Unified Approach in the DSS Development Process

Claudiu BRANDAŞ
West University of Timişoara, Faculty of Economic Sciences

The structure of today's decision support environment become very complex due to new generation of Business Intelligence applications and technologies like Data Warehouse, OLAP (On Line Analytical Processing) and Data Mining. In this respect DSS development process are not simple and needs an adequate methodology or framework able to manage different tools and platforms to achieve manager's requirements. The DSS development process must be view like a unified and iterative set of activities and operations. The new techniques based on Unified Process (UP) methodology and UML (Unified Modeling Language) it seems to be appropriate for DSS development using prototyping and RAD (Rapid Application Development) techniques. In this paper we present a conceptual framework for development and integrate Decision Support Systems using Unified Process Methodology and UML.
Keywords: *Decision Support Systems, Unified Process, UML, Prototyping, DSS Tools.*

Introduction

Decision support systems are one of the most important concepts in the Management Information Systems field. DSS can be viewed as a third generation of computer-based applications. First, mainframe computers were used mostly for transactions processing [13]. Then, there was a growing realization that computers and information technology could be used for purposes other than automating paper work, for example, for management reporting, so the field of management information systems (MIS) was taking hold. Meanwhile, assistance for decision making was the domain of management scientists and operations researchers who created structured models, for which computers served primarily as computation engines.

DSS was both an evolution and a departure from previous types of computer support for decision making. Management Information Systems (MIS) provided 1) scheduled reports for well-defined information needs, 2) demand reports for ad hoc information requests, and 3) the ability to query a database for specific data. Operations Research/Management Science (OR/MS) employed mathematical models to better analyze and understand specific problems. Each was lacking some of the attributes needed to support decision making attributes such as focus, development methodology, handling of managerial data, use of analytic aids, and dialog between user and system.

The structure of today's decision support environment become very complex due to new generation of Business Intelligence applications and technologies like Data Warehouse, OLAP (On Line Analytical Processing) and Data Mining. These applications increase the organization's business intelligence process in order to develop new perspectives of building and implementing Decision Support Systems.

DSS developers must use and integrate various technologies and tools if they want to cover decisional requirements based on business intelligence process. In this respect DSS development process are not simple and needs an adequate methodology or framework able to manage different tools and platforms to achieve manager's requirements.

The classical approaches regarding to DSS development, such as SDLC (System Development Life Cycle) are to rigid [8]. The requirements in decisional process often change quite rapidly and the SDLC structure may force too much structure on end-user requirements too early in the development process. A wide variety of schemes and frameworks attempt to classify DSS design and development tools. A common and easily generalizable classification scheme for DSS tools was proposed by Sprague and Carlson [12]. The development tools are classified

according to three levels of technology: (1) DSS Tools, (2) DSS Generators, and (3) specific DSS applications. Sprague's development framework focused only on the tools that can be used for DSS construction.

In the last years DSS developers use successfully the prototyping technique. This technique evolved in response to the deficiencies of the SDLC and appears to be an expedient and efficient method of "hitting the moving target" of DSS user requirements. Prototyping involves performing the analysis, design and implementation phases concurrently and repeatedly. System prototypes are quickly developed and presented to users. The use of prototyping in DSS development process needs an adequate methodology for systems analysis and modeling based on unified iterative process.

The integration of Decision Support Systems and Business Process will affect competitive advantages of firms. In order to develop Decision Support Systems, modeling of business process is a fundamental work of system analysis and design. System dynamics is useful to solve non-linear, complex, time delay and feedback problems of business processes. Recently, some technologies to model real world and to develop software systems have been presented, Object-Orientation, Function-Orientation, State-Orientation and Formal [9]. Object-Orientation is the most popular method which could make software product to be reusable, portable, and interoperability [11]. For enterprises, if they model business process by object method, they can adjust their system more flexibility.

The DSS development process must be view like a unified and iterative set of activities and operations. The new techniques based on Unified Process (UP) methodology it seems to be appropriate for DSS development using prototyping and RAD (Rapid Application Development) techniques. Unified Process (UP) methodology known as Unified Software Development Process (USDP) [7] is an industry standard in software engineering process. UP aims to build robust system architecture incrementally.

Modeling DSS architecture with UML (Unified Modeling Language) and using the UP as main methodology, DSS developers are able to construct and implement rapidly, accurate and scalable DSS. The UP and UML offers appropriate activities and modeling language for developing modern DSS.

DSS Development based on Unified Process and UML

The DSS development process must be view like a unified and iterative set of activities and operations. Researchers assert the need for new business paradigms to drive a unified approach to development of new active decision support capabilities [10]. Jacobson, Booch and Rumbaugh introduced in 1999 the **Unified Software Development Process (USDP)** as a software engineering process standard [7]. It is commonly referred to as the **Unified Process or UP**. This methodology consists of cycles that may repeat over the life of systems development. UP have three basic axioms [1]:

• *Use case and risk driven*. UP employee Use Cases to capturing users requirements and predicating software construction on the analysis of risk.

