Chapter 6

Semiotic Analysis of Computer Visualization

Vladimir L. Averbukh

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/67729

Abstract

The purpose of this chapter is to discuss the semiotic approach to form theory of computer visualization. Such theory should be the foundation of design, development, and evaluations of visualization systems. The "direct" semiotic analysis of visualization is defined and the scheme of the analysis is considered. This analysis reveals "who is who" in the process of the visualization semiosis and helps in design and development of the real visualization systems. The analysis allows to describe the problems arising at developments of specialized systems in terms of the semiotics and showing how this analysis can serve as a tool for the visualization systems design. It is important to analyze the sign nature of the human-computer interface and the visualization. Such conceptions as computer metaphor, metaphor action, and metaphor formula are defined. The properties of metaphors are analyzed with a view to possible usage of metaphors for specific applications. The properties are considered by the example of the hierarchical sequence of the natural Room-Building-City (Landscape) metaphors. Also the properties of the molecule metaphor are considered in the context of software visualization systems. In conclusion, some approaches to the theory of computer visualization are outlined.

Keywords: computer visualization, semiotic analysis, visualization metaphor, entities of metaphors, analysis of metaphors

1. Introduction

This chapter connects our previous research studies on semiotics approaches to computer visualization theory and visualization metaphors. It is an extended and revised version of Ref. [1].

In 1987, the special issue of ACM SIGGRAPH Computer Graphics Journal was published. The issue was devoted to the definition and description of computer visualization. The description computer visualization as the independent discipline summed up the great practice of Computer

Graphics since the beginning of 1960s. In this issue, the main conceptions of the new discipline were defined. The visualization is considered as 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. The goal of visualization is to support the analysis and interpretation stages in framework of the computer modeling cycle. One can consider three main directions in research studies and developments for the computer visualization domain. That is — computer graphics (hardware and software including mathematical and algorithm components), software engineering, and human factors. Our interests lie in the human factor subdomain. The process of human dealing with visualization consists of three stages "*Perception→Cognition→Inter pretation.*" In the frameworks of semiotics, processes of interpretation are considered.

Computer visualization contains three main subdomains: scientific visualization, information visualization, and software visualization.

It is shown that the human-computer interaction and visualization have a semiotic nature. The conceptions of a visualization language and a figurative (visual) text described on this language are considered. The computer metaphor is considered as a basis of the visualization language. The semiotics analysis of computer metaphors allows to evaluate known metaphors and to search new ones for specialized visual systems. Thus, the semiotics analysis can be an important tool for the visualization systems design and development.

Semiotics, dealing with sign systems and with practice of their functioning, may be considered as tools for descriptions of theories of HCI and computer visualization just as mathematics is tool for description of physics theories.

The obvious semiotic nature of the human-computer interface and visualization allows to reveal the sign systems that determine interactions, visualization, and communications. Human-computer interaction in this connection may be described precisely as sign process. Visualization may also be described as sign process similar to the human-computer interaction. Processes of human computer interaction and visualization contain user interpretation of visual and dialog objects as their essential part. In turn, the process of sign interpretation is researched in frameworks of semiotics. That is why one may consider semiotics as the base of theories of HCI and computer visualization. If human-computer interface and visualization have the sign and language nature, then each interface and visualization system contains the language as its core. The language in this case is understood as the systematical description of entities under consideration, methods of their representation, modes of changes of visual display, as well as, techniques of manipulations, and interaction with them. The language (or rather a base sign system) is built upon some basic idea of similarities between application domain entities with visual and dialog objects, i.e., upon a computer metaphor (that is interface metaphor and visualization metaphor).

Semiotic analysis is an important tool for the visualization system design and development. Below we consider the "direct" semiotics analysis of the visualization that reveals "who is who" in the process of the visualization semiosis. It allows to describe the problems arising at developments of specialized systems in the terms of the semiotics and showing how this analysis can serve as a tool for the visualization systems design. Further, metaphor properties are considered to analyze the possibility of the metaphor use for specific applications.

2. Related works

The development of the "semiotics" approach to the theory of computer visualization and human-computer interaction started in the 80th years of the twentieth century. The statements of the classical semiotics were used to describe visual sign processes in connection with a computer graphics and visualization [2–6]. Using the semiotic engineering of human-computer interaction is described in this chapter [7, 8]. The design principles for information visualization based on a combination of algebraic abstract data type theory, semiotics, and social theory were suggested in reference [9]. In many articles, the semiotics of graphics and visualization is considered from the perspectives of dating back to 1980s [10]. This approach involves the study of individual sets of signs and pictographs that are often associated with cartography. However, modern visualization systems depict huge volumes of data possibly without a natural or familiar imagery. These set of displays may be considered as a visual text, which corresponds to visualization languages. The semiotics research studies of visualization languages are the basis of our approach described in Refs. [1, 11–13]. The description of visualization language is important to evaluate the already-existing systems so as to analyze decisions in the phase of design and development.

The concepts of a visualization language and a visualization text depicted on this language are considered. The concept of pictorial (graphical) text was used to describe the petroglyphs and ancient illustrative pictures depicted some narratives. Interpretation of such texts is possible only if the "readers" of the text have an external information [14]. If the graphical texts are considered, then one may consider corresponding graphical languages. They are rich and complex languages, based on natural imagery. It is static languages of fine arts, communications, illustrations and advertising, and dynamic languages of cinema and animation. Similarly, we can define graphical texts associated with computer visualization. The examples of those visualization texts are:

- isolated displays (static pictures);
- dynamic logically related shot changes with the inclusion of interaction, which may define the logic of the change of the conclusions;
- animations also with the inclusion of interaction.

In turn, we can consider the concept of languages of computer visualization. In this case, the language is understood as the systematical description of entities under consideration, methods of their representation, modes of changes of visual display, as well as techniques of manipulations and interactions with them. The language (or rather a base sign system) is built upon some basic idea of similarities between the application domain entities with the visual and dialog objects, i.e., upon a visualization metaphor.

We consider the notion of visualization language from the perspective of semiotics as unity of lexicon, syntax, semantics, and pragmatics [15, 16]. Let us use to describe the notion of visualization lexicon for the ideas from [17] where some formalized models of visualization are proposed for the case of parallel performance data. Among others, such entities as *performance view* and *performance display* are considered. We use the synthesis of these notions to describe visualization languages.

Generalized view of visualization system is defined as a visualization abstraction containing specifications of visual objects, their attributes, their relationships, possible dynamics, and methods of interaction. Thus, view design provides valid picture changes and animations, and interactions with visual objects. View may be understood as a technique of data depiction, a kind of a visualization procedure. A visualization system realizes linking view "arguments" with real data and supports an output to graphics. Resulting pictures (*displays of a visualization system*) are represented visual abstractions. A set of displays considers as a visual text corresponding to a visual language. The same views may be used in a variety of visualization systems and therefore constitute elements of lexicons of different visualization language. The set of views of the visualization system defines its visualization lexicon.

Syntax of a visualization language may be considered as a set of rules describing: (a) relationships of visual objects; (b) possible dynamics of visual objects; and (c) techniques and results of interaction with visual objects of the view. Rules of display changing may also be part of the syntax of a visualization language.

Semantics of a visualization language is set by goals and tasks of visualization. Most importantly that semantics is specified by goals and tasks of computer modeling which data are under analysis and interpretation during the visualization system.

Pragmatics of a visualization language is also related with goals and tasks of visualization and modeling. Pragmatics is determined by meaning, which can draw users of visualization systems. Visualization metaphor is considered as the basic idea of likening between interactive visual objects and model objects of the application domain. Its role is to promote the best understanding of semantics of interaction and visualization and also to determine the visual representation of dialog objects and a set of user manipulations with them. Visualization metaphors form the basis of views of specialized visualization systems.

The notion of computer metaphors (interface and/or visualization metaphor) is rather popular in the scientific literature after research studies of Kuhn [18, 19]. Studying of the literature on problems of a computer metaphor allows drawing some conclusions. One of them—there is a certain consensus on the computer metaphor theory. First of all this consensus consists of the recognition in cognitive approach to the metaphor theory as the base of the theory of interface metaphor. This approach is linked with names Lakoff and his colleagues [20, 21]. The cognitive approach to a metaphor considers a metaphor as the basic mental operation, as a way of cognition, structuring, and explanation of the world. Second, the Peircean semiotics is applied to user-interface metaphor [22]. Our approaches to problems of computer metaphors are described in Refs. [1, 23–25]. Also, some new ideas will be considered below.

3. Semiotical analysis

The sign process (or semiosis) is considered on the five-term relation between a sign, its meaning, its interpretant, a context where the sign meets, and, at last, a sign interpreter. The sign causes the interpreter to certain reaction or predisposition to it (interpretant) on a certain kind of object under certain conditions (in some context).

The human-computer interaction and visualization, necessarily, have a semiotic nature. The sign nature of visualization allows to reveal sign systems, determining interactions, visualization, and communications. There are relationships between the visual representation of an object, that is, relationships between a signified (a denotatum) and a visual sign. A user or an observer (an interpreter) in determined context recognizes the idea caused by visualization that is the interpreting idea (an interpretant). They are all relations described semiosis (the process of interpreting signs or the sign process).

A set of classical semiosis "roles" in human-computer interaction should be broadened. There is another process actor — the author of the message. The sense intended by the "author" may differ from the interpretant that is the ideas understood by the interpreter.

We consider the "direct" semiotics analysis of visualization that reveals "who is who" in the process of the visualization semiosis. It allows to describe problems arising at developments of specialized systems in terms of the semiotics and showing how this analysis can serve as a tool for the visualization systems design.

First of all, it is necessary to pay attention to the pair "sign-denotatum." Revealing of denotatum and a corresponding choice of a sign is the important problem of the semiotics analysis. Note, that in any concrete case of visualization there are "nonsign" aspects. Not everything is reduced to sign forming.

There are some simple examples (see **Figures 1** and **2**). Suppose we need to represent the progress of a simple process. One may use the conventional technique to represent—to draw a usual 2D graph. Here the process is the denotatum, and the whole of graph is the sign. If further the task to represent the change of the progress of a process, then change the direction of the graph simply and obviously indicates the change of the progress of a process. In this case, the denotatum is the change of the progress of a process and the sign is the change of the graph direction (but not the whole graph as in the previous case). For more complex cases, one may use the more complex (and more interesting) technique of visualization, for example, to animate the process basing on its natural imagery. But, in this (animation) case, one has to construct the more sophisticated and complex sign to represent the same denotatum (the change of the progress of a process).

Let us consider the next example that is the simplification of real specialized system of scientific visualization for the model of pollution of the environment. In the beginning of the system development, the task on visualization provided the real imagery of pollution and animation—the smoke from factory chimneys is diffused in the town air and the dirt from the factory tubes is diffused in the town pond. This animation may be interesting for regional authorities, factory managers, and environment defenders. That is originally the process of

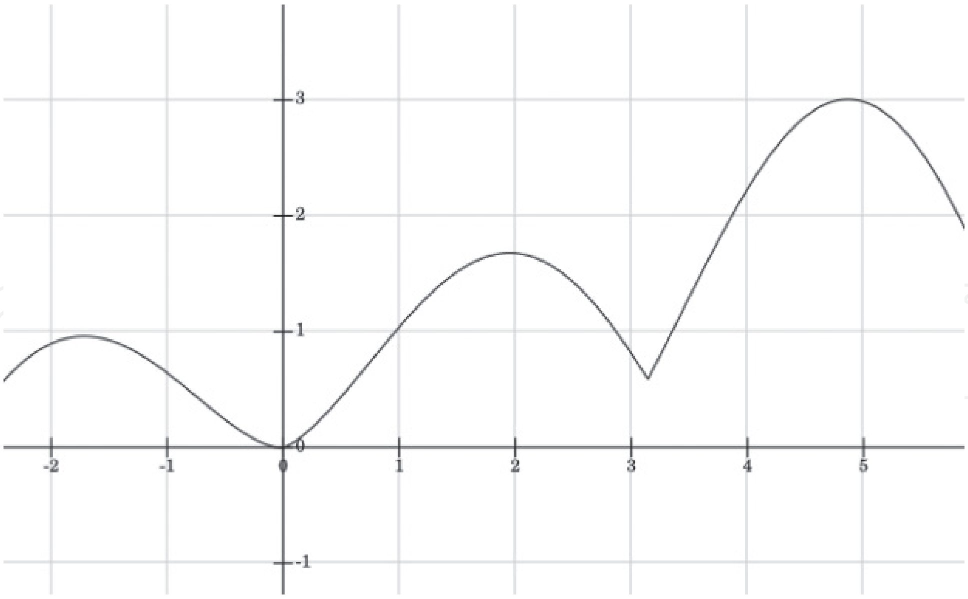


Figure 1. Sample of the plot of a function.

