DATABASE ARCHITECTURE FOR SPECIFYING AND MODELING SPATIO-TEMPORAL RELATIONS

Sergey Salibekyan[a], Peter Panfilov[a,b]

[a]National Research University – Higher School of Economics, Myasnitskaya St. 20, Moscow 101000,
Russian Federation
[b]Fachhochschule Technikum Wien, Höchstädtplatz 6, 1200 Wien, Austria

Abstract

The applications of systems for spatiotemporal relations representation and reasoning range from natural language processing to computer vision and robotics. Our most recent research is concerned with a problem of specifying and modeling spatiotemporal and cause-effect relations between objects in a real physical environment. We propose a method based on defining environmental database as a network (graph) model which originates from the object-attribute (OA) dataflow computing architecture and is similar to a frame knowledge representation scheme. In this paper, we propose a pseudo-physical database architecture which integrates spatiotemporal aspects of the application domain and provides support for reasoning with an OA-grammar of the graph transformation system, a proprietary calculus for spatial reasoning, and an OA-language. A prototype of the OA-DB has been implemented and used to model an application domain-*Natural Language Processing*.

Keywords: physical environment database; spatiotemporal relations; object-attribute architecture; object-attribute grammar; calculus for spatial reasoning; pseudo-physical logics; graph transformation; inference rule

This Publication has to be referred as: Salibekyan, S[ergey] & Panfilov, P[eter] (2016). Database Architecture for Specifying and Modeling Spatio-Temporal Relations, Proceedings of the 26th DAAAM International Symposium, pp.0589-0598, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-07-5, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/26th.daaam.proceedings.080

1. Introduction

This paper describes our efforts of an implementation of pseudo-physical (spatiotemporal) database architecture according to the object-attribute technology – OA-DB. An OA-technology is now under development in Moscow Institute of Electronics and Mathematics at National Research University Higher School of Economics (HSE) in Russia. It includes object-attribute dataflow computer architecture (OAA), object-attribute parallel programming language (OA-PL), and object-attribute software environment (OA-SE) for application design and analysis [1]. Other components of the OA-technology framework targeted at various application domains are planned for implementation.

An object-attribute approach has been used to design and implement systems in application areas ranging from ambient assisted living (AAL) to dataflow supercomputer architecture, to natural language processing (NLP), and to artificial intelligence including neural networks, to name a few. In industrial production/automation domain, research on the application of object-attribute ideas to intelligent manufacturing systems (IMS) has come a long way in the past few years from our works on distributed automation systems through embedded and real-time systems to current work on conceptual models, logical data models, system modeling and implementation [2, 3]. The modern industrial production/automation systems are progressing very rapidly to exhibit high-level intelligent behavioral features such as semi-autonomous and/or autonomous operation of components, self-organization, self-orientation, cooperation and coordination in a complex industrial environment. Modern IMS (in particular, multi-robot production/manufacturing systems, BAS [4], and so on) rely heavily upon accurate information about physical environment, spatiotemporal and other phenomena and objects in it. For this purposes a dedicated pseudo-physical (spatiotemporal) database management systems are developed – IMS-DB.

The object-attribute approach can be applied at all stages of the IMS lifecycle from conceptual modeling (of industrial automation phenomenon), through logical modeling (of industrial automation database) and software modeling (of IMS), its design and implementation to use. The object-attribute semantic models are used to describe spatiotemporal, cause-effect phenomena and objects in physical environment of IMS. An object-attribute spatiotemporal database management system (IMS-DB) is designed according to the object-attribute logical data model, and an object-attribute industrial automation software is developed based on the OAA model, IMS-DB and OA-PL. Some techniques on object-attribute analysis, object-attribute system design, and object-attribute system implementation are introduced in this paper.

We first briefly describe the related work in section 2. Section 3 discusses the conceptual model of pseudo-physical (spatiotemporal, cause-effect, etc.) phenomena by object-attribute analysis. Then we present object-attribute logical spatiotemporal data model in section 4. This is followed by a discussion of design of object-attribute spatiotemporal database management engine (OA-DB) in section 5. Software modeling implementation of an OA-industrial automation software in section 5. We then conclude and discuss future work.

2. Related work

To describe the space, a special branch of science, called the spatio-temporal calculus, has been developed, where the intelligent systems are studied. These calculi deal with methods for describing the spatiotemporal relations between objects by determining the relative position of physical objects, the time-variations in their location or characteristics, the direction of their motion or the object orientation in space, and the cause-effect relations between phenomena. Any similar calculus also has an apparatus for obtaining new information from the already known information – *inference rules.* Several similar calculi are available nowadays. We here list some of them:

- Allen’s Interval Algebra [5] introduces thirteen basic relations between temporal intervals.
- The calculus of cardinal directions, CDA [6].
- The relative orientation calculus [7].
- The Δ Calculus [8] operates with interval relations.
- RCC-8 (Region Connection Calculus) [9] introduces eight relationships between two spatial regions.
- The cardinal direction calculus (CDC) [10] is based on 218 basic relations and can describe almost 2004 different configurations of entities.
- The direction calculus described in [11] introduces a vector of two points, the space about the two points is divided into 15 parts, and an entity is located at one of this parts.

