

# Towards Semantic Knowledge Base Definition

Marek Krótkiewicz<sup>1</sup>, Krystian Wojtkiewicz<sup>1,3</sup> , and Marcin Jodłowiec<sup>2,3</sup>

<sup>1</sup> Department of Information Systems,  
Wrocław University of Science and Technology,  
Wrocław, Poland

{m.krotkiewicz,k.wojtkiewicz}@kieg.org

<sup>2</sup> Institute of Computer Science,  
Opole University of Technology, Opole, Poland

m.jodlowiec@kieg.org

<sup>3</sup> Institute of Control Engineering,  
Opole University of Technology, Opole, Poland

**Abstract.** The paper is a wide survey over one of the knowledge representation and processing solutions, namely knowledge bases. Due to current terminological inconsistency authors propose the complex definition of knowledge base in the field of knowledge representation. The overview of the most common reality description methods is provided in order to discuss its usefulness in knowledge base design. Authors not only give the definition of the knowledge base but also prove its completeness on the example of Semantic Knowledge Base project. The project aims at developing the general domain knowledge base using ontology base and semantic networks as basic knowledge representation methods.

**Keywords:** Knowledge representation · Knowledge base · Ontology  
Semantic network

## 1 Introduction

Knowledge has been the source and target of studies from the very beginning. Plato was probably the first philosopher who tried to set the definition of knowledge. He started from “Knowledge as perception”, to propose another “Knowledge is True Judgment” to finally provide definition, that is: “Knowledge is true judgment with an account”. The Plato’s Theaetetus [3] just started the discussion that still lasts. After over 2000 years, we still cannot provide one universal definition of Knowledge. However for the purpose of this article we will define it as “Data with information how to use/interpret it”.

Knowledge design and processing is one of the most complex contemporary research issues in the computer science. The design of the knowledge base is crucial for the effectiveness of the expert system build over it [10]. Main problem associated with this task is the availability of models flexible enough to build not only data structures but also structures for storage of knowledge. Relational

model, that is the most common model in databases design, is at the same time very inefficient in the means of complex, hierarchical data structures [6, 13, 27, 31].

The description of complexity of the reality is an important aspect of research regarding artificial intelligence. The study developed many different approaches that not only intermingled with each other but most importantly they enabled to increase the expression of the phenomena and facts described. Below, the methods of describing the reality conceptualization most frequently mentioned in the literature of the subject are presented.

## 2 Basic Methods of Conceptualization of Reality

### 2.1 Abstraction

One of the basic methods used from the beginning of modelling. Its main assumption is the principle of functional decomposition that is based on defining operations as more elementary operations. Each level of decomposition leads to the increase of detail in the description of primary operation by introducing new elements of description that on the higher level of abstraction are not important. As the example, let us consider the operation of “building of a house” as the highest level of abstraction. Defining such an operation the following sub-operations can be specified:

1. House construction preparations,
2. House construction realization,
3. House construction finalization.

The presented lower level operations are only the examples. By creating the hierarchic structure of the operation the aspects that may constitute the independent dimension of this description can be taken into account. For example, the operation “house construction preparation” can have its aspect (dimension), e.g.: financial, organizational, technical.

In each of this dimensions there is an independent hierarchy of the operation in the sense of division into aspects, what was presented on the Fig. 1, what is not contradictory to the possible relations between operations only, in the functional sense.

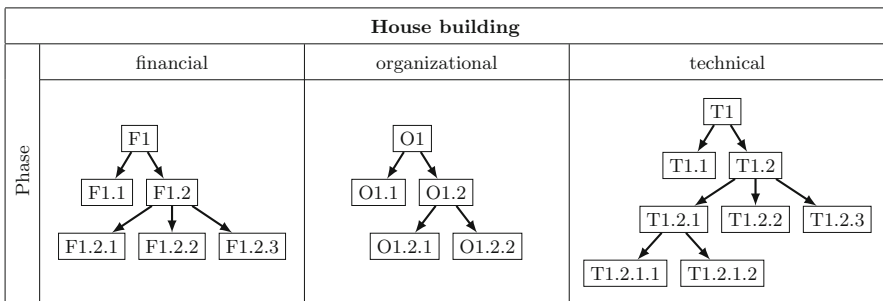


Fig. 1. House building abstraction

For instance, the preparation in the technical and organizational sense will certainly depend on the financial aspect. Moreover, there are other close relations between organizational and technical aspects. In practice, it means that procedural abstraction can be transformed from the simple tree to the form of multidimensional labelled directed graph, defined as:

$$G = (V, E, \gamma) \quad (1)$$

where:

$V$  – set of nodes,

$E$  – set of edges,

$\gamma$  – function over nodes and edges.

In the classical approach to the procedural abstraction [20,28] this issue comes down to the principle, according to which any operation that reaches defined goal can be treated as a whole, regardless to the fact that the operation itself may actually consist of a sequence of lower level operations.

The abstraction can also be used to describe not only operation but also data. Data abstraction is based on their description in the operation categories. This idea is very strongly used in the object-oriented programming paradigm, where, in contrast to the procedural paradigm, a strong connection of structures (classes) with operations (methods) that can be performed on the instances of this structures (objects) has occurred.

## 2.2 Encapsulation

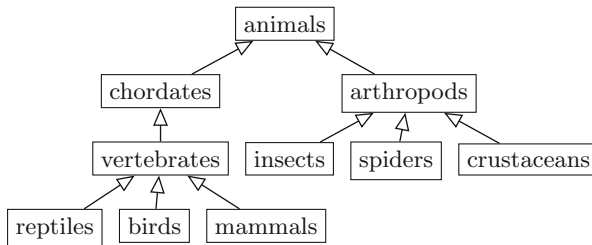
It is the concept that is based on the principle of minimization of information available outside. An element makes the information available through interfaces, that is set methods of communication [12,25,30]. This method of reality description ensures that the surrounding “knows” only what is absolutely necessary and subsequently there is no access to inner information of encapsulated element. It is very important from the point of view of consistency, because it protects encapsulated element from the unauthorised access that can involve reading or modifying the value of this element. In object-oriented encapsulated modelling the element is a class, where by default all components have private visibility range. As a result, all that has not been explicitly declared as public may be modified only by the operations defined in this class. This protects items of a particular class from the modification of attributes or from performing an operation that is reserved only for the inner purposes of items of a particular class. Thus, the encapsulated element has private (hidden) and public (accessible from the outside) part, whereby the principle of encapsulation implies that only those elements that need to be public are public, in contrast to others, in particular procedural approach, where there has been no mechanism protecting the structure of data and functions against misuse.

