

CONCEPTUAL OBJECT-RELATIONSHIP-PROPERTY APPROACH: THREE DIFFERENT INTERPRETATIONS OF THE SAME ENTITY

Bronius PARADAUSKAS

Kaunas University of Technology
Vytautas Magnus University
Studentų St. 50-108, 3028 Kaunas, Lithuania

Abstract. In this paper the universal structural type of entities is presented by recursive applications of two operators, i.e., operator which constructs positioned subsets of entity name universum and operator which constructs a functional set on the entity name universum. The object subtype is determined using an especial commutational diagram with the compositions of inner functions. The main integration rule of conceptual object-relationship-property scheme is determined by means of external containment function. A compositions of inner and external functions serve as a tool for normalization of entities systems. The advantages of proposed approach to the modelling of entities system are pointed out.

Key words: entity, object, relationship, property, conceptual scheme, relational scheme, structural types, commutational diagram, compounded functional dependence, extent function.

1. Introduction. The information system is a part of the enterprise which may be viewed under consideration from the goal, from the action and from the object perspective. The primary purpose of an information system is to provide a model of an object system. Defining a model of an enterprise Boman *et al.* (1993) use the conceptual model term to denote a model from the object perspective. The object system term has been used to refer to some part reality. An information system consists of three parts: a conceptual schema, an information base, and an information processor (ISO, 1982). A conceptual schema and information base are totally static unless an information processor operates on it to cause change.

Classical ENTITY-RELATIONSHIP formalism (Chen, 1976) and all the

descendents attempt to describe static structure of an object system. Entity–relationship diagrams are graphic notation for showing entity types, their attributes, and the relationships that connect them. Enhancement to the ER formalism have been pursued for increasing the semantic richness of conceptual schema (Brodie, 1984; Hull, 1987). The reason for this aim of conceptual scheme to capture the various constraints through its structure. We will analyse in this paper universal object type, which covers structural constraints. Constraints of this kind are presented in a declarative manner and later they must be converted always to procedural form. The constraints restrict the values that entity or relationship attributes can assume.

The approach (Dardenne *et al.*, 1993) to requirements acquisition involves three levels of modelling: the meta-level, the domain level and the instance level. In other words all three level are made from meta-types, types and type instances. The instance level consists of instances of domain level concepts. The concepts of domain level are instances of meta-level (e.g., AGENT, EVENT, ACTION, ENTITY, RELATIONSHIP). The domain level refers to concepts specific to the application domain (e.g., CLERC, ORDER, VENDOR). Meta-concepts and meta-relationships have been defined according to approach to goal directed requirements acquisition. The ENTITY, RELATIONSHIP, EVENT and AGENT concepts are considered as specializations of the OBJECT meta-concept. A relationship may be considered as a subordinate object. An object as well as a relationship have instances; the existence of relationship instances depends upon the existence of the corresponding object instances connected by the relationship. The object meta-concept is involved in a member of meta-relationships. Usually OBJECT concept means a single thing, e.g., object instance, and OBJECT concepts refers to a class of similar things. The abbreviation OBJECT we will use instead of object class because we will use it as type, as a set of object instances with identical structure. We will capture the static structure of object system by showing objects, relationships between the objects, and the attributes, and procedures that characterize each object. Objects are material things, the units into which we divide the world and object system model capture those concepts from the real world that are important to all goal of information system and that appear in all goals directed requirements. The paper (Wangler, 1993) presents the ENTITY–RELATIONSHIP–TIME model, the extended ER-model to describe historical information and to model the

functional requirements. A TIME is introduced as distinguished entity class. Each object or relationship can be connected with TIME and can be presented as time stamped object or time stamped relationship. This is good idea from a point of objects naturalization and scheme presentation. In fact the temporal objects and relationships in ER-formalism language are relationships.

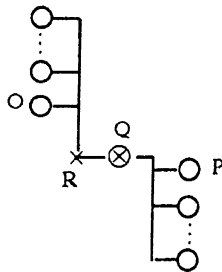
The functional methodologies (DeMarco, 1979; Yourdon, 1989) are based mainly in functional decomposition although object-oriented methodologies (Booch, 1991; Rumbaugh *et al.*, 1991) focus first on identifying objects, then fitting procedures around them. In both cases methodologies use three kinds of models to describe a system – object model, state – transition diagram, and data flow diagram. These three views of the system must be consistent and compatible with one another. Functions sometimes may be decomposed to subfunctions and, similarly, objects may be subdivided into subobjects. A state–transition diagram shows decisions which depend on object values and invoke functions from the data flow diagram; events become procedures on objects. Each store on data flow diagram must correspond to an object or relationship in the object model; a function must be involved by procedures in the object model and actions in the state diagram; a function operates on data values specified by the object model. In this paper we emphasize on objects defined by data structure. An object is autonomous and instances may exist independently from other object instances. If we have two detailed objects and given decomposed procedure uses the values of both objects, then they will be connected by data relationship and they will create new object. So, a procedure may be considered as object property and it makes influence upon object distinguishing.

The meta-relationship between the ENTITY and the OBJECT is often vaguely used in the various approaches. According to Dardenne *et al.* (1993) the ENTITY is a specialization of the OBJECT. And in contradiction Rumbaugh *et al.* (1991), the ENTITY concept includes object–instances, OBJECT–classes, attributes, links and associations. According to this statement the OBJECT concept is a specialization of the ENTITY concept. In these and many other approaches both an ENTITY and an OBJECT are things of significance information about which needs to be known and may be specified. The same ENTITY appears different and behaves differently when it is used in different contexts by the same or by different persons. We will give first not formal OBJECT definition by a set of features.

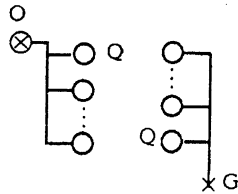
Let is given the entity Q of domain level. The entity Q is assumed as object Q if it necessarily has a name and its feature is to be a relationship R involving at least one another object O and/or its feature is to have at least one property P (Fig. 1,a); the second feature of the object Q is its possibility either to be involved in new relationship G and/or to be a property of another object O (Fig. 1,b).

Graphically:

- a) "Object Q has property P and is relationship R involving another object O ".



- b) "Object Q is a property of O and object Q is involved in relationship G ".



In the process of the requirements acquisition and knowledge representation the system specialists and experts manipulate entities the names of which are included in conceptual model presenting the semantical data properties and the structural constraints. Cause-effect phenomena of the application domain causes taxonomic connections between the entities. Connections of this kind are paths of transition from more general entities to more detailed ones. Formally these connections represent partial ordering: they are transitive, reflexive and antisymmetric. The approach used in this paper follows approach used in (Paradauskas, 1993) and allows to choose the formal apparatus for ordering the

universum of entities names, and to define universal structure type.

Without doubt, proposed formalism for semantic data models and object structured type is not only one. Here are known a number of different mathematical schemes, for example Brachman *et al.* (1985); Sciore (1980). The approach used in this paper follows the method for knowledge representation on abstract semiotic space (Ambrazevitchius and Paradauskas, 1983). In the second section the method mentioned above is simplified and reviewed with respect to the normalization of relational databases through synthesis (Maier, 1983). Relational approach to object system modeling is presented in (Paradauskas, 1994).

2. Formal definition of structural types of object names. Analysis of the structural entities name sets has been based on the initial assertion that name universum D is complete partially ordered set (CPOS). For this purpose the least element \perp such that $\forall x \in D[\perp \sqsubseteq x]$ is introduced and there is a least upper bound $\bigsqcup X \in D$ for any directed subset $X \subseteq D$. The approximation symbol \sqsubseteq indicates the partial ordering of the elements of a directed subset $X \subseteq D$. The subset $X \subseteq D$ is considered to be directed because it is not empty and

$$\forall x, y \in X \quad \exists z \in X[x \sqsubseteq z, y \sqsubseteq z].$$

Let us introduce two operators r and p on the set D as CPOS:

r – operator of constructing positioned subsets of D ;

p – operator of constructing a functional set on D .

