1+a_a)/2$. If in the moment t automate was in a state φ_i, $i=1,m$, to which the following action $f_{ai}=F(\varphi_i)$ corresponds, then a possibility p_{ij} of the transition to the φ_i is equal to the $p_{ij}=p_{ai}a_{ij}(1) + q_{ai}a_{ij}(0)$, $i,j =1, m$. Matrix $p = ||Pij||$ is stochastic and process of automate functioning is Markov process. The mathematical expectation of automate L winning in the environment C is:

$$M(L,C) = \sum_{a=1}^{\infty} \sigma_a \alpha_a$$

,where σ_a – is a probability of the operation f_a

and $min(a1, a2, \ldots aj) \leq M(L,C) \leq max(a1, \ldots, aj)$.

For the automaton, which is operating independently from the environment (σ_a - uniform probable distribution) mathematical expectation is:

$$M = \frac{1}{j} \sum_{a=1}^{\gamma} a_a$$

III. Proposed approach

There is a necessity to change f(t) and $\varphi\ (t)$ in the non static *environment.* It means, that automaton have to be trained and retrained according to environmental changes. Time of training (or retraining) depends on the automate description and on the time, which is needed to detect the state of environment.

We shall consider joint functioning of automate group with procedural model graph implementation. This kind of graph orders the procedure amount and name. The arrows on this graph show the directions, which contain information or/and make actions. Moreover, information may be provided from outside (from environment), or be produced in that group, by executing the given procedures.

Each procedure may be final and may lead to the certain decision, but it may also be intermediate and lead to the new procedure.

Let us consider that each automata of a group can make a decision, by using any procedure from the list of procedures, or can delegate the decision making to (by fulfilling any set of the procedures from the list) another automata. In general case assignment of the feature of each automaton (for example in the form of different procedure usage probability distribution and probability of decision making delegation). Using them we will describe a procedure graph:

$$p_s^{(i)} = p_{s^0}^{(i)}(1 - \exp - t / \tau_s^{(i)})$$

, where t – time is needed to make a decision;

$\tau^{(i)}_s$ – time constant of decision making, while doing *i* procedure on automaton *S*.

$p^{(i)}_{s0}$ – maximal probability of making decision S_0 with total number of automaton (with $t \to \infty$) with fulfilling procedure *i*.

Matrix of transition probabilities is:

$$P_s = \| P_{s(i,j)} \|, i, j = \overline{1, s_0}$$

Time of transition is described by matrix:

$$\tau_s = \| \tau_{s(i,j)} \|; i, j = \overline{1, s_0}$$

Full probability of making decisions by position:

$$\boldsymbol{P_s^{(i)}} = \sum_{i=1}^{s_0} \boldsymbol{P_s^{(i)} P_{si}}$$

Probability of delegating decision making to automaton *s+1* is:

$$P_{s+1} = \sum_{i=1}^{s_0} [(i - P_s^{(i)}) \sum_{i=1}^{S_0} P_{s_i} P_{s_{ij}}]$$

, where j is a number of j automaton of group

Procedures can be conditionally formalized and represented as an ordered algorithm actions' sequence. The way of ordering, similarly as concrete meaningful essence of procedures, can be changed depending on necessarily functional and morphological levels of consideration of assigned tasks. Quantitative list of procedures may be any.

Automatons of the group for the effective successful execution of a procedure may each exchange of results of procedures following the scheme, which is described by orientated sub graphs. We will assign a probability of finding a solution (in some known class of decision making) and time of decision making to each vertex of the graph, as well as we will assign time of transition necessary for executing next procedure, if a decision has not been made, to each edge of the graph.

The repetition of any procedure is possible. It is possible to calculate limit evaluations, time of decision making, critical paths. i.e., sequence of delegating decision making, which can be applied to reach minimal and maximal time of decision making. The graph can be optimized according to the limitations: getting the minimal time T_r of decision making with known probability P_r of solution or maximization of decision probability in a specified time. Herewith we consider, that each automaton is operating independently in the limits of its own competence.

