Event Driven Objects

V.E.Wolfengagen*
Vorotnikovskiy per., 7, bld. 4
JurInfoR-MSU Institute for Contemporary Education
Moscow, 103006, Russia
e-mail: `vew@jmsuice.msk.ru`

Abstract

A formal consideration in this paper is given for the essential notations to characterize the object that is distinguished in a problem domain. The distinct object is represented by another idealized object, which is a schematic element. When the existence of an element is significant, then a class of these partial elements is dropped down into actual, potential and virtual objects. The potential objects are gathered into the variable domains which are the extended ranges for unbound variables. The families of actual objects are shown to be parameterized with the types and events. The transitions between events are shown to be driven by the scripts. A computational framework arises which is described by the commutative diagrams.

Key words: concept, data object, event, individual, metadata object, state, script, type, variable domain

Introduction

The language for a computational environment depends on the means of data object identification.

The consideration below is given more formal. The essential notations are summarized to characterize the object that is distinguished in the problem domain. The term 'problem domain' is enriched to capture the object of current research or investigation. In particular, the database, mathematical theory, programming system etc. are the problem domains.

The distinct object is represented by another idealized object, which is called an element. For convenience this representation is also called object. The representation is fixed within the language framework by the formal description. Thus, the term 'element, that represents the generic object' would be replaced by the term 'description-as-object'.

Furthermore the attention of the researcher is paid to the analysis of the representation currently mentioned. There is no need in formal distinction between 'object' and 'element'. By need, to resolve the possible misunderstanding the object from the problem domain and the object as its representation would be separated (recall that the representation generates the objects in a mathematical sense). The consideration is not restricted to a class of the total elements. When the existence of an element is significant, then a class of the partial elements is covered. Therefore the possible elements, or objects, are included and this kind of objects is possible with respect to some a priori theory.

Consider an image of some problem domain that contains the objects and their relationships. This representation becomes possible when the possibility to distinct the objects and their relationships does exist. The objects and their relationships are fixed within some logical language. Suppose that the logical language contains a set of special individualizing functions. Any its subset distinguishes an individual. The further consideration is straightforward: some individualizing functions are truth valued, thus, preserving the predicativity. Before analysing the logic thus generated, a preliminary discussion should give insight to understand the nature

---

*Also: Kashirskoye Ave., 31, Moscow Engineering Physical Institute, Moscow, 115409 Russia

Proceedings of the Workshop on Computer Science and Information Technologies CSIT'99
Moscow, Russia, 1999

of the object or its representation.

The separation of all the objects into assignments, individuals (actual, possible and virtual) and concepts presupposes a special kind of logic with the central notion of an object or element. Any actual object is implemented by the appropriate assignment which is applied to the possible object, and the last one is possible with respect to the prescribed theory. Under this assumption the elements exhibit their partial nature. This means that they are determined and defined with respect to some subdomains but not with respect to others.

Thus, the extended evaluation of the unbound variables is necessary to represent the partial elements so that they range over the sets of actual objects (the actual objects which *do* exist). Therefore, the partial objects are the schematic entities and their schemes are to be considered any time the unbound variables are evaluated. The schemes give advantage in applying theory of computations. A fruitful hint is the following. The object in a theory of computations is represented by some scheme that actually transforms the 'flow' of information.

For the purely mathematical reasons the term 'assignment' is often used instead of 'event' as well as 'evolvent' instead of 'script'.

The material covered in this paper is as follows.

A general conceptual background is given in Section 1. Its feasibility is based on potentiality, relativity, and separability of a source set for individuals. The actual individuals are assigned with the indices while virtual individuals are added as the idealized entities to increase the expressivity of a formal language, giving rise to some natural taxonomy. Some stages of a hypotetical procedure with a problem domain pre-representation are selected out as studying of the problem domains, descriptions with the data, and map to environment.

Section 2 contains a short outline of the metadata base issue. Getting started with a global domain $\mathcal{H}$ the two ways to construe the local domains and concepts are discussed. The individuals are determined via the functional entities $\langle I, h, T \rangle$ for the further capturing both the transactioning and cloning effects. The brief outline, in fact, some extractions, for a type system are included.

Section 3 deals more with the semantic considerations. An approach gives an opportunity to determine the modularity of persistent couples of data. A primitive frame for the evaluation map is given by the rules which are the principles of evaluation.

The basics of variable sets are due to [Law75], [Sco80]. The reasons for types and structures are influenced, but in a rather distinct way, by [AL91]. The semantic considerations for information systems are reviewed in [BSW94]. Overall and general view for the variety of relations is influenced by [Bun79].