The operator $r: D \rightarrow r(D)$ is used in developing a new set and extends the domain of the definition to

$$r(D) = D + D^2 + \dots + D^{|D|}. \tag{2.1}$$

According to (2.1) set $r(D)$ is obtained by union of sets D^n , $n = 1, 2, \dots, |D|$ with artificially introduced least elements

$$\perp_{D^n} = \left\langle \underbrace{\perp_D, \perp_D, \dots, \perp_D}_n \right\rangle, \quad n = 1, 2, \dots, |D|. \tag{2.2}$$

The ordering $\sqsubseteq_{r(D)}$ on the set $r(D)$ is induced by \sqsubseteq_D on CPOS D :

$$\langle x_1, x_2, \dots, x_n \rangle \sqsubseteq_{r(D)} \langle y_1, y_2, \dots, y_n \rangle, \tag{2.3}$$

only when $x_i \sqsubseteq_D y_i$, $i = 1, 2, \dots, n$; $n \leq |D|$.

The upper bound for two elements

$\langle x_1, x_2, \dots, x_n \rangle \in X$ and $\langle y_1, y_2, \dots, y_n \rangle \in X$, $X \subseteq D^n$,
is calculating in the following way:

$$\begin{aligned} \langle x_1, x_2, \dots, x_n \rangle \bigsqcup^{D^n} \langle y_1, y_2, \dots, y_n \rangle \\ = x_1 \bigsqcup^D y_1, x_2 \bigsqcup^D y_2, \dots, x_n \bigsqcup^D y_n. \end{aligned} \quad (2.4)$$

From here sets D^n , $n \leq |D|$ are CPOS as they contain the least element \perp_{D^n} and any subset $X \subseteq D^n$ has least upper bound $\bigsqcup^{D^n} X \in D^n$. The obtained set $r(D)$ in (2.1) is CPOS too, while the union is made by coupling (pasting) the least elements

$$\perp_{D^n}, \quad n = 1, 2, \dots, |D|.$$

From the conditions (2.3), (2.4) it follows that the operator r is monotonous, i.e., the directivity of the subset $X \subseteq D^n$, $n = 1, 2, \dots, |D|$ is not lost while extending the domain of definition from D to $r(D)$.

In addition from the equation

$$r \left(\bigsqcup^D X \right) = \bigsqcup^{r(D)} r(X)$$

it follows that the operator $r(D)$ is continuous.

Let us consider the second operator $p: D \rightarrow p(D)$ constructing a set of continuous functions:

$$p(D) = [D \rightarrow D]. \quad (2.6)$$

The ordering $\subseteq_{p(D)}$ on $p(D)$ is stimulated by the ordering \subseteq_D on D :

$$\text{only when } \left. \begin{array}{l} x_1 \rightarrow y_1 \subseteq_{p(D)} x_2 \rightarrow y_2 \\ x_1 \subseteq_D x_2 \text{ and } y_1 \subseteq_D y_2 \end{array} \right\}. \quad (2.7)$$

The set $p(D)$ is CPOS, because the least element $\perp_{p(D)} = (\perp_D \rightarrow \perp_D)$ is artificially introduced for it and any subset $X \subseteq p(D)$ has a least upper bound (supremum) $\bigsqcup^{p(D)} X \in [D \rightarrow D]$. The upper bound for the two elements $(x_1 \rightarrow y_1) \in p(D)$ and $(x_2 \rightarrow y_2) \in p(D)$ is calculated in the following way:

$$(x_1 \rightarrow y_1) \bigsqcup^{p(D)} (x_2 \rightarrow y_2) = \left(x_1 \bigsqcup^D x_2 \right) \rightarrow \left(y_1 \bigsqcup^D y_2 \right). \quad (2.8)$$

From the conditions (2.7), (2.8) it follows that the operator p remains monotonous, i.e., the directivity of the subsets $X \subseteq p(D)$ is not lost in transition from D to $p(D)$. In addition from the equation

$$p\left(\bigsqcup^D X\right) = \bigsqcup^{p(D)} p(X) \tag{2.9}$$

it follows that the operator p is continuous.

As the sets $r(D)$ and $p(D)$ are CPOS it is correct to expand the domain D of definition for the operators r and p in the following way:

$$r: \left\{ \begin{array}{l} r(D) \rightarrow r(D) \\ p(D) \rightarrow r(D) \end{array} \right\}, \quad p: \left\{ \begin{array}{l} r(D) \rightarrow p(D) \\ p(D) \rightarrow p(D) \end{array} \right\}. \tag{2.10}$$

A composition of the monotonous and continuous functions is the monotonous continuous function, too. The domain of values of the operators r and p on (2.10) remain CPOS and are determined as:

$$\left. \begin{array}{l} r(D) = r(D) + r \circ p(r(D)) + \dots \\ p(D) = p(D) + p \circ r(p(D)) + \dots \end{array} \right\}. \tag{2.11}$$

As a result of the multiple use of the operators r and p the boundaries \mathbf{R} and \mathbf{P} are obtained as fixed point of continuous monotonous mapping of complete partially ordered sets in themselves. The fixed point theorem for a complete lattice has been proved by Tarski (1981). A variant of the fixed point theorem for CPOS is presented in (Barendregt, 1981). The bounded sets \mathbf{R} and \mathbf{P} are recursively determined by the operators r and p on the initial CPOS D of entities names. On the boundaries \mathbf{R} and \mathbf{P} the operators turn into operations $\overset{\circ}{r}$ and $\overset{\circ}{p}$ and the formulae (2.11) become:

$$\left. \begin{array}{l} \mathbf{R} = \overset{\circ}{r}(\mathbf{R} + \mathbf{P}) \\ \mathbf{P} = \overset{\circ}{p}(\mathbf{R} + \mathbf{P}) \end{array} \right\}. \tag{2.12}$$

By disjunctive joining of \mathbf{R} and \mathbf{P} as CPOS the largest type \mathbf{Q} is obtained:

$$\mathbf{Q} = \mathbf{R} + \mathbf{P}. \tag{2.13}$$

The type is CPOS and contains the possible types \mathbf{R} and \mathbf{P} constructed by means of operations $\overset{\circ}{r}$ and $\overset{\circ}{p}$.

3. The relational approach to representation of abstract objects

DEFINITION 3.1. Let S is a set. T is an extent over S when T is a set of functions over S .

EXAMPLE 3.1. Fig. 1 shows extents over $S1=\{\text{ORDER_NR, VENDOR, CUSTOMER}\}$ and $S2=\{\text{VEND_NR, NAME, F_NAME, CHIEF}\}$. Extents T1 and T2 are shown in form of tables.

T1			
	ORDER_NR	CLERK	CUSTOMER
t_1 :	2002	16	043
t_2 :	2003	06	008
t_3 :	2017	06	043

T2				
	VEND_NR	NAME	F_NAME	CHIEF
t_4 :	16	Long	Nicole	03
t_5 :	06	Thomas	Pat	03
t_6 :	03	Vidoni	Rick	⊥

Fig. 1. An order extent T1 and a vendor extent T2.

