Web Based Energy Audit and Accounting Software for Power Distribution Utilities

www.serd.ait.ac.th/reric

R. P. Gupta * and A. K. Khastagir **

* Department of Electrical Engineering
Indian Institute of Technology Kanpur, Kanpur – 208 016
** Interra Software (India) Pvt. Ltd.
E-14, NEPZ, Noida 201305
INDIA

ABSTRACT

This paper presents simple and novel method to estimate the total energy loss (technical and commercial) in different parts of the power distribution system. Based on the proposed method, a comprehensive software solution has been designed and developed using the Internet technology for Web based energy audit and accounting for power distribution utility. Further, a set of test cases has been designed and performed to check the proper functioning of the software and proposed methodology.

1. INTRODUCTION

The economic growth and development of a country depends heavily on the reliability and quality of the power. In most of the countries the gap between demand and supply of electricity is increasing due to rapid growth in industrialization. Generally, rigorous planning is done for the addition of the generation and expansion of the transmission networks using modern technology. However, the distribution systems have generally grown in an unplanned manner resulting in high technical and commercial losses in addition to poor quality of power. The technical loss is due to Ohmic loss or I^2R loss in the conductor besides Iron loss or No-load loss in the transformer and usually determined through load flow study. The commercial loss, on the other hand, is due to theft and pilferage, meter tampering, unauthorized connections and un-metered supply. In general, most of the commercial losses occur in the power distribution system. In most of the developing countries, the commercial loss is as high as 2/3 of the total Transmission and Distribution (T & D) losses. Though most of the existing literatures focus primarily on the technical loss calculation [1, 2], there is a need to address the commercial loss in distribution system.

In view of the above, a concept has been proposed to estimate the energy losses (technical and commercial) in the power distribution utilities in this paper. Based on the proposed method, a web based Energy Audit and Accounting Software (EAAS) system is designed, developed and tested. The developed software has capability to provide both monthly and yearly record of energy losses (technical and commercial) at different hierarchical levels of the power distribution utilities.

2. ENERGY LOSS CALCULATION

A typical distribution system consists of primary, secondary and low voltage (LV) distribution feeders. These feeders operate at different voltage levels. The primary distribution feeder emanates from sub-transmission network and terminates at a distribution substation. At substation, voltage is converted from primary distribution voltage level to secondary distribution voltage level. The secondary distribution feeders emanate from distribution substation and terminate at distribution transformer. The distribution transformer converts secondary distribution voltage into LV distribution voltage level. Consequently, a set of LV distribution feeders comes out from a distribution transformer and runs in different load areas. As an example primary, secondary and LV distribution voltage levels are 33 kV, 11 kV and 415 V in most of the Indian power distribution system. The energy loss calculation method has been conceptualized and described at different voltage levels in the next section.

2.1 Energy loss in a primary distribution feeder

This involves energy metering at all the primary distribution feeders emanating from sub-transmission substation and another metering at low voltage side of the distribution substation. Consider a primary feeder, which is feeding three distribution substations S1, S2 and S3. Then monthly energy loss in this primary feeder is given by:

$$E_{loss\ in\ pri\ feeder} = E - (E_1 + E_2 + E_3 + E_4) \tag{1}$$

where, E = monthly energy unit sent out on this feeder at the sub-transmission substation,

E_1, E_2 & E_3 = sum of energy units sent out on all the secondary feeders emanating from the distribution substations S1, S2 and S3 respectively month wise, and

$E4$ = sum of the consumptions of all the HT consumers on this primary distribution feeder month wise.

2.2 Energy loss in secondary distribution feeders

Energy loss calculation for secondary distribution feeders requires installation of energy meter at each secondary feeder or a group of feeders if controlled by a single breaker at the distribution substation. The month wise energy loss in a secondary feeder is given by:

$$\begin{aligned} E_{loss\ in\ sec\ feeder} &= energy\ sent - energy\ billed - agricultural\ energy\ sales \\ &= E_S - (E_P + E_{HVS} + E_{HT} + E_A) \end{aligned} \tag{2}$$

where, E_S = monthly energy sent out on this feeder,

E_P = monthly energy billed from the report of private accounting agency,

E_{HVS} = monthly energy billed for high value services,

E_{HT} = monthly energy consumption of all the HT consumers on the feeder,

E_A = monthly agricultural energy sales on this feeder.

