Bull. Nov. Comp. Center, Comp. Science, 29 (2009), 17–32
© 2009 NCC Publisher

An approach to visualization of knowledge portal content*

Z. V. Apanovich, P. S. Vinokurov, V. A. Elagin

Abstract. The process of development of an ontology-based knowledge portal and creation of its content is time-consuming and labor-intensive. It is very desirable to have special analysis facilities for maintenance and development of such a portal. The subject of our paper is a tool for visual analysis of content and ontology of a knowledge portal.

The basis of our approach is application of graph generation and graph visualization methods. Browsing of the content of a knowledge portal is organized as a multilevel stepwise process. Appropriate placement algorithms are used at each step of this process. They take into account certain types of ontological relations and their combinations. Radial, circular and layered tree placement algorithms are used for visualization of the inheritance relation. Several force-directed algorithms are used for visualization of role relations between different classes as well as between class instances. Finally, we have two special placement algorithms for visualization of the superposition of partonomy and role relations. A number of useful graph transformations are implemented. They essentially improve understandability of a portal under investigation.

This approach provides users with general understanding of ontology and a possibility of visual estimation of the sizes of classes and relations. It allows the user to select and analyze specific relations between classes and between objects as well as to identify errors. The program was tested on real knowledge portals and appeared to be rather helpful.

Key words: *knowledge portal, ontology, content, information visualization, force-directed methods, tree drawing algorithms*

Introduction

Currently, there exist plenty of ontology based knowledge portals. The process of their development is time-consuming and labor-intensive. Large groups of developers can be involved in this process. Big amount of data is entered manually that is potentially dangerous. Besides, knowledge portals need to be maintained, developed and reused. Therefore, the problem of maintenance of long-term functioning and development of such a portal during its life cycle is very important. To make this task easier, a knowledge engineer needs special analysis facilities simplifying comprehension of the portal content. Ontology comprehension is also important for reuse of

*This research is partially supported by RFBR grant 09-07-00400) and RAS project 2/12.

already developed ontologies and knowledge bases. It is much easier to develop already existing ontology than to create another one from the scratch.

A generally recognized way to facilitate understanding of large and complex data sets is to use graph visualization methods [1–3]. Ontology can be represented as a graph, where the graph vertices correspond to the ontology entities, such as classes and instances, and the graph edges correspond to the ontology relationships. By browsing the drawings of various sub-graphs, the portal developer can detect the errors provoked by manual input of information, as well as the design errors.

An essential feature of a knowledge portal content is a considerable quantity of various objects and relations between them. For example, one of our test data contained four thousand objects and fifteen thousand relations between them. But visualization of a graph containing more than a few hundred vertices and edges results in incomprehensible representations with many overlappings and occlusions, which makes almost impossible differentiation and closer investigation of individual vertices and edges. To investigate these large and complex data sets, it is necessary to provide a user with tools for their controlled decomposition. This decomposition should reduce the amount of presented information and simultaneously limit its geometric complexity in order to enable interactivity. In addition, this decomposition should not lose essential information. For example, one of the standard ways of graph browsing is generation of focus vertex neighborhoods with specified radii. It is evident that neighborhoods generation is useless for cycles detection or other kinds of global information extraction.

That is why the basis of our approach is relation-based content decomposition. Users can select one or more relationships to browse classes and instances connected by the selected relationships and to choose methods of their visualization. These methods depend on types of relationships and their combination. A number of graph transformations providing their structured visualization are also implemented. In particular, transitive reduction enables application of tree drawing algorithms instead of force-directed algorithms and provides more simple and comprehensible drawings. Further on, the specific features of the user interface intended to simplify the visual analysis of a knowledge portal content will be described. Also, the examples of errors that can be detected using the proposed visualization methodology will be shown.

1. User interface

Our program gets the input data in the form of two xml-files. The first file (ontology.xml) contains the information on classes and role relationships, and the second one (data.xml) contains the information on instances of classes and their relations. These data are transformed into an internal format. The usage of xml-files is not an essential restriction and we have also experimented with the data presented in RDF-format. The structure of the user interface is subordinated to the main goal: to provide a user with as much facilities of the content analysis as possible. For these purposes, the following decisions have been implemented:

- The user interface consists of a set of menu tabs corresponding to the basic "views" of portal ontology and content;
- A user can choose appropriate visualization methods according to the types of concrete relations, and combinations of relations;
- There is a wide set of facilities for interaction with the content by means of extraction and visualization of various sub-graphs. There also exist editing features allowing users to move and resize these drawings.
- The names of classes, instances, relationships, and values of attributes can be displayed either next to their vertices and edges or in description windows.

