

Cognitive Systems for Monitoring: Architectural View

Alexander Vodyaho¹, Evgeniy Postnikov², Alexander Ekalo², Vasilii Osipov³, Nataly Zhukova³, Michael Chervontsev^{*2}

¹*Saint Petersburg Electrotechnical University, Saint-Petersburg, 197022, Russia*

²*Research and Engineering Center of Saint Petersburg Electro-Technical University, Saint-Petersburg, 197022, Russia*

³*Saint Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, Saint-Petersburg, 197022, Russia*

ARTICLE INFO

Article history:

Received: 22 January, 2019

Accepted: 08 May, 2019

Online: 24 May, 2019

Keywords:

*Cognitive Systems,
Cognitive Monitoring System,
Architectural Cognitive Approach
Design of Monitoring Systems*

ABSTRACT

The modern stage of information technology development is characterized by the acute need to use cognitive technologies for the solution of tasks of practice, in particular, in the sphere of real world objects monitoring and management. Practical usage of cognitive systems of monitoring is significantly limited to two factors now: operation at the level of knowledge leads to loss of speed, and the high complexity of software solutions leads to increase in cost of development. In order to solve these problems it is suggested to use architectural cognitive approach to design of systems of monitoring. In the article the concept of cognitive monitoring is defined. A new approach to creation of cognitive systems of monitoring which functioning is based on use of model of a target object and the model of monitoring system presented in terms of knowledge are proposed. The idea of this approach is generation of the loaded architecture according to these models. The generalized structure of cognitive system of monitoring is given, the concept of the cognitive monitoring machine, which basic elements are the subsystem of creation of models, a subsystem of transformation of models, a subsystem of processing of models, a generator of architecture, a generator of reports and reactions, a repository and a set of adapters is entered. The structure of the platform focused on realization of this approach is described. The example of cognitive monitoring system is given.

1. Introduction

Cognitive information technologies begin to get into everyday life actively. More and more widely such smart devices as smart phones, the smart TV, the smart house, the smart systems of monitoring of a condition of the patient, etc. are used. Fast extension of scope of the Internet of things (IoT) leads to appearance of the heterogeneous network structures, huge by the size, including very large number of the various elements using a large number of various stacks of protocols. At the same time the huge volume of structured and unstructured data is created, and the structure of such systems permanently changes. Efficiency of the control of such huge information systems (IS) with permanently changing structure of interconnections and device types significantly depends on the evaluable information about the IS state, i.e. the solution of the task of monitoring. Existing systems for monitoring are not always applicable in the considered

conditions. Essentially new paradigms of monitoring are strongly required. They can be built on the base of cognitivity concept.

The term cognition comes from Latin *cognito* that means I think. In particular, in the Cambridge dictionary the term cognition is defined as rational cerebration. In case of such determination emphasis is placed on how a human studies, remembers and argues, but not on discrete facts. The term cognitivity came from philosophy, and then began to be used also by psychologists. In particular, cognitive psychology studies how a human think. The concept of cognitivity is very closely coupled with a concept of artificial intelligence (AI). In essence, the AI IS allow understand a situation, to study it and based on it to make decisions on those actions which are required to be undertaken. The classical AI systems can solve those problems which can be accurately formulated and programmed. From this point of view the cognitive systems (CS) are capable to realize processing just as it is done by a human, so CS can be considered as extension of traditional AI IS.

*Corresponding Author: Michael Chervontsev, Email: chervontseff@mail.ru

One of the main goals of CS development is minimizing the semantic gap between the human and machine i.e. computer acts as a human. In order to reach this effect a computer must be able to work with knowledge (to mine knowledge, to process knowledge). Modern software platforms as a rule do not suggest effective support for knowledge processing and there are limited number of analyst, architects and programmers who can solve problems in terms of knowledge. So, we have a gap between cognitive architectures and modern software development platforms. Another problem with CS is that their implementation requires logical inference, knowledge processing, etc. This leads to serious problems with the speed of processing in CS and as a result limits the scope of CS usage.

The article suggests a new approach to cognitive monitoring. It is an architectural approach that assumes usage of target system models which can be used for generation scripts for cognitive behavior realization. Models can also be used for generation of "loadable architectures". Loadable architecture is peaces of code to be loaded into processors of distributed cognitive monitoring systems. In the second section of the article, an analysis of modern CS, the levels of their cognitivity and approaches to CS systems implementation are analyzed. In the third section the solved problem is formulated. In the fourth section basic principles of cognitive monitoring systems are considered. Generalized structure and the models of the systems are described in the fifth and the sixth sections. The architectural approach to cognitive monitoring systems development is presented in the section seven. In the last section an example of practical use of the proposed approach for systems of operational management of networks of cable television is given.

2. Modern state of Cognitive computing (CC)

On the idea [1, 2] cognitive systems shall differ essentially from traditional IS, including the AI IS. They shall not be programmed rigidly, and shall study like the human in the course of functioning and communication with people, i.e. over time they shall function more and more effectively [3]. According to this paradigm the CS shall be based not on the algorithms and rules which are a priori put in them, but shall be based preferentially on training [2]. It is expected that usage of cognitive technologies can expand abilities of IS and allow them solve problems which are traditionally solved by human. It can be such problems as planning, reasoning, training, operation with incomplete and doubtful data, etc. [4].

