SCIENTIFIC PAPERS, UNIVERSITY OF LATVIA, 2009. Vol. 751
COMPUTER SCIENCE AND INFORMATION TECHNOLOGIES 22–39 P.

# Formal Concept Analysis for Concept Collecting and Their Analysis*

**Darius Jurkevicius**[1] and **Olegas Vasilecas**[2, 3]
[1] Department of Information Systems, Faculty of Fundamental Sciences, Vilnius Gediminas Technical University, Sauletekio al. 11, LT-10223 Vilnius-40, Lithuania, *d.jurkevicius@isl.vgtu.lt*
[2] Information Systems Research Laboratory, Faculty of Fundamental Sciences, Vilnius Gediminas Technical University, Sauletekio al. 11, LT-10223 Vilnius-40, Lithuania, *olegas@isl.vgtu.lt*
[3] Department of Computer Science, Faculty of Natural Sciences, Klaipeda University, Herkaus Manto 84, LT-92294 Klaipeda, *olegas.vasilecas@ik.ku.lt*

The method how to collect concepts and analyse them using formal concepts analysis is presented in this paper. This method allows representing of the hierarchical tree of concepts without analyzing terms in a specific domain. The main idea of the proposed method is to select concepts (objects) from collected data of specific domain using templates. Later the collected terms are analyzed using the formal concept analysis method. This allows to simplify the steps of term analysis and ontology representing in the ontology development process. We propose to use the logical structure of context that allows to extend the traditional context. It allows to save more data in the context and to simplify the usage of context in information systems. We call this extended formal context distributional formal context. In this paper, the real estate domain was selected for the experiment and its results are presented.

**Keywords:** formal concepts analysis, formal context, formal concept, multi-formal context, distributional formal context, concept collecting.

## Introduction

Usually, different types of ontology are used for designing of information systems (Fig. 1). During ontology development process the general concepts (entity, event, date, process, etc) are defined at top level. At the next step, concepts of domain (domain ontology), concepts of process (ontology of process), and concepts of tasks (ontology of problem) are defined. In that way principle of independence of the domain knowledge and knowledge about process proposed by Guarino (5) are implemented. Processes are described by the terms of the actor. The processes in relation with the problem are

* The work is supported by Lithuanian State Science and Studies Foundation according to High Technology Development Program Project "Business Rules Solutions for Information Systems Development (VeTIS)", Reg. No. B-07042.

linked to the entity of domain. It is useful to find components that can be repeatedly used for designing information systems. Ontology is one of such components. Ontology of process can be repeatedly used, because its structures are the same in different domain processes. Generally, the ontology development process includes 4 stages: collecting of terms, analysis of terms, correction of terms, and representation of terms. The diagram of this process is shown in Figure 2.

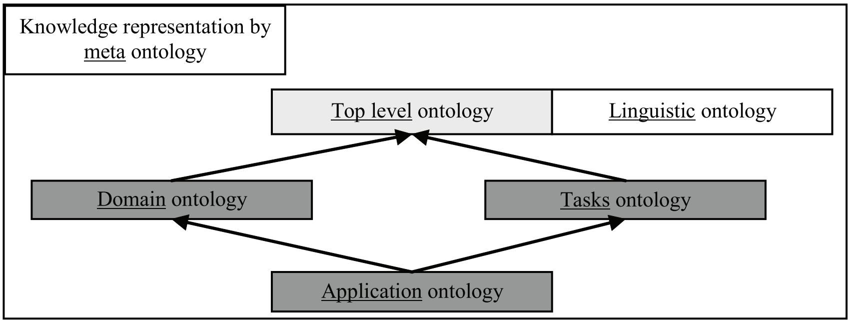


*Fig. 1.* Types of ontology

The identification of all terms of a specific domain, their relationship and definitions are included into the term collection stage. Terms are analysed at the stage of term analysis. During this stage different terms that describe the same objects and processes are searched for. After the searching of terms is finished, the usage of one general term must be achieved. This is done during the third stage of ontology development. The fourth stage is representation of ontology using a specific language (for example, OWL), and during this stage a tool is used. The choice of ontology representation language belongs to the ontology development tool. A user without a specific knowledge can develop ontology just by using ontology development tool. For example, an information system engineer can perform this work.

The ontology development process described above and presented at Fig. 2 is slow and time-consuming.

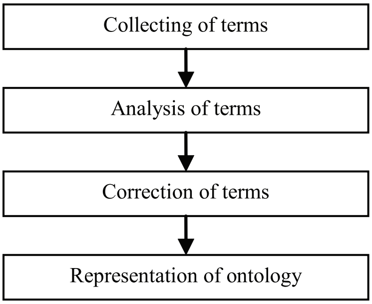


*Fig. 2.* Process of ontology development

Automating of the stages of term analysis and ontology representation is one of the ways to develop ontology more quickly. There are no systems that could automate these stages at this moment. However, researchers' community in the ontology development field believes that in achieving the technology of semantic Internet, specific systems will be developed that should automate ontology development stages. This paper proposes the method for automation of ontology developed process by mean of formal concept analysis.

The authors apologize for the readers because some pictures of application are in the Lithuanian language. The application and ontology for it was used for analysing data in Lithuanian. The remaining part of this paper consists of the following sections.

- Formal concept analysis and its usage for ontology developed is described in the second section.
- Collecting of terms in a specific domain and the context logical structure that allows to extend the traditional context is described in the third section.
- The experiment and its results are described in the fourth section. The system architecture is presented.
- The conclusions are drawn in the last section.

## What is Ontology?

According to [1, 14, 15], in the context of computer and information sciences, ontology is defined as a set of representational primitives with which a domain of knowledge or discourse is modeled. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application.