Each additional exchange of information may increase P_r by increasing T_r.

Let's consider that P_{si} – probability of decision making by S automate is related to information exchange

$$P_{s_i}^{(s_j)} = (1 - P_{s_i}^{(o)}) \prod_{j=1}^{s_j} (1 - P_{s_j} P_{ji}),$$

where P_{sj} – probability of information exchange to *j* automaton,

S_j – number of automatons, which are exchanging information

P_{ij} – probability of successful information exchange with *j* automaton by *i* procedure, $P_{ji} \le 1$

From formula $P_s^{(i)} = \sum_{i=1}^{s_0} P_s^{(i)} P_{si}$ we will extract

$$P_{s_i} = 1 - P_{s_i}^{(s_j)} = 1 - (1 - P_{s_i}^{(0)}) \prod_{j=1}^{s_i} (1 - P_{s_j} P_{ji})$$

with $P_{sj} = 0$ $P_{ij} = 0$ or $S_j = 0$, $P_{s_i} = P_{s_i}^{(0)}$ and $P_{sj} = 1$

$$P_i = 1 - (1 - P_{s_i}^{(0)}) \prod_{j=1}^{s_i} (1 - p_{ij}) > P_{s_i}^{(0)}$$

In this case graph optimization is dependent on choice of procedures and information exchanges between automaton, which fulfil these conditions:

$(T_R[P_R \ge P^*]) \to \min$, or

$(P_R[T_R \le T^*]) \to \max$, where

P_R^*, T_R^* are corresponding threshold values.

This way we can concretely specify limitations, which are related with uncertainty of system functioning "critical time" and more uncertain characterization of real time.

In the case of more precise and simple definition of system functioning critical time is appropriate to use – time of decision making by system or system self-organization time.

IV. EXAMPLE.

The process of conflict resolution [5] can be depicted as a flowchart or, more precisely, a connected locally finite and oriented graph (Fig. 2) of tree type with a root. The graph vertices denote the following events: αC(m), after a time segments a collision of m transmissions occurs; bT, after b time segments a successful transmission occurs, where a, b ∊ {0, 1, 2}. The arc denotes transitions from one event (or vertex) to another. Each path (route from root to leaf) corresponds to one process, while all possible paths correspond to the set of all possible processes that pass through the first (root) vertex). Arcs that depart from the same vertex correspond to classes of outcomes, and therefore it is convenient to number them in the same order, while the probability P_i of appearing in the i-th class is the probability that a process will proceed further from vertex C(m) along the i-th arc.

We denote the length of the path (or of the path segment under consideration) by R, and we define it as the number of vertices that are encountered on this path (or segment). Then the list of all possible paths of length ≤R can be compiled either in the form of a string of events (sequence of graph vertices) or in numerical form (sequence of arc numbers). The probabilities of all the arcs making up a path are already known, and therefore we can use the multiplication rule to determine the probability P_K that a process will follow the k-th path. For example, for m = 2 according to Fig. 2 we have R ≤ 3. As we know, both forms of path notation uniquely designate a single process. Inspection of all paths of length ≤R means enumeration, with constraints, of all R-