2.3 Energy loss calculation in low voltage distribution network

This again requires meters to be provided on the low voltage side of distribution transformer to obtain energy sent out. Further, billing database is used to compute the energy billed for different class of customers connected on the distribution transformer. The energy sales on each distribution transformer can be automatically calculated if customer-billing database is linked with the location of the distribution transformer. To achieve this, a location code has to be associated with each distribution transformer. The location code has been conceptualized in terms of district code, circle code, village/ town code censes code, ward and serial number of distribution transformer. Next, the consumers connected on each distribution transformer have to be identified. This is achieved by adding an additional field “transformer location code” in the consumer database. This enables to calculate energy billed for each distribution transformer automatically. This record of energy billed for each distribution transformer can be maintained in the billing database by the private accounting agency. Thus, monthly energy loss in low voltage distribution network down to a distribution transformer is given by:

$$E_{loss\ in\ LV\ network} = E_{TS} - E_{TB} \tag{3}$$

where, E_{TS} = monthly energy sent out in low voltage network down to a distribution transformer, and

E_{TB} = distribution transformer wise monthly energy billed from the report of private accounting agency.

2.4 Energy loss calculation in low voltage distribution feeder

As said earlier, a set of low voltage feeders emanates from a distribution transformer and travels to different load areas. Monthly energy loss for each such LV feeder can be calculated using the similar concept described earlier for distribution transformer wise monthly energy loss. The energy sent out can be achieved by installing separate energy meter for each LV outgoing feeder from a distribution transformer. Further, LV feeder wise monthly energy billed can be computed easily from the billing database if the identification codes for the LV feeders are added as a field in the billing database. To achieve this, LV feeder code has been conceptualized in terms of district code, circle code, village/town code censes code, ward, serial number of distribution transformer and feeder name. Next, adding an additional field “LV feeder code” in the consumer database can easily identify the consumers connected on each LV distribution feeder. This enables to calculate energy billed for each LV distribution feeder automatically. This record of energy billed for each LV distribution feeder can be maintained in the billing database by the private accounting agency. The difference of monthly energy sent out and energy billed gives the energy loss for each LV feeder emanating from any distribution transformer.

3. SALIENT FEATURES OF DEVELOPED SOFTWARE

A user-friendly software has been developed for the purpose of energy audit and accounting using the proposed methods to calculate the energy losses at different voltage levels in the power distribution system. The developed software is based on the web technology and uses 3-tier architecture. This software can be operated in two modes - monitoring and input mode through the Internet / Intranet Web browser. The energy loss information for a given distribution system can be retrieved by any one whereas data can be inputted only by the authorized users in input mode. The major features of the developed software are given below:

- The developed software provides energy loss information at different hierarchical levels of the power distribution utility. For an example, the power distribution organizational hierarchy in India is state, area, zone, circle, division, subdivision, distribution substation, distribution transformer and low voltage distribution feeders. This software generates energy loss report month, season and year wise both in the tabular and bar chart form. Apart from this, there is a provision to generate a consolidated energy loss report at the end of the year. Trend analysis feature of the software enables to check the impact of the corrective action taken to reduce the total energy losses. Energy information at a glance is available through this software and it is useful for higher-level authorities in the power distribution utilities.