CS can have the following properties:

- Ability to self-learning. CS shall minimally use traditional "hard" programming. They shall acquire permanently knowledge from the environment. At the same time CS shall create and check permanently hypotheses, at the same time learning goes preferentially in the mode "without teacher".
- Ability to adaptation. CS shall have ability to adapt both to specific features of the problems to be solved (content), and to the changing environment (context).
- Dynamism. The dynamism can be defined as ability to work in the real time mode.

- Interactivity – ability to interact with different stakeholders for the purpose of obtaining knowledge from them.
- Ability to process unstructured data, in particular, the data provided in a natural language.
- Ability to scaling. Ability to add auxiliary resources in order to improve performance.
- Ability to manage own structure. Assumes a possibility of self-diagnostics, reconfiguring in case of failure of separate elements, solving problems of performance optimization, security management, etc. [6].

To estimate the level of cognitivity of IS one can use discrete or continuous measures of approximation (maturity model) which, in particular, can be defined in terms of levels of a maturity of the technologies. These levels define what cognitive opportunities and in what order can be added in case of creation of an IS, forming some road map which defines transition from traditional IS to cognitive IS. It is possible to define 2 alternative approaches to delimitation of a cognitive computing: to define them in terms of ability to solve problems just as it is done by a human or in terms of methods of the decision of tasks.

The first approach assumes comparing of behavior of the machine and the human in case of the decision of target tasks. For example, if an expert is not able to define with confidence, who solves a problem: a human or a computer, then it is possible to speak about cognitive computing. It is possible to define, how intensively "human" approaches to problems solving is used. For example, for service oriented IS the level of cognitivity can be defined by means of calculation of what part of services can be considered as cognitive. The problem consists that how the human solves the problem is not always known. It should be noted that for practice such approach is of no use.

The second approach according to which cognitive computing is defined by the used architectural concepts is of bigger practical interest. When using this approach, cognitive computing can be defined as a family of architecture solutions which use models of the environment and model of the system, and these models shall be built in terms of knowledge. Taking into account the definition of the term software architecture [5] where architecture includes the IS development, then for cognitive IS this definition is to be expanded and formulated as a family of architecture which in the process of operation and (or) in the process of development use the models of the environment and model of the IS provided in terms of knowledge. If cognitivity is implemented only at a development stage, then existence of tools for transformation of cognitive solutions for class IS in not cognitive solutions for a concrete solution is required. Such approach can be useful in cases when use of cognitive approaches in the run time mode is not possible because of strict requirements on speed and (or) on available resources.

Implementation of CS is based on existing approaches to creation of the AI systems. In [7] 4 alternative approaches to creation of the AI systems are defined: IS which think as a human, IS which think rationally (reasonably), IS which operate as a human, IS which operate rationally (reasonably). It is necessary to mark that concepts to think or to operate as a human and to operate rationally are different things.

The existing CS use all 4 approaches. In particular, in order to understand how a human solve problems, it would be useful to analyze what types of mistakes are made by a human and by a computer when they solve similar problems. If they make similar mistakes, then they thing in the same way [8]. It is obvious that this approach is more useful to physiologists, but not IT specialists as for last it is important that the IS be able to solve problems with the required characteristics of quality. The more important is that when a human thinks, he or she operates with knowledge. Therefore from creation of the CS as a system which operates with the models in terms of knowledge is more useful for IT practice. The detailed and fresh analysis of the state of the art of CS relating to 2018, one can find in [8].

The CS commonly integrates many technologies and platforms thus it can be considered as integration technology. So CS is closely coupled with MAS. MAS is a multidimensional concept. It can be considered, at least, from 3 points of view: i) from the point of view of the theory, MAS can be considered as an approach to creation of systems of the distributed AI, ii) from the point of view of the application programmer, the MAS can be considered as separate architectural style or as a combination of several architectural styles [9], iii) from the point of view of the system programmer, the MAS is a hardware-software platform.

3. The problem of cognitive monitoring systems creation

One can define monitoring as a process of acquisition data about the current status of a target system (TS), data processing and (or) transformations for presenting needed information to stakeholders in a proper form, particularly to persons who makes decisions (DMP). The process of monitoring can be conceded as a business process which can be created both in statics and in runtime.

The monitoring system (MS) can be defined as systems which realize monitoring process. MS can be realized both as stand alone IS or IS subsystem. In the latter case we have self monitoring. Results of monitoring are used for solving problems of target system (TS) management (diagnostics, self-repair, optimization of performance, reconfiguration, etc. [10], [11]).

One can conceder MS and TS as a single system, or a TS can be considered as a context in which MS operates. The choice of approach depends on specifics of the problem to be solved. If the monitoring subsystem cannot send control information to TS, then it is better to conceder TS as an environment. If MS can manage TS, then we have monitoring control system (MCS). As a rule, TS has connections with other objects and IS. The set, consisting from target object and related objects forms the TS. Both TS and MS may include a program component (business logic) and the infrastructure component (IC). IC is sensors, the executive mechanisms, servers and network equipment, etc. The generalized structure of MS is shown in fig. 1.