The extent T1 consists of three functions:

$$t_1 = \{(\text{ORDER_NR}; 2002), (\text{CLERK}; 16), (\text{CUSTOMER}; 043)\},$$

$$t_2 = \{(\text{ORDER_NR}; 2003), (\text{CLERK}; 06), (\text{CUSTOMER}; 008)\},$$

$$t_3 = \{(\text{ORDER_NR}; 2017), (\text{CLERK}; 06), (\text{CUSTOMER}; 043)\}.$$

The extent T2 consists of three functions:

$$t_4 = \{(\text{VEND_NR}; 16), (\text{NAME}; \text{Long}), (\text{F_NAME}; \text{Nicole}), (\text{CHIEF}; 03)\},$$

$$t_5 = \{(\text{VEND_NR}; 06), (\text{NAME}; \text{Thomas}), (\text{F_NAME}; \text{Pat}), (\text{CHIEF}; 03)\},$$

$$t_6 = \{(\text{VEND_NR}; 03), (\text{NAME}; \text{Vidoni}), (\text{F_NAME}; \text{Rick}), (\text{CHIEF}; \perp)\}.$$

In example $\text{dom}(t_1)=\text{dom}(t_2)=\text{dom}(t_3)=S1$ and $\text{dom}(t_4)=\text{dom}(t_5)=\text{dom}(t_6)=S2$. Fig. 1 shows order and vendor extents relevant to a certain selling company at a particular moment. Extent information represents some entities or abstract information objects $Q1$ and $Q2$, abstract order and abstract vendor objects.

DEFINITION 3.2. Let QS is a set of abstract objects and Q is considered as element of set QS . g is a set function if g is function and for every $Q \in \text{dom}(g)$ $g(Q)$ is a set. If g is a *set function* and ν is a function over $\text{dom}(g) = QS$ then $\nu(Q)$ is an *extent over $g(Q)$* and ν is *state function over $\text{dom}(g)$* ; we will call ν the *object's state over g* .

For representation of objects we will use function ν which assigns for each object $Q \in QS$ an extent. In example at one particular moment $\nu_1(Q_1)=T_1$ and $\nu_1(Q_2)=T_2$. The state of object Q_1 or Q_2 at an other moment can be represented by an other function ν_2 over g_1 . We consider that state of objects in QS is time-varying. We do assume, though, that set function g_1 is time-invariant.

If g is a set function then set U of all time-varying and permissible states of $\text{dom}(g)$ is called the *extent universe over g* .

Let g is a set function and $g(Q) = S$. If T is a set of functions over S then S is called a *relation scheme of object Q* and T is called *Q – projection onto relation scheme S* . If P is an element of S then P is called the *P -property of object Q* . If $R \subseteq S$ then R is called the *R -property set of object Q* .

DEFINITION 3.3. An object Q is *projected* on property set R if for each state $\nu \in U$ over g and for each function $t \in \nu(Q)$ there is defined exactly one function $t|_R$ restricted to R , i.e.,

$$t|_R = \{(P; t(P)) | P \in R\} \subseteq t.$$

Let the order object Q_1 is projected on $R_1 = \{\text{VENDOR}\}$, $R_1 \subseteq g_1(Q_1)$ and the vendor object Q_2 is projected on $R_2 = \{\text{NAME}, \text{F_NAME}, \text{CHIEF}\}$, $R_2 \subseteq g_1(Q_2)$. For state ν_1 over g_1 , $\nu_1(Q_1)=T_1$, and for

$$t_1 = \{(\text{ORDER_NR}; 2002), (\text{CLERK}; 16), (\text{CUSTOMER}; 43)\} \in T_1$$

is defined function

$$t_7 = \{(\text{CLERK}; 16)\} \subseteq t_1.$$

For state ν_1 over g_1 , $\nu_1(Q_2)=T_2$, and for

$$t_4 = \{(\text{VEND_NR}; 16), (\text{NAME}; \text{Long}), (\text{F_NAME}; \text{Nicole}), (\text{CHIEF}; 03)\} \in T_2$$

is defined function

$$t_8 = \{(\text{NAME}; \text{Long}), (\text{F_NAME}; \text{Nicole}), (\text{CHIEF}; 03)\} \subseteq t_4.$$

By Q_1 project onto R_1 from T_1 it may be created a new extent, let T_{11} in Fig. 2, and by Q_2 project onto R_2 from T_2 it may be created a new extent, let

T_{21} in Fig. 2. Project operator in the relational databases theory is defined as unary operator on relations and the projection of relation T onto R is written $\pi_R(T)$.

	T11		T21															
$t_7 :$	<table style="border-collapse: collapse; width: 100%;"> <tr><th style="border: none;">CLERK</th></tr> <tr><td style="border: none;">16</td></tr> <tr><td style="border: none;">06</td></tr> </table>	CLERK	16	06	$t_8 :$	<table style="border-collapse: collapse; width: 100%;"> <tr><th style="border: none;">NAME</th><th style="border: none;">F_NAME</th><th style="border: none;">CHIEF</th></tr> <tr><td style="border: none;">Long</td><td style="border: none;">Nicole</td><td style="border: none;">03</td></tr> <tr><td style="border: none;">Thomas</td><td style="border: none;">Pat</td><td style="border: none;">03</td></tr> <tr><td style="border: none;">Vidoni</td><td style="border: none;">Rick</td><td style="border: none;">⊥</td></tr> </table>	NAME	F_NAME	CHIEF	Long	Nicole	03	Thomas	Pat	03	Vidoni	Rick	⊥
CLERK																		
16																		
06																		
NAME	F_NAME	CHIEF																
Long	Nicole	03																
Thomas	Pat	03																
Vidoni	Rick	⊥																

Fig. 2. Extents defined by projections $T_{11}=\pi_{R_1}(T_1)$, $T_{21}=\pi_{R_2}(T_2)$.

Let $g(Q) = S$, $\nu(Q) = T$, $R \subseteq S$ and $P \in R$. A set R is called a *set with identifying property* for object Q if different elements of T have different values for at least one P -property on R . Formally an identifying set R can be defined using Definition 3.4.

DEFINITION 3.4. Let t is function and R is a set. t is *restricted to R* , i.e., $t|_R$, if $t|_R = \{(x, y) \in t \mid x \in R\}$. R is a set with an identifying property for object Q if for each $t \in T$ and each $t' \in T$ when $t|_R = t'|_R$ then $t = t'$. R is a key set or simply a *key* of Q if no proper subset R' of R shares identifying property for each $\nu(Q) \in U$.

A set of the keys of object Q we will write $KS(Q)$. In Example 1 $R_1 = \{\text{ORDER_NR}\}$ is a key of order object Q_1 and $KS(Q_1) = \{R_1\}$; $R_2 = \{\text{VEND_NR}\}$ and $R_3 = \{\text{NAME, F_NAME}\}$ are keys of vendor object Q_2 and $KS(Q_2) = \{R_2, R_3\}$.

4. An abstract object as structural subtype. Analysis of the structural entities names was based (Paradauskas, 1993) on the initial assertion that name universum D is a complete partially ordered set (CPOS). For this purpose the least element \perp such that for all $x \in D$ $\perp \sqsubseteq x$ was introduced and for any directed subset $X \subseteq D$ the upper boundary point $\bigsqcup X \in D$ was used. As result of the multiple use of monotonous and continuous operator of constructing positioned subsets of D and monotonous and continuous operator constructing a functional set on D there were created two boundary sets: complete partially ordered set

of lists \mathbf{R} and complete partially ordered set of functions \mathbf{P} . By disjunctive joining of \mathbf{R} and \mathbf{P} the largest type \mathbf{Q} , $\mathbf{Q} = \mathbf{R} + \mathbf{P}$ was obtained. The type \mathbf{Q} is a complete partially ordered set and contains the possible types \mathbf{R} and \mathbf{P} . In this section we will analyse the object Q as a subset of \mathbf{Q} .

