INTEGRATED METHODOLOGY FOR INTERNET-BASED ENTERPRISE INFORMATION SYSTEMS DEVELOPMENT

Sergey V. Zykov, Ph.D.
ITERA Oil and Gas Company L.L.C.

Keywords: ERP, enterprise information system, data modelling, abstract machine, computational environment, content management, software development methodology, application integration

Abstract: The paper considers software development issues for large-scale enterprise information systems (IS) with databases (DB) in global heterogeneous distributed computational environment. Due to high IT development rates, the present-day society has accumulated and rapidly increases an extremely huge data burden. Manipulating with such huge data arrays becomes an essential problem, particularly due to their global distribution, heterogeneous and weak-structured character. The conceptual approach to integrated Internet-based IS design, development and implementation is presented, including formal models, software development methodology and original software development tools for visual problem-oriented development and content management. IS implementation results proved shortening terms and reducing costs of implementation compared to commercial software available.

1 INTRODUCTION

Due to the tendency of almost annual doubling of enterprise information size, state-of-the-art IS and DBMS are currently operating terabyte information arrays, which are going to change for petabyte size shortly.

Tremendous heterogeneous data volumes demand new software development tools that provide continuous iterative design, implementation and support for IS integrating both DB and meta-DB (MDB) containing specific information for DB handling. Since contemporary scholars use MDB for knowledge acquisition and processing, solving problem of integrated DB and MDB management by means of IS provides potentially new level of information handling, thanks to data and knowledge cooperation.

Under the socio-economic globalization conditions combined with growing dependency of nation-wide and worldwide development level upon IT perfection degree, adequate IS functioning demands novel conceptual and methodological approaches to software design and implementation.

Under rapidly growing IT penetration into each and every activity sphere of the modern society, software integration becomes a critical issue, particularly considering DB and MDB sizes and co-existing heterogeneous (and often contradicting) concepts, methodologies, models and approaches to handling them. The fact that major software development companies (such as Microsoft, IBM, Oracle, SAP, BEA etc.) have not produced a uniform approach to complex IS construction yet (including Internet-based ones) speaks for itself; universal terminology has not been worked out either. Quite a number of research groups focus on a uniform theoretical language and SDK solution for enterprise-scale IS design, implementation and integration. However, the problem is still far from its adequate solution.

The above considerations require a conceptually new approach to integrated IS design, implementation and support, which provides dynamic tracking (meta)data integrity and high IS fault tolerance in heterogeneous computational environment.

Therewith, a number of interrelated problems to be solved arises, namely:

- Multiple CASE and RAD methodologies complicate supporting continuous, integrated, problem-oriented IS lifecycle;
- Non-uniformity of mathematical and methodological foundations and programming features of (meta)data object description languages and IS construction software toolkits makes it really complicated to provide continuously integrated software extension and improvement;

V. Zykov S. (2005).
INTEGRATED METHODOLOGY FOR INTERNET-BASED ENTERPRISE INFORMATION SYSTEMS DEVELOPMENT.
In *Proceedings of the First International Conference on Web Information Systems and Technologies*, pages 168-175
DOI: 10.5220/0001227301680175

- Flexible and reliable user interface to IS and DBMS (including personalized data access restriction and interface customization) is a problem to implement;
- Computational environment features can hardly be used completely and uniformly for data objects (DO) and metadata objects (MDO) manipulation, including interface elements.

2 THEORETICAL BACKGROUND

Theoretical and experimental research on IS implementation, improvement and support reveals the following problems to be solved:

- Building a uniform conceptual approach to large-scale IS design, implementation and support for global, distributed, heterogeneous computational environment;
- Creating methodology for continuous integrated support of IS during the entire lifecycle;
- Developing a theoretical background and formal methods for representation, testing and verification of both (meta)data and IS on the whole;
- Implementing (M)DO computational models for highly dynamic IS in extendable heterogeneous environment;
- Developing problem-oriented software design and implementation toolkits;
- Forming choice criteria for integrated solutions of globally distributed enterprise IS.