By the feature of expressivity, two types of ontology can be distinguished: heavyweight and lightweight ontology. The main difference between them is the role played by axiomatization. Heavyweight ontologies are extensively axiomatized and lightweight ontologies often are presented as simple taxonomic structures and are either slightly or not axiomatized.

Contemporary ontologies share many structural similarities, regardless of the language in which they are expressed. Most ontologies describe individuals (instances), classes (concepts), attributes, and relations. In this section, each of these components is discussed in turn.

Common components of ontologies can include:

- individuals: instances or objects (the basic or "ground level" objects);
- classes: sets, collections, concepts, types of objects, or kinds of things;
- attributes: aspects, properties, features, characteristics, or parameters that objects (and classes) can have;
- relations: ways in which classes and individuals can be related to each other;
- function terms: complex structures formed from certain relations that can be used in place of an individual term in a statement;
- restrictions: formally stated descriptions of what must be true in order for some assertion to be accepted as an input;

- rules: statements in the form of an “if-then” (antecedent-consequent) sentence that describe the logical inferences that can be drawn from an assertion in a particular form;
- axioms: assertions (including rules) in a logical form that together comprise the overall theory that the ontology describes in its domain of application. This definition differs from that of axioms in generative grammar and formal logic. In these disciplines, axioms include only statements asserted as *a priori* knowledge. As used here, axioms also include the theory derived from axiomatic statements.

## Understanding of Formal Concept Analysis

One of the ways to transform available data in a hierarchic form is formal concept analysis. Dau [2] noticed that scientists making plots could not lean on them as on arguments. To separate formally the mathematical structure from its schematical presentation, the work environment was created in which diagrams could be used to make formal substantiations. Now we will define some terms used in this paper.

**Concept** can be defined as:

- an abstract or general idea inferred or derived from specific instances [9, 23];
- an abstract idea or a mental symbol typically associated with a corresponding representation in language or symbology that denotes all of the objects in a given category or class of entities, interactions, phenomena, or relationships between them [15, 23];
- having an intention (deep definition), extension (set of objects or exemplars) [10];
- the definition of a type of objects or events; a concept has an intentional definition (a generalization that states membership criteria), and an extension (the set of its instances) [11].

**Formal Concept Analysis** (FCA) [17] method is:

- a mathematization of the philosophical understanding of concept;
- a human-centred method to structure and analyze data;
- a method to visualize data and its inherent structures, implications, and dependencies.

FCA is based on the philosophical understanding that a concept can be described by its extension – that is, all the objects that belong to the concept and its intension which is all the attributes that the objects have in common [13], and this can be represented as a table.

**Formal context** is the mathematical structure used to formally describe these tables of crosses (or, briefly, a context) [22].

FCA is a method used in data analysis, knowledge representation, and information control. Rudalf Wille suggested FCA in 1981 [17], and this method is successfully developed nowadays. For the first 10 years, FCA was researched by small groups of scientists and Rudalf Wille’s students in Germany. FCA was not known worldwide because the bulk of publications were presented at mathematicians' conferences. After getting the sponsorship, some projects were implemented in this area. Most of them were knowledge research projects used for systems delopment. This system was known only in Germany. During the last 10 years, FCA became the research object of the international scientific community. FCA was used in linguistics, psychology, as well as in software engineering and in the areas of artificial intelligence and information search.

Some of the structures of FCA appear to be fundamental to information representation and were independently discovered by different researchers. For example, Godin et al. [5] used concept lattices (which they call "Galois lattices") for information retrieval.

Now, we shall introduce the definition of formal concept analysis [4]. Let us present an example: G is the set of objects that we are able to identify in some domain (e.g. if, when, than). Let M be the set of attributes. We identify the index I as a binary relationship between the two sets, G and M, i.e. $I \subseteq G \times M$. The triple (G, M, I) is called a formal context. For $A \subseteq G$, we define

$$A' := \left\{ m \in M \;\middle|\; (g,m) \in I \;\; for \;\; all \;\; g \in A \right\} \qquad (1),$$

and dually, for $B \subseteq M$

$$B' := \left\{ g \in G \;\middle|\; (g,m) \in I \;\; for \;\; all \;\; m \in B \right\} \qquad (2).$$

A formal concept of a formal context (G, M, I) is defined as a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' \subseteq B$ and $B' \subseteq A$. Sets A and B are called the extend and intend of the formal concept. The set of all formal concepts of a context (G, M, I) is called the concept lattice of the context (G, M, I).

## The Proposed Method for Concept Collecting and Their Analysis

Our method for term collection and analysis of data is proposed in this section. The sequence of term collection, analysis, and representation is shown in Figure 3. Hereinafter we will describe the process of our proposed method. Firstly, analysts (or users) select the specific domain. The second step is acquisition of information for analyzing the domain. Then the user defines the attributes. Attributes are needed for a specific tasks (for example, criteria attributes are needed for task search). Next, the user inputs all attributes into formal context and the information is analysed (Fig. 5). The user searches for the objects (concepts) in information and, when the object (concept) is found, it is inputted into the formal context and the next step is information processing.

When all objects are inputted into formal context and information processing is executed, the analyst can create the concept net (Fig. 4). Using the created concept net, the analyst can search the dependences, matching, repeated structures, exceptions, etc. The analyst can also develop ontology using the concept net (Fig. 4).

Information system can use the created formal context to solve a specific task.