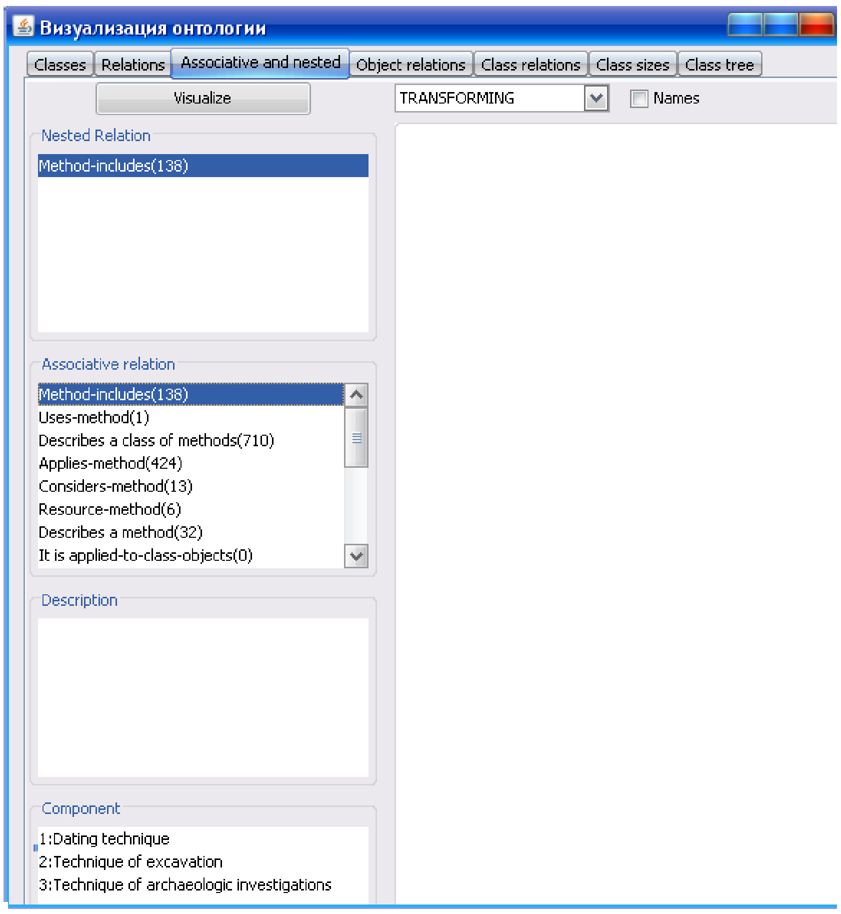
Figure 1 shows the user interface of our program. The top part of the window consists of seven menu tabs, corresponding to the basic modes of visualization: **Classes**, **Relations**, **Associativity and enclosure**, **Relations between objects**, **Relations between classes**, the **Sizes of classes**, and the **Tree of classes**.

- The **Tree of classes** and the **Size of classes** tabs correspond to two different modes of visualization of the inheritance relationship between classes.
- The **Relationships between classes** tab allows users to draw a subgraph induced by any chosen subset of relationships between classes.
- The **"Relationships between objects"** tab makes possible exploration of any chosen subset of relations between objects. The "**Associativity and enclosure"** tab serves for combined visualization of associative relationships and enclosure relationships.
- There exist two auxiliary tabs **"Classes"** and **"Relationships"** that are used for adjustment of visualization parameters.

Further it will be shown how the specified modes help the user to understand the structure of a knowledge portal and its content.

2. Representation of the inheritance relationship and associative relationships between classes

The best way to start an ontology exploration is to examine its global view by generating a tree-like node link diagram of the inheritance relationships



Nested Relation
Method-includes(138)

Associative relation
Method-includes(138)
Uses-method(1)
Describes a class of methods(710)
Applies-method(424)
Considers-method(13)
Resource-method(6)
Describes a method(32)
It is applied-to-class-objects(0)

Component
1:Dating technique
2:Technique of excavation
3:Technique of archaeologic investigations

Figure 1. Visualization panel and "Associative relations and nesting" tab

between classes. To display this view, a user has to choose "**Inheritance Relationships"** tab and to push the **"Visualization"** button. The corresponding sub-graph will be created and displayed in the Graph View Panel with one of the three available tree-drawing algorithms: level-based, radial or circular. Figure 2 shows the radial view of the inheritance relationships in a linguistic ontology. Experiments with various ontologies have shown that empty classes, without instances, occurred very often. Our program highlights these empty classes in gray to remind the user about them and to prompt the user either to delete these useless classes or to fill them with instances.