Spatiotemporal representation and reasoning is important in many application domains including interactive 3D computer graphics and virtual reality. The scene graph whose vertices are associated with objects or sets of objects in virtual environment and whose arcs indicate semantic relations between objects has been used widely in this field. In the context of our study, the scene graph based on the “entity-relations” (ER) model deserves special attention [12].

In our studies on OA-technology, we first approached the problem of spatiotemporal relation modelling, representation and reasoning, when implementing the object-attribute design of NLP application (OA-NLP) [13]. For this application, we was to construct a spatiotemporal database format for describing the meaning of a text in a natural language. The spatiotemporal DB in the OA-NLP system has to describe a sufficiently wide application domain up to the entire physical real world environment (no constraints should be imposed on the subject matter of the text under study).

Because descriptions of spatiotemporal and cause-effect relations are encountered in real texts rather often, it is necessary to develop techniques (including those of the original calculus) for their description and analysis. Later in the course of our study, we have discovered that the OA-methods and techniques developed for the NLP application can find applications not only in analysis of a natural language and its automatic translation but also in the other application areas including intelligent systems and industrial automation systems.

We have applied the OA-approach to describe the text meaning that allowed us to construct data structures of network (graph) type which are pretty similar to the frame knowledge representation scheme. These data structures can be used as a basic OA-model of spatiotemporal database. We call such a structure a *semantic OA-graph*. The semantic OA-graph format has currently been developed to the level to permit us to describe an extremely wide data domain (general DB) including the spatiotemporal and cause-effect relations between physical objects (pseudo-physical DB). In what follows, we first describe the general DB (the description of spatiotemporal phenomena is an integral part of the general DB) and then give a more precise account of the pseudo-physical DB. The pseudo-physical DB can be divided into three parts, namely, the passive DB, search/retrieve engine, and active DB. The first part serves to describe the spatiotemporal relations between entities in physical environment. The second part is necessary to implement the information search/object retrieve operations on the semantic OA-graph. The third part is necessary to extract new data from the already known information contained in the DB (inference rule operations).

3. Object-attribute database

The OA-approach which underlies the environmental (spatiotemporal) DB architecture [14] allows us not only to organize a highly efficient parallel computation process but also to create a spatiotemporal DB of network (graph) type and to implement the object-attribute database management system (OA-DBMS) for it. The OA-approach is also used not only to represent the text meaning but also to synthesize a semantic OA-graph from a text in a natural language. In the present paper, we do not describe methods for synthesizing an OA-graph from a text in a natural language but focus on the DB format and methods for analysis and transformations of DB.

The OA-model of data structuring and modelling is close to the frame knowledge representation concept [15]. In the case of this organization, the data are arranged as a set of frames (set of slots which can contain a constant or a reference to another frame). The references reflecting the semantic relations between objects associated with frames are used to unite the frames into a network. Such a network can describe a complex object or a system. The frame network concept is widely used in linguistics, for example, in Ch. Fillmore's theory of semantic roles [16] and the frame semantics which was developed on the basis of this theory [17]. According to the frame semantics, the text meaning can be described by a graph. The nodes of this graph are associated with objects which play definite roles (usually, a node of the graph is associated with a noun). Ch. Fillmore distinguished the following roles: agent (initiator of action), contractor (force resisting to the action), object, place, addressee, patient (thing experiencing changes, for example, «to break a cup»), and result, tool, source (initial state of the object). But according to the frame semantics, each part of speech is associated with a frame which determines possible semantic relations to other words. Such relations are called *valences*, by analogy with chemistry, where atoms form molecules by attaching other atoms through their valences. For example, the verb has the following valences: subject (who acts), object (who experiencing the action), tool, addressee, etc. Later, the construction grammar (CxG) was developed from the theory of semantic roles and the frame semantics [17]. For example, [18] presents an attempt to perform a CxG analysis of the space description given in the text.

The OA-principle permits formalization of the frame network approach and its computer implementation (and such formalization ensures not only the data organization but also the information search in the network and the information transformation, which will be discussed below). As a result, the model of a system of objects and their semantic relations was implemented as a semantic OA-graph (the data base (DB) of network (graph) type). The notes of a semantic OA-graph (an analog of the frame) describe the objects and their characteristics and states. One characteristic of the object, properties, or state is described by an information pair (IP) (IP is an analog of the frame slot) consisting of the following two fields: an attribute (the property identifier) and a load (a constant константа or a pointer). Namely, an IP is represented as $c=<a,l>$, where $a \in A$ is the attribute (A is the set of attributes which permit identifying the load l) and $l \in L = \{\Re \cup nil \cup S \cup \Omega\}$ is the load (L is the set of data in the load, $\Re$ is the set of rational numbers, nil stands for the empty load, Ω is the set of indices of the OA-graph nodes, and $S \in \sum^*$ is the set of chains of symbols belonging to the alphabet $\sum$, where the * symbol denotes the transitive closure). The set of IP (the information capsule (IC)) is used to describe an object and its characteristics or states. The semantic relations (i.e., the relations between IC) are implemented by information pairs (IP) whose loads bear pointers to other IC. A set of several IC mutually connected by references form a semantic OA-graph which describes a complex object or a system. A fragment of the semantic OA-graph is shown in Figure 1. This fragment presents a description of objects and their properties and states. The attribute 'Atr' labels the reference to IC with a description of the object characteristics; 'Class' labels the class code of the object contained in the load of IP; 'Obj' is a pointer to the description of the object of action, 'subj' is a pointer to the subject; 'Atr1, Atr2, Atr3' label the attributes characterizing the object under study. The meaning is represented as an OA-graph as in the frame network, but this representation is more flexible because it does not rigidly specify the number of slots required to form relations typical of the description of certain object classes. The number of relations (valences) and characteristics of the object description is arbitrary, i.e., an IC can contain arbitrarily many IPs. The IP can be added to and removed from an IC at any time.