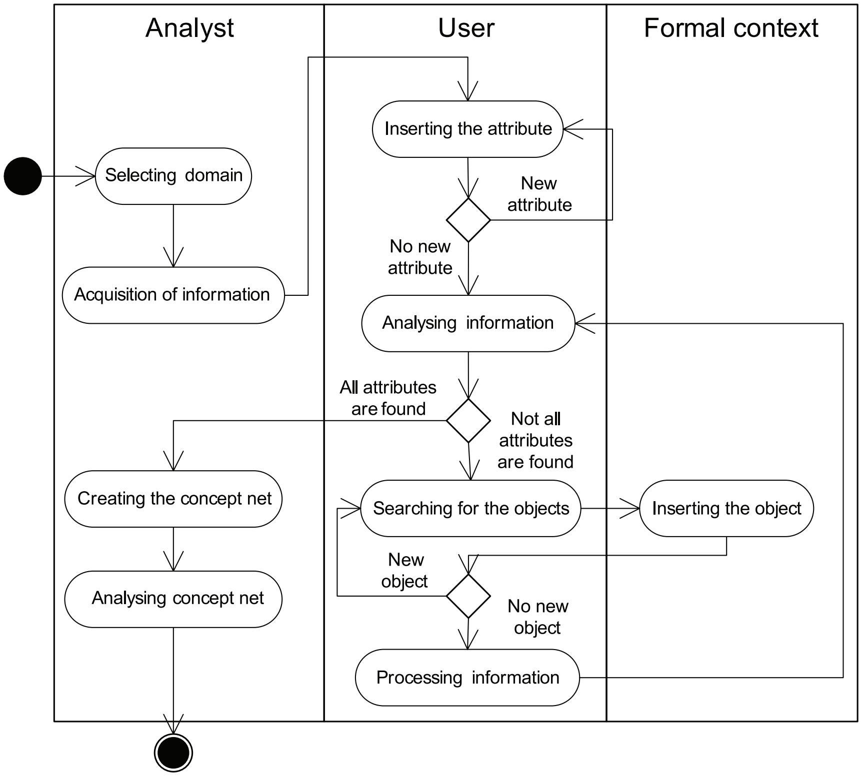


*Fig. 3.* The process of term collection, analysis, and representation

It is well known that knowledge is important part of modern information systems. Knowledge used in information systems is stored in knowledge bases, data bases, ontology, and other knowledge sources. Formal context is one of knowledge sources.

The traditional formal context is used in formal concept analysis. The traditional context is a triple (G, M, I) consisting of a set of formal objects G, a set of formal attributes M, and a binary relation $I \subseteq G \times M$ (expressing the attributes pertaining to each object) [3, 18, 19, 20, 21]. The Formal Concept Analysis method is: a mathematization of the philosophical understanding of concept; a human-centred method to structure and analyze data; a method to visualize data and its inherent structures, implications, and dependencies. However, this method is quite difficult in using it for information systems because there is not enough data stored in the context.

In comparison with formal context structure, we review the ontology structure because these structures are quite similar. They have objects (classes), attributes, and relations between objects and attributes. Ontology in computer science and information science is a formal representation of a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to define the domain. Most ontologies [7, 12, 16] describe individuals (instances), classes (concepts), attributes, and relations. In this section, each of these components is discussed in turn.

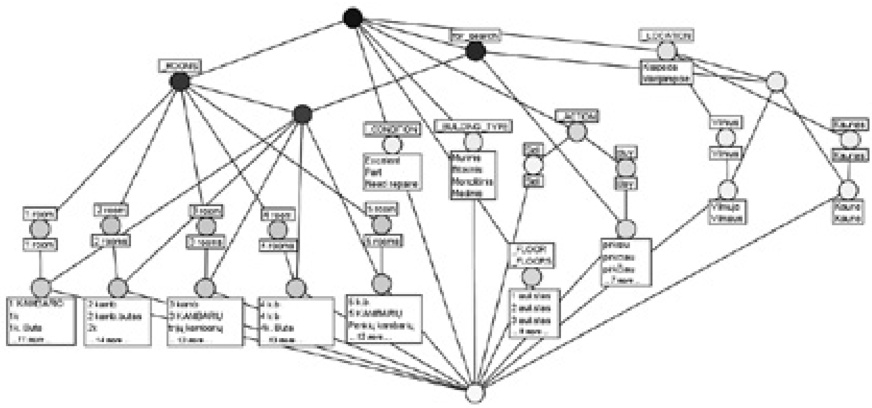

*Fig. 4.* The net of concepts created with ToscanaJ-1.6 tool from formal context

Ontology structure is more complex than formal context structure.

In the next section, we propose the context logical structure that allows to extend the traditional context. That solution permits saving more data in the context and simplifying the usage of context in information systems.


*Fig. 5.* The form for information analysis

## Distributional Formal Context

A *traditional formal context* in formal concept analysis is a triple (G, M, I) consisting of a set of *formal objects* G, a set of *formal attributes* M, and a binary relation $I \subseteq G \times M$ (expressing the attributes pertaining to each object) [3, 18, 19, 20, 21]. The traditional logical data model and graphical representation of formal context [4, 22] in formal concept analysis is shown in Figure 1.

For more information storage in formal context, we propose to extend the logical scheme of formal context. That allows keeping general features of formal concept analysis. By [4, 19], for the mathematical definition of *formal concepts*, we introduce the derivation operators “ ' ”.

Using the derivation operators, we can derive *formal concepts* from our *traditional formal context* with the following routine:

1) pick a set of objects A;
2) derive the attributes A';
3) derive (A')';
4) (A'',A') is a *formal concept.*

An example of generating formal concept from traditional formal context (Fig. 6):

1) pick any set of objects A, e.g. A = {Object 3};
2) derive the attributes A' = {Attribute 3, Attribute 4, Attribute n};
3) derive (A')' = {Attribute 3, Attribute 4, Attribute n }' = { Object 3, Object 4};
4) (A'',A') = ({Object 3, Object 4},{Attribute 3, Attribute 4, Attribute n}) is a *formal concept*.

