Toward the Conceptions of Visualization Language and Visualization Metaphor

Vladimir L. Averbukh
Inst. Math. Mech. & Urals State University
GSP 384, Inst. Math. Mech. 16. S. Kovalevskoi.
Ekaterinburg. Russia. 620219
averbukh@oso.imm.intec.ru

In this article the conceptions of Visualization Languge and Visualization Metaphor are suggestbed. The structure of a metaphor and the some conception linked with visualization languages are considered. The article includes the critical overview of metaphor used in CHI theory and in the modern practice of the interactive and visual environment design. The approaches to evaluating of visualization systems based on adequacy in visualization are suggested.

Keywords visualization language, visualization metaphor, visualization abstraction, adequacy in visualization

1 Inroduction

"Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen. It enriches the process of scientific discavery and fosters profound and unexpected insights. In many fields it is already revolutionizing the way scientists do science." This classical definition was published in [14]. Since then hundreds of visualization systems were realized. These systems can be related to three main categories such as scientific visualization, information visualization, and software visualization.

It should be note that in spite of differences both in their purpose and methods of realization in each visualization system it is possible to point out their unity in mapping of a computer model into graphical representation based on the visual model of the phenomenon in question.

We suggest an approach that will make it possible to unite in the common frameworks the researches of different visualization systems and also to provide the base for formalized techniques of visualization system evaluation. This approach is based on extraction and analysis of visualization languages. In the article the following outline is reflected:

1. Each visualization system contains as its core the language considering as an unity of the vocabulary, syntax, semantics, and (the last but not the least) pragmatics that is the user interpretations of language sentences.

2. Visualization languages are constructed on some basic idea of similarities between application domain entities and visual objects that is visualization languages are based on the visualization metaphor.

3. It is necessary to evaluate metaphors and visualization languages based on such a parameter as their ability to satisfy user's needs to represent by visual way results of user's problem decisions, that is on adequacy in visualization.

4. Developing of adequacy in visualizations as certain functional dependence on user features and parameters of application domain models has to supply the formalized design of "good" visualization systems.

2 Visualization Language

The study of different purpose visualization systems points out unified methods of visual display descriptions. This unification exists in spite of multiplicity of system goals and tasks and visualization techniques. The visualization language describes rules and method of mapping application objects and its singularities into the set of visual objects and its graphic and ungraphic attributes. A investigation and a formalized description of visualization languages are useful to improve and to speed up processes of visualization system designing. The visualization system design also have to contain the analysis of user knowledge of application entities providing basis for an in-

terpretation of various aspects of application model. In [11] the design process was described for the case of the visualization of parallel performance data (that enter into Software Visualization domain). Below we shall describe some conceptions that are necessary for more precise understanding of a design and development process in the general case of visualization systems.

In our opinion it is necessary to consider, first, computer model of an interesting entity and, secondly, mental model of this entity being in the minds of the visualization system users and/or its designers. Defining the conception of visualization it is important to consider the stage of mapping of the given computer model into the visual model based on given mental model.

The *model entity* is considered as the object under investigation, the object whose condition and behavior, characteristics, attributes and features are interesting for researchers and as the result are to be visualized.

The *view* is defined as the abstraction of graphical display including the specification of visual objects, their attributes, their interplacement, potential dynamics and interaction techniques.

The *visualization abstraction* implies connecting of *model entities* with *view* in such way as to present the content, behavior, features, and attributes of the concrete graphical display exactly identifying all visual characteristics wherein attributes of corresponding *view* are mapped.

Analyzing visualization systems it is possible to find out their community implied in the existence of visualization language. This language are revealed from the collection of graphical displays presented to the observer. Thus the language dictionary consists of *view* set used in given visualization system. (The concrete graphical displays are constructed on the base of *views*.) The language grammar (the rules of language composition) is prescribed by means the sequence of exchanged views. The significant units of the visualization language are not only 2D or 3D images but changes of their graphical and nongraphical attributes. The analysis of real visualization languages shows that true dictionary may be wide of the dictionary attributed by system designer. This is due to unaccounted visual factors affecting users.