From the point of view of artificial intelligence the example of encapsulation may be the abstract model of reality called the Chinese Room [8]. In this approach encapsulation is based on hiding grammatical rules and communicational language semantics, in contrast to the object-oriented model, where data and operations are considered.

### 2.3 Inheritance

It is the concept of reality description focused on the relation between generalization and specialization. This description comes down to the creation of taxonomic trees (Fig. 2), in which the superior elements (base elements) describe the common set of features, properties, operations for all derivative elements (inherited elements). This mechanism is highly important and commonly used to the description of phenomena, both in information technology and in other fields. There are many relations between this and other mechanisms. For instance, in object-oriented modeling the idea of inheritance is closely related to such issues as encapsulation, visibility ranges and interfaces, resulting in coherent description (modeling) system that has a wide range of expression.

Inheritance has several aspects. One of them is the problem of multiple inheritance that is based on the possibility of inheriting from more than one base element. The result is that the classical generalization tree becomes a directed graph (acyclic graph).



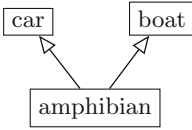
**Fig. 2.** Exemplary animals taxonomy tree

Multiple inheritance is a strong description tool, because it enables the simple and quite intuitive method of presenting a situation, in which a particular element constitutes a kind of hybrid of two or more superior elements, regardless of whether it refers to structures, operations or other elements that are subject to inheritance (Fig. 3).

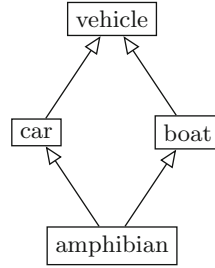
The issue of inheritance involves the problem of virtuality that protects against unauthorised duplication of elements or of different classes of components that are in fact the same element. It is particularly important in the inheritance structures called diamonds or their derivatives (Fig. 4).

### 2.4 Connotation

The description of reality through the associations is based on creating relations between elements that in some aspect or dimension have common or similar features, properties or functionalities [2, 7]. Such relationships are ambiguous or unnamed and they do not specify precisely the nature of this relation but rather loose connections between elements. It is also restricted to the specified aspect.



**Fig. 3.** An example of multiple inheritance



**Fig. 4.** An example of diamond structure in a inheritance graph

For instance, some events may be associated in the dimension of time, what would suggest that they can occur in e.g. in the close distance of time from each other, they can constitute consequents or precedents to some other events. As a result, associations are inherent to the concept of similarity, however, they do not imply the necessity of a strict definition of metrics but only to more or less set out the proximity in a particular dimension or space. Other more complex aspect of similarity that constitutes the basis for associations is for instance the structure. The structure, as the system composed of elements and relations between them. The classic example of associations related to the aspect of structure is the term “tree” that is associated with many concepts including the concept of a physical vegetal object and the abstract concept of data structure that derives from the graphs theory. Completely independently, but close to that concept is the concept of e.g. decision tree used in expert systems of artificial intelligence. Analogically, we can list directory trees, trees as organizational structures in management sciences, taxonomic trees, trees defining inheritance etc. In this case, the combining (associating) elements are the features of structure referred to as the tree, which here is an incoherent directed acyclic graph.

**2.5 Scale**

The method of description based on the scale relates on the idea to look at the described model from a distance that causes some details to be invisible [29]. This idea is quite natural, however it is highly important not only for the description of reality in the abstract sense, e.g. knowledge base, but also it constitutes a good illustration for the method known in the modelling of 3D scenes named “elision”. It is based on optimization of number of displayed polygons (the smallest element of which 3D scene is build). This optimization is focused on the reduction of polygons, that is details of displayed items, in the function of distance of element and observer. We can imagine a very complex scene composed of a very large number of complicated solids that additionally change their location in time creating a dynamic scene. The example of such scene can be a dynamic 3D image presenting an epic battle, which involves 10 000 warriors, each composed of thousands elements that are polygons. If the

observer is able to see the whole scene, it raises the question if it would be wise to overload a processor by the necessity of calculation the location and other parameters for each of this billiards of polygons. Taking into account that by transforming this 3D scene to the form of projection onto a plane, individual warriors will cover only a few pixels. Thus, the number of processing elements of a 3D scene depends on the distance from observer, so the processor is not overloaded with the necessity perform calculations on details that will not even have a chance to appear in the form of visual effect.

And this is a key to the idea of reality description based on the scale, where by presenting any model, some of its details needs to be hidden and which of them will be hidden, and specifically to which level of significance the element is visible, depends on the scale at which the model is presented. It has extremely significant for the ability to perceive the model and its effective analysis. Entering into details on every level of zoom by the observer will cause the necessity to take into account too many details. This is why, there are several levels of model granularity. From the detailed model the elements can be segregated so that the model starts to be “light” and easy to analyze and comprehend by a person or persons working on it.

The idea of scale is commonly used for a long time during the use of all kinds of maps. There are general maps, where placing of details would obscure the image, if at all it was possible because of their richness. However, the detailed maps can and should consist of possibly a large amount of information, so that the change of scale is not a simple operation based on mathematical homothetic transformation, and at the same time provide new information hidden on the higher levels of scale [14]. The scale can relate to many aspects, e.g. it can be connected with the relation between a part and a whole. For example a technical drawing, where some of the parts can be visible and some of them hidden and what decides about it is the scale of such drawing.

It can also relate to more abstract relations, e.g. generalization-specialization, where while considering for instance taxonomy, we can focus on the upper part of a tree going down to the direction of leaves that are in fact elements that in a specific scale are seen as the last, the darkest and the most detailed. In this case the scale means moving inside the tree of taxonomy.

### 3 Complex Methods of Conceptualization

#### 3.1 Object-Oriented Modeling

Object-oriented modelling [21] is an example of idea, in which several methods of modelling were combined, including a few presented above. Therefore, the description of features of such modelling focuses only on its selected elements, in order to avoid redundancy of description. One of the more important concepts of object-oriented modelling is class-object relation. The class is a description of a construction and the way of creating objects, whereas the object itself constitutes the “physical” emanation of class, as its instance. It is necessary to emphasize that the object is not a part of the class and it is able to store information

in attributes and to interact with other objects through methods. In this approach methods are part of the class, that is they are defined in it, whereas they are induced for objects. In a specific case, the object can have an interaction through the method with itself; then it refers to the methods induced for and inducing object. In the natural language it is the equivalent of inducing operation described as the reflexive verb. (an example: washing yourself) Another already mentioned concept are the attributes. Attributes are defined in the class but the ability to assign them the values is reserved for the objects. Due to that, all objects of a particular class have the same set of attributes but each of the objects can consist of other set of values assigned to these attributes. A special case of attribute is static attribute, where it is defined and stored on the level of the class and all objects of a particular class referring to this attribute, refer to one and the same element. Object-oriented modelling does not imply aggregates that have the ability to store objects, that is in the same idea there is no separate category with such function. In practice, it means that each implementation can, according to its rules, create sets, lists, vectors, bags and multisets, however, as such, they are not the element of object-oriented modelling.

