1. INTRODUCTION

Digital simulation of power system gathers importance in various engineering studies for power system design, planning and training, ranging from a simple load flow to complex contingency, security and transient stability analysis (Pandit et al., 2000). Power system computation programs were traditionally developed using function-oriented methodology and implemented in Fortran with its associated advantages and limitations. Due to escalating power demand and expanding power system network, the operation and control of contemporary power system become more complex and cumbersome. The requirements on the software are increasing day by day and the application software being used in energy management system (EMS) and distribution management system (DMS) has to be upgraded to meet the requirements. Simulation and operation of the deregulated power system requires sophisticated software tools. Hence, the power industry computer applications are undergoing a revolution to significant phenomena such as the replacement of mainframe computer solutions with networked microcomputers and workstations, as well as the movement toward object-oriented (OO) software development methodologies. Modern control centers make use of the open distributed systems with client server architecture to distribute their functions among different computers (Dy-Liacco, 1994).

![Simulation software development procedure.](image)

Fig. 1 shows the steps involved in the development of steady-state analysis and time domain simulation software. The development process begins with the real world model, which is the physical system. The behavior of the physical system is mathematically modeled using linear and/or nonlinear, algebraic and/or differential equations. The mathematical equations and the data corresponding to the physical components are translated into the software model, which describes the data specifications on which all the operations are performed. Software modeling plays a significant role in representing the mathematical model of the system into the computer memory. The developed software model is then implemented in any programming language to obtain a full-fledged simulation module.
In practice, developing a better software model, which supports various analyses of the problem domain, is a challenging task. In traditional function-oriented programming, where the problem is decomposed into various functions performing a particular task, the behavior of the system is completely decoupled from the characteristics or attributes of the system. The data of the problem domain is kept global and is passed as arguments into the functions for performing desired computations. The relationships between different physical components are not replicated in the software. This reduces the ‘one to one’ matching between the physical system and the software model. Consequently, the software model turns out to be very difficult to realize when the physical system becomes more complex. This is where the object-oriented modeling can provide a better solution. Object-oriented modeling decomposes the problem domain into various objects and performs a particular task by the interaction between objects (Booch, 1994; Ambler 2001). Researchers exploited the advantages of object-oriented methodology (OOM) to address the issues involved in the development of power system analysis applications since early ’90s (Neyer et al., 1990; Zhou, 1996).

2. MOTIVATION

Over the last decade, a large number of power system problems, which were investigated by many researchers in the past, have been re-engineered from the object-oriented point of view. Recently, object-oriented models for radial and weakly meshed distribution systems have been provided in (Losi and Russo, 2003). The developed object model is limited to only three-phase balanced distribution system and the unbalanced distribution system was not addressed.

Object-oriented approach was applied for the development of efficient sparse matrix computation tools (Hakavik and Holen, 1994; Pandit et al., 2001 b), network topology processor (Foley and Bose, 1995; Pandit et al., 2001 a), load flow analysis (Neyer et al., 1990; Fuerte-Esquivel et al., 1998) and dynamic simulation (Manzoni et al., 1999). Object-oriented design patterns were introduced to power system domain in (Zhu and Jossman, 1999) and also employed in (Li and Broadwater, 2004) for designing a software framework. Recently, OOM has been applied for optimal power flow analysis in (Bouktir and Belkacemi, 2003). The method proposed is to handle only cost minimization objective function and no provision is given to interchange the solution technique.
Most of the works reported earlier are focused on modeling of the physical components of power system and little amount of attention is provided on modeling of the conceptual objects such as algorithms for power system computation. Accordingly, the present work has made an attempt for modeling the unbalanced distribution system and employing the design patterns for modeling the conceptual objects of power system computation.

3. OBJECTIVES OF THE PRESENT WORK

1. To eliminate few drawbacks of the existing object-oriented model for balanced operation of the distribution system and to model the unbalanced three-phase distribution system components by reusing the objects of single-phase components of balanced distribution system.

2. To present an object-oriented model for radial distribution system that can handle the topology changes due to switching operations.

3. To model the conceptual objects of power system computation by employing the design pattern technique, thus enhancing the code reusability and design extensibility.