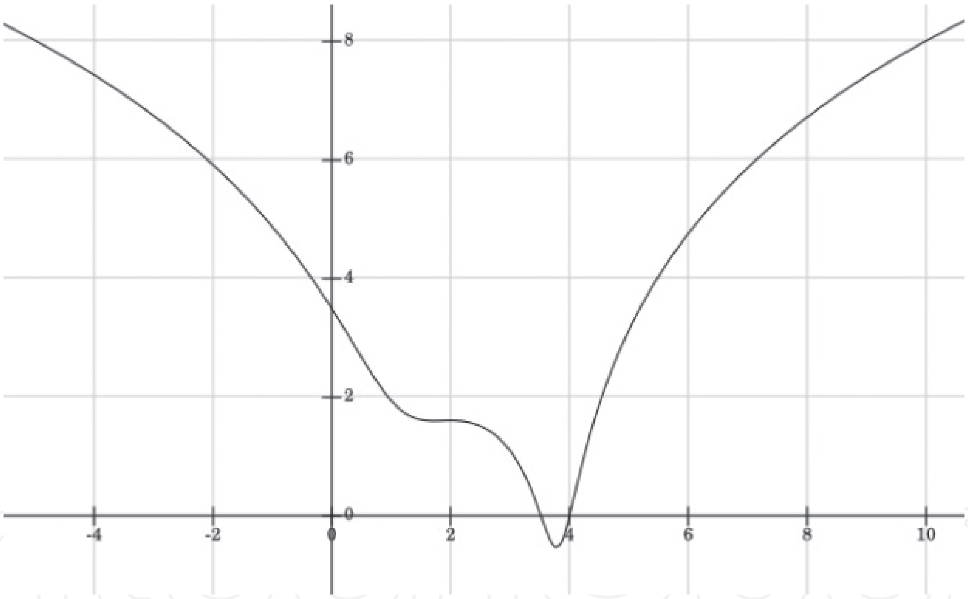
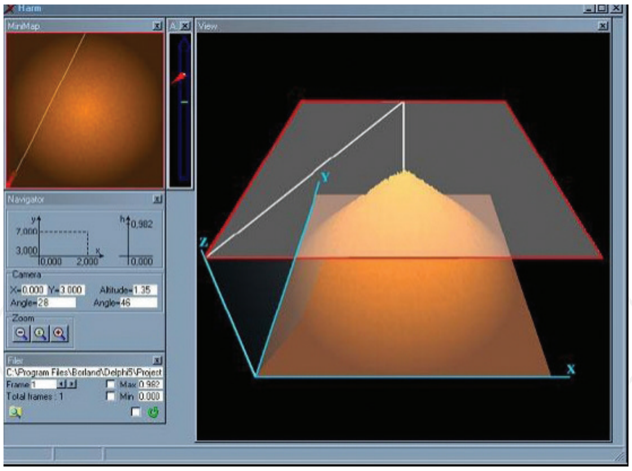


Figure 2. Changing of the direction of a process.

pollution that was considered as a denotatum. The realistic animation has to be the basis of sign representations. Note that in this case the realistic animation is not too suitable to depict the process uniquely. However, analysis revealed that the main problem of this mathematical and computer modeling resided in the reconstruction of values of emission rates basing on available information. Thus, the denotatum and the subject of visualization were not in the least process of pollution of the environment but some properties of the same mathematical model. The use of an abstract imagery to visualize the model is not surprisingly. Just we used the 3D surface to depict the model. In particular, isolines showing equal pollution loads are the sign for the process of pollution (see **Figure 3**).



X=0.000 Y=3.000 Altitude=1.35
Angle=28 Angle=46

C:\Program Files\Borland\Delphi5\Project
Frame 1
Total frames : 1
□ Max 0.982
□ Min 0.000
□

Figure 3. Visualization for modeling of environment pollution.

Another real example is the simulation of excitative process in cardiac chambers. At once, note that, in this case, the excitative process in cardiac chambers is the denotatum. Experts suggested the scientific metaphor to represent pathways of myocardium as the set of interconnected cells. These cells may send signals to each other.

The model depicts myocardium and simulates the excitative process in cardiac chambers by means of simple animations. Basing on this animation we succeeded to visualize the simulations such pathologies as tachycardia and extrasystoles. The simulation of the cardio disorders was realized by means of the system parameterization. In particular, time intervals corresponding to different states of the cardio cells were varied. Such parameters as preparedness to receive/transmit; process of receive/transmit; unpreparedness to receive/transmit were under user's control. The presence of pathology was depicted by types of hesitation. Really dangerous pathologies are chaotic animations. The stable (even not norm) animation is a sign out of the deadly condition. Three-dimensional (3D) model of the heart generated at the first stages of development was rejected because, first, it was inadequately for chosen scientific metaphor and, second, visual perception of 3D animation was difficult for users. Flat representation in this case turned out to be more accurate and winning in terms of user experience. Despite a number of restrictions, the model completely satisfied the expert requirements. In this case, the sign indicating the presence of simulated pathology is the type of oscillation. The heart itself, which is not a matter of designation, does not need in visualization in this case (see **Figure 4**).

Consider the following examples related to the algorithm visualization. Algorithm visualization and animation systems are considered as education means but they may be used as instruments for algorithm evaluation and debugging. Let's ask a question—what is denotatum in the case of algorithm animation. It will be recalled that in the frameworks of theory of visualization the conception of “algorithmic operation” is considered. Algorithmic operations are such operations of the algorithm that are important to understand the program's semantics. For example “compare” and “exchange” in a sorting algorithm [26]. That is, in the case of algorithm visualization, its base operations may be considered as denotatum rather than the algorithm itself. (As we know algorithm is rather complicated conception.)

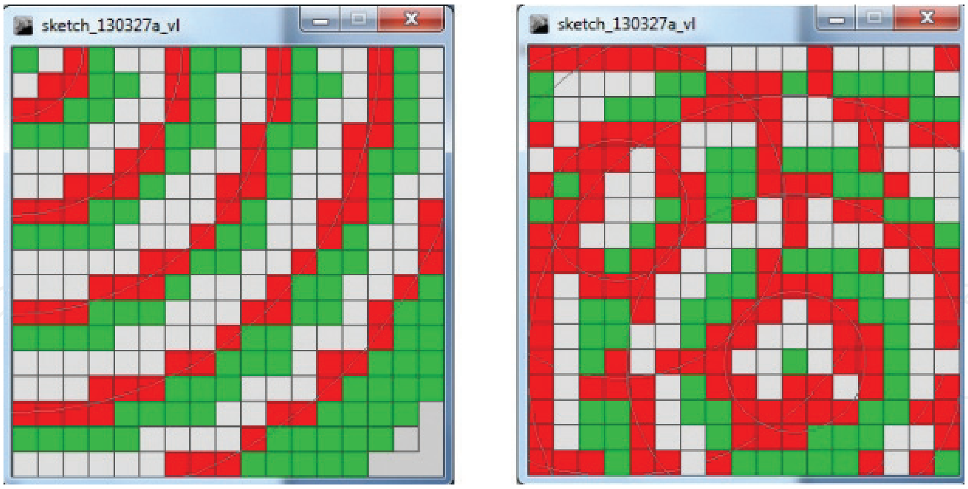


Figure 4. Normal (left) and pathological (right) variants of excitative processes in cardiac chambers.

Starting in the 1980s of the twentieth century, a number of algorithm animation systems were developed. In these systems, the designation was conducted by creating dynamic images that demonstrates the behavior of the algorithm. Here visual dynamic images are considered as signs. In the "classical" systems of algorithm animation only "exchange" operation was depicted when sorting algorithms were realized. "Compare" operation seemed as self-evident for users-observers of animation. In the 1990s, we have researched some problems of representation of both operations in sorting algorithms. On our opinion, the value of variable is preferable to depict by the size of bars. Whereas the color for that end may be used only in certain cases. Some approaches for visualization of "compare" operation were suggested. For example, a harpoon (or an arrow) was used for this purpose. A "harpoon" is moving up from the end of current (lower) object to compare next objects. If the "harpoon" collides with other object then it becomes lower, and the former current object goes up one step (see **Figure 5**).

There are also a number of other successful examples of algorithm animation systems, but majority of these animations deals with sorting and graph algorithms. Sometimes systems depict and animate the process of program execution rather than algorithms.

Considered examples of revealing of a denotatum at semiotics phase of the visualization design show that answers two questions that are important:

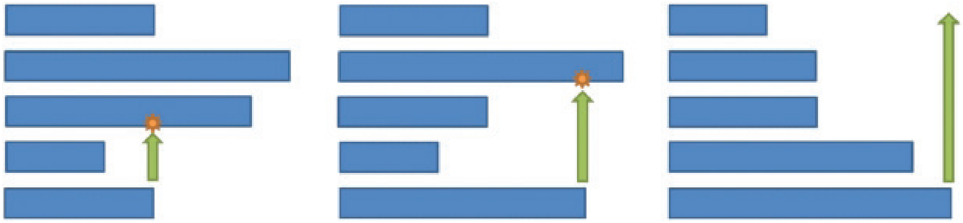


Figure 5. Animations of sorting algorithm using "harpoon" for depicting "compare" operation.

- “what are the objectives of visualization?”
- “what are the subjects of visualization?”

The answer to the second question as one may see needs the special analysis; it is not trivial but sometimes it is unknowns.

Searching methods of the denotatum representation and designation is connected with the conception of a computer metaphor.

4. Computer metaphors

Informally, the visualization metaphor is understood as mapping from an application domain to the visual world. The visualization metaphors have to help in understanding the complex and abstract concepts and in clarifying the relationships between objects. Metaphors are used to define the activities of computer systems users and to depict forms of her/his vision and operations on program objects. Theoretically, any visualization is metaphoric. Whereas in the literature, the traditional methods of data visualization and new (metaphorical) ideas are often opposed [24, 27]. The semiotics analysis of computer metaphors allows to evaluate known metaphors and to search new ones for specialized visual systems. Thus, the semiotics analysis is an important tool for the visualization systems design and development.

The metaphor essences consist in interpretation and experience, the phenomena of one sort in terms of the phenomena of other sort. Metaphorization is based on interaction structures of source and target domains. During process of metaphorization, some objects of target domain are structured on an example of objects of target domain and there is a metaphorical mapping (projection) of one domain onto another. That is the metaphor can be understood as a map from source domain onto target domain, and this map is strongly structured.

Cite an example of a classical metaphor LIFE IS A JOURNEY, where LIFE is target domain, and JOURNEY is source domain. Some structures of JOURNEY (beginning, ascent, descent, end, etc.) are considered in the given metaphor as a basis for the description of life structure. Image-schemas are image-like reasoning patterns, consisting of a small number of parts and relations, made meaningful by sensorimotor experience. There are a CONTAINER schema (things that have an inside, an outside, and a boundary), a PART-WHOLE schema (something can be seen as a whole or as its constituent parts), a LINK schema (two or more things have a link between them), a SOURCE-PATH-GOAL schema (or sometimes, just a PATH, which goes from a source along a path to a destination). There are an UP-DOWN schema, a BACK-FRONT schema, and so on. Schemas are gestalts—structured wholes—that structure our direct experiences. Image-schemas may in fact be the kind of structure, which is preserved by computer metaphors [20, 21].

One can define a computer metaphor (interface and/or visualization metaphor) as an operator from concepts and objects of the application domain under modeling to a system of similarities and analogies generating a set of views and a set of techniques for interaction with and manipulation by visual objects. Computer metaphor is considered as the basic idea of likening between interactive visual objects and model objects of the application domain. Its role is to promote the best understanding of semantics of interaction and visualization and also to determine the visual representation of dialog objects and a set of user manipulations with them. Visualization metaphors form the basis of views of specialized visualization systems whose design is the important part of whole design the "human factor" aspects of these systems.

A set of requirements imposes on source and target domains during the selection of metaphors for visual interactive systems. Among them, there are such as *similarity of properties of source and target domain objects; "visualizeness" (in a broad sense) of source domain; habitualness (recognizability) of its objects; rich structure of interrelationships between objects*.

The concept of habitualness and recognition in the specialized visualization systems should be connected mostly not with everyday realities, but with potential user activity in that sphere for which the interactive system is created. In general, computer metaphors may refer less as to exact matching of reality than conversely may need in additional "irreal" (or "magic") opportunities. "Magic" in the computer metaphor means that "metaphorical" interfaces and visualizations that do not imitate prototypes from real world. The presence of "magic" attributes in a metaphor means that its target domain has properties nonexistent in the source domain. "Magic" in metaphors is closely related to the conception of intuitively usable interface. The "correct magic" of the interfaces and visualizations has to be based on this principle of intuitive usage. Understanding of the magic is interlinked as of cultural background of potential users as of context of using interfaces and/or visualizations. In connection with this context, one should be paid attention to the requirement of the metaphor naturalness. There are a variety of approaches to appraisal of its role. Some authors consider as metaphor such as only those where source domains have based on everyday realities. Really such metaphors, for example, Mosaic, Information Wall, Fish Tank, gain widespread acceptance in interfaces and in information visualization systems. But no less frequently then "natural" (real life), the "quasi-natural" (habitual for a given domain) imageries are used in visualization systems. There are such examples as the techniques of molecule depictions in chemistry or biology. Also, one may consider the visual formalisms as some kind of metaphors. Such visual formalisms as flow charts, data flows, Petri nets, etc. are actively used in diagrammatic visual programming languages. The visual formalisms have abstract imageries but these imageries are interpreted monosemanticly by users-specialists.

We consider the metaphoricalness of any visualization. In our opinion, in the general case, there are no "metaphorless" visualizations of computer models and program entities. The survey of the corresponding bibliography shows on "pictureness" of all metaphors and accordingly on metaphorness of any images in computer visualizations. Per se every computer visualization may be considered as a metaphor because it associates model entities and images and represents one by another for adequate user interpretation. One may show the community of metaphor design and usage in all subdomains of computer visualization. In the case of visualization metaphors, the transition to some world of visualization, where imageless objects obtain their visual representations, takes place. The process of metaphor generation (metaphorization) first of all includes (may be implicit) analysis of target domain of the future metaphor. The hierarchical structure of object interrelations of target domain and their properties is revealed on a basis of the metaphor objects and its properties. At the following stage, a source domain and its main object are searched. Criteria of a choice are criteria of metaphor quality.

First, the main object of a source domain should have the properties, similar (closed) to properties of metaphorization object. The structure of these object interrelations and its properties should be *similar* to structure of interrelations of object under metaphorization and its properties, at least on the first level of a structural tree. Second, a source domain should be visualized. That means that the nature of the *source domain* should be like, that its objects have dimension, extent, length, form, color, or other visual characteristics. For example—a metaphor of the railway for the functional description of operational systems.

5. Metaphorical domain

The metaphorization is based on the interaction structures of the source and target domains. During the process of the metaphorization, some objects of the target domain are structured on the example of objects of the target domain, and there is a metaphorical mapping (projection) of one domain onto another. Moreover, not all objects are selected (and not even all of their properties, or structure elements), but only those that are the most interesting for us. The analogues of these objects are searched in the source domain (in frameworks of structures, the qualitative properties, etc.). Further, the following operation takes place. The objects of the target domain together with the object from the source domain are located now in the common "metaphorical domain" or more exact in doing so this "metaphorical domain" is generated. In this domain, the investigated object now starts to function. It is possible to consider, that it is already a new object of a new domain. The metaphorical domain gets autonomy from the domains generated in it. Many properties of its objects only mediately are connected (if at all are connected) to the properties of the source domain objects. By means, the projection of some characteristics of the target domain onto the source domain its own logic of development of metaphorical domain appears. So, for example, the use of the scientific metaphor of an electromagnetic field its intensity is studied. But it is obviously absent on a field of wheat.