One of the basic method of object-oriented modelling is the concept of association. Association is the connection that links two classes. This relation is unnamed, which means that its semantic is not predetermined, in contrast to e.g. composition or generalization. Association has some other features, such as navigation, multiplicity of association or roles, however these aspect will not be discussed in detail in this article.

The object-oriented model despite its size and generality has restrictions and does not include all possible concepts of description of world complexity.

Considering the restrictions, we should notice that in many cases they rather do not result from the model itself but from its implementation. For instance, already mentioned association normally is realized by the attributes. The result is that the relation of association automatically became purely conceptual, that is it does not consist of its own separate categories or their instances. Simply put, in languages using object-oriented paradigm usually it is impossible to create an entity that would be an instance of association. The special case of association are  $n$ -ary association, for  $n \neq 2$ . Each of the roles of this association consist of multiplicity, but only on the side of a class. As a result, it is possible to limit the number of objects that take part in association, but it is impossible to determine the number of association, in which the object of a particular case can take a part. This stand in a sharp contrast to the binary association, where on both sides there are multiplicities, which means that ternary associations and higher are a special case that should be consider separately in relation to the binary associations. It is inconsistency that goes much further, because in the model unary relations were not included, although they are useful in the modelling.

By reviewing the existing methods of the world complexity and researching their mutual relations, way of mutual complementation, redundancy, strengths and weaknesses, what was developed is the skeletal system of a knowledgebase which in the possibly most general way is able to store the information of the

possibly most general character. This system is hybrid both from the point of view the conception of complexity description and in respect of the character of data that are stored there. It connects many simple and complex ideas, the latter of which are often extended versions of their primary concept.

### 3.2 Ontology

Ontology in the sense of science is the area known to philosophy and it is defined as “as it is”. It means that ontology consists of information about the state of things (state of the world), entities and relations between them. From the point of view of information technology, it can be determined that ontology is focused on entities and relations between them, not, however, on the answer to questions why is it like this or how does it happen, etc. The most popular definition of ontology in the information technology is Tom Gruber’s phrase saying that ontology is “*specification of conceptualization*” [9]. This definition is elegant because of its simplicity and generality, however it does not explain too much. From the moment, when the concept of ontology in information technology became commercialized, it started to be identified with the particular solution, and even with implementation of this solution in the form of OWL<sup>1</sup> and its derivatives. Ontology in this sense has lost on its generality and moved away from that idea, indicated in 1994 by Tom Gruber. Moreover, many studies present the specific implementations based on e.g. taxonomy, pompously called ontology of a particular problem. Sometimes it is even called ontological base, which is a terminological mistake, because ontology itself constitutes a structured set of information having a specific structure [5].

Aside from the issues of terminology, ontology is undoubtedly one of the most important methods of world complexity description. It should, however, be emphasised that it surely does not exhaust all the aspects of this description. In particular, ontology is not intended for storing the rules and facts beyond the entities and relations between them. Ontologies are usually presented as sets of types (classes), objects (instances) and relations (associations), which can be connections of any type. In practice, solutions based on ontologies very often make a predefined assumption, that is predetermined number of types (classes) and kinds of relations. Restriction that is particularly strong is the determination of kinds of relations, usually for generalization-specialization, part-whole and other for special uses. Lack of possibility to freely define relations is particularly burdensome for a modelling person and it significantly narrows the possibilities of a specific system.

### 3.3 Semantic Networks

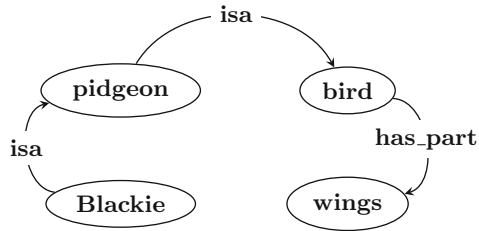
Semantic networks [4] are the method of complexity description based of the graph theory. In semantic networks there are nodes and edges, whereas both nodes

---

<sup>1</sup> Web Ontology Language – a family of knowledge representation languages endorsed by the World Wide Web Consortium (W3C).



and edges consist of labels (Fig. 5). What is particularly important for the semantic networks, is the fact that they do not have predetermined semantics. It is their fault, but also a huge advantage. It can be a fault because when seeing any defined network without the description of its semantics, it is impossible to clearly determine the meaning of nodes and arches. It means that by creating any semantic network, it is necessary to precisely determine the semantics of arches and nodes and possible grammatical structure accepted by these networks. It is at the same time an advantage, because it proves the generality of this method and allows to use them for the description of incredibly wide range of problems: from simple arithmetical equations, which can be easily represented by such networks, to complicated rules or facts that apply to the artificial intelligence systems. The simplicity of a structure of such network, that is nodes, edges, their labels and a function determining which of the nodes are connected with which and with which arches, allows to express the complexity of the world. However, the classical semantic networks also have some disabilities [17].



**Fig. 5.** An exemplary semantic network

They are, for instance, unable to take into account some of the details important for the description of reality. Many problems relating to the semantic networks have been solved in the Hendrix's studies over 50 years ago [11]. Unfortunately, the literature review proves that incredibly small number of authors describing the networks include the solutions proposed in this studies, referred to as the partitioned networks. The mechanism described by Hendrix is strong and it ensures the hierarchic structure of a network. Still, there are some issues left that in the classical networks and in networks extended by Hendrix are not included. An example of which is a mechanism of associations multiplicity, which is not present in the classical semantic networks. The problem relating to associations multiplicity is practically insolvable on the level of graphs and only after introducing the structures known as hypergraphs<sup>2</sup> to the description of semantic network structure, the problem is possible to be modelled. Semantic networks in the classical approach consist of more important restrictions, which, however, are not the subject of interest of this article.

<sup>2</sup> Hypergraph's edges are called hyperedges; they can be incident to any number of vertices.