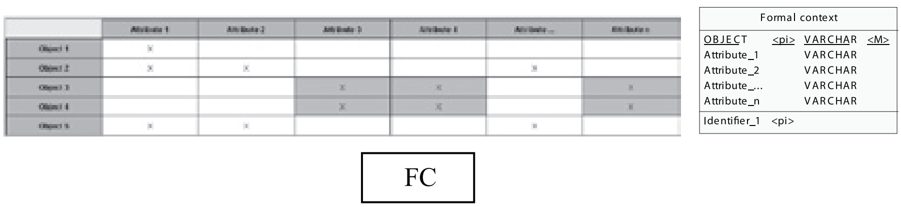


*Fig. 6.* Traditional formal context (FC) scheme and symbol

We propose to divide the traditional formal context into three parts:

- objects are described in the first table (Objects);
- attributes are described in the second table (Attributes);
- relations between objects and attributes are described in the third table (Relation). The proposed physical data model is shown in Figure 7.

To separate the distributional formal context (DFC) from traditional formal context (FC), we propose graphical notation:

- the traditional formal contexts are represented as rectangles (Fig. 6);
- the distributional formal contexts are represented as rectangles with bias (Fig. 7).

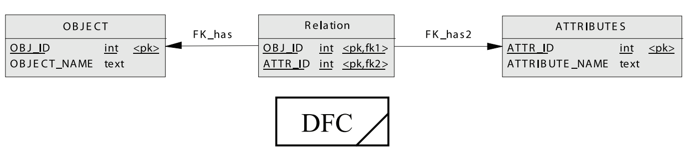


*Fig. 7.* The proposed logical scheme and notation of distributional formal context (DFC)

However, formal concept analysis needs traditional context (Figure 6). To convert distributional formal context to traditional context, additional transformation is needed. We propose the transformation table where relations between objects and attributes are saved. That transformation is used for formal concept analysis. In Figure 8 relation table transformations into traditional context are shown. The traditional formal context is created after transformation.

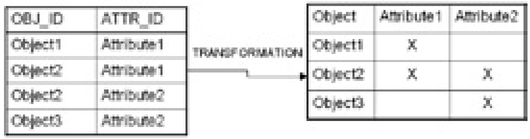


*Fig. 8.* Transformation of distributional formal context into traditional formal context

This solution allows to connect additional data to distributional formal context. Additional data can be either traditional formal context or distributional formal context. The type of context depends on one reason. Distributional formal context is always parent. Relationships between contexts are generally represented in context rectangles by a line. Context is parent when the line ends with dot.

Logical data model where the relations between two formal contexts are represented is shown in Figure 9. One is distributional formal context and the other is traditional formal context.

Logical data model where the relations between two distributional formal contexts are represented is shown in Figure 10.

Our proposed method is deriving *formal concepts* from *distributional formal context* with the following routine (first method when Context 1 is the main and used for data analysis; Context 2 will be used for saved additional data):

1) pick a set of objects A from Context 1;
2) derive the attributes A' from Context 1;
3) derive all formal concepts B' from Context 2;
4) derive (A')' from Context 1;
5) derive all formal concepts B'' from Context 2;
6) (B'', B') is a *formal concept.*

Let us present an example (Fig. 9) of generating formal concept from distributional formal context (where A is a distributional formal Context 1 and B is an additional formal Context 2):

1) pick any set of objects A from Context 1, e.g. A = {Object 3};
2) derive the attributes A' = {Attribute 3, Attribute 4, Attribute n};
3) derive all attributes B' (has relation with attributes from Context 1) from Context 2, e.g. B' = {Attribute 3 ({Object 1},{Attribute 1, Attribute 2}), Attribute 4 ({Object 2},{Attribute 1, Attribute 3}), Attribute n ({Object 3, Object 4},{Attribute 3, Attribute 4})};
4) derive (A')' = {Attribute 3, Attribute 4, Attribute n }' = {Object 3, Object 4};
5) derive all objects B' (has relation with objects from Context 1) from Context 2, e.g. B'' = {Object 3 ({Object 1},{Attribute 5, Attribute 6}), Object 4 ({Object 5},{Attribute 1, Attribute 2})};
6) (B'',B') = ({Object 3 ({Object 1},{Attribute 5, Attribute 6}), Object 4 ({Object 5},{Attribute 1, Attribute 2})},{Attribute 3 ({Object 1},{Attribute 1, Attribute 2}), Attribute 4 ({Object 2},{Attribute 1, Attribute 3}), Attribute n ({Object 3, Object 4},{Attribute 3, Attribute 4})}) is a *formal concept* from Context 1.

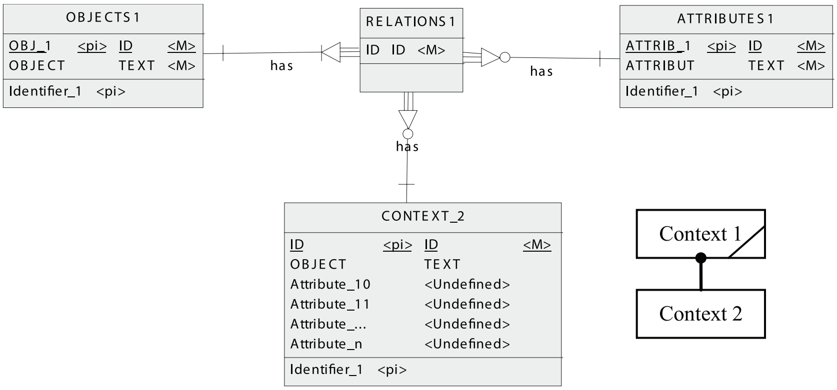


*Fig. 9.* Relationship between traditional formal context and distributional formal context

