Integration of HCI Needs with SE Methods Using OODPM Methodology

Offer Drori

The Hebrew University of Jerusalem, SHAAM – Information Systems,
P.O. Box 10414, Jerusalem, ISRAEL
offerd@cc.huji.ac.il

Abstract: Many information systems softwares currently being developed relate to the system's user interface. Notwithstanding, the separation between the fields of HCI and SE is customary only in a part of the academic arena. In most cases, human computer interaction (HCI) affects the overall process of software engineering (SE). The challenge, as we see it, relies on systems analysts' and software developers' understanding of organization and user needs, and the ability to translate these needs into functional information systems. In this paper, we will describe OODPM methodology that offers explicit support for HCI using object-oriented methods from the SE field.

Keywords: OODPM, SE and HCI, Prototype methodology

1 Introduction

In object-oriented design, the object is the building block upon which the system is constructed. It comprises a method (the algorithm according to which the object operates), data that the object activates, and an area in which data is transferred between the object and other programs or objects (Alter, 1996) (Booch, 1994) (Rumbaugh et al., 1991). Although in OO design an object is a building block, objects do not exist (or behave) in isolation. Rather, they participate in actions together with other objects in order to satisfy some objective. Also, relationships between objects defined by relationship invariants define the stable properties of the system that should not be violated by those actions (Kilov, 2002) (ISO/IEC).

Prototype methodology bases system planning and design upon the construction of a prototype, which is used to test, demonstrate, and evaluate the proposed system. Even though there are several kinds of prototypes, all allow the system to be perceived from the user's point of view. Instead of relying on the user's imagination, the prototype provides a tool that fully simulates the future system, thereby minimizing the risk of errors and misunderstandings (Martin, 1991).

OODPM embodies an integration of these approaches. The system is planned by defining objects (as well as relationships and behavior), while the user interface (the graphical design) is presented by means of a prototype. This approach has been presented in several publications (Drori, 2001) (Drori, 2000) (Drori, 1998) (Drori, 1996). A recent survey of large software houses in Israel (Drori, 2003), which examined the use of methodologies and tools in information systems development phases, found that 42% of the firms use UML methodology, 15% use OODPM (Drori, 2002), 12% use OMT, 10% use Mafteach (Mafteach), and 7% use Code & Yourdon.

2 OODPM Components

In the OODPM model, the activity (object) comprises six components (see Fig. 1):

(a) The user interface - window/screen of the computer system, depicting the activity of the future system.

(b) Verbal description of the process used by the activity (free text or pseudo-code). If the process will be automated, this description must be explicit, integral, and unambiguous – in other words, pseudo-code.

(c) Input data that will be entered into the window/screen of the information system.

(d) Output of the computer system window/screen (for example, a computer record consisting of the input fields and additional data, such as a user ID, date, etc.).

(e) Data format used in the activity.

(f) System state (Common communication area for objects, for example, Response code table).

Components (a), (b), (c), (e), and (f) are preconditions of the activity (i.e., they should be true before the activity can occur), and components (d) and (f) are post conditions of the activity (i.e., they should be true as an immediate result of the activity).

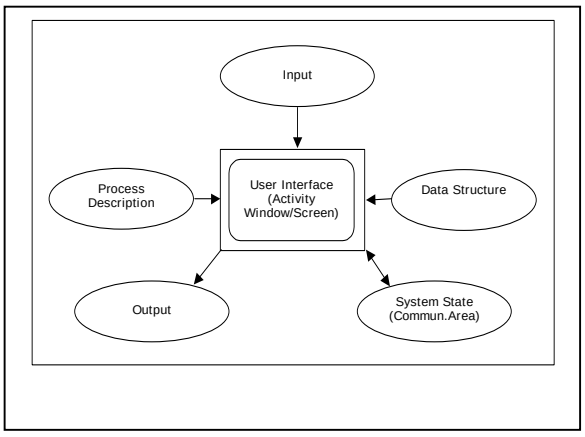

Figure 1: Object Components in OODPM

3 Template for OODPM (Ver. 5)

A template for the methodology has been developed to achieve the goal of planning an information system that gives the user the ability to influence his needs. The template includes 5 chapters that assist the system analysis to define the requirements of the future system with the user's collaboration. The five chapters are: 1. Initiation And Introduction;
2. Description of Existing System; 3. Feasibility Research; 4. Defining the New System;
5. Designing the System. The most important chapter in the user point of view is chapter 5 since he gets from it the user interface of the future system as defined by the system analyst.