### 3.4 Natural and Artificial Languages

The most natural way of complexity description is the natural language [22–24]. Its undeniable feature is its incredible flexibility allowing to describe almost every idea. It is, however, burden with a very difficult and, surprisingly unnatural for a human, method of recording. It is one dimensional recording that is linear and strongly marked with cultural connotations and connected with the set of terms and concepts being at the authors disposal for a specific statement in the natural language. It is a huge obstacle to the analysis of a text, because the only way for its acquisition is to combine terms as sequences that needs to be in accordance with a set grammar of a natural language. Already mentioned unidimensionality excludes the possibility to move freely on the record. The reality described is usually multidimensional and multifaceted and can include many levels, points of view and transformation of very complex information to the form of a sequence of terms. This needs to cause, first of all, difficulties, second of all, illegibility, and the third, a huge effort of the receiver to transform this sequence back to the complex structures that are functional for our minds. This is why, the natural language has been displaced from the technical sciences, especially from information technology, where two- or three-dimensional structures are dominant, also constituting the (often graphic) representation of a concept, idea or basically the complexity of the world. An example for this could be dozens of diagram types used for the description of equivalent issues. What is used within the modeling of information systems is for instance UML, which alone consists of several number of diagrams. Additionally, there are many diagrams in other sciences, such as BPMN [1], which is widely used in the combination of information and management science.

Formal languages [18] have been created for a very specific needs. Currently, the primary most universal formal language is language of logic, on which other fields of mathematics were based. Formal languages, e.g. programming languages, consist of very strict grammar and semantics, what differs them from the natural language. They are so unambiguous that it is possible to design machines able to communicate in such way. It is, however, burdened with illegibility for an average person and often with a considerable complexity of records. It should be emphasized that formal languages do not necessary need to be based on the text. Incredibly efficient way of communication are already mentioned formal languages based on the graphic symbols. Automatic processing of records in such language can be realize by appropriate algorithms, what is another advantage of this languages. Examples of this include automated reasoning systems, which perform symbolic operations basing on simple but incredibly effective methods of processing information.

## 4 Semantic Knowledge Base

Here, authors would like to introduce a term: *semantic knowledge base*. It is so crucial issue that all discussions on the databases and knowledge bases start and often end with a question: “*what do we mean by the term knowledge base*”

or database?”. Contrary to appearances, these terms are understood differently depending on the areas of studies of the researchers. The terminological order is essential, not only because of the methodological bases of the science but also for the explanation of solutions proposed by the authors.

#### 4.1 Database

Database is understood as the basic, lowest in the hierarchy of information storage and processing element, from which other levels are built. The definition of the database depends on the domain of the area interested in databases and will not be analyzed herein. Usually the term **database** is understood as stated in Definition 1 or in Definition 2.

**Definition 1.** *One or more large structured sets of persistent data, usually associated with software to update and query the data. A simple database might be a single file containing many records, each of which contains the same set of fields where each field is a certain fixed width. A database is one component of a database management system.*

**Definition 2.** *Ordered (having a specific structure) set of information logically connected with each other, which are intended to mirror the fragment of reality. Database should enable the storage of information in a permanent and coherent way that enables access to them (to read, add, delete, modify) anytime in a synchronous way.*

Database should comply with the following requirements:

- to guarantee data integrity,
- to ensure the effective data processing,
- to correctly mirror the relations in the real world represented by a database,
- to protect from the unauthorised access,
- to ensure synchronous access to date to multiple users,
- to make metadata (information about data structure) available.

From the point of view of definition, what is discussed here is not a particular database model: *hierarchical, network, relational, object-oriented, association-oriented*. These are only the frames setting the principles of creating categories and relations between them, from which particular bases are constructed. Database consists of a structure. The database structure is defined as a set of elements belonging to the database category together with their mutual relations. For instance, in relational model the structure constitutes a set of tables and relations between this tables. On different levels of structure representation (conceptual, logical, physical), this structure can have different form, degree of precision – generally speaking representation. However, what is really important is the fact that it does not have an established and included in the structure semantics. It means that structure of a database is a structure and semantics describing the meaning of particular elements and relation of these elements

is something that should be added in a form of verbal description, and then implemented in a form of application processing data in accordance with a particular structure. Here, it is important to draw attention to the next term, that is database consistency. The term database consistency [19,26,32] is described by the concept verifying whether a database is consistent with a model, that is with a simplified image of reality, which is being represented. The term was mentioned mainly because its relation with the concept of semantics. It is impossible to discuss consistency with a simplified image of the real world and its representation in the form of a structure and limitations without the knowledge of importance of particular elements of this structure. Semantics combines structural elements (syntax or specific structure within a specific syntax) with the outside world, specifically with the understanding (perception) of this world by a modelling person. What is important in the above considerations is that a database in a sense of structure can exist individually and be correct, and the semantics in database is not integrally related with a database itself. In practice, however, it is impossible to image a situation of creating a database structure in isolation of semantic aspects. It is possible only in theory without data, which refers to attributes, combining them together into wholes called in the relational model relations and in object-oriented model classes, creating relations between tables and classes, where all elements function without any relation to anything.

## 4.2 Knowledge Base

Database can have its structure that can be referred to as a syntax. To this structure it is possible to include semantics, that is determine relations between syntax and what it represents. The next stage and level of development is set of methods, algorithms. Such functions include information about the way of processing data. They can be relatively simple, elementary, e.g. function searching for data that meet specified criteria on the structure and data. However, methods (algorithms) that are being created, in their complexity and level of difficulty are beyond the simple operations on data, e.g. methods of logical inference based on contents of a database. It raises the question that is both philosophical and very practical, from which moment we can or we should say that the database (structure+semantics) consisting of more or less complex function implementing specific tasks can be called a knowledge base. The knowledge base is usually described as a set of information (database) with the ability to interpret those information. Similarly to functions of searching for information, e.g. according to value of the attribute, is hard to call knowledge and to classify the whole system as the knowledge base. However in the case of information system operating on information and consisting of implemented complex algorithms, the inferred situation is no longer unequivocal. Here, the authors wish to emphasize a certain imprecision associated with a blurry statement of “ability of interpretation”. It is possible to list a number of examples that are completely analogical to this problem, including Turing’s criteria or Chinese Room. In both cases reasonable doubts appear concerning the abilities to “intelligent” processing of information.

In conclusion, the authors would like to propose a certain approach to the issue related to clarification of the concept under the term of knowledge base.

Authors propose the following definition of the knowledge base.

