Pattern-Based Development of Enterprise Systems: from Conceptual Framework to Series of Implementations

Dr. Sergey V. Zykov, Ph.D.
National Research University Higher School of Economics, Moscow, Russia
szykov@hse.ru

Abstract

Building enterprise software is a dramatic challenge due to data size, complexity and rapid growth of the both in time. The issue becomes even more dramatic when it gets to integrating heterogeneous applications. Therewith, a uniform approach is required, which combines formal models and CASE tools. The methodology is based on extracting common ERP module level patterns and applying them to series of heterogeneous implementations. The approach includes a lifecycle model, which extends conventional spiral model by formal data representation/management models and DSL-based "low-level" CASE tools supporting the formalisms. The methodology has been successfully implemented as a series of portal-based ERP systems in ITERA oil-and-gas corporation, and in a number of trading/banking enterprise applications for other enterprises. Semantic network-based airline dispatch system, and a 6D-model-driven nuclear power plant construction support system are currently in progress.

1. Introduction

The objective of the paper is a systematic outline of the new technology for developing large-scale integrated heterogeneous applications. Currently, the multinational enterprises possess large, geographically distributed infrastructures, aimed at the same business goals. Each of the enterprises has accumulated a tremendous and rapidly increasing data burden, comparable to an avalanche. In certain cases, the data bulk exceeds petabyte size, and it tends to double every five years. Undoubtedly, management of such data is a serious challenge. The problem becomes even more complicated due to heterogeneous nature of the stored data, which varies from well-structured relational databases to non-normalized trees and lists, and to weak-structured multimedia data. The technology presented in the paper is focused at more efficient heterogeneous enterprise and uniform data management procedures. The technology involves a set of novel mathematical models, methods, and the supporting software engineering tools for object-based representation and manipulation of heterogeneous enterprise data.

2. Heterogeneous Enterprise Systems: Concepts, Problems, Models and Tools

Brute force application of the so-called “industrial” enterprise software development methodologies (such as IBM RUP, Microsoft MSF, Oracle CDM etc.) to heterogeneous enterprise data management, without an object-based model-level theoretical basis, results either in unreasonably narrow “mono-vendor” solutions, or in inadequate time-and-cost expenses. On the other hand, the existing generalized approaches to information systems modeling and integration (e.g., category and ontology-based approaches, Cyc and SYNTHESIS projects – [1-4] – do not result in practically applicable (scalable, robust, ergonomic) implementations since they are separated from state-of-the-art industrial technologies (CASE, RAD etc.).

Thus, the suggested technology of integrated development and maintenance of heterogeneous internet-based enterprise software systems has been created. The approach is based on rigorous mathematical models and it is supported by software engineering tools, which provide integration to standard enterprise-scale CASE tools, commonly used with software development methodologies. The approach eliminates data duplication and contradiction within the integrated modules, thus increasing the robustness of the enterprise software systems (ESS). The technology integrates a set of ESS development levels: data models, software applications, “industrial” methodologies, CASE, architecture, and DB management. The novel technology elements are: (i) conceptual framework of ESS development; (ii) a set of object models for ESS data representation and management; (iii) CASE tools for semantic-oriented ESS development (ConceptModeller) and intelligent content management (ICMS); (iv) ESS implementations [5]. For adequate modeling of heterogeneous ESS, a systematic approach has been developed, which includes object models for both data representation and data management [5,6]. The general technological framework of ESS development provides closed-loop, two-way construction with re-engineering.

The general technological framework of ESS development contains stages, which correspond to data representation forms for heterogeneous software system components, communicating in the global environment.The object nature of the "class-object-value" model framework provides compatibility with traditional object-oriented analysis and design approach (OOAD), as well as with other certain promising approaches (such as D.S.Scott's variable domains [7], V.E.Wolfengagen's conceptual method [8]) and helps to extend the mentioned approaches to model the ESS internet-based environments. The following technological transformation sequence is suggested: (i) a finite sequence object (e.g., a lambda calculus term); (ii) a logical predicate (higher order logic is used); (iii) a frame (as a graphical representation); (iv) a XML object (class definition generated by the ConceptModeller engineering tool); (v) an UML diagram (CASE tool data scheme) in the ESS (meta)data warehouse [5].