Research in the above directions resulted in a novel conceptual approach to integrated IS design and implementation, represented by a collection of the following interrelated elements:

- Set of formal models (problem domain conceptual model, abstract machine model for computational environment and software development toolkit);
- Methodology for IS design, implementation and support;
- Choice criteria set for IS prototyping, design and implementation SDK;
- Original software development tools (*ConceptModeller* for visual problem-oriented IS development, content management IS).

3 THE METHODOLOGY

3.1 Structure and organization

The suggested conceptual approach to IS design, implementation and support is based on (meta)data integration and aimed primarily to overcome the problems mentioned.

One of its essential components is conceptual software design, which provides a unified (meta)data computational model in the form of (M)DO, language and software development tools for handling them.

Another essential ingredient of the approach is the methodology that supports continuous multi-level iterative software design and implementation (including reengineering) from problem domain entities to (meta)data and IS schemes. Therewith, (meta)data actuality, completeness, consistency and integrity is controlled throughout the entire IS lifecycle.

One more innovative aspect of the research deals with IS and (M)DB CASE-, RAD-, and integration tools. Application development requires conceptual and methodological generalization of (M)DO management processes to provide integrated DO and MDO management on the basis of uniform, open and extendable interface, language and software tools.

Research objective is analysis, development and approbation of IS construction concept and methodology for acquisition, processing and providing report information on enterprise resources, and practically testing the results obtained for prototyping, design, implementation and support of full-scale software for global information networks. Major tasks solved during the research are the following ones:

- Analyzing design process for global network software;
- Building concept of design, implementation and support of Internet-oriented software;
- Development and approval of the unified approach to solve the formulated task in a form of the methodology for continuous IS integrated design, implementation and support, (M)DO problem domain model, and an abstract machine as computational environment model;
- Software development tools implementation for visual problem-oriented design (*ConceptModeller*) and RAD (content management IS) for the enterprise applications;
- Designing system interfaces and supporting architecture of a generalized global networking IS;

- Implementing fast prototypes and full-scale software for portal information resources management in a large international corporation.

Research methods used to solve the above tasks are based on a creative synthesis of fundamental statements of finite sequence theory (Barendregt, 1984), category theory (Curry, 1958; Cousineau, 1987), computations theory (Scott, 1982) and semantic networks theory (Roussopulos, 1976).

Research contains a novel reasoning and presentation of conceptual and methodological foundations of continuous, integrated enterprise IS design for global networking environment, during their entire lifecycle. All of the approaches known as yet either have methodological “gaps” or do not result in enterprise-level solutions with practically applicable implementation features (scalability, expandability, availability, fault tolerance etc.).

The developed data models (DM) for problem domains (variable domain-based) and software development tools (abstract machine-based) manipulate dynamic and static features of heterogeneous weak-structured environments in a more detailed way. Another aspect of research novelty is the architectural solution for open, extendable, distributed, interoperable, heterogeneous environment that supports personalized front-end and back-end (meta)data processing based on component technology with embedded script procedures.

The variable domain-based DM with states features event-driven DO and MDO control of heterogeneous problem domains. Therewith, the range of possible (meta)data sources is extended up to virtually arbitrary data warehouses (including (M)DB), that support state-of-the-art front-end architectures of globally distributed IS, intermediate and legacy systems. The software development solution features content-oriented (M)DM, which is modeled by an abstract machine. As far as implementation part is concerned, traditional technologies and software development tools are extended by continuous conceptual iterative integrated IS design based on a creative combination of COM model, UML language and BPR technology applied to a decentralized, personalized, globally distributed component environment.

Research attention is focused on object-oriented IS design and implementation that provides front-end interfaces with heterogeneous data warehouses by built-in procedures and embedded dynamic modules.