The *language semiotic analysis* consists of extracting and describing of the language spatial syntax, semantics and pragmatics (the user interpretations of language sentences).

Analysis of visualization language semantics have to be based on the question – how it is realized choice of the main objects for visualizations that is the choice of the main visualized entities of given computing models, their features and conditions.

Similar to considered above language dictionary the true pragmatics of visualization languages may be different from pragmatics attributed by visualization system designer.

It is necessary to provide the correctness and naturalness of the visual text interpretation taking into account concrete user national and/or professional cultures. There is no visualization pragmatics common for all applications and all users. The problem of visualization pragmatics is closely involved with subjectivity of visual text perception depending on cultural, psychological and physiological factors. Studding of user perceptions of visualization language texts led to the following conclusion – a success of a visualization system (as a rule) is connected with the visualization of limited number of entities and functions.

The analysis of visual programming languages have to provide in a like manner by recognizing that the nature of visual programming languages is similar to the nature of visualization languages.

Thus, the consideration of visualization languages presumes not only language description based on user's guide but more fine analysis of system entities and ways of their displaying including user manipulations with visual objects on the screen. Visualization languages as a rule, are "reading" languages. The significant elements of their dictionary serve for the user perception and for the further interpretation of visual sentences.

3 Visualization Metaphor

Any visualization is founded on imagery similarities of model entities and visual objects representing them. That is visualization is founded on an idea of a metaphorical representation. The development of this idea is the basis of any visualization language.

3.1 Metaphor in Computer Science

The phenomenon of metaphor has been investigated in the context of literary analysis, philology and philosophy and now in the framework of such domains of Computer Sciences as human-computer interface (HCI) and visualization.

Several approaches to understanding of metaphor in Computer Sciences can be mentioned.

Prevailing now views in literature perceive the conception of metaphor as explaining of facts, which are completely new or rather unusual for the user, by means of some other facts, which are well known to the user from everyday life and which share the main features of the facts they shall explain. [21]

The main achievement of this intention is well-known. It is generally received desktop metaphor connecting office realia with programming concepts. After this manifest success several more or less successful attempts of realizing of global interface metaphor have been undertaken. It is necessary to mention such interesting examples as following:

- geographic space metaphor used specifically in software development environment. [8],
- flying and fish tank metaphors used in information systems (and also mixed "flying and fish tank" metaphor in [10]),
- theatre, cinema and comics metaphors in programming by demonstrations systems [22].

Metaphor in this case is mapping between different areas of human activity and is useful for the stimulation of associations. Such metaphors define the whole set of concepts which users apply when solving their problems. Visual symbols are used for the idea action and command description. Global metaphors promote a better understanding of interaction semantics as well as ensures a visual presentation of interactive objects and define sets of user manipulations.

Besides there are examples of local problem-oriented visualization metaphors also base on using of everyday and technical notions. [20], [21].

There also exists a different understanding of metaphor when said for example that a given visual system supports different graphical metaphors used in programming, such as finite automatons, block diagrams, dataflows diagrams, etc. Corresponding metaphor, as a rule, is supported by its own visual programming language. The design of this language is connected with the choice of correspondents between certain aspects of program and graphics. At the same time this choice determines the behavior of graphics model.

At last notice articles containing in-depth analysis of metaphor conception and interpreting it in the context that is close to modern philosophy one. In [12] metaphors in connection with computer discourse are considered. It is necessary to mention that the computer discourse essentially requires ubiquitous using of more or less appropriate metaphors for description of new conceptions arising from system development. (The name of new area "Data Mining" provides a comparatively fresh example of pure metaphorical approach.) In [13] systematic approach to metaphoric design is offered and theoretical basis for metaphor choices and associating of entities are considered. In [19] the interaction of conceptions "Metaphor" and "Illusion" is considered and their using during design of immersive virtual reality systems are analyzed.