TS can be defined as S, EM, EQ, BP, M, D, where: S – set of sensors, EM – a set of the executive mechanisms, EO – a set of elements of the equipment, BP – a set of the business processes which realize business logic, M - a set of signals (messages) which are used for information interchange, L - records in log-files about events like beginning and the end business process, D - sources of signals (data) which are generated in the context of specific

business process. Message has a format: BP, T where BP – the identifier of business process, T – time tag, each time tag has a format N, V, T, where N is a parameter name, V – a parameter value, T – a time stamp.

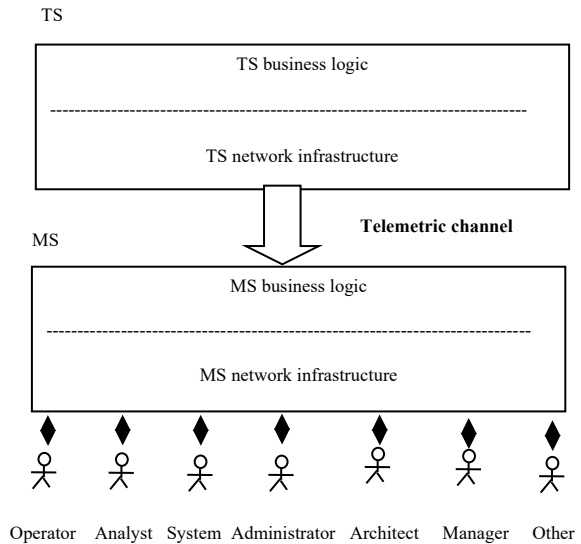


Fig. 1. MS structure

At the top level MS can be classified by three parameters: structure of TS (centralized, distributed), structure of MS (centralized, distributed) and processing method (postponed, on line).

MS can be presented in the form shown in fig. 2.

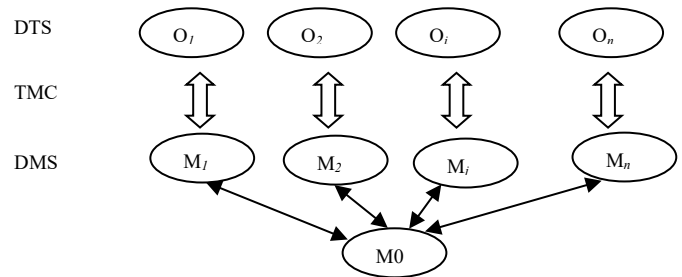


Fig 2. MCS structure

This variant corresponds to a case when TS is a set of distributed objects (DTS). It can be, in particular, mobile objects. MCS also is realized as distributed system (DMS). In fig. 2 M are modules, a tree topology of MS is shown, but also other topology can be used. DTS and DMS can be connected by means of several channels of telemetry (TMC).

It is possible to define the following main problems solved by MS:

- The analysis of telemetry in the postponed mode. On an input of MS the event stream in a format time-name-value arrives. It is necessary to define time point or an event since which deviations are observed and to trace history of development of the situation.
- On line processing of telemetry and TS control. On an input of MS the event stream in a format time-name-value arrives. It is required to check, if the message comes in a proper time and parameters are in norm it means that

situation develops in a proper way. If the events show that the situation has deviations from the sample one, then procedure of correction is to be started. It can be a procedure of change of settings or reconfiguration by means of sending message of name value type.

- Network infrastructure of MS management. If MS is realized as distributed system, the task of monitoring and control of MS own network infrastructure appears. The task is similar to the previous task.
- Debugging of a process of operation with telemetry. Assumes carrying out different experiments at a stage of debugging of procedures of monitoring of real MS.
- Simulation of operation of MS. Carrying out different experiments at a stage of primary debugging of MS.

For solving the enumerated problems monitoring systems should be able to solve such “human” tasks as find and usage of the most informative sources, reasonable distribution of existing resources, observed objects modeling, etc. Also MS should flexibly respond to changes in internal and external monitoring conditions. For this cognitive monitoring systems are required. To build such systems appropriate models and architectures should be developed.

4. Basic principles of cognitive monitoring systems

The Cognitive Monitoring Systems (CMS) can be defined as the monitoring systems realized on the basis of the principles of cognitive computing. It should be noted that most often usage of CMS makes IS cognitive. It, in particular, concerns the systems constructed on the basis of the concept of IoT. Cognitivity is not inherent feature of IoT, but CS are often built on IoT platform. It is possible to speak about 3 aspects of implementation of the cognitive monitoring (CM) concept. In this case it is necessary to answer 3 questions:

- How cognitive processing is organized?
- How the functionality is distributed between the CMS subsystems?
- How the network media is organized?

The answer to the first question consists in the description of what mechanisms of cognitive processing are used in the case of creation of MCS and by means of what mechanisms they are implemented. The answer to the second question consists in describing the CMS structure and platform. A CMS can be realized as a system with the client-server architecture, as a distributed fog system, on the base of multi-agent platform, etc. It should be noted that sensors can be cognitive also. The answer to the third question assumes the description of how the monitoring data communication media is organized. In view of the fact that modern TS more and more often represent as a set of distributed mobile objects, usage of a communication media based on ideas of Cognitive Internet of Things (CIoT) quite evident.