3.2 The (meta)data object model

Upon analyzing ERP problem domain features, the computational data model (CDM) is suggested in a form of object calculus.

DO of the CDM can be represented as follows:

DO = <concept, individual, state>,

where under a *concept* a collection of functions with one and the same definition range and one and the same value range is implied. An *individual* means an entity that an expert identifies within a problem domain by specifying its unique properties. *State* changes model problem domain individual dynamics.

Compared to results obtained as yet, principal benefits of the CDM suggested are more adequate mapping of heterogeneous weak-structured problem domain dynamics and statics as well as event-driven (meta)data control in a global computational environment. As to architecture and interface, the CDM provides "penetrating" iterative design of open, distributed, interoperable IS based on UML and BPR methodologies and .NET web services. The implementation supports an integrated front-end information management for various data warehouse types of heterogeneous enterprise IS on the technological basis of event-driven procedures and dynamic SQL-based component modules.

The globally distributed information management problem domain requires support for multi-level profile-based access to heterogeneous distributed databases and data warehouses based on unified Internet information interchange protocols and web services. To meet the IS interface requirements, dynamically variable (meta)data representation, flexible personalized access granting and consistent support of data integrity should be provided. IS architecture should be open, extendable, flexibly adaptable to problem domain state; it should also provide online data and metadata personalized correction (i.e. depend on the computational environment state).

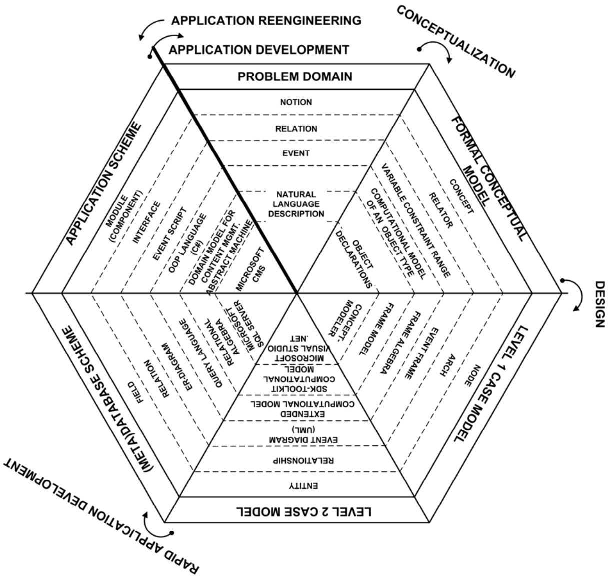

Figure 1: Generalized design and implementation procedure of IS for global network environments.

A multi-level integrated IS design and implementation scheme is suggested (fig. 1) that provides fast component-based development of integrated, open, extendable IS for global networks with continuous (meta)data adequacy and integrity control. During the development procedure IS specification is transformed from problem domain notations to formal CDM entities, further, by CASE-toolkit to OR(M)DB scheme with Oracle Portal as (M)DO manipulation tool, and, finally, to target IS formal architecture and interface component description.

The developed computational model is based on two-level *conceptualization* (Wolfengagen, 1999), i.e., the process of establishing relations between problem domain concepts.

Individuals *h*, according to assigned types *T*, are assembled into assignment-dependent collections, thus forming variable domains

$$H_T(I) = \{h | h : I \rightarrow T\},$$

which model problem domain dynamics and statics.

When fixing CDM individuals, uniqueness of DO *d* individualization within problem domain *D* by formula *Φ* should be maintained:

$$||Ix\ \Phi(x)||i = d \ \Leftrightarrow \ \{d\} = \{d \in D|\ ||\ \Phi(\text{đ})||i=1\}.$$

Computational model semantics for (meta)data and states are adequately and uniformly formalized by multi-sort typed λ-calculus (Barendregt, 1984) and combinatory logic (Curry, 1958), semantic network-based situational description (Roussopulos, 1976) and state-based abstract (virtual) categorical machine (Cousineau, 1987).

