## Conclusions

In this article unenriched logical relationship systems are introduced on the basis of the applied logic languages. Every such a system represents a class of enriched logical relationship systems. Every enriched logical relationship system determines a set of its solution. In this article notions of equivalence, isomorphism, homomorphism, and product are introduced for unenriched logical relationship systems.

## References

[Artemjeva et al, 1996] Artemjeva I.L., Gavrilova T.L., Kleshchev A.S. Logical Relationship Systems with Elementary Objects. In Scientific andTechnical Information, Series 2, 1996, 1: 11-18 (in Russian).

[Artemjeva et al, 1997] Artemjeva I.L., Gavrilova T.L., Kleshchev A.S. Logical Relationship Systems with Parameters. In Scientific and Technical Information, Series 2, 1997, 7: 19-23 (in Russian).

[Kleshchev et al, 1999] Kleshchev A.S. & Artemjeva I.L. Domain Ontologies and Knowledge Processing. Techn. Report 7-99, Vladivostok: Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences, 1999. 25 p.

[Kleshchev et al, 2005a] Kleshchev A.S. and Artemjeva I.L. A mathematical apparatus for ontology simulation. An extendable language of applied logic. In International Journal Information Theories & Applications. 2005.

[Kleshchev et al, 2005b] Kleshchev A.S. and Artemjeva I.L. A mathematical apparatus for ontology simulation. Specialized extensions of an extended language of applied logic. In International Journal Information Theories & Applications. 2005.

## Authors' Information

Alexander S. Kleshchev – kleschev@iacp.dvo.ru

Irene L. Artemjeva – artemeva@iacp.dvo.ru

Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences 5 Radio Street, Vladivostok, Russia

# GRID-ENABLING SATELLITE IMAGE ARCHIVE PROTOTYPE FOR UA SPACE GRID TESTBED[1]

Andrey Shelestov, Alexander Lobunets, Michael Korbakov

***Abstract**: The paper describes practical approach to implementation of satellite data archive using Globus Toolkit 4 components. The solutions consists in converting a hierarchy of remote data files available via FTP into Grid-enabled archive. All etries of such archive will be indexed using arbitrary but pre-defined XML schema. The information will be exposed via MDS4 Index service and the actual data will be exposed via GridFTP. The schema used in our solution is simple enough for understanding but in a real life applications we should use metadata standards such as ISO 19139 and ISO 19115 in particular. A working prototype of the archive described in this paper is deployed on the Grid testbed of Space Research Institute of National Academy of Science and National Space Agency of Ukraine (SRI-NASU-NSAU). The SRI-NASU-NSAU testbed is briefly described in this paper as well.*

***Keywords**: Grid, distributed systems, parallel computing, satellite data, image processing, file archive, programming languages.*

***ACM Classification Keywords**: C.2.4 Distributed Systems:* Distributed applications

---

[1] The work is partially supported by the grant of National Academy of Science of Ukraine for the support of scientific researches by young scientists № 24-7/05, " Розробка Desktop Grid-системи і оптимізація її продуктивності ".

## Introduction

Grid systems are becoming a standard solutions for enabling remote computations execution and distributed data access and processing in environments of the different level of scalability. As it was stated by Ian Foster "The real and specific problem that underlies the Grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.... This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form what we call a virtual organization (VO)" [Anatomy, 2001].

The task of Grid system could be formulated as connecting data, processing powers and algorithms distributed over the network for solving of particular problems. Grid system should be universal up to some degree so these problems should not be hardcoded during its development. Instead a set of problems being solved in a Grid environment must be open for modifications and addition of new problems. This goal is achieved by introducing standard interfaces for communicating between different kinds of Grid resources and clients.

Space agencies all over the world have started working on evaluation of Grid technology in their application area. Space data processing is an important subject domain for application of Grid computing among high energy physics, biology, economics, etc, which produces huge amounts of data. Thus, problems of storing, indexing for quick retrieval on application's demand arise within mentioned above the area. One of the possible solutions of the problem of organization of Grid-enabled data archive is presented below. The development was performed within the framework of UA Space Grid project.