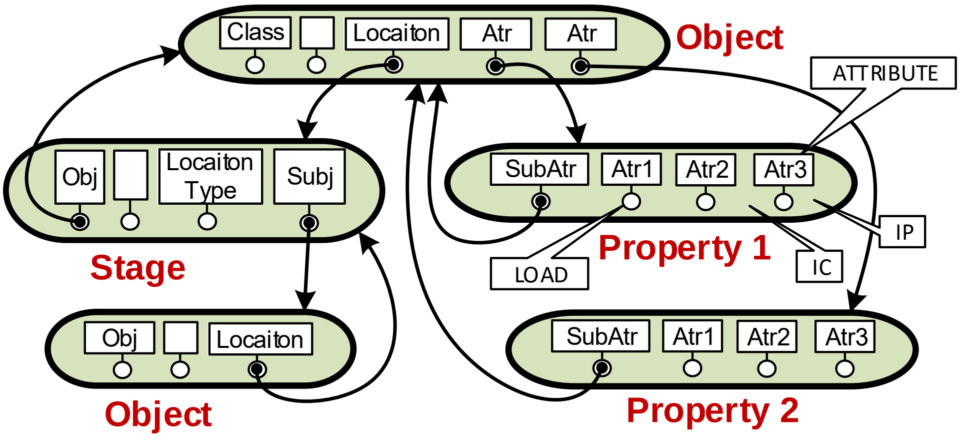

Fig. 1. Semantic OA-graph.

4. Format of object-attribute database

An important stage of studies is the design of the rigid format of general OA-DB for the purpose of simplifying the information search in it and performing its transformations (the information search algorithm becomes simpler on a rigid structure).

As a result of our studies, we developed an AO-DB format consisting of the following four levels (Fig. 2):

1) Descriptions of objects,
2) Descriptions of the properties and states,
3) Descriptions of spatiotemporal and cause-effect relations,
4) Descriptions of numbers and measurements.

Each level is divided into the following three sublevels: description of the set of objects, list of elements of the set, description of the objects contained in the set. This division is necessary to be able to describe the sets of objects, states, or locatives (points in the space-time). For example, the expression «Object1 and Object2» means an unordered set of type «AND», i.e., a set of several objects which are present simultaneously. «Object1 or Object2» means an unordered set of type «OR» (one or more objects are present simultaneously). And the expression «First Object1 and then Object2» means an ordered set. The set can comprise not only objects but also their properties and states («One goes and carries a weight», «large and strong») and spatio-temporal and cause-effect relations between the objects («He came home and looked for slippers», here the states «came» and «looked for» form an ordered set, because one states precedes the other), numbers and measurements («10 or 15 meters», «5 or 10 objects»). This format permits describing a wide class of data domains and performing semantic search of information in the database.

The sets are described by OA-graphs as follows. An IC located at the first sublevel (of any level) necessarily contain an IP whose attribute points to the set type and whose load contains a pointer to the IC with the list of the set elements. The first sublevel can also contain a pointer to the number of the set elements (cardinality of the set). The sublevel corresponding to the list of the set elements contains an IC whose IP can have one of the following two attributes: ‘Element’ (an element is contained in the set) and ‘Exclude’ (an element is not contained in the set). The load of such an IP contains a pointer to the IC with the description of the element, which is contained or not contained in the set. The elements of the set at the first level of the OA-graph are objects, the elements at the second level are states, the elements at the third level are locatives, and the elements at the fourth level are numbers. As an example, we consider the following types of sets: ‘Set’ (unordered set), ‘SetOr’ (at least one element of the set exists), ‘SetXor’ (only one element of the set actually exists), ‘SetOrder’ (ordered set), ‘SetRevers’ (set is ordered in the opposite direction), ‘SetToFro’ (set which can be regarded as an ordered set in one direction and as an unordered set in the opposite direction). The application of these types of sets will be illustrated below.