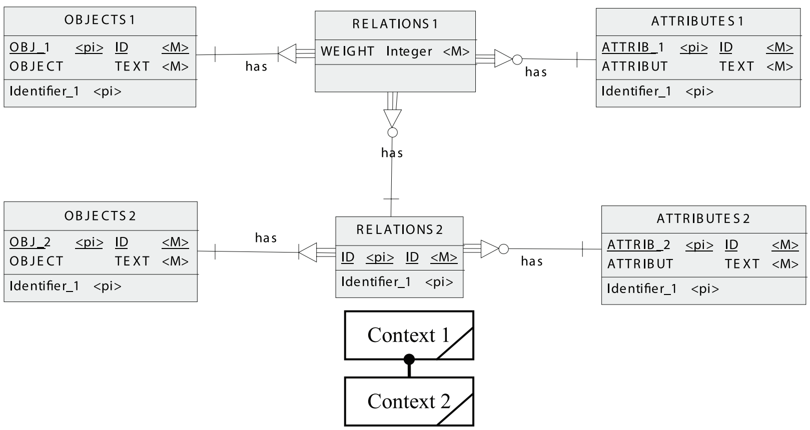


*Fig. 10.* Relationship between two distributional formal contexts

The second method of using distributional formal context is when parameters to select the specific context from Context 1 are described in Context 2. An example of the method is given in the next section.

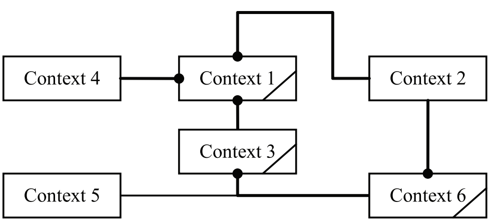


*Fig. 11.* Network of contexts

The proposed method theoretically and practically allows to compose the network from contexts (Fig. 11). We can see from this network example that some contexts are traditional (Contexts 2, 4, 5) and others are distributional (Contexts 1, 3, 6). One stage of the future work is to research and analyse the network of contexts.

## Distributional Formal Context for Collecting Many Contexts

The second method of using distributional formal context originates when parameters for selecting the specific context from Context 1 are described in Context 2. The Context 1 is multiple context and it means that from Context 1 we can get many contexts.

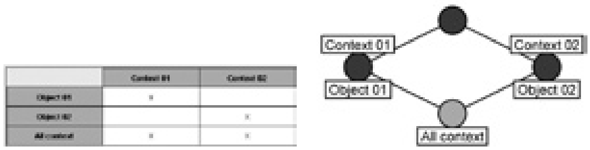


*Fig. 12.* Contexts described in Context 2

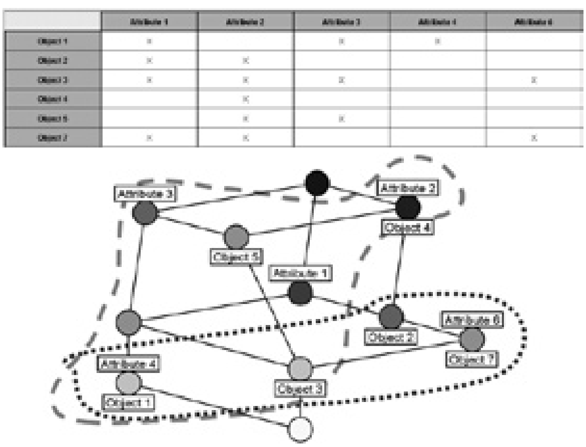


*Fig. 13.* Context 1 includes Context 01 and Context 02 (dashes show concepts of Context 2; squared dots show concepts of Context 1)

Let us present an example (Fig. 13) of generating formal concept from distributional formal context (where A is a distributional formal Context 1 and B is an additional formal Context 2):

1) pick any set of objects A from Context 2, e.g. A = {Object 01};
2) derive the attributes (from Context 2) A' = {Context 01 };
3) derive (A')' = {Object 01}' = {Context 01};
4) (A'',A') = ({ Object 01},{ Context 01}) is a *formal concept* of formal Context 2;
5) pick any set of objects B from Context 1 (has relation with formal concept (A'',A') from Context 2 (Table 1)), e.g. B = {Object 2};

6) derive the attributes from Context 1 (has relation with formal concept (A",A') from Context 2 (Table 1)) B' = {Attribute 1, Attribute 2};
7) Derive (B')' = {Object 2 }' = {Attribute 1, Attribute 2};
8) (B",B') = ({Object 2},{Attribute 1, Attribute 2}) is a *formal concept* of formal Context 1.

Figure 14 presents the lattice when we have selected all formal concepts from Context 1 (when Context 02 is selected (generated) from Context 1).

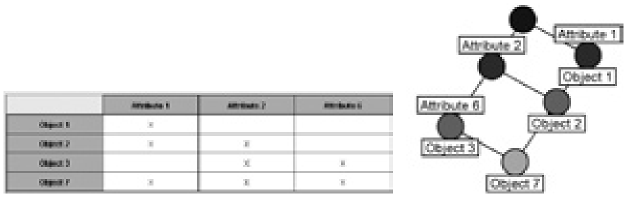


*Fig. 14.* Context 1 and lattice of Context 1

If we set the Object 2 from Context 2, we obtain lattice shown in Figure 15.