## UA Space Grid Testbed Environment

Among different Grid research tasks the most challenging one is the establishment of production-use Grid infrastructure or at least working prototype. This activity includes creation and maintenance of both technological (development, deployment, testing, evaluation of software components) and political (covering managerial issues) infrastructures.

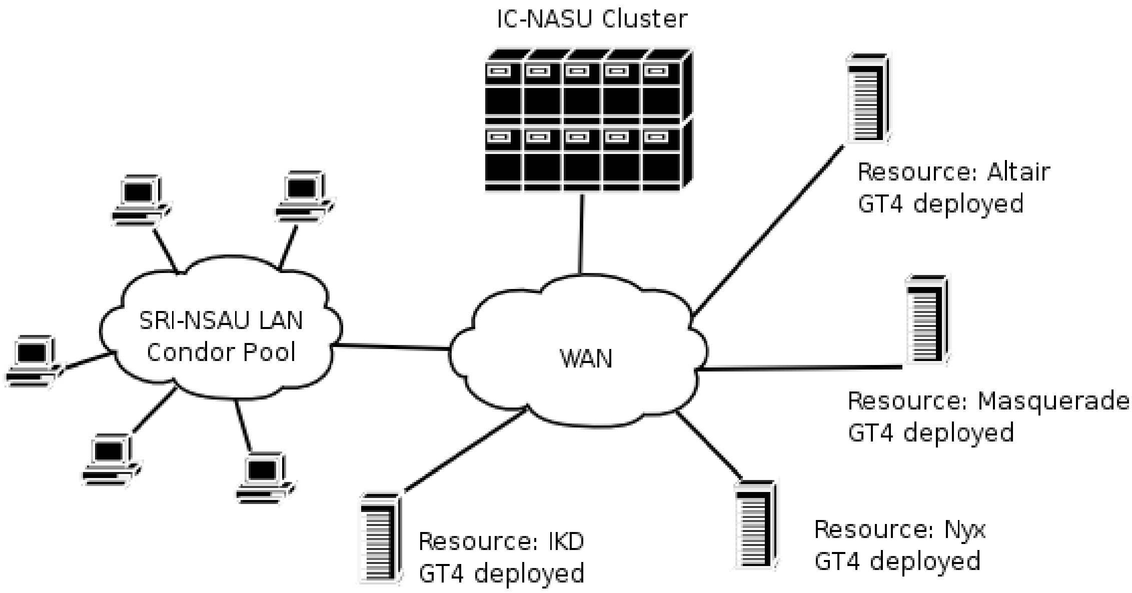


Fig. 1. Current structure of UASG Virtual Organization testbed.

Space Reseach Institute NASU-NSAU (SRI-NASU-NSAU) is developing its own Grid testbed under the draft name UA Space Grid (UASG). The primary focus of the UASG is to integrate available computing resources of NASU institutions such as Institute of cybernetics NASU (IC-NASU) for solving spacial data processing related problems [KSPKK, 2005]. These problems include:

- Data preprocessing (noise reduction, quality adjustment);
- Cloud mask extraction (algorithms are based on Markov Random Fields approach);
- Fractal features extraction (modelling based on fractional Brownian Motion process).

The current structure of UASG is depicted in Figure 1.

UASG represents single Virtual Organization (VO) with a Certification Authority (CA) located in SRI-NASU-NSAU. Its basic computing elements reside in the SRI-NASU-NSAU, IC-NASU and 3rd party's co-locations in Kyiv, Ukraine. Currently most of the resources are dynamic. This perplexes the maintenance activities and requires a flexible and reliable monitoring system deployed over the testbed to provide operational information about resources availability.

