UDC 004.655.2:519.76(6.2+7.4)

# About language for data structures modeling

A. G. Zhytaruk, G. N. Zholtkevych
*V.N. Karazin Kharkiv National University,*
*4, Svobody Sqr., Kharkiv, 61077, Ukraine*

The pre-scheme theory for data models description and data structure model by means of concept samples is considered. A representation of samples by means of acyclic graph within of pre-scheme theory is suggested. Language for description of structural constraints for samples is offered. Method of their verification is developed.

***Key words:****.pre-scheme, acyclic graph, sample of concept, labeled tree, constrained sample, free sample, interoperability.*

Для описания моделей данных в работе предложена теория предсхем и образцы концептов как модель структуры данных. В рамках этой теории предложена структура данных – ациклический граф как способ задания образцов. Также предложен язык описания структурных ограничений для образцов и разработан метод проверки выполнимости такого рода ограничений.

***Ключевые слова:*** *предсхема, ациклический граф, образец концепта, маркированное дерево, образец со связями, свободный образец, интероперабельность.*

В роботі запропонована теорія предсхем для описання моделей даних та зразків концептів як моделей структур даних. В межах цієї теорії запропонована структура даних – ациклічний граф для визначення зразків концептів. Також запропонована мова описання структурних обмежень для зразків концептів та метод перевірки такого роду обмежень.

***Ключеві слова:*** *предсхема, ациклічний граф, зразок концепту, маркіроване дерево, зразок зі зв'язками, вільний зразок.*

## Introduction

One of the important requirements to modern information systems is ability to enhance their capabilities due to the use of the software components. This property of system is called interoperability [3].

There is a class of problems such as data exchange between applications or data management or storing semi-structured data and so on [2, 1]. All the tasks within each of these problems require a common semantic model.

In most cases, a data model and a data scheme are fixed at the design stage of the information system development. It is necessary to develop a high-level data model for providing semantic interoperability that allows carrying out data transformation. There are three classic models of data. They are hierarchical, network and relational.

A hierarchical data model is a data model in which the data is organized into a tree-like structure. Not tree-like structures of data lead to the problems of the creation of data model and data processing.

The network model is an improvement of a hierarchical model in which every record in network node has relationship with other nodes. This model allows to increase the time of data access but to decrease the time of change of the data model.

The relational model used the basic concept of a relation to provide a declarative method for specifying data and queries. However, it is not possible to explicitly describe the recursive structures. There are two levels of data representation according to relational model; they are data and metadata. A metadata must be static.

Thus, among the existent models of data there is not a common high-level data model providing possibility of data verification, management by metadata and tools of design of the known data structures.

A rigorous mathematical language of data modeling will be offered in this paper. The proposed language is simple and has large expressive possibilities.

**1 Preliminaries**

In this context, pre-schemes can be considered as models of data. The pre-schemes theory permits to describe the arbitrary structures of data but also has mechanisms of verification of data structures. A pre-scheme templates allow to describe the known structures of data such as lists, arrays etc.

**Definition 1.**

We shall say that the pre-scheme of subject domain is three unitary predicate $C(x)$, $R(x)$ and $Q(x)$, binary predicate $D(x,y)$ and triadic predicate $M(x,y,z)$ if the following conditions are satisfied:

1. $(\forall x,y)D(x,y) \to C(x) \wedge Q(y)$
2. $(\forall x,y,z)M(x,y,z) \to Q(x) \wedge R(y) \wedge C(z)$.

$C(x)$ is true if $x$ is a concept. $R(x)$ is true if $x$ is a role. $Q(x)$ is true if $x$ is a qualifier. $D(x,y)$ is true if $y$ is a qualifier of concept $x$. $M(x,y,z)$ is true if $y$ is a role from qualifier domain $x$ and concept $z$ is a value of role $y$.

Let $W$ be a set of names of subject domain. Every pre-scheme has to satisfy the following conditions:

1. Condition of names not intersection

$$(\forall x \in W)C(x) \to \neg(R(x) \vee Q(x)) \quad (1)$$

$$(\forall x \in W)R(x) \to \neg(C(x) \vee Q(x)) \quad (2)$$

$$(\forall x \in W)Q(x) \to \neg(C(x) \vee R(x)) \quad (3)$$

In other words, any name cannot be concurrently used as a name of a concept or a name of a role or a qualifier.