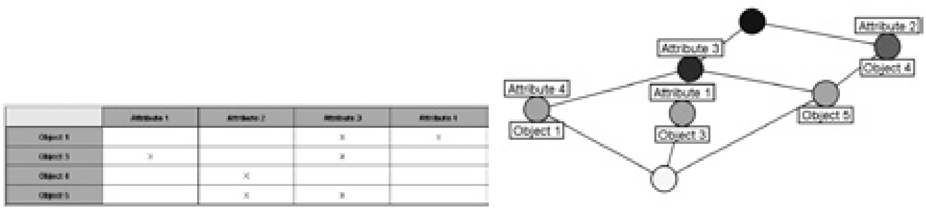


*Fig. 15.* Context 2 and lattice of Context 2

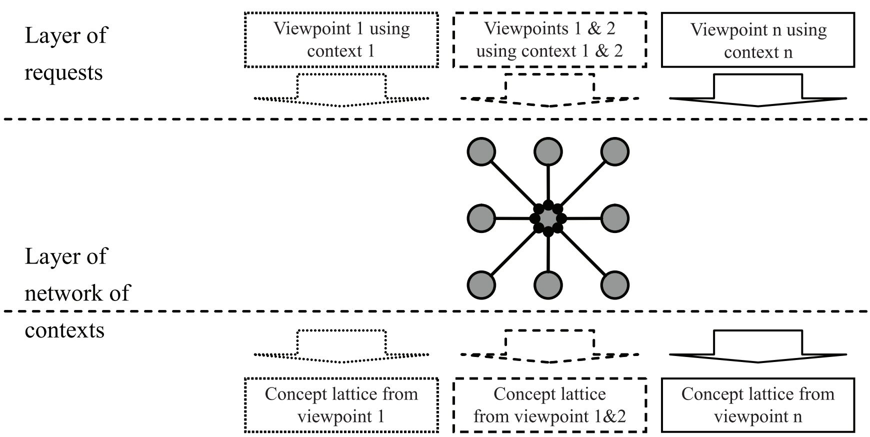


*Fig. 16.* Diagram of the process where usage of the context network is shown

*Table*

**Relations between objects and attributes of Context 1 and objects of Context 2**

| Objects (Context 1) | Objects (Context 2) |
|---|---|
| Object 1 | All context |
| Object 2 | Context 01 |
| Object 3 | All context |
| Object 4 | Context 02 |
| Object 5 | Context 02 |
| Object 7 | Context 01 |

| Objects (Context 1) | Objects (Context 2) |
|---|---|
| Attribute 1 | All context |
| Attribute 2 | All context |
| Attribute 3 | Context 02 |
| Attribute 4 | Context 02 |
| Attribute 6 | Context 01 |

The process of using distributional formal context for selecting the specific contexts from multiple formal contexts is presented in Figure 16.

## The Experiment

The program agent (Fig. 17) for information search tasks was used for the experiment. Its task was to collect information (from a specific domain) from Internet resources.

While developing the ontology, a set of notices of real estate were explored and analysed.

These notices were acquired from the following websites: *http://www.aruodas.lt, http://www.skelbiu.lt, http://www.domoplius.lt, http://www.alioreklama.lt, http://www.skelbimai.lt, http://www.edomus.lt, http://www.enamai.lt, http://www.city24.lt, http://www.namai.lt; http://www.muge.lt.*

Web Scraper Plus+ tool was used for data acquisition from the websites. The data was collected and saved in a database.

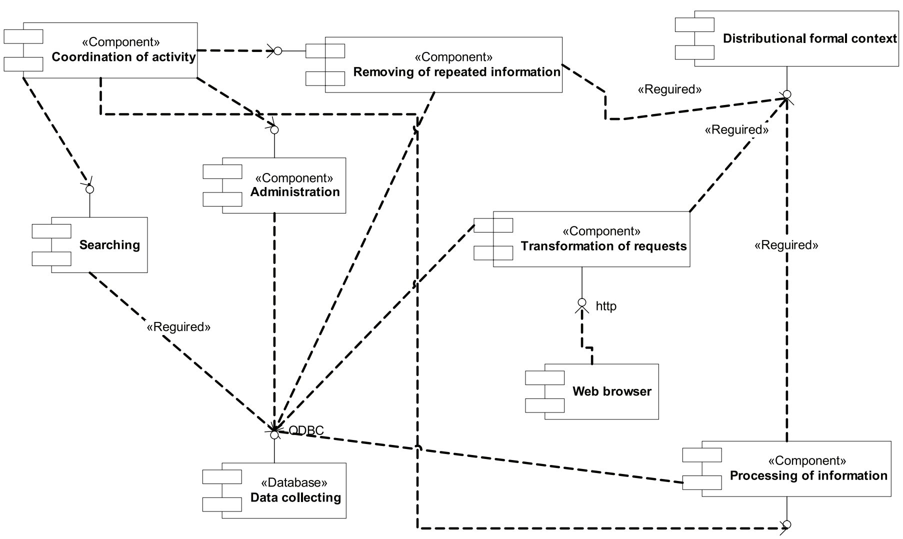


*Fig. 17.* Architecture of program agent for information search

During the information-processing step, objects (concepts) are acquired from collected information (Fig. 18). To perform this task, formal context and ontology of real estate are needed.

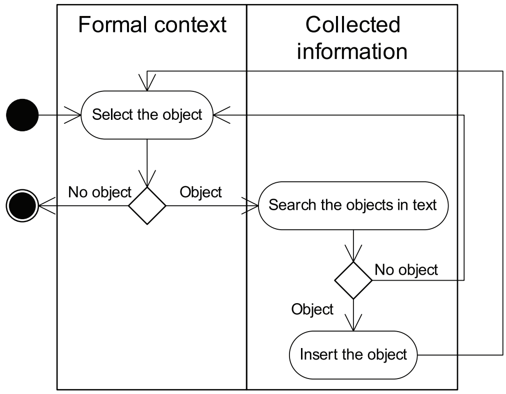


*Fig. 18.* Activity diagram of information processing

Next, we will describe how ontology and formal context was made. Two ontologies were developed: one – using the common method, and the other – using our proposed method (formal context).

### Development of Ontology Using the Common Method

Common ontology method was used to develop the first ontology (Fig. 1).

Protégé 3.2 and Gate 3.1 tools were used for the process of analysing and correcting terms. The terms were collected, then the text was analysed and annotated using the Gate tool (Fig. 19).

The developed ontology was presented with the Protégé tool (Fig. 20).

