Problems and features of data structure modeling in distributed Web-oriented production systems

A D Danilov, O Ja Kravets and D V Terekhov
Voronezh State Technical University, 14, Moscow ave, Voronezh, Russian Federation
E-mail: danilov-ad@yandex.ru

Abstract. The goal is to develop a system model necessary for structuring the subject area, decomposing the functions performed by a web application; formalize the design methods, determine the general method of synthesizing Internet-oriented information systems. The main part of the development is the creation of a model, which components will be the objects of the domain, while attention should be paid to balance - the flexibility of the model and the development costs, taking into account the possibility of reuse. Too flexible components will have too many properties and will be difficult to customize adjust. On the contrary, models that implement too particular situation will be useless in terms of reuse, and, perhaps, the cost of its implementation will not make sense. For a more complete understanding of the functions and requirements for data processing, it is necessary to identify the main objects of the domain, information flows, logical connections, as well as rules and restrictions when working with data. The result of building a model should be a DBMS-independent data schema (metadata) corresponding to the internal data structure of the Web application, focused on the implementation of the required functions when interfacing with the user.

1. Introduction
In the process of building new and modernizing existing units in the production systems of nuclear power engineering, evolutionary changes are constantly taking place in terms of using automated process control systems (APCS). At the same time, the structure of the automated process control system is changing, new, modern hardware and software are used, including web applications. A challenge is that in the process of developing the system, the unification of information solutions is violated. Moreover, each time, naturally, problematic issues arise at the stages of equipment introduction, its operation and repair, as well as with further building up and modernization of the process control system and integration of various complexes into a single control system.

Thus, an important task is the development of a system model necessary for the structuring of the subject area, the decomposition of the functions performed by Web applications; formalization of design methods, definition of general methods of synthesis Internet oriented information systems.

2. Problems of data modeling in distributed Web-based information systems
Consider the Web-based production system management system [1] as a set of abstract information objects:

\[ \mathcal{O} = \{O_i\} \tag{1} \]
Information objects have an interface defined by attributes, functions, and also hidden implementation of their structures and processes. Information objects contain extensible attribute lists [2]. Attributes of an information object, will be understood as the properties of an object, which determine its state and relationships with other objects. Attributes can contain both state and process descriptions.

Each object will be assigned a description of its state in the form of the next tuple:

\[ S_i = \langle AO_i, DO_i \rangle, \]  

where \( S_i \) - a description of the state of the information object, \( AO_i = \{ ao_{i1}, ao_{i2}, \ldots, ao_{im} \} \) - a set of names of the properties (attributes) of the information object and \( DO_i = \{ do_{i1}, do_{i2}, \ldots, do_{in} \} \) - and a set of domains of the corresponding attributes.

2.1. Interfaces (relations) modeling

Let a set of \( n \) types (or domains) be given \( T_i (i = 1, 2, \ldots, n) \), all of that do not have to be different. Then \( r \) will be a relation defined on these types if it consists of two parts of the heading and the body, where

a) the heading – is a number of \( n \) attributes of \( A_i : T_i \); where \( A_i \) - unique names of the relations attributes \( r \), and \( T_i \) - relevant domains \((i = 1, 2, \ldots, n)\);

b) the body – is a number of \( m \) tuple \( t \), where \( t \) - many components of the type \( A_i : v_i \), where \( v_i \) - a value of type \( T_i \), i.e. the attribute value for the attribute \( A_i \) in the tuple \( t \) \((i = 1, 2, \ldots, n)\).

Hereunder \( m \) and \( n \) - cardinality and relations rate \( r \). The title of the relationship can be considered a predicate, and the body a true statement derived from the predicate by substituting the value of the corresponding type instead of parameters. The heading for the base relation (i.e. the value of the base variable - relation) is specific and known to the system because it is part of the definition of the corresponding base variable - relation [3].

To model relations between objects (primary and foreign key constraints), the following elements should be introduced.