There are the questions: what are the nature and the structure of the metaphorical domain; how its generation is produced? First of each metaphor generates some sign system, that is the integral sign set, in which exists the internal relations between the signs somehow that map the relations between the designates. Our metaphorical domain as a matter of fact is a sign system. The understanding of a metaphor as a sign system gives us the basis for the evaluations of the metaphors offered in the concrete cases. If the used affinity (comparison or a set of comparisons) matches the systemness requirements, then we may speak about the existence of a useful metaphor. If not, if the conditional changes of the source domain objects are connected with the changes of the objects from the target domain poorly, then such comparisons usage cannot help us to understand an investigated the situation better. In case of a metaphor, the generation of a sign system is possible to consider as the adaptation of two metaphor operators, the basic:

> *"Let A be similar to B"*

and the additional operator:

> *"The following attributes/elements/characteristics of A are selected for assimilation to the following attributes/elements/characteristics of B."*

Here A stands for a source domain, whereas B stands for a target domain.

6. Metaphor action and metaphor formula

Let us define the concept of "*metaphor action*" to describe the [potential] results of metaphor uses. This conception allows to analyze structurally specific computer metaphors. In turn, the analysis is necessary to understand causes of the successes of one and the failures of another visualization and interface metaphors. Further, the analysis of the logic of metaphor searching and choice enables to formulate the evaluation criteria for the "human factor" aspects of visualization systems.

The concept of "*metaphor action*" is connected with answers to the following questions:

- *"How can this metaphor assist to represent the information?"*
- *"How can this metaphor assist to interact with data or to manipulate them?"*
- *"What properties of metaphorical objects (that is visual and/or dialogue objects generated by the metaphor) take place?"*
- *"What actions or ideas are arisen from the process of the user interaction (including observations of pictures) with metaphorical objects?"*

It is possible to construct a "formula" of the metaphor actions. The metaphor "formula" includes simplified descriptions of the source and target domains, an idea of likening using in the metaphor and the results of metaphor actions.

Note that the computer metaphors do not need to obtain the completeness and precision of similarities. Therefore, in formula (as in a metaphor), only a limited set of required objects is described.

In the general case, metaphor formula is as follows:

> **Source domain**: *description [+ set of the objects participating in a metaphorization]*
>
> **Target domain**: *description [+ set of the objects participating in a metaphorization]*
>
> **Idea of likening**: *{object of Source domain_1} = {object of Target domain_1}*
>
> …

{object of Source domain_n} = {object of Target domain_n}

{operations over objects of Source domain_1} = {operations over objects of Target domain_1}

…

{operations over objects of Source domain_n} = {operations over objects of Target domain_n}

*[***Magic idea***]: the description additional, often impossible in reality, but useful properties of new objects and/or operations over them.*

*[***Result***]: the description of resultant (metaphorical) domain with a set of objects and operations over them.*

The purpose of our analysis is to reveal the structures of the successful metaphors and to build the basis for the comparison and evaluation of metaphors. Such concepts as "metaphor action" and "metaphor formula" are considered to construct the basis of analysis. We begin our analysis with one of the most popular "desktop" metaphor. Originally, this metaphor was offered for the office automation systems, but then it was expanded for the general case of the interface for operating systems. "Desktop" metaphor in the 90th years of the twentieth century became the most frequent practice. This metaphor is in many respects a basis of the modern visual interfaces. The success of the "desktop" metaphor, undoubtedly, is connected not only (and not so much) with the natural figurativeness of icons those are [not always] clear to users, but with logicality and systematicity of all activity in the frameworks of visual environments based on this metaphor. The "desktop" metaphor generates the unfussy sign system that is the base of the corresponding metaphorical domain.

In the case of desktop metaphor, the formula may be written as follows:

Source domain: The desk with folders containing documents (documents are structured, but folders may be disordered);

Target domain: The office automation system;

Idea of likening: "Folders with papers" = "structure of the data, a set of files";

"Opening of a folder" = "demonstration of file structures and/or files";

"Processing of documents" = "execution of functions, by means commands of the visual language."

Result: The direct access to the data structures by means of the manipulations of icons placed on the screen; calls of some [user] predetermined functions by means of the visual dialog language. The early versions of Microsoft Windows use the extended version of this metaphor.

Addition of source domain: The desk is combined with the control panel where starting buttons are placed. Besides the "magic" idea is added: all actions within the framework of system are made by means of **double click** on icons.

Result: icons those can represent the data structures and the programs calls.

The data structures and programs are executed in the same way corresponding to the classic von Neumann computer.

There is also one more idea: opening of the new windows when program execution begins. One can speak about carrying out of the "metaphorical" interface domain constructed on the basis of the desktop realities. However, not every entities of the real desktops (the source domain of the metaphor), which are richer and poorer than the metaphorical objects in the same time, were equally useful in the new metaphorical domain. Often icons moving on the screen are needed only for grouping and for the convenience of the concrete user. The images of folders do not play a main role in the users' actions with operational system and frequently they are not placed on "desktop." But the major value (not having analogues in initial area) double "click" using for program starts has obtained. Usually the double "click" results is an opening of new window, and, in the case of Internet-browsers windows are opened in almost a literal sense. As the result, we have the logical commands system of the visual (iconic) language, based on the set of icons and "Double-Click" operation.

7. Properties of visualization metaphors

Objects of the new metaphorical domain, the relationship between them and the possible actions in this domain have a number of properties, which we call metaphor properties. The success or failure of visualization systems depends on many factors. One approach to the evaluation of visualization involves the examination of properties of visualization metaphors. We analyze the properties to consider the possibility of metaphor using for specific applications of software visualization. It is important to understand what objects may be represented with one or another metaphor. We need to analyze the possibility of the visualization metaphors (more precisely—the views based on the visualization metaphors) to represent large and huge volumes of data and details required to understanding the program's operations. The positive effects of a 3D display and virtual and augmented reality environments are possible in these cases. Therefore, it is important to analyze possible applications of metaphors in the frameworks of visualization systems using modern computer graphics environment, in particular the virtual reality environment. For all this, we need to describe how to verify the suitability of metaphor for solving problems under consideration.

Note that such metaphor properties such as "ability to contain any objects inside itself", "restriction of a perception context", "closeness", "inclusion in structure", "presence a structure inside", and "naturalness of a metaphor." These properties are connected with using within the framework of metaphors such basic image-schemas, as CONTAINER, UP-DOWN, and BACKFRONT. These image-schemas and other visual characteristics are the base of depiction techniques in many visualization systems. We will analyze the metaphor properties by the example of the hierarchical sequence of the natural boom-building-city (landscape) metaphors and the molecule metaphor. These metaphors are used in a variety of information visualization and software visualization systems.

8. Properties of room metaphor

In the beginning, room metaphor was considered as an extension of desktop metaphor. In this case, data and system control objects were placed inside 3D room space. Objects in the room link to specific information (see **Figure 6**) [28]. Furthermore, room metaphor is used to depict objects of software visualization systems and systems of visual programming.

The room metaphor possesses the following properties:

1. *Ability to contain any objects inside itself*. The room not only represents the separate object but is also the container for others ones.
2. *Restriction of a perception context*. The objects inside the room are considered in the separation from "external worlds."
3. *Closeness*. There are no any additional elements to use the room metaphor (excepting possible inner objects).
4. *Inclusion in structure*. It is possible "to build buildings of rooms," that is to consider the set of rooms. Therefore, the room may be an element of construction of some complex construction.
5. *Naturalness of a metaphor*. The room is the natural metaphor, with the presence of the corresponding objects in the real world. This property makes intuitively understandable all properties described above. There are no additional analogies and unnatural images. The functionality and characteristics of the real objects are transferred in the virtual world with only minor extended understanding.

It is possible to consider various ways of objects locations inside a room using the container property. The information may be represented by the type of the objects without considering

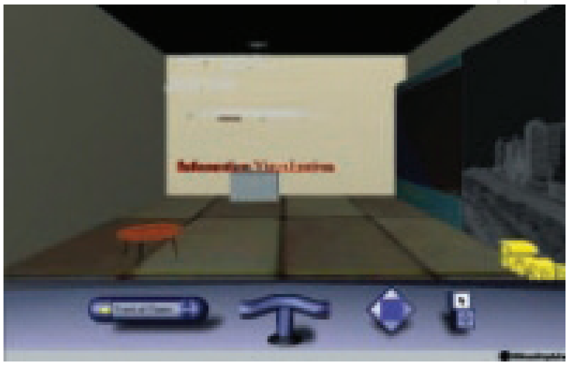


Figure 6. Data vault based on the 3D room information space metaphor [29].

their location. One-type objects may be represented by their location in the room. It is more natural to place the visual objects onto “walls” of the room. Also, it is possible to use for the information representation the location of 3D objects indoors. Certainly, one may use both methods together and additionally forms and colors of objects. The collection of rooms may represent a set of program classes. It is possible to observe the dynamics of the program execution “on the inside” by using the special form to the depict kinds of the program constructions. The color in the room may be determined on the base of the contents of the “room-function” (see **Figure 7**).

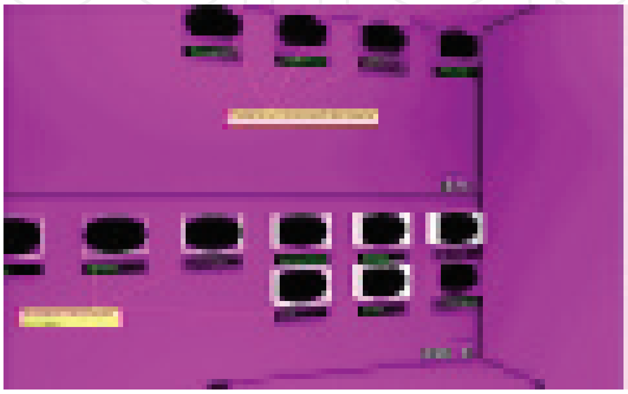


Figure 7. Example of room metaphor realization to represent the links of program elements [30].

One may consider a set of different types of rooms. In this case, the connection between the rooms may represent the structural relationships in the complex object. It is also possible to provide a predetermined, strictly defined location in the room space (wall, skyscraper, etc.). However, such arrangement can represent less information about the rooms forming the structure. The dynamical change of the characteristics of the room may be an additional source of the information. It is possible to use the animation at all rooms. In addition, the animation may affect not only the change in space, but also other characteristics of the room: object colors, sizes, shapes, etc.

9. Building metaphor

In its turn, building metaphor may be considered as an extension of room metaphor. As a rule, visualizations based on this metaphor were represented a structured set of rooms and other accommodation. This metaphor is used to represent in information visualization (see **Figures 8** and **9**) and software visualization systems (see **Figures 10–12**). There is interesting (although rather old) example of building-like metaphor in the case of software visualization. In avatar system, virtual reality was used for performance analysis of parallel systems. The user was inside a “room.” The performance analysis data are depicted on its “walls” and “floor.” The user could be moving between “rooms” in some paths (see **Figure 13**).

Building metaphor possesses the following properties:

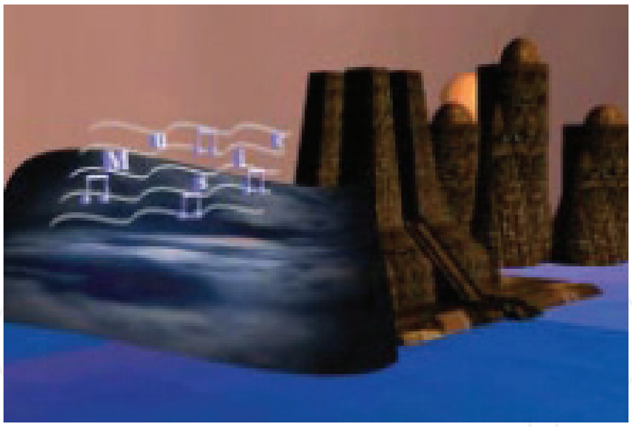


Figure 8. Visualizing a "building" containing the query result [31].

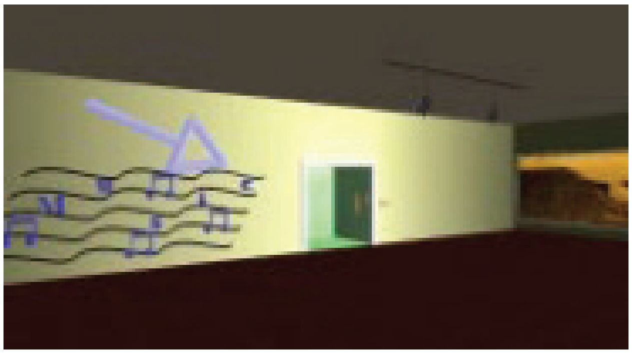


Figure 9. Inside the "building" [31].

1. *Ability to contain any objects inside itself*. The building is the container for others objects. In comparison with the room metaphor, the building metaphor possesses bigger "depth." This metaphor suggests not so much the presence of some visual information objects as the presence of containers with the objects.

2. *Restriction of a perception context*. Everything that is placed inside the building is perceived as connected in a whole, affinitive through some characteristics.

3. *Closeness*. The building metaphor inherits closeness property of the room metaphor in the sense that in the frameworks of this metaphor it is not required external objects, however, the internal filling of the building is very important.

4. *Inclusion in structure*. This property is similar to the corresponding property of the room metaphor. It is possible to construct the city including single building or collecting them in structures (city quarters).

5. *Presence a structure inside*. It is necessary to distinguish the use of the building metaphor and the multiple arbitrary structured uses of the room metaphors. The building in this sense has quite fixed structure in the kind of a location of "rooms" on "floors," and also a set of variations in the structure of each floors, for example, available general "corridor"

between them in the hotel metaphor, in the strict location of rooms of rather up-down neighbors, etc.