2. Completeness condition

$$(\forall x \in W)(C(x) \vee R(x) \vee Q(x)) \quad (4)$$

In other words, every name from subject domain is either a name of a concept or a name of a role or a name of a qualifier and nothing else.

3. Condition of qualifiers completeness

$$(\forall y)(\exists x,q,z) : Q(y) \to D(x,y) \wedge M(y,q,z) \quad (5)$$

In other words, every qualifier has a non-empty domain.

4. Condition of unambiguous roles definition

$$(\forall x)(\forall y,q,z)((q \neq z) \wedge \neg(M(x,y,q) \wedge M(x,y,z)) \quad (6)$$

5. Condition of unambiguous concepts definition

We introduce the following auxiliary predicate $T(x,y,z)$. It can be done as follows

$$T(x,y,z) \equiv C(x) \wedge R(y) \wedge C(z) \wedge (\exists u)(Q(u) \wedge D(x,u) \wedge M(u,y,z))$$

$$(\forall x,y)(T(x,y,z) \wedge T(x,y,z')) \to (z = z') \quad (7)$$

6. Condition of unambiguous qualifiers definition

$$(\forall x_1, x_2)((\forall y, z)(M(x_1, y, z) \leftrightarrow M(x_2, y, z)) \rightarrow (x_1 = x_2)) \quad (8)$$

**Definition 2.**

A concept x is called a basic concept if

$$C(x) \wedge (\forall y)\neg D(x, y) \quad (9)$$

The set of all basic concepts is denoted by $N_0$.

**Definition 3.**

Let sample be an acyclic graph with the following properties:

a) every non-leaf node is marked by the selected qualifier
b) every leaf node is marked by a concept from $N_0$
c) every edge is marked by a role name

## 2 Methods of pre-scheme definition

Description of pre-scheme by means of predicates (as discussed above) is a non-trivial task. Any pre-scheme can be described by means of the predicates. But this description is difficult for users without the special mathematical experience. Therefore, a task of development of simple language with the same expressive possibilities is actual.

### 2.1 Graphic Method

The following graphic notation for pre-schemes representation has been suggested in paper [4].

1. A circle represents a qualifier.
2. A rectangle represents a concept.
3. A rectangle and a circle are joined with each other by a line if:
    a. a qualifier is associated with the concept (straight line).
    b. a qualifier is associated with the concept (straight line).
    c. a concept is used inside a qualifier(arrowed line directed from qualifier to the concept).

Pre-scheme of polyline is a simple picture, as the following example shows.

**Example 1.**

*W = { polyline; segment; point; list; null; Real; head; tail; begin; end; nothing; x; y; broken; structure; empty; coordinates}*
*D(x; y) = { (polyline; broken); (list; broken); (list; empty); (segment; structure); (point; coordinates)}*
*M(x; y; z) = {(broken; head; segment); (broken; tail; list); (structure; begin; point); (structure; end; point); (coordinates; x;Real); (coordinates; y;Real); (empty; nothing; null)}*
*C(x) = {polyline; segment; point; list; null; Real}*
*R(x) = {head; tail; begin; end; nothing; x; y}*
*Q(x) ={broken; structure; empty; coordinates}*

An example of the pre-scheme is given in Figure 1.

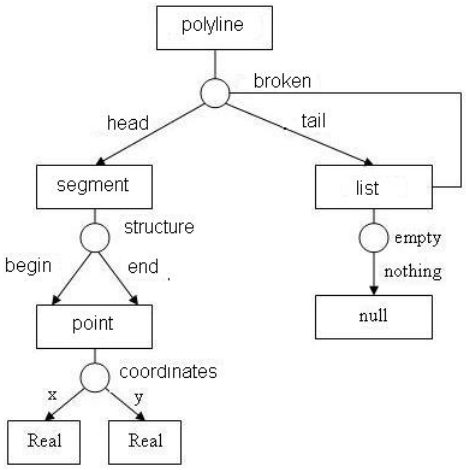



*Fig. 1. Graphic method of pre-schemes*

The graphic method allows good visual representation. However, there is a risk of construction of inconsistent logical model. Therefore, the task of verification of logical model is important. Algorithms of verification for logical model do not exist in case of a graphic method. Thus, graphic method is useless for real subject domains.