Therewith, the warehouse content representation is based on semantic network situation model, which provides intuitive transparency for problem domain analysts when they construct the problem domain description. The model can be ergonomically visualized through a frame-based notation. Warehouse content management is modeled as a state-based abstract machine and role assignments, which naturally generalizes the processes of similar engineering tools, such as (portal page template generation, portal page publication cycle, role/access management etc. Therewith, the major content management operations (declaration, evaluation, personalization etc.) are modeled by the abstract machine language. The language has a formal syntax and denotation semantics in terms of variable domains. The transformation sequence of the model is: (i) a term of variable domain algebra (D.S.Scott's computations theory is used)[7]; (ii) a domain-based function (higher order logic is used) [7]; (iii) a frame (a graphical notation); (iv) a XML object (a template for a ICMS portal page); (v) HTML code (ICMS portal page code) of the ESS portal.

The architecture of the integrated heterogeneous enterprise content warehouse provides unification due to generalized object association-based relationships at the data at metadata levels. Uniform heterogeneous ESS content management is based on a uniform portal foundation, which serves a meta-level enhancement over the enterprise data warehouse. Assignments act as code scripts; they change ICMS machine states, and provide dynamical, scenario-driven content management.

The ConceptModeller tool assists in semantically-oriented visualized development of heterogeneous ESS data warehouse scheme [6]. A semantic network-based model is suggested, which works in nearly natural-language terms, intuitively transparent to problem domain analysts. Model visualization is based on frame representation of the ESS data scheme. Deep integration with mathematical models and ESS CASE tools provides a closed-loop, continuous lifecycle with reengineering. The ICMS tool is based on an abstract machine, and it is used for problem-oriented visualized heterogeneous ESS content management and portal publication. ICMS features a flexible scenario-oriented management cycle and role-based mechanisms. ICMS provides a unified portal representation of heterogeneous (meta)data, flexible content processing by various user groups, high security, ergonomics and intuitively transparent complex data object management.

3. Pattern-Based Development for Enterprise Systems

The general ESS development framework [5] potentially allows the following benefits: (i) applying a "spiral-like" lifecycle to the general ESS development framework; (ii) ESS "tuning" by applying a "spiral-like" lifecycle and subsequent verification; (iii) requirement "tracing"; (iv) building a repository of ESS "meta-snapshots", with which the system and/or warehouse could be "reincarnated" to virtually any previous state using component-wise strategy; (v) building a "pattern catalogue" [9] for heterogeneous ESS, based on the integrated repository of various ESS state "meta-snapshots"; (vi) developing a repository of "branches" for "cloning" slight ESS variations for the "basis; (vii) developing a formal language specification (e.g, a DSL technology-based one) [11]; (viii) adjusting" the existing ESS "meta-snapshot" repository components to match requirements; (ix) reuse of the desired components.

The preferable ESS development framework tends to be iterative; in certain cases waterfall is an option.

An essential feature of the general ESS development framework is its two-way organization. The approach provides reverse engineering possibility both for ESS in general, and their components in particular. The practical value of the approach is provided by the verifiability of heterogeneous ESS components at the uniform level of the problem domain model, which is practically independent upon the hardware and software environment of the particular component. Therewith, a major theoretical generalization is a possibility of mathematically rigorous verification of the heterogeneous ESS components by a function-based model [7,10]. The ESS engineering models are oriented at a promising "pure" objects approach, which is a strategy of .NET and Java technologies, where any program entity is an object.

An essential benefit of the approach suggested is a possibility of adaptive, sequential "fine tuning" of ESS heterogeneous component management schemes in order to match the rapidly changing business requirements.