The testbed is based on Globus Toolkit 4 (GT4) framework [http://www.globus.org] which is known as the "de facto standard" by numerous agencies. GT4 provides the following components for implementation of Grid infrastructures:

- Security infrastructure based on X.509 certificates
- Data management services (such as data replication, secure and reliable file transfer)
- Execution management services (WS GRAM, Pre-WS GRAM, WMS, CSF)
- Information services (Index service, Trigger service, Aggregation framework)

In this paper we would like to emphasize the WS GRAM (Web Services based Grid Resource Allocation & Management) component [WS-GRAM, 2005]. It provides a basic mechanism for initiation, monitoring, management, scheduling, and/or coordination of remote computations via Web Services based interfaces. A typical remote execution workflow using WS GRAM is depicted in Figure 2.

According to Figure 2 a user signs in into Grid testbed using his/her X.509 certificate, arranges a job description and send job descriptor to WS GRAM service. A general job descriptor contains the name of the executable to be launched remotely, arguments, logging options, files staging in and out options. A WS GRAM service performs file staging options to retrieve all required data for job execution and contacts Local Resource Scheduler (for example, Linux fork command [http://www.linuxforum.com/man/fork.2.php], Condor scheduler [Condor, 2005], etc) which performs the actual execution. Files staging in and out is performed via GridFTP and RFT (Reliable File Transfer) components.

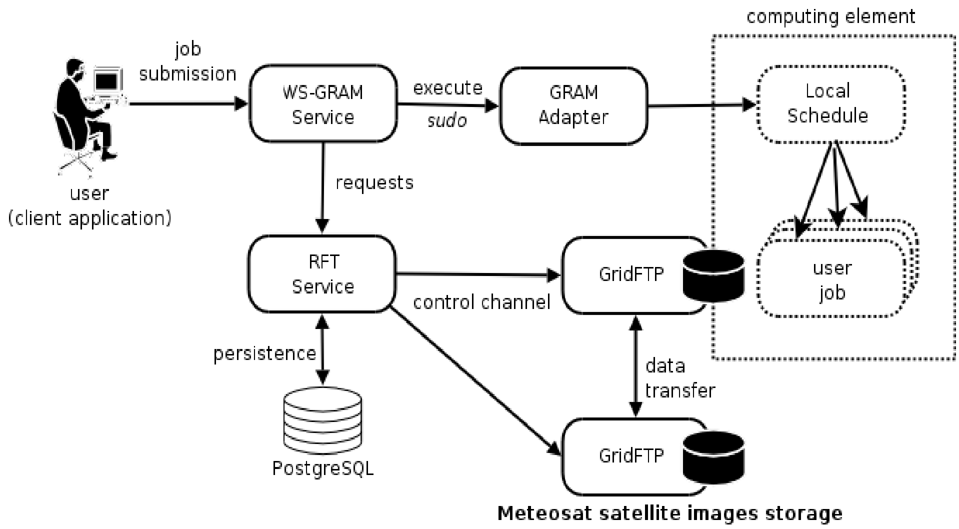


Fig.2. Remote execution with WS GRAM

Another important GT4 component heavily used by application within UASG testbed is Information management which is represented by Monitoring and Discovery System (MDS). MDS "is a suite of web services to monitor and discover resources and services on Grids. This system allows users to discover what resources are considered part of a *Virtual Organization (VO)* and to monitor those resources. MDS services provide query and subscription interfaces to arbitrarily detailed resource data and a trigger interface that can be configured to take action when pre-configured trouble conditions are met" [MDS,2005]. MDS includes a framework for data acquisition and aggregation, an WSRF[http://www.globus.org/wsrf/] compliant Index and Trigger services.