4. SCOPE OF THE PRESENT WORK

- Only the steady state analyses and no time domain simulations are considered.
- Transmission and distributions systems are studied.
- Only radial configuration is considered in the unbalanced distribution system.
- Load flow and optimal power flow analyses of transmission system are considered for modeling the conceptual objects using design patterns.
- FACTS devices such as SVC, TCSC, STATCOM and UPFC are included.

In the following sections of this synopsis, unified modeling language (UML) class diagrams (Rumbaugh et al., 1999; Ambler, 2001) have been employed to express the proposed object-oriented design of power system. In this synopsis, class names within the text are given in italic letters and starts with CAPITAL letters in order to differentiate it from the normal text. In class diagrams, they are given in all CAPITAL letters to improve the clarity.
5. DESCRIPTION OF THE RESEARCH WORK

5.1 OBJECT-ORIENTED MODELING OF DISTRIBUTION SYSTEM

5.1.1 Balanced Distribution System

Although the distribution system is unbalanced, resulting into unbalanced three-phase line currents and line voltages, approximate analyses are performed assuming a balanced operation to obtain ballpark answers (Kersting, 2002). Accordingly, the three-phase distribution system is assumed to be balanced and it can be represented by a single line diagram.

In the present work, initially, radial distribution system has been modeled using OOM and then the extensibility property of OOM has been utilized to extend the objects to model weakly meshed system and to incorporate the dispersed generators. Analyzing the radial distribution system in object-oriented point of view, the physical components such as load, shunt capacitor, shut reactor, bus, feeder, series branch, substation etc. have been modeled as objects. The proposed class diagram of the radial distribution system is shown in Fig. 2.

Unlike the object decomposition proposed in (Losi and Russso, 2003), the present work has modeled the feeder, the load, and the shunt devices as objects. Feeder is modeled as an object that aggregates several branch objects. This enhances the application of proposed object decomposition not only in the load flow analysis but also in the short circuit analysis and feeder reconfiguration etc. Since the load is modeled as separate object, various load representations such as constant power, constant current and constant impedance loads can be easily included in the analysis without any modification in the class Bus.
When the objects are extended to model the weakly meshed system, few new specialized classes have been derived. They are Tie and Dummybus. Tie is modeled as a specialized class of Feeder consisting of only one SeriesBranch. The Dummybus is a specialization of class Bus. In order to incorporate the dispersed generators that can be modeled as PV buses for analysis, a new class PVBus is derived from the class Bus. Moreover, in practical distribution system these generators may be located at the fork node, terminal node or at any one bus along the feeder. Hence, the classes FPVNode (PV Fork Node), TPVNode (PV Terminal Node) have also been derived using multiple inheritance technique of object-oriented methodology. Using the proposed object model, a load flow analysis module employing backward forward sweep method has been developed and tested with 33-bus, 69-bus and IEEE 37-bus system. The results obtained have been verified with the results provided in the literature.

5.1.2 Unbalanced Distribution System

An important contribution of the present work is effective reuse of the balanced system objects, with little modifications if required, to model the three-phase unbalanced system using “aggregation” and “composition” principles of object-oriented design. Unbalanced distribution system has three-phase, two-phase and single-phase components, where the connected phases (a-b, b-c, c-a for two-phase and a, b, c for single-phase) have to be taken into consideration. Three-phase and two-phase objects are modeled as a composition of three and two single-phase objects respectively. In order to represent the connected phases, an additional attribute is included in the single-phase objects.

The unbalanced distribution system has been modeled as an object of class UBDistSys (UnBalanced Distribution System), which is an aggregation of all three-phase feeders, two-phase feeders, single-phase feeders, three-phase, two-phase and single-phase buses etc. Fig. 3 shows the proposed class diagram of an unbalanced distribution system. Only important base and derived classes are shown to improve the clarity. All the three-phase, two-phase and single-phase buses have been modeled as objects. Similarly, three-phase, two-phase and single-phase transmission lines and cables have been considered. Three-phase transformers of various winding configurations have also been included in the proposed object modeling. Single-phase loads connected between phase and neutral as well as between two phases have been considered. Star and delta connected three-phase loads of different types such as constant power, constant current and constant impedance have also been modeled in this work.
Three-phase shunt capacitors and voltage regulators have also been included. The complete details of object modeling of all the components in the unbalanced distribution system are described in the thesis. The proposed object model has been used for the implementation of unbalanced load flow and short circuit analysis. The developed programs have been tested with IEEE 13-bus, 34-bus and 123-bus systems. The advantages of the proposed method are discussed in the thesis.

