PATTERN-BASED DEVELOPMENT OF ENTERPRISE SYSTEMS
From Conceptual Framework to Series of Implementations

Sergey V. Zykov
Software Engineering Dept., Higher School of Economics, Moscow, Russia

Keywords: Design pattern, Enterprise software system, Lifecycle management.

Abstract: Building enterprise software systems (ESS) 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 suggested methodology is based on extracting common ERP module level patterns and applying them to series of heterogeneous implementations. The approach includes an innovative lifecycle model, which extends conventional spiral model by: (i) formal data representation/management models and (ii) 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 International Oil and Gas Corporation, and in a number of trading/banking enterprise applications for other enterprises. As for work-in-progress, currently underway are semantic network-based airline dispatch system, and a 6D-model-driven nuclear power plant construction methodology.

1 INTRODUCTION

The paper outlines a new technology for developing large-scale integrated ESS. Currently, the enterprises have accumulated a huge and rapidly increasing data burden, managing which is a challenge. The problem becomes more complex due to heterogeneous nature of the data. The technology suggested is focused at more efficient and uniform data management procedures. It involves a set of novel mathematical models, methods, and the supporting CASE tools for object-based representation and manipulation of heterogeneous ESS data. The architecture of the approach involves internet/intranet portals.

Unfortunately, a brute force application of the so-called “industrial” enterprise software development methodologies (IBM RUP, MSF, etc.) to heterogeneous ESS data management, without an object-based model-level theoretical basis, results either in unreasonably narrow solutions, or in inadequate time-and-cost expenses. On the other hand, the existing generalized approaches to ESS modelling and integration - (Guha, and Lenat 1990), (Lenat and Reed, 2002), (Güngördü and Masters 2003), (Birnbaum, Forbus, and Wagner, 2005), (Kalinichenko and Stupnikov, 2009) – do not result in practically applicable implementations since they are separated from state-of-the-art industrial technologies. A number of research programs supported by UN, UNESCO, USA, EU, Japan, Russia, proves that the heterogeneous ESS problems are essential (Fujiwara, Kanazawa, et al., 2009).

Thus, the suggested technology of integrated development and maintenance of heterogeneous internet-based ESS has been created. The approach is based on rigorous mathematical models and it is supported by CASE 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 ESS. The technology includes conceptual framework, a set of object models and CASE tools for semantic-oriented ESS development and content management (Sushkov and Zykov, 2009), (Zykov, 2010).

2 THE MODELS AND TOOLS

For adequate modelling of heterogeneous ESS, an approach has been developed, which includes object models for both data representation and data management. The general technological framework

V. Zykov S..
PATTERN-BASED DEVELOPMENT OF ENTERPRISE SYSTEMS - From Conceptual Framework to Series of Implementations.
DOI: 10.5220/0003426104750478
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 475-478
ISBN: 978-989-8425-56-0

of ESS development provides closed-loop, two-way construction with re-engineering and verification, which increases ESS robustness and reliability.

The general technological framework of ESS development contains stages, which correspond to data representation forms for components, communicating in the global environment. Such data representation forms include natural language, mathematical models, engineering tools integration, and content management. Data representation forms are further detailed by the representation levels.

Content-oriented approach to ESS data management allows data/metadata generalization on the common model basis, unified management of heterogeneous objects, and adequate modelling of the global internet environment, which is critical for ESS robustness and reliability. The object nature of the "class-object-value" model framework provides compatibility with traditional OOAD; it extends the earlier approaches to model ESS environments.

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 modelled as a state-based abstract machine and role assignments, which naturally generalize the processes of similar CASE tools. The major content management operations are modelled by the abstract machine language. The language has a formal syntax and denotation semantics in terms of variable domains.

The technological transformation sequence of the model is as follows: (i) a term of variable domain algebra (Scott, 1981); (ii) a domain-based function; (iii) a frame (Roussopulos, 1976); (iv) a XML object; (v) HTML code of the ESS portal.

The architecture of the integrated heterogeneous ESS provides unification due to generalized object association-based relationships at the data at metadata levels. The ESS content management is based on a uniform portal foundation, which serves a meta-level enhancement over the ESS data warehouse. Therewith, assignments, implemented as software scripts, which change the states of CMS virtual machine, provide dynamical, scenario-driven content management within the portal architectural framework. Other kinds of script implement personalized content management, which is supported by a multi-parameter functional model and the CMS virtual machine.

The ConceptModeller CASE tool (Zykov, 2010) assists in semantically-oriented visualized development of heterogeneous ESS data warehouse scheme. Therewith, 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 warehouse data scheme. Thus, the ConceptModeller tool provides a closed-loop, continuous ESS lifecycle with re-engineering.

The ICMS engineering tool is based on an abstract machine model, and it is used for problem-oriented visualized heterogeneous ESS content management and portal publication cycle. The ICMS tool features a flexible content management cycle and role-based mechanisms, which allow personalized content management. Due to scenario-oriented content management, the ICMS provides a unified portal representation of heterogeneous data and metadata objects, flexible content processing, high data security, a higher ergonomics level and intuitively transparent complex data object management.

3 PATTERN-BASED DEVELOPMENT

The ESS development framework (Zykov, 2010) has the following benefits:

- ESS "tuning" by applying a "spiral-like" lifecycle and subsequent verification;
- Building a reusable repository of ESS component "meta-snapshots", with which the ESS previous state could be rendered;
- Building a "pattern catalogue" (Fowler, 1997) for heterogeneous ESS, based on the integrated repository of state "meta-snapshots" and a repository of "branches" for "cloning" ESS variations for the "basis";
- Developing a DSL language specification (Cook, Jones, Kent, and Wills, 2008) to specify ESS requirements, and "adjusting" the "meta-snapshot" repository to them.