DEFINITION 4.1. Let \mathbf{IN} denotes the set of natural numbers. r is m -tuple if r is a function over $\{k \in \mathbf{IN} \mid k < m\}$. Notation $r = \langle p_1; p_2; \dots; p_m \rangle$ denotes m -tuple r defined by $r(0) = p_1, r(1) = p_2, \dots, r(m-1) = p_m$. R is a sequence if there exists $m \in \mathbf{IN}$ such that R is m -tuple; natural number m is called the length of R .

DEFINITION 4.2. Let Q is a set, g is a set function. The sequence $\langle P_1; P_2; \dots; P_m \rangle$ is called the P -properties sequence of Q if for all $i, P_i \in g(Q)$.

DEFINITION 4.3. Let Q is a set, g is a set function, ν is state function over g . An m -tuple $r = \langle p_1; p_2; \dots; p_m \rangle$ is a r -instance of P -properties sequence $\langle P_1; P_2; \dots; P_m \rangle$ if for each $i = 1, 2, \dots, m, p_i = t(P_i)$ and $P_i \in g(Q)$. A set $\{\langle t(P_1); t(P_2); \dots; t(P_m) \rangle \mid t \in \nu(Q)\}$ of r -instances is Q -projection onto P -properties sequence $\langle P_1; P_2; \dots; P_m \rangle$ at the "moment" ν . Naturally Q -projection (as well as Q -volume) depends on ν but for simplicity we will note it in this section as $Q.R = \langle P_1; P_2; \dots; P_m \rangle$ and call the inner relationship of object Q . When it is understood we write R for a P -property subset of $g(Q)$ and $Q.R$ for internal relationship, created onto sequence $\langle P_1; P_2; \dots; P_m \rangle$, where $P_i \in R; i = 1, 2, 3, \dots, m; m = |R|$. An R -instance of P -properties sequence $\langle P_1; P_2; \dots; P_m \rangle$ we will call the instance of relationship $Q.R$ and will write $r \in Q.R$. Note that such an ordering of P -properties adds nothing to the information content of extents over $g(Q)$. The representation of inner relationship $Q.R, R \subseteq g(Q)$, in form of tables is more natural than representation of extents over $g(Q)$. Consider the inner relationship $Q.R = \langle P_1; P_2; \dots; P_m \rangle$. We will say that inner relationship $Q.R$ covers the properties $P_i, i = 1, 2, \dots, m$ and P_i is a part of $Q.R$, written $P_i \in Q.R$.

DEFINITION 4.4. Let $Q \in QS, g$ is a set function, $g(Q) = S, \nu$ is state function over $g, Q.S$ is Q -projection onto P -properties sequence which covers the properties $P \in S$. If $R \subseteq S, P_i \in R, Q.R = \{\langle t(P_1); t(P_2); \dots; t(P_n) \rangle \mid t \in \nu(Q)\}$ is both subrelationship of $Q.S$ and Q -projection onto P -properties

sequence which covers the properties $P_i \in R$.

Let $Q \in QS$; g is a set function, $g(Q) = S$, $R \subseteq S$; $\hat{R}, \hat{R}' \in KS(Q)$ and $Q.S, Q.R, Q.\hat{R}, Q.\hat{R}' \subseteq R$ are Q -projections onto P -properties sequences. Consider Q as a subset of largest structural type Q , i.e., $Q \subseteq Q$. Since Q is subset of type Q hence new subtype Q will be defined using Q -commutational diagram (Fig. 3).

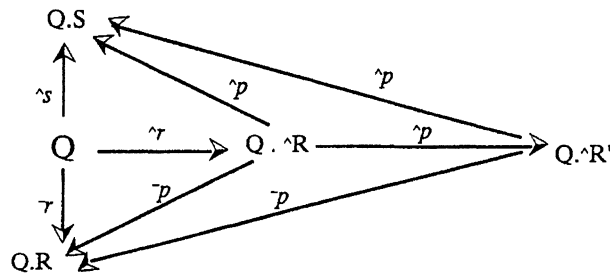


Fig. 3. Q -commutational diagram.

Mappings

$$\left. \begin{aligned}
 \hat{s}: & \quad Q \rightarrow Q.S, \\
 \hat{r}: & \quad Q \rightarrow Q.\hat{R}; \\
 \hat{p}: & \quad \left\{ \begin{aligned}
 & Q.\hat{R} \rightarrow Q.\hat{R}', \\
 & Q.\hat{R} \rightarrow Q.S, \\
 & Q.\hat{R}' \rightarrow Q.S; \hat{p} \subseteq P,
 \end{aligned} \right\}
 \end{aligned} \right\} \quad (4.1)$$

in Q -commutational diagram (Q - CD) are bijective, and mappings

$$\left. \begin{aligned}
 \bar{p}: & \quad \left\{ \begin{aligned}
 & Q.\hat{R} \rightarrow Q.R, \\
 & Q.\hat{R} \rightarrow Q.R; \bar{p} \subseteq P;
 \end{aligned} \right\} \\
 \bar{r}: & \quad Q \rightarrow Q.R,
 \end{aligned} \right\} \quad (4.2)$$

are subjective.

The subset $Q \subseteq Q$ is entirely formal and is defined as domain of subsets constructing function \hat{r} , \hat{s} and \bar{r} , which satisfies requirements of Q -commutational diagram. Real thing or process $q \in Q$ is characterized by P -property values. The bijective mapping \hat{s} assigns to each object–instance q the m -tuple $\hat{s}(q) = (t(P_1); t(P_2); \dots; t(P_m)) \in Q.S$, $P_i \in S$.

For representation of objects–instances $q \in Q$ we have sufficient for use one key relationship $\hat{R} \in KS(Q)$. The bijective mapping \hat{r} assigns to each

object-instance $q \in Q$ the z -tuple

$$\begin{aligned} \hat{r}(q) &= (t(P_1); t(P_2); \dots; t(P_z)) \in Q.\hat{R} \subseteq \mathbf{R}, \\ P_j &\in \hat{R}; \hat{R} \in KS(Q), \hat{R} \subseteq S. \end{aligned}$$

The surjective mapping \hat{r} assigns to each object-instance $q \in Q$ the relationship-instance $\hat{r}(q) \in Q.\hat{R} \subseteq \mathbf{R}$, where $Q.\hat{R}$ is subrelationship of $Q.S$.

DEFINITION 4.5. Let we have three sets A, B, C and two mappings $a: A \rightarrow C$ and $b: B \rightarrow C$. Two elements $x \in A$ and $y \in B$ are equivalent, i.e., $x \sim y$, if $a(x) = b(y)$. Two sets A and B are equivalent, i.e., $A \sim B$, if for each element $x \in A$ exists exactly one $y \in B$ such that $x \sim y$ and on the contrary: for each $y' \in B$ there exists exactly one $x' \in A$ such that $y' \sim x'$.

Naturally, the sets Q and $Q.S$ in Q - CD are equivalent, i.e., $Q \sim Q.S$, by means of bijective mappings \hat{r}, \hat{s} and $\hat{p}: Q.\hat{R} \rightarrow Q.S \subseteq \mathbf{P}$. The sets Q and $Q.\hat{R}$ are equivalent by means \hat{r}, \hat{r} and $\hat{p}: Q.\hat{R} \rightarrow Q.\hat{R} \subseteq \mathbf{P}$ too. On the analogy $Q.\hat{R} \sim Q.\hat{R}'$. A composition $\hat{p} \circ \hat{r}$ of two bijective mappings is bijective mapping and $Q \sim Q.\hat{R}'$. The equivalences $Q \sim Q.\hat{R}$ and $Q \sim Q.S$ is the *first structural feature* of subset $Q \subseteq Q$. We can consider the set Q in Q - CD as subset $Q.S \subseteq \mathbf{R}$ or as subset $Q.\hat{R} \subseteq \mathbf{R}$.