5.2 TOPOLOGY PROCESSING IN RADIAL DISTRIBUTION SYSTEM

Distribution system is subjected to frequent switching operations due to fault isolation, service restoration and feeder reconfiguration, which alter the topology of the network. Hence, an efficient network topology processor (NTP) program is essential in the DMS to track the existing network connectivity. A dynamic topology processor algorithm for radial distribution system has been proposed in this work and implemented using object-oriented methodology.

The classes RootNode, ForkNode and TerminalNode are inherited from the class Bus in the object-oriented design provided in (Losi and Russo, 2003) and the new design proposed in this work in Sec. 5.1. But the creation of fork node and terminal node are dynamic in nature, when the switching operations are taken into consideration. The number of feeder sections
existing in a distribution system and the series branches that make the feeder sections may also differ from one topology to other. Hence, during the object decomposition of distribution system, the topology related attributes should be untied from the attributes representing the physical nature of the distribution system components. Fig. 3 shows the modified object model of radial distribution system, which contains both physical and topology based objects. Decorator design pattern (Gamma et al., 1995) has been used to separate the topology-based attributes from the attributes representing the physical characteristics of the objects. State design pattern has been used to model the switching states of the tie and sectionalizing switches.

The proposed method has been tested on the standard 69-bus (Venkatesh and Ranjan, 2003) distribution system. The actual 69-bus distribution system is modified to include 5 switches. The developed topology processor algorithm is implemented in C++ and executed in a Pentium III, 1.13GHz processor with 512 MB RAM and the CPU time for reading data, creating initial feeder sections and for the switching operations are shown in Table 1. For comparison, the dynamic data structure provided in (Venkatesh and Ranjan, 2003) has also been implemented and tested on the modified 69-bus system. The average CPU time for topology processing following a switching operation is found to be 0.25ms. The CPU time for the present method (Table 1) is more than this due to the time involved in message passing between the objects. The topology update method proposed in this work can handle different switching options described in the thesis, while the method provided in (Venkatesh and Ranjan, 2003) has been found to be not suitable for all switching options.

Fig. 3 Modified class diagram of radial distribution system.
The major difference between the present method and other available methods lies in the model of Feeder. In earlier methods, the Feeder is modeled as a collection of buses existing along the feeder. On the contrary, Feeder is modeled in the present work as an aggregation of SeriesBranch objects, which resembles the real world feeder. Moreover, most of the radial distribution system analyses are based on branch power flows rather than nodal power injections. Hence, the proposed object model is more suitable for topology processing together with different load flow algorithms such as Newton-Raphson based (Losi and Russo, 2003), forward-backward sweep based (Shirmohammadi et al., 1988), ‘DistFlow’ equations based (Baran and Wu, 1989 a) and power loss based (Venkatesh and Ranjan, 2003) as well as the reconfiguration algorithm based on branch exchanges used in (Baran and Wu, 1989 b).

5.3 OBJECT-ORIENTED MODELING OF TRANSMISSION SYSTEM

The following objectives are emphasized in the proposed object-oriented decomposition of power transmission system:

1. Chosen objects should be analysis independent and should realize the physical nature of the system.
2. Relationship between the physical objects should be replicated in the software objects in order to make the software model more understandable.
3. Conceptual objects, which are mostly designed for computational purposes, should be different from the physical objects and may vary according to the analysis.
4. Code reusability and design extensibility.

5.3.1 Physical Object Modeling

The physical components in the transmission system are abstracted as software objects. The main features of the physical components, which expose their physical characteristics, are encapsulated into the individual objects. The proposed class diagram for the physical components of the power transmission system is shown in Fig. 4. In the present work, more concentration is given to the modeling of conceptual objects involved in load flow and optimal power flow analysis. It is briefly described in the subsequent sections.

<table>
<thead>
<tr>
<th>Operation</th>
<th>CPU Time (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading data and creating feeder objects.</td>
<td>0.01</td>
</tr>
<tr>
<td>100 switching operations</td>
<td>0.04</td>
</tr>
<tr>
<td>1 switching operation</td>
<td>0.0004</td>
</tr>
</tbody>
</table>

Table 1 Execution time for topology processing.
5.3.2 Conceptual Object Modeling

In the earlier works, design patterns (Gamma et al., 1995) were exploited to obtain reusable software objects for the physical components of power system (Zhu and Jossman, 1999; Li and Broadwater, 2004). The present work has investigated the applications of design pattern in modeling the conceptual objects of power system computation such as load flow and optimal power flow analysis.