Thus, the ESS development framework implies software lifecycle variations according to waterfall, spiral, evolution, and incremental approach.

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 or the entire ESS. Therewith, a major theoretical generalization is a possibility of mathematically rigorous verification of the heterogeneous ESS components. The approach benefits adaptive, sequential "fine tuning" of ESS heterogeneous component management schemes, which match rapidly changing business requirements due to reverse engineering down to model level with rigorous component-wise ESS verification. Conventional reengineering and verification can be enhanced by flexible correction and "optimization" of the target ESS according to business requirements. 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 integration problems of "standard" ESS components and/or combinations, which have been obtained previously. The approach allows serious SE project savings for clients, provided the ESS developer's "meta-snapshot" repository already stores a similar integrated solution to the system required. The above consideration gives rise to "meta-snapshot" repository development, which stores chronological ESS solutions sequence as a tree with the "baseline" and slightly different "branches" for ESS variations. This is analogous to software engineering tools for version control. The approach allows a reasonable selection of most valuable deliverables of the ESS lifecycle phases, and organization of similar solution "cloning" for various client enterprises, and for different companies of a single enterprise.

Further discussion could cover the prospective areas of "meta-snapshot" repository development. To describe the metadata warehouses and the related enterprise-level business requirements a new DSL was developed for meta-warehouse description and requirement specification. Semantic network-based search mechanisms with frame visualization will help to obtain the ESS "meta-snapshot" repository components matching the new requirements best. Terms-and-cost-effective transformation of ESS components is provided to match the new requirements with minimum labour expenses.

4 THE IMPLEMENTATIONS

The methodology has been practically approved by development of portals in ITERA Group. The implementation terms and costs have been reduced about 40% (on the average) compared to commercial software available, while features range has been essentially improved. The results proved reducing terms and costs of implementation compared to commercial software available, higher scalability, mobility, expandability and ergonomics.

Another implementation example was a trading corporation with a proprietary .NET-based message delivery system for information exchange between headquarters and local shops. An approach based on DSL and domain-driven development (Evans, 2003) has been suggested. The external XML-based DSL extended the scope of the enterprise application programming language. The methodology instance included DSL scope detection, problem domain modelling, DSL notation development, DSL restrictions development, and DSL testing. The lifecycle model was iterative. The solution was based on a redesigned architecture pattern. The DSL included message transfer rules/parameters and new types of messages. Different shops had different configuration instances, which produced the client-side message processing/transfer structure. Based on DSL class model, messaging maps were built. Parser used the maps to generate system configuration. DSL syntax and semantics were built. DSL-based refactoring resulted in enterprise trade management system with transparent configuration and a standard object-based model. The DSL solved the messaging management problem. Since changes were chiefly localized within the transfer configuration /map, the changes management was dramatically simplified. The DSL-based methodology assisted in conquering complexity, made the proprietary system an open, scalable, and maintainable solution. The approach can be customized to fit similar proprietary systems.

An air traffic planning system is an area of work-in-progress. The problem is to get remote access to the planning data. The current operating solution is based on a legacy TAXXI-Baikonur technology, which 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 an XML browser, i.e. a "thin" client. The suggested approach is going to unify architecture-level update and application migration. The methodology will also simplify integration of the global air traffic management ESS.

Another implementation is related to high-level

template-based software re-engineering for nuclear power plants (NPP). For a competitive NPP, it is necessary to meet the quality, security standards and to provide term-and-cost reduction. The above conditions could be satisfied only under a systematic approach, which combines state-of-the-art production potential, advanced control methods, and SE 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. During NPP lifecycle, ESS acquire information. Heterogeneous data objects and the number of units, make NPP a huge and complex information object.

A major competitiveness criterion in nuclear power industry is a set of electronic manuals, which helps to assemble/troubleshoot NPP. Such a 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 ESS throughout the lifecycle. 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 to build a typical optimized nuclear reactor by selecting invariant units for rapid "template-based" development of a slight variety set. Applying the methodology to the 6D information model of the nuclear reactor, is a promising approach to pattern-based component-wise development of NPP series.

The other implementation examples include pattern-based development of a number interacting sets of ERP applications for several Russian banks.

5 CONCLUSIONS

Implementation of the suggested approach allowed to developing an integrated ESS, which includes 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 provided a number of successful results in ITERA Group. The approach features 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 domain-specific, so the ESS cloning process 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.

REFERENCES

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

Cook S., Jones G., Kent S., Wills A. C.: *Domain-Specific Development with Visual Studio DSL Tools*, Pearson Education, Inc, 2008, 524 pp.

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

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

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

Güngördü Z., Masters J.: Structured Knowledge Source Integration: A Progress Report. In: *IKIMS 2003*, Cambridge, MA, USA, 2003

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

Kanazawa S., Fujiwara M., Nishizawa H., Yokoyama M.: R&D Trends for Future Networks in the USA, the EU, and Japan. *NTT Technical Review*, 7(5), 2009, p.p.1-6

Lenat D., Reed S.: Mapping Ontologies into Cyc. In: *AAAI 2002 CWOSW*, Edmonton, Canada, 2002

Roussopulos N.: *A semantic network model of databases*. Toronto Univ., 1976

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

Sushkov, N.; Zykov, S.: Message system refactoring using DSL. *In:CEE-SECR'09*, Moscow, Russia, pp.153-158

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