# Collaborative Conceptual Exploration as a Tool for Crowdsourcing Domain Ontologies

Sergei Obiedkov and Nikita Romashkin

National Research University Higher School of Economics, Moscow, Russia
sergei.obj@gmail.com, romashkin.nikita@gmail.com

**Abstract.** Domain ontologies are essential in disciplines as diverse as software engineering, medicine, or political science to name just a few. This paper describes an ongoing effort to develop a methodology for collaborative ontology construction by geographically spread communities of experts and implement a web-based prototype supporting this methodology. A distinctive feature of the proposed approach is the use of conceptual exploration techniques, which make it possible to organize the process of ontology construction by automatically identifying and explicitly highlighting issues that remain to be addressed. Given a set of objects (facts, situations, etc.) of a subject domain, which is known to have considerably more such objects, and their unified descriptions in terms of presence or absence of certain attributes, a conceptual exploration system maintains a compact representation of implications behind the currently built ontology and offers them for experts to accept or falsify by entering new objects or extending the description language with new attributes. Upon termination, exploration results in identification of a (relatively small) representative part of the domain from which a conceptual hierarchy of the entire domain can be automatically constructed. We consider theoretic, algorithmic, representational, and pragmatic issues of transforming the exploration methods into a toolset useful for domain experts.

## 1 Introduction

Domain ontologies provide common interfaces to knowledge accumulated in their respective fields through achieving consensus on terminology used therein and its meaning. Their construction is thus an important topic in knowledge engineering. In this paper, we describe a project aimed at developing a platform for online collaborative ontology construction by geographically distributed groups of users that would not simply support sharing and common editing of formalized knowledge, but would also include means for on-the-fly validation of the ontology being constructed and automatic generation of guidelines for its completion.

The proposed approach is based on formal concept analysis, a mathematical theory oriented at applications in knowledge representation, knowledge acquisition, data analysis and visualization [5]. It provides tools for understanding the structure of data given as a set of objects with certain descriptions, e.g., in terms of their attributes, which is done by representing the data as a hierarchy of concepts, or more exactly, a concept lattice (in the sense of lattice theory). The objects, attributes, and relation between them constitute a formal context; hence, the definition of a concept is necessarily contextual. Every concept has extent (the set of objects that fall under the concept) and intent (the set of attributes or features that together are necessary and sufficient for an object to be an instance of the concept). Concepts are ordered in terms of being more general or less general (i.e., covering more objects or fewer objects).

The concept lattice, being a rather universal structure, provides a wealth of information about the relations among objects and attributes, which made possible applications in areas ranging from sociology [8] to ontology construction [14]. Indeed, it can help in processing a wide class of data types providing a framework in which various data analysis and knowledge acquisition techniques can be formulated.

One such technique is attribute exploration [4], which, in its basic version, can be summarized as follows. Given a set of objects of a domain, which is known to have many more such objects, and their descriptions in terms of presence or absence of certain attributes, attribute exploration builds an implicational theory of the entire domain and a representative set of its objects. The implicational theory is the set of sentences of the form "if an object has all attributes from set $A$, then it also has all attributes from set $B$" that are believed to hold for all objects of the domain. It is always possible to find a minimal representation of such a set, the Duquenne–Guigues or canonical basis of implications, from which all valid implications can be inferred [6]. The representative set of objects must respect all implications from the generated canonical basis and provide a counterexample for every implication that cannot be inferred from the basis (in this case, the concept lattice of the domain is isomorphic, i.e., structurally identical, to the concept lattice of this relatively small set of objects).

The process of attribute exploration is interactive: it consists in computer suggesting implications and the user accepting them or providing counterexamples. Attribute exploration is designed to be efficient, i.e., to suggest as few implications as possible without loss in completeness of the result. It can work even if the initial set of objects is empty. Attribute exploration is domain-independent: although its application is more straightforward in precise domains such as mathematics [11], [12], it can certainly be used in other fields [10].

Advanced versions of attribute exploration can take into account background information, such as relations among attribute values (not necessarily given as implications), thus, avoiding suggesting trivial implications [4]. There are also methods of concept exploration, relational exploration, and rule exploration that, each in its own way, generalize attribute exploration, making it possible to work with a broader class of dependencies [15, 13, 16]. We refer to all such methods collectively as *conceptual exploration*.