The classification of possible variants of CSM organization is given in fig. 4. The approaches can be classified by 3 main features: from the point of view of used structural and architectural concepts and from the point of view of used models. From the point of view

of architectural solutions, it is possible to select 4 basic alternative approaches: use of classical solutions, use of vertical solutions, use of horizontal solutions and use of hybrid solutions. Traditional solutions assume, for example, use of solutions on the basis of SNMP and cognitive processing can be executed on the server. Vertical solutions are multi-layer solutions in which cognitive processing is implemented at several levels. Examples of such approach is fog computing. Horizontal solutions assume one level organization. In this case cognitive processing is implemented by a group of the processors working at one level. As a rule, it is P2P systems. Service oriented architectures and multi-agent solutions can be conceded as examples of horizontal approach. Hybrid solutions are a combination of horizontal and vertical approach. MAS working on the several levels can be conceded as an example of this approach.

CMS assumes use of 2 main types of models: TS model and own model. If the dynamic (compiled) architecture is used, then it is possible not to use models in loadable modules. In CMS only separate model can be used, for example, only TS model. The models can be either static, or dynamic. Static models are generated in the process of the CMS development. Dynamic models can be created and (or) changed in run time.

In the process of CMS development different architectural concepts can be used. In the simplest case it can be the fixed architecture. It can be either an architecture designed manually or the architecture generated from architectural meta models according to an approach which will be described below. When using the virtual environment use of the loaded (on demand) architectures from architecture library is also possible. The most effective and difficult solution is use automatically generated dynamic architecture which can adapt both to content, and to context.

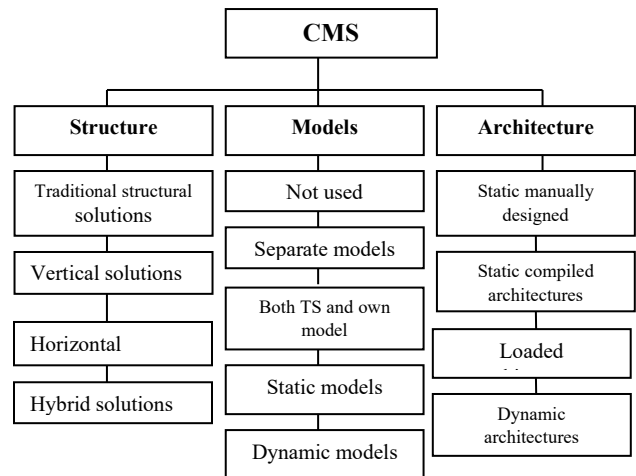


Fig. 3. Classification of approaches to CMS implementation

5. Generalized structure of cognitive monitoring systems

The generalized structure of CMS is shown in fig. 4. The TS can have the built-in sensors which can realize separate elements of cognitive processing, as a rule at the level of signals. The core element of CMS is cognitive machine which is responsible for building required models, their maintenance in actual state, transformation of models and realize decision-making on models. Communication between the cognitive machine and the field of

cognitive sensors can be realized with the help of a cognitive network [8].

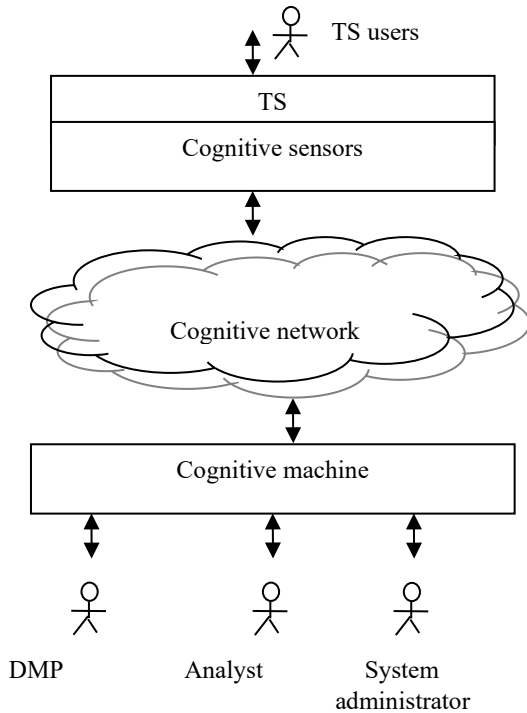


Fig. 4. Generalized structure of CMS

The basis of CMS is the cognitive machine of monitoring (CMM) which can be considered as an architectural framework as the platform or as CMS toolkit depending upon the point of view. CMM consists of a set of services of different level which can be used for creation of concrete CMS. The conceptual structure of CMM is shown in fig. 5. CMM includes 7 subsystems: Model Mining Module (MMM), Model Transformation Module (MTM), Model Processor (MP), Architectures Generator, Generator of Report and Responses, Repository and a set of Adapters.

MMM is a module which is responsible for creation and maintenance of models in actual state. MTM is a module which is responsible for model transformations. PM is a module which is responsible for processing of models, in particular realizes reasoning on models, the generator of architecture can generate loadable modules and is used in CMS with dynamic architecture. The repository is used as storage for models, data, knowledge, scripts, etc. The report generator and responses performs several functions. If a CMS operates in MS mode this module presents information about TS status to DMP. If the CMS operates in MCS mode, then it sends controlling impacts to TS. In the case of use of the dynamic loaded architecture, this module is responsible also for loading of architectural modules in distributed environment of CMS. Adapters are used for information and knowledge search in external sources.