The template paragraphs are:

1 Chapter On Initiation And Introduction

The process of initiating the project; the central problem that led to the project; the initiator, referral to the analyst, etc.

1.1 Initiation

1.2 Description of Background

1.2.1 General description of the organization

{Include here general organizational structure diagram}

1.2.2 Characteristics of activity

1.2.3 Quantitative operational data

1.3 Definition of the Problem

1.3.1 General diagnosis of the problems from the point of view of the consumer.

1.4 Aims

1.4.1 Aims of the organization

1.4.2 The aims of the organizational unit examined

1.4.3 Aims and gauges of an information system

1.5 Fundamental Assumptions, Framework, and Limitations

1.5.1 Scope of operation and its domain

1.5.2 Budgetary limitations

1.5.3 Timetable

1.6 Program For Carrying Out the Survey; Previous System Analysis Work

1.6.1 Method

1.6.2 Budget estimate

1.6.3 Timetable

1.6.4 Approval from the authorized decision-making level

2 Chapter on Description of Existing System

OOPDM focuses primarily on system planning, but also addresses the business specification stage. According to this approach, user needs to be implemented in the future system must be studied, but time must also be dedicated to studying the current situation in order to complete the requirements definition. Experience has shown that users tends to focus on those needs that have not been met by the current system, and tend to ignore the parts of the system that have met their needs. Without a stage to examine the current situation, only a partial definition of the requirements is likely to achieved.

2.1 System and Structure

2.1.1 Organizational structure

{Include here organizational unit structure diagram}

{Include here organizational work team structure diagram}

2.1.2 Technological character of the organization

2.1.3 Management characteristics

2.1.4 Development plans

2.2 Flow of Information

2.2.1 The flow of information among principals in the organization

{Include here functional flow chart}

{If needed use here Data Flow Diagram (DFD) content only}

2.2.2 Analysis of information

2.3 Procedures

2.3.1 Description of procedures

2.3.3 The timing and coordination of times between the procedures

2.4 Processes

2.4.1 Description of processes

2.4.2 The timing and coordination of times between the processes and activities

2.4.3 Handling inadequacies

2.5 Forms

2.5.1 Description and characteristics of forms

2.5.2 Examples of forms

2.5.3 Division of fields in forms

2.6 Files

2.6.1 Description, characteristics, and structure of files

2.6.2 Records

2.6.3 Special fields and specifications

2.7 Reports

2.7.1 Description, structure, and characteristics of reports

2.7.2 Contents of reports

2.7.3 Examples of reports

2.8 Linkages with Other Systems

2.8.1 Direction and depth of linkages

2.9 Resources in the System

2.9.1 Manpower

2.9.2 Equipment

2.9.3 Software

2.9.4 Hardware

2.9.5 Communication

2.9.6 Miscellaneous

2.10 Definition of Problems of Existing System

2.10.1 Process problems

2.10.2 Information problems

2.10.3 Coordination problems

2.10.4 Technological problems

3 Chapter on Feasibility Research

This chapter deals with the description of all alternatives that can be used to solve the problems as responses to requirements and problems from the previous chapter.

3.1 Requirements of the New System

3.1.1 Required tasks

3.1.2 Inadequacies to be corrected

3.1.3 Additional improvements

3.1.4 Aims and gauges

3.1.5 Constraints

3.2 Alternative Solutions

3.2.1 Description of solutions

3.2.2 Describe the solutions as responses to requirements and problems

3.2.3 Advantages and disadvantages of each solution

3.3 Economic Feasibility

3.3.1 Lifetime of the system

3.3.2 Development cost

3.3.3 Operating cost

3.3.4 Comprehensive analysis for each solution.

3.4 Technological Feasibility

3.4.1 Possibility of development from a technological standpoint

3.4.2 Possibility of integration into an existing technology

3.4.3 Technological stability and its reliability

3.4.4 Comprehensive analysis for each solution

3.5 Operational Feasibility

3.5.1 Extent of required change to the existing situation

3.5.2 Anticipated resistance and/or assistance

3.5.3 Comprehensive analysis for each solution

3.6 The Chosen Solution

3.6.1 Comprehensive analysis

{Include here Delphi model's weighted grading table summary}

3.6.2 Recommendations for deciding on the system

3.6.3 Decision

3.7 Action Program for Project Development

3.7.1 Manpower required for development

3.7.2 Hardware and software required for development

3.7.3 Timetable

3.7.4 Approval of the authorized decision-making level

4 Chapter on Defining the New System

This chapter deals with the definition of the new system and, in practice, with its design.