The list of attributes and domains determine the structure of the data, their scheme. Specifying a specific values for attributes from the set \( AO_i \) we can get a relation consisting of the following tuples:

\[ S_{O_i} = \langle ao_{ij}, d_k, j = 1, \ldots, m, d_k \in do_{ij} \rangle. \]  

Each of the tuples (3) corresponds to a certain object of the domain or its state at a specific point in time.

The values of individual attributes or some of their groups must have a unique property in order to avoid data duplication and to be able to identify a tuple [4]. Such attributes or groups of attributes in the theory of relations [11] are called key attributes. You can set key attributes for each object \( S_i \) by listing them:

\[ K_i = \{ ao_{ij} \}, \]  

where the index \( j \) matches the key attribute numbers. The combination of attribute values included in \( K_i \) must be unique among all tuples \( S_{O_i} \).

On the set of states of information objects \( S_O = \{ S_{O_i} \} \) you may figure out relations \( R = \{ r_k \} \), that define the relations between data elements, where \( R \subseteq S_{O_i} \times S_{O_j} \). Each such relationship links
two relations $S_{Oi}$ and $S_{Oj}$ (figure 1). In addition, each tuple from $S_{Oi}$ matches one tuple or set of tuples in $S_{Oj}$, relation $S_{Oi}$ is a dominant and $S_{Oj}$ is a subordinate.

Relations $R$ can be set by enumerating attribute sets for relations $S_{Oi}$ and $S_{Oj}$ as follows:

$$FK_j \subseteq AO_i,$$
$$REF_j \subseteq AO_j,$$

where $FK_j$ is a set of related $S_{Oi}$ attributes, while $REF_j$ is a set of related $S_{Oj}$ attributes.

![Figure 1. Set of possible object states and state transition.](image)

Data modification in dominant relation $S_{Oi}$ may result in deletion or modification of $S_{Oj}$ subordinate relation attribute values, which are included in $R$. For maintaining data integrity [5] one needs the ability to link every such relation with a procedure, which will perform modifications in $S_{Oj}$ subordinate relation.

Consequently, a set (3) represents number of informational entities, with each of them having a certain corresponding object in a database.

2.2. Application logic modelling

Let us define changes in data states based on system functioning, as a transition graph (figure 2), with each node unequivocally corresponding to internal state of a system.

The graph provided has four levels [6]:

- ER level describes domain objects, their attributes and dependencies. At this level objects, their names, attributes, methods and interconnections are identified.
- Information flows level describes all possible object data states and functions, which model information technology, i.e. transfer objects to different states.
- Function interfaces level models data processing functions: their objectives, as well as input and output parameters.
- Functions realization level elaborates the previous level. It describes function performance in program operations.

It has been established [7, 8] that a large number of recursive structures and other transition graph features determine the choice of source code generation – with XML for transition graph and XSLT
for transformation using specialized templates.

The functioning of information objects can be described as a set of information states

\[ S_o = \{ S_{oi} \}. \]  \hspace{1cm} (6)

Based on a set \( S_i \), we can define a set of functional relations:

\[ F = \{ F_i \}. \]  \hspace{1cm} (7)

where \( F \subseteq S_o \times S_o \) is a set of functional relations, defined by state-to-state functions \( S_{oi} \), i.e. set of data processing functions (figure 1).

The examples of typical data processing functions are: information retrieval, addition, removal and modification of data.

Any typical function can be translated into a sum of four relational data operations executed in a certain order.

We can assign a following description to each data processing function:
\[ S_{F_i} = \langle AF_i, DF_i \rangle , \]  

(8)

where \( AF_i = \{ af_{i1}, af_{i2}, \ldots, af_{im} \} \) is a set of property (attribute) names of \( i \) data processing function; \( DF_i = \{ df_{i1}, df_{i2}, \ldots, df_{im} \} \) is a set of domains of corresponding attributes can serve as conversion function parameters.