## Sample Applications

For testing and benchmarking purposes we've developed a set of applications that use standard data processing algorithms often found in space science. This application also solves some practical tasks of UASG. This set of applications consists of:

- cloud mask extraction (heterogeneous Markovian segmentation and motion detection algorithms)
- fractal-based analysis of atmosphere (approximation by the fractional Brownian motion)
- visualization (spatiotemporal interpolation, 3d-modeling)

The problem of cloud mask extraction does not put any special requirements for data archive. Depending on underlying algorithm it requires one satellite image or a short sequence of them. The particular algorithms used is heterogeneous Markovian segmentation and optical flow motion analysis (early stage of development). Fractal-based cloud modeling have a similar requirements for data archive. Horever data visualization is putting much more stronger requirements on underlying data services. Data visualization requires heterogeneous data such as basic topography, DEM data and current satellite images for processing.

Each application follows the same business logic (or workflow) with the only difference in executable file and input/output data and visualization service location. A typical sample application workflow is depicted in Figure 3:

A client daemon component of the sample application runs continually and relies on MDS4 Trigger service for monitoring newly appeared image in the archive. "New image event" triggers a job submission logic which composes an XML job descriptor [WS-GRAM, Int] and submits the job to WS GRAM service. A typical submitted job has the following structure:

- Perform file staging in (Download required input files from specified sources)
- Execute processing applications on the resource (The executable will produce result files)
- Upload/Copy/Move result fils to the visualization services location (For instance, a WWW directory of a MapServer)

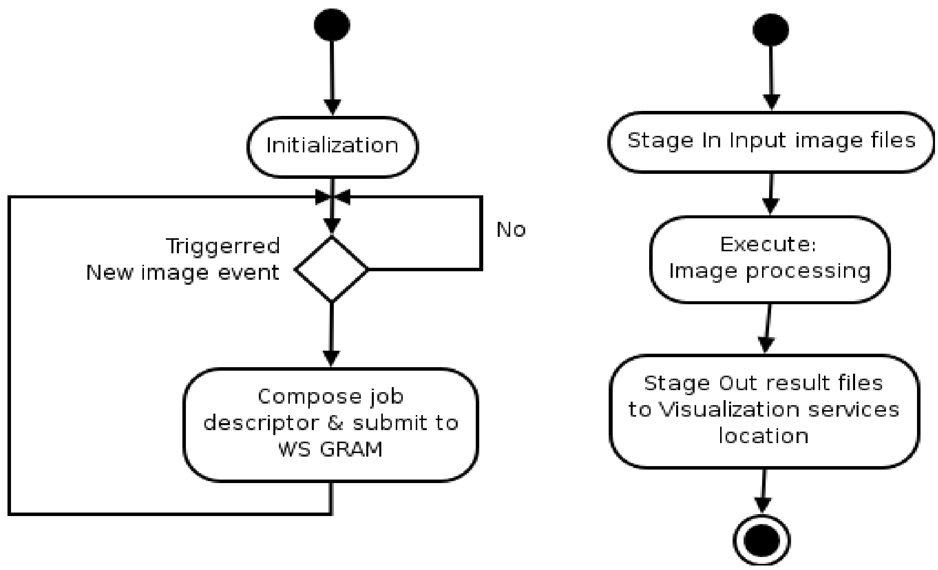


Fig.3. Sample application workflow. Left: client daemon. Right: Job structure.

All applications mentioned above require access to satellite data provided by receiving station. A specific metainformation should be attached to each file which describes image-specific attributes such as range, resolution, dimensions, date, etc. At the moment a very simple scheme is used for metadata with a very limited set of fields such as dimensions, satellite band and time. In future we plan to use standard schemas for metadata, ISO 19139 and ISO 19115 in particular.

Thus a unified interface to data files is required which will be based on Grid concepts and utilize available in the UAGS technologies to avoid formation of so called shop of frameworks or technologies.

The GT4 framework which is used for development of UASG provide all basic services required for setting up a Grid-enabled archive.