**Definition 3.** *Knowledge base (KB) is a system consisting of four elements:*

1. *metastructural database,*
2. *database semantics,*
3. *primary information,*
4. *primary methods of data processing.*

*It is proposed to distinguish the following features of KB:*

1. *the possibility of freely defining the structure of knowledge,*
2. *the possibility to introduce knowledge,*
3. *the possibility to generate questions to the knowledge base,*
4. *the automatic processing of knowledge.*

The term metastructure should be understood as database structure that is constructed in a way allowing storage of information on the knowledge structure, and only through it the knowledge itself can be stored.

For instance, in a database we can put information of the employee; in a relation model it would be a relation consisting of attributes describing this employee and relationships with other relations, thanks to which it is possible to store information on e.g. his history of employment. It is the data structure. Whereas metastructural database would have the ability to define such type as an employee. This means that it would have the ability to freely, dynamically determine the attributes, which can be possessed by such entity and relations with other entities, both relating to their quantity and other features. Due to that, in a dynamical way, that is the way that does not interfere with a database structure. It is possible to introduce new entities and in the case knowledge base we would refer to terms and their definitions. It is the most important structural element differing the database form the knowledge base. It is, however, not the only difference. Database also can and even should consist of semantics. In the case of knowledge base the semantics is also present, but on the metastructural level. Another important remark is that in theory database can exist without semantics connecting it to the world to be described.

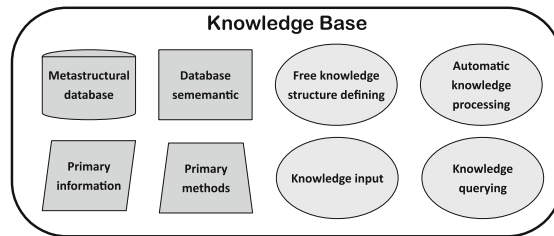
Database does not have to consist of primary data understood as information that is equally important as the structure itself, without which data interpreting and processing would be impossible. In the case of knowledge base, primary data e.g. in the form of primary terms definitions, primary relations (generalization-specialization, part-whole) typical for the knowledge bases, are absolutely essential for the knowledge base to be filled with knowledge.

### 4.3 Methods of Data Processing

A database does not have to consists of any methods of data processing embedded into its structure. It is a kind of database usage method, the application part

that depends on database but is not essential for its existence. The proposed definition, in contrast to unclear, blurry and very general definition describing the knowledge base as a database able to make use of that data, ensures the possibly clear and precise separation of what is a database and what can be called a knowledge base. This is an important issue because there are many publications, in which these terms appear in an unauthorized way, sometimes extremely incorrectly. The example for this include publications, in which non-structuralized sets in the form of text are called knowledge bases. In reality, they are simple repositories that do not even meet the requirements to be treated as knowledge bases. Sets of answers for essential questions are often called knowledge bases, mainly due to the fact that they are sets of data and they refer to some knowledge of a particular field. However, they are not knowledge bases. The classic example are services such as Wikipedia that, as we all know, is an extremely simple technology based on a simple text with references, called hyperlinks. Undoubtedly, it consists of some part of a human knowledge, but it does not make it a knowledge base. It is hard to even imagine that it could be the basis, on which any knowledge base could be built on. The terminological problem can be considered secondary, if not for the methodological issue related to the creation of knowledge bases, and consequently establishing of essential requirements and components of such system.

In the simple definition of the structure of knowledge we should refer to the abolition of restrictions on those that are not a domain of the database. It means that increasing the level of abstraction and entering the metastructure level, where the knowledge structure is possibly optional. The restriction concerns only the metastructure that has the feature of large generality. The possibility to introduce knowledge is analogical to features without data and is based on introducing mechanisms of putting knowledge on the specific structures, and consequently it is related to the mechanisms of structural and semantic correctness control for the particular knowledge base.



**Fig. 6.** Knowledge Base construction

The language of questions is also an analogical solution to the language of questions in knowledge bases. It is to ensure the possibility of acquiring knowledge both on the elementary level (such as in databases, where there are mechanisms to read information recorded before) and on the higher level, that is using

mechanisms such as acquiring knowledge not only recorded, but also knowledge based on the already acquired knowledge. The example of it can be a mechanism that is able to extract all attributes of a given term taking into account its negative features, but also all features that it has obtained as a result of using inheritance mechanism. It is a simple mechanism, however, it already requires searching the knowledge base and creating more complex semantic response. In databases, such mechanism is not required (Fig. 6).

Automatic processing of knowledge is a definite development of question mechanism. Similarly to the questions mechanism it required the ability to extract data on the basis of information included and more complex information (derived), the mechanism of automatic processing of knowledge relates to much more complex and subtle methods and its purpose is to modify the content of a database. The example of simple knowledge base mechanisms are methods of automatic detection of contradictories and methods of simplifying knowledge, that is conversion from one structure to another in order to optimize efficiency. There are the examples of elementary methods, which does not mean that there are simple to implement, and the level of complexity strictly depends on the level of knowledge base complexity.

#### 4.4 An Example of Knowledge Base

This chapter will present an example of the knowledge base that meets the predefined criteria. At the beginning one more terminological issue needs to be discussed. The developed knowledge base has been named *Semantic Knowledge Base* while in previous chapter it was noted that one of the elements to determine the system as a knowledge base is defined semantics of its database meta structure. Experience shows, however, that very often the term *knowledge base* determines standard data sets, as mention before. This phenomenon is so common that it has become customary, and even though incorrect to fight with it is virtually impossible. Therefore, to distinguish between the knowledge base, understood as different sets of information gathered by people, and the true knowledge base, i.e. one that meets the predefined criteria, the latter will be named *semantic knowledge base*. The term knowledge base is supplemented by “semantic” to clearly highlight the difference between sets of information, that do not have references to the meanings of elements stored in them, and the base, which is focused on the ability to define structures describing these meanings.

As part of research in the field of knowledge engineering *Semantic Knowledge Base* (SKB) system has been developed. It is framework system, i.e. it is not dedicated to a domain-specific applications. This means that it is able to store both, common and specialized knowledge. The common sense knowledge is used to define and to precise more specialized one. SKB is modular and uses logical modules, i.e. there is no strict separation between modules. This is due to the fact that each of the modules cooperates with the other according to the characteristic of the knowledge being processed at the moment.