Our aim is to extend these methods so as to make them suitable for collaborative ontology construction over the web by a geographically spread community of researchers working in the same domain. The idea is to provide tools that would allow them to use attribute exploration and related techniques to refine the language in which their domain is described and boost their knowledge about it. Being suggested, an implication will be accepted only if no expert has a counterexample for it. In the end, there will be a list of implications correctly describing objects under study and a representative context. If, at a later stage, a counterexample becomes known, it is added to the context and the implication basis is modified accordingly. Thus, an up-to-date list of open problems of the domain can be maintained.

We start with a short introduction into the relevant aspects of formal concept analysis and then define attribute exploration. After that, we discuss possible ways to make this procedure collaborative. Finally, we describe the current state of our web-based exploration system and the work still to be done.

## 2 Concept Lattices

We briefly introduce necessary mathematical definitions [5] and then explain them less formally. Given a *(formal) context* $\mathbb{K} = (G, M, I)$, where $G$ is called a set of *objects*, $M$ is called a set of *attributes*, and the binary relation $I \subseteq G \times M$ specifies which objects have which attributes, the derivation operators $(\cdot)^I$ are defined for $A \subseteq G$ and $B \subseteq M$ as follows:

$$A^I = \{m \in M \mid \forall g \in A : gIm\}$$

$$B^I = \{g \in G \mid \forall m \in B : gIm\}$$

In words, $A^I$ is the set of attributes common to all objects of $A$ and $B^I$ is the set of objects sharing all attributes of $B$.

If this does not result in ambiguity, $(\cdot)'$ is used instead of $(\cdot)^I$. The double application of $(\cdot)'$ is a closure operator, i.e., $(\cdot)''$ is extensive, idempotent, and monotonous. Therefore, sets $A''$ and $B''$ are said to be *closed*.

A *(formal) concept* of the context $(G, M, I)$ is a pair $(A, B)$, where $A \subseteq G$, $B \subseteq M$, $A = B'$, and $B = A'$. In this case, we also have $A = A''$ and $B = B''$. The set $A$ is called the *extent* and $B$ is called the *intent* of the concept $(A, B)$.

A concept $(A, B)$ is a *subconcept* of $(C, D)$ if $A \subseteq C$ (equivalently, $D \subseteq B$). The concept $(C, D)$ is then called a *superconcept* of $(A, B)$. We write $(A, B) \leq (C, D)$. The set of all concepts ordered by $\leq$ forms a lattice, which is called the *concept lattice* of the context $\mathbb{K}$.

The formal context makes precise the scope of the discussion by specifying the domain to which it applies (listing all the objects of this domain) and defining the terms in which it is going to be discussed (listing the attributes to be used in object descriptions).

To flesh this out a bit, we give a small example based on the data from the O*NET Resource Center (`http://www.onetcenter.org/`), which essentially provides an interface to a taxonomy of occupations, organizing occupations in various groups and describing the knowledge, skills, and abilities required by

| | Computers and Electronics | Education and Training | Administration and Management | Mathematics | Physics |
|---|---|---|---|---|---|
| Computer and Information Research Scientists | × | × | × | × | |
| Computer Programmers | × | | × | × | |
| Mathematicians | × | | | × | × |

**Fig. 1.** A formal concept of some computer and mathematical occupations.

each occupation. Here, we focus on the knowledge required by occupations from the Computer and Mathematical job family.

A formal context encompassing three occupations is shown in Fig. 1. Here, rows correspond to objects (occupations) and columns correspond to attributes (areas of knowledge). A cross indicates that the corresponding object has the corresponding attribute; in our case, it means that an occupation requires knowledge in a certain area.

A line diagram of the concept lattice of this context is shown in Fig. 2. Nodes correspond to formal concepts, with more general concepts placed above less general ones. Two concepts are connected with a line if one is more general than the other and there is no concept between the two. Every concept in the diagram is described extensionally, by a group of objects, and intensionally, by attributes shared by all the objects in the extent of this node. The names of the objects in the extent of a node can be read off from the diagram by looking at the labels immediately below this node and below all nodes that can be reached from this node by downward arcs. Conversely, the set of attributes forming the intent of a node consists of labels immediately above this node and those above nodes that can be reached from this node by upward arcs. For example, the bottom-right node corresponds to the concept whose extent consists of a single occupation, Computer and Information Research Scientists, whereas its intent includes all attributes but *Physics*. The top concept is labelled by *Computers and Electronics* and by *Mathematics*, which means that all the three occupations require knowledge in these two areas.