- The developed software solution enables distribution utility to monitor the consumers in a user-friendly manner through the Internet. Under this, the complete list of consumers is available for a specific distribution substation, distribution transformer and low voltage distribution feeder. In addition to this, it provides a comprehensive list of these consumers who are overdrawing the power and also who are not paying the bills regularly both at the distribution transformer and LV feeder levels. All such information helps distribution utilities to take the necessary action very quickly in order to increase the cash flow.
- A simple mechanism has been provided to manage the consumer's data through the Internet. This includes adding new consumer connection, modifying the consumer related data, deleting a consumer connection and reallocating a consumer connection from one feeder to another feeder.
- The developed software provides comparative performance report for different part of the distribution system at various hierarchical levels of the utility on the basis of monthly energy loss.
- An easy interface between user and computer has been designed and developed in this software in the form of single line diagram both in the monitoring and data input modes. Initially at the time of software installation, it needs to draw the single line diagram of the distribution system and to input the relevant and necessary data in the input mode. Afterwards, it provides an easy access of all the major components (distribution transformers and feeders) and associated static information besides the energy audit information. This type of information helps distribution utility to manage the distribution infrastructure in a better manner.
- A generic search utility has been provided to trace a consumer connection and its details. Also, a distribution component such as transformer or feeder with its details can be retrieved very easily from the database using the search utility. This feature helps software users to go to the location of their interest directly.
- This software supports ODBC (Open Database Connectivity) interface to connect a commercial database such as ORACLE-8 or SQL Server 2000. Consequently, the developed software can retrieve the energy-billed information from the customer-billing database maintained by private accounting agencies if any. In case if billing database is not available, the software has provision (1) to manually feed the meter reading and (2) to feed the meter reading through the meter reading instruments (MRI). The software has also an optional feature of consumer bill calculation using the pre defined tariff structure for different class of consumers.
- An advanced feature of electricity bill payment mechanism has been incorporated in the software using Internet technology. Also, an interface has been provided to upload the metering information from the database to hand held computer and to download the billing information from hand held computer to the billing database.
- The developed software can be easily customized to fulfill the specific requirements of any distribution utility. These customizations include definition of the consumer connection code, tariff code for different class of consumers, overdrawn limit for different category of consumers, color codes in single line diagram at different voltage levels and organizational hierarchical structure of the distribution utilities. These customization features provide flexibility to use this software in varying environment of different distribution utilities.
- A separate module for administrator has been designed and developed as a part of the overall software system. This enables administrator to create new user, modify user profile, list the current users and remove an user. The administrator performs the major customization activities at the time of software installation.
- The other utilities such as printing the reports and logging the data and events with time stamp have also been integrated with the main energy audit software.

4. SOFTWARE ARCHITECTURE

A 3-tier architecture is selected to design and develop the EAAS utilizing the Internet technology. Fig.1 shows a simplified form of 3-tier architecture.

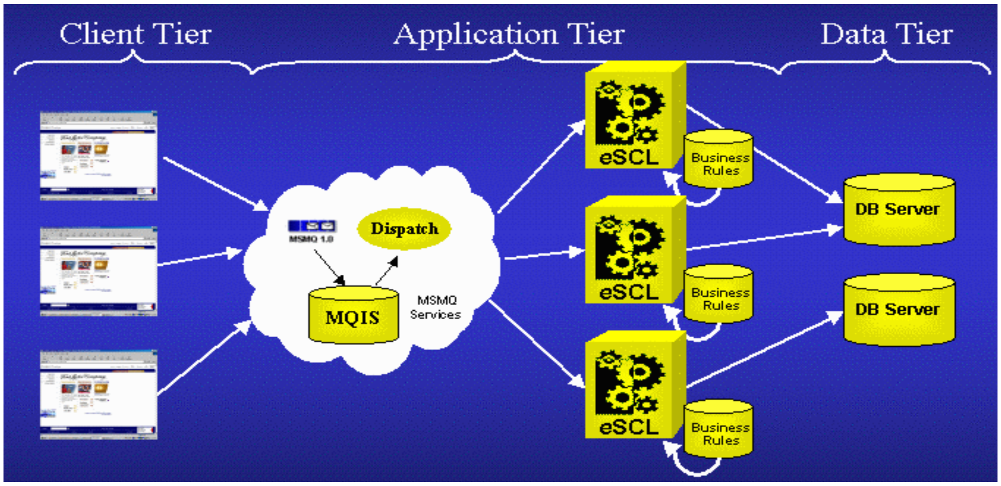

Fig. 1 3-tier architecture for EAAS

The 3-tiers in this architecture are generally named as client-tier, application-server-tier and data-server-tier. It is important to note that boundaries between tiers are logical. It is quite possible to run all three tiers on one physical machine. The main point is that the system is neatly structured, and that there is a well-planned definition of the software boundaries between the different tiers. These tiers are briefly described below:

4.1 Client-tier

It is responsible for the presentation of data, receiving user events and controlling the user interface. The actual logic (e.g. energy loss calculation) has been moved to an application-server. Now-a-days, Java-applets offer an alternative to traditionally written PC-applications.

4.2 Application-server-tier

This tier is new, i.e. it isn't present in 2-tier architecture in this explicit form. The application objects, that implement the actual internal logic of EAAS here, are available to the client-tier. This level now forms the central key to solving 2-tier problems. This tier protects the data from direct access by the clients. The object oriented analysis (OOA) aims in this tier to record and abstract energy audit processes in logical-objects. This way it is possible to map out the applications-server-tier directly from the CASE-tools that support OOA.