3.2 Analysis and critiques of metaphorical approach

In spite of doubtless successes of metaphors in the human-computer interface should note fair criticism of metaphorical approach, which is contained in [18]. For example it is a important remark considering the meaning displacement that supports spatial visual metaphors by means of resemblance or analogies with situations of the real world. This displacement can be both positive, and negative, when the restrictions of real situations are transferred to metaphoric meanings. It is possible the simplification of essence phenomena understanding, loss of details and some specific notions, which have no analogues in the chosen metaphor. Often metaphor using reveal "metaphoric artifacts" that is metaphor object characteristics absent in target application are transferred on application model objects. Additional and undesirable analogies connected with everyday metaphors may appear in user minds. Note the inadmissibility of metaphor choices on verbal resemblance. Often such metaphors are connected with programmer argot and are language and/or cultural oriented. It makes the metaphors incomprehensible and bad for users who are speakers of different languages. The simplest example of such a cultural and language dependent metaphor is using of "crossed-bug" icon for the presentation of the searching error function (debugger). This argot-based icon is incomprehensible as metaphor for many Russian-speaking students and programmers. Though some of critical remarks may be to dispute, but experience of using the visual methods shows that a grain of criticism useful as warning for visual system designers. Note in this connection that the given criticism concerns a narrow understanding of metaphors as using of everyday life realia. In our opinion exactly this narrow understanding rather then using a metaphor itself is the reason of the unsuccessful decisions and the base for critiques. It is impossible to transfer understanding of a metaphor successfully used in HCI to all cases of visualization. It is necessary to develop a more

broad understanding of visualization metaphors including actual traditions of HCI metaphor uses but free from little guidance in every details of source natural domain.

3.3 Definition of a visualization metaphor

We consider a metaphor as *the main idea that determines the mapping from application domain to the visual world.* Our approach to understand a metaphor is extended in comparison with traditional one and in the significant measure is based on semiotics. This approach has to formalize the search, design, and generation of visualization *views.* In our opinion there are no "metaphorless" visualizations of computer models and program entities (in spite point of view declared in [7] or [16]). In the literature it has long been observed that any metaphor is a picture and accordingly that all graphical images of visualization are based on metaphors and so are of metaphorical nature. All cases of visualization are properly metaphors since it represents one things (model objects) as something else (visual objects) in order to interpret the results of computing. In this connection it is necessary to consider the term of *visualization metaphor* as a mapping that provides correspondence between notions and objects of modeled application domain and a system of similarities and analogies. This system of similarities and analogies generates a representational set (a set of *views*) and techniques of interaction with visual objects.

4 Structure of visualization metaphor

In connection with the semiotic analysis of visualization languages it is necessary to define notions of a sign process and a sign system. Sign process is considered as a set of relation between the sign, the sign interpreter, his/her predisposition to the certain reaction on the sign, the sign signification and context. Sign system is considered as a set of signs, where relations between detonates are mapped by some way into internal relations between signs.

The choice of a metaphor is a choice of sign system, which is used in visualization. The other function of metaphor is the determination of context to assist the correct interpretation of the given visualization language elements. By this means visualization metaphor provides understanding of mapping application domain entities and also it provides the creation of new entities based on internal logic of the metaphor. We may consider as components of visualization metaphor imagery and operations generated and directed by metaphor. Operations are considered as animations of visual images as user's manipulations with visual objects.

The analysis of visualization systems shows the presence of metaphor "focus". Focus has to supply the main influence of the perception of the language building on the visualization metaphor. Sometimes the metaphor focus is based on any differences of application and metaphor entities. It is significant that metaphor focus is the subjective conception and that there are examples of visualization metaphor without any foci.