Other related topics, with an encirclement of computation environment for different models of information systems, are covered in the rest of references.

1 Conceptual background

The modern trends in developing the majority of information systems are based on some notion of an object. A common place is that the meaning of an object in use depends on the aims and targets of a particular developer. Typically, the creation of an information system involves both the database development and implementation stages. Assume now, that both database and metadata base are under development. As a rule, not only the data sets but also the metadata couples are appeared and used as the various universes of concepts.

The metadata objects reflect the rules and restrictions which are superimposed on the data sets. In addition, the metadata objects originate an existence of the data objects and their families. Intuitively, metadata object is observed as a source to generate these families. Thus, from a mathematical point of view metadata object is *not* represented by a total function.

On the other hand a database gives a valid family of the actual data objects. For the reasons as above this is just a local universe, or partial domain for metadata.

The feasibility of behaviour for both the sets of data objects and metadata objects meets the difficulties for this partial nature of the domains. At least, the grounds and rules how to select out, fix and apply the families of concepts need to be verified with a care. One of the ways to get started is to revise an imaginable and thus idealized development procedure.

1.1 Development grounds

In practice, a database development, even under an adopted data model and tool, meets a great amount of restrictions and exclusions. The process seems to be a multistage and iterative, taking into account the excessive and complicated case study.

An idealization is usually fruitful and needed to concentrate the efforts on the significant features and effects and do not lead to an oversimplification. The formulation of basic development grounds both for databases and metadata bases needs, at least, the three stages. Some stages of a hypothetical procedure are the studying of a problem domain, descriptions with the data, and map to environment. They are straightforward as follows.

1.1.1 Studying of the problem domain

At the first stage, when *some problem domain D is studying* the objects-as-individuals are selected out. They are treated as the generic notions which are intuitively clear and transparent, and this is safety from a mathematical point of view.

Potentiality

By the reasons of a formalism the individuals are assumed to be collected into a single set $\mathcal{H}$, and this set $\mathcal{H}$ is observed as a class of all the *potential* individuals. This assumption seems not to be dramatically restrictive, an important is namely the *possibility* to construe such a single set. Thus, better way is to fix this set *before* the study to avoid the later contradictions in a target formal framework. Otherwise, the starting formalism would be revised under the renewed ideas concerning this initial universe.

Relativity

The members of this class are evaluated relatively some other *assigned* sets Asg which can be indexed, so that they can have their own inner structure. For pure notational reasons to indicate explicitly this inner structure, instead of Asg, the notation I will be used. At this point of discourse this feature of potential indexing is not necessary but leads to some reasonable simplifications.

This feature of potentiality reflects the variations of $\mathcal{H}$ with a time: the individuals *enter* some set starting their existence, and *leave* this set cancelling out their existence relatively previously assigned index. Let observe this index as a cross reference with the set of all potential individuals.

Separabilty

To get started with a case study for the behaviour of individuals, a choice should be done with *actual* U, *possible* H, and *virtual* V individuals which are taken from $\mathcal{H}$. The actual individuals are assigned with the indices while virtual individuals are added as the idealized entities to increase the expressivity of a formal language. The natural taxonomy principle reflects all these three sets:

$$U_i \subseteq H \subseteq V,$$

where i ranges Asg.

The Figure 1 reflects the transformations of an in-language individual $\overline{h}$ under evaluation map into the individual h, and those of the individual h under the events i, j.

1.1.2 Descriptions with the data

At the second stage a problem domain is described with the *data*, and the truth value of a description becomes sensitive to the assignment $i \in Asg$, where Asg is a set of all the assignments. The description is determined by the operator $\mathcal{I}$, the prefix $\mathcal{I}x$ is to correspond rather to the words 'the one and only object (entity) x such

$$\begin{tikzcd}
\bar{h} \arrow[r, two heads, "\|\cdot\|"] & h & & \\
& i \arrow[r, two heads] & h(i) & \\
& & j \arrow[r, two heads] & h(i)(j)
\end{tikzcd}$$

Figure 1: An evaluation map $\|\cdot\|$ maps the in-language individual $\overline{h}$ to h. This possible individual $h \in H$ is driven by *assignment* (or: index, event) $i \in Asg$ resulting in the actual individual $h(i) \in U_i$. The next step could be driven by the event $j \in Asg$ where the map $h(i)$ results in the state $h(i)(j)$.