**2.2 Relation Method**

The method of presentation of pre-schemes with relational databases in the form of two relations has been proposed in paper [7]. This method allows describing some algorithms for pre-scheme verification. However, there is no pure relational query to identify a concept that has no sample. Thus, in general it is not possible to verify a given model by means of relational approach.

**2.3 SDL notation**

The method of presentation of pre-schemes with SDL notation has been proposed in paper [9]. Let us define SDL notation.

SDL notation is a text document containing a description of the pre-scheme. The pre-scheme is a set of items.

The item can have the following forms:

1. [***define*** *identifier (selectors)* ] defines a complex qualifier where
   (a) ***define*** is a keyword.
   (b) *identifier* is an ID of the qualifier.
   (c) *selectors* is a set of selectors.
2. [*identifier = definitions*] defines a concept and associated qualifiers where
   (a) *identifier* is a concept identifier.
   (b) *definitions* is a set of concept qualifiers.
3. [*identifier* ***is atomic***] defines a basic concept where
   (a) *identifier* is a concept identifier .
   (b) ***is atomic*** is a keyword.

The qualifier can have the following forms:

4. [*identifier (selectors)*] defines a simple qualifier where
   (a) *identifier* is an ID of the qualifier,

(b) *selectors* is a set of selectors.

3. [*identifier (ID qualifier)*] is used if qualifier is already defined.

*Selectors* is a sequence of selectors separated by commas. Each selector is [*identifier role : identifier concept* ].

The description of pre-scheme from Example 1 rewritten in terms of SDL notation is given in Figure 2.

Pattern structures such as an array or a list can be described using SDL- notation. An advantage of SDL-notation is its extensibility. It allows adding new elements for pre-scheme description. Hence, there are at least three approaches for pre-scheme definition. SDL notation is a powerful and expressive method for describing pre-schemes.

```
define broken (tail: list, head: segment)
define structure (begin: point, end : point)
define coordinates(x : Real, y : Real)
polyline = broken
list = broken; empty(nothing : null)
segment = structure
point = coordinates
Real is_atomic
null is_atomic
```

*Fig. 2. SDL notation for a pre-scheme of the polyline.*

**3 A review of data structure model descriptions**

As mentioned above, pre-scheme represents the data model. Then concept samples are the data structure models. Below there are several methods for sample description.

**3.1 Labeled trees**

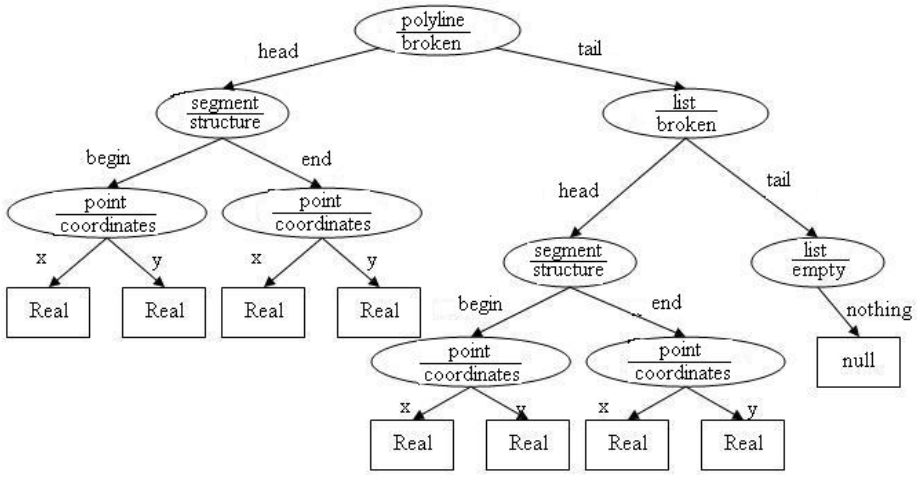
The method of presentation of the sample with labeled trees has been proposed in paper [5]. For example, the labeled tree of pre-scheme sample from Example 1 is given in Figure 3.

However, the set of all pre-scheme samples containing one or more recursively defined concepts can be cumbersome. This is a considerable disadvantage regarding problems of information storing and processing. It is necessary to reduce the set of all samples to a set of samples allowed in the semantics of a subject domain.