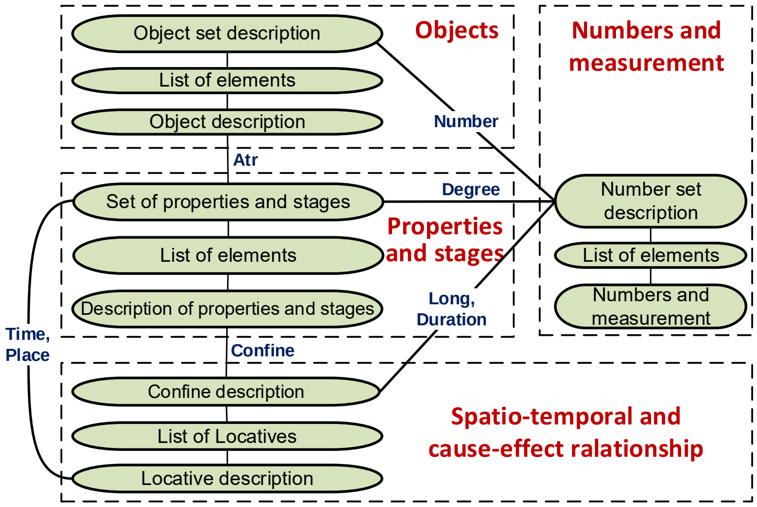

Fig. 2. Format of the OA-database

The sets are described by OA-graphs as follows. An IC located at the first sublevel (of any level) necessarily contain an IP whose attribute points to the set type and whose load contains a pointer to the IC with the list of the set elements. The first sublevel can also contain a pointer to the number of the set elements (cardinality of the set). The sublevel corresponding to the list of the set elements contains an IC whose IP can have one of the following two attributes: ‘Element’ (an element is contained in the set) and ‘Exclude’ (an element is not contained in the set). The load of such an IP contains a pointer to the IC with the description of the element, which is contained or not contained in the set. The elements of the set at the first level of the OA-graph are objects, the elements at the second level are states, the elements at the third level are locatives, and the elements at the fourth level are numbers. As an example, we consider the following types of sets: ‘Set’ (unordered set), ‘SetOr’ (at least one element of the set exists), ‘SetXor’ (only one element of the set actually exists), ‘SetOrder’ (ordered set), ‘SetRevers’ (set is ordered in the opposite direction), ‘SetToFro’ (set which can be regarded as an ordered set in one direction and as an unordered set in the opposite direction). The application of these types of sets will be illustrated below.

The attributes (semantic relations) used in OA-DB can be divided according to their belonging to a certain level of the semantic OA-graph. So, at the object description level, we need the following attributes: ‘Class’ (class of the object), ‘Name’ (name of the object), and ‘Atr’ (property or state of the object). At the level of properties and states, we need the following attributes: ‘Stage’ (state of the object), ‘Color’ (color of the object), ‘Obj’ (object of action), ‘Subj’ (subject of action), ‘Confine’ (spatio-temporal constraint on the state of the object), etc. The attributes used at the level of space-time description will be discussed in the next chapter, and the attributes requires at the level of number description are beyond the scope of the paper.

The levels and sublevels of OA-graphs can be semantically related to each other (in Fig. 2, such relations are denoted by solid lines). So, the sublevels of the first three levels of the set description can contain pointers to IC at the level of numbers and measurements, for example, in the case where the set cardinality is indicated, for example, «crowd of 100 people», «1000 houses», etc. In this case, at the sublevel of the set description, there is an IP with the attribute ‘Number’ and a pointer to the description of the number determining the set cardinality. The object description can be related to the level of properties and states. For this, at the sublevel of object description, there is an IP with the attribute ‘Atr’ whose load stores a pointer to the IC with the description of the object properties. The relation between the sublevel of properties and state to the sublevel of space-time description is indicated by the attribute ‘Confine’. The sublevel of the set description belonging to the level of space-time description can be related to the sublevel of numbers and measurements; such a relation is indicated, for example, by the attribute ‘Long’ (length of the object trajectory) or ‘Duration’ (duration of the object motion or the state where it stays).

Now we pay our attention to the third level of the semantic OA-graph, where an IC with the description of spatiotemporal and cause-effect relations between physical objects is contained.

5. Passive database

To solve real computational problems related to the semantic analysis of texts, it is necessary that the pseudo-physical DB contains all possible spatio-temporal relations, namely, the spatio-temporal relations between several entities, the object shape and dimensions, the object trajectory, the direction (the vector of the object position in space and the motion direction), and the cause-effect relations between events. It is also necessary to be able to describe relations which are equipped with a metric or not. In this case, the DB and the calculate must be implements on basis of a network (graph) model so that it could be written in the framework of a semantic OA-graph. As an analog of the proposed technique, we can consider the system [19] implementing the spatio-temporal calculus RCC-8 on the basis of the network (graph) DB RDF/OWL.

Unfortunately, the currently available spatio-temporal calculi do not allow any implementation of the OA-approach. Therefore, we developed our own format of data representation and an original calculus for searching the information in DB and obtaining new data from the set of available information. The DB used to describe the space-time is called the passive DB. The pseudo-physical DB also contains a search/retrieve part which is intended for information search and object retrieval in the pseudo-physical DB. We first describe the passive part.