Thus the visualization metaphor may be described as set consisting of:

- metaphor imagery;
- operations directed by metaphor both animation operations and user's manipulations (in degenerated case the observation may be considered as these operations);
- the set of similarities between model and metaphoric entities and/or elements of semantic nonconcurrence;
- metaphor focus.

Consider several interadditional approaches to the formalized description of visualization metaphor.

1. At the first of them metaphor is described as a set:

Thus visualization metaphor may be described as tuple $\{I, O, S, F\}$ where:

I is the set of images generated by metaphor;

O is set of operations ordered by metaphor; these operations may be animations of visual images as well as user manipulation with visual objects;

S is the set of metaphor similarities or its differences;

F is the metaphor focus.

2. The second approach conceives the description of a metaphor as map of $[E]$ that is a set of application domain entities into $[O]$ that is a set of visual objects of visualization language. Thus metaphor is the map J of set $[E]$ into set $[O]$; $J([E])=[O]$.

Further, define $\{[E]\}$, $\{[O]\}$ as the sets of meanings included in $[E]$, $[O]$ accordingly.

$\{[E]\}$ and $\{[O]\}$ belong to set of meanings SEN. Let M_s is the degree of closeness on SEN. If $M_s(\{[E]\})$ is close to $M_s(\{[O]\})$ then metaphor J is successful (good).

3. The third approach is to consider the visualization metaphor as an origin of visualization language grammar. That is metaphor is considered as a specification including the cores of language dictionary, syntax, semantics and pragmatics.

Consider a metaphor as an analogy (or analogy system) of a given application domain with other domain, whose main entities are well-known and assign generally accepted meanings. Using this analogy (similarity), we immediately define a possible visual dictionary, whose elements also have generally accepted meanings. The recognition of the language elements and sentences is also predetermined by similarities of used metaphor. In a similar manner we may consider a layout of language elements and techniques of descriptions of their relations. They also basically are predestined by customary allocation of source domain objects used as analogies. Thereby it is possible to present visualization language as a developing of cores of dictionary, syntax, semantics, and pragmatics including in the metaphor specification.

5 Quality evaluation of visualization metaphors

Traditionally the goal of visualization is considered as "finding a graphical representation for program (or model) behavior which provides a good mapping to the way the programmers (or researchers) themselves tend to formulate solutions" [9]. Thus the quality of visualization is connected with "goodness of mapping" or "goodness" of visualization metaphor.

However, in spite of interesting articles published in recent years and denoted to methods of visualization quality evaluation as a whole a problem of formalized evaluation is not solved. Our approaches to formalization are a subject of the further consideration.

It is necessary to consider notions which are important for the construction of approaches to the systematic evaluation and analysis of visualization systems such as visual expressiveness, visual informativeness, adequacy in visualization.

1. Let's define informally a notion of visual informativeness. The term *"visual informativeness"* are considered as a subjective characteristics of quantity of the useful information received by a user (receiver of the information) from the visual text.

Such approach to the determination of informativeness reflects a correlation of subjective an objective, syntax and semantic characteristics of visual messages. The traditional approach to the evaluation of information assume that a semantics of message does not depend on user characteristics whereas the notion of informativeness allows as far as possible to define subjective semantics and to evaluate pragmatical properties of visualization languages.

2. The term *"visual expressiveness* (expressiveness of visualization language)" is considered as a capacity for this language to express the maximum quantity of senses and shades of senses using the minimal number of language elements. There is the link between conceptions of "visual expressiveness" and "informativeness of visual message". The increase of the informativeness of some visual "word" causes to the increase of the visual expressiveness of all visual text. The expressiveness of any text is determined by presence of images describing senses of events or situations contrary to retelling of their sequences. Quantitative parameters of both informativeness and expressiveness may be evaluated by experiments. [1].