CMM operates in a following way. Data from sensors come to the CMM machine. If the model is unknown or requires specification and (or) verification, then for this purpose MMM used. The standard mode of operation assumes permanent checking of models on correctness and reorganization of models in case of arrival of messages about events. All models are stored

in the repository. In CMM main decisions are made as a result of logical inference on knowledge base. For this purpose it is required to mine knowledge from an input data stream and external sources. This procedure realizes MMM which integrate a number of lower level services. MMM can mine knowledge from data and events. Processing in MMM can be realized according to well-known JDL model [12]. MP realizes functions of making different operations with models, by the most part by means of SPARQL [13] requests.

Such solutions are relevant first of all in the cases when TS and (or) a connective network is built as IoT. If the structure of the serviced TS is known and stable then loadable modules can be stored in repository libraries. The use of the loaded (on demand) architecture [14] is relevant, first of all, for fog structures when resources of modules of the intermediate level are restricted.

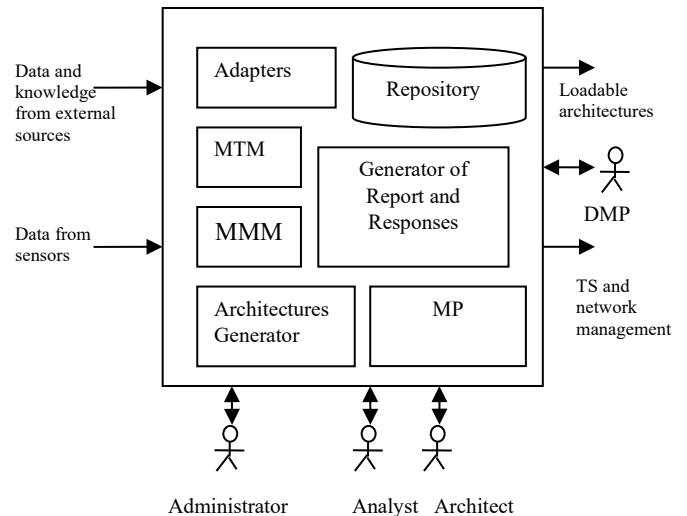


Fig. 5. Cognitive machine of monitoring structure

6. Models and models transformation in cognitive monitoring systems

As it was stated above, cognitive computing and, respectively, cognitive monitoring, assumes the widest use of models, including the models provided in terms of knowledge. Nowadays creation of models in the case of design of the IS undoubtedly is a bottleneck and in the most cases continues to be executed manually by analysts. This activity needs both big expenses and does not exclude errors. If the structure of the IS to be modeled permanently changes, then the task of creation of adequate models becomes even more difficult. Thus, the problem of automatic formation of models is rather actual one.

In relation to CMS models shall provide the solution of the following main problems: estimation of a current status of TS, prediction of TS behavior, to estimation of CMS own condition, prediction own status, synthesis of architectural models from the architectural description. For CMS models in terms of knowledge are of prime interest. First of all, it is ontological models.

Nowadays different methods of receiving models from data and information on events are known. In our context at least 3 technologies are of interest: extraction of knowledge from data (data mining) [15], creation of models of processes from log-files in the form of Petri nets (process mining) [16], creation of automata, in particular, multilevel automata models [17]. Today it

is separate loosely interconnected technologies. It is advisable to consider these technologies as a single technology, which can be called Model Mining (MM). MM can be conceded as umbrella technology and can be determined as $MM = \{ Mod, Met, Tmet, I, R \}$ where Mod – models, Met – methods of a mining of models from data and messages about events, Tmet – methods of transformation of models, I – instruments of models mining of, R - an implementation of models. The MM can be considered as integration technology. Usefulness from introduction of this concept is that it allows: i) to expand a concept of Model Driven Engineering (MDE), ii) gives the chance to look at a problem of automatic synthesis of models integrally, iii) allow understand what elements which are necessary for realization of systems of automatic model synthesis are absent, iv) allow accumulate knowledge in the form of models (model knowledge).

According to the MDE approach to describe a system a stack of models is created. The key concepts of the approach are the following: model (M), meta model, (MM), model wiring (MW) and model transformation (MT). Representation of a set of models in the form of a stack needs to determine the procedures of transition from model of the top level to models of the bottom level and vice versa. Process of model transformation is a process which defines how it is possible to receive target model from one or several initial models. Models can be transformed either horizontally, or vertically. In the first case the model is transformed to model which belongs to the same level. In the second case the transformed model is model of higher or lower level. The special type of models - VM are used for binding of models of different types. The basic goal of these models consists in establishment of compliance between separate elements of models (low level binding) [18].

In relation to CSM both horizontal, and vertical transformations of models are of practical interest. Not always it is possible to realize transformations procedures automatically and to receive needed model. For example, a service allows receive results of a mining in the form of the multi-level finite state automata, but MP can work with ontology, and for generation of executable code UML description is needed.

7. Architectural approach to cognitive monitoring systems development

In the process of CMS development an architect faces at least with two problems.