4.1 Perceptions and Principles

4.1.1 Perception of the computerized system

4.1.2 Organizational aspects

4.2 Constraints and Limitations

4.2.1 Constraints and limitations of the computer system

4.2.2 Constraints and limitations of the organization

4.3 General Description

4.3.1 System components

{Include here a structural chart of the system}

4.3.2 System flow

4.3.3 Scope and frequencies of operations

4.4 Data Structure for System Components

4.4.1 Input contents

4.4.2 Output contents

4.4.3 Structure of files and their contents

4.5 Sub-systems

4.5.1 Components of each sub-system.

4.5.2 Inputs of each sub-system.

4.5.3 Outputs of each sub-system.

4.5.4 Files of each sub-system

4.5.5 Operations of each sub-system.

4.5.6 Operational screens of the sub-system.

4.5.7 Developing phases recommended

4.6 Linkages with Other Systems

4.6.1 Technical linkages

4.6.2 Operational and organizational links

4.7 Hardware and Software Requirements

4.7.1 Equipment requirement

4.7.2 Software requirements

4.7.3 Information security and backup requirements.

4.8 Operation, Conversion, and Assimilation Requirements

4.8.1 Testing and trial requirements

4.8.2 Conversion requirements

4.8.3 Essential requirements for preliminary operation

4.9 Control Requirements and Feedback

4. 9.1 Quality

4.9.2 Timetable

4.9.3 Resources

4.9.4 Reporting

4.10 Possibilities For Expansion

4.10.1 Possible requirements for increasing the volume of data

4.10.2 Possible requirements for the increase of functions

4.10.3 Possible requirements for technological changes

4.10.4 Possible requirements for linkages with other systems

5 Chapter on Designing the System

This chapter relates to designing the system, which is the last stage before programming.

5.1 Detailed Description of the System

5.1.1 Description of the system

5.1.2 Function of various components

5.1.3 Processing procedure

5.2 Detailed Description of Each System Object (Component)

5.2.1 Drawing of the system screen (or of another main component, such as a printout)

{Include here drawing of the system screen/window comprise the user interface of the object (GUI)}

5.2.2 Input from each object

5.2.3 Output to each object

5.2.4 Processing of each object (the method)

5.2.5 Messages from each object

5.2.6 Data structure of each object

5.3 System Navigation Diagram

{Include here system navigation diagram}

5.4 Files / Tables

5.4.1 Description and characteristics

5.4.2 Examples of records

5.5 Databases

5.5.1 Logical organization

5.5.2 Physical organization

5.6 Software and Development Environment

5.6.1 Infrastructure software

5.6.2 Programming languages and application generators

5.7 Equipment / Hardware

5.7.1 Power of processors

5.7.2 Peripheral equipment

5.7.3 Communication equipment

As we can see, chapters 4 and 5 are based on the user's needs. In chapter 5, paragraph 5.2 (detailed description of each system object) is the most important part of the planning work. In this paragraph, the designer expresses his or her understanding of the user's needs by demonstrating the user interface of the future system. Using this technique, the user can test the system's attributes and processes, and, most importantly, can modify options or attributes and assert his or her needs.

4 Advantages of Designing Information Systems with OODPM

OODPM combines the SE method for developing information system with HCI needs. Prototyping is a low-cost means of demonstrating the idea behind a design. A prototype can be easily modified, and the design evolves gradually to an optimal solution before the system is implemented. Prototyping offers

savings in time and cost to develop something that can be tested with real users (Luqi, 2003). Using the prototyping environment represents significant support for the usability of the prototyping effort. It can be expected to improve the quality of the systems developer, decrease the time and reduce the cost of software development, enhance the developer's satisfaction and productivity, and in most ways makes software development a more delightful and exciting task (Johnson, 1999). Using Prototype as a main part of the object definition is one of the major integrations of SE and HCI. The advantages of designing an information system with OODPM are:

1. It creates greater understanding between the developer and the user.
2. It improves the understanding between the designer and the developer.
3. Part of the developing work is dictated by the design process
4. The system is finely tuned to the user's needs.
5. It enables a reuse of the software, thereby economizing on organizational resources.
6. Reusing proven software components improves the reliability of the system.
7. It enables the creation of object-oriented systems in conventional programming languages.