The basic notions of the passive part of a pseudo-physical DB are «Locative» and «Confine». The attribute ‘Locative’ means a point or a domain in the space-time and the attribute ‘Confine’ describes the set of mutually related locatives or distinguishes (bounds) a part of space-time where, for example, the object is located or where it can be present (probability of the object presence). The locatives are divided into spatial (indicate a point in space), temporal (time point), and spatio-temporal ones. The confines can be of the following types: dynamics (describes the object displacement, i.e., distinguishes the object trajectory in space-time), direction (determines the vector of direction in space), topology (indicates the relative position of objects), and shape (determines the object shape and dimensions, i.e., bounds the space parts occupied by the object). The confines can be equipped with a metric (indicating the dimensions and duration) or a coordinate system (the origin of the coordinates and the direction of the coordinate lines are given).

Three sublevels are separated in the OA-graph (Fig. 2) to describe the confines. The first sublevel describes the confine itself, i.e., it describes, for example, the type of the confine, the length of the route, the time of motion along the route, the coordinate system, the type of the set of locatives, etc. At the next sublevel (list of locatives), there is an IC combining IPs with pointers to the locatives contained in the confine. At the third level, there are ICs describing locatives (such an IC can contain two IPs which describe the position (attribute ‘Place’) and/or the time (attribute ‘Time’)). Now, we consider several examples of confine description which practically illustrate the DB format and the technique for its organization to explain the methods for describing the spatiotemporal relations.

First, we consider an example of the confine-direction description. For example, it is required to analyze the situation given by the sentence «He saw a car through the window», where the observer look direction is described. Figure 3 outlines the scene as follows: ZeroPoint is the origin of the coordinate system (this is a locative, i.e., the observer position); AxisPoint is the point through which the coordinate vector passes (in our case, through the window); Direction is the vector direction (in our case, Forward, i.e., along the coordinate vector direction).

In this case, the level of numbers and measurements is not used, because no metric is described in the text. So the figure presents the description of three objects (Person, Window, and Car). Each object is in a certain state, namely, ‘Person’ is in the state of looking, and the other two simply exist (therefore, the load of the IP with the attribute ‘Stage’ contains the zero pointer, i.e., the state is unknown). The DB contains one confine of direction, which contains three locatives, i.e., the places of ‘Person’, ‘Window’, and ‘Car’. The IC at the sublevel of track description contains references to the coordinate system origin (where the ‘Person’ is located) and the point through which the coordinate axis passes (where the ‘Window’ is located) and a pointer to the IC containing the set of pointers to the track locative. In this case, a set of the type of ‘SetOrder’ is used, because the order of elements (locatives) is essential here. All three objects forming the scene are successively located along the vector direction, and therefore the list of locatives contains three pointers to the places where Person, Window, and Car are located.

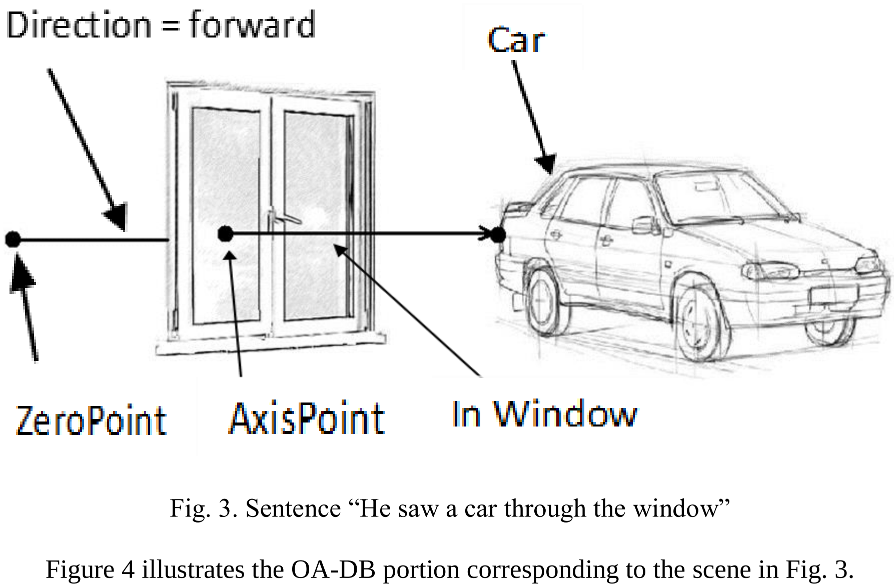

Fig. 3. Sentence “He saw a car through the window”

Figure 4 illustrates the OA-DB portion corresponding to the scene in Fig. 3.