## Archive Architecture Overview

As it was mentioned in previous chapter, a unified interface to spacial data is required for application. This will avoid increase of technologies types in the system and simplify developer's efforts in accessing data in the Grid. Unfortunately, no elegible solution was found in respect to implementation of Grid interface to file archive. As a result, the architecture featured in Figure 4 was proposed to support sample application within UASG.

Let us describe the Grid-enable file archive idea illustrated with Figure 4 in more details.

The system consists of the following components: receiving station, archive host and VO index service. Receiving station is a 3rd party organization which periodically uploads satellite image files to archive host filesystem via dedicated line using well known File Transfer Protocol (FTP). This data flow is marked with number "1" in Figure 4.

The archive host consists of the following subcomponents:

- Filesystem
- Execution aggregator source
- MDS4 Index service
- GridFTP server

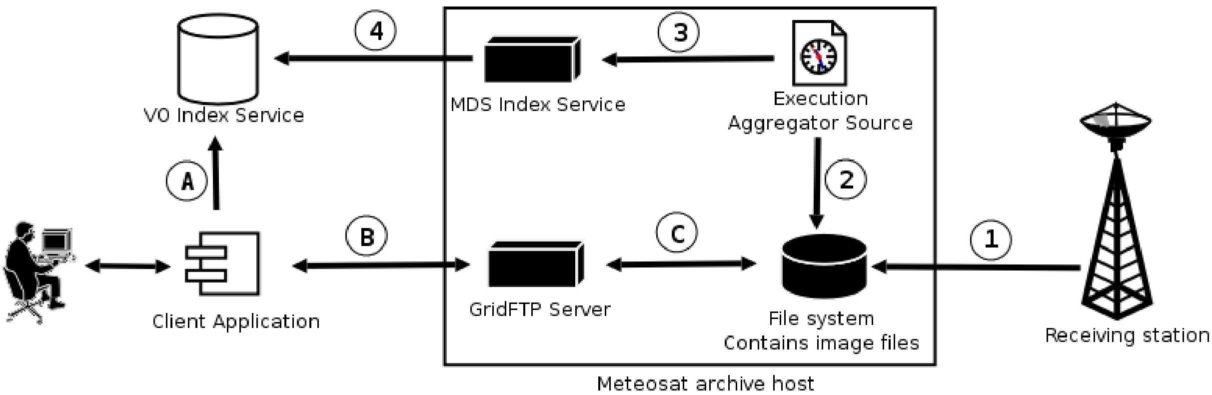


Fig. 4. Satellite data archive architecture

Execution aggregator source is a MDS4 component that performs periodical lookup to filesystem, gathers metainformation and aggregates it into local MDS4 Index service (a subcomponent of archive host as well). Thus, Index service is a temporal storage of metainformation. The persistence of data can be achieved using MDS4 Archive service which is currently under the development. If the Index service will be restarted or will reset its data the metainformation will be re-acquired and in this way database will be recovered.

Metainformation acquision and aggregation is marked with numbers "2" and "3" accordingly in Figure 4. The aggregator source script was implemented with Ruby programming language [Ruby] using external FTP command line tools such as [NCFTP-Client]. The Ruby language was chosen as a scripting language for sample applications in UASG because of its simplicity and tight integration with standard UNIX shell. This features make Ruby the perfect alternative to bash or csh scripts that tends to become very obscure with time.

The simplified script is provided in Listing 1 below:

```
#!/usr/bin/ruby
$listCommand = "ncftpls -x \"-1\""
$server = "ftp://ikd.d23.dedicated"
$rootFolder = "/pub/meteosat"
$urlPrefix = "archive.d23.local:2811/pub/meteosat/"
def getDirectoryInfo(dirName, channelName)
...(skipping lines)
end
result = Array.new
result += getDirectoryInfo "LOOP15", "VIS"
puts "<?xml version=\"1.0\" standalone=\"yes\"?>"
puts "<MeteosatInformation>"
result.each do |item|
puts "\t\<MeteosatPhoto\>"
... (skipping lines)
puts "\t</MeteosatPhoto>"
end
puts "</MeteosatInformation>"
```

Listing 1. Simplified aggregator script

All metainformation, which is used in the script listed above, is acquired inside *getDirectoryInfo* function's body. Obtained metanformation is converted into Extensible Markup Language (XML) and printed into standard system output. The output is read by MDS aggregation framework, parsed and included into Index service's Resource Properties.