Let Q - CD is created for all $Q.\hat{R}_j \subseteq \mathbf{R}, \hat{R}_j \in KS(Q), j = 1, 2, \dots, k$ and $Q.R = \langle P_1; P_2; \dots; P_n \rangle \subseteq \mathbf{R}$ (Fig. 4). An object $Q \in QS$, a new subtype $Q \subseteq Q$ will be defined using *second structural feature*: an object $Q \subseteq Q$ satisfies Q - CD functional constrains, given as \hat{p} in Eq. (4.1) and \hat{r} in Eq. (4.2).

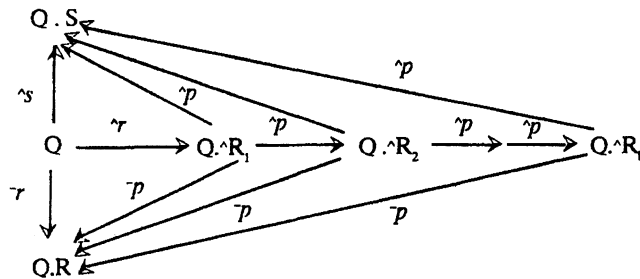


Fig. 4. Q - CD creation for all key relationships of Q and $R \subseteq S$.

Note, that an object Q can be interpreted as compound functional dependence $(Q.\hat{R}_1; Q.\hat{R}_2; \dots; Q.\hat{R}_k) \rightarrow Q.R; \hat{R}_j, R \subseteq S, S = g(Q)$, with equivalent left sides $\hat{R}_j, j = 1, 2, \dots, k$ (Maier, 1983). If $R = S - \{P \mid P \in \hat{R}, \hat{R} \in KS(Q)\}$ mappings (4.1) and (4.2) may be interpreted as compound functional dependence with reduced right side. But there is an essential difference: every bijective mapping $\hat{p}: Q.R \rightarrow Q.R' \subseteq P$ is a surjective mapping $\bar{p}: Q.R \rightarrow Q.R' \subseteq P$; each surjective mapping $\bar{p}: Q.R \rightarrow Q.R' \subseteq P$ is functional dependence $Q.R \rightarrow Q.R' \subseteq P$. The reverse assertions are wrong.

If object $Q \subseteq Q$ satisfies both features, formulated using Q -CD we will call such object Q *the structural object*.

The subtype Q will be restricted in addition using *3-th feature*, the “volume”feature of $Q \subseteq Q$, i.e., when the object–instance q is ascribed to set Q , so it is assumed that q must have only relevant P -properties values:

$$t(P) \in D \quad \text{and} \quad t(P) \neq \perp, \quad P \in Q.S, \quad S = g(Q),$$

where D is entity name universum.

If for q there exists such $P \in S$ that $t(P) = \perp$ we will say that property P is not relevant to q . So, we will consider a set Q consisting of such elements that for every element $q \in Q$ the properties $P \in Q.S$ are relevant.

Example 4.1. In Example 4.1 CHIEF–property is relevant for two object–instances q_4 and q_5 , which are represented by two 4-tuples correspondingly

$$\langle 16; \text{Long}; \text{Nicole}; 03 \rangle \quad \text{and} \quad \langle 06; \text{Thomas}; \text{Pat}; 03 \rangle.$$

These 4-tuples make up the clerk–object projection onto property sequence

$$\langle \text{VEND_NR}, \text{NAME}, \text{F_NAME}, \text{CHIEF} \rangle.$$

The object–instance q_6 represented by one 3-tuple $\langle 03; \text{Vidoni}; \text{Rick} \rangle$ makes up the shief–object projection onto property sequence

$$\langle \text{VEND_NR}, \text{NAME}, \text{F_NAME} \rangle.$$

Fig. 5 shows order–object projection $Q1.S1$ and two projections $Q21.S21$ and $Q22.S22$, which are made for new object, clerk and chief objects.

Q1.S1			
	ORDER_NR	CLERK	CUSTOMER
q1:	2002	16	043
q2:	2003	06	008
q3:	2017	06	043

Q21.S21				
	VEND_NR	NAME	F_Name	CHIEF
q4:	16	Long	Nicole	03
q5:	06	Thomas	Pat	03

Q221.S22			
	VEND_NR	NAME	F_NAME
q6:	03	Vidoni	Rick

Fig. 5. An example of projection of three objects with relevant properties.

If all objects $q \in Q$, $Q \in QS$, $Q \subseteq \mathbf{Q}$ are characterized by relevant property values then notation $t(P) = \perp$ can be reserved for object $q \in Q$ on situation when P -property value for object q is *not known*.

A structural subtype Q , $Q \subseteq \mathbf{Q}$, $Q \in QS$, satisfying all considered types of requirements, i.e., structural and the “volume” requirements we will call the relevant (to problem area) object Q .

We will use the graphical notation of objects $Q \in QS$, $Q \subseteq \mathbf{Q}$. In Example 4.1 the $Q1$, $Q21$ and $Q22$ can be represented as shown in Fig. 6.

5. Extent functions and semantics of external relationship. It will be convenient to have names for objects and its components, i.e., for inner relationships and for object properties, in order to be able refer to them inside formal problem area specification language. For this purpose the following injective mapping will be used:

$${}^{-}d: \begin{cases} QS \rightarrow D; \\ Q.RS \rightarrow D, Q \in QS; \\ g(Q) \rightarrow D, Q \in QS. \end{cases} \quad (5.1)$$

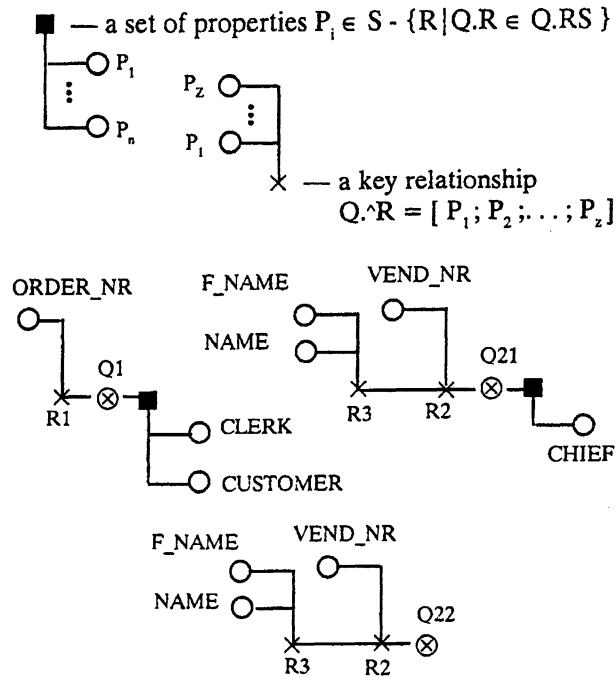


Fig. 6. Graphical notation of objects Q1, Q21 and Q22 from Example 4.2 (graphically: \otimes denoted an object; \circ – a property of the object).

Here $Q.RS$ is a set of inner relationships $Q.R$ including key relationships of Q ; this set should be chosen for specification.

According to injectivity of τ -d-mappings in (5.1) all abstract objects have unique names; all internal relationships inside the same object have unique names; all properties of the same object have unique names. Note that we allow that relationships and properties of different objects may have the same name; the same name may designate object, its relationship and its property inside the same object.