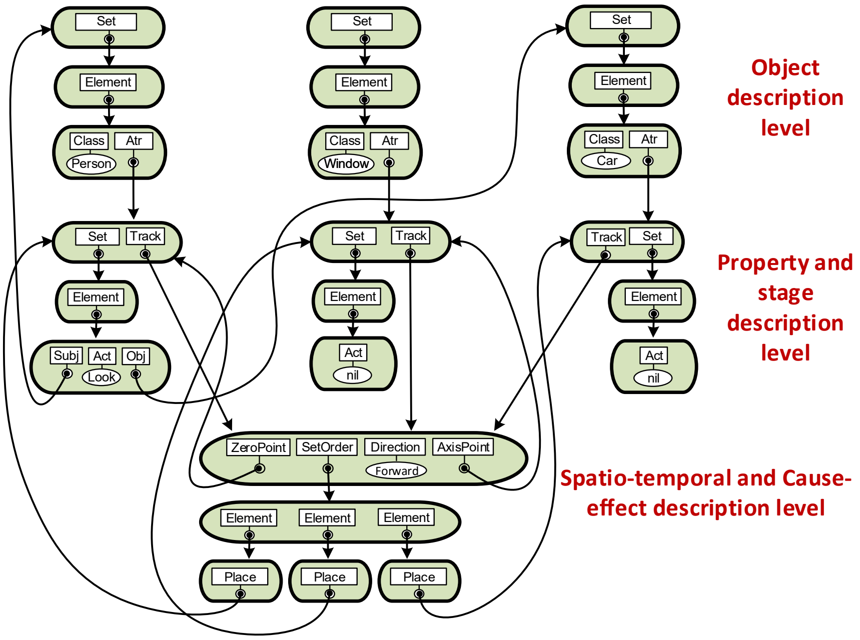

Fig. 4. OA-database representation for the sentence "He saw a car in the window"

Further, we consider an example of a cause-effect confine. For this, we examine the sentence «A boy pushed the door, and it opened». The corresponding DB portion (Fig. 5) contains descriptions of the two objects, boy and door. The boy is in the state of pushing. The door is in the two successive states: first in an unknown state, and then in the state of opening. To show the sequence of the two states of the door, they are united by a confine of the type 'Order' (successive events) as follows: the first element in the ordered list indicates the first (unknown) state of the door, and the second element indicates the state of opening. The confine of cause-effect relation unites the boy's state of pushing and the door's state of opening, because the result of the boy's action is that the door is open. But the object of the boy's action (Obj) is the door in the first state, because the boy's action was applied to the door before it was open.

The passive part of a pseudo-physical DB first describes topological relations between the objects and the direction, because the OA-graph can contain confines of the type «over», «under», «inside», «outside». Each of such confines contains two locatives. The relation «between» is determined by a confine containing three elements. So, the object «Window» in Fig. 4 is located between the observer and the «car», and this conclusion is made because the set of locatives is ordered in this case, and one can separate the objects located ahead and behind a certain object. Second, the passive part describes the metric in the space, which can be introduced in the following two ways: by an absolute value («at the distance of 10 meters from something») and by a relative value («large», «long», etc.). The time and the temporal interval can also be absolute («in 10 minutes after something», «a week before something») and relative («soon after something», «sufficiently long»).

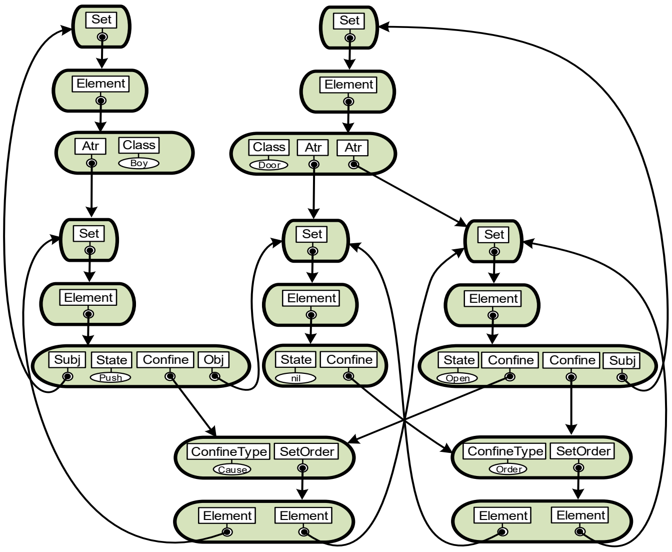

Fig. 5. OA-database for the sentence "He left the house and entered a shop"

It is most convenient to introduce the relative values by using the Linguistic Variable [20]. The Linguistic Variable in an OA-system is implemented by a subprogram built in the OA-graph. Third, the passive part describes the cause-effect relations (causality logics) which are introduced by a track of «cause-effect» type. The description of such a confine contains a reference to an ordered set where the first pointer refers to the state of an object which is the cause, and the second pointer refers to the state of another object which the effect. Fourth, the passive part describes the object shape and dimensions. In the OA-DB, one can also specify the information uncertainty, for example, the probability of the event that the confine contains a locative. This permits using the apparatus of fuzzy sets [21, 22] to search and analyse information in the OA-DB. Fifth, the passive part describes the direction.

6. Search/retrieve engine

The search/retrieve operation in the OA-graph is the following procedure. The text (or any other information with a description of the environment) where it is required to look for the information is first transformed into an OA-graph (general DB). Then a query is introduced (for example, the user makes a query in the natural language) which is also transformed into an OA-graph (query graph). Further, it is determined whether the query graph is a subgraph in the general DB. Thus, for example, one can determine whether the situation (environment) corresponds to the given description. In this section, we consider the algorithm for searching/retrieving a subgraph in a pseudo-physical DB. The main difficulty arises when different types of sets of locatives are compared. Here we list sets of several types, which are used in the pseudo-physical DB:

1. Set: unordered set (for example, used to describe the relation «near»).
2. SetOrder: ordered set (used to describe dynamic tracks).
3. SetToFro: set ordered in both directions (i.e., a set which can be ordered in either direction depending on the situation).
4. Subset: unordered subset.
5. SubsetOrder: ordered subset.
6. SubsetToFro: subset ordered in both directions.
7. SubsetOrderNotFull: ordered incomplete subset.
8. SetHead: first element of an ordered set (track head).
9. SetTail: last element of an ordered subset (track end).