The full version of the script is available at the URL: http://nyx.linuxside.net:2500/globus/show/HomePage.

The process or registration of a new execution aggregator source is well documented in [Miller, 2005].

A GridFTP server component serves as provider of data files on user's request by given direct URL.

And, finally, local MDS4 Index service registers itself (in other words, publishes) at centralized Index service of VO or so called Grid Information Service (marked as “4” in Figure 4). Thus information about satellite images becomes available to all VO users.

A typical scenario of interaction with image archive will be reviewed in the next chapter.

## Examples of Interaction

Consider the typical scenario of Grid user interaction with Grid-enabled image archive. First of all user logs into VO using his/her credentials. For instance:

```
alex@eclipse ~ $ grid-proxy-init
Your identity: /O=Space Research Institute/OU=D23/CN=Alexander Lobunets
Enter GRID pass phrase for this identity: ***********
Creating proxy .................................... Done
Your proxy is valid until: Sun Sep 25 17:23:19 2005
```

Listing 2. Initialization of working in Grid.

Then a user query VO Index service for available satellite data (in our case - meteosat image files) using common tool *wsrf-query* provided by GT4 for querying Resource Properties of WSRF service:

```
alex@eclipse ~ $ wsrf-query -s \
https://gis.d23.local:8443/wsrf/service/DefaultIndexService \
“//*[local-name()='MeteosatInformation']”
```

Listing 3. Querying VO Index service.

In the listing above a user specified a URL of VO Index service using "-s" option (in current example: https://gis.d23.local:8443/wsrf/service/DefaultIndexService ). The last argument is an expression in XPath language that allows to specialize the exact data to retrieve. The result of query execution is provide in Listing 4.

```
alex@eclipse ~ $
<?xml version="1.0" standalone="yes"?>
<MeteosatInformation>
  ... (skipping lines)
  <MeteosatPhoto>
    <Year>05</Year>
    <Month>07</Month>
    <Day>29</Day>
    <Hour>03</Hour>
    <Minute>51</Minute>
    <Channel>VIS</Channel>

<URL>archive.d23.local:2811/pub/meteosat/LOOP1505072903.515.tiff</URL>
  </MeteosatPhoto>
  ...(skipping lines)
</MeteosatInformation>
```

Listing 4. Query output.

Mentioned above actions are marked with "A" in Figure 4. As you can see from Listing 4 a user receives detailed information about image file including direct URL to image file via GridFTP server. Next, a user can decide to transfer an image he/she is interested in using common commandline tool *globus-url-copy* for example (marked as "B" in Figure 4):

```
alex@eclipse ~ $ globus-url-copy -vb \
gsiftp://archive.d23.local:2811/pub/meteosat/LOOP1505072903.515.tiff \
file:///home/alex/
```

Listing 5. Transfering selected file via archive GridFTP server.

## Conclusions

In this paper there was provided practical approach to implementation of a Grid-enabled satellite data archive. That was only a working prototype description aimed to evaluate concepts of basic services provided by GT4 application. This approach can be applied to any kind of file archive and is presumed as general approach for solving similar issues.

In a production use Grid-systems large amounts of data are distributed among different host which will require more sophisticated solutions for implementing efficient archiving. Besides processing logic will require to replicate subsets of the archives into locations which are efficient from the processing performance point of view. These issues provide reasons for continuing research in this area and finally arranging the general approach to organization and maintenance of satellite data archive with a Grid-enabled access.