AODB is a database metamodel designed by authors i.e. for the purpose of knowledge base implementation. It is considered to be the novel database

solution, that stems its conceptions from Entity-Relationship approach (E-R) and object-oriented paradigm, representing the database schema as a data and relationship constraint graph whereas the data model is the hypergraph. The syntax and semantics of the metamodel has been deeply elaborated in [15–17]. The two main intensional categories of the metamodel are association and collection. Collection is equivalent to class and is used to store and define objects, and the association is a category of elements that form the relations described by roles. The structure and tasks of the SKB *Structural Module* will be presented as to fulfill the definition of knowledge base in the sense of its metastructure, semantics, as well as, primary information it stores. The description of other modules is not significant since, it would only bring more prove on the same matter, while that description is not the key factor for this paper.

The *Structural Module* consist of two sub-modules: *Ontological Core* (Fig. 7) and *Relationship Module* (Fig. 8). The *Ontological Core* derives its name from the ontology concept, namely the key ontology elements. The most original and structurally situated in the center is an abstract collection CONCEPT. The abstract type is important, since authors assumed that there are no concepts that could not be classified into one of predefined type of concepts. Following collections inherit from an abstract CONCEPT collection: CLASS, INSTANCE, FEATURE, VALUE, VALUESPEC, RELATIONSHIP, SET, which are in turn:

- CLASS – represents class in object-oriented model, such as an animal, a person, a feeling.
- INSTANCE – represents object in object-oriented model, such as Mickey Mouse, Barrack Obama, i.e. instance that describes a particular element defined by specific class. SKB provides the solution in which it is possible to assign multiple classes to a single instance. Authors do not favor such a solution, however, it was decided to allow the implementation and usage of such a concept.
- FEATURE – used to define set of characteristics for a given concept. In the particular case, you can specify the characteristic of the class, what will mean that all instances of this class are described by such features. This mechanism is analogous to attributes assignment for the class in object-oriented model, but much more general, because it can describe the characteristics of any concept, such as relationships, collections, or even other features.
- VALUE and VALUESPEC – used to store and define the type of values.
- REALATIONSHIP – describes the relationships between concepts through predefined set of attributes as well as any additional features defined by FEATURE. It should be noted that the RELATIONSHIP collection do not represent specific relationships in terms of connections between concepts, but only hold its characteristics. The mechanism of building relations between concepts will be presented later in this paper.
- SET – used to describe sets of concepts.

The *Ontological Core* consists not only of collections, but also associations. These associations, according to AODB model, realize the following relationships.



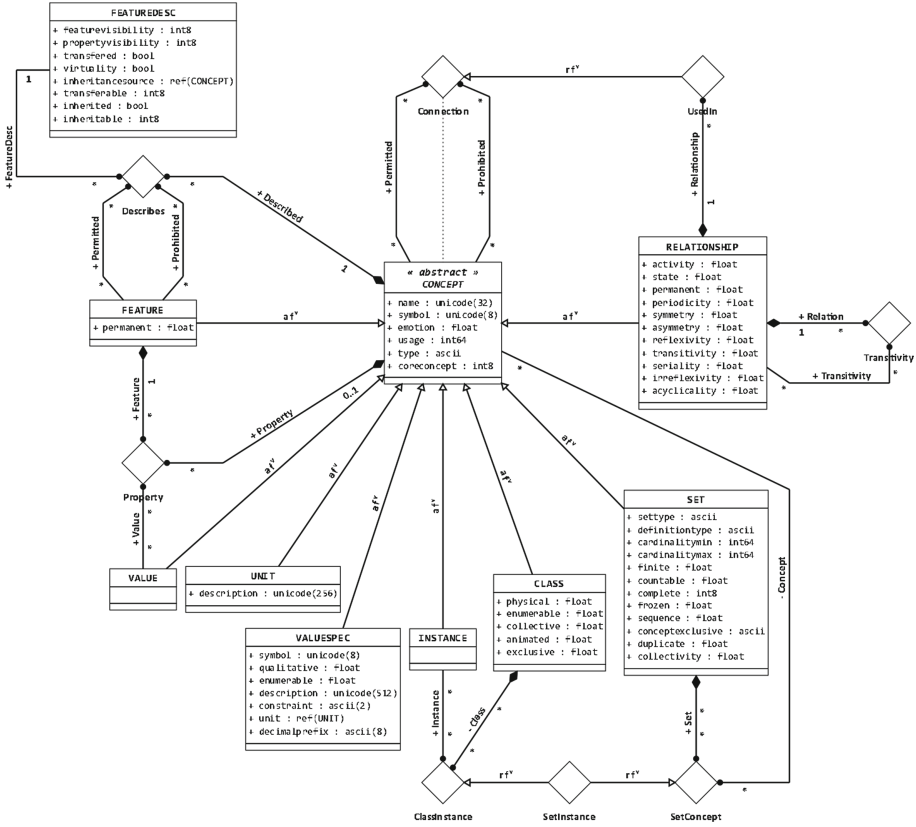


Fig. 7. The AML diagram of the SKB Ontological Core

**Property** – ties together the collections that are used to describe the properties of a particular concept, i.e. the value of the attribute (**FEATURE**). The role **Property** binds the concept with the property, **Feature** points to the feature being described and **Value** allows to assign value for the property.

**Connection** – allows to specify permitted and prohibited connections between concepts. The connection is understood as any possible junction of two concepts, that can be defined on the level of *Ontological Core* or *Relationship Module*.

**UsedIn** – specialization of **Connection** used to identify concepts that are permitted or prohibited to define within particular relation.