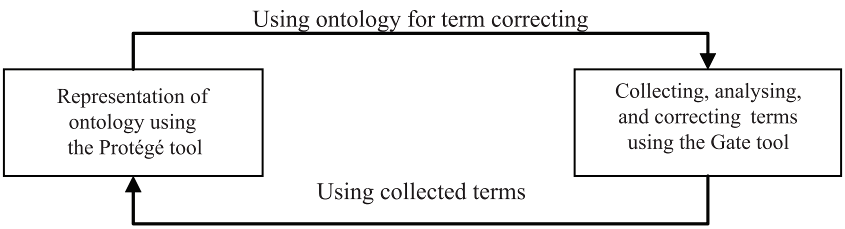


*Fig. 19.* The process of development of the ontology of real estate

### Creating the Classes

After the analysis of collected concepts, the tree of concepts was created (Fig. 20): real estate, location, action with real estate. Every concept has a lower-level concept.

The class “Real estate” has homes; flats; condition.
The class “Action with real estate” has sell; buy; rent; change.
The class “Location” has

- Vilnius:
  - Naujamiestis;
  - Fabijoniškės;
  - Šnipiškės;
  - Karoliniškės;
- Klaipėda:
  - Laukininkai;
  - Alksninė;
  - Pietinis;
- Kaunas:
  - Šilainai;
  - Dainava;
  - Žaliakalnis etc.

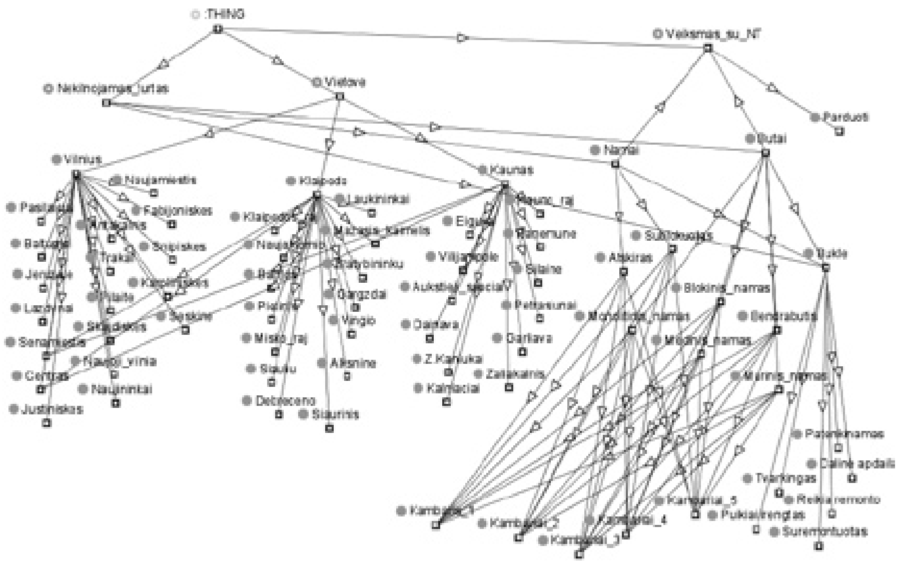


*Fig. 20.* A hierarchical diagram of the real estate ontology

**Development of Ontology (Formal Context) Using the Proposed Method**

The tool “Search of real estate system” was developed to carry out the experiment (Fig. 5, 21). This tool allows to fill the formal context in a database using a web browser. Searching and data analysis forms are created by using formal context. All data for creating these forms is saved in the formal context. That solution allows to change the content of forms dynamically. The tool was developed in PHP language and MySQL database was selected. 11 465 real estate commercials were used to perform the experiment. The main task was “Search”. Here we present more descriptions.

**Formal Context Filling**

"Search of real estate system" tool was developed (Fig. 5, 21) for formal context filling. The tool was created in PHP language and MySQL database was selected. This tool can be used to create search criteria and administrate the formal context. It allows to edit the formal context.


*Fig. 21.* Real estate system: inserting object in the formal context

ToscanaJ tool was used to analyze the data collected in the formal concept. ToscanaJ is a pure viewer/browser for conceptual schemes; it is optimized for a non-expert audience, it comes with additional tools for creating the data displayed and offers options of additional and more technical analysis. The four main tools are:

- ToscanaJ: the viewer/browser component;
- Elba: an editor for conceptual schemes on relational databases. Database-aware and offering extra tools like exporting SQL scripts;
- Siena: in many ways similar to Elba (mostly thanks to shared code), Siena edits conceptual schemes that store their data in memory;
- Lucca: an experimental editor that is supposed to make use of implication analysis of SQL clauses to allow very explorative and intuitive creation of database-connected systems.

ToscanaJ is the most prominent, however, possibly the most important program of the ToscanaJ suite. It is a very advanced viewer for conceptual schemes that is able to display information queried from the database in lattice diagrams or just using memory-mapped data structures (Fig. 4). There is an opportunity to insert the new concept into formal context quickly (Fig. 17).

The form hyperlink "INSERT" means that there are no searched concepts in the formal context. By clicking the hyperlink "INSERT", the concept can be inserted into the formal context (Fig. 21). Then information is processed.

## Results and Conclusions

The review of ontology and formal concept structures shows that they are quite similar and both are represented in a hierarchical form. Lightweight ontology has classes, attributes, and relationships. Objects, attributes and relationships are used for formal concept analysis as well. Nevertheless, heavyweight ontology allows to describe the domain in more detail because it has function terms, restrictions, rules, and axioms. Heavyweight ontology is extensively axiomatized and lightweight ontology often is presented as simple taxonomic structures, and is either slightly or not at all axiomatized.

This paper has proposed a structure of distributional formal context. That solution allows to save more data in the context and to simplify the usage of context in information systems. Two methods of using distributional formal contexts are described in this paper. One of these methods allows to connect the many contexts and to use multi-formal context in the network.