What should be chosen sets QS , $Q.R$ and $g(Q)$ and what corresponding names of their elements should be specified? The relevance of objects and their properties must be determined by the end users of the information system. The relevance of internal relationships follows from the possibility to specify external relationships between the pairs of objects – as much as possible. A subject

of this section is to define the conditions of external relationships concerned below.

A variant of objects and its components specification is given in Fig. 7.

Let be given the surjective mapping \bar{d} from (5.1). We will write $d_1 = \bar{d}(Q)$ for object $Q \in QS$; we will write $d_1.d_2 = \bar{d}(Q).\bar{d}(R)$ for object inner relationship $Q.R$ and $d_1.d_2.d_3 = \bar{d}(Q).\bar{d}(R).\bar{d}(P)$ for $Q.R.P$ (i.e., for $P \in Q.R$) or $d_1..d_3 = \bar{d}(Q).. \bar{d}(P)$ for $P \in g(Q)$.

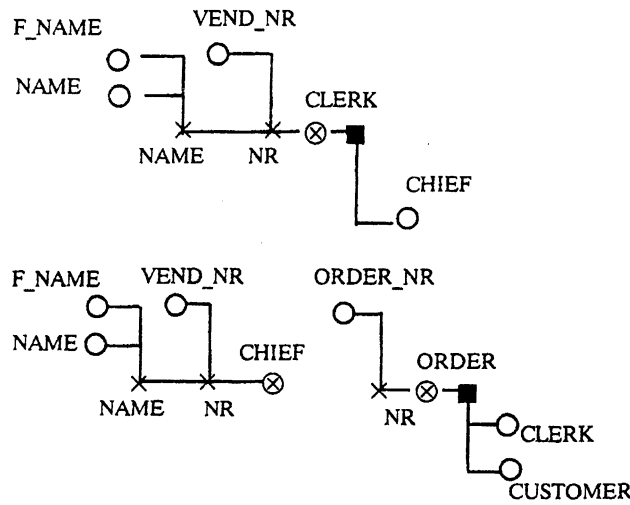


Fig. 7. An example of three object with specified objects, internal relationship and property names.

EXAMPLE 5.1. Consider two objects from Example 3.1. CLERK and ORDER, and internal relationship ORDER.TO and CLERK.NR. We confirm that every CLERK who has received at least one ware's order is CLERK (Fig. 8). Formally ORDER.TO is ORDER – object' projection onto property sequence TO which covers the property CLERK, i.e., $CLERK \leftarrow ORDER.TO, CLERK \in g(ORDER)$.

Let $U1$ be an extent universum over set function $g1$ introduced for our example $CLERK.NR \in CLERK.RS$ and $ORDER.TO \in ORDER.RS$. If for every $\nu \in U1$ and every CLERK-value in extent ν (ORDER) also appears as VEND_NR-value in ν (CLERK), i.e., at each moment every clerk having received at least one order belongs to list of clerks working on selling company

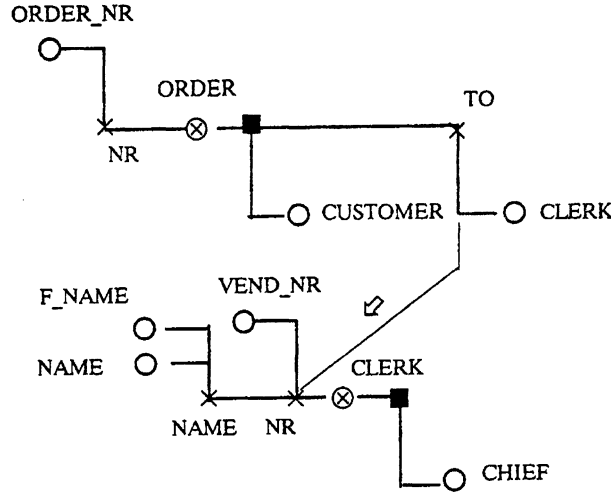


Fig. 8. Two objects CLERK and ORDER with external relationship ORDER.TO ⇒ CLERK.NR

at the same moment, then this induces for order pair (ORDER.TO; CLERK.NR) a containment function $F1(\nu)$ from ORDER.TO into CLERK.NR.

DEFINITION 5.1. Let consider general situation in which: $O.R \in O.RS$ and $Q.R' \in Q.RS$ are two internal relationships with equal length n , $O \neq Q$;

$$P_i \leftarrow O.R; P'_i \leftarrow Q.R'; 1 \leq i \leq n; \tag{5.2}$$

$$g \text{ is set function, } R \subseteq g(O), R' \subseteq g(Q); \tag{5.3}$$

$$\nu \text{ is state function over } \text{dom}(g); \tag{5.4}$$

$$U \text{ is extent universum over } g. \tag{5.5}$$

$F(\nu)$ is containment function for order pair $(O.R; Q.R') \in O.RS \times Q.RS$ if for all $\nu \in U$ and all $t \in \nu(O)$ when $\langle t(P_1); t(P_2); \dots; t(P_n) \rangle \in O.R$, then exists $t' \in \nu(Q)$ such that $\langle t'(P'_1); t'(P'_2); \dots; t'(P'_n) \rangle \in Q.R'$ and $t'(P'_i) = t(P_i); i = 1, 2, \dots, n$.

According to this definition containment function $F(\nu)$ is induced by the time-invariant relation $Q.R \subseteq Q.R'$. Extent function F for $(O.R; Q.R') \in O.RS \times Q.RS$ over U is defined by

$$F(\nu) = \{ (t_{|R}; t'_{|R'}) \in \pi_R(\nu(O)) \times \pi_{R'}(\nu(Q)) \mid t(P_i) = t'(P'_i); P_i \leftarrow O.R, P'_i \leftarrow Q.R'; i = 1, 2, \dots, n \}. \tag{5.6}$$

Containment function $F(\nu)$ is function and this proof is almost trivial. We still have to check that $F(\nu) \in \pi_R(\nu(O)) \rightarrow \pi_{R'}(\nu(Q))$, $\text{dom}(F(\nu)) = \pi_R(\nu(O))$ and $\text{rng}(F(\nu)) \subseteq \pi_{R'}(\nu(Q))$ (see Fig. 9). According to Definition 3.3 the projection operator π_R is function-valued function over $\nu(O)$ assigning to each function $t \in \nu(O)$ exactly one function $t|_R = \{(P; t(P)) \mid P \leftarrow R\}$. By analogy operator $\pi_{R'}$ assigns to each $t' \in \nu(Q)$ exactly one function $t'|_{R'} = \{(P'; t'(P')) \mid P' \leftarrow R'\} \in \pi_{R'}(\nu(Q))$. $\pi_{R'}(\nu(Q))$ is set (not multiset) because of the relation axiom, and $F(\nu)$ is function-valued function.

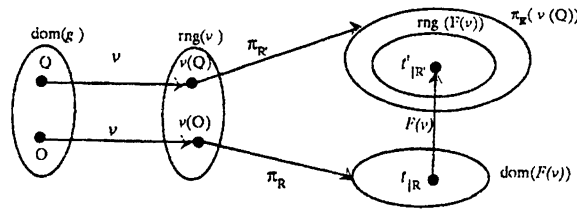


Fig. 9. Illustration of containment function $F(\nu)$ for the pair $(O.R; Q.R')$.

The containment function $F(\nu)$ for pair $(O.R; Q.R')$ induced by relation $(O.R; Q.R')$ is injective function for all $\nu \in U$. The extent function F specified in conceptual scheme for $(O.R; Q.R')$ is function-valued function and we will call it the *external relationship*.