**Describes** – used to build lists of features permitted or prohibited to be used for the description of a concept.

**ClassInstance** – an association in which the instance is ascribed to specific class or classes.

**SetConcept**, **SetInstance** – used for building sets of concepts or sets of instances respectively.

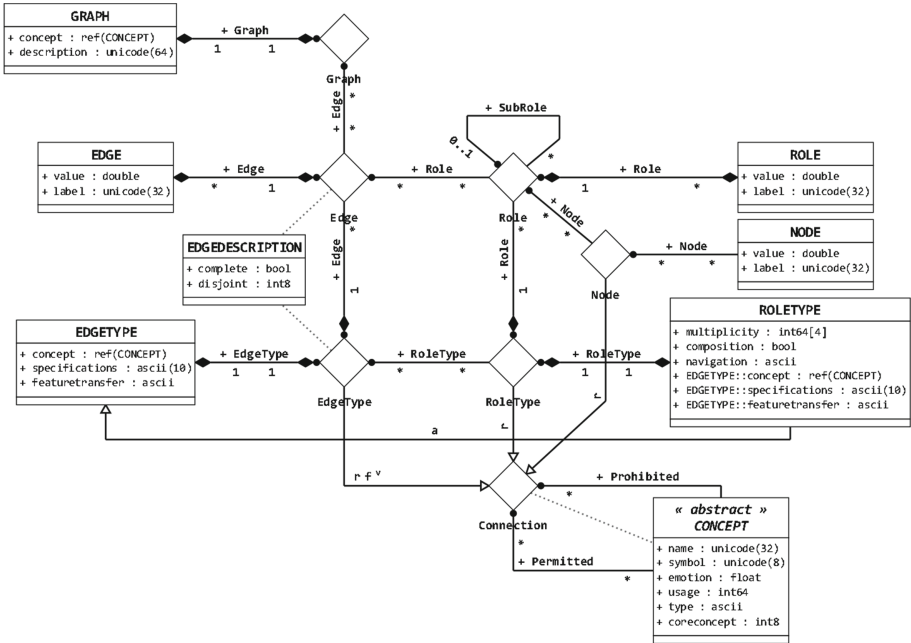


Fig. 8. The structure of Relationship Module

Presented *Ontological Core* module uses Gruber’s principles regarding ontology design, however it is not a full implementation of the ontology following this approach. This module is used to store the information about concepts and their properties as well as possibilities to establish relations between concepts. In order to create full-fledged ontology, i.e. that, which contains information about the relationships between concepts an *Relationship Module* has been introduced.

*Relationship Module* consist of two conceptual elements. The first is a system used for building relationship templates, defining type of relation and roles building it. It corresponds to the UML class diagram. The second one is used to determine the specific relationships between concepts defined in *Ontological Core*. It corresponds to object diagram in UML. Very important issue is that in the process of defining relations and roles only concepts previously defined in *Ontological Core* may be used. Given that *Ontological Core* module is capable of defining any terms, which later might become components of relations in the *Relationship Module*, it should be noted that it gives ability to create any possible relations. Therefore this module is not limited to standard relations, e.g. whole-part, generalization-specialization.

A description of each collection, and association that make up this module is omitted, as it would address the postulate that meeting has already been presented in the description of the *Ontological Core* sub-module. The presented diagram shows the structure of *Relationship Module*. It is worth noticing that

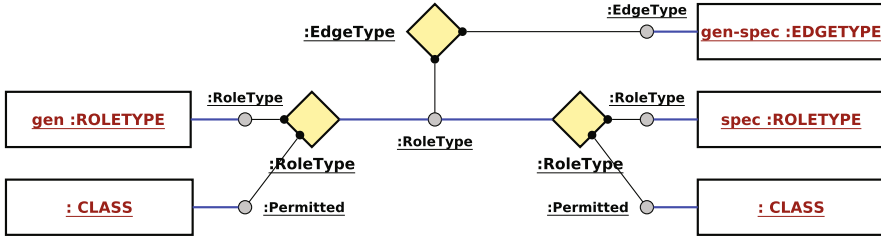


Fig. 9. The definition of generalization-specialization relationship

the relationships are built as hypergraphs, in contrast to classic links derived from object model. The association is treated as an edge, which in turn is made up of roles. The role, in general, is a tree-like structure, that may consist of any number of sub roles. In addition, for each role one can specify the types of terms that may be used by it. This means that at the end of each of the role or sub role may be any number, but at least one node representing the type of the concept. At the stage of defining the role, you can also specify: the multiplicity, the navigation, composition, as well as information regarding inheritance of the attributes of concepts taking part in it.

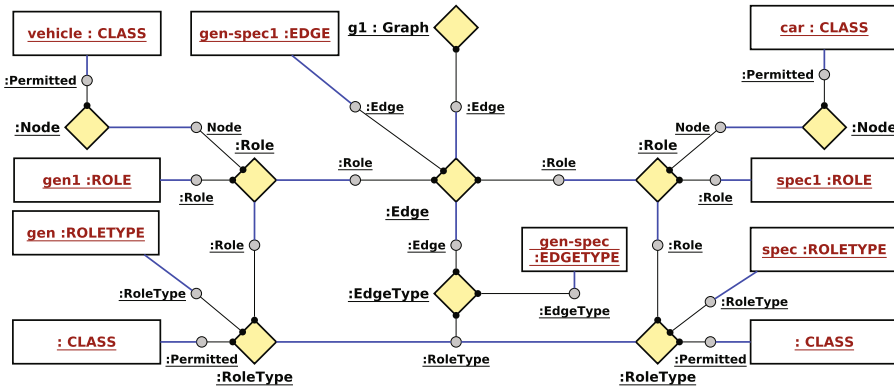


Fig. 10. The example of generalization-specialization relationship implementation

The Fig. 9 shows an example of a simple binary relationship. In the diagram, there is a definition of relationship presented, while in Fig. 10. concrete implementation of this relationship.

## 5 Summary

The aim of this study was to propose a definition of *Semantic Knowledge Base* in the context of nowadays used terminology and solutions in the field of databases and reality description. Presented methods used for description of reality were

not intended to develop discussion in this direction, but rather to briefly present the most common solutions. Each of these may be individually expanded to some extent and interpreted, as well as, implemented in many different ways. The list of solutions is not closed, as there may be a more specific solution or a hybrid one, combining together several of simple methods. Knowledge Bases, depending of on their applications, degree of generality or specialty and implementation may contain different combination of showed methods. This raises the important question about the nature of knowledge base. The paper pointed out that the term Knowledge Base is very often used to name any collection of information resulting from the accumulation of human knowledge. In particular, different repositories, such as sets of answers to frequently asked questions. It is a popular, commercial approach to naming, but has nothing to do with the scientific approach. The source of this is the simple fact that the term database has ceased to be a trendy and catchy carrier, at the moment knowledge is a term often used to catch interest of the recipient. The paper shows that these repositories of information do not have the structure and other attributes that allow them to qualify even as a database. Therefore they might not have nothing to do with the knowledge base. In defining the term of Knowledge Base authors come from the observation that the knowledge is the information along with the possibility of its use. This is an important assumption, however, very general and, unfortunately, very rarely respected. In order to clarify the criteria for determining whether a set of information is a knowledge base, four components and the four properties that the system must meet has been introduced. The authors believe that it is necessary to draw attention to the mass scale abuse of terminology in the field of knowledge engineering. This is very dynamic area and it can be assumed that its development will only accelerate. As a result, it should be a very clear distinction between advanced knowledge base systems solutions and extremely simple repositories of information. For this purpose, authors proposed to supplement the knowledge base term with “semantic” in front. Author is concerned that the term knowledge base has been so widespread that in practice it is impossible to separate legit knowledge base from commercial solutions reaching the marketing terminology.