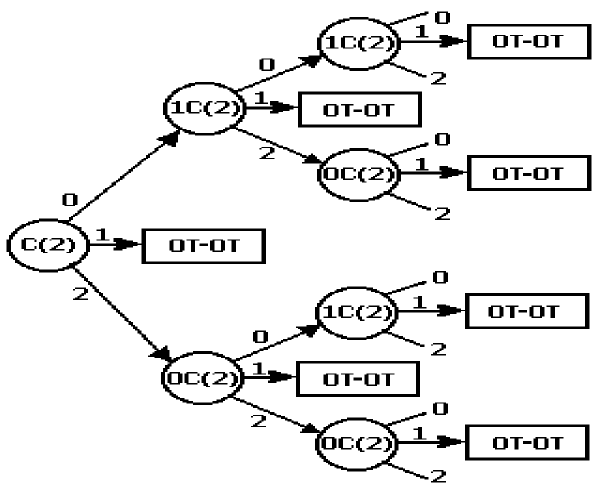

Fig. 1. Flowchart of conflict-resolution process

00=C(2)-1C(2)-1C(2)...	$P_{00}=P_0^2$
01=C(2)-1C(2)-0T-0T.	$P_{01}=P_0P_1$
02=C(2)-1C(2)-0C(2)....	$P_{02}=P_0P_2$
1=C(2)-0T-0T.	$P_1=P_1$
20=C(2)-0C(2)-1C(2)...	$P_{20}=P_2P_0$
21=C(2)-0C(2)-0T-0T.	$P_{21}=P_2P_1$
22=C(2)-0C(2)-0C(2)...	$P_{22}=P_2^2$

position numbers in the (m+1)-th number system; in the general case, different numbers have different bases. All possible paths of specified length, belonging to the initial vertex C(m), form a complete system of paths, and therefore $\sum_k P_k = 1$. If all outcomes turn up either in the zero or m-th classes, then the length of this process R = ∞. Therefore the flowchart is also infinite, and is drawn only in part. At the same time, the most probable processes are short. The longer the process is, the lower its probability, and, therefore, the probability of an infinitely long process is equal to zero.

It is more convenient to employ finite flowcharts in which the zero and m-th arcs are transformed into loops (Fig. 3), the zero loop yielding a delay t_S. Flowcharts with loops are much simpler, since they do not explicitly show repeat collisions (number of steps around loops). In them, however, it is necessary to determine the average delay D_m of the process at the vertices C(m). For this we compile a list of all half-segments of length r = 1, 2, 3, ..., that close at vertex C(m), and we determine the probability p(m, x) of each of them. A segment of such a path begins by arriving at vertex C(m) and ends by departing from it along the j-th arc, where j = 1, 2, ..., m - 1. We also determine the instantaneous sum of the probabilities of all the paths already considered, and we terminate the length increase for r = r_{max}, when the instantaneous sum $\sum_x p(m,x)$, with sufficient accuracy, becomes equal to 1. The length of each such a segment of the process is l(m,x)=z+cr, where z is the number of zero loops in the given segment, i.e., the number of delays of magnitude t_S in A1-TR transitions; r is the number of repeat collisions C(m) on the given segment; and c is the average collision length in time segments. As we know, the collision length lies in the range ≈0-t_S, and therefore, with a view to the worst case, we will assume that c = 1 in what follows.

Now we can determine the average length of the process segment, in other words, the average delay of the process at vertex C(m) in time segments, via the formula $\overline{D}_m = \sum_x p(m,x)\cdot l(m,x)$. Finally, it is necessary to determine the new probability distribution Q_j that the process will proceed further from C(m) along the j-th arc, where j = 1, 2, ..., m - 1, since entries into loops are already taken into account in D_m. The probabilities Q_j can be determined in the form of a sum of probabilities p(m, x) of those segments that terminate by exiting from C(m) along the j-th arc, i.e.,

$$Q_j = \sum p(m,x)_j \ ,$$

or by the formula

$$Q_j = P_j(1 + P_0 + P_m + P_0^2 + P_0P_m + P_m^2 + P_0^3 + P_0^2P_m + \\ + P_0P_m^2 + P_m^2 + \ldots + P_0^{\sigma} + \ldots + P_m^{\sigma}),$$

w

here a = r_{max}-1. The calculations are made only for modest m values, since, as m increases, the delay D_m →1 and Q_j → P_i, j=i, while P_0, P_m → 0.