Performing the experiment using the method that realised regular process, concepts were collected and corrected in 1 week. Concepts were collected and corrected in 3 hours when the experiment was performed using our proposed method. Please note that the time of development of “Search of real estate system” tool is not included into evaluation. We also would like to stress that for performing the first experiment, an additional analyst was needed. To perform the second experiment, only the user was needed and the analyst could have been needed if another type of ontology was developed.

The experiment showed that the proposed method allows to search the terms (objects) quickly while analysing a large amount of data. We can show the hierarchical tree (structure) of concepts by using data collected in the formal context. Later, we can also develop ontology from the collected data using another representation language. That allows searching for the dependences, matching, repeated structures, exceptions, etc.

## References


1. Bugaite D., Vasilecas O. Ontology-Based Elicitation of Business Rules. In: A. G. Nilsson et al. (eds.) Proc. of the Fourteenth International Conference on Information Systems Development 2005, *Advances in Information Systems Development: Bridging the Gap between Academia and Industry.* Karlstad, Sweden, 15–17 August, 2005, Springer, 2006, p. 795–806.
2. Dau F. (2004) Types and Tokens for Logic with Diagrams. In: K. E. Wolff, H. Pfeiffer, & H. Delugach (eds.), *Conceptual Structures at Work: 12th International Conference on Conceptual Structures.* Berlin: Springer, p. 62–93.
3. Ganter B., Wille R. (1998) Applied Lattice Theory: Formal Concept Analysis. In: G. Grätzer: *General lattice theory.* 2nd ed. Birkhäuser Verlag, Basel.
4. Ganter B., Wille R. (1999) *Formal Concept Analysis: Mathematical Foundations*. Berlin-Heidelberg: Springer.
5. Godin R., Gecsei J., Pichet C. (1989) Design of browsing interface for information retrieval. In: N. J. Belkin & C. J. van Rijsbergen (eds.), *Proc. SIGIR ’89*, p. 32–39.
6. Guarino N. (1997) Understanding, building and using ontologies. *Int. Journal of Human-Computer Studies*, vol. 45, no. 2/3 (Feb/Mar).
7. Lavbic D., Krisper M. Rapid Ontology Development Model Based On Business Rules Management Approach For The Use In Business Applications. Proceedings of the 6th ICEIS Doctoral Consortium (DCEIS 2008) in conjunction with 10th International Conference on Enterprise Information Systems (ICEIS 2008), Cardoso, Jorge (ed.), Barcelona, Spain, pp. 24–35 (81).
8. Ling Liu and M. Tamer Özsu (eds.) (2008) *Encyclopedia of Database Systems*, Springer-Verlag.
9. Margolis E., Laurence S. (2007) The Ontology of Concepts – Abstract Objects or Mental Representations? *Nous*, vol. 41, issue 4, p. 561.
10. Martin J., Odell J. (1994) *Object-Oriented Methods: A Foundation*. Prentice-Hall, p. 52.
11. Mayers A., Maulsby D. (2004) Glossary (Online). Available: *http://acypher.com/wwid/BackMatter/Glossary.html.*
12. Sasa A., Matjaz B. J., Krisper M. Service-oriented framework for human task support and automation. *IEEE transactions on industrial informatics.* Nov. 2008, vol. 4, no. 4, p. 292–302.

13. Tilley T. (2004) *Formal Concept Analysis Application to Requirements Engineering and Design.* School of Information Technology and Electrical Engineering, University of Queensland, PhD Thesis, December. Available: *http://www.tomtilley.net/publications/tilley04formal.pdf*
14. Vasilecas O., Bugaite D. Ontology-based Information Systems Development: the Problem of Automation of Information Processing Rules. In: E. Neuhold, T. Yakhno (eds.), Proc. of the Fourth International Conference Advances in Information Systems (ADVIS‘2006), Izmir, Turkey, 18–20 October, 2006, Springer, *LNCS 4243*, p. 187–196.
15. Wikipedia (2008) Concept. Wikipedia: The Free Encyclopedia. [Accessed on 2008-02-22]. *http://en.wikipedia.org/wiki/Concept.*
16. Wikipedia (2008) Ontology (computer_science). Wikipedia. The Free Encyclopedia. Available: *http://en.wikipedia.org/wiki/Ontology_(computer_science)*
17. Wille R. (1982) *Restructuring Lattice Theory: an approach based on hierarchies of concept. Ordered sets* (editor I. Rival). Reidel, Dordrecht-Boston, p. 445–470.
18. Wille R. (1992) Concept Lattices and Conceptual Knowledge Systems. *Computers & Mathematics with Applications*, 23, 493–515.
19. Wille R. (1997a) Introduction to Formal Concept Analysis. In: G. Negrini (ed.), *Modelli e modellizzazione*. (Models and modelling.) Consiglio Nazionale delle Ricerche, Instituto di Studi sulli Ricerca e Documentazione Scientifica, Roma, 39–51.
20. Wille, R. (1997b) Conceptual Graphs and Formal Concept Analysis In: Dickson Lukose, Harry Delugach, Marry Keeler, Leroy Searle, and John F. Sowa (eds.), Conceptual Structures: Fulfilling Peirce’s Dream, Proc. of the Fifth Int. Conf. on Conceptual Structures (ICCS’97), August 3–8, University of Washington, Seattle, USA, *LNAI, Number 1257*, Springer Verlag, Berlin, 290–303.
21. Wille R. (2001) Why Can Concept Lattices Support Knowledge Discovery in Databases? *ICCS’01 International Workshop on Concept Lattices-Based KDD*, 7–20.
22. Wolf K. (1993) A first course in formal concept analysis. In: Faulbaum, F. (ed.), *SoftStat’93 Advances in Statistical Software 4*, p. 429–438.
23. WORDNET (2008) A lexical database for the English language [Online]. Available: *http://wordnet.princeton.edu/ perl/webwn?s =concept*