5.3.3 Load Flow Analysis

The physical objects such as bus, branch, generator and load etc. are stored in different types of data structures by different clients. Hence, it is mandatory for an analysis tool to have a unique method for traversing different data structures. Iterator pattern has been used to solve this problem. Class diagram illustrating the application of iterator pattern is shown in Fig. 5.

Generally, different types of buses involved in load flow analysis are defined as sub classes of class Bus. While performing computation, a PVG bus (bus with a generator) may be converted into PQ bus due to violations in the reactive power generation limits. Similarly, other types of PV buses (buses with SVC or STATCOM) may also be converted into PQ bus due to operating limit violations. Usually, this situation is handled by including a flag in the class PVBus, which toggles from ‘0’ to ‘1’, when the bus type changes. The use of inheritance and the use of flag will be cumbersome, when two or more voltage controlling devices are connected at the same bus. Decorator design pattern, which can be dynamically attached to and removed from the object, has been proposed and used in this work, as shown in Fig. 6, to model the different roles played by Bus.

Several numerical methods such as Gauss-Seidal, Newton-Raphson and Fast Decoupled are available for solving the load flow problem. All those methods share several common steps.
Template method design pattern, which defines a skeleton of an algorithm in an operation, redefining some steps in the subclasses, has been used to develop load flow analysis programs employing different solution techniques as shown in Fig.7. This enhances the code reusability.

5.3.4 Optimal Power Flow Analysis (OPF)

OPF involves different kinds of objective functions, a large number of equality and inequality constraints. Researchers proposed number of classical and heuristic techniques to solve optimal power flow problem. The programs written for solving OPF, with a particular objective, using a particular solution technique are not suitable to handle other objective and other solution technique. Succinctly, the developed OPF program is neither extendable nor flexible to make amendments. Few of these issues are addressed in this work employing design patterns.

In the present work, different objective functions are implemented as objects. The objects of various objective functions should have common interfaces so that they can be easily accessed by a particular solution methodology, which means that the abstraction of objective function should be independent of its implementation. Bridge design pattern, which decouples the abstraction of an object from its implementation, has been employed as shown in Fig. 8 to accomplish this.

Inequality constraints are modeled as conceptual objects. Due to the large number of constraints, a large number of objects are created in the program, which occupy huge amount of memory space and may reduce the execution speed. Since the intentions of all the constraints are same, flyweight design pattern, which uses sharing to support large number of fine-grained objects efficiently, has been selected in this work to model inequality constraints. Fig. 9 shows the application of flyweight design pattern.

Interchanging of solution techniques in OPF program has been made feasible through strategy design pattern as shown in Fig. 10. Different solution techniques are implemented as subclasses of class Strategy and a new solution technique can also be included by deriving a subclass without affecting other classes. The computational classes developed for load flow and optimal power flow analyses are shown in Fig. 11 and Fig. 12 respectively. The developed programs have been tested on IEEE benchmark systems. The implementation details and the advantages obtained using design patterns are described in the thesis in detail.
Fig. 5  Class diagram illustrating the application of iterator pattern.

Fig. 6  Class diagram illustrating the application of decorator pattern.

Fig. 7  Class diagram illustrating the application of template method pattern.

Fig. 8  Class diagram illustrating the application of bridge pattern.

Fig. 9  Class diagram illustrating the application of flyweight pattern.

Fig. 10  Class diagram illustrating the application of strategy pattern.

Fig. 11  Computational classes in load flow analysis.

Fig. 12  Computational classes in OPF.
6. CONCLUSIONS

• Unbalanced distribution system is a complex system involving three-phase, two-phase and single-phase components. Steady state analysis of unbalanced distribution system requires flexible and analysis independent software models for its components. Composition technique of object-oriented methodology has been identified more suitable for modeling unbalanced distribution system components.

• In order to obtain extendable objects for distribution system, the topology-based attributes of distribution system components should be untied from the attributes representing the physical nature of the components. The number of feeders existing in a distribution system and the series branches that make a feeder will differ from one topology to other. Hence, the feeder objects should be created dynamically during the execution of the program.