Now we can determine the length of the entire specified process that begins at vertex C(m):

$$T_k=\sum D_m+\sum d+\sum T.$$

where =$\sum D_m$ is the sum of the average delays at those vertices that are encountered in the k-th process; ∑d is the sum of the pauses, equal to 2t $_S$, which are encountered in the k-th process in A2-TR transitions; ∑T is the sum of the time intervals that are occupied by all m successful transmissions (in time segments). Here ∑T is the useful part of the process, while

$\sum D_m + \sum d = Y_k$ is the time loss required to implement the access method.

Knowing the losses Y_k and the probabilities P_k for each process, we can determine the average value of the time loss with respect to all processes belonging to C(m), using the formula:

$$Y_m = \sum_k P_k Y_{k_i}$$

where k = 1, 2, ..., is the ordinal number of the process.

Let us determine the time losses for elementary processes in accordance with Fig. 2 for p = 0.5:

m=2 1=C(2)-0T-0T $Y_2=D_2=2.5$

m=3 11=C(3)-0T-0C(2)-0T-0T $Y_{11}=D_3+D_2=4$

21= C(3)- 0C(2)-0T-0T-2T $Y_{21}=D_3+D_2+2=6$

$Y_3=Q_1Y_{11}+Q_2Y_{21}=0.5\cdot4+0.5\cdot6=5.$

m=4 111=C(4)-0T-0C(3)-0T-0C(2)-0T-0T

121=C(4)-0T-0C(3)-0C(2)-0T-0T-2T

211=C(4) -0C(2) -0T-0T-2C(2)-0T-0T

311=C(4)-0C(3)-0T-0C(2)-0T-0T-2T

321=C(4)-0C(3)-0C(2)-0T-0T-2C(2)-0T-0T.

$Y_{111}=D_4+D_3+D_2=1{,}22+1{,}5+2{,}5=5{,}22$

$Y_{121}=D_4+D_3+D_2+2=5{,}22+2=7{,}22$ } 6,22

$Y_{211}=D_4+D_2+2+D_2=1{,}22+2{,}5+2+2{,}5=8{,}22$

$Y_{311}=D_4+D_3+D_2+2=5{,}22+2=7{,}22$

$Y_{321}=D_4+D_3+D_2+2+D_3=7{,}22+2{,}5=9{,}72$ } 8,47

$Y_4=0{,}2855\cdot6{,}22+0{,}429\cdot8{,}22+0{,}2855\cdot8{,}47=7{,}22$

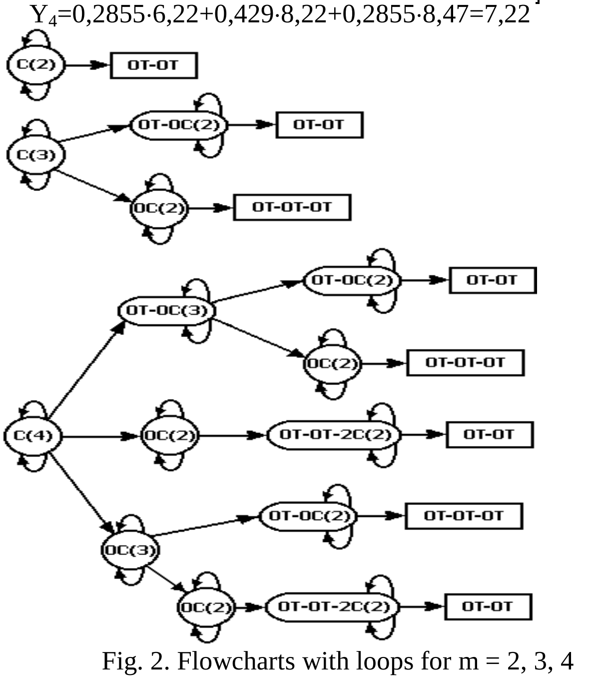

Fig. 2. Flowcharts with loops for m = 2, 3, 4

Conclusion

The proposed approach provides an opportunity for flexible and simultaneously utilizing of different descriptions at different systems and descriptions levels.