## Bibliography

[Anatomy, 2001] I. Foster, C.Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations International Journal of High Performance Computing Applications, 15 (3). 200-222. 2001. Available at http://www.globus.org/research/papers/anatomy.pdf

[KSPKK, 2005] N. Kussul, A. Shelestov, N. Phuong, M. Korbakov, A. Kravchenko. Parallel Markovian Approach to the Problem of Cloud Mask Extraction, Proceedings of XI-th International Conference "Kowledge-Dialog-Solution", June 20-30, 2005, Varna Bulgaria, pp.567-569.

[WS-GRAM, 2005] WS-GRAM Documentation // http://globus.org/toolkit/docs/4.0/execution/wsgram/

[WS-GRAM, Int] WS GRAM Public Interfaces // Available at http://www.globus.org/toolkit/docs/4.0/execution/wsgram/WS_GRAM_Public_Interfaces.html

[Condor, 2005] Condor Project Home // http://cs.wisc.edu/condor/

[MDS, 2005] Information Services (MDS) : Key Concepts // http://globus.org/toolkit/docs/4.0/info/key-index.html#id2770162

[Ruby] Ruby Programming Language // http://www.ruby-lang.org

[NCFTP-Client] A command line ftp client for Unix // Available at http://www.ncftp.com/ncftp/

[Miller, 2005] Naill Miller. How to write a simple Execution Aggregator Information Provider for MDS4 // Available at http://www-unix.mcs.anl.gov/~neillm/mds/writing_a_gt4_information_provider.html

[Raicu, 2005] Ioan Raicu. A performance study of the Globus Toolkit and Grid services via DiPerF, an automated distributed performance testing framework // M.S. Thesis, The University of Chicago, May 2005

## Author's Information

**Andrey Y. Shelestov, PhD.** – senior scientific researcher, Space Research Institute NASU-NSAU; 40 Glushkov Ave 03187, Kyiv, Ukraine; e-mail: inform@ikd.kiev.ua

**Alexander G. Lobunets, MSc.** – junior scientific researcher, Space Research Institute NASU-NSAU; 40 Glushkov Ave 03187, Kyiv, Ukraine; e-mail: alexander.lobunets@gmail.com

**Michael B. Korbakov, Bcs.** – system developer, Space Research Institute NASU-NSAU; 40 Glushkov Ave 03187, Kyiv, Ukraine; e-mail: alexander.lobunets@gmail.com

# REPRESENTATION OF NEURAL NETWORKS BY DYNAMICAL SYSTEMS

Volodymyr Donchenko, Denys Serbaev

*Abstract: Representation of neural networks by dynamical systems is considered. The method of training of neural networks with the help of the theory of optimal control is offered.*

*Keywords: neural nets, dynamical systems, training.*

*ACM Classification Keywords: G.1.6 Optimization, G.1.2 Approximation, I.2 Artificial Intelligence*

## Introduction

At present time neural networks have received the wide circulation and are successfully applied to the decision of various complicated problems such as, for example, control and identification of nonlinear systems, the analysis of the financial markets, modelling of signals etc. Quality of work of neural networks depends from efficiency of the chosen algorithm of definition of weights of a network for achievement of required accuracy on training and test samples. The method of adjustment of weights of neural net on the basis of the theory of optimal control and representation of neural network by dynamical system is offered below.

## Representation of a Network by the System of Recurrent Equations

As is known [4], neural network - is set of the same elements - neurons, - divided in to the parts-layers consistently connected among themselves. Each of neurons from, actually, answers scalar function of vector argument $y=F(w^T x)$, that is superposition of the linear form with a vector of the linear form $w$ which name as vector of weights, - and scalar function $F$. Function $F$ is activation function of neuron. Vector argument x is an input of the neuron, scalar value y - an output. Inputs of the neuron belong to this or that layer on which the network is broken. These layers are ordered consistently so, that outputs of all neurons of the previous layer move on inputs of any of neurons of the following layer. An input of the first layer is the signal which is input