The DO model compression principle

$$C = Iy : [D]\ \ \forall x : D(y(x) \spadesuit \Phi) = \{x: D \mid \Phi\}$$

allows model application to concepts, individuals and states separately and to the DO on the whole.

Metadata computational model extends Codd's relational model by the compression principle:

$$x^{j+1} \equiv I z^{j+1} : [\ldots[D]\ldots]\ \ \forall x^j : [\ldots[D]\ldots](z^{j+1}(x^j) \leftrightarrow \Phi)$$

where

z^{j+1}, x^{j+1} are metalevel predicate symbols in relation to level *j*;

x^j is an individual of level *j*;

Φ^j is a DO definition language construct of level *j*.

The above integrated object model for data, metadata and states is characterized by structural hierarchical organization, scalability, metadata encapsulation and readability. Extendibility, adequacy, neutrality and semantic soundness of the formalization provide problem-oriented IS development with (M)DO adequacy maintenance throughout the entire lifecycle.

During the research, automated translation procedure is developed that transforms (M)DO of the above CDM into target (M)DB schemes and (meta)data management abstract machine codes; the formal language specification presented provides data completeness, consistency and integrity.

On the basis of multi-parameter functional

$$F = F\ ((v), (e), \ldots)\ (s)\ (p),$$

where assignment values represent:

s – IS user personal preferences;
p – IS user registration status;
v – IS client interface parameters;
e – IS data access device parameters,

semantics object model and formal generalized procedure of (meta)data instantiation have been built, depending on the above assignments and based on functional *F* evaluation function $||\circ||$ (Zykov, 2002).

3.3 The content management model

Abstract machine for content management (AMCM) (Zykov, 2004a) is suggested as a formal model of content management IS for information resources, which is an improved version of categorical abstract machine (CAM) (Cousineau, 1987). At any given moment AMCM is determined by its *state*. AMCM work *cycle* can be formalized by explicit enumeration of possible state changes, which define the procedure of AMCM state *dynamics* modeling.

From the formal model viewpoint, when portal page templates are mapped into the pages, variable *binding* actually occurs, which is evaluation of variables that characterize template elements with their values, or portal page elements.

AMCM semantics can be described on the basis of D.Scott semantic domain theory (Scott, 1982). Therewith, atomic template types are chosen out of standard domains, while more complex template types are built using domain constructors.

AMCM formal semantics is built in the following order:

1) *Standard* (most commonly used within the model framework) domain enumeration;

2) *Finite* (containing explicitly enumerable elements) domain definition**;**

3) Domain *constructor* (operations of building new domains out of the existing ones) definition, which are domain combination operators;

4) Aggregate domain formalization using standard domains and domain constructors.

Domain constructors include functional space $[D_1 \rightarrow D_2]$, Cartesian product $[D_1 \times D_2 \times \ldots \times D_n]$, sequence D^* and disjunctive sum $[D_1 + D_2 + \ldots + D_n]$.

Let the AMCM language contain expression set *E* (including constant set, identifier set *I*, assignment operation (content “write operation” to template “slot”) etc.), and command set *C* (comparison, command sequence etc.).

AMCM syntax is completely defined by the following syntax domain description:

$$Ide = \{I \mid I - identifier\};$$
$$Com = \{C \mid C - command\};$$
$$Exp = \{E \mid E - expression\}.$$

Let us collect all possible language identifiers into *Ide* domain, commands – into *Com* domain, and expressions – into *Exp* domain.

State-based computational model of AMCM environment is as follows:

$$State = Memory \times Input \times Output; \qquad (s)$$
$$Memory = Ide \rightarrow [Value + \{unbound\}]; \qquad (m)$$
$$Input = Value^*; \qquad (i)$$
$$Output = Value^*; \qquad (o)$$
$$Value = Type1 + Type2 + \ldots \qquad (v)$$

AMCM state is defined by “memory” state considering input values (i.e. content) and output values (i.e. web-pages) of the abstract machine. Therewith, under *memory* a mapping from identifier domain into value domain is implied, which is similar to lambda-calculus variable binding. For correct exception handling, *unbound* element should be added to the domains. *Value* domain is formed by disjunctive sum of domains, which contain content types of AMCM language.