The formalism of stochastic automata allows generalization for successful self-organization as some general start program, which is capable of operating in the determined conditions, improving system behaviours, by increasing quantity of contenting in media and its subsystems information.

Formalism of stochastic automata, which is functioning at group, provides possible specifications of "critical time systems" or "real time systems" definitions as self-organization time systems.

Acknowledgements

This research has been supported by Riga Technical University and has been performed in the framework of the projects funded by the Latvian Council of Science (09.1541, 09.1237 / 09.1345). This work has also been supported by the European Social Fund within the framework of the project "Support for the implementation of doctoral studies at Riga Technical University".

References

[1] AsbyW.R., Design for a Brain.London,1952.

[2] V.Zagursky, A.Morozov, Macromodel for uncertainty estimation of mixed-signal system elements under extreme operating condition,Proceedings of VIII International Conference, ISBN 978-5-91450-027-3, "..System Identification and Control Problems" SICPRO-09, Moscow,January26- 30,2009, pp.821-827.

[3] C..W. Yi, P.J. Wan, X.Y. Li, and O. Frieder, Asymptotic distribution of the number of isolated nodes in wireless ad hoc networks with Bernoulli nodes, *IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, Louisiana, Procreedings pp.131-135, March 2003.

[4] J.Hong, S.Lu, D.Chen *and oth.* Towards Bio-Inspired Self-Organization in Sensor Networks:Applying the Ant Colony Algorithm, 22nd International Conference on Advanced Information Networking and Applications,2008, 05:01:05 EST from IEEE Xplore pp.1054-1062.

[5] V. Zagursky, Dz. Zibin, Random Multiple Access Method for Wire and Wireless Local networks, 17-th International Conferenceon Computer Theory and Applications, ICCTA 2007, Egypt , Alexandria, 1-3 September, Proceedings, 2007g., pp137-141.

[6] Я.Цыпкин. Основы теории обучающихся систем. Москва. Наука.1970.

[7] T.Collier,C.Taylor,Self-Organization in Sensor Networks, Elsevier Science 10 December 2003.,pp.1-17.

[8] A.Srivastav,A.Ray, Self-organization of sensor networks for detection of pervasive faults, Springer-Verlag London Limited 2009, DOI 10.1007/s11760-008-0101-4.

[9] F.Yu, T. Wu, and S.Biswas, Toward In-Band Self-Organization in Energy-Efficient MAC Protocols for Sensor Networks, IEEE Transactions on mobile computing,Vo.7,No2, February 2008,pp.156-170.

[10] В.Дружинин. Д.Канторов, Проблемы системологии, С предисловием В.М. Глушкова , Радио,1976.

[11] E.Asarin, O.Maler, Scheduling with timed automāta.Theor. Comp.Sc.354.2(2006), pp272-300.

[12] E.Asarin, O. Maler, Time optimal control for timed automāta, Proceedings of the 2th International Workshop on Hybrid Computation and Control, Springer,1999,pp.19-30.

[13] P.Bouy, U. Fahrenberg , G. Larsen, N. Mark, Quantitative analysis real-time systems using priced timed automata. Communications of the ACM september 2011 | vol. 54 | no. 9, pp.78-87.

Valerijs Zagurskis, received his M.S. in computer science in 1965 from Riga Technical University (RTU) and his Candidate of technical science (Ph. D) in circuits and systems in 1972 from the Latvian Academy of Science, Doctor of Technical Science in 1990 from the Ukrainian Academy of Science and Dr.Habil.Comp.Sc. in 1992 from the University of Latvia. He is a professor at RTU and head of the department of Computer networks and systems technology (DTSTK), as well as a member of the IEEE and ACM. He also is an expert in the Latvian Council of Science. His research interests include networks, mixed signal system design, MAC protocols, resource scheduling, cross-layer design, and joined job of systems, wireless ad hoc and sensor networks.

