Table of contents

TABLE OF CONTENTS
ABOUT THE AUTHOR

PREFACE
P1. CONTENT AT A GLANCE
P2. SYSTEM DESIGN “PHILOSOPHY”

CHAPTER 1
GENERAL SOFTWARE DESIGN AND IMPLEMENTATION CONCEPTS AND SOLUTION PARADIGMS
1.1. SOME SPECIFIC FEATURES OF SOFTWARE DEVELOPMENT PROCESS
1.1.1. Art and science of software development
1.1.2. Software development trends
1.2. OPERATING SYSTEM. FOUNDATION OR A MODULE HOME IN ITSELF?
1.2.1. The role of operating system in the design process. Real life example and some inferences
1.2.2. Dependence of software applications on the operating system
1.3. UNDERSTANDING SOFTWARE SYSTEMS’ COMPLEXITY AND FUNCTIONALITY
1.3.1. Reducing system design’s complexity and compacting implementation solutions
1.3.2. System design as an iteration process in a multidimensional environment
1.3.3. Relationship of application’s functionality, complexity and size
1.4. STANDARDIZATION, UNIVERSALIZATION AND GENERALIZATION AS SYSTEM DESIGN CONCEPTS, AND THEIR RELATIONSHIPS
1.4.1. Standardization paradigm. Generic and universal solutions
1.4.2. Consequences of excessive dependence on the third party’s design tools
1.4.3. Optimizing design solutions by combining custom and standardized approaches

CHAPTER2
SYSTEM DESIGN APPROACHES AND TECHNOLOGIES
2.1. SYSTEM DESIGN AS A DELICATE BALANCE OF MANY FACTORS
2.1.1. Art and science of system design
2.1.2. How close should we customize solution to the problem?
2.1.3. Choosing the system design toolset
2.1.4. Iterative and incremental nature of real things. Defining the project goals
2.2. TYPES OF SYSTEMS AND APPLICATIONS
2.2.1. Transaction processing systems
2.2.2. Office automation systems
2.2.3. Management information systems
2.2.4. Artificial intelligence systems
2.2.5. Using functional classification of software systems
2.2.6. Software systems integration
2.3. SOFTWARE SYSTEM’S DESIGN CYCLE
2.3.1. Straightforward approach

2.3.2. Every next project is new
2.3.3. Inherent features of software projects
2.3.3.1. Complexity
2.3.3.2. Conformity
2.3.3.3. Changeability
2.3.3.4. Invisibility
2.4. PEOPLE IN SOFTWARE DEVELOPMENT. CRAFTING THE RIGHT ENVIRONMENT
2.4.1. What factors should be considered with regard to stakeholders interests?
2.4.2. Fairness as a condition of productive environment
2.4.3. Compromising and conceding
2.4.4. Multifactor nature of software development projects
2.4.5. Commitment
2.4.6. Feeling of accomplishment. Freedom and control of creativity and self-realization

CHAPTER 3
DISSECTING DESIGN APPROACHES AND METHODS
3.1. THE SPIRAL APPROACH
3.2. IBM RATIONAL UNIFIED PROCESS ® (RUP ®)
3.3. MODEL DRIVEN ARCHITECTURE (MDA)
3.3.1. MDA overview
3.3.2. MDA and platform independence
3.3.3. MDA' s platform specific models
3.4. AGILE SOFTWARE DEVELOPMENT
3.4.1. Origin of this approach
3.4.2. Prototyping as a foundation of agile software development
3.4.3. Agile development methods
3.4.4. Summing up
3.5. SPECIFICS OF OBJECT ORIENTED DESIGN
3.5.1. Fundamentals of object oriented design (OOD)
3.5.2. Pros and cons of OOD
3.5.3. OOD advantages
3.5.4. OOD cons
3.5.5. Wrapping things up
3.6. UNIFIED MODELING LANGUAGE UML
3.6.1. UML and development environment
3.6.2. UML basics
3.6.3. UML discussion
3.7. TERMINOLOGY, NOTATIONS AND NAMING CONVENTIONS
3.8. DOCUMENTING DESIGN PROCESS
3.8.1. The role of documentation development in the project's process
3.8.2. Documentation development process
3.9. HARNESSING THE POWER OF OBJECT ORIENTED DESIGN IN C++
3.9.1. Generic algorithms and function objects
3.10. COMPLETING THE FIRST PART OF A SYSTEM DESIGN STORY

CHAPTER 4
HIGH PERFORMANCE INTERPROCESS COMMUNICATION DESIGNS
4.1. ORIGIN OF INTERPROCESS COMMUNICATION BASED ON MEMORY MAPPED FILES
4.2. DISTRIBUTED COMPUTER SYSTEMS AND PARALLEL COMPUTING CONCEPTS
4.3. MEMORY MAPPED FILES AND INTERPROCESS COMMUNICATION
4.4. COMPARING PERFORMANCE OF INTERPROCESS COMMUNICATION MECHANISMS
4.5. ACCESSING DATA AND SENDING MESSAGES THROUGH MEMORY MAPPED FILES
4.6. ONE-TO-ONE INTERPROCESS COMMUNICATION
4.6.1. Mapping files into process's space
4.6.2. Communicating via the memory mapped files. Inherent two-way nature of this communication
4.6.3. Closing Joiner connection
4.7. ONE-TO-MANY JOINER INTERPROCESS COMMUNICATION
4.7.1. One-to-many communication in Joiner's "any" mode
4.7.2. Dynamic configuration of the overall system in "any" mode
4.7.3. Joiner's "every" mode for one-to-many type of communication
4.7.4. Dynamic reconfiguration of Joiner connection in "every" mode
4.8. MANY-TO-ONE JOINER'S INTERPROCESS COMMUNICATION
4.8.1. Many-to-one Joiner's connection in "every" mode
4.9. JOINER COMMUNICATION OF MANY-TO-MANY TYPE
4.9.1. Improving software systems' concurrency
4.9.2. Implementation of many-to-many type of communication
4.9.3. Implementation of many-to-many type of communication without semaphores
4.10. SOME SPECIFIC ISSUES OF JOINER CONNECTION
4.10.1. Joiner connection via the paging file
4.10.2. Joiner's connection security issues
4.11. JOINER INTERPROCESS COMMUNICATION OVER NETWORKS
4.11.1. Emulating Joiner connection over networks
4.11.2. Using operating system's signal mechanism for Joiner communication over networks
4.12. GENERAL NOTES ABOUT THE JOINER INTERPROCESS COMMUNICATION MECHANISMS

EPILOGUE
REFERENCES
Index table