The term, "component" describes visual components on the client-side and configurable objects which when put together forms an application processes.

4.3 Data-server-tier

This tier is responsible for data storage. Besides the widespread relational database systems, existing legacy systems databases are often reused here.

4.4 The advantages of 3-tier architecture

As previously mentioned, 3-tier architecture can eliminate many problems of 2-tier architectures.

- The 3-tier architecture provides a clear-cut separation of user-interface-control and data presentation from application-logic. Through this separation more clients are able to have access to a wide variety of server applications. The two main advantages for client-applications are (1) quicker development through the reuse of pre-built application-logic components and (2) a shorter and modular test cycle.
- Re-definition of the storage strategy does not influence the clients. Relational Database Management System (RDBMS) offers a certain independence from storage details for the clients. It may be necessary to change database table attributes to suit the client requirements. Even in case of switching from a RDBMS to an Object Oriented Database System (OODBS) will not influence the client. In well-designed systems, the client still accesses data over a stable and well-designed interface, which encapsulates all the storage details.
- Application-objects and data storage should be brought as close as possible, ideally they should reside physically on the same server. This way especially, with complex accesses network load is eliminated. The client only receives the results of a calculation through the application-object.
- In contrast to the 2-tier model, where only data is accessible to the public, application-objects can place applications-logic or "services" on the network. As an example, energy loss report in the form of application objects will be available to the clients, and the logic for calculation of energy loss is made available on the server.
- As a rule servers are "trusted" systems. Their authorization is simpler than that of thousands of scattered client-PCs. Data protection and security is simpler to obtain. Therefore, it makes sense to run critical application processes, that work with security sensitive data, on the server.
- If bottlenecks occur in terms of its performance, the server process can be moved to other servers at runtime.
- It is also easy and faster to exchange a component on the server than to furnish numerous PCs with new program versions. The new application logics can be developed and put in the application server in such a way that the clients automatically work with the latest version from the exact date that it has to be run. It is, however, compulsory that interfaces remain stable, the old client versions are still compatible and proper configuration management with version control is in place. In addition such components require a high standard of quality control. This is because low quality components can endanger the functions of a whole set of client applications.
- It is relatively simple to use wrapping techniques in 3-tier architecture. As implementation changes are transparent from the viewpoint of the object's client, a forward strategy can be developed to replace legacy system smoothly. This requires defining the object's interface. However, the functionality is not newly implemented but reused from an existing host application. That is, a request from a client is forwarded to a legacy system, processed and answered there. In a later phase, the old application can be replaced by a state-of-the-art solution. It is also possible to interface legacy application with front end web based application tiers using third party software like Websphere, Weblogic, etc. If it is possible to leave the application object's interfaces unchanged, the client application remains unaffected.

5. SOFTWARE TESTING AND RESULTS

The Software Development Life Cycle (SDLC) has been followed for the development and testing of the energy audit software [3]. SDLC provides the guidelines for the systematic development of the software and also ensures the quality of the software. The basic activities of SDLC are:

- Requirement analysis
- Design of the software
- Coding of the software
- Testing of the software
- Implementation

To meet the objectives of energy audit and accounting, a detailed requirement of the software is laid down in the form of Software Requirement Specification (SRS) after analysis phase. The SRS is prepared as per the IEEE standard [4]. The SRS of the software elements is organized as following:

- Introduction: Purpose, Scope, Definition, Acronyms and abbreviation, Overview
- Overall Description: System interface, User interface, Hardware interface, Software interface, Memory constraints, Operations
- Specific Requirements

Each component of software has been designed in a modular fashion. The design of the software took place in two phases – (1) system level design and (2) detailed design which is documented in Software Design Document (SDD). After the coding phase, computer codes are available that can be executed for testing purposes. Software testing is performed not only to detect errors introduced during coding, but also errors introduced during the previous phases – requirement analysis and design. Consequently, different levels of testing are performed for the developed software as given below:

- Unit testing
- Integration testing
- System testing
- Acceptance testing

Unit testing is performed on each module separately. The coder himself often performs this test at the time of the coding of the module. The purpose is to detect the coding error in different parts of the module. Once a module is made and tested separately, it is integrated with the other modules to form the complete software system. During integration of modules, integration testing is performed. The objective of this testing is to detect design errors, while focusing on testing the interconnection between modules. Once all the modules are put together to form the software system, system testing is performed. Here, the software system is tested against the software requirements to see if all the requirements are met and software system performs as specified by the requirements (SRS). Finally, acceptance testing is performed to demonstrate the operation of the software system on the real life data of the client.