It can be seen from this diagram that all occupations in our context requiring knowledge in Education and Training also require knowledge in Administration and Management. This is formally captured by the notion of an *implication*,

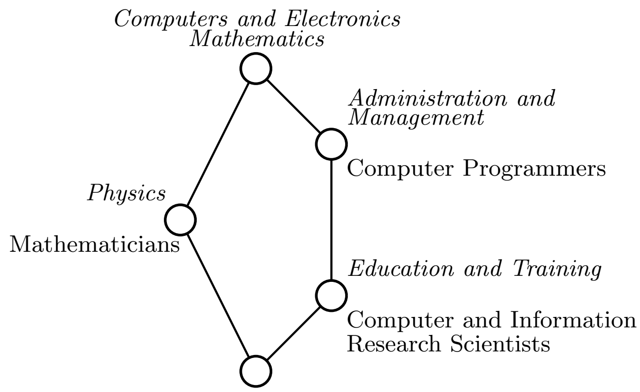


**Fig. 2.** The concept lattice of the context in Fig. 1.

which is, formally, an expression $A \to B$, where $A, B \subseteq M$ are attribute subsets. It *holds* or *is valid* in the context if $A' \subseteq B'$, i.e., every object of the context that has all attributes from $A$ also has all attributes from $B$.

An attribute subset $X \subseteq M$ *respects* or is a *model* of an implication $A \to B$ if $A \not\subseteq X$ or $B \subseteq X$. Obviously, an implication holds in a context $(G, M, I)$ if and only if $\{g\}'$ respects the implication for all $g \in G$. If an object $g \in G$ is such that $\{g\}'$ is not a model of $A \to B$, we will call $g$ a *counterexample* to this implication.

If $A' = \varnothing$ for $A \subseteq M$, then the implication $A \to M$ necessarily holds in the context. We will sometimes write such an implication as $A \to \bot$, with $\bot$ standing for "contradiction", meaning that attributes of $A$ never occur all together.

All valid implications of the context can be summarized by means of the *Duquenne–Guigues basis*:

$$\{P \to P'' \setminus P \mid P \subseteq M \text{ is pseudo-closed}\},$$

where a set $P \subseteq M$ is recursively defined to be *pseudo-closed* if $P \neq P''$ and $Q'' \subset P$ for every pseudo-closed $Q \subset P$. The models of these implications are precisely the models of all implications valid in the context, and the Duquenne–Guigues basis has the smallest number of implications among all implication sets with this property [6]. Any valid implication can be inferred from the basis using Armstrong rules [2].

The Duquenne–Guigues basis of the context in Fig. 1 consists of three implications shown in Fig. 3. Although they look similar, it may be more intuitive to read them differently:

- The first implication says that all occupations require knowledge both in Computers and Electronics and in Mathematics.
- The second implication says that, if an occupation requires knowledge in Computers and Electronics, Mathematics, and Education and Training, then it also requires knowledge in Administration and Management.

$\varnothing \to \{\text{Computers and Electronics}, \text{Mathematics}\}$ 3

$\{\text{Computers and Electronics}, \text{Mathematics}, \text{Education and Training}\} \to$
$\{\text{Administration and Management}\}$ 1

$\{\text{Computers and Electronics}, \text{Mathematics}, \text{Administration and Management}, \text{Physics}\}$
$\to \{\text{Education and Training}\}$ 0

**Fig. 3.** The Duquenne–Guigues basis of the context in Fig. 1. The number in the end of each line is the number of objects that "support" the implication, i.e., contain all attributes from its premise.

– The third implication means that there are no occupations requiring at the same time knowledge in Computers and Electronics, Mathematics, Administration and Management, and Physics.

These implications are valid in our context, but the context contains only three occupations. How do we know if the three implications are valid for all computer and mathematical occupations out there? This is where attribute exploration becomes useful.