Polyline sample is the set of all samples that comply with the following condition.

*Every end of the segment has to coincide with the beginning of the next segment. (10)*

It leads to the concatenation of two nodes into one in a labeled tree. As a result, the data structure obtained will contradict to the definition of a labeled tree. Thus, labeled trees cannot store samples with structural constraints. Therefore, it is necessary to improve the sample description theory so it can take that kind of constraints into account.




*. Fig. 3. A sample of the polyline concept as a labeled tree.*

**3.2 Acyclic graph**

We now define free samples to be a labeled tree and constrained samples to be a free sample satisfying some structure constraints. The model of a constrained sample for a pre-scheme is an acyclic graph where unique id corresponds to each vertex, denoted by the qualifier name from $Q(x)$.

The sample of the polyline, satisfying 10 presented as an acyclic graph is shown in Figure 4.

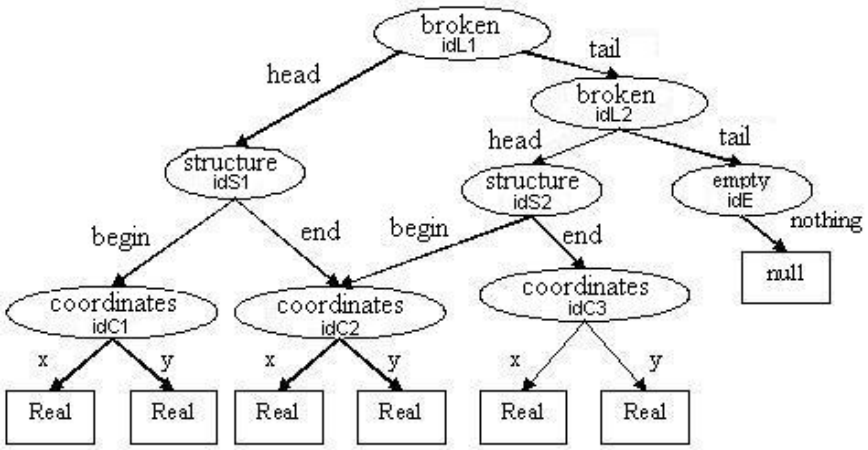



*Fig. 4. An acyclic graph of a polyline concept.*

We can see from Figure 4 that constraint 10 corresponds to merging of two nodes into one marked by a tuple *(coordinates; idC2)*.

So, we will use labeled trees for presenting free samples and acyclic graphs for presenting constrained samples.

**4 Method of verification for constrained samples of concept.**

As mentioned above, free samples are labeled trees and constrained samples are acyclic graphs. We now define language for description of structure constraints. It is based on a graph path.

**Definition 4.**

$P$ is a path of free sample such that

$$P \to QT$$
$$T \to \varepsilon \mid .R : QT$$

where £ - empty, $Q$ - name of qualifier and $R$ - name of role.

The selector **"dot"** defines a selected role and the selector **"colon"** defines a selected qualifier.

Denote by $Len(P)$ the length of path $P = w_1 ... w_n$. Then $Len(P) = n$. An example of the path in a labeled tree shown in Figure 3 is

*broken* **.** *head* **:** *structure* **.** *end* **:** *coordinates*

**Definition 5.**

Suppose

$AG = (V, E, L \cup N_0, R, beg : E \to V, end : E \to V, m_V : V \to L \cup N_0, m_E : E \to R)$

is an acyclic graph and $P = v_1 ... v_n$ is a path in it. We say that $P$ is attached to the node $v$, if $v = v_1$ and $P$ is defined.

**Definition 6.**

The sample defined as the acyclic graph satisfies the structural constraint $P_1 = P_2$, if the following condition holds.

$$\left((\forall v \in V)(\exists P_1)(P_1 = vv_2 ... v_n) \wedge (\exists P_2)(P_2 = vv_2' ... v_k') \Rightarrow (v_n = v_k')\right)$$

This means that for any graph node $v$ such that paths $P_1$ and $P_2$ are attached to it, we have $v_1 = v_k$.