Now we present several examples of comparison of different types of sets. To find out whether a locative is the head of a dynamic confine, we must introduce a confine with a set of SetHead-type locatives in the query OA-graph. A set of this type can be compared only with a SetOrder-type set. Two ICs with lists of locatives are assumed to coincide if the first element of the SetOrder-type set coincides with an element of the SetHead-type set. If it is required to learn whether there are two coinciding confines consisting of unordered sets of locatives (a Set-type set is used, for example, to describe the relation «near»), then we use the Set-type set. In this case, elements of the set in DB must contain elements identical to the IC of the query graph (the sequence order of IC is here of no importance). If it is required to find out whether a dynamic confine coincides with another confine, then it is necessary to use SetOrder-type sets. Sets of this type coincide if they have the same number of elements and if each element of one set is in correspondence with the same element which has the same sequence number in the other set. If it is required to verify whether the attribute «window» is located between the attributes «observer» and «car», then it is necessary to form a subset ordered in both directions (the SubsetToFro type originates from the word combination «to from») in the query graph, because both the ordered set {«Observer», «Window», «Car»} (here «{» and «}» denote the head and end of IC) and the set {«Car», «Window», «Observer»} satisfy the relation «between». These two situations can be described by the construction {SubsetToFro={«Car», «Window», «Observer»}} (here «=» denotes an IP (the attribute name is on the left of the symbol «=» and the load name is on the right of «=»)). In this situation, the coincidence of locative sequences is verified in the forward and backward directions. To determine whether the locative «car» is the end of the track, it is necessary to introduce the construction {SubsetTail={«car»}} in the query and then to verify whether the element in the query set coincides with the last locative in IC in the text DB.

7. Active DB

To obtain new information from the already available information, we earlier developed an apparatus of OA-grammar (in fact, the OA-grammar is a graph transformation system). The nearest analogue of such an apparatus is the Head-Driven Phrase Structure Grammar (HPSG) [23]. The HPSG is also intended for describing operations of structure transformations of data describing the meaning, but the HPSG can operate only with tree-type data structures but not with network structures, where as the OA-grammar is intended for describing the operations of transformations of the network data structure, more precisely, of OA-graphs. Formally, an OA-grammar can be described as a quadruple: OAG={A,L,P,G}, where A is the alphabet of attributes; L is the alphabet of IP loads (this alphabet contains not only numbers and rows but also references (indices) to the IC); G is an OA-graph (a set of ICs united by references); and P denotes the OA-graph transformation rules. The OA-graph transformation rules consist of two (left and right) parts separated by the symbol «→». The OA-graph pattern in placed on the left, and the OA-graph which replaces this pattern stands on the right. The following notation is used to describe the OA-graph:

a = l denotes an IP ($a \in A$, $l \in L$, where A is the set of attributes and L is the set of IP loads);
Name{…}is an IC; (the IPs belonging to the IC are written between the symbols «{» and «}»); an IC is a set of IPs; Name is the name of IC (or a reference to IC);
{…a1= ICName{Ip1,Ip2…} …} is the IP whose load contains a pointer to another IC in the load; the IC name can be omitted;
* is the IC concatenation (the representation IC1*IC2 means that the capsule IC2 must be linked with the capsule IC1);
^{F} denotes the first IP in the IC;
^{L} denotes the last IP in the IC;
‘ denotes the fact that the first or the last IP is deleted from IC.

For example, there are two dynamic confines consisting of two locatives. The final locative of one confine is the starting locative of the other confine. Thus, these two confines actually represent one confine consisting of three locatives. Therefore, it is necessary to perform the following transformation which is described by the OA-grammar rules: {Confine={SetOrder=T{ ^{L}temp}} Confine={SetOrder=T2{ ^{F}temp}}} → {Confine={SetOrder=T1*’T2}}}

Thus, using the OA-grammar, one can describe the OA-graph transformations intended for adding new information to OA-DB and optimizing the OA-DB or adding new information to OA-DB.

8. Conclusion

The OA-technique presented above for modeling and implementation of an environmental pseudo-physical DB capable of representing spatiotemporal and cause-effect relations between objects (the passive, active and search parts of DB) allows one to describe practically any arbitrary relations between physical objects, namely, topology, time, cause-effect relations, shape, etc. Because of their versatility and flexibility, such pseudo-physical DB are widely used in various fields of science including intelligent manufacturing systems in industrial automation field.

The OA-DB implementation based on OAA model promises significant advantages. In this case, the unique data representation is ensured at the level of computer hardware, system software and the DB itself. The OA-computer system also ensures implementation of distributed DB and maximal parallelism of computations (the computations are self-parallelized due to the dataflow paradigm).