## 3 Attribute Exploration

The idea of attribute exploration is simple: consider every implication in the Duquenne–Guigues basis of the context and add a counterexample if the implication is not valid generally. To be more precise, we are dealing with two contexts here: one corresponds to the entire subject domain (computer and mathematical occupations, in our case) and may not be immediately observable in its entirety, while the other contains only a selection of objects. This smaller context is the one that we can put into a cross-table such as one in Fig. 1 and for which we can compute the implication basis. The goal is to make the smaller context representative of the larger context in a very precise sense: the two contexts must be models of exactly the same implications. Implications valid in the larger context are always valid in the smaller context; we can force the converse to become true as well by adding counterexamples to implications valid in the smaller context, but not in the larger one. When we are done, the two contexts will share the same implication basis and, furthermore, have isomorphic concept lattices: in other words, the concept intents will be the same in the two contexts.

As an example, consider again the lattice in Fig. 2 and the corresponding implication basis in Fig. 3. The implication

$$\varnothing \to \{\text{Computers and Electronics}, \text{Mathematics}\}$$

is valid in our current context in Fig. 1, but according to the O*NET data, knowledge of Mathematics is not really essential for Clinical Data Managers, whose role is to "apply knowledge of health care and database management to analyze clinical data, and to identify and report trends". Out of the five areas we consider, they must have knowledge of Computers and Electronics and of Administration and Management. Therefore, we add Clinical Data Managers to our context as a counterexample to the above implication:

| | | | | | |
|---|---|---|---|---|---|
| Clinical Data Managers | × | | × | | |

Note that Clinical Data Managers still must have knowledge of Computers and Electronics; therefore, we still have to consider the implication

$$\varnothing \to \{\text{Computers and Electronics}\}.$$

It seems that knowledge of Computers and Electronics is a requirement for all computer and mathematical occupations; so, we accept the implication.

Since we changed the context by adding a new object, the implication basis has also changed. Now, it includes the implication

$$\{\text{Computers and Electronics}, \text{Physics}\} \to \{\text{Mathematics}\}$$

suggesting that, for an occupation within the job family under consideration, knowledge of physics is useless without knowledge of mathematics. If we are to believe the O*NET data, the knowledge of Physics is not required for anyone within this job family but Mathematicians, for whom the knowledge of mathematics is obviously a must. Thus, we accept the implication.

For the same reason, we accept the third implication in Fig. 2: the only occupation category requiring knowledge of Physics is Mathematicians, but it does not require knowledge in Administration and Management; hence, there is no occupation satisfying the premise of the implication and the implication trivially holds.

In our modified context, there remains only one implication to consider: Is it true that, if an occupation requires knowledge both in Computers and Electronics and in Education and Training, then it also requires knowledge in Mathematics and in Administration and Management (as it is the case with Computer and Information Research Scientists)? The answer is negative, because this does not hold for Informatics Nurse Specialists, who are there to "apply knowledge of nursing and informatics to assist in the design, development, and ongoing modification of computerized health care systems". They "may educate staff [...] to promote the implementation of the health care system", and, therefore, need knowledge in Education and Training, but knowledge of Mathematics is not a requirement for them. We add a new object to our context:

| | | | | | |
|---|---|---|---|---|---|
| Informatics Nurse Specialists | × | × | × | | |

Proceeding likewise, we add

| | | | | | |
|---|---|---|---|---|---|
| Statisticians | × | × | | × | |

to give an example of those who, according to O*NET, typically need knowledge of Education and Training, but not of Administration and Management; accept the implication indicating that no occupation in this job family typically requires the simultaneous knowledge of Physics and of Education and Training; and we are done. The resulting concept lattice is shown in Fig. 4. Below, we present the resulting implication basis:[1]

- $\varnothing \to \{\text{Computers and Electronics}\}$
- $\{\text{Physics}\} \to \{\text{Mathematics}\}$
- $\{\text{Administration and Management}, \text{Physics}\} \to \bot$
- $\{\text{Education and Training}, \text{Physics}\} \to \bot$