Semantic statements describe *denotates* (i.e. correct construct values) of AMCM (M)DO manipulation language.

Semantic statement for an expression reads:

$$\boldsymbol{E}: Exp \rightarrow [\ State \rightarrow [[Value \times State] + \{error\}]];$$

Expression evaluation in AMCM environment results in such a state change that the variable is bound to its value, or (in case the binding is impossible due to variable and value type incompatibility) an error is generated.

Semantic statement for a command reads:

$$\boldsymbol{C} : Com \rightarrow [State \rightarrow [State + \{error\}]].$$

Generally speaking, expression evaluation in AMCM environment results in AMCM state change and a situation is possible (e.g., assignment type incompatibility), when an error is generated.

Constant denotates are their respective values (in a form of ordered pair of <*variable*, *value*>), while program state remains unchanged.

Identifier denotates are identifiers bound with their values in case binding is possible (in a form of ordered tuples of <*variable_in_memory*, *identifier*, *state*>), while the state remains unchanged, and an error message is generated in case the binding is impossible:

$$\boldsymbol{E}\ [I]\ s = (m, I = unbound)\ error, \rightarrow (m, I, s).$$

Semantic function for assignment command of AMCM language has the following type:

$$\boldsymbol{C}: Com \rightarrow State \rightarrow [State + \{error\}].$$

Thus, content management IS template binding with the content may result in AMCM state change and in a number of limited, predefined cases (particularly, under template and content type incompatibility) – in error generation.

Semantic statement for an AMCM command, which assigns content to template element, results in state change with substitution of content value *v* by identifier ***I*** in the memory:

$$\mathbf{C}\ [I{=}E] = \boldsymbol{E}\ [E] * \lambda v\ (m\ , i, o)\ .\ (m\ [v/I], i, o).$$

4 FROM MODEL TO IMPLEMENTATION

4.1 General implementation scheme

IS development concepts and methodology have been instantiated for enterprise resource management, including information and interface (i.e. data and metadata) of enterprise information resources (Internet and Intranet sites). Detailed design scheme consists of the following stages (Zykov, 2004b):

1) Management board outlines objectives and measures to control information resources that map into formal business rules of IS computational model;

2) Information resource managers build detailed structure and functional enterprise conceptual business model as a DO map;

3) System analysts consider OLAP-instantiations of the enterprise business model versions for various development scenarios;

4) IS and DB developers formalize architecture and interface logics with OO scripting languages (in terms of state-based abstract machine), which is later translated into UML-based DM (abstract machine code) by synthetic CASE methodology;

5) DB, LAN, Internet, Intranet and security administrators, designers and content managers implement and support target configurations for IS, DB and information network resources.

According to detailed development sequence for resource management IS, a generalized heterogeneous data warehouse processing procedure is suggested that allows users to interact with distributed (M)DB in a certain state, depending upon dynamically script-activated assignments. Therewith, scripts are initiated (in the form of (M)DB connection profiles and state-based abstract machine OO script procedures) depending on user-triggered events, which provide transparent intellectual client-server front-end interaction. Dynamically variable profiles for information resource and interface (M)DB access provide strict and flexible personalization, high fault tolerance and data security for ordinary and privileged IS users in heterogeneous environment.

4.2 Component integration algorithm

According to resource management IS design and implementation scheme, a generalized algorithm is outlined for new component integration into existing IS. The algorithm is based on semantically preferred DO analysis, and it provides completeness, consistency and integrity of extendable DO models and an iterative IS design procedure by business model reengineering. The algorithm unifies heterogeneous object-based IS integration process; ERP and document management IS integration is overviewed (Zykov, 1998).