6. *Naturalness of a metaphor*. The building is also the natural metaphor. There is an analog for it in the real world. The metaphor does not associate additional analogies and unnatural images.

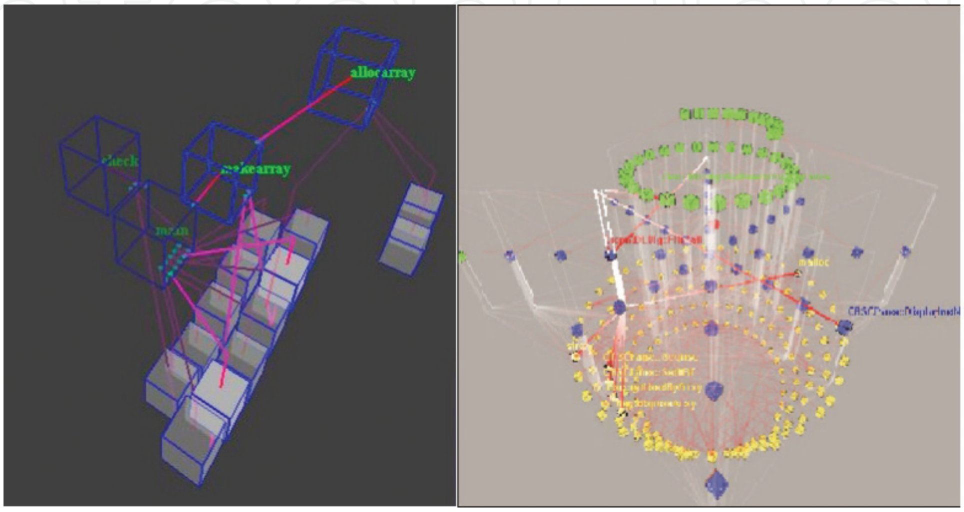


Figure 10. Call graph visualizations based on building metaphor [30].

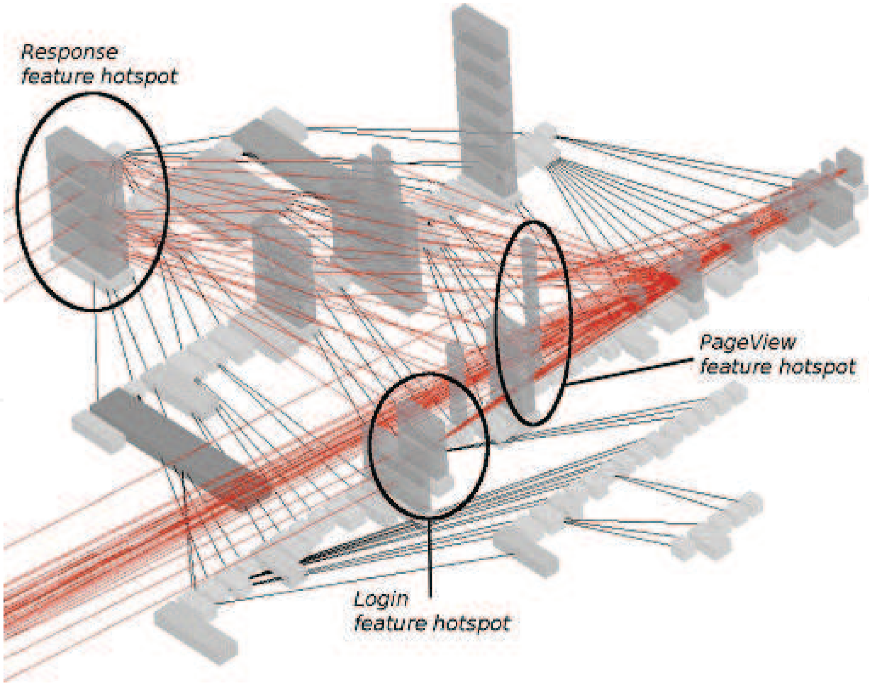


Figure 11. Building metaphor using in software visualization system.

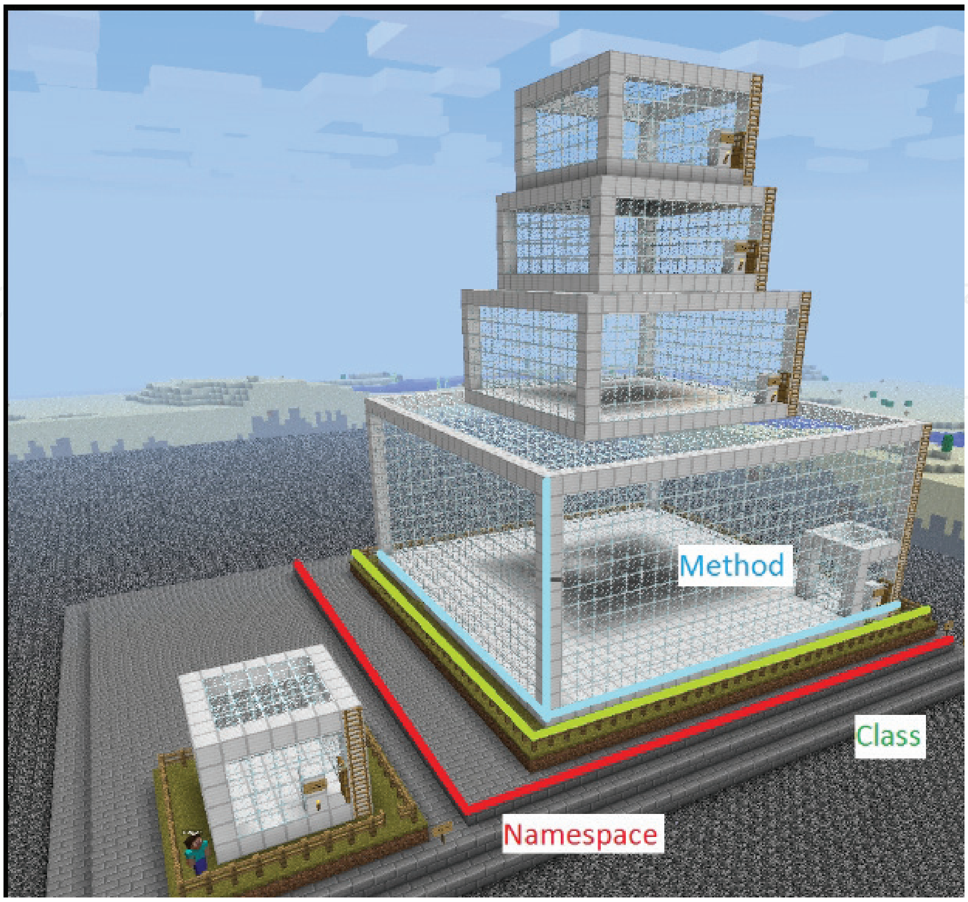


Figure 12. The depiction of program code [32].

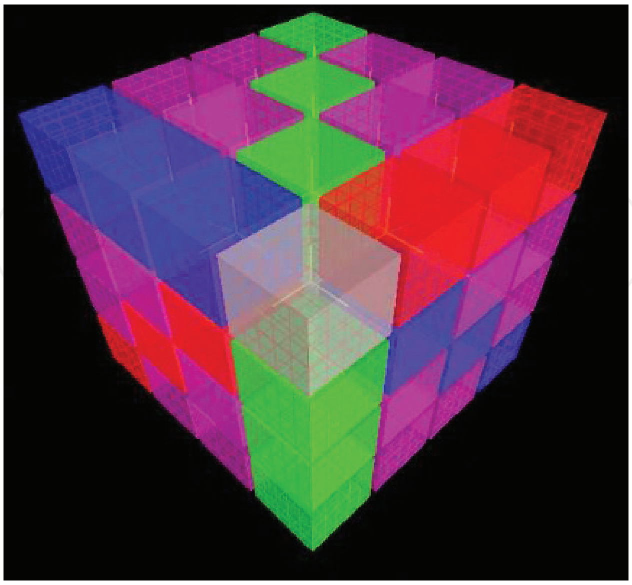


Figure 13. Scatercube—the 3D extension of 2D scatterplot and scatterplot matrix [33].

10. Properties of city and landscape metaphors

City and landscape metaphors are well-known beginning the 90th of twentieth century as in information visualization as in software visualization systems. For example, the city-like metaphor was used to visualize hierarchical graphs (see **Figure 14**). As early as 1993, the landscape metaphor was used to represent a corpus of documents (see **Figure 15**). Note that ideas of information landscape are very popular in information visualization [34–36]. Relationships between the individual objects (e.g., articles) are identified using citations, descriptive terms, or textual similarities. Objects are then clustered using a force directed placement algorithm to produce a terrain view of the many thousands of objects (see **Figures 16** and **17**). Also city and landscape metaphors are actively used in software visualization. Urban streetscape may represent the progression of the program system development [37] (see **Figure 18**).

City metaphor and similar landscape metaphor (and their modifications such as industrial landscape metaphor and factory metaphor are popular in software visualization systems to represent execution traces and call graphs of parallel programs (see **Figures 19**, **21** and **22**). Note the interesting idea united the metaphor of hierarchical edge bundles and city metaphor. In Ref. [40], the adaptation of the existing two-dimensional (2D) hierarchical edge bundles technique to represent relations in a 3D space on top of city metaphors is described. This visualization technique inspired by the 2D hierarchical edge bundles technique is converted into 3D hierarchical attraction points, which affect edge paths across the city visualization. In this way, edges are grouped together, resulting in a more understandable visualization of relations (see **Figure 20**).

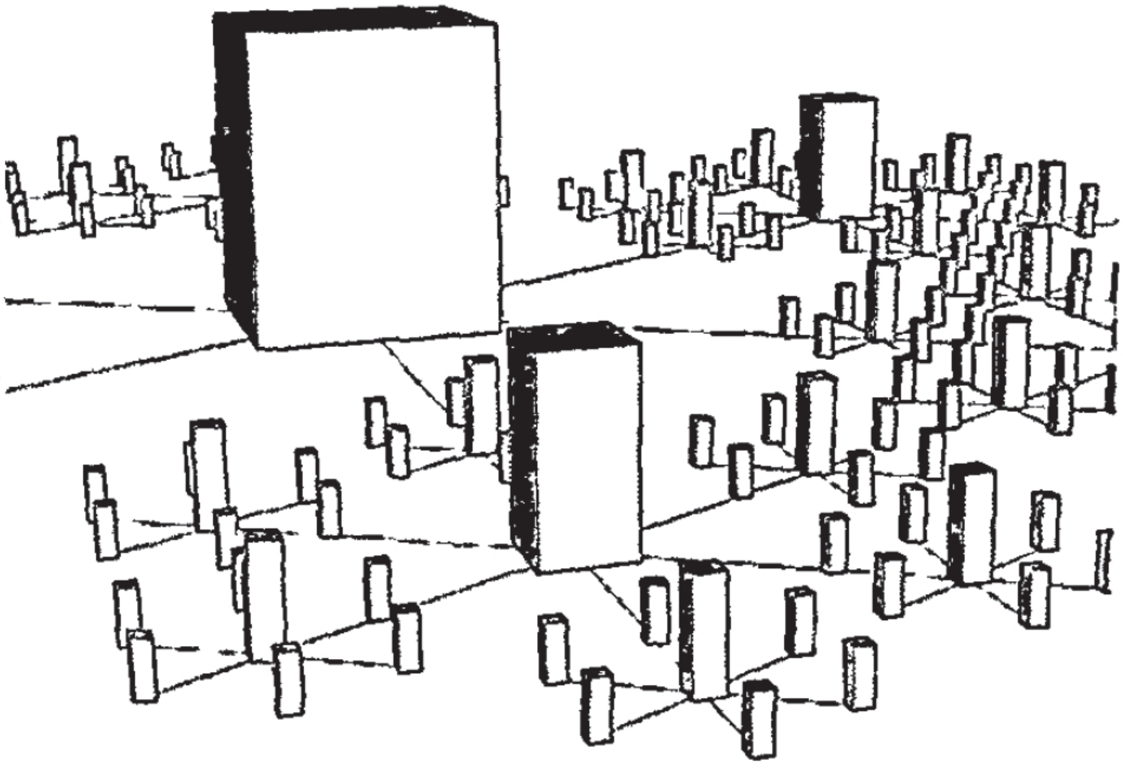


Figure 14. Viewing details in the cityscape-visualization of the tree [38].