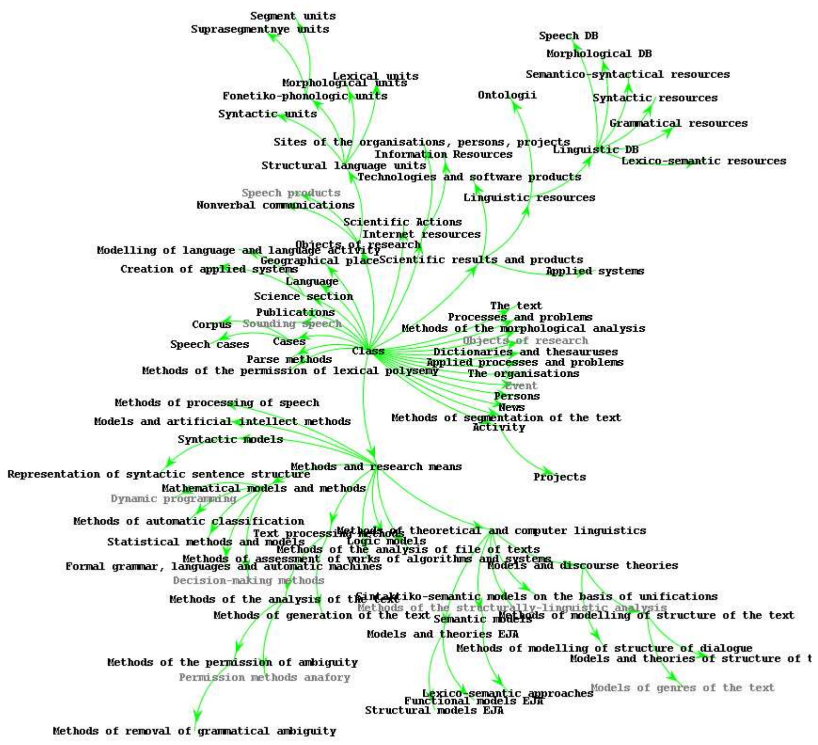

Segment units
Suprasegmentnye units
Speech DB
Morphological DB
Semantico-syntactical resources
Lexical units
Morphological units
Fonetiko-phonologic units
Ontologii
Syntactic resources
Syntactic units
Grammatical resources
Sites of the organisations, persons, projects
Information Resources
Linguistic DB
Lexico-semantic resources
Structural language units
Technologies and software products
Speech products
Nonverbal communications
Linguistic resources
Scientific Actions
Internet resources
Objects of research
Modelling of language and language activity
Geographical place
Scientific results and products
Creation of applied systems
Applied systems
Language
Science section
The text
Publications
Processes and problems
Corpus
Sounding speech
Methods of the morphological analysis
Objects of research
Speech cases
Cases
Class
Dictionaries and thesauruses
Parse methods
Applied processes and problems
Methods of the permission of lexical polysemy
The organisations
Event
Persons
Methods of processing of speech
News
Models and artificial intellect methods
Methods of segmentation of the text
Activity
Syntactic models
Methods and research means
Representation of syntactic sentence structure
Projects
Mathematical models and methods
Dynamic programming
Methods of automatic classification
Text processing methods
Methods of theoretical and computer linguistics
Statistical methods and models
Logic models
Methods of the analysis of file of texts
Methods of assessment of works of algorithms and systems
Formal grammar, languages and automatic machines
Models and discourse theories
Decision-making methods
Methods of the analysis of the text
Sintaktiko-semantic models on the basis of unifications
Methods of the structurally-linguistic analysis
Methods of generation of the text
Semantic models
Methods of modelling of structure of the text
Models and theories EJA
Methods of modelling of structure of dialogue
Methods of the permission of ambiguity
Models and theories of structure of t
Permission methods anafory
Lexico-semantic approaches
Functional models EJA
Models of genres of the text
Structural models EJA
Methods of removal of grammatical ambiguity

Figure 2. The radial view of the inheritance relationships in a linguistic ontology

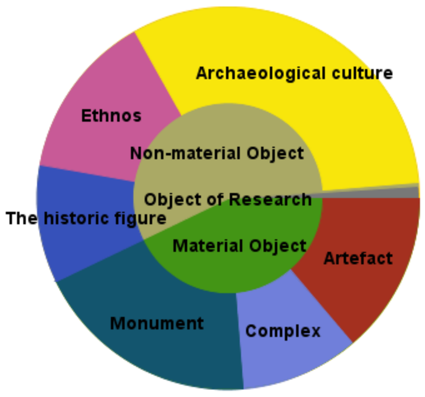


Figure 3. The space-filling radial view of the inheritance relationship

As an additional possibility, a space-filling version of the radial algorithm is available. The menu tab "Size of classes" provides this kind of drawing as it is shown in Figure 3. In this view, each class of ontology is represented with a sector segment, and the angular size of the segment is proportional to the amount of objects in the class. This option is very useful for visual estimation of the quantity of instances in each class. Out test ontologies had rather moderate sizes and the static version of the radial tree drawing algorithm appeared to be a convenient tool for their exploration [6].

3. Representation of combination of the inheritance relationship and associative relationships between classes

An essential feature of the knowledge portals content is a considerable quantity of various kinds of relations between classes. In addition to the inheritance relation, there exists a wide spectrum of associative relationships and plenty of partonomy relationships. We propose to improve understanding of the portal structure by providing the user with special browsing facilities. The user should be able to select and display sub-graphs induced by separate relationships, and to choose a suitable way of visualization for certain types of relations and combinations of relations. Our experiments have shown that a combined representation of the inheritance relation with associative relations between classes is very helpful for understanding of a portal structure. Two ways of visualization for such combinations were implemented in our program.

The first way of visualization is available in the mode "Inheritance of classes". When a node in the drawing is selected by a mouse click, associative relationships of this node appear (Figure 4). This kind of visualization appeared to be rather useful because it allows the user to compare associative relations of the nodes connected by the inheritance relation. In particular, it is not always easy to distinguish classes connected to subclasses via some inheritance relation, and the objects of the same class connected by a partonomy relation. We suppose that a child class should be distinctly different from the parent one. This difference can be expressed in the form of existence of additional relationships with other classes, a distinctive property that the child concept bears, or in limitations on slot filters that are distinct from limitations on other classes. To make the drawing clearer, our program does not show the relationships of the child classes that are inherited from the parent class. All the subclasses that have not their own relationships or attributes are visible at the first glance.