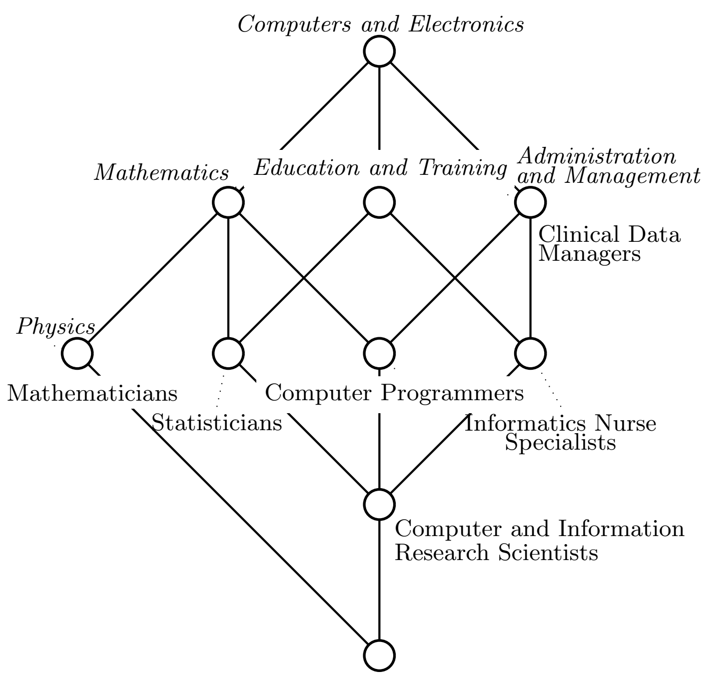


**Fig. 4.** The concept lattice resulting from attribute exploration on the context in Fig. 1.

Although we have considered only a small fraction of computer and mathematical occupations, these four implications completely characterize the implicational theory of *all* such occupations (at least, as long as we trust the O*NET

[1] Here, implication premises are given in an abbreviated form using their so-called minimal generating sets: instead of a pseudo-closed set $P$, we use its minimal subset $Q$ satisfying $Q'' = P''$. The resulting implication set is equivalent to the Duquenne–Guigues basis.

data) and the six occupations covered by the concept lattice in Fig. 4 are representative of the job family in this sense. The concept lattice contains all relevant combinations of knowledge areas that may be required for computer and mathematical occupations, and each such occupation can be placed into one of the ten concepts of this lattice. For example, Web Administrators must have knowledge of Computers and Electronics and of Administration and Management, but not of any of the other three areas; therefore, they are covered by the same concept as Clinical Data Managers.

But this is counterintuitive: surely, Web Administrators might need knowledge in areas foreign to Clinical Data Managers. The problem is that the six occupations we identified during attribute exploration and the concept lattice generated from them are representative only with respect to the five knowledge areas we have chosen before. However, the choice of knowledge areas might not be representative itself. To correct this, we may use *object exploration*, a process dual to attribute exploration, which involves working with object implications, such as

> {Informatics Nurse Specialists} $\rightarrow$ {Computer and Information Research Scientists}.

In our case, this implication is interpreted as follows: "Every knowledge area essential for Informatics Nurse Specialists is also important for Computer and Information Research Scientists." This is not so, since Informatics Nurse Specialists are typically required to have knowledge also in Medicine and Dentistry, which forces us to introduce a new attribute into the context. Thus, object exploration ensures that the language we use to describe the domain is sufficiently rich to differentiate between what must be differentiated.

Alternating between object and attribute exploration in this way, we can arrive at a complete conceptual hierarchy (in the form of a concept lattice) of the domain under consideration. Such a hierarchy modeling the subconcept–superconcept relationship is an essential part of any domain ontology. Advanced versions of attribute exploration, such as rule exploration [16], may be used to identify other types of relations between objects or classes of objects.

## 4 Making Exploration Collaborative

The construction of the concept lattice is automatic, but the data needed for that is gathered through exploration in an interactive fashion: the exploration system asks the user questions and the user replies positively by accepting an implication or negatively by providing a counterexample. Extending this approach to the case of many users will make it possible to apply it in construction of ontologies for large domains of which experts have only partial and mutually complementary knowledge.