An application system with OA-DB for semantic analysis of a natural language is now developed. At present, an prototypical OA-NLP application, which is a linguistic processor generating OA-DB from texts in English or Russian, is realized. Our plan for the near future is to develop a system of semantic search in OA-DB and the simulation of OA-computer systems for improving the data representation in DB and developing the optimal architecture of computer systems for solving similar computational problems.

Also future components of the OA-technology framework include an object-attribute knowledge base (OA-KB), object-attribute neural networks (OA-NN), object-attribute database management system (OA-DBMS), object-attribute automation system (OA-AUTO), and so on.

9. Acknowledgements

The study was conducted in the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE) in 2015.

10. References

[1] S.M. Salibekyan and P.B. Panfilov, Object-attribute architecture is a new approach to object systems design, Information Technologies, No.2(186)/2012, pp.8-13. (in Russian)

[2] P.Panfilov, S. Salibekyan, Dataflow computing and its impact on automation applications, in: Procedia Engineering, Vol.69 (2014), pp.1286-1295. ISSN: 1877-7058 © 2014 by Elsevier Ltd. DOI: 10.1016/j.proeng.2014.03.121

[3] S. Salibekyan, P. Panfilov, A new approach for distributed computing in embedded systems, in: Procedia Engineering, Vol.100 (2015), pp. 977-986. ISSN: 1877-7058 © 2015 by Elsevier Ltd. DOI: 10.1016/j.proeng.2015.01.457

[4] I. K. Kukushkin, B. Katalinic, P. Cesarec, and R. Kettler, Modeling of mobile robot behavior for line-less bionic assembly system, in: Annals of DAAAM for 2012 & Proceedings of the 23nd International DAAAM Symposium, Vol. 23, No. 1, ISBN 978-3-901509-91-9, ISSN 2304-9679, pp 0641-0642, Editor B[ranko] Katalinic, Published by DAAAM International, Vienna, Austria 2011

[5] J.F. Allen, Maintaining knowledge about temporal intervals, Communications of the ACM. 26(11), Nov. 1983, pp.832-843.

[6] A. U. Frank, Qualitative spatial reasoning about distances and directions in geographic space, Journal of Visual Languages and Computing. 3, 1992, pp.343–371.

[7] C. Freksa, Using orientation information for qualitative spatial reasoning, in: A. U. Frank, I. Campari, and U. Formentini (Eds), Proceedings of GIS — from Space to Territory: Theories and Methods of Spatio-Temporal Reasoning, Springer, Berlin, 1992.

[8] K. Zimmermann, Measuring without measures - the Δ-calculus, Graduiertenkolleg Kognitions-Wissenschaft, Universität Hamburg, Report 39, 1994.

[9] D. Randell, Z. Cui, and A. Cohn, A spatial logic based on regions and connection, in: Proc. KR, 1992.

[10] R. K. Goyal and M. J. Egenhofer, Similarity of cardinal directions, in: SSTD, 2001, pp.36–58.

[11] K. Zimmermann and C. Freksa, Qualitative spatial reasoning using orientation, distance, and path knowledge, in: Applied Intelligence 6 (1996), pp.49-58.

[12] G. Miaoulis, D. Plemenos (Eds.), Intelligent scene modelling information systems, in Series: Studies in Computational Intelligence, Vol. 181 Springer, 2009. DOI: 10.1007/978-3-540-92902-4

[13] S.M. Salibekyan and P.B. Panfilov, Language analysis aided with object-attribute computer system architecture, Software Engineering, No.9/2013, pp.9-16. (in Russian)

[14] J. Silk, B. Robic, and T. Ungerer, Asynchrony in parallel computing: From dataflow to multithreading, Institut Jozef Stefan, Technical Report CDS-97-4, September 1997.

[15] Unknown, Minsky's frame system theory, in: Proceedings of the 1975 Workshop on Theoretical Issues in Natural Language Processing - TINLAP '75, Association for Computational Liquistics, Stroudsburg, PA, USA, 1975, pp. 104–116. DOI:10.3115/980190.980222

[16] The Case for Case, in: Bach and Harms (Ed.), Universals in Linguistic Theory. New York: Holt, Rinehart, and Winston, 1968, pp.1-88.

[17] Frame semantics and the nature of language in: Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech Volume 280 (1976) pp.20-32.

[18] T. Hoffmann and G. Trousdale (eds.), The Oxford handbook of construction grammar (Oxford Handbooks). Oxford: Oxford University Press, 2013. Pp. xxii + 586.

[19] A. Cienki, Spatial cognition and the semantics of prepositions in English, Polish and Russian. Munchen: Verlag O. Sagner, 1989.

[20] M. Stocker, E. Sirin, PelletSpatial: A hybrid RCC-8 and RDF/OWL reasoning and query engine, 2009.

[21] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-I,II,III. Information Sciences, 1975, 8(3), pp.199-249; 8(4), pp.301-357; 9(1), pp.43-80.

[22] L. A. Zadeh, Fuzzy sets. Information and Control 8 (3), 1965, pp.338–353

[23] S. Müller, Unifying everything: Some remarks on simpler syntax, construction grammar, minimalism and HPSG, Lan¬guage 89(4), 2013, pp.920–950.