In this notation, the constraint 10 is an equality in the form:

*broken* **.** *head* **:** *structure* **.** *end* **:** *coordinates* =

*broken* **.** *tail* **:** *broken* **.** *head* **:** *structure* **.** *begin* **:** *coordinates* (11)

The database model for storage of the sample in the form of two relations $(IDQ, IRD)$ has been proposed in paper [8].

The relation $IDQ$ has the scheme $(C, ID)$ where $C$ is an attribute corresponding to the name of the qualifier and $ID$ is an attribute corresponding to the name of the identifier.

The relation $IRD$ has the scheme $(ID, R, IDD)$ where $ID$ is an attribute corresponding to the name of the identifier, $R$ is an attribute corresponding to the name of the role and $IDD$ is an attribute corresponding to the name of the identifier. The axioms for database model for storage of the sample have been formulated in paper [8].

Now we introduce the following algorithm of method of structural constraint checking. This algorithm is described by means of relational algebra.

Suppose $IDQ(C, ID)$ and $IRD(ID, R, IDD)$ are relations to specify the sample. $P_1 = P_2$ is a constraint where $P_1 = w_1 ... w_{n_1}$ and $P_2 = w_1 w_2^{'} ... w_{n_2}^{'}$

**Step 1.** $P = P_1$, $n = Len(P)$

**Step 2**. If $n > 1$

$$rv_0(beg, IDD) = \rho_{ID:=beg}(\pi_{ID,IDD}(\sigma_{C=w_1}(IDQ) \triangleright\triangleleft \sigma_{R=w_2}(IRD)))$$

$$rc_i(beg, ID) = \pi_{beg,ID}(rv_{i-1}[IDD = ID]\sigma_{C=w_{2i+1}}(IDQ)) \quad i = 1...\frac{n}{2}$$

$$rv_i(beg, IDD) = \pi_{beg,IDD}(rc_i \triangleright\triangleleft \sigma_{R=w_{2i+2}}(IRD)) \quad i = 1...\frac{n}{2} - 1$$

$$path(beg, end) := \rho_{ID:=end}(rc_{\frac{n}{2}})$$

If $n = 1$, i.e. $P = w$

$$rc(beg) = \rho_{ID:=beg}(\pi_{ID}(\sigma_{C=w}(IDQ)))$$

$$path(beg, end) := rc[beg = end]\rho_{beg:=end}(rc)$$

Go to the step 4.

**Step 3.** Similarly, $P = P_2$ , $n = Len(P)$ and go to the Step 2.

**Step 4.** Denote $path(beg, end)$ by $path1(beg, end)$ for $P_1$ and denote $path(beg, end)$ by $path2(beg, end)$ for $P_2$. If the relation $path1(beg, end) \cup path2(beg, end)$ satisfy the functional dependence $beg \to end$, then the sample satisfy the constraint $P_1 = P_2$. In the converse case, the sample does not satisfy the constraint $P_1 = P_2$.

Thus, it is possible to describe structural constraint for the sample of concept by means of graph paths. This method allows to store the constrained samples and to check them.

**5 Data Structure Transformation**

The pre-scheme theory within the problems of data exchange could be an intermediate level of transformation of one model of data to other. Such approach allows transforming data from one model to other and also allows verification of data structure. An example of transformations from a hierarchical model into pre-scheme and pre-scheme into relational model is given in figures 5, 6, 7.