Proper selection of the test cases is essential in order for successful testing. There are two different approaches to select test cases – functional testing and structural testing. In functional testing the software or the module to be tested is treated as a black box, and the test cases are decided based on the specifications of the software system or the module. For this reason, this form of testing is also called "Black Box Testing". The focus here is on testing the external behavior of the software system. In structural testing the test cases are decided based on the logic of the modules to be tested. A common approach here is to achieve some type of coverage of the statements in the code. One common coverage criterion is the statement coverage, which requires that test cases be selected so that together they execute each statement at least once. Structural testing is sometimes called "Glass Box Testing". The forms of testing are complementary: one tests the external behavior and the other tests the internal structure. Often structural testing is used only for lower levels of testing, while functional testing is used for higher levels.

Testing usually starts with a test plan. This plan identifies all the testing related activities that must be performed and specifies the schedule, allocates the resources and specifies guidelines for testing. The test plan specifies conditions that should be tested, different units to be tested, and the manner in which modules are integrated together. Then for different test units, test case specification document is produced, which lists all different test cases, together with the expected outputs, that will be used for testing. During the testing of the unit the specified test cases are executed and the actual result is compared with the expected output. The final outcome of the testing phase is the test report and error report. Each test report contains the set of test cases and the result of executing the code with these test cases. The error report describes the errors encountered and the action taken to remove the error.

A representative set of data has been taken from a power distribution utility. These data have been fed to EAAS to prepare the necessary database for a distribution network and consumer billing information through the Internet. Afterwards, a series of test cases have been designed and performed for acceptance testing in the utility environment. Some of these test cases are described below:

- Aim of this test is to verify the feature of 'feeder wise annual energy loss report generation' in the developed software. For this, a low voltage feeder "FDR1012" and the period of 1993-2002 are selected through the graphical user interface at the client-tier level. The yearly energy loss report, available from the developed software, is given in Fig. 2 in bar-chart form. This verifies the feature of annual energy loss report generation in graphical form.

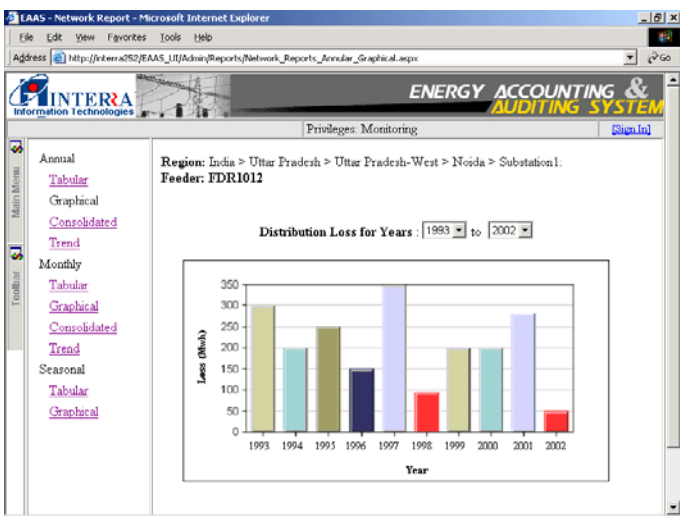

Fig. 2 Energy loss report generation

- Purpose of this test to verify the feature of network presentation through the single line diagram, a distribution substation has been selected from the menu-bar. As a result of this, the single line diagram, as depicted in Fig. 3, appears on the computer screen for this substation. An authorized user in input mode can edit the single line diagram for this substation. Editing single line diagram includes adding, modifying and deleting a component such as transformer, feeder and circuit breaker.