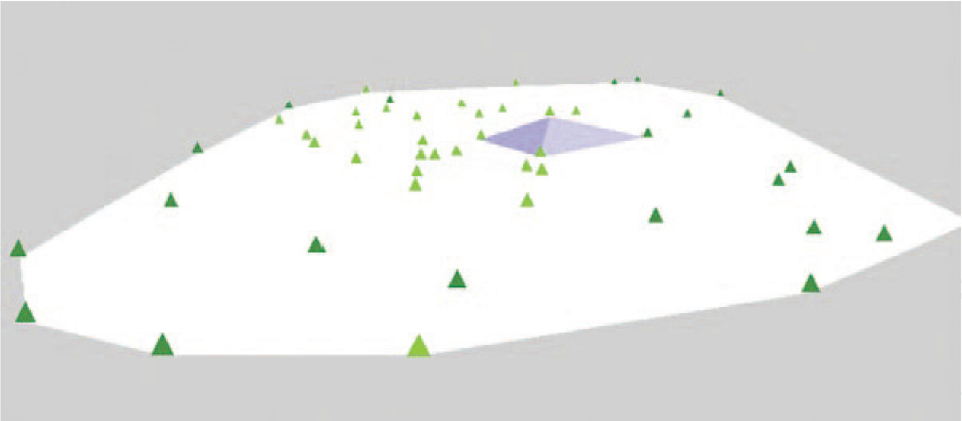
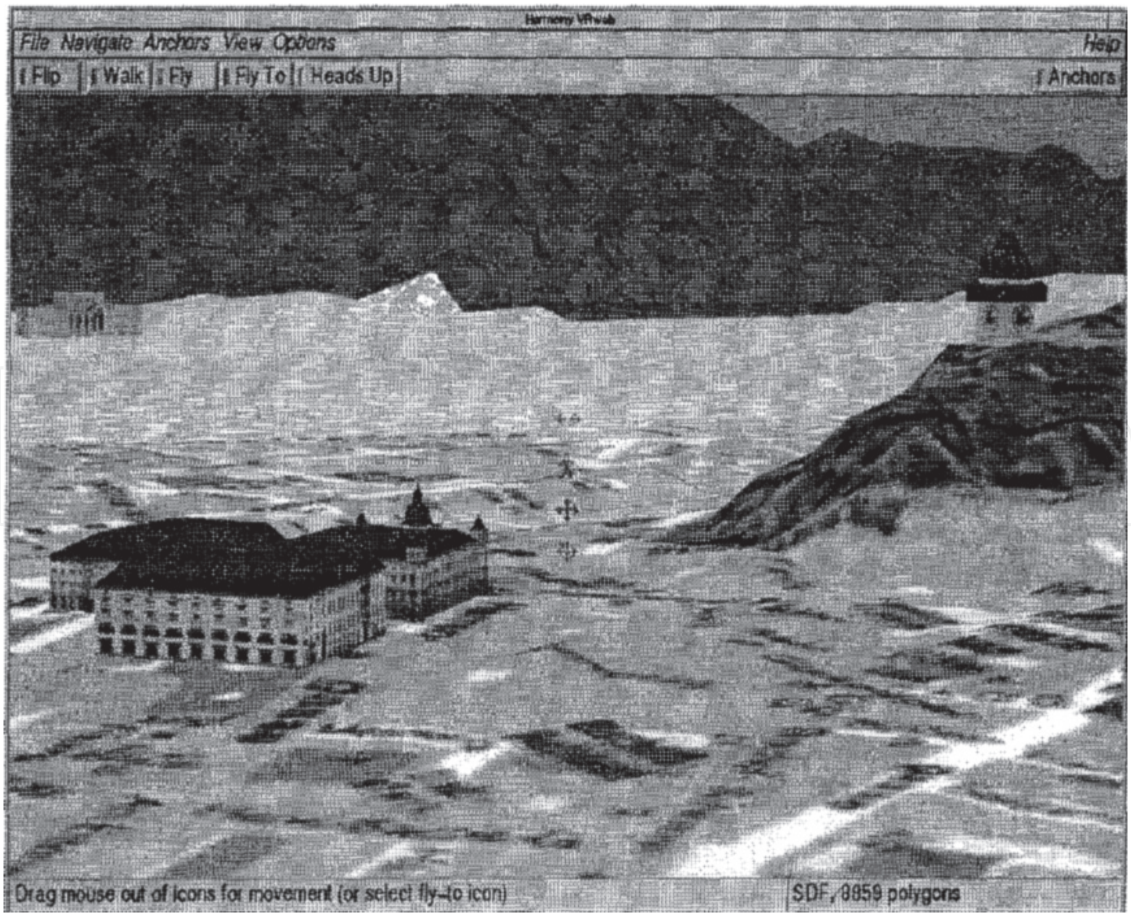


Figure 15. An information landscape bead system, depicting articles from an HCI conference (CHI'91) [39].

One may consider the following properties.

Unlimited context

The user context is not limited artificially in *City* and *Landscape Metaphors*. As a result, additional user's efforts are required to identify needed objects among many others. At



Drag mouse out of icons for movement (or select fly-to icon)

SDF, 8859 polygons

Figure 16. Harmony's VRWeb 3D Viewer [34].

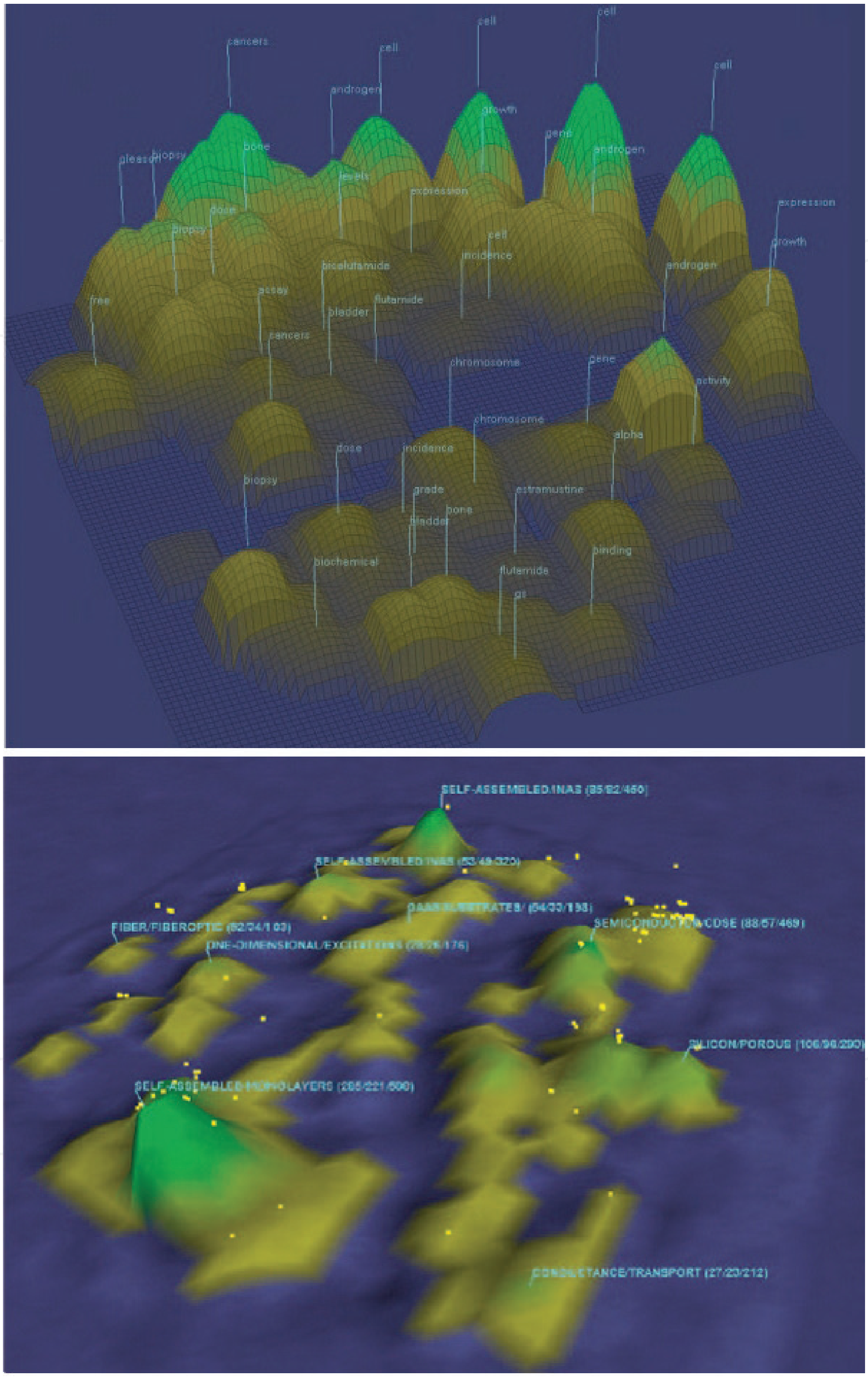

cancers
cell
androgen
cell
growth
cell
gene
androgen
cell
gleason
biopsy
bone
levels
expression
dose
biopsy
bicalutamide
incidence
cell
androgen
expression
growth
free
assay
flutamide
bladder
cancers
chromosome
gene
activity
chromosome
alpha
dose
incidence
biopsy
grade
estramustine
bone
bladder
binding
biochemical
flutamide
gs
SELF-ASSEMBLED/INAS (85/82/450)
SELF-ASSEMBLED/INAS (53/49/320)
GAAS/SUBSTRATES/ (54/33/158)
FIBER/FIBEROPTIC (82/34/103)
ONE-DIMENSIONAL/EXCITATIONS (28/26/176)
SEMICONDUCTOR/CDSE (88/67/469)
SILICON/POROUS (106/96/290)
SELF-ASSEMBLED/MONOLAYERS (285/221/590)
CONDUCTANCE/TRANSPORT (27/23/212)

Figure 17. Screenshots of ThemeScapes (up) and VxInsight (down) information landscapes [37].

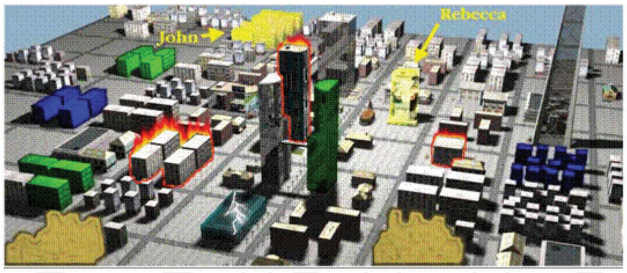
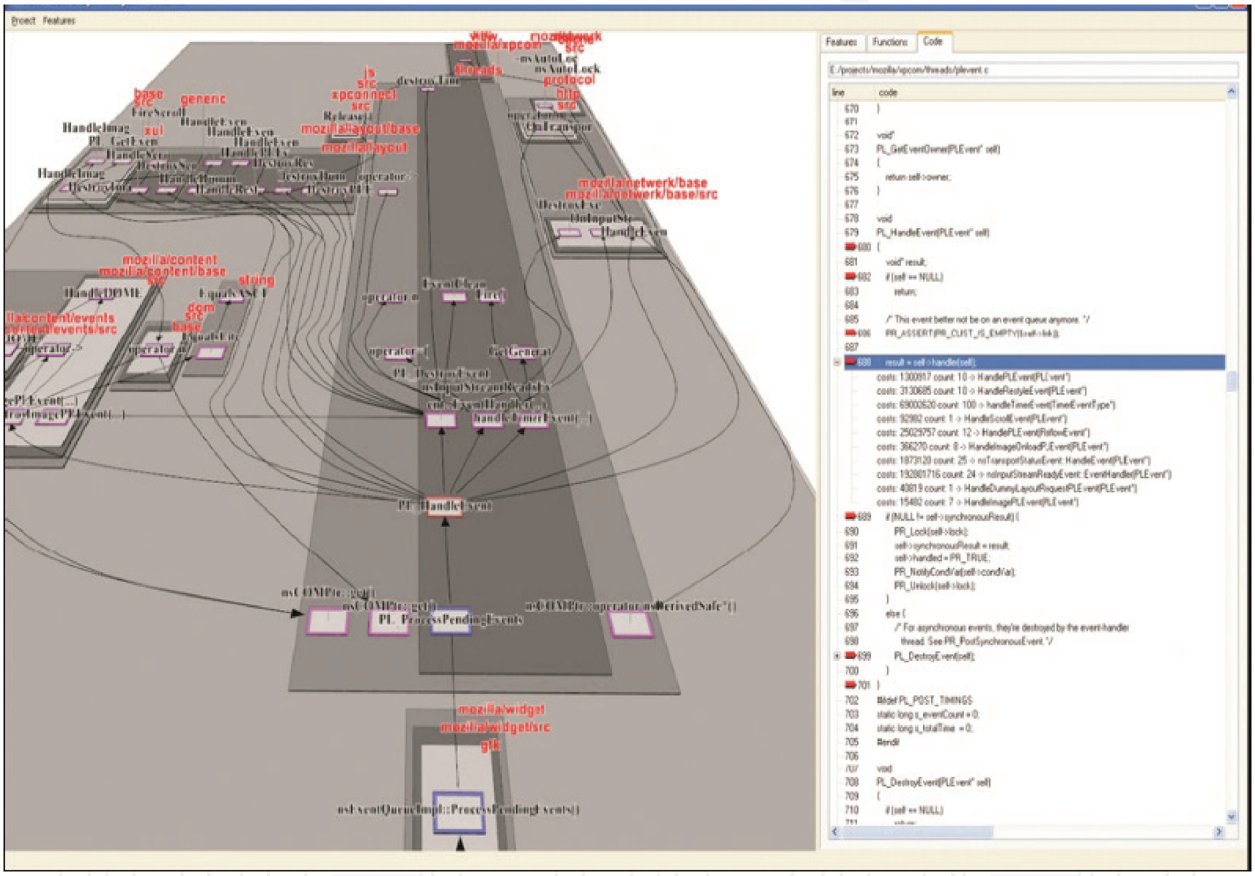


Figure 18. Depiction the program system development [37].



E:/projects/mozilla/xpcom/threads/plevent.c

```
line  code
    }

    void*
    PL_GetEventOwner(PLEvent* self)
    {
        return self->owner;
    }

    void
    PL_HandleEvent(PLEvent* self)
    {
        void* result;
        if (self == NULL)
            return;

        /* This event better not be on an event queue anymore. */
        PR_ASSERT(PR_CLIST_IS_EMPTY(&self->link));

        result = self->handler(self);
      costs: 1300917 count: 10 -> HandlePLEvent(PLEvent*)
      costs: 3130685 count: 10 -> HandleRestyleEvent(PLEvent*)
      costs: 69000620 count: 100 -> handleTimerEvent(TimerEventType*)
      costs: 92982 count: 1 -> HandleScrollEvent(PLEvent*)
      costs: 25029757 count: 12 -> HandlePLEvent(ReflowEvent*)
      costs: 366270 count: 8 -> HandleImageOnloadPLEvent(PLEvent*)
      costs: 1873120 count: 25 -> nsTransportStatusEvent::HandleEvent(PLEvent*)
      costs: 192801716 count: 24 -> nsInputStreamReadyEvent::EventHandler(PLEvent*)
      costs: 40819 count: 1 -> HandleDummyLayoutRequestPLEvent(PLEvent*)
      costs: 15482 count: 7 -> HandleImagePLEvent(PLEvent*)
        if (NULL != self->synchronousResult) {
            PR_Lock(self->lock);
            self->synchronousResult = result;
            self->handled = PR_TRUE;
            PR_NotifyCondVar(self->condVar);
            PR_Unlock(self->lock);
        }
        else {
            /* For asynchronous events, they're destroyed by the event-handler
              thread. See PR_PostSynchronousEvent. */
            PL_DestroyEvent(self);
        }
    }
    #ifdef PL_POST_TIMINGS
    static long s_eventCount = 0;
    static long s_totalTime  = 0;
    #endif

    void
    PL_DestroyEvent(PLEvent* self)
    {
        if (self == NULL)
```

Figure 19. Complex graphical view and textual code view of call graphs based on factory metaphor [41].

the same time, the *unlimited context* allows a quick look at the entire "picture" and quickly identification of key elements.