The first problem is that very often MS must operate in real time. In relation to CMS the main problem is that implementation of intellectual elements of behavior requires manipulation with knowledge. Manipulations with knowledge, assumes implementations of a logical inference, i.e. execution of a large number of machine operations which cannot always be executed in parallel mode that makes difficult to use CMS not only in hard real-time systems, but also in online IS with moderate requirements to speed of operation. Besides, it is not always possible to predict time required for realization of a logical inference precisely. It is possible to solve this problem due to increase of performance of the used servers and (or) usage of parallel algorithms. All these approaches, at least, when using the modern platforms, give effect only in the case of very moderate requirements to real time.

The second problem is that CMS realization assumes implementation of mechanisms of knowledge processing which are most often provided in the form of ontology. It is necessary to mark that development of such type of IS requires from the analyst and the programmer rather high qualification and experience or it is necessary to invite knowledge engineers, i.e. the process of CMS development becomes rather expensive.

For solving these problems architectural cognitive approach to CMS development is suggested. The idea of this approach consists in development of the domain oriented platform for CMS which includes the architectural framework (AF) and tools for development on its basis of concrete CMS. AF includes a set of services. The generalized structure of the platform is shown in fig. 6.

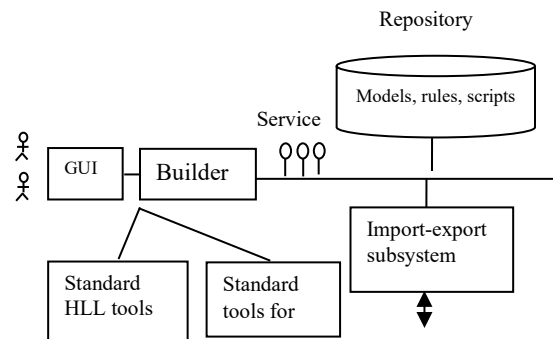


Fig. 6. Cognitive machine of monitoring structure

The platform represents a set of tools for architectural development of CMS. The platform consists of the following elements: a repository for models, rules, scripts and data storage, a data, information and knowledge import export subsystem, a set of services. Besides, the platform encapsulates toolkit for CMS architectural development – builder, with the help of which one can build models and business processes of monitoring information processing. Besides it standard high level languages (HLL) design tools and object-oriented modeling (OOM) can be used. The basis of the platform are services, such as: services of extraction rules from data (data mining), services of automatic creation of models (model mining), services of models transformation, services of generation of modules of the loaded architecture, models finding services, engines services, services of access to data and knowledge models access, etc.

The platform can support model transformation, i.e. generation of automata or neuron network solutions from cognitive models. For example, when we have strict requirements on time, use of automata may help to solve problem. In the case of context is changed then new automate can be generated and loaded as module. For loading of modules it is possible to use separate temporal slots or it can be done when model mining subsystem defines the model is to be corrected. In this case one can speak about the adaptive (agile) architecture in a sense [19].

In general, the described above approach to use of models can be defined as Cognitive Supported Architecture (CSA). Perspectives of practical application of this approach are defined by accessibility of services of transformation which allow use instead of heavy models in the form of knowledge other more light models such as automata machines, tables, decision trees etc which

allow receive higher speed of computations. Such approach allows use the platform as a modeling and development tool. The offered approach is, first of all, the integration approach directed to integration of already available technologies and solutions. Now the platform is in a development stage.

8. Use case. CMS of resources on networks of cable television

Let us concenter as an example of practical use of the described earlier approach for system of operational management of networks of cable television.

Generalized structure. The Systems of a Cable Digital Television (SCDT) are one the traditional directions of telecommunication technologies. The modern SCDT are sophisticated distributed IS with hundreds of thousands of subscribers, which include the client side and sever side equipment. The server side equipment is installed in Data Acquisition and Processing Centers (DAPC) of operators. Server side equipment is, as a rule, powerful servers clusters which realize procedures of client side equipment monitoring and measurements. The network is divided to segments. Each segment has its own segment server (SS). The receiver of a digital television (the TV-tuner, the Set-top box, STB) which provides basic and expanded functionality of the TV set is installed on the side of the subscriber.

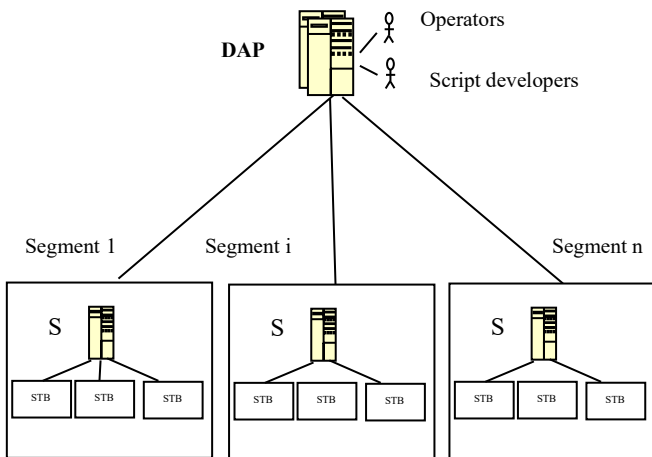


Fig. 7. Structure of SCDT

For management of SCDT the systems of Operation Management System (OMS) which standard structure is shown in fig. 7 are traditionally used. Command centers realize monitoring of STB of a network for the purpose of determination of their current status and provide technical support. Besides, the solving this task, OMS realizes functions of information distribution and management of computational resources.