5 IMPLEMENTATION FEATURES

The suggested conceptual approach to IS design and implementation has been practically approved by development of *UniQue* HR management IS, Internet and Intranet portals in ITERA International Group of Companies.

As to information management software for global computational environment, its *Menu* component is aimed at portal navigation data storage and processing. *Pages* subsystem is related to *Menu* and tracks events of assignment, transfer, and deletion of portal pages (meta)data and navigation menu items. *Images* module provides storage, retrieval and portal web publication of digitized photos and graphics. *News Columns* supports periodical portal publications (press releases, media news etc.), including data from related third-party IS modules. *Special sections* organizes visual management of portal content (i.e. data) and design (i.e. metadata) by given criteria set. *Administration* implements profiling, data access policies and (M)DB synchronization.

In terms of system architecture, the IS provides assignments (depending on front-end position in data access hierarchy) with a certain level of (meta)data entry, modification, analysis and report generation (from administrator down to reader). Problem-oriented form designer, report writer, online documentation and administration tools make an interactive interface toolkit. (M)DB supports integrated storage of data (for online access) and metadata (DO dimensions, integrity constraints, data representation formats and other information resource parameters).

During the resource management IS design, problem domain DM specifications (in terms of semantic networks) are transformed by *ConceptModeller* to UML diagrams, then by Oracle Developer/2000 integrated CASE tool – to ER diagrams (or by AMCM and Oracle Portal toolkit – into AM code) and, finally, into target IS and (M)DB schemes.

Using the suggested information model, the architectural and interface solution has been customized for resource management IS with (M)DB interaction assignments for various classes of users and administrators.

IS implementation process included fast prototyping (with SQL and Perl stored OO scripts) and full-scale integrated Oracle-based implementation.

The fast IS prototype has been designed to prove adequacy of the formal (meta)data models, methods and algorithms, it uses a generalized architecture scheme and linking interfaces. As a result of comparative analysis the environment has been chosen that included Sybase S-Designor and PowerBuilder CASE-and-RAD toolkit, Perl language and mySQL DBMS.

Upon prototype testing, a full-scale enterprise OO toolkit has been customized for applied IS development for information resources management including Internet and Intranet portals. Web pages automatically generated by information resources (meta)data management IS are published at ITERA Group Intranet portal and official Internet site (www.itera.ru).

To provide the required industrial scalability and fault tolerance level, the integrated Oracle IS design and implementation toolkit (Portal, Developer/2000) has been chosen to support UML and BPR methodologies.

All of the components are designed, implemented and customized according to technical specifications outlined by the author and tested for several years in a heterogeneous enterprise environment. The resulting implementation terms and costs have been reduced about 40% (on the average) compared to commercial software available, while functional characteristics have been essentially improved. Advanced personalization and access level differentiation allows to substantially reduce risks of (meta)data damage or loss (Zykov, 2003).

6 RESULTS AND CONCLUSION

A comprehensive conceptual approach to IS design, implementation and maintenance has been outlined. Based on the approach, a novel IS development methodology in terms of multi-level iterative procedure has been developed for information resource management, which provides adequate, consistent and integrate (meta)data manipulation during the entire lifecycle (Zykov, 2002).

Upon customizing theoretical methods of finite sequences, categories, semantic networks, computations and abstract machines, a set of models have been constructed including problem domain conceptual model for (M)DO dynamics and statics as well as a model for development tools and computational environment in terms of state-based abstract machines, which provide integrated (M)DO manipulation in (weak-structured) heterogeneous problem domains. For the model collection, a generalized development toolkit choice criteria set has been suggested for IS prototyping, design and implementation.

A SDK has been implemented including *ConceptModeller* visual problem oriented CASE-tool and IS for content (i.e., (M)DO) management.