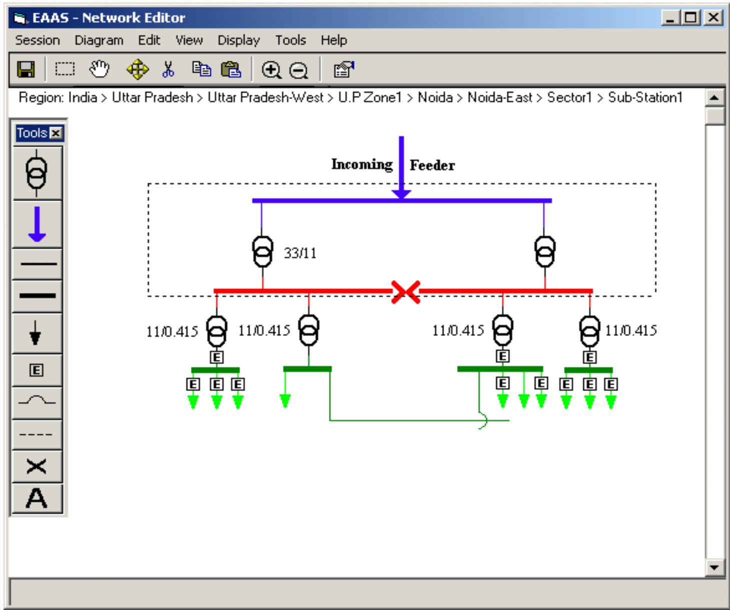

Fig. 3 Graphical representation of network

- This test case is designed to verify the customization feature to define the consumer connection code in a secure mode. It is found in the software that the system administrator can define the structure of consumer connection code at the time of software installation from the dialog-box given in Fig. 4.

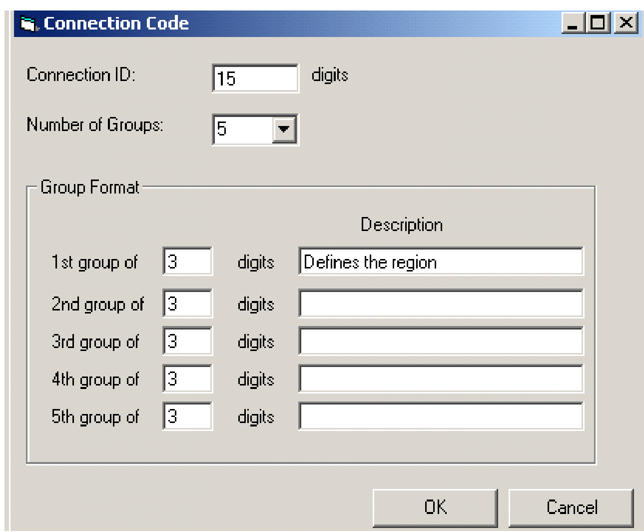

Fig. 4 Dialog-box for consumer connection code definition

- As said earlier that this software has capability to perform monthly trend analysis of energy losses. Fig. 5 shows the trend of energy loss on monthly basis. Analysis of this graph shows reduction of the energy loss after the month of September. This is because of the improvement in the bill collection method in September and onwards.

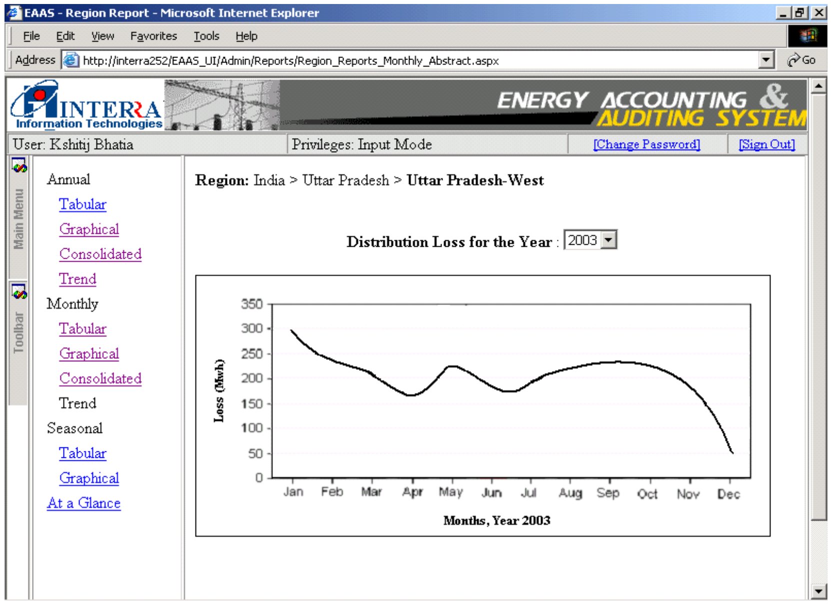

Fig. 5 Energy loss trend analysis

- The software is expected to give a list of consumers who are overdrawing power from the sanctioned value on a specific feeder. To verify this feature, the feeder named “Feeder-1” and month “March-2003” have been chosen in the monitoring mode of the software. In response to this, the software gave a list of consumers who have overdrawn power in the month of March-2003 on the feeder “Feeder-1”. This is shown in Fig. 6.