• *Architecture centric*. UP approach to developing software systems is to develop and evolve a robust system architecture. Architecture describes the strategic aspects of how the system is broken down into components, and how those components interact and deployed on hardware.

• *Iterative and incremental*. The iterative aspect of UP means that we break the project into small subprojects (the iterations) that deliver system functionality in chunks, or increments, leading to a fully functional system.

There are several commercial variants of UP available. The most widely used commercial variant is **RUP (Rational Unified Process)**. This product supplies all of the standards, tools, etc. that are not included in UP and that one would otherwise have to provide oneself. Both UP and RUP model *the who, when and what* of the software development process, but they do so very slightly differently.

UP have 4 main phases and one iteration workflows for each phase:
- Inception
- Elaboration
- Construction
- Transition
- Iteration: Requirements – Analysis – Design – Implementation - Test

For modeling process and visual representation UP use UML (Unified Modeling Language).
DSS developed based on UP and UML can be easily implemented using Prototyping. The research of use UP in DSS development process is very poor. Most of the research papers refer only on use UML to modeling DSS process.
Using UP as main methodology, UML for system modeling and Prototyping technique for implementation DSS, we have developed a conceptual framework.
The **DSS Unified Development Framework (DSS-UNIDEF)** is an integrated set of methodology, activities, operations, modeling language, tools and technologies for conception, building and implementing decision support systems in a unified manner [3]. This is more a **conceptual framework** than a software framework. This framework is based on Unified Process as methodology, UML as modeling language and Prototyping, DSS Tools / DSS Generators as implementation tools and technique.
We defined this conceptual framework to be an integrated environment for organizational and complex DSS development, not a software framework.
The DSS-UNIDEF can be structured in three layers (figure 1).

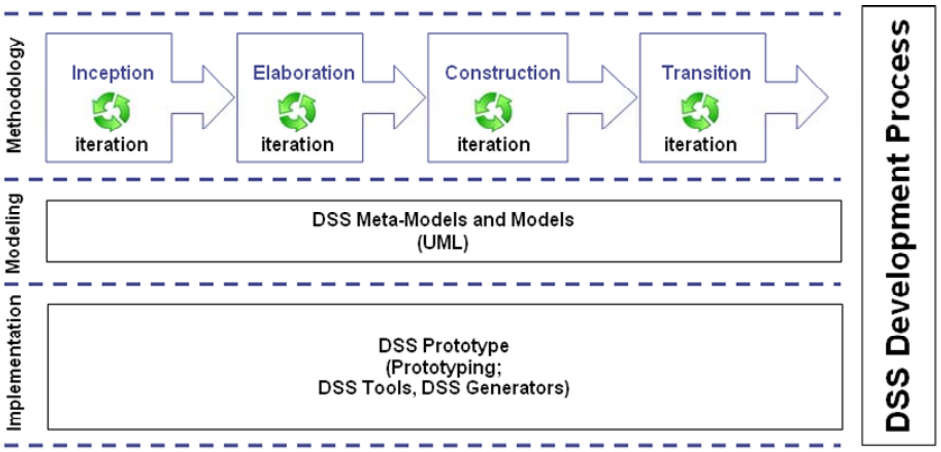

Fig.1. DSS-UNIDEF main layers

I. Methodology. On the first layer is UP methodology. UP have four main phases and one iteration workflows for each phase:
- Inception
- Elaboration
- Construction
- Transition
- Iteration: Requirements – Analysis –Design – Implementation – Test.

II. Modeling. On the Modeling layer is UML. UML is a graphical language that is suit-able to express software or system requirements, architecture, and design. It can be used to communicate with other developers, project clients, and increasingly, with automated tools that generate parts of the system. UML has at least 13 official diagrams and several semiofficial diagrams. Diagrams are windows or views into the model. The diagram is not the model itself. UML diagrams (figure 2) can be divided into *Structural diagram (systems structure)* and *Behavioral diagram (system behavior)*.