Let us consider, for example, the class **Person** that has the classes S**tudent** and **Researcher** as subclasses. The class **Person** is connected with other classes via the following associative relationships: "Acquaintances" (the class "**Person**"), "Event-Participants" (the class "**Event**"),

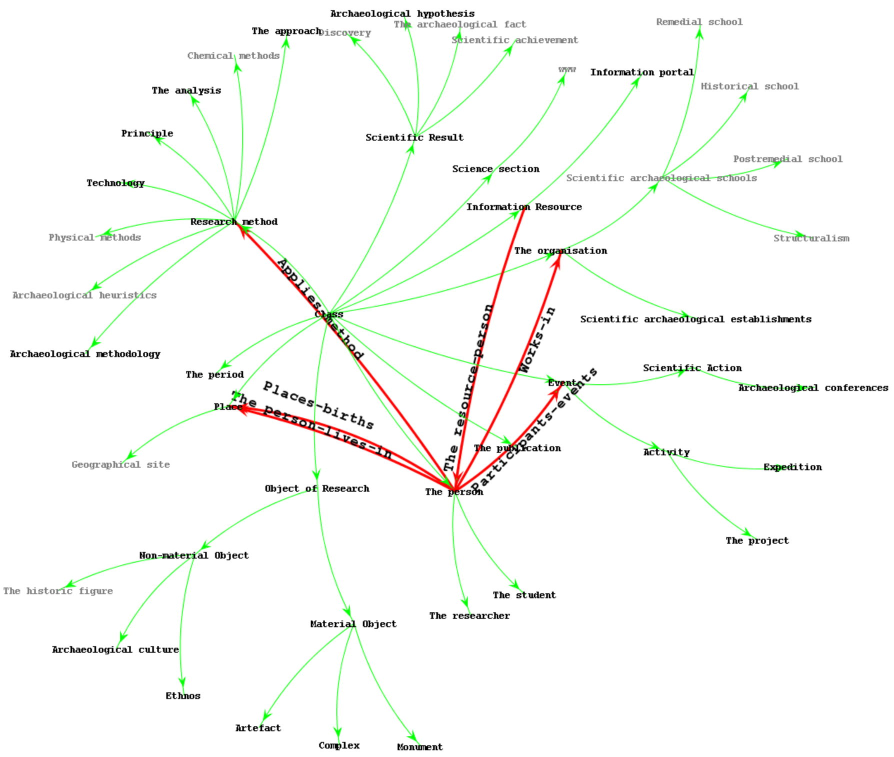


Figure 4. The inheritance relationship combined with associative relationships of the class **Person**

”Resource-of-a-Person” (the class “**Information Resource**”), ”Applies-Method” (the class “**Investigation Method**”), and “Student” (the class “**Researcher**”).

The subclass **Researcher** inherits relationships from the class **Person** and, in addition, it has its own relationships. It is connected with:

- the class “**Scientific Division**” via the “Investigation-Direction” relationship,
- the class **Publication** via the “Author-Of” relationship,
- the class **Period** via the “Investigates-Period” relationship,
- the class **Project** via the “Project-Participant” relationship,
- the class **Person** via the “Student-Of” relationship.

At the same time, the subclass **Student** of the class the **Person** has no its own relationships with other classes. This feature is easily discovered when the node **Student** is selected by a mouse click. When detecting this problem, a knowledge engineer has a choice. He can expand the **Student**

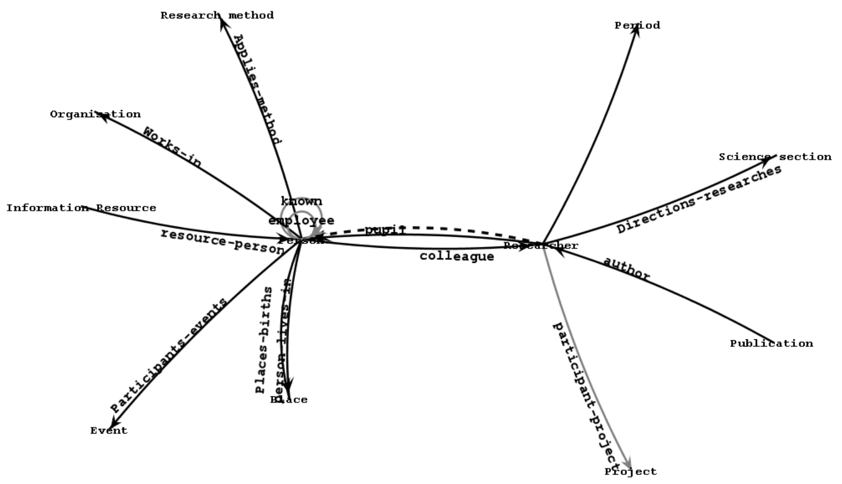


Figure 5. Relationships between the classes **Person** and **Researcher**

subclass definition with relations distinguishing it from the classes **Person** and **Researcher**, or to remove this subclass from the ontology. Similarly, it is possible to check all the subclasses of a given ontology. When the number of classes is moderate, such inspection takes just several minutes.

When checking-out the information content of a knowledge portal, it is also very important to have a global view of associative relations in the ontology. A facility allowing a user to choose arbitrary subsets of nodes or relationships and to get visualizations of sub-graphs induced by the chosen nodes and relationships can be very helpful. It can serve as a prompt for the ontology developer. For example, links between the classes **Person** and **Researcher** are shown in Figure 5. The dashed edge represents the inheritance relationship. When looking through the associative relations of the classes **Person** and **Researcher, a** knowledge engineer can be surprised by discovering that a relationship "Applies-the-Method" connects the class **Method-of-Investigation** with the class **Person** and not with the class **Researcher**.

One more additional facility provides us with a drawing where the size of nodes is proportional to the quantity of objects in the corresponding classes and the thickness of edges is proportional to the quantity of links between objects of these classes (Figure 6). This kind of drawing allows one to estimate in advance the volume of picture showing relations between instances of classes.