It is important to note that users of such a system must not be knowledge engineers: they never have to explicitly describe the ontology of the domain under consideration using a specialized knowledge representation formalism. Instead, they only supply data sufficient for building such ontology automatically, while the exploration system directs them through questions/implications. If exploration is used within a scientific project, the users will typically be domain experts. In other cases, it is possible that they even come from the "general public". For example, if we were to collect data about occupations, we would welcome input not only from job market research analysts, but also from people working in HR departments, who know their companies' requirements, as well as from job seekers, who know about such requirements from their own experience. Thus, conceptual exploration provides a foundation for a toolset for crowdsourcing data based on which a domain ontology (or its part) can be automatically constructed.

One of the first experiments of collaborative exploration was conducted within lattice theory already twenty years ago. The aim was to study relations between various properties of lattices: algebraic, atomistic, finite, etc., and, indeed, interesting non-trivial dependencies have been discovered [11]. In that setting, valid implications corresponded to theorems, which had to be formally proven. The resulting concept lattice (of lattice properties) was found valuable by lattice theorists.

Obviously, the Internet together with appropriate software can make the process much better organized and more realizable in practice. We experimented with building a web system supporting attribute exploration within a joint project with political scientists at Higher School of Economics in Moscow. The project had as one of its goals developing a hierarchy of concepts of democracy defined in literature and implemented in practice. The focus was on so-called "defective democracies", such as "delegative democracy" or "exclusive democracy". To make the construction of this concept hierarchy easier and more transparent, a website was set up in order to allow political scientists to collectively build a corresponding formal context using attribute exploration and related tools. Objects of this context were countries and/or regimes described in political science literature, and attributes were their various properties. During exploration, users consider implications of the form:

> "If a regime is characterized by all attributes from set $A$, it is also characterized by all attributes from set $B$,"

which they can accept or reject. In the latter case, the user must describe a regime characterized by all attributes from $A$, but not by all attributes from $B$. Similarly, object implications of the form

> "Every attribute characterizing all regimes in $A$ is shared by all regimes in $B$"

propose the user to differentiate between regimes from sets $A$ and $B$ by adding a new attribute shared by all regimes in $A$ but missing from some regimes in $B$.

It turned out however that there are a number of issues that must be resolved before exploration techniques become a truly useful tool for people not experienced in knowledge representation and formal concept analysis. Therefore, we settled down to developing a general paradigmatic model of collaborative conceptual exploration and its prototype implementation. A website is under development that currently supports only the basic version of attribute exploration, at the same time, providing some standard facilities of a collaborative environment such as user profiles, separate projects, etc.

Upon signing up, the user can start a new exploration project or join an existing project by contacting its owner. Of course, one user may participate in several projects. A typical workflow is as follows: project members start attribute exploration by defining an initial list of attributes (which can afterwards be altered in any way) and continue by reviewing the dynamically updated implication basis as described above, i.e., by accepting implications they deem valid and providing counterexamples to other implications. The formal context maintained in the process can be modified directly, as long as the modifications do not conflict with the accepted implications.

There are many issues that need to be addressed to allow for exploration of complex domains:

- Object exploration was discussed earlier; it is essential for identifying features differentiating objects from each other.
- Incomplete specification of examples. Users must still be able to add a counterexample for an implication to the system even if they are unsure about its status with respect to attributes not occurring in this implication. Its description will be completed at later stages, either manually or automatically (if accepting an implication forces certain values for some attributes that have been left unspecified). This requires a modified definition of the implication basis allowing for incompletely specified objects.
- Background knowledge. There must be a way to explicitly add known facts about the domain to speed up exploration and concentrate on new knowledge. The implication basis must be built relative to these facts summarizing the part of knowledge behind the context not covered by them.
- Algorithmic issues. Since, for large ontologies, the canonical basis may also be large, algorithms allowing for incremental update [9] of and navigation through the basis must be developed.
- The policy of collaboration. It is essential to develop methods for resolution of conflicts arising when different users have different views on the status of an implication or when new information becomes available providing a counterexample for an already accepted implication.
- Supporting tools. There must be a way for the user initiating any modification to annotate it: provide a proof for an accepted implication, describe the meaning of the new attribute, upload a document with evidence for the new object, etc. In some domains, it may be possible to automatically prove implications or search the Internet for potential counterexamples.
- More complex languages for object description. Attribute exploration can easily be extended to the case of many-valued contexts (non-binary object–attribute tables) by means of conceptual scaling [5]. However, which scales

to use for a many-valued attribute should be a matter of collective decision; support for organizing the process of decision making must be envisioned. Another possible extension is to the case when objects are described by arbitrary formulas rather than only by conjunctions of attributes.