Such benefit is possible due to the reverse engineering feature of the integrated general iterative framework of ESS development. The reverse engineering is possible down to model level, which allows rigorous component-wise ESS verification. Thus, conventional reengineering and verification can be enhanced by flexible correction and "optimization" of the target ESS in strict accordance with the specified business requirements. This is possible due to the suggested model-level generalization of the iterative, evolutionary ESS development framework. Another benefit of the suggested ESS development framework is a possibility of building a "catalogue of templates for heterogeneous ESS", which is based on an integrated metadata warehouse, i.e., a "meta-snapshot" repository. Thus, the software development companies get a solution for storing relatively stable or frequently used configurations of heterogeneous enterprise software systems. The solution potentially allows avoiding the integration problems of "standard" ESS components and/or combinations, which have been obtained previously. The approach allows serious software engineering project savings for clients, provided the ESS developer's "meta-snapshot" repository already stores a similar or an analogous integrated solution to the system required. The above consideration clears the way for "meta-snapshot" repository development, which stores the chronological sequence of ESS solutions as a tree with the "baseline" and slight variations of ESS "branches".

This is similar to version control CASE tools. The approach allows a reasonable selection of most valuable deliverables of the ESS lifecycle phases, and organization of similar solution "cloning". Therewith, the "clones" may be created both for different client enterprises, and for different companies of a single enterprise.

Further discussion could cover the prospective areas of "meta-snapshot" repository development. First of all, to describe the metadata warehouses and the related enterprise-level business requirements it seems reasonable to develop new DSL-type problem-oriented meta-languages. Let us call them the MetaWarehouse Description Language (MWDL) and the Requirement Specification Language (RSL) respectively. Further, the formal models, outlined in the paper and given a more detailed coverage [5], allow interrelating the RSL and MWDL entities. Semantic-oriented search mechanisms assist in revealing ESS "meta-snapshot" repository components, which provide the closest matching to the new requirements. The approach potentially allows terms-and-cost-effective and adequate transforming of the existing ESS components in order to match the new requirements with minimum corrections effort and, consequently, with minimum labor expenses. Therewith, the global perspective it becomes possible to reuse certain ESS components for current or new clients. Selection criteria for such "basic" components may be percentage of reuse, ease of maintenance, client satisfaction, degree of matching business requirements etc.

4. Implementations: Results Obtained and Projects-In-Progress

ITERA Oil-and-Gas Group: a Portal-Based Solution. The suggested methodology has been practically approved by development of Internet and Intranet portals in ITERA International Group of Companies. During the design stage, problem domain model specifications are transformed by the innovative ConceptModeller SDK to UML diagrams, then by Oracle Developer/2000 integrated CASE tool – to ER diagrams and, finally, into target IS and enterprise content warehouse storage schemes.

Using the suggested data model, the architectural and interface solution has been customized for enterprise resource management IS with content personalization for a wide spectrum of user and administrator types.

To provide the required industrial scalability and fault tolerance level, the integrated Oracle design and implementation toolkit has been chosen to support UML and business process reengineering.

A set of models have been constructed including problem domain conceptual model for enterprise content dynamics and statics as well as a model for development tools and computational environment in terms of state-based abstract machines, which provide integrated object-based content management in heterogeneous enterprise portals. For the model set, a generalized development toolkit choice criteria set has been suggested for information system prototyping, design and implementation. A set of SDKs has been implemented including ConceptModeller visual problem oriented CASE-tool and the CMS. According to the approach, a generalized interface solution has been designed for Internet-portal, which is based on content-oriented architecture with explicit division into front-end and back-end sides. Portal design scheme is based on a set of data models integrating object-oriented methods of management of data and metadata (or knowledge).The major implementations of portals in ITERA Group were: CMS for network information resources, official Internet site, and enterprise Intranet portal.