But from the user's point of view the main characteristic of visualization language is the ability of the language to satisfy needs for application problems solving. That is it is necessary to evaluate how quickly and exactly users may interpret visual texts. If visualization language is laconic, informative and well interpreted (is translated into user's mental language) then this language accords adequately with user's needs.

3. The term *"adequacy in visualization"* defines such properties of visualization languages that allow given user (or given user category) to solve concrete application problems. The negative factors are the existence of additional meaning in visual texts or the lack of convergence between user and designer text interpretations. The values of adequacy in visualization relate to visual informativeness and visual expressiveness. To describe the adequacy in visualization it is necessary to analyze the application domain and to build the user model. [2].

In connection with the analysis of application domain it is necessary for the beginning to assess possibilities of mapping application domain objects with one or another visualization metaphor into a visual space generated by this metaphor. It is necessary to describe the main entities of application domain taking into account analysis of visual imagery, and to choice the main application notions with the possible clarification of their advantageous resemblance with images generated by a given metaphor. Also it is necessary for visualization to choice key entities of given application domain. These entities need not be the

main ones in the given application domain but they have to allow solving visualization problems. The application domain imagery and imagery generated by given metaphor should be analyzed in connection with visualization (for example based on resemblance of terminology using in both source and target domains). Also it should be estimated the possibility of using of abstract symbols and icons, as well as using of images connecting with one or another computing algorithms.

A quality evaluation of visualization and visualization metaphors have no to be based only on the simple tests and experimental situations (as for example in [7] or [6]). As discussed in [17] only a long-term commercial or educational usage of a visualization system is a reliable indicator of its quality.

In our opinion the key feature of visualization systems is adequacy in visualizations. Practically, exactly adequacy in visualizations instead visual expressiveness or other quality features of visualization metaphors (as we considered earlier) we had tried to estimate by indirect techniques studying model tasks (described in [1]) user's perception speed, perception uniqueness, fatigue level, availability (or lack) of an aesthetic and emotional satisfaction and one or another user's preference. We try to prove it experimentally and also to find and to research experimental techniques of evaluation of adequacy in visualization.

The ParaVision project [3] is an attempt to develop a system for evaluation of adequacy in visualization. ParaVision is linked with problems of visualization of behavior of parallel systems for their monitoring and debugging. In this domain the visualization is the most useful and effective meaning. Setting up a problem have assumed that common model task should be presented to a sizable and uniform group of testees. The visualization *views* using for task solving are based on different metaphors. Further an experementer has to analize statistical data and to compare the average time spent on decision of model task and on this base we may determine what metaphors is the best (the most adequate) for visualization. The model problem is a popular problem of parallel programming – The Dining Philosophers problem which allows to illustrate situations arising during parallel computing processes.

There are three visual metaphors offered to users: natural, symbolic, and abstract. In the natural metaphor, philosophers, forks, etc., are represented by natural images. In the symbolic metaphor, philosophers and their operations are represented by circles. In the abstract metaphor, the parallel process states (idle, active and waiting-for-lock) are represented by Gantt charts.

ParaVision system allows to consider parallel processes by examples of a model task, to interface with program changing its parameters, to consider the common situation by means different visualization metaphors, to test an user offering him to find the solution of parallel problems.

Next prototypal systems for evaluating of adequacy in visualization were oriented to possibilities of user's own choice of entities for visualization languages under fixing model problems. The main goal of this environment was to provide the construction of visualization language by mean choosing a kit of visual elements, rules and techniques of a linkage of sentences and senses. As a model application domain the sorting algorithm animation was chosen.

6 Discussion and Conclusion

The approaches to a definition of a metaphor conception offered in this article allow to generate new fruitful ideas for visualization and CHI without literal following to (often casual, for example [7], [21], [15]) similarities of entities that occurs with traditional understanding of metaphors. Yet our approach providing a choice of adequate in visualization metaphors and languages unites as traditional understanding as other cases of using of analogies and similarities between entities during the process of visualization.