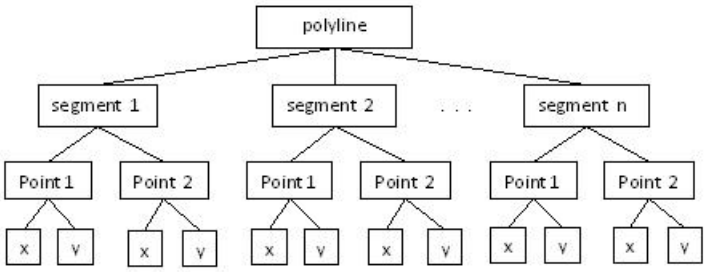



*Fig. 5. A hierarchical data model.*

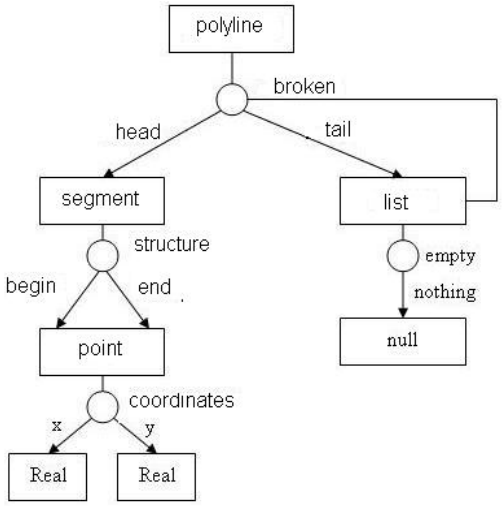



*Fig. 6. A pre-scheme.*

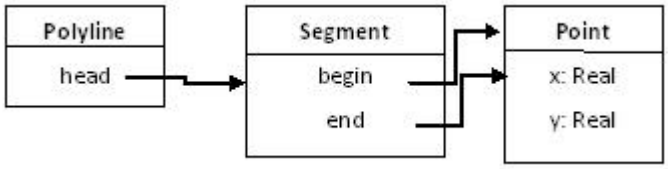



*Fig. 7. A relational data model.*

**Conclusion**

Thus, the theory of pre-schemes for presentation of data model is offered in this paper. Under this theory a sample as structure of data is the model of data schema. Description of concept samples by means of labeled trees allowed to describe arbitrary data schema. The acyclic graph for presentation of data schema that satisfies of the set of structural constraints was suggested.

As a result, the acyclic graph can be regarded as a method of defining the constrained samples. Database model for storage and deserialization of the constrained samples was constructed.

In present paper a method of specifying the structural constraints for samples of concepts in terms of the graph paths had been developed.

The method of verification of accordance of sample to the set of the structural constraints was developed for the samples of concepts defined as acyclic graphs. This method allows to check acceptability of data structure defined as a sample of concept within the bounds of the semantics of subject domain. Also the algorithm for checking the accordance of the sample to the set of the structural constraints was developed. This algorithm was described by means of relational algebra.

Consequently, the obtained results will allow to create an intermediate level for transforming data from one data model to other, that will allow to provide semantic interoperability. The description of model of the data structures by means of the acyclic graphs will allows to provide the control of input data at conceptual level.

REFERENCES

1. Clark,J. , DeRose, S.: XML Path Language (XPath) Version 1.0, W3C Recommendation, November (1999)
2. Chamberlin,D., Clark, J., Florescu, D., Robie,J., Simeon J., Stefanescu, M.: XQuery 1.0: An XML Query Language. W3C Working Draft (2002)
3. Lambert, M., Mariam, T., Susan, F.: Semantic Interoperability. Betascript Publishing (2010)
4. Semenova, T.: Use of Semi-scheme Templates in the Construction of the Logical Model of Subject Domain. Bulletin of NTU "KhPI", Kharkov, 19. (2006) 77{86[in Russian]
5. Zholtkevych G., Ahmad Yusef Ebrahim Ebrahim: On the Opportunity for Marked Tree Representation of Semi-scheme Notion Samples. Systems of information processing. - Kharkiv: KhUAF, Vol. 2(51) (2006) 20{26 [in Russian]
6. Zholtkevych, G., Semenova, T.: To the problem of formalization of information system design. Bulletin of V. Karazin Kharkiv National University, 605. Series "Mathematical Modelling. Information Technology. Automated Control Systems", Issue 2 (2003) 33{42 [in Russian]
7. Zholtkevych, G. , Semenova, T. , Fedorchenko, K.: Representation of Information System Domain Semi-schemes To the Relational Databases. Bulletin of V. Karazin Kharkiv National University, 629. Series "Mathematical Modelling. Information Technology. Automated Control Systems", Issue 3. (2004) 11{24 [in Russian]
8. Zhytaruk,A.,Zholtkevych, G.: Representation of the Concept Samples of the Information Systems by Means of Acyclic Graph. Systems of information processing. Kharkiv: KhUAF, Vol. 6(87). (2010) 215{219 [in Russian]
9. Zhytaruk, A., Zholtkevych, G.: Translation and Veri_cation of SDL-notation by Means of PROLOG Language. Bulletin of KNTU. 2(35). (2009) 200{208 [in Russian]

Надійшла – 26.04.2011.