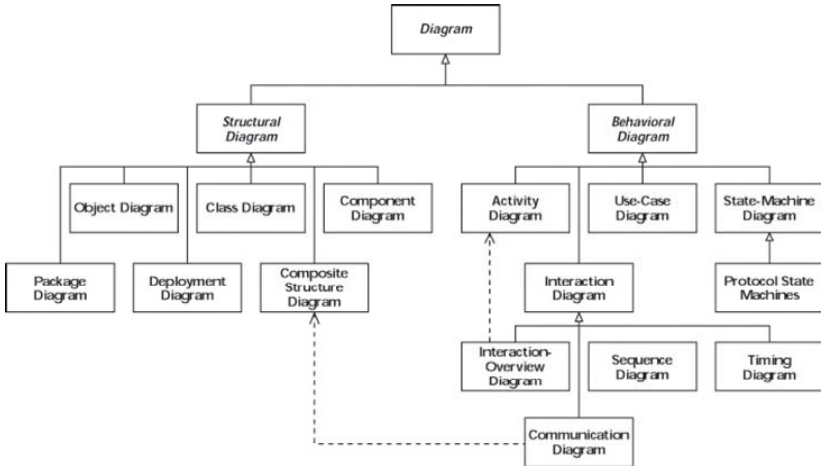

Fig.2. UML diagrams

III. Implementation. On the last layer are Prototyping technique, DSS Tools and DSS Generators.

DSS development process using our framework can be structured in four main phases according with UP methodology. In every phase are used UML models to describe DSS. In table 1 we present the DSS-UNIDEF process.

Table 1 **DSS-UNIDEF Process.**

Unified Process	**DSS Modeling workflows**	**UML Models**	**Deliverables**
Inception	• Identify the needs for DSS development • Identify business risks	Initial Use Case Model of DSS	• DSS Project Plan • Initial DSS Use Case Model • Risks Assessment Document • DSS Throwaway Prototype
Elaboration	• Decision Process Analysis and Modeling (Functional Requirements; Interface requirements; Coordination Requirements) • Decision Makers Profiles Modeling • DSS Design (Reports, Models, Database, Knowledgebase, Interface) • DSS Architecture Modeling	• Static Model • Dynamic Model • Use Case Model	• DSS Meta-model / Model • DSS executable architectural baseline
Construction	• DSS Programming/Design (DSS Tools, DSS Generators) • Develop users manual • DSS Testing	DSS Architecture Model	DSS Software product (beta testing)
Transition	DSS installation/ integration	Deployment Model	Final DSS Software product

Conclusions

Current frameworks and methodologies for DSS Development must be able to integrate Business Process with new DSS applications like Business Intelligence technologies. In this respect DSS developers must have a unified view of DSS building and implementation. From our opinion UP, UML, Prototyping and DSS Tools / DSS Generators are the most recommended for current DSS development and implementation.

DSS models elaborated with UML can be rapid integrated with other business functions in the new MDA (Model-Driven Architecture). With UML, DSS developers can develop complex decision models and databases (Data warehouse, OLAP and Data Mining Applications). In the same time UML offer a high level of component reusability.

Our conceptual framework DSS-UNIDEF can be a basic foundation for development flexible, accurate and reusable DSS.

References

[1] Arlow, J., Neustadt, I., UML and the Unified Process – Practical object-oriented analysis and design, Addison Wesley, Boston, 2002.

[2] Booch, G., Rumbaugh, J., Jacobson, I., The Unified Modeling Language User Guide, 2nd Edition, Addison Wesley, Boston, 2005.

[3] Brandas, C., Conceptual Framework for DSS Development based on UP and UML, Annals of the Tiberiu Popoviciu Seminar of Functional Equations, Approximation and Convexity, Mediamira Science Publisher, Cluj-Napoca, 2006.

[4] Chang, LC, Tu, Y.M., Attempt to Integrate System Dynamics and UML in Business Process Modeling, Proceedings of 21th International Conference of System Dynamics Society, 2004.

[5] Eom, B.S., Decision support systems research: current state and trends, Industrial Management & Data Systems, 99/5, MCB University Press, pp. 213-220, 1999.

[6] Filip. F.G.. Sisteme suport pentru decizii, Editura Tehnica, Bucuresti, 2004.

[7] Jacobson, I., Booch,, G., Rumbaugh, J., Unified Software Development Process, Addison-Wesley, Boston, 1999.

[8] Marakas, G.M., Decision Support Systems in the 21st century, 2nd edition, Prentice Hall, New Jersey, 2003.

[9] Peters, J.F., Pedrycz, W., Software Engineering: An Engineering Approach, JohnWiley & Sons Inc, 2000.

[10] Ring, J., Intelligent enterprises, INCOSE INSIGHT, Vol. 6 No. 2, 2004.

[11] Schach, S.R., Object-Oriented and Classical Software Engineering, 5th ed., McGraw-Hill Companies Inc, 2002.

[12] Sprague, R. H., and Carlson, E. D., Building Effective Decision Support Systems, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[13] Sprague, R.H., Jr., Watson, H.J., Decision Support for Management, Prentice Hall, New Jersey, 1996.

[14] Turban, E., Aronson, J.E., Decision Support Systems and Intelligent Systems, 6th edition, Prentice Hall, New Jersey, 2001.

[15] Zaharie, D., Albescu, F., Bojan, I., Ivancenco, V., Vasilescu, C., Sisteme Informatice pentru asistarea deciziei, Dual Tech, Bucureşti, 2001.

[16] ***, Decision Support Systems, Journal, Elsevier B.V., Vol. 34, 35, 36, 37, 38.