Naturalness

It is known that naturalness of a metaphor reduces efforts on the resultant image interpretation.

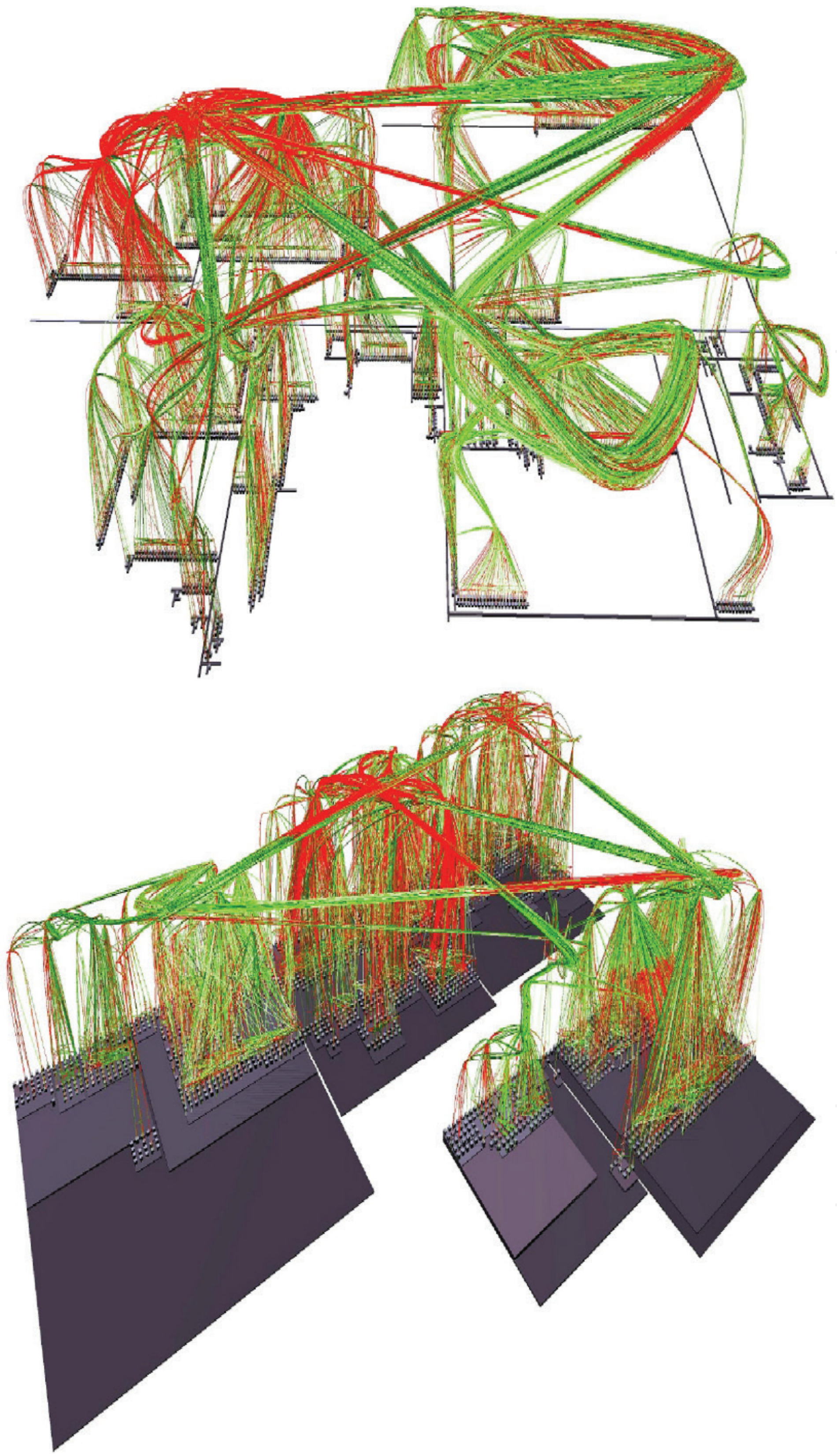


Figure 20. Visualization of dynamic call relations on an execution of JEdit, Java JRE classes included. 2710 classes, 10,870 edges representing 4,632,680 calls [40]. Up: Relations on top of the nested layout of the software city metaphor. Down: Relations on top of the street layout of the software city metaphor.

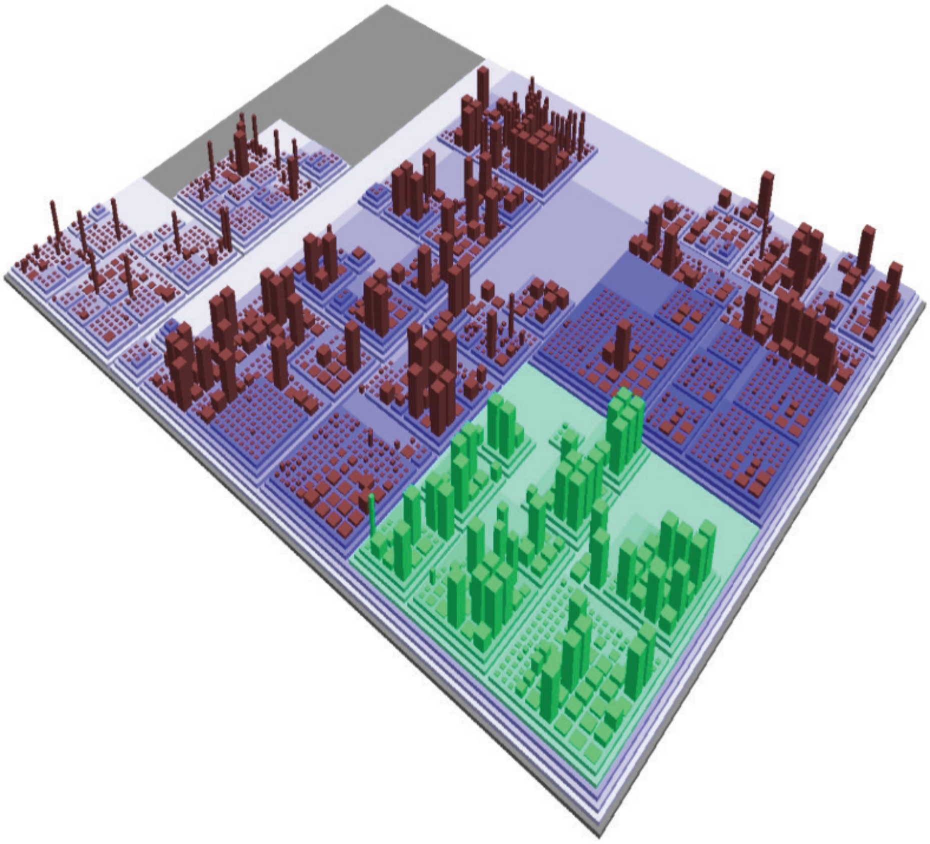


Figure 21. The example of city view of the software system [42].

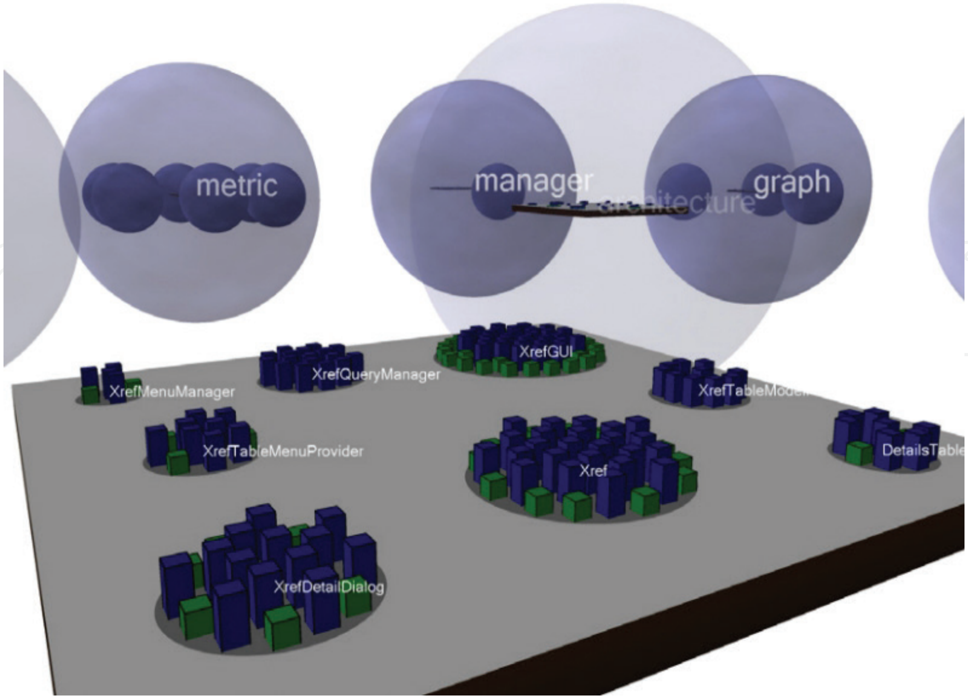


Figure 22. The example of landscape view of the software system containing packages, classes, methods and attributes [43].

In the cases of *city* and *landscape metaphors* not only naturalness of spatial orientation, but also naturalness of navigation takes place. In case of a city metaphor, the method of navigation is defined by the metaphor itself.

Organization of inner structure

Metaphors suggest the existence of an inner structure. In case of a *city* metaphor, this structure is dictated by the metaphor itself, and it is defined rather rigidly—there are buildings, quarters, streets, and districts. In *landscape* metaphor, a structure choice is nondedicated. In this case, one may say about landscape nesting.

Key elements

Metaphors suggest a representation of large volume of information, and in most cases, this information is rather homogeneous in visual sense. Users need the key elements to interpret this information. If we want to use a metaphor to reveal specific features and/or exceptions (for example bugs in programs), these elements have to be depicted by easily distinguished image-keys. One may design some key elements in frameworks of city or landscape metaphors. In these cases, some forms of guidance signs or markers may be used as key elements.

Resistance to scaling

These metaphors are stable in the case of increase in information volumes. Moreover, applications of city and landscape metaphors are reasonable only in the cases of large information volumes. In the cases of *city* and *industrial landscape* metaphor transport corridors help to design software visualization systems. Transport corridors may be used as means to represent control flows, data flows, and other relations between program constructions or parts of program complex.

Note that unlike in the case of landscape metaphor, the choice of city metaphor strongly limits the set of possible views. Thus, city and landscape metaphors may form base to represent considerable volumes of the structured information with identifications of specific interest cases that is necessary in the systems for performance tuning and program debugging for parallel computing. Additionally possibility to fly over a city/landscape creates prerequisites to easy navigation. Flight with changes of height allows to carry out scaling and zooming. Interpretation of the graphical displays based on these metaphors seems to be simple.

11. Properties of molecule metaphor

Now, let us consider the properties of molecule metaphor that also may be used to visualize call graphs of parallel programs (see **Figure 23**). This metaphor may be used in software visualization, for example, to visualize dynamic object relationships in Java programs. The metaphor of a chemical molecule is used to aid comprehension and to help in reducing the size of the object graph [44] (see **Figure 24**).

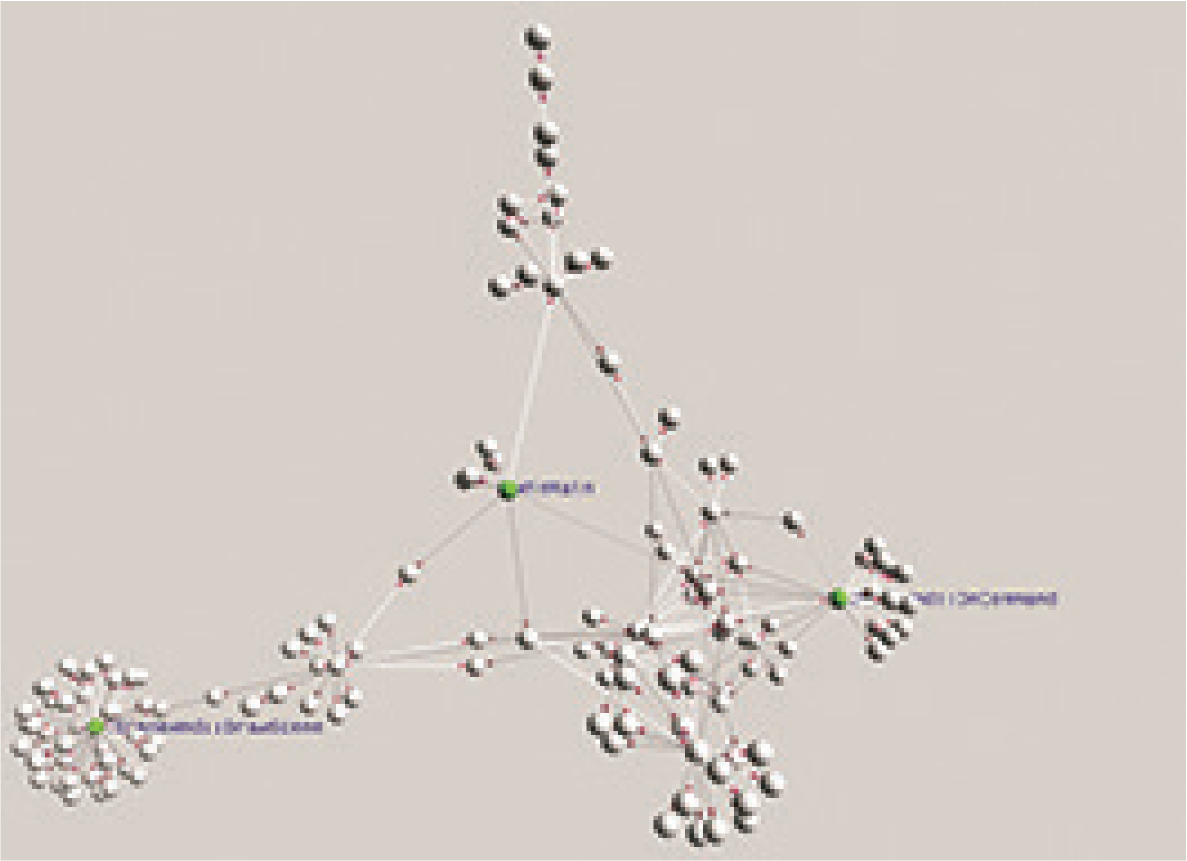


Figure 23. Call graph visualization based on molecule metaphor [30].