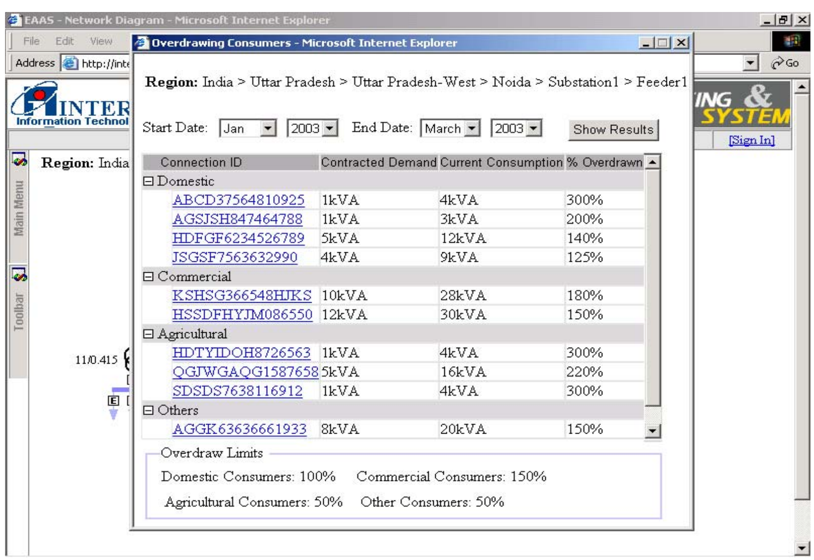

Fig. 6 List of consumers overdrawing power

All these test reports verify the proper functioning of the developed software in actual utility environment.

6. CONCLUSIONS

Energy Audit and Accounting, in general, has been addressed for power distribution system in this paper. Methods to calculate energy losses at different voltage levels have been proposed and discussed. Based on these methods, an user friendly software for energy audit and accounting has been designed and developed using Internet technology. The design of the software is based on the concept of 3-tier architecture. The features of this software and the process of development are also described in detail. There after, a set of test cases have been designed and performed in utility environment. Test results validate the proposed method of loss calculation and indicate proper functioning of the software. The developed software solution is useful to determine the high loss feeder at different voltage levels in the distribution system.

7. ACKNOWLEDGEMENTS

The authors would like to express his deepest sense of gratitude to late Professor Sachchidanand for his continuous pin pointed technical discussions and suggetions in completing this work successfully. Authors are also thankful to Professor S. C. Srivastava for his valuable comments, continuous encouragement and moral support at various stages of this work. At last but not the least, both the authors cannot forget the utility persons Er. R. K. Jain, Er. P. K. Yadav, Er. R. Singh, Er. R. K. Goel and Er. S. Bhattacharya from Uttar Pradesh Power Corporation Lilmited, Lucknow who gave their feedbacks and pragmatic suggestions while developing this software solution.

8. ACRONYMS

T & D	=	Transmission and Distribution
EAAS	=	Energy Audit and Accounting Software
LV	=	Low Voltage
HV	=	High Voltage
CASE tool	=	Computer-Aided Software Engineering tool
OOA	=	Object Oriented Analysis
OODBS	=	Object Oriented Database System
SDLC	=	Software Development Life Cycle
SRS	=	Software Requirement Specifications
SDD	=	Software Design Document

9. REFERENCES

[1] Khodr, H.M.; J. Molea, Garefa, I.; Hidalgo, C.; Paiva, P.C.; Yusta, J.M.; and Urdaneta, A. J. 2002. Standard Levels of Energy Losses in Primary Distribution Circuits for SCADA Application. *IEEE Transactions on Power Systems* 17(3): 615-620.

[2] Oliveira, C.C.B.; Kagan, N.; Meffe, A.; Jonathan, S.; Caparroz, S.; and Cavaretti, J. L. 2001. A New Method for the Computation of Technical Losses in Electrical Power Distribution Systems. In *Proceedings of Electricity Distribution, Part 1: Contributions. CIRED. 16th International