According to the conceptual approach, a generalized interface solution has been designed for Internet-oriented information management IS, based on content-oriented architecture with explicit division into front-end and back-end sides.

To solve multi-aspect applied task of information resource management, a fast event-driven prototype has been developed using *ConceptModeller* toolkit and PowerScript and Perl programming languages. After the prototype testing, a full-scale enterprise object-oriented IS has been designed and implemented as an information resource management application.

The full-scale enterprise IS has been customized for HR and information resource management (in terms of Internet and Intranet portals) and implemented in a corporation employing nearly 10,000 people. The obtained results proved shortening terms and reducing costs of implementation compared to commercial software available, and revealed high mobility, expandability, scalability and ergonomics of the IS. IS iterative multi-level design is based on a formal model synthesizing OO methods of management of DO (i.e. data) and MDO (i.e. knowledge).

Practical value of the results obtained is defined by the benefits of large-scale IS development by the methods suggested. Particularly, formal DM is used, which synthesizes methods of finite sequences, category theory, computation theory and semantic networks and that provides joint DO and MDO management in heterogeneous interoperable globally distributed environment. Due to formalized IS testing and verification procedures, costs of (meta)data maintenance, fault tolerance and integrity support have been essentially reduced, while IS modernization, customization and performance optimization procedures have been simplified.

The results obtained have been used for development of a number of IS in ITERA Group:

- *UniQue* HR ERP (1998-2000);
- Content management IS for network information resources (2001-2002);
- Official Internet site, www.itera.ru (2003-2004);
- Enterprise Intranet portal (2004).

Models, methods and tools developed by the author (event-driven state-based DM, AM for distributed IS) served as a basis for IS implementation for distributed information resource management in ITERA International Group of Companies. According to ITERA experts, the IS implementation has resulted in annual cost reduction of hundreds of thousands of US dollars, while data management efficiency has increased substantially.

The author is going to continue his studies of the formal models and related SDK that support enterprise content management IS.

REFERENCES

Barendregt H.P. *The lambda calculus (revised edition)*, Studies in Logic, 103, North Holland, Amsterdam, 1984.

Cousineau G., Curien P.-L., Mauny M. The categorical abstract machine. *In Science of Computer Programming* 8(2): 173-202, 1987.

Curry H.B., Feys R. *Combinatory logic, Vol.I,* North Holland, Amsterdam, 1958.

Roussopulos N.D. *A semantic network model of data bases*, Toronto Univ., 1976.

Scott D.S. Domains for denotational semantics. *In ICALP* 1982, 577-613.

Wolfengagen V.E. Event-driven objects. In *Proceedings of the 1st International Workshop on Computer Science and information Technologies CSIT'1999.* MEHhI Publishers, Moscow, Russia, 1999, Vol.1. p.p. 88-96.

S.V.Zykov. Human Resources Information Systems Improvement: Involving Financial Systems and Other Sources Data. In: *W.Litwin, T.Morzy (Eds.) Advances in Databases and Information Systems,* Springer, 1998, p.p. 351-356.

S.V.Zykov. The Integrated Approach to ERP: Embracing the Web. *In Proceedings of 4th International Workshop on Computer Science and Information Technologies, CSIT'2002.* Patras, Greece, 2002.

S.V.Zykov. Integrating Enterprise Software Applications with Web Portal Technology. *In Proceedings of 5th International Workshop on Computer Science and Information Technologies, CSIT'2003.* Ufa, Russia, USATU Publishers, 2003, p.p.60-65.

S.V.Zykov. Abstract Machine as a Model of Content Management. *In Proceedings of 6th Workshop on Computer Science and Information Technologies CSIT'2004,* Budapest, Hungary, 2004, Vol.1, p.p.251-252.

S.V.Zykov. Enterprise Portal: from Model to Implementation. *In Proceedings of 6th Workshop on Computer Science and Information Technologies CSIT'2004*, Budapest, Hungary, 2004, Vol.2, p.p.188-193.