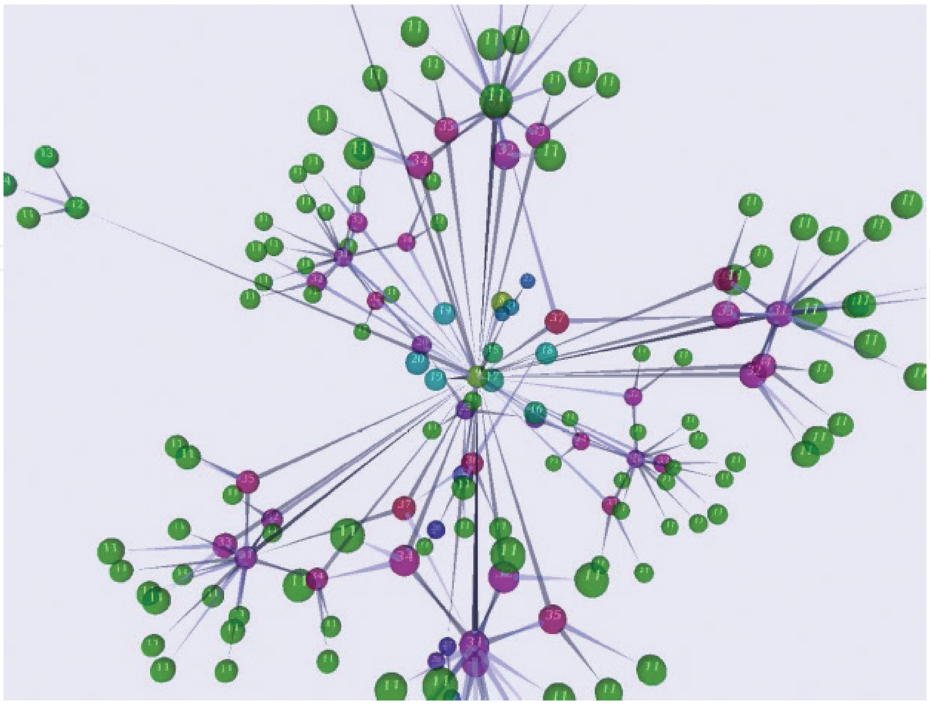


Figure 24. The main cluster from the object diagram [44].

Three-dimensional visualization techniques may improve the quality of graph structure perception. Suggested idea is to search analogies with natural objects. Let us place in nods-functions (which is usually represented like spheres) an electrostatic charge [30].

Connections between nods are replaced by elastic interaction. Let's name the metaphor "a molecule metaphor" because at the given approach the visualization similar to the structure of benzol molecules. Thus, there are two types of interactions: springy between bound nods and electrostatic between all other nods-"atoms." The electrostatic interaction may reflect temporary features of the calling functions, then springy—a number of calls. The consideration of the "molecule" energy allows us to construct the effective drawing algorithm for about thousands of objects. The displays meet the symmetry criteria. Animation (molecule rotation) allows exploring graph structure better. Color may be used for accentuation of interesting features of visualized graphs. The molecule metaphor is constructed on analogies to natural objects [24]. There is an interesting example of using landscape metaphor for visualization of molecular similarities [45].

Physical particles metaphor substantially similar to molecule metaphor may use in software visualization [46] (see **Figure 25**).

The metaphor supports the selection of key elements for example, by coloring or size changing of the molecules elements and changing of thickness of communications between them (key elements). Moving and navigating in pictures related to molecule metaphor may be

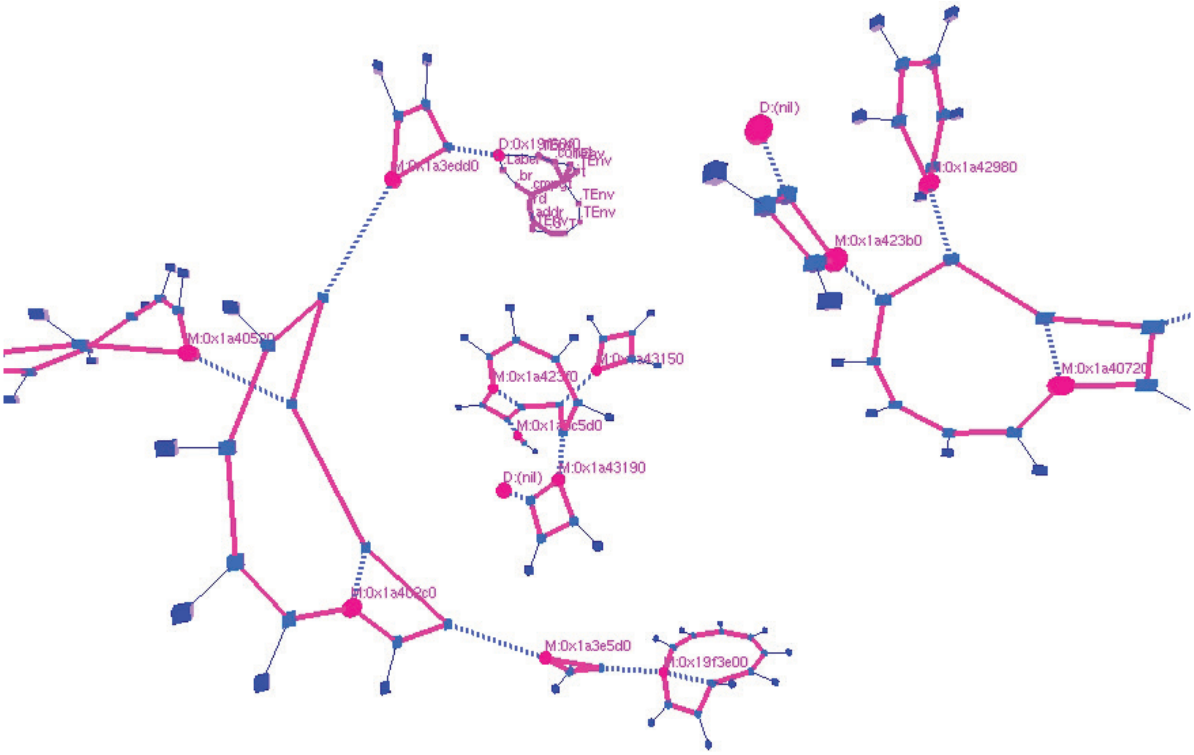


Figure 25. Compilation graph visualization based on physical particles metaphor [46].

performed by flying around molecule. There is the experience of visual "entering" a separate "atom" and viewing internal visual information inside a single sphere [46]. It is possible to implement similar "entering" spheres in the frameworks of virtual reality environments (organization of inner structure) (see **Figures 23** and **25**).

Note that in general, the use of virtual reality enhances the visualization based on *building*, *city*, *landscape*, and *molecule* metaphors.

12. Context, interpreter, and interpretant

According to semiosis, a metaphor defines techniques of designation and an imagery of visualization. Also, a metaphor defines a context of interpretation. Interpretation of visualization (and also interactive manipulations) based on given metaphor reconstructs (or creates anew) a set of user's cognitive structures in which the picture of the phenomena is represented. A process of interpretation is exactly the generation of representative cognitive structures on bases of the visual images. This process is inverse or more exactly dual to visualizations.

Process of visualization, in turn, is considered as construction of visual (geometrical) images on the basis of abstract representations of objects. These abstract representations are the model of objects under researches, the phenomenon, or the process, somehow connected with the user's cognitive structures that describe these entities.

The context is defined as a metaphor, and an individual of the interpreter. The interpreting context defined by the metaphor is revealed in the individual of the user of visual systems—the interpreter of the sign visualization process.

The answer to a question "who is the interpreter of visual texts?" defines that part of a context which depends on the interpreter. Against this background user modeling is very important. One can consider user models of various levels, for example, the general model of visual perception, or by contrast the concrete model of user manipulations with the concrete input device. Now, the research domain of user modeling is "under construction." For obvious reasons, researches related to modeling of users of mass interfaces (such as educational or informational systems, e-shopping, and social network sites) are carried out most actively. Also, there are interesting researches on modeling users of specialized visualization systems, for example, systems based on virtual reality.

As already noted, the meaning of "visual texts" implied by a developer of the visualization system (an author of the text) can be significantly different from the meaning obtained by a user of visualization systems (an interpreter of the text). Thus, in many cases it is impossible to determine accurately the content of interpretant in computer visualization systems. For the design of visualization systems, it is necessary to consider possibility of meaning distortion, appearance of "descriptive artifacts," partial or full misunderstanding of senses implied in visual texts. Development of the user model and its analysis has to help with an explanation of similar negative occurrences, or (better) have to prevent them. On the other hand, there is possibility of some positive occurrences connected with partial determinacy of *interpretant*. These situations are frequent at the first stage of development of some specialized visualization systems when there are not fully understood, the algorithms and methods to implement them, and often there are not clearly defined the mathematical models themselves. The successful metaphor, well designed and developed views of one or another scientific abstractions often allow the user who really understands an essence of the phenomena under researching, to find more valuable meanings, more than interpreted information in the resulted picture, than the designer of visualization supposes. Thus, indeterminacy or partial determinacy of interpretant (if to consider it from the designer point of view) can occur in those cases of computer modeling, when a new, hitherto unknown knowledge about a given application domain are gained.

13. Design of visualization

In summary, let us describe our approach to the semiotics design of visualization systems. Design of visualization itself is the part of the process of the development of specialized visualization systems. This process includes among other such stages as search/choice/designing of visualization metaphors. The next stage is the design views, based on these metaphors. We define a view as the abstraction of a graphic display, containing specification of visual objects, their attributes, their interpositions, possible dynamics, and ways of interaction. After determination "*who is who*" in visualization in terms of semiosis let us translate resulting scheme of semiotics analysis into the language of visualization design for specialized visualization systems.

The first point of our scheme concerns the recognition of denotatum (designatum) in semiosis. For the scheme of design of visualization systems, this point corresponds to such questions as "*what is the goals of visualization?*" and "*what is the subject of visualization?*" Thus, the definition of denotatum is related in the process of visualization to the definition of the objects of special interest, their states, features and specifications, as well as moments of transition from one state to another. Note that the same set of model objects can be visualized in a few views by different methods.

The next point is associated with the search for methods of signification for the denotatum, that is, with the choice of sign.

For the design of visualization systems, it is important to understand that whole graphical display (a picture) rarely appears as a sign. It is necessary to determine, which elements of the image should (and can!) be recognized, understood and interpreted by the user specifically as such. It is known that the choice of imagery for the view is primarily dependent on the visualization metaphor. Moreover, the metaphor sets the context of interpretation.

The context does not exist by itself. In principle, it is subjective, as it bases on the senses of the interpreter. In this regard, let us make one more remark. Signs (or more exactly the text) are interpreted only by those who can do it, who has the necessary knowledge. For example, a hunter "reads" animal tracks in the snow forest clearing and reconstructs exactly the events what happened there. And an inexperienced person cannot do it. Hence, another important question in the design of visualization stage is the following: *"Who is the interpreter of visual texts"*, *"what experiences and what knowledge he has?"*

As already mentioned, there is another important (if not the most important) actor of the design process—the author of visual text (that is, the designer of visualization). She/he should have knowledge of the application domain, allowing precise identification of the main objects of interest to be visualized, and understand what type they are. However, there is an example of the visualization environment, which may independently choose by certain criteria a way of visual representation from a set of the available ones. This environment should be belonging to the class of cognitive visualization systems. Here, the current author of the visual text is the computer program; therefore, it is difficult to say about the presence of some primary, preembedded sense put in the visual text. In the meantime, a user of this system does the analysis successfully and interpretation of pictures, getting new (hitherto unknown information) from presenting graphical displays. Note, once again, that the problem of the source of the interpretant in the visualization process is still not fully explored.

Due to the projection on the process of visualization design, the scheme of semiotic analysis is a useful tool for the design of visualization systems of various types. It was successfully demonstrated their ability to create new visualization techniques.

14. Conclusion

Semiotics approach to the description of visualization does not isolate us from other approaches. On the contrary, the fact that signs have to be recognized, understood, and interpreted, requires the research studies of the perception of signs and their recognition among the other elements of the pictures. These issues are studied in the framework of Gestalt psychology. There are the well-known publications on Gestalt design of human-computer interaction and visualization [47–49]. However, their results are not always taken into consideration by system designers.

Another approach to research studies on visualization is connected with psychological studies. In the report [50], it was noted that the goal of visualization is to leverage-existing scientific methods by providing new scientific insight through visual methods. Visualization should form (or facilitate to form) holistic mental models and as a consequence to create insight. The occurrence of insight is considered as one of the main criteria in evaluating the visualization quality [51].

Ideas of Brushlinskiy [52] on insight are important in connection with analyzing user experience of systems of computer visualization. Insight is regarded as an event, in which an individual "immediately formulates the basic thought has arisen." Also "noninstant" insight is considered. The process of solving a task lies in revealing the relations between its elements, its conditions, and demands. The individual solving a task is performing analysis via synthesis. New characteristics and relations of elements of the tasks are laid out and synthesized with each other, until the solution is at last found.