• Association between the objects makes the process of tracing the path between each node and the substation easy as compared to procedure-oriented implementation.

• In the present work, Decorator design pattern is identified and explored for modeling conceptual objects representing different types of buses involved in load flow analysis due to different kinds of voltage controlling devices. Decorator design pattern is also suitable for modeling conceptual objects resulting from the devices controlling the power flow through a transmission line.

• Bridge, strategy and flyweight design patterns are suitable to obtain a general framework for optimal power flow analysis. Selection of appropriate design patterns to model the conceptual objects plays a vital role in obtaining extendable object-oriented design for power system computation software.

7. REFERENCES


9. **Gamma, E., R. Helm, R. Johnson** and **J. Vlissides** *Design Patterns- Elements of Reusable Object-Oriented Software*. Pearson Education Asia, 1995.


8. PROPOSED CONTENTS OF THE THESIS

Acknowledgements
Abstract
List of tables
List of figures
Abbreviations
Notation

Chapter 1 Introduction

1.1 Background
1.2 Issues to be considered in developing power system analysis/simulation programs
1.3 Literature review
1.4 Scope of the present work
1.5 Organization of the thesis

Chapter 2 Object-Oriented Paradigm

2.1 Introduction
2.2 Object-oriented methodology
2.2.1 General
2.2.2 Main features of object-oriented methodology
2.3 Phases of object-oriented methodology
2.4 Unified modeling language
2.5 Design patterns
2.5.1 General
2.5.2 Classification of design patterns
2.6 Selected patterns for the present work
2.6.1 Iterator pattern
2.6.2 Decorator pattern
2.6.3 Template Method pattern
2.6.4 Bridge pattern
2.6.5 Flyweight pattern
2.6.6 Strategy pattern
2.6.7 State pattern
2.7 Summary

Chapter 3 Object-Oriented Modeling of Distribution System

3.1 Introduction
3.2 Modeling of balanced distribution system
3.2.1 Modeling of radial distribution system
3.2.2 Modeling of weakly meshed distribution system
3.2.3 Modeling of distribution system with dispersed generation
3.3 Modeling of unbalanced distribution system
3.3.1 Single-phase load modeling
3.3.2 Three-phase load modeling
3.3.3 Single-phase and three-phase shunt modeling
3.3.4 Three-phase bus modeling
3.3.5 Three-phase feeder modeling
3.3.6 Three-phase voltage regulator modeling
3.3.7 Two-phase components modeling
3.3.8 Unbalanced distribution system modeling
3.4 Object-oriented implementation of distribution system analysis
3.4.1 Balanced load flow analysis
3.4.2 Unbalanced three-phase load flow analysis
3.4.3 Short circuit analysis of unbalanced distribution system
3.5 Summary

Chapter 4 Dynamic Topology Processing in Radial Distribution System

4.1 Introduction
4.2 Modified object model of radial distribution system
4.3 Static and dynamic information of objects
4.4 Creation of feeder objects
4.5 Network topology processing
4.5.1 Switch opening operation
4.5.2 Switch closing operation
4.6 Case study
4.7 Summary

Chapter 5 Application of Design Patterns in Transmission System Power Flow Studies

5.1 Introduction
5.2 Existing object models for load flow and optimal power flow
5.2.1 Objectives of the proposed object model
5.3 Object model of power transmission system
5.4 Selection of design patterns for load flow analysis
5.5 Application of design patterns in load flow analysis
5.5.1 Application of Iterator pattern
5.5.2 Application of Decorator pattern
5.5.3 Application of Template Method pattern
5.6 Computational classes for load flow analysis
5.7 Selection of design patterns for optimal power flow analysis
5.8 Application of design patterns in optimal power flow analysis
5.8.1 Application of Bridge pattern
5.8.2 Application of Flyweight pattern
5.8.3 Application of Strategy pattern
5.9 Computational classes for optimal power flow analysis
5.10 Summary

Chapter 6 Conclusions

6.1 Conclusions
6.2 Contributions
6.3 Scope for further work

Appendices
References
List of papers published based on this thesis
9. LIST OF PUBLICATIONS BASED ON THE RESEARCH WORK

Journals


Conference Proceedings


Papers under review