Our methods also may use in the cases of scientific and information visualization. Thus, this approach was realized in the set of specialized visualization systems using for optimal control and differential games applications. New views and techniques of user's manipulation with images for supporting effective interpretations of modeling results were developed basing on analysis of characteristics of adequacy in visualization [4], [5].

Note that specialized visualization systems supporting new views and interaction techniques are absolutely necessary on stages of a new computer model development because studying of model objects requires new techniques of visual representations. These problems are traced also in the design of visualization and modeling environments for biomedical researches.

Some visual systems (for example Microsoft Windows) are based on desktop metaphor adopted from office automation domain. That is why users-professionals in medicine (and also professionals in

other domains) must use distant metaphors and visual languages. However it becomes unacceptable, when disturbs the decision of important problem to analyze source large data describing patients. It is necessary to design complex views including for example as elements of statistical graphics as natural images of human organs.

The traditional point of view on the problem of visualization language considers 2D or 3D graphical primitives, colors, textures, etc. as elementary language units. In modern visualization systems we have to consider as languge units some kinds of multimedia interactive movie consisted of 3D frames, soundtracks, possibillities of user manipulations, etc. In this connection it is possible to consider *generalised views* included as a set of animated images as potential user manipulations with visual objects. Naturally that in a degenerate case the view may consist of static images. Also it is possible to display images without any interactive actions.

Designers of specialized visualization systems have to develop the techniques of constructing such views which answer to concrete user's notions about nature of application domain entities, to user's goals, and to his/her methods of problem solving.

On our opinion it is necessary to develop individualized visualization systems (but not only powerful widespread visualization systems). These individualized systems have to orient to concrete users and their concrete classes of problems. Analysis of user's preferences in his/her choice of visualization language facilities during problem solving is the base for evaluations of adequacy in visualizations. In this connection full-blooded techniques of evaluations of visualization system quality based on results of psychological researches are necessary. New approaches to the study of adequacy in visualizations may help to solve an important problem – a problem of generating the visualization metaphors and corresponding visualization languages which have to provide the decisions of problems in the given application domain. The other important problem is the search of mathematical foundation for adequately describing of visualization metaphors formalisms.

Thus it is possible to formulate unsolved tasks in connection with our plans in the visualization domain:

- development of application domain models
- development of user models;
- development of the adequate mathematical means.

ACKNOWLEDGEMENT

The authors are grateful to Professor Erkki Sutinen for most helpful discussion.

References

[1] Averbukh V.L., Konovalov A.V., Tarskikh I.V., Vorzopov V.V. Analyses of Visual Metaphors and Languages. Toward Prototyping of Software Visualizing Systems // "East–West". International Conference on Human-Computer Interaction EWHCI'94. St.-Petersburg, Russia, 2–6 August, 1994". Proceedings. ICSTI. Moscow. 1994. V.I, pp. 244–254.

[2] Averbukh V.L. Toward Formal Definition of Conception "Adequacy in Visualization" // Proc. 1997 IEEE Symposium on Visual Languages, Sept. 23-26, 1997. Isle of Capri, Italy. - S.l.: IEEE Comput. Soc., 1997. p. 46-47.

[3] Averbukh V.L., Konovalov A.V., Vorzopov V.V. An Approach to Evaluations of Software Visualization // Human Factors in Computing Systems. CHI 97 Extended Abstracts. Atlanta, Georgia USA, 22-27 March 1997. ACM, 1997, p. 42.

[4] Averbukh V.L., Kumkov S.S., Shilov E.A., Yurtaev D.A., Zenkov A.I. Specialized Scientific Visualization Systems for Optimal Control Application // Nonsmooth and Discontinuous Problems of Control & Optimization: Proc. IFAC Workshop Chelyabinsk, Russia, 17–20 June, 1998. New York etc.: Pergamon, 1999. P.71–76.