Cognitive visualization is aimed at helping the researcher to see all elements of the task at hand, to evaluate their relations with each other. One may say that a search for a solution by a user of scientific visualization system largely matches the activity of a researcher busy with a scientific problem, and that may include both instant and noninstant insight. According to Brushlinskiy, in the latter the thought is being formed during several seconds before one's eyes (it is not originally available and is actually formed, not simply formulated). Study of activity of a researcher analyzing and interpreting data with the help of visualization is a major task, which would allow raising the efficiency of computer modeling as whole.

The process of visualization means building a visual image upon abstract ideas of an object. These abstract ideas constitute a model of an object, a phenomenon, or a process researched, which relates to representational cognitive structures of a user, that describe this entity [53]. Visual images representing an entity being modeled serve to create or restore the cognitive structures upon it. The task of visualization is to obtain a visual image, by means of which a mental image (idea) of the object in question may be correctly restored. On the basis of those preliminary structures, the set of views for the visualization system is designed. On the other hand, the specific visual images representing the modeling entities provide for the formation or reconstruction of an updated version of cognitive structures. Generation of *representational cognitive* structures basing on visual images supports interpretation processes.

Consideration of computer visualization and visual human-computer interface in terms of visual communications is another source of analysis techniques and experience. The analysis of visual communication may be also performed from a perspective of semiotics.

It is very significant the problem of formalizing the visualization theory. There are different approaches to the formalization basing as on semiotic as mathematical analysis methods. The problem of the mathematical formalizing the visualization theory basing on category theory and semiotics was posed and considered in the important chapter [54]. In Ref. [55], formal approaches to evaluation of visual texts and visualization effectiveness are considered. Effectiveness of visualization is defined as a multivariable function. The parameters of this function are partial derivatives of visual text by its informative characteristics. Visualization metaphor may be considered as persistence mapping analogous to denotational semantics using in the programming domain. Also, persistence mapping may be defined throw little varying of visualization parameters.

The formal approaches are one of further directions of our researches.

Author details

Vladimir L. Averbukh

Address all correspondence to: averbukh@imm.uran.ru

Krasovskii Institute of Mathematics and Mechanics, Ural Federal University, Yekaterinburg, Russia

References

[1] L. Averbukh. Approach to semiotic theory of computer visualization. The Advances in Computer Science: An International Journal, 4:44-55, 2015.

[2] P.B. Andersen. A Theory of Computer Semiotics. Updated ed. of 1990. Cambridge University Press, New York, 1997.

[3] J. Goguen. Semiotic morphisms, representations, and blending for user interface design. In F. Spoto, G. Scollo, and A. Nijholt, editors, Proceedings of the 3rd AMAST Workshop on Algebraic Methods in Language Processing (AMiLP 2003), pp. 1-15, Verona, Italy, 2003.

[4] M. Nadin. Visual semiosis applied to computer graphics. In Annual Conference Proceedings of the ASEE, Hanover, 1986.

[5] M. Nadin. Cognitive aspects of visualization. Interactive multimedia on the world-wide web. geoinformatik online, 1, 1997. www.gio.uni-muenster.de.

[6] K. Liu, C. Tan. Semiotics in digital visualisation. In J. Cordeiro, S. Hammoudi, L. Maciaszek, O. Camp, J. Filipe, editors, Enterprise Information Systems. ICEIS 2014. Lecture Notes in Business Information Processing, vol. 227. Springer, 2015. Cham.

[7] C.S. de Souza. The Semiotic Engineering of Human-Computer Interaction. The MIT Press Cambridge, Massachusetts, 2005.

[8] C.S. de Souza. Semiotics and human-computer interaction. In M. Soegaard, R.F. Dam, editors, Encyclopedia of Human-Computer Interaction. The Interaction Design Foundation, Aarhus, Denmark, 2012. http://www.interaction-design.org/encyclopedia/semiotics_and_human-computer_interaction.html

[9] J.A. Goguen, D. Harrell, Information visualization and semiotic morphisms. In G. Malcolm, editor, Multidisciplinary Approaches to Visual Representations and Interpretations, pp. 83-98, Elsevier, Oxford, United Kingdom, 2004.

[10] J. Bertin. Semiology of Graphics: Diagrams, Networks, Maps. University of Wisconsin Press, Madison, WI, 1983.

[11] V.L. Averbukh. Visualization metaphors. Program and Computer Software, 27:227-237, 2001.

[12] V.L. Averbukh. Toward the theory of computer visualization. Computational Technologies, 10(4):21-51, 2005.

[13] V.L. Averbukh. Theory as a bridge between education, research and development in human-computer interaction and computer visualization. Journal of International Scientific Publication: Educational Alternatives, 10:151-158, 2012.

[14] D. Raevskiy. Toward interpretation of pictorial texts. In Proceedings of Conference “Nonverbal Domains of Culture”, pp. 144-145, Moscow, 1995.

[15] C. Morris. Foundations of the theory of signs, volume 1 of International Encyclopedia of Unified Science, University of Chicago Press, Chicago, 1938.

[16] J. Lotman. The structure of the artistic text, volume 7 of Michigan Slavic Contributions, University of Michigan, Department of Slavic Languages and Literatures, Ann Arbor, 1977.

[17] M. Heath, A. Malony, D. Rover. The visual display of parallel performance data. IEEE Computer, 28(11):21-29, 1995.

[18] T. Kuhn. The Structure of Scientific Revolutions. University of Chicago Press, 3rd edition, Chicago, 1996.

[19] T. Kuhn. Metaphor in science. In A. Ortony, editor, Metaphor and Thought, pp. 409-419, Cambridge University Press, Cambridge, England, 1979.

[20] G. Lakoff. The contemporary theory of metaphor. In A. Ortony, editor, Metaphor and Thought, pp. 202-251, Cambridge University Press, Cambridge, England, 1993.

[21] G. Lakoff, M. Johnson. Metaphors We Live By. University of Chicago Press, Chicago, 1980.

[22] P. Barr, J. Noble, R. Biddle. A semiotic model of user-interface metaphor, chapter in "Virtual Distributed and Flexible Organisations". Kluwer Academic Publishers, Dordrecht, 2003. (Edited proceedings of the Sixth International Workshop on Organisations Semiotics in Reading, UK.)

[23] V.L. Averbukh, M. Bakhterev, A. Baydalin, D. Ismagilov, P. Trushenkova. Interface and visualization metaphors. In J. Jacko, editor, Human-Computer Interaction, Part II, HCII 2007, volume 4551 of Lecture Notes on Computer Sciences, pp. 13-22, Springer-Verlag, Berlin Heidelberg, 2007.

[24] V.L. Averbukh, M.O. Bakhterev, A.Y. Baydalin, D.Y. Gorbashevskiy, D.R. Ismagilov, A.Y. Kazantsev, P.V. Nebogatikova, A.V. Popova, P.A. Vasev. Searching and analysis of interface and visualization metaphors. In K. Asai, editor, Human-ComputerInteraction, New Developments, chapter 3, pp. 49-84, In-tech, Vienna, 2008.

[25] V.L. Averbukh. The origins of computer metaphors. survey and analysis. In K. Blashki, editor, Proceedings of IADIS International Conference "Interfaces and Human Computer Interaction 2010" (part of the IADIS Multi Conference on Computer Science and Information Systems 2010—Freiburg, Germany), pp. 391-395, IADIS Press, Lisbon, Portugal, 2010.

[26] J.T. Stasko. Tango: a framework and system for algorithm animation. IEEE Computer, 23(9):27-39, 1990.

[27] C. Knight, M. Munro. Software visualisation conundrums. Computer Science Technical Report, University of Durham, 2001.

[28] J. Pirker, C. Gutl. Virtual worlds for 3D visualizations. Advancing Physics Learning Through Traversing a Multi-Modal Experimentation Space. In D. Preuveneers, editor, Workshop Proceedings of the 11th International Conference on Intelligent Environments, pp. 265-272, Prague, Czech Republic, vol. 19, IOS Press, July 2015.

[29] R.M. Rohrer, E. Swing. Web-based information visualization. IEEE Computer Graphics and Applications, 17(4):52-59, 1997.

[30] V.L. Averbukh, A.U. Baydalin, D.R. Ismagilov, A.U. Kazantsev, S.P. Timoshpolskiy. Utilizing 3d visualization methophors. In Proceedings of 14th International Conference on Computer Graphics and Vision, Graphicon 2004, pp. 295-298, Moscow, Russia, 2004.

[31] M.F. Costabile, D. Malerba, M. Hemmje, A. Paradiso. Metaphors for supporting user interaction with multimedia databases—a demonstration. In Y. Ioannidis, W. Klas, editors, Proceedings of the FLIP TC2/WG 2.6 Fourth Working Conference on Visual Database Systems. Visual Database Systems 4 (VDB4), pp. 154-160, 27-29 May 1998, L'Aquila, Italy. vol. 4, 1998.

[32] G. Balogh. Validation of the city metaphor in software visualization. In O. Gervasi et al., editors, Computational Science and Its Applications—ICCSA 2015, vol. 9159 of Lecture Notes in Computer Science, pp. 73-85, Springer, Cham, 2015.

[33] D. Reed, W. Scullin, L. Tavera, K. Shields, Ch. Elford. Virtual reality and parallel systems performance analysis. FEEE Computer, 28(11):57-67, 1995.

[34] K. Andrews. Case study. Visualising cyberspace: information visualisation in the harmony internet browser. In Proceedings of FEEE Symposium on Information Visualization, pp. 97-104, 1995.

[35] S.I. Fabrikant, D.R. Montello, D.M. Mark. The natural landscape metaphor in information visualization: the role of commonsense geomorphology. Journal of the American Society for Information Science and Technology, 61(2):253-270, 2010.

[36] K.W. Boyack, B.N. Wylie, G.S. Davidson. Domain visualization using VxInsight for science and technology management. Journal of the American Society for Information Science and Technology, 53(9):764-774, 2002.

[37] Th. Panas, R. Berrigan, J. Grund. A 3d metaphor for software production visualization. In Proceedings of Seventh International Conference on Information Visualization, pp. 314-319, 2003.

[38] C. Keskin, V. Vogelmann. Effective visualization of hierarchical graphs with the cityscape metaphor. In NPIV'97, pp. 52-57, Las Vegas, Nevada, USA, 1998.

[39] M. Chalmers. Using a landscape metaphor to represent a corpus of documents. In A.U. Frank and I. Campari, editors, Spatial Information Theory. A Theoretical Basis for GIS, vol. 716 of Lecture Notes in Computer Science, pp. 377-390, Springer, Berlin, 1993.

[40] P. Caserta, O. Zendra, D. Bodenes. 3d hierarchical edge bundles to visualize relations in a software city metaphor. In Proceedings of 2011 6th IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT), pp. 1-8, 2011.

[41] J. Bohnet, J. Dollner. Facilitating exploration of unfamiliar source code by providing 2-1/2D visualizations of dynamic call graphs. In Proceeding of 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis, pp. 63-66, 2007.

[42] R. Wettel, M. Lanza. Visualizing software systems as cities. In 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis, VISSOFT 2007, pp. 92-99, 2007.

[43] M. Balzer, A. Noack, O. Deussen, C. Lewerentz. Software landscapes: visualizing the structure of large software systems. In Proceedings of the Sixth Joint Eurographics—IEEE TCVG Conference on Visualization VISSYM'04, pp. 261-266, 2004.

[44] B.A Malloy, J.F. Power. Using a molecular metaphor to facilitate comprehension of 3d object diagrams. In Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, pp. 233-240, 2005.

[45] M. Gronemann, M. J?nger, N. Kriege, P. Mutzel. The landscape metaphor for visualization of molecular similarities. In: S. Battiato, S. Coquillart, R. Laramee, A. Kerren, J. Braz, editors, Computer Vision, Imaging and Computer Graphics—Theory and Applications, VISIGRAPP 2013, Communications in Computer and Information Science, vol 458, Springer, Berlin, Heidelberg. 2014.

[46] V.L. Averbukh, O. Annenkova, M. Bakhterev, D.D. Manakov. Analysis and assessment of visualization systems used in software for parallel computations. Voprosy Atomnoy Nauki i Tekhniki, series "Mathematical Modeling of Physical Processes", pp. 58-70, 2015.

[47] D. Chang, L. Dooley, J.E. Tuovinen. Gestalt theory in visual screen design—a new look at an old subject. In A. McDougall, J. Murnane, D. Chambers, editors, Selected Papers from the 7th World Conference on Computers in Education (WCCE'01), Copenhagen, Computers in Education 2001: Australian Topics, vol. 8, pp. 5-12.,Australian Computer Society, 2002.

[48] L. Graham. Gestalt theory in interactive media design. Journal of the humanities and Social Sciences, 2(1):1-12, 2008.

[49] R. Fraher, J. Boyd-Brent. Gestalt theory, engagement and interaction. In Proceedings of the 28th of the International Conference Extended Abstracts on Human Factors in Computing Systems, pp. 3211-3216, ACM, 2010.

[50] B.H. McCormick, Th.A. DeFanti, M.D. Brown, editors. Visualization in Scientific Computing, vol. 21. ACM SIGGRAPH Computer Graphics, November 1987. number 6.

[51] Ch. North. Toward measuring visualization insight. IEEE Computer Graphics and Applications, 26(3):20-23, 2006.

[52] A.V. Brushlinskiy. Subject: Thinking, Learning, Imagination. MPSI, Moscow, 2003.

[53] N.I. Chuprikova. The Psychology of Mental Development: The Principle of Differentiation. Stoletie, Moscow, 1997.

[54] P. Vickers, J. Faith, N. Rossiter. Understanding visualization: a formal approach using category theory and semiotics. IEEE Transactions on Visualization and ComputerGraph, 19:1048-1061, 2013.

[55] D.V. Manakov, V.L. Averbukh, P.A. Vasev. Visual text as truth subset of the universal space. Scientific Visualization, 8(4):38-49, 2016.