Let we refer to objects and to their components by entities names from D and let the pair $(O.R; Q.R')$ induces the containment function over U ; $P_i \leftarrow O.R$, $P'_i \leftarrow O.R'$. We will $d_1.d_2.d_3 = \text{"}d(O).\text{"}d(R).\text{"}d(P_i)$ and $d'_1.d'_2.d'_3 = \text{"}d(Q).\text{"}d(R').\text{"}d(P'_i)$ call the *compound names* and say that the compound name $d_1.d_2.d_3 \sqsubseteq d'_1.d'_2.d'_3$, and $d_1.d_2 \sqsubseteq d'_1.d'_2$. From $d_1.d_2.d_3 \sqsubseteq d'_1.d'_2.d'_3$, follows $d_1..d_3 \sqsubseteq d'_1..d'_3$. In our last example $\text{ORDER.TO.CLERK} \sqsubseteq \text{CLERK.NR.VEND_NR}$, $\text{ORDER.TO} \sqsubseteq \text{CLERK.NR}$ and $\text{ORDER..CLERK} \sqsubseteq \text{CLERK..VEND_NR}$.

6. Various cases of external relationship implication. Fig. 10 shows four cases of the containment function application.

The case in Fig. 10a is the general case. It was defined by (5.6) under the condition (5.2–5.5). The rest cases are particular. Let consider 10b case, in

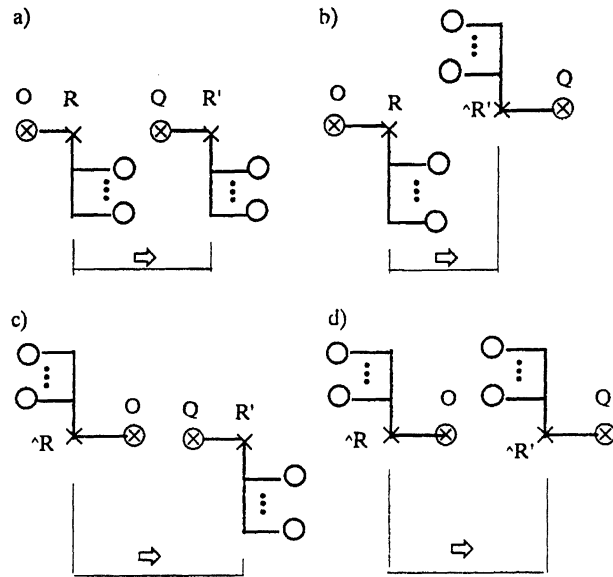


Fig. 10. Four cases of external relationship implication.

which $O.R \Rightarrow Q.^R R'$.

DEFINITION 6.1. A is an object valued function if A is function and for all $x \in \text{dom}(A)$ $A(x)$ is a structural object.

Let $\nu \in U$ is state function over $\text{dom}(g)$, g is set function; $R \subseteq g(O)$, $^R R' \subseteq g(Q)$; $\nu, Q \in QS$; $O, Q \subseteq Q$. The order pair $(O.R; Q.^R R') \in O.RS \times Q.KS$ induces for each $\nu \in U$ the object valued function $L(\nu)$ defined by

$$L(\nu) = \{(t|_R; q') \in \pi_R(\nu(O)) \mid t(P_i) = t'(P'_i); P_i \in O.R, P'_i \in Q.^R R'; i = 1, 2, \dots, n\}. \quad (6.1)$$

$L(\nu)$ is injective function because $\text{dom}(L(\nu)) = \pi_R(\nu(O))$, $\text{rng}(L(\nu)) \subseteq Q$ and because of the bijection $^R r: Q \rightarrow Q.^R R'$ in $Q-CD$.

DEFINITION 6.2. a is arrangement of ν if a is function over $\text{dom}(\nu)$ and for all $Q \in \text{dom}(\nu)$ a is bijective function onto $\nu(Q)$. l is enumeration of Q if l is injective sequence and $\text{rng}(l) = Q$. o is an ordering function for a if o is function over $\text{dom}(a) = QS$ and for all $Q \in \text{dom}(a)$ $o(Q)$ is an enumeration of $\text{dom}(o(Q)) = Q$.

Note that an enumeration of Q after an arrangement of state function ν adds nothing to the information content of $\nu(Q)$ but this process is essential for the formal basing of the P -properties generalization and for the implementation of main rule in conceptual scheme integration. After the substitution for each $(t_{|R}; q') \in L(\nu)$ the number notation of q' from set Q instead of $t_{|R} \in \pi_R(\nu(O))$ we write $O.R. \Rightarrow Q$, $O.R. = \langle P^* \rangle$ (see Fig. 11a) or $O..P^* \Rightarrow Q$ (see Fig. 11b). Here P^* is new structural P -property of O . The function L is function valued function over U and is extent function. It may be specified in conceptual schema.

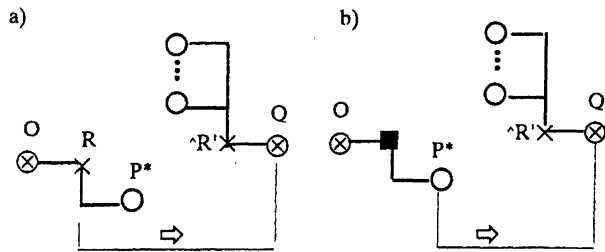


Fig. 11. The specification of extent function induced by $O.R. \Rightarrow Q.\hat{R}'$.

EXAMPLE 6.1. Fig. 12 shows the modified order extent after enumeration of clerk object at the moment ν and by the use of new P -property TO.

The order extent was created according to the integrity rule shown in Fig. 13.

Let consider Fig. 10c case in which $O.\hat{R} \Rightarrow O.R'$. Containment function (see formula 5.6) for $(O.\hat{R}; Q.R') \in O.KS \times Q.RS$ can be modified and defined for every ν by

$$F'(\nu) = \{(q; t'_{|R'}) \in O \times \pi_{R'}(\nu(Q)) \mid t(P_i) = t'(P'_i); P_i \in O.R, P'_i \in Q.R'; i = 1, 2, \dots, n\}. \tag{6.2}$$

$F'(\nu)$ is modified containment function defined over $O \subseteq Q$. $F'(\nu)$ is injective function because of the bijection $\hat{r}: O \rightarrow O.\hat{R}$ in O -CD. $F'(\nu)$ is function-valued function because $\text{rng}(F'(\nu)) \subseteq \pi_{R'}(\nu(Q))$. The function F' is function valued function over U and is extent function. It may be specified in conceptual scheme (see Fig. 14b).

	ORDER_NR	TO	CUSTOMER
q_1 :	2002	1	043
q_2 :	2003	2	008
q_3 :	2017	2	043

	VEND_NR	NAME	F_Name	CHIEF
$q_4 \rightarrow 1$:	16	Long	Nicole	03
$q_5 \rightarrow 2$:	06	Thomas	Pat	03

Fig. 12. The modified order extent with new property TO of ORDER-object.