Characteristics of active web application can be described as a set of included modules’ conditions:

\[ S_F = \{ S_{F_i} \} . \]  

(9)

Mapping of set into another is crucial in analysis. Suppose \( A \) and \( B \) are two sets. Let every element \( a \) in set \( A \) has a certain element \( b = g(a) \), contained in set \( B \). In this case mapping \( g \) of set \( A \) into set \( B \), can be written as:

\[ g : A \rightarrow B . \]  

(10)

Element \( b \) is called an image of element \( a \) under mapping \( g \), while element \( a \) is a preimage or one of preimages of element \( b \). Often element \( a \in A \) is called a variable or an argument of mapping \( g \), and element \( g(a) \in B \) - a value \( g \) at element \( a \).

Mapping \( g : A \rightarrow B \) is called a function with domain \( A \) and range \( B \). In some branches of mathematics, depending on nature of sets and mapping characteristics, mapping \( g \) is called operator, functional, etc.

While analysing a certain web application, we can note, that there are relations between functional states \( S_O \) and \( S_F \), defined by the following relation:

\[ F_i : S_O \rightarrow S_F . \]  

(11)

Equation (11) should be interpreted as a mapping \( F_i \) of system elements from one functional state into another [9, 10] (or accordance of typical modules with typical data processing functions).

Each function \( F_i \) is described via an interface, i.e. a set of input, output parameters and returned value:

\[ F_i = \{ P_i^i \cup P_i^o \} , \]  

(12)

where \( P = \{ P_i^i, P_j^i, \ldots, P_s^i \} \cup \{ P_i^o, P_j^o, \ldots, P_s^o \} \) is a set of vectors defining input and output functional parameters.

The returned value is a special case of an output parameter.

\[ P^i = \langle AI, DI \rangle . \]  

(13)

3. Conclusion

In summary, we received the following results:

- We defined that it is important to develop a system model for structuring the application domain, web application functions decomposition, design techniques formalization, defining general technique for synthesis of Internet-oriented data management systems.
- We defined information object attributes as object properties, that determine its state and links with other objects.
- We proposed a method for key attribute representation by listing, which takes into account the requirement for uniqueness throughout information space.
- We analysed and formalised typical distributed data processing functions.
References

[1] Wang X V, Wang L and Givehchi M 2015 ICMS: A Cloud-Based System for Production Management, *IFIP WG 5.7 International Conference on Advances in Production Management Systems (APMS 2015)* (Tokyo, Japan) doi: 10.1007/978-3-319-22759-7_52

[2] Infoblox NIOS 8.2 Administrator Guide 2017 Available from https://docs.infoblox.com/display/NAG8/About+Extensible+Attributes

[3] Jamali H R and Nikzad M 2011 Article title type and its relation with the number of downloads and citations, *Scientometrics, 88 (2) 653-61*

[4] Coyier C 2018 The Difference Between ID and Class, *CSS-tricks* Feb 24 Available from https://css-tricks.com/the-difference-between-id-and-class/.

[5] Pérez J R 2014 Maintaining Data Integrity *Quality Progress* Available from http://asq.org/quality-progress/2017/03/data-quality/field-notes-maintaining-data-integrity.html

[6] Nagalla R and Hellestrand G 1994 Signal transition graph constraints for synthesis of hazard-free asynchronous circuits with unbounded-gate delays *Formal Methods in System Design* 5(3) 245-73

[7] Ham F, van de Wetering H and van Wijk J J 2001 Visualization of State Transition Graphs *DBLP IEEE Symposium on Information Visualization (INFOVIS’01)* (San Diego, CA, USA) doi: 10.1109/INFVIS.2001.963281

[8] Henderson K et al. 2011 It’s who you know: graph mining using recursive structural features *KDD’11 Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* pp 663-71

[9] Roberts M 2018 Serverless Architectures Available from https://martinfowler.com/articles/serverless.html

[10] Vu T-D 2009 Vehicle Perception: Localization, Mapping with Detection, Classification and Tracking of Moving Objects. *Computer Science [cs]* (Institut National Polytechnique de Grenoble – INPG) Available from https://tel.archives-ouvertes.fr/tel-00454238/PDF/thesis.pdf

[11] Grunert K G 1989 Attributes, attribute values and their characteristics: A unifying approach and an example involving a complex household investment *Journal of Economic Psychology* 10(2) 229-51