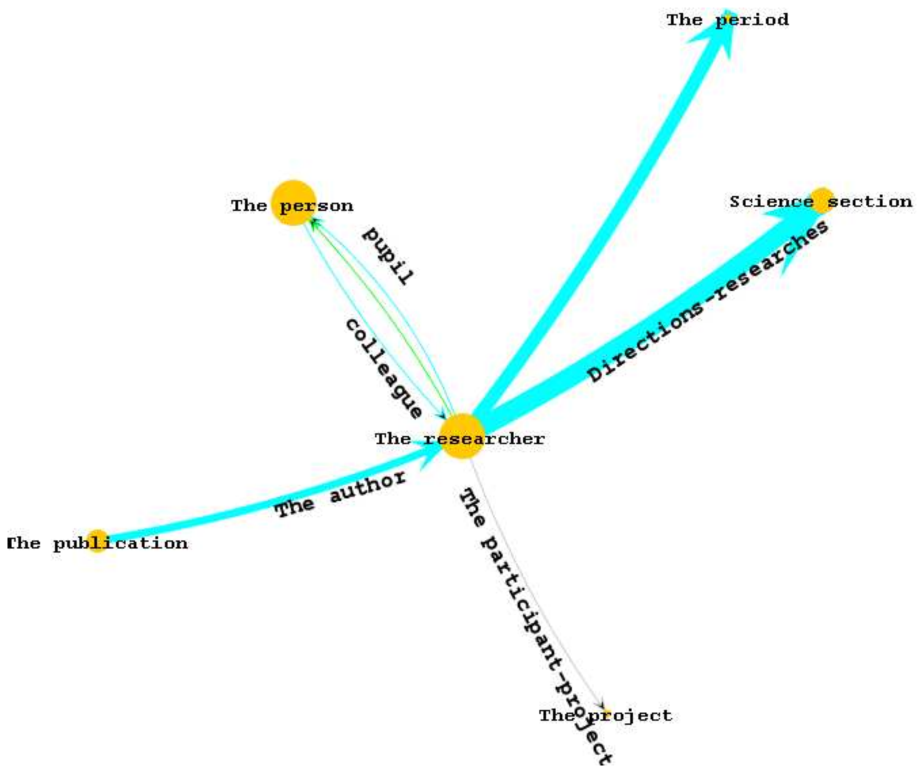
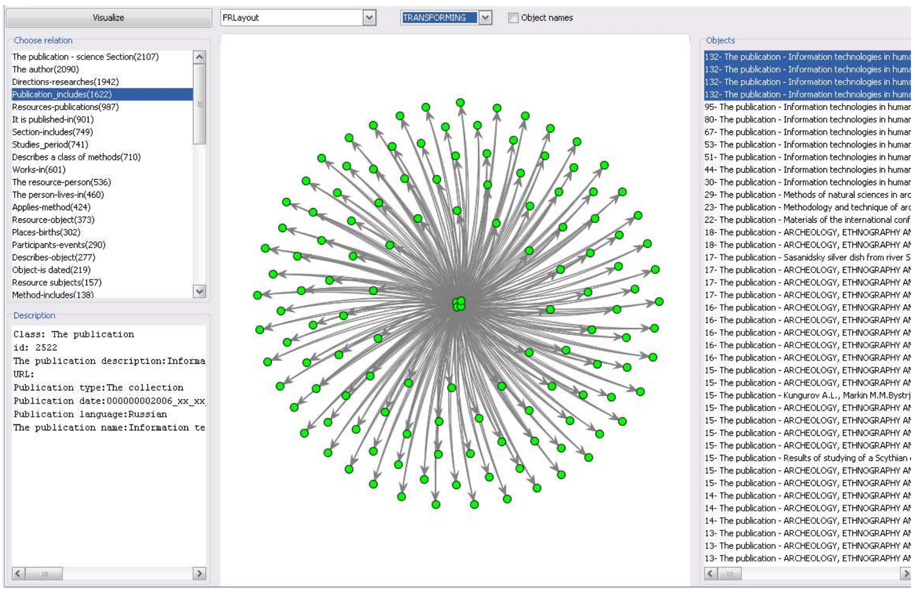


Figure 6. The size of nodes is proportional to the quantity of objects in corresponding classes and the thickness of edges is proportional to the number of links between objects of these classes

4. The drawing of relations between instances of classes is a useful tool of analysis of a knowledge portal content

The "**Relations between instances**" tab is designated for visualization of the class instances and subsets of relations between them. To get a drawing of a sub-graph induced by the chosen instances and relations, a user can select vertices and relations of interest in the drawing of classes (if it exists). It is also possible to get the same drawing directly by choosing necessary relations from the list of available relations in the "Choice of relationship(s)" Panel (Figure 7). When a subset of relations is selected in the Panel "Choice of relationship(s)" on the left, the list of instances connected by the chosen subset of relations appears in the Instance Panel on the right. A user can select an arbitrary subset of these instances and get their drawing. Our program searches for the connected components in the graph and displays them with one of the available placement algorithms. Currently, Kamada–Kawai [9] and Fruchterman–Reingold [7] algorithms are used for visualization of the available data. Visualization of instances and



```
The publication - science Section(2107)
The author(2090)
Directions-researches(1942)
Publication_includes(1622)
Resources-publications(987)
It is published-in(901)
Section-includes(749)
Studies_period(741)
Describes a class of methods(710)
Works-in(601)
The resource-person(536)
The person-lives-in(460)
Applies-method(424)
Resource-object(373)
Places-births(302)
Participants-events(290)
Describes-object(277)
Object-is dated(219)
Resource subjects(157)
Method-includes(138)
```

```
Class: The publication
id: 2522
The publication description:Informa
URL:
Publication type:The collection
Publication date:000000002006_xx_xx
Publication language:Russian
The publication name:Information te
```

```
132- The publication - Information technologies in huma
132- The publication - Information technologies in huma
132- The publication - Information technologies in huma
132- The publication - Information technologies in huma
95- The publication - Information technologies in humar
80- The publication - Information technologies in humar
67- The publication - Information technologies in humar
53- The publication - Information technologies in humar
51- The publication - Information technologies in humar
44- The publication - Information technologies in humar
30- The publication - Information technologies in humar
29- The publication - Methods of natural sciences in arc
23- The publication - Methodology and technique of arc
22- The publication - Materials of the international conf
18- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
18- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
17- The publication - Sasanidsky silver dish from river S
17- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
17- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
17- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
16- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
16- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
16- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
16- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
16- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
15- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
15- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
15- The publication - Kungurov A.L., Markin M.M.Bystrj
15- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
15- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
15- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
15- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
15- The publication - Results of studying of a Scythian
15- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
15- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
14- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
14- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
14- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
13- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
13- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
13- The publication - ARCHEOLOGY, ETHNOGRAPHY AN
```