Distributed Trading Company: a Domain-Driven Messaging System. A trading corporation used to commercially operate a proprietary Microsoft .NET-based message delivery system for information exchange between the headquarters and the local shops. The system was client-server based. The client included a local database and a Windows-based messaging service, while the server side consisted of a Web service and central

database. The operation/maintenance challenges were: complicated client-side code refactoring; difficult error localization/reduction; inadequate documentation; and decentralized configuration monitoring/management for remote shops. To solve the problems mentioned, an approach based on domain-driven development [11] and Domain Specific Languages (DSL) has been suggested. The approach included problem domain modeling and problem domain DSL development. A XML-based DSL was used, which extended the scope of the ESS programming language. The methodology instance included DSL scope detection, problem domain modeling, DSL notation/constraint development, and DSL testing. The lifecycle model was iterative with reengineering support. The Windows service containing a DSL parser is a part of the application. The DSL parser input is a current message transfer map.

The DSL scope included message transfer rules/parameters, and adding new types of messages. The next methodology stage was building semantic model of the objects handled by DSL. Three types of the objects were revealed: messages, message transfer channels and message transfer templates. DSL describes object metadata, i.e., configurations and manipulation rules. Templates were core elements of the model, and channels were links between template instances. Templates and channels together make message maps. DSL described the maps, i.e. the static part of the model, while messages referred to its system dynamics and store the state. Templates define actions with messages, i.e. transform or route them. Based on DSL class model and implementation, messaging maps were built, which were later used by parser to generate system configuration. At this stage, DSL syntax and semantics were built. Messaging map was built as an XML document, defining system configuration and containing templates for routing, message processing, transfer channels and their relationships. The parser creates channel objects based on DSL channel descriptions. Then it configures the messaging system by creating message processing objects similarly. Finally, it instantiates the I/O channels, and creates relationships between channels and message processor.

Thus, the DSL-based refactoring resulted in an enterprise trade management system with transparent configuration and a standard object-based model. The DSL developed solved the problem of messaging management. Since changes are chiefly localized within the transfer configuration /map, the change management has been dramatically simplified. The DSL-based methodology instantiation assisted in conquering complexity, made the proprietary system an open, scalable, and maintainable solution. The approach can easily be customized to fit a broad class of similar proprietary systems.

Air Transportation Planning System. Air traffic planning system is an area of work-in-progress.

The problem is to develop remote access to the planning data. An operating solution currently exists. However, it is based on an outdated TAXXI-Baikonur technology, which is no longer evolving after early 2000s. The technology involves component-based visualized assembling of the server application. The ready-made VCL library components from Borland had been integrated with proprietary TAXXI components. The client side is TAXXI Communicator, i.e. an XML browser, which is a "thin" client. The TAXXI technology is limited Windows framework, which is the only possible basis for both client and server-side applications. According to the State Program of Planning System Updates, the Main Air Traffic Management Centre is going to create the new remote access solution. The internet-based architecture is to be implemented in Java technology and to operate on the Apache web server platform. The solution is to query Oracle-based data centre, process the query output and retrieve the results of the air traffic planned capacities to an intuitive and user-friendly GUI. The practical application of the solution is the global integrated ESS, which is providing a uniform and equal information access to all of the international air traffic participants. The suggested pattern-based and component-wise approach is going to unify the issues of the architecture-level update and application migration in Russia. The methodology will also simplify the integration challenges of the global air traffic management software solution.

Nuclear Power Plant: Approaching a 6D-Model Based Implementation. Another challenging aspect of the methodology implementation is related to high-level template-based software re-engineering for nuclear power plants (NPP). To provide worldwide competitiveness, it is necessary to meet the following requirements: (i) meeting quality standards throughout the lifecycle; (ii) high security under long-term operation; (iii) term-and-cost reduction for new generation facilities development. The approach combines state-of-the-art production potential, advanced control methods, and software engineering tools. Each stage of the NPP lifecycle is mapped into a set of business processes, where not only people, but also enterprise systems are interacting.