– Integration with state-of-the-art tools for ontology development, especially those based on description logics. Attribute exploration has been used for completing description logic knowledge bases, see, e.g., [3], although in a slightly different context. Another relevant line of research to be taken into account is learning ontologies in the framework of exact learning with queries [1], which has a lot in common with attribute exploration; see, e.g., [7].
– Non-implicational knowledge. In some cases, it may be desirable to use a richer subset of propositional logic to summarize the theory of the domain.
– Relational exploration. Attribute/object exploration is, for the most part, concerned with the subsumption hierarchy of domain concepts. It may be desirable to extract knowledge about other relations between concepts; relational or rule [16] exploration may be adapted for this purpose.
– Merging results of several explorations. It may be more appropriate for users to work with subcontexts corresponding to their field of expertise rather than with a larger context. Methods for combining the results of several explorations must be devised.

## 5 Conclusion

We believe that conceptual exploration techniques provide a powerful framework for collaborative development of domain ontologies. Recent developments in Web applications open new prospects for tools based on exploration. A proper implementation may result in a platform effectively supporting online social networks of experts exchanging knowledge in a structured way and working together towards constructing an ontology of their field. Nevertheless, a considerable amount of research, development, and testing is still needed to unleash the full potential of conceptual exploration in a distributive environment.


## Acknowledgment

The first author was supported by the Russian Foundation for Basic Research grant no. 14-01-93960.


## References


1. Angluin, D.: Queries and concept learning. Machine Learning 2, 319–342 (1988)
2. Armstrong, W.: Dependency structure of data base relationships. Proc. IFIP Congress pp. 580–583 (1974)
3. Baader, F., Ganter, B., Sertkaya, B., Sattler, U.: Description logic knowledge bases using formal concept analysis. In: Veloso, M.M. (ed.) Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07). pp. 230–235 (2007)

4. Ganter, B.: Attribute exploration with background knowledge. Theoretical Computer Science 217(2), 215–233 (1999)
5. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin/Heidelberg (1999)
6. Guigues, J.L., Duquenne, V.: Famille minimale d'implications informatives résultant d'un tableau de données binaires. Mathématiques et Sciences Humaines 24(95), 5–18 (1986)
7. Konev, B., Ozaki, A., Wolter, F.: A model for learning description logic ontologies based on exact learning. In: Proceedings of AAAI-16 (2016)
8. Mohr, J.W., Duquenne, V.: The duality of culture and practice: Poverty relief in New York City, 1888–1917. Theory and Society 26, 305–356 (1997)
9. Obiedkov, S., Duquenne, V.: Attribute-incremental construction of the canonical implication basis. Annals of Mathematics and Artificial Intelligence 49(1-4), 77–99 (April 2007)
10. Obiedkov, S.A., Kourie, D.G., Eloff, J.H.P.: Building access control models with attribute exploration. Computers & Security 28(1-2), 2–7 (2009)
11. Reeg, S., Wei, W.: Properties of finite lattices. Diplomarbeit, TH Darmstadt (1990)
12. Revenko, A., Kuznetsov, S.O.: Attribute exploration of properties of functions on sets. Fundam. Inform. 115(4), 377–394 (2012), http://dx.doi.org/10.3233/FI-2012-660
13. Rudolph, S.: Relational Exploration—Combining Description Logics and Formal Concept Analysis for Knowledge Specification. Universitätsverlag Karlsruhe (Dec 2006), dissertation
14. Stumme, G., Maedche, A.: FCA-MERGE: Bottom-Up Merging of Ontologies. In: Proc. of the $17^{th}$ International Joint Conference on Artificial Intelligence. pp. 225–234 (2001)
15. Stumme, G.: Concept exploration: knowledge acquisition in conceptual knowledge systems. Ph.D. thesis, TH Darmstadt (1997)
16. Zickwolff, M.: Rule exploration: first order logic in formal concept analysis. Ph.D. thesis, TH Darmstadt (1991)