5 Summary

In most cases, human-computer interaction affects the overall process of software engineering. Prototyping is the main bridge between these two fields. SE is a platform that empowers software designers to develop software based on the requirements specification. Requirements specification is based, first of all, on the human factor that is involved in the HCI process. In this paper, we describe methodology (OODPM) in the domain of the object-oriented field that offers explicit support for HCI using the SE method. A template for word 2000 is available for convenient use of OODPM methodology. The first version of OODPM was published in 1994, and the most recent version (5.0) was published in February 2003. The latest version is the first one to be in English (all other versions were published in Hebrew). More information about the methodology can be found at the methodology site (OODPM). The template may also be downloaded at the site.

References

Alter, S. (1996), *Information Systems - A Management Perspective, 2nd ed.*, Menlo Park, CA: The Benjamin/Cummings Publishing Company.

Booch, G. (1994), *Object Oriented Analysis and Design with Applications. 2 ed.*, Redwood City, Calif.: Benjamin/Cummings.

Drori, O. (2003), Use of CASE Tools and Object Oriented Methodologies, Maydaon - The Israeli Journal for Information Technology, Tel-Aviv: The Israel Chamber of System Analysts, No. 133, June 2003, 21-26 (in Hebrew).

Drori, O. (2002), *Planning information systems using OODPM methodology - user guide, version 5*, Jerusalem, Academon, 2002 (in Hebrew).

Drori, O. (2001), HyperCASE - Case Tool which Supports the Entire Life Cycle of OODPM, *Proceedings of the ECOOP2001 conference workshop on Automating Object-Oriented Software Development Methods* (June 2001, Budapest, Hungary).

http://shum.huji.ac.il/~offerd/papers/drori062001.pdf

Drori, O. (2000), Analysis and Design of Information Systems Using OODPM - Practical Implementation, *Proceedings of the OOPSLA 2000 Workshop on Behavioral Semantics* (Minneapolis, Minnesota, USA, October 15, 2000).

http://shum.huji.ac.il/~offerd/papers/drori102000.pdf

Drori, O. (1998), Definition of requirements for an OODPM-Based information system using hypertext, in: *Proceedings of the ECOOP'98 Workshop on Precise Behavioral Semantics with an Emphasis on OO Business Specifications* (Brussels, Belgium, July 24, 1998), ed. by H. Kilov, B. Rumpe, Munich University of Technology, TUM-I9813, pp. 75-84.

http://shum.huji.ac.il/~offerd/papers/drori071998a.pdf

Drori, O. (1996), Planning and Design of Information Systems Using OODPM, in: *SIGSOFT Newsletter*, New York: ACM SEN (Software Engineering Notes), Volume 21 Number 5, p. 95.

http://shum.huji.ac.il/~offerd/papers/drori091996.pdf

ISO/IEC JTC1/SC21. Open Distributed Processing – Reference Model: Part 2: Foundations (ITU-T Recommendation X.902, ISO/IEC 10746-2).

Johnson, C. (1999), Why Human Error Analysis Fails to Support System Development, *Interacting with Computers*, Vol. 11, No. 5, 517-524.

Kilov, H. (2002), Business Models: A Guide for Business and IT, N.J.: Prentice-Hall.

Luqi, Z. Guan. (2003), A Software Prototyping Framework and Methods for Supporting Human's Software Development Activities, *Proceedings of Bridging the gaps between SE and HCI workshop*, held at ICSE'03, 114-121.

Martin, M. (1991), *Analysis and Design of Business Information Systems*, New York: Macmillan Publishing Company.

Mafteach http://www.methoda.com/

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991), *Object-Oriented Modeling and Design*, Englewood Cliffs, N.J.: Prentice-Hall.

OODPM http://oodpm.huji.ac.il