Figure 7. A sub-graph induced by instances of the class **Publication** connected by the relation “Publication-includes”

links between them has found to be a very helpful analysis tool. It makes possible quick identification of errors, arising during manual input of a portal content or at some stage of ontology design.

The presence of cycles or loops (the edges connecting a vertex with itself) in a graph of instances, as well as graph disconnectedness, can be a sign of erroneous data input. For example, Figure 8 shows a cycle in the sub-graph induced by instances of the class **Period**. This cycle is created by the relation “Historically-follows” connecting three objects “Lower Paleolithic”, “Middle Paleolithic” and “Upper Paleolithic”.

The time borders defined for these periods are as follows:

Period Name	Lower Paleolithic
Start of the Period	1500000 BC
End of the Period	100000 BC
Period Name	Middle Paleolithic
Start of the Period	100000 BC
End of the Period	40000 BC
Period Name	Upper Paleolithic
Start of the Period	40000 BC
End of the Period	14000 BC

According to the time borders defined for the specified periods, they should be ordered linearly: “the Lower Paleolithic”–> “the Middle Paleolithic”–> “the Upper Paleolithic”, and there should be no cycles in the

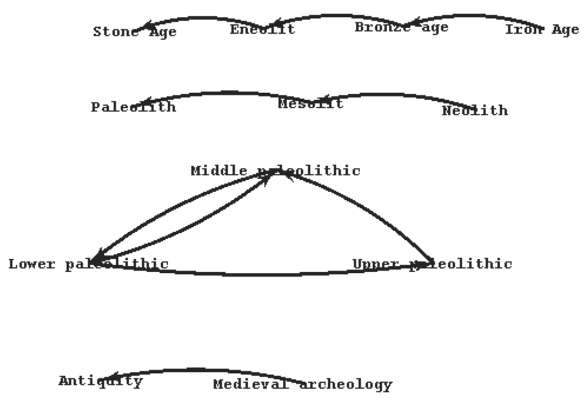


Figure 8. The presence of cycles in the sub-graph induced by instances of the class **Period**, connected by the relation "Historically-follows"

graph. This is a typical error of manual data input. Moreover, the sub-graph induced by the relationship "Historically-follows" is not connected that is not natural at all. Thus, the generated drawing helps the ontology developer and prompts him to add constraints on the sub-graph induced by the relationship "Historically-follows" to make it acyclic and connected. This can be done by incorporation of the relevant axioms in the ontology.

One more example demonstrates utility of drawing of sub-graphs selected by relationships for detection of errors arising during the ontology design. Figure 7 shows a drawing of a sub-graph induced by the instances of the class **Publication**, connected by the relation "Publication-includes". It is possible to see four vertices situated in the center of the drawing. They are too close to one another. Since the depicted relationship is "Publication-includes", this drawing means that the corresponding four publications include all the other publications shown in the figure. A user can select any of these four vertices with mouse and see its description in the bottom left component of the Visualization Panel. The descriptions of the four vertices are very similar:

- Description of the publication: Information technology in humanitarian research: Digest. Issue 9. Novosibirsk: Novosibirsk State University, 2005. 90 P.
- Type of the publication: digest.

Other three vertices have precisely the same descriptions and the only difference is the number of the issue. That is, the specified vertices represent four issues of the digest "Information technology in humanitarian research", for the years 2004, 2005, 2006 and 2007.

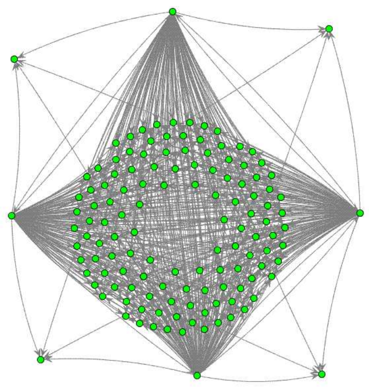


Figure 9. A sub-graph induced by instances of the class **Publication** connected by the relation "Publication-includes"

At this moment, there is a question: whether there was an error in the description of links of other publications (articles) to the specified four digests? It is possible to see that every visible article is included in each of four digests. But we can refine the drawing by moving vertices. Using a mouse, we place four vertices-issues in the top left, top right, bottom left, and bottom right corners of the drawing as shown in Figure 9. Now it is possible to see that the four issues-vertices include each other. This is an evident pointer to a design error. It seems that in the ontology under investigation there is no general concept that unites issues of the digests corresponding to the years of issue.