OMS functionality. Standard OMS solves 2 groups of tasks: the tasks of technical support of the SCDT and the task of statistics receiving for the benefit of analytical and marketing departments; the continuous monitoring the status of services, operational informing the operator in case failures, registration of originating errors, their identification, localization; assessment of possible ways of elimination, operational debugging, detection of origins of problem situations, the analysis of a situation in the network in

general or in separate subnets, detection of dependences between originating errors on different components, dependences of origin of errors and network condition, actions of the user, the forecast of origin of errors, early diagnostics and preventing of appearance of errors, etc. The main OMS task is provision to operators of information on a status of SCDT, detection and localization of the failures.

The standard requirements to OMS. The main requirements to the modern and perspective OMS are about a possibility of operation with big data, flexible logic of operation (a possibility of adaptation in real time to an environment status, actions of users), the minimum requirements to technical characteristics of STB and parameters of a network, high rates of readiness. At the same time OMS shall be updated quickly taking into account the new realized logic; new needs of analysts for solving analytical tasks.

Typical problems of OMS development and usage. In the process of OMS operation the high dynamism of the environment takes place. Work loading of a network and devices is permanently changing depending on behavior of users and operability of technical means. In this situation it is very difficult to forecast the SCDT future status. As a result, the situation when additional loading from a monitoring system leads to system failure is possible. In the course of OMS functioning there is always a danger of origin of effects of "avalanches" when in case of origin of a malfunction there is the avalanche increase in traffic between the local server and STB caused by repeated attempts of receiving and sending data.

OMS developers often face the following problems: i) the long and sophisticated cycle of software debugging; the cycle of testing and delivery of the new version of a software takes about 6 months that makes impossible to change program code often because of complexity of process and high cost of release of new versions; ii) the logic which is to be realized by the STB is defined by many factors such as network parameters, structure of a network, specific tasks of the concrete operator; as a result, development of many modifications of the systems is required, at the same time there are the limited number of systems of each type is used.

Possible approaches to solving the problem of SCDT monitoring. The task of SCDT monitoring can be considered as the task of establishment of causes and effect relationships on a set of events where the reason is a failure, and the result is an event message which can be either message from STB to the server, or creation of a monitoring artifact created on a server side. The problem is that the structure of SCDT is permanently changing because of appearance of subscribers, appearance of new equipment, etc.

Appearance of the new STB types which, on the one hand, have new functional capabilities and on the other hand, they have new architectural features such as a modified set of statuses, the new message formats, and it is necessary to develop big number of new drivers. Really, in case of appearance of the new STB type it is necessary to develop new diagnostic scripts, thus, a new role appears – a scripts developer that, naturally, leads to increase in the total cost of ownership (TCO).

From the point of view of the SCDT owner is extremely desirable to automate the procedure of scripts development, or, at

least, radically simplify it. One of possible approaches to this problem solution is use the domain specific languages (DSL) [20] for script development. This approach allows to simplify the process but not to automate it.

Ontological approach. The main idea of ontological approach is that the diagnostic script is created on the basis of ontological descriptions of elements, in particular, of STB. When ontological approach is used then when it is necessary to add new element type to SCDDT the script developer makes its ontological description and adds it to working ontology. The ontological description is used for generation of automata model.

Ontological approach can be considered as a superstructure over other approaches. It, first of all, concerns automata approach. The ontology in this case can be conceded as an instrument of dynamic structures description. Ontological approach can be also used together with model approach. In this case for scripts generation instead of object models ontological models are used. It is obvious that the systems which use build-in ontology cannot show high performance but for this case it is not very important. Use of ontological approach, on the one hand, simplifies operation of the analyst because it allows build ontological descriptions of elements on the basis of existing,

but on the other hand, requires from the analyst to have the skills of ontologies usage.

Diagnostics system operation algorithm. The generalized algorithm of system operation of monitoring includes 4 steps:

- Step 1. Algorithm execution is initiated in case of origin of an erratic situation in STB
- Step 2. Parameters which values are required for bug fixing in operation of a receiver are defined.
- Step 3. Sending to STB of the requests for obtaining parameters with the help of standard commands is carried out. A command execution result is information messages with parameter values or log-file. The analysis of the received values is made; if necessary, additional requests are made.
- Step 4. On the base of received data the way of elimination of failure is defined and implemented.

In article [21] the formalized algorithm of diagnostics and elimination of erratic situations in STB is described. The mathematical apparatus for solving the problem of automatic scripts synthesis, models and indices of efficiency are suggested. The stage of determination of parameters of a status of STB which are necessary for identification, localization and elimination of an erratic situation is in details considered. The algorithm of automatic synthesis of programs of additional parameters acquisition from receivers is offered. The algorithm is constructed on the basis of conditional finite state operational automata. For solving the problem of diagnostics the models which describe receivers, the environment in which they function, actions of the users are used.