This article is not supposed to be only a theoretical consideration of the terminology issues, but it also presents a solution namely, *Semantic Knowledge Base* system, developed by the authors. Due to the volume of the studies, the article presents only the main idea, and the most important characteristics of the modules, describing in general terms the most key modules constituting the system core. This system is fully defined by the syntax and semantics in terms of Association-Oriented Database model. AODB is a new model for database modeling, as a key and direct solution, which formed on the basis of the concept of semantic knowledge base. The main idea that joins those studies is the idea of associations being widely used in modeling. A detailed description of the AODB grammar and semantics, due to its size, will be the subject of monographic studies currently being prepared for printing.

## References

1. Aagesen, G., Krogstie, J.: BPMN 2.0 for modeling business processes. In: Handbook on Business Process Management 1, pp. 219–250. Springer, Heidelberg (2015)
2. Barnes, W.H.F.: The doctrine of connotation and denotation. *Mind* **54**, 254–263 (1945)
3. Brown, M.S.: Theaetetus: knowledge as continued learning. *J. Hist. Philos.* **7**(4), 359–379 (1969)
4. Collins, A.M., Quillian, M.R.: Retrieval time from semantic memory. *J. Verbal Learn. Verbal Behav.* **8**(2), 240–247 (1969)
5. Dudycz, H.: Approach to the conceptualization of an ontology of an early warning system. In: Information Systems in Management XI, Data Bases, Distant Learning, and Web Solutions Technologies, pp. 29–39 (2011)
6. Duhl, J., Damon, C.: A performance comparison of object and relational databases using the sun benchmark. In: ACM SIGPLAN Notices, vol. 23, pp. 153–163. ACM (1988)
7. Feng, S., Bose, R., Choi, Y.: Learning general connotation of words using graph-based algorithms. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1092–1103. Association for Computational Linguistics (2011)
8. French, R.M.: The chinese room: Just say “no!” In: Proceedings of the Cognitive Science Society, vol. 1 (2000)
9. Gruber, T.R.: A translation approach to portable ontology specifications. *Knowl. Acquisition* **5**(2), 199–220 (1993)
10. Hao, C.: Research on knowledge model for ontology-based knowledge base. In: 2011 International Conference on Business Computing and Global Informatization (BCGIN), pp. 397–399. IEEE (2011)
11. Hendrix, G.G.: Encoding knowledge in partitioned networks. In: Associative Networks: Representation and Use of Knowledge by Computers, pp. 51–92 (1979)
12. Joyce, D.: An identification and investigation of software design guidelines for using encapsulation units. *J. Syst. Softw.* **7**(4), 287–295 (1987)
13. Kalantari, R., Bryant, C.: Comparing the performance of object and object relational database systems on objects of varying complexity. In: Data Security and Security Data, pp. 72–83 (2012)
14. Korzynska, A., Zdunczuk, M.: Clustering as a method of image simplification. *Inf. Technol. Biomed.* **47**, 345 (2008)
15. Krótkiewicz, M.: Asocjacyjny metamodel baz danych. Definicja formalna oraz analiza porównawcza metamodeli baz danych (eng. Association-Oriented Database Metamodel). No. z. 444 in *Studia i Monografie, Oficyna Wydawnicza Politechniki Opolskiej*, Opole (2016)
16. Krótkiewicz, M.: Association-oriented database model - n-ary associations. *Int. J. Softw. Eng. Knowl. Eng.* **27**, 281 (2017)
17. Krótkiewicz, M., Wojtkiewicz, K., Jodłowiec, M., Pokuta, W.: Semantic knowledge base: quantifiers and multiplicity in extended semantic networks module. In: Knowledge Engineering and Semantic Web: 7th International Conference, KESW 2016, Prague, Czech Republic, 21–23 September 2016, Proceedings. Springer, Cham (2016)
18. Lange, K.J.: Complexity and structure in formal language theory. *Fundam. Inf.* **25**(3, 4), 327–352 (1996)

19. Lin, K.J.: Consistency issues in real-time database systems. In: Proceedings of the Twenty-Second Annual Hawaii International Conference on System Sciences, 1989. Vol. II: Software Track, vol. 2, pp. 654–661. IEEE (1989)
20. Macewen, G.H., Martin, T.P.: Abstraction hierarchies in top-down design. *J. Syst. Softw.* **2**(3), 213–224 (1981)
21. OMG: Unified Modeling Language™ (UML®) Version 2.5 (2013). <http://www.omg.org/spec/UML/2.5/www.omg.org/spec/UML/2.5/Beta2/PDF/>
22. Przepiórkowski, A.: Slavonic information extraction and partial parsing. In: Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies, pp. 1–10. Association for Computational Linguistics (2007)
23. Przepiórkowski, A., Górski, R.L., Lewandowska-Tomaszyk, B., Lazinski, M.: Towards the national corpus of polish. In: LREC (2008)
24. Przepiórkowski, A., Marcińczuk, M., Degórski, L.: Dealing with small, noisy and imbalanced data. In: Text, Speech and Dialogue, pp. 169–176. Springer, Heidelberg (2008)
25. Schärli, N., Black, A.P., Ducasse, S.: Object-oriented encapsulation for dynamically typed languages. In: ACM SIGPLAN Notices, vol. 39, pp. 130–149. ACM (2004)
26. Seligman, L.J., Kerschberg, L.: Knowledge-base/database consistency in a federated multidatabase environment. In: Proceedings of the Third International Workshop on Research Issues in Data Engineering: Interoperability in Multidatabase Systems, RIDE-IMS 1993, pp. 18–25. IEEE (1993)
27. Soutou, C.: Modeling relationships in object-relational databases. *Data Knowl. Eng.* **36**(1), 79–107 (2001)
28. Stroustrup, B.: What is object-oriented programming? *IEEE Softw.* **5**(3), 10–20 (1988)
29. Su, H., Bouridane, A., Crookes, D.: Scale adaptive complexity measure of 2D shapes. In: 18th International Conference on Pattern Recognition, ICPR 2006, vol. 2, pp. 134–137. IEEE (2006)
30. Voigt, J., Irwin, W., Churcher, N.: Class encapsulation and object encapsulation: an empirical study (2010)
31. Wislicki, J., Kuliberda, K., Adamus, R., Subieta, K.: Relational to object-oriented database wrapper solution in the data grid architecture with query optimisation issues. *Int. J. Bus. Process Integ. Manag.* **2**(1), 17–25 (2007)
32. Zhangbing, L., Wujiang, C.: A new algorithm for data consistency based on primary copy data queue control in distributed database. In: 2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN), pp. 207–210. IEEE (2011)