At the moment, the only way to detect similar errors is to explore all available sub-graphs one by one, which is a time- and labour-consuming problem. Therefore, it is desirable to analyze in advance semantics of essential relations of ontology and to automatically check correctness of the input data. Our tools are the first step on the way to this automatic verification.

One more feature of sub-graphs corresponding to relationships between instances of classes is that many relations have transitivity property. The sub-graphs corresponding to these relations have a significant amount of transitive edges. These edges overcharge the sub-graph drawing and create useless clutter. After deleting the transitive edges, it is possible to re-layout the graph vertices by one of the available tree drawing algorithms and to get much more comprehensible drawing. Figure 10 shows a drawing of the relationship "Method-includes" between instances of a class "Method-of-Investigation" before and after transitive edges deletion.

We have noticed also that during the manual input of data many transitive edges are usually lost just because a user forgot to input them. Moreover, a layout without transitive edges looks more natural and comprehensible from the human point of view. But on the other hand, all transitive edges should be used when answering the knowledge base query. Therefore, an optimum operating mode with transitive relations is the mode when the data entered manually are processed by the algorithm of a transitive reduction. The result of transitive reduction is stored and laid out. But for a knowledge base query processing, a transitive closure of the corresponding sub-graph is used.

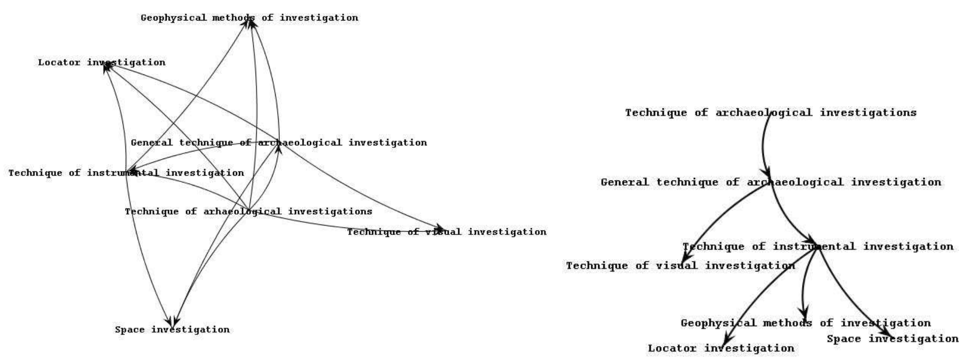

Geophysical methods of investigation
Locator investigation
General technique of archaeological investigation
Technique of instrumental investigation
Technique of arhaeological investigations
Technique of visual investigation
Space investigation
Technique of archaeological investigations
General technique of archaeological investigation
Technique of instrumental investigation
Technique of visual investigation
Geophysical methods of investigation
Space investigation
Locator investigation

Figure 10. A drawing of the relationship “Method-includes” between instances of the class “Method-of-Investigation” before and after transitive edges deletion

5. Combined representation of a partonomy relation and associative relations

One more feature of a knowledge portal structure is a considerable quantity of associative relations connecting instances that are a part of the hierarchical structure induced by the partonomy relationship. Let us consider, for example, a big hierarchy of instances of the class “**Methods of research**” connected by the relation “Method-includes”. Suppose that we need to know the persons applying these methods. A layout of the corresponding sub-graph generated by one of the force-directed placement algorithms looks rather confused and incomprehensible (see Figure 11).

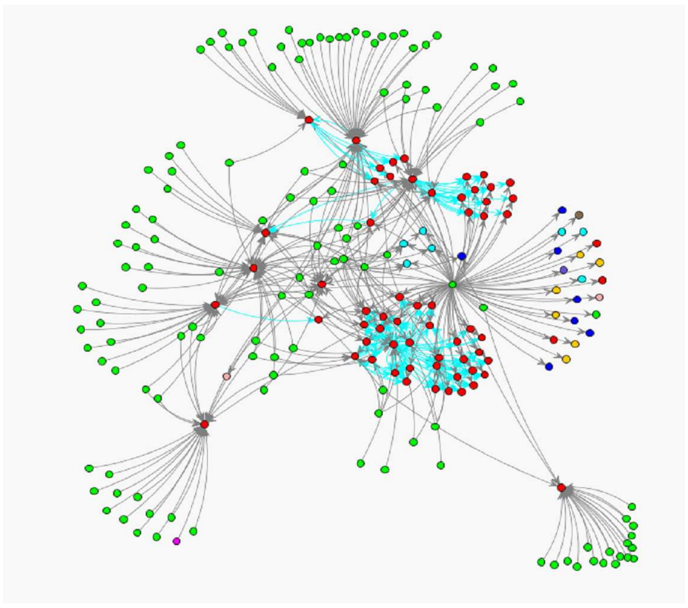


Figure 11. A combined representation of the partonomy relationship "Method-includes" and relationship "Applies-the-Method-of-Investigation"