Faculty of Computer Science and Information Technology
Riga Technical University

E-mail: vzagursky@gmail.com

Dmitrijs Bliznuks, received M.sc.eng. degree from Riga Technical University in 2008. He currently is a doctoral student at Riga Technical University. He has been holding the researcher position at Riga Technical University since 2009. His research interests include wireless networks and computer based control.
E-mail: dmitrijs.bliznuks@rtu.lv

Romans Taranovs, received M.sc.eng. degree from Riga Technical University in 2009. He currently is a doctoral student at Riga Technical University. He has been holding the assistant position at Riga Technical University since 2009. His research interests include microprocessors, embedded systems wireless sensors networks and computer based control

Valerijs Zagurskis, Dmitrijs Bļizņuks, Romāns Taranovs. Pašorganizacijas princips laika kritiskos uzdevumos. Rakstā tiek pētītas problēmas saistītas ar laika kritiskām sistēmām. Šī jautājuma kodolā ir dinamiskā resursu izdalīšana un sistēmas funkcionālas izmaiņas laikā. Plašs pētījums ir kritisks priekš sistēmas plānošanas, veiktspējas novērtēšanas, laika sinhronizācijas, resursu piekļuves vadības un sistēmas infrastruktūras. Šajā rakstā arī mēs apskatam formalizētus stohastiskus automātus, kas darbojas grupās. Kā arī tiek piedāvāta kritiskā laika sistēmas jeb reāla laika sistēmas definīcija sakarā ar dinamisko un stohastisko sistēmas uzvedību. Tādas sistēmas funkcionēšana tiek determinēta izmantojot funkcionālus, morfoloģiskus un loģiskus aprakstus. Funkcionēšanas procedūra var būt specificēta dažādos veidos, tādos kā makro modeļos [2], virzīts grafs vai procedūras [4]. Formalizētie stohastiskie automāti ļauj vispārināt veiksmīgu pašorganizāciju, kas spēj uzlabot sistēmas uzvedību, palielinot daudzumu satura informāciju tajā. Rakstā tiek piedāvāts un apskatīts piemērs, kas bāzējas uz konfliktu atklāšanas procesa. Kas ir piedāvāts orientēta grafa veidā, ar kuru tiek pētīti kolīziju aprakstīšanas un viņu iespaids komunikācijas sistēmai jautājumos. Rakstā tiek piedāvāta pieeja kā matemātiski vispārināt augšminētas definīcijas. Šī pieeja var būt izmantota priekš sistēmu infrastruktūras pašorganizācijas vai pieņemot gala lēmumu. Un beigās rakstā tiek prezentēta kritisko laika sistēmu optimizācijas un projektēšanas pieeja.

Валерий Загурский, Дмитрий Близнюк, Роман Таранов. Парадигма самоорганизации в системах с критическим временем функционирования.
В статье исследуются проблемы, связанные с критичными по времени системами. В основе этого вопроса лежит динамическое выделение ресурсов, и функциональные изменения системы во времени. Широкое исследование необходимо для планирования, оценки производительности, временной синхронизации, управлением доступом к ресурсам таких систем. В статье рассмотрен формализм стохастических автоматов, работающих в группе. Представлена возможная спецификация дефиниций «системы реального времени», «системы критического времени» в связи с динамическим и стохастическим поведением систем. Оно определяется функциональным (информационным), логическим и морфологическим описанием, причём процедуры функционирования могут быть заданы как на уровне макромоделей [2] ,так и в виде ориентированного графа процедур [4]. Предложен математический подход для генерализации вышеуказанных дефиниций как время самоорганизации инфраструктуры системы для принятия окончательного решения. Предложен возможный подход к оптимизации и дизайну систем с критическим временем принятия решений. Формализм стохастических автоматов позволяет обобщить успешную самоорганизацию, которая улучшает поведение системы, повышая количество информации. В работе предложен пример, которой базируется на процессе разрешении конфликтных ситуаций, который представлен в виде ориентированного графа, с помощью которого исследуются коллизии, а также их влияние на функционирование коммуникационной системы.