The set of models includes the following models: function models which describe the functions provided to users, model of program components which describe the software of a receiver,

models of parameters of components which determine parameters of a status of software components. The set of models includes also the following main models: i) the model of platform-independent parameters, ii) models of platform-dependent parameters which are defined by STB model type, iii) context-sensitive models (model of the environment and methods of use of STB) which define a context of formation of algorithms of monitoring, iv) model of a data communication network, v) behavior model of the user, vi) status model of STB and their components, vii) status models of STB describe a receiver status for the given context according to the volume of the supported user functionality, viii) model of erratic situations. All listed above models in details are described in [20].

Author's experience in development and use of the described system shows that despite limited cognitive opportunities, use of the cognitive approach based on application of models gives a certain positive effect from the point of view of minimization of TCO. System testing shows that it allows create rather effective diagnostic scripts.

It is necessary to mark that described above OMS realizes rather low level of a maturity from the point of view of cognitivity, in particular now self learning mechanisms are not yet realized. Authors try to do the best to realize this facility in the next system version.

Conclusion

It is possible to claim that today cognitive technologies reach the sufficient level of a maturity for use at design of real systems. Not least it concerns the CS of monitoring. In many cases existence of a cognitive subsystem of monitoring, allows to concede the IS as a CS.

But there are a number of moments which prevents a wide use of CS. The first problem is that for creation CS and respectively cognitive monitoring needs it is necessary to invite high skill developers who can work with knowledge based IS. For many small companies very often it is too expensive. Suggested framework can be used by architects who have limited experience in the field of knowledge based IS development. The second problem under consideration is a problem of CS operation in real time mode. For solving this problem it is suggested to use loadable generated architectures.

The further direction of research will be pointed on integration of suggested architectural framework with existing architectural framework and further investigations in the field of automatic architectures generation.

References:

- [1] Kelly, J.E.: Computing, cognition, and the future of knowing. IBM Research. Oct 2015
http://www.research.ibm.com/software/IBMRResearch/multimedia/Computing_Cognition_WhitePaper.pdf
- [2] Kelley, J.: Smart Machines: IBM's Watson and the Era of Cognitive Computing. Columbia Business School Publishing. Sep 2013 *Kelly III, Dr. John (2015)*. "Computing, cognition and the future of knowing" (PDF). *IBM Research: Cognitive Computing*. IBM Corporation. Retrieved February 9, 2016
- [3] Balani, N. Cognitive IoT (2015) <http://navveenbalani.com/>
- [4] Schatsky, D., Muraskin, C., Gurumurthy, R.: Cognitive technologies: the real opportunities for business. Deloitte review, no. 16, pp. 115–129 (2015)
- [5] International Standard ISO/IEC/IEEE 42010 Systems and software engineering — Architecture description <http://www.iso.org>

- [6] How is cognitive computing different from big data and NLP?
<http://www.coseer.com>
- [7] Russell, S, Norvig, P. Artificial Intelligence: A Modern Approach, 3rd Edition, 2010
- [8] Sangaiah A. K., Thangavelu A., Sundaram V. M. S. Cognitive Computing for Big Data Systems Over IoT. Frameworks, Tools and Applications. Cham, Switzerland. Springer International Publishing AG -2018. — 375 p.
- [9] Bass, L., Software Architecture in Practice. 3rd ed. / L Bass., P.Clements, R. Kazman. — Upper Saddle River, NJ.: Addison-Wesley. 2013. — 661 p.
- [10] Okhtilev M. Yu., Sokolov B.V., Yusupov R.M. Intelligent technologies of status monitoring and control of the structure of complex technical objects. M. Nauka.-2005. — 291 p. (In Russian)
- [11] Beliaev S.A., Vasiliev A.V., Kudriakov S.A. The monitoring of information trends system's architecture based on the free software // Programmnye produkty i sistemy (SOFTWARE&SYSTEMS) №4, 2016, pp.85-88 (In Russian)
- [12] Blasch E., Bosse E, Lambert D. High-Level Information Fusion Management and System Design, Artech House Publishers, Norwood, MA. 2012
- [13] Gasevic D., Djuric D., Devedzic V. Model Driven Architecture and Ontology Development. Springer-Verlag, 2006.
- [14] Sommerville, I., Software Engineering. — Boston, Massachusetts, Addison-Wesley, 2011. -773 p.
- [15] Zaki M., Meira W., Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, 2014
- [16] van der Aalst W. Process Mining. Data Science in Action. 2nd ed., Berlin Heidelberg. Springer-Verlag. —2016
- [17] Osipov V. Yu. Automatic Synthesis of Action Programs for Intelligent Robots // Program. Comput. Software. 42 (3), 2016. Pp. 155 – 160.
- [18] Zivin B. E., Jouault J, and Valduriez P., On the Need for Megamodels. <https://scinapse.io/papers/195085068>
- [19] Babar M. A., Brown A. W., Mistrik I. Agile Software Architecture. Waltham, MA: Elsevier Inc. 2014. -- 392 p.
- [20] Kelly S. Tolvanen J. Domain-Specific Modeling: Enabling Full Code Generation. — John Wiley & Sons, 2008. — 340 p.
- [21] Vodyaho A.I., Mustafin N.G., Zhukov N.A. Ontological approach to creation of monitoring systems of resources on networks of cable television, Izvestiya SPbGETU "LETI" 2017, No. 2, pp. 29-38 (In Russian)