Identifying operation sequences, the systems form business process automation standards. For example, workflow mechanisms can assist in building enterprise standards on electronic documents validation and approval. During a certain NPP lifecycle, the enterprise systems acquire information on it. Finally, each of the enterprise systems reveals certain NPP aspects: design, technology, economics etc. Thus, various objects, the systems together describe NPP as a huge object. Heterogeneous nature of the data objects, and a million

number of units make NPP a high complexity information object. A major competitiveness criterion in nuclear power industry is a set of electronic manuals, which helps to assemble, troubleshoot, repair NPP etc. Such manual set provides transparent information models of NPP (units), which allow getting information on the object without directly contacting it. Such a versatile description, combined in a single data model is often referred to as a 6D model, which includes 3D-geometry, time and resources for operating the plant. Since mechanisms for information searching, scaling, filtering and linking, should provide complete and non-contradictory results, the information models should have well-defined semantics. The uniqueness of data entry assumes information model data acquisition by the enterprise systems throughout the lifecycle. While a single information model can be derived out of a single system, the 6D model should combine information models of a number of systems. The methodology for building a 6D model suggests portal-based system integration, which can be based on a "platform" capable of entire lifecycle support. The further information model development assumes monitoring system state changes and their influence to the other parts of the system. This helps to immediately react on critical issues in NPP construction, which can be used for decision making. A major nuclear industry challenge is a typical optimized nuclear reactor. The idea is to select typical invariant units for rapid "template-based" development of a set of slightly varying versions. Applying the methodology to the 6D information model of the nuclear reactor, is promising for pattern-based component-wise NPP series development.

5. Conclusion

Implementation of the suggested approach allowed to developing a unified ESS, which integrates a number of heterogeneous components: state-of-the-art Oracle-based ERP modules for financial planning and management, a legacy HR management system and a weak-structured multimedia archive. The implementation of internet and intranet portals, which manage the heterogeneous ESS warehouse content, provided a number of successful implementations in diversified ITERA International Group of companies (approx. 10,000 employees in over 20 countries). The systematic approach to ESS framework development provides integration with a wide range of state-of-the-art CASE tools and standards of ESS development. Other implementations and work-in-progress areas include: air transportation planning system, messaging system for a trading enterprise, a nuclear power plant and banking solutions. Each of the implementations is a domain-specific one, so the system cloning process is not straightforward, and it requires certain analytical and CASE re-engineering efforts. However, in most cases the approach reveals patterns for building similar implementation in series, which results in substantial term-and-cost reduction of 30% and over. The series can be applied both to subsidiaries and to different enterprises.

References

[1] Guha R., Lenat D.: *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley, 1990

[2] Lenat D., Reed S. Mapping Ontologies into Cyc. *In: AAAI 2002 Conference Workshop on Ontologies for the Semantic Web*, Edmonton, Canada, 2002

[3] Birnbaum L., Forbus K., et al.: Combining analogy, intelligent information retrieval, and knowledge integration for analysis: A preliminary report. *In: ICIA 2005*, McLean, USA, 2005

[4] Kalinichenko L., Stupnikov S.: Heterogeneous information model unification as a pre-requisite to resource schema mapping. In: *ITAIS 2009*, Springer, 2009, pp. 373-380

[5] Zykov S.: Integrated Methodology for Internet-Based Enterprise Information Systems Development. *In: WEBIST2005*, Miami, FL, USA, May 2005, p.p.168-175

[6] Zykov S.: Pattern Development Technology for Heterogeneous Enterprise Software Systems. *Journal of Communication and Computer*, 7 (4), 2010, pp.56-61

[7] Scott D.: *Lectures on a mathematical theory of computations*. Oxford Computing Laboratory Technical Monograph. PRG-19, 1981, 148 pp.

[8] Wolfengagen V.: Event Driven Objects. *In: CSIT'99*, Moscow, Russia, 1999, p.p.88-96

[9] Fowler M.: *Analysis Patterns: Reusable Object Models*, Addison Wesley, 1997, 223 pp.

[10] Curry H., Feys R.: *Combinatory logic*, Vol.1, North Holland, Amsterdam, 1958

[11] Evans E.*: Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison Wesley, 2003, 560 pp.