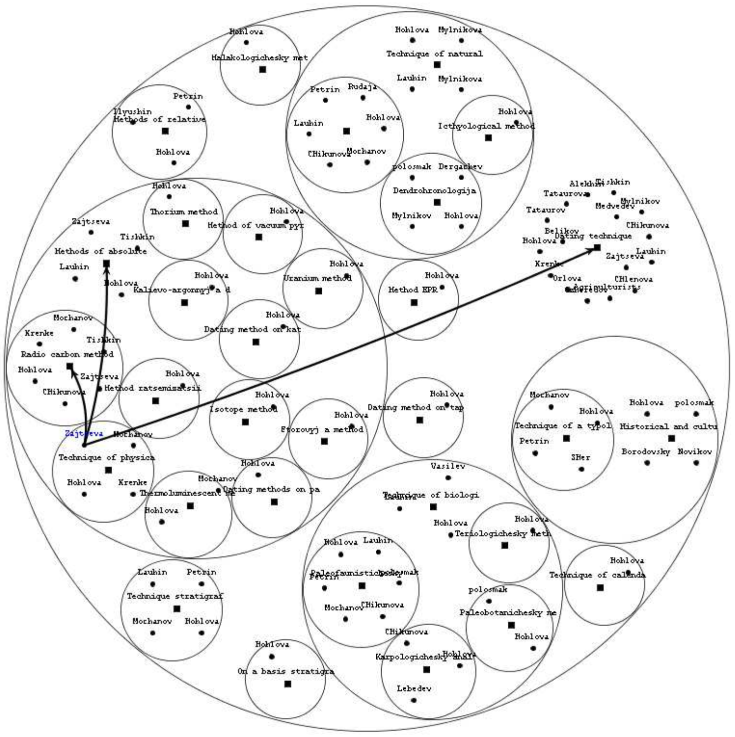


Figure 12. A combined representation of the partonomy relationship "Method-includes" and relationship "Applies-the-Method-of-Investigation"

We have implemented a specialized method of visualization for this kind of combination. It represents the members of the partonomy structure by circles and the partonomy relationship between them by means of a geometrical nesting of vertices connected by the partonomy relation. A heuristic algorithm is used for their placement. Every circle contains all the instances connected with a node of the partonomy structure by an associative relationship. These vertices are placed by the Kamada–Kawai [9] algorithm inside the nodes of the partonomy structure. Figure 12 shows the nesting structure for the relationship "Method-of-research-includes". The central circle represents an instance "Methodology of excavations". The methods contained in the methodology of excavations are represented with the nested circles. Every node-circle contains vertices connected to nodes of the partonomy structure by an associative relationship. Figure 12 shows such associative relationship as "Applies-the-method". Note that the partonomy structure can contain several copies of instance connected to the partonomy by the associative relationship. When such an instance is selected, the program shows all the nodes of the partonomy structure connected with the chosen instance. Figure 12 shows the edges that correspond to methods of investigation applied by a person whose name is Mochanov. From our point of view, this layout is much more structured and comprehensible than that of Figure 11.

Conclusion

A subsystem for visual analysis of the knowledge portal content is described. This subsystem has been tested on real data and proved to be rather useful. It makes possible detection of errors of manual data input that can hardly be found in text representation of data or by means of standard methods of navigation in large graphs. This subsystem can be useful at the stage of knowledge portal development, as well as during its life cycle.

Experiments with visualization of the knowledge portal content have shown the necessity of further development of the visualization program in several directions. First, for effective visualization of the classes containing a considerable quantity of instances and relations, the existing set of placement algorithms needs to be extended with clustering methods and cluster placement. Second, it is necessary to investigate the types of the existing relationships and to define the most effective methods of visualization both for certain types of relations and for various types of sub-graphs generated during the analysis of the knowledge portal content. Third, the development of specialized methods of analysis and verification of this content is necessary for effective work with knowledge portals. When developing our subsystem, we used a free library of classes Jung [8].

Acknowledgements. The authors are very grateful to Ju.A. Zagorulko and S.V. Bulgakov for the test data in the xml-format.

References

[1] **Apanovich Z.V.** Methods of navigation for graph visualization // Vestnik NGU. – Vol. 6, Iss. 3. – 2008. – P. 35–47 (In Russian).

[2] **Apanovich Z.V.** Methods of interactive information visualization problems of control and modeling in complex systems // Proc. of X Internat. Conf. (Samara, July 23–25 2008). – 2008. – P. 478–489 (In Russian).

[3] **Apanovich Z.V.** Tools for large graphs processing: construction and optimization of floorplans // System informatics. Iss. 10: Methods and models of modern programming. – Novosibirsk: SB RAS Publishing House, 2006. – P. 7–58 (In Russian).

[4] **Zagorulko Yu.A., Borovikova O.I., Kholushkin Yu.P.** Construction of ontology for archeological knowledge portal // Information Technologies in Humanitarian Research. Iss. 10. — Novosibirsk, 2006 (In Russian).

[5] **Kholushkin Yu.P., Grazhdannikov E.D.** System Classification of Archeological Science (Elementary Introduction in Science of Science). – Novosibirsk, 2000. – 58 p. (In Russian).

[6] **Di Battista G., Eades P., Tamassia R., Tollis I.G.** Algorithms for drawing graphs: an annotated bibliography // Computational Geometry, Theory and Applications. – 1994. – N 4. – P. 235–282.

[7] **Fruchterman T. M. J., Reingold E. M.** Graph drawing by force-directed placement software // Practice and Experience. – 1991. – Vol. 21, N 11. – P. 1129–1164.

[8] **Madahain O., Fisher D., Smyth P., et al.** Analysis and visualization of network data using jung // J. of Statistical Software. – VV(II).

[9] **Kamada T., Kawai S.** An algorithm for drawing general undirected graphs // Information Processing Letters. – 1989. –Vol. 31. – P. 7–15.

[10] **Katifori A., Halatsis C., Lepouras G., et al.** Ontology visualization methods – a survey // ACM Comput. Surv. – 2007. – Vol. 39, N 4. – P. 10.