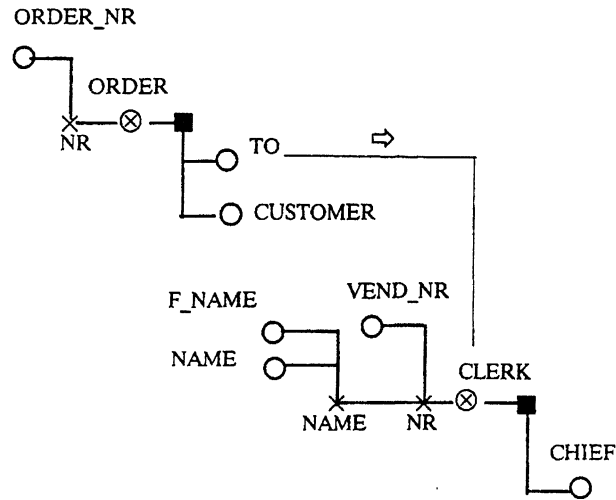


Fig. 13. The integrity rule ORDER..TO ⇒ CLERK

Let us consider external relationship $O \hat{=} R \Rightarrow Q \hat{=} R'$. Object valued function (see formula 6.1) can be modified and defined for every ν by

$$L'(\nu) = \{(q; q') \in O \times Q \mid t(P_i) = t'(P'_i); \\ P_i \in O \hat{=} R, P'_i \in Q \hat{=} R'; \quad i = 1, 2, \dots, n\}. \quad (6.3)$$

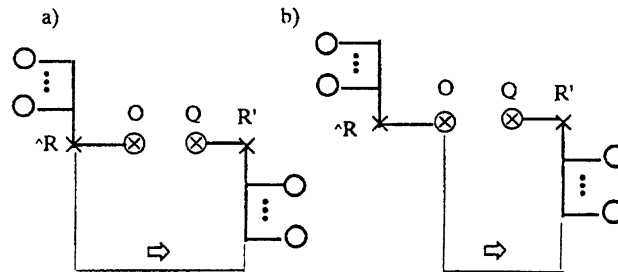


Fig. 14. The specification of extent function $O \Rightarrow Q.R'$ induced by $Q.\hat{R} \Rightarrow R'$.

$L'(\nu)$ is injective function because $\text{dom}(L'(\nu)) = O$, $\text{rng}(L'(\nu)) \subseteq Q$ and because of the bijections $\hat{r}: O \rightarrow O.\hat{R}$ in $O\text{-}CD$ and $\hat{r}': Q \rightarrow Q.\hat{R}'$ in $Q\text{-}CD$. $L'(\nu)$ is *modified object valued function*.

The function L' is function valued function over U and is extent function. It may be specified in conceptual schema (see Fig. 15b).

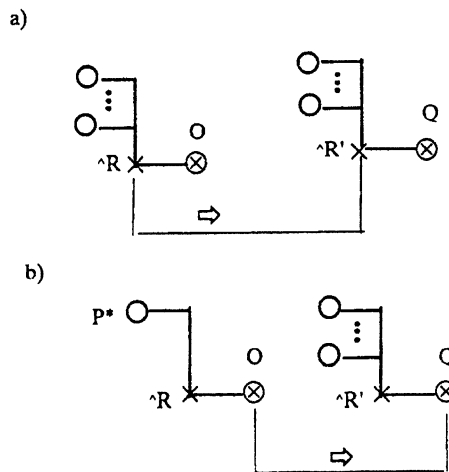


Fig. 15. The specification of extent function $O \Rightarrow Q$ induced by $O.\hat{R} \Rightarrow Q.\hat{R}'$.

7. Concluding remarks. The proposed approach to the object system modeling is characterized by four main advantages.

1. Reflexiveness is an important feature of the given typing approach, and from this it is possible to assign the types by means of design operators r and p to the initial CPOS D without any initial types and pre-typing. The transition from the initial set D to bounded sets \mathbf{R} and \mathbf{P} enables a set of entity names to be considered as structural types with respect to isomorphism: any structural entity is either a list or a function (2.12) and identified reflexively and cross-recursively (2.11) by means of monotonous and continuous operators r and p .

2. The category of the same entity may have three different interpretations: entity TO in context ORDER may be considered as property ORDER..TO; in context CLERK may be considered as object CLERK or as relationship CLERK.NR. The flexible interpretation of entities category is essential for distributed modelling of large information system.

3. For the description of Q -object structure it was used the Q -commutational diagram with surjective and bijective functions like the compound functional dependency. The Q - CD serves for a purpose of object structure normalization; it may be reduced and minimized. Constraints of Q - CD eliminate dual uncertainty or *unknown/or not adequate* of P -property values.

4. Surjections and bijections between Q -projections inside object Q and external bijections between objects–projections of a different objects pair allow to use common mathematical approach to conceptual scheme integration and manipulation: connections between different scheme components (objects, inner relationships, properties) are functions compositions of various types. Computed composition may be “degenerating” into partial functional dependence, into the weakest connection type.

REFERENCES

- Ambrazevitchius, E., and B. Paradauskas. (1983). The method of knowledge representation for semiological design of complex system models. *Cybernetics Problems: Situational Management and Semiological Modelling*. State Academy of Sciences, Moscow. pp. 3–13 (in Russian).
- Barendregt, H.P. (1981). The lambda calculus: its syntax and semantics. *Studies in Logic and the Foundations of Mathematics*, 103, North-Holland.
- Brachman and H.J. Levesque (Eds) *Readings in Knowledge Representation*. Morgan

- Kaufmann, Los Altos.
- Boman, M., J. Bubenko, jr., P. Johannesson and B. Wangler (1993). *Models, Concepts, and Information: an Introduction to Conceptual Modelling for Information Systems Development*. Royal Institute of Technology and Stockholm University, Sweden, Oktober.
- Booch, G. (1991). *Object-Oriented Design*. Benjamin/Cummings, Redwood City.
- Chen, P.S. (1976). The entity-relationship model – towards a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9–36.
- Dardenne, A. van Lamsweerde and S. Fickas (1993). Goal-directed requirements acquisition. *Science of Computer Programming*, 20, 3–50.
- DeMarco, T. (1979). *Structured Analysis and Systems Specifications*. Yourdon Press, Englewood Cliffs, New Jersey.
- Hull, R. (1987). A survey of theoretical research on typed complex database objects. In J. Paredaens (Ed.), *Databases*. Academic Press, London.
- ISO. *Concepts and Terminology for the Conceptual Schema and the Information Base*. ISO/TC9/SC5/WG3, 1982.
- Maier (1983). *The Theory of Relational Databases*. Computer Science Press, Rockville.
- Paradauskas, B. (1993). Conceptual and relational schemes of entities. *Informatica*, 4(3–4), 351–359.
- Paradauskas, B. (1994). Denotational semantics of objects relationship. *Proc. of the Baltic Workshop on National Infrastructure Databases: Problems, Methods, Experiences*. Trakai, Lithuania. pp. 230–241.
- Rumbaugh, J., et al. (1991). *Object-Oriented Modeling and Design*. Prentice-Hall International, Inc., Englewood Cliffs, New Jersey.
- Sciore, E. (1980). The universal instance and database design. *Doctor Diss*, Princeton Univ., Princeton, NJ.
- Tarski, A. (1981). A lattice-theoretical fixed point theorem and its syntax and semantics. *Studies in Logic and Foundation of Mathematics*, 103, North-Holland.
- Wangler, B. (1993). Contributions to functional requirements modelling. *Doctoral Thesis*. Royal Institute of Technology and Stockholm University, Sweden.
- Yourdon, E. (1989). *Modern Structured Analysis*. Yourdon Press, Englewood Cliffs, New Jersey.

Received October 1995

B. Paradauskas received the degree of Doctor of Technical Sciences from the Kaunas Polytechnic Institute in 1974. He is head of the Information Systems department of Kaunas University of Technology. His research interests include conceptual modelling of data bases.

**KONCEPCINIS OBJEKTŲ–SĄRYŠIŲ–SAVYBIŲ POŽIŪRIS:
TRYS SKIRTINGOS VIENOS ESYBĖS INTERPRETACIJOS**

Bronius PARADAUSKAS

Straipsnyje pateiktas esybių vardų struktūrinis tipizavimo būdas, kurio esybė gali būti interpretuojama kaip objektas, kaip objekto vidinis sąryšis arba kaip objekto savybė. Išnagrinėta objektų išorinių sąryšių semantika, panaudojant įvairias tipizuotas funkcijas.