[5] Averbukh V.L., Kumkov S.S. Pykhteev O.A., Yurtaev D.A. Visualization of Level Sets and Singular Surfaces in Differential Games // Proceeding of ALGORIMY 2000, 15-th Conference on Scientific Computing, Vysoke Tatry – Podbanske, Slovakia, September 10–15, 2000. Slovak University of Technology, Bratislava. 2000. pp.196–206.

[6] Bederson B.B., Boltman A. Does Animation Help Users Build Mental Maps of Spatial Information? // HCIL Technical Report No. 98-11 (September 1998) http://www.cs.umd.edu/hcil

[7] Blackwell A. Green T.R.G. Does Metaphor Increase Visual Language Usability? // 1999 IEEE Symposium on Visual Languages VL'99,

Tokyo, Japan, September 1999/ http://www.-cl.cam.ac.uk/ afb21/publications/VL99.pdf

[8] Christensen H. A Software Development Environment based on a Geographic Space Metaphor www.daimi.aau.dk/ hbc/Ragnarok/gsm.ps.gz

[9] Eisenstadt M., Domingue J., Rajan T., Motta E. Visual Knoledge Engineering // IEEE Transactions on Software Engineering. Vol. 16 No 10 (October 1990). pp. 1164–1177.

[10] Fleet D., Ware C. An Environment that Integrates Flying and Fish Tank Metaphors // Human Factors in Computing Systems. CHI 97 Extended Abstracts. Atlanta, Georgia USA, 22–27 March 1997. ACM, 1997, pp. 8–9.

[11] *Heath M., Malony A., Rover D.* The Visual Display of Parallel Performance Data // IEEE Computer, V.28, N 11, (November 1995), pp. 21–29.

[12] Johnson G.J. Of Metaphor and the Difficulty of Computer Discourse // Communication of the ACM. Vol. 37, No 12 (December 1994). Pp. 97–102.

[13] Madsen K.H. A Guide to Metaphorical Design // Communication of the ACM. Vol. 37, No 12 (December 1994). Pp. 57–62.

[14] McCormick, B.H., T.A. DeFanti, M.D. Brown (ed), Visualization in Scientific Computing // ACM Computer Graphics Vol. 21, No. 6, (November 1987).

[15] Musil S. Monitoring Parallel Programs with INHOUSE//http://www.ani.univie.ac.at/-ani/research/Monit.html

[16] Nardi B.A., Zarmer C.L. Beyond Models and Metaphors: Visual Formalism in User Interface Design // Journal of Visual Languages and Computing (1993) No 4, pp.5–33.

[17] Price B.A., Baecker R.M., Small I.S. A Principled Taxonomy of Sofware Visualization // Journal of Visual Languages and Computing V. 4 (1993) N 3, pp. 211–266.

[18] Repenning A. Agensheets: A Tool for Building Domain-Oriented, Dynamics, Visual Environment // Ph.D. Dissertation. University of Colorado at Boulder. Department of Computer Science. CU-CS-693-93. December 1993.

[19] Riva G. From Technology to Communication: Psycho-social Issues in Developing Virtual Environments // Journal of Visual Languages and Computing (1999) No 10, pp.87–97.

[20] Samofalov V.V. Scharf S.V. The Visual Process: Design, Development and Application in Parallel Programming // Algoritmy i programmnye sredstva parallel'nyh vychislenii. (Ed. A. Kukoushkin and S. Scharf). Yekaterinbourg. IMM UrO Of RAN. 1995. pp. 170-181. (In Russian)

[21] Tscheligi M., Musil S. An Information Manipulation Environment for Monitoring Parallel Programs // AVI 1994: Bari, Italy. AVI'94, Proceedings of the Workshop on Advanced Visual Interfaces, June 1-4, 1994, Bari, Italy. ACM, 1994. P. 246–248

[22] Watch What I Do. Programming by Demonstration. (Ed.- Allen Cypher) MIT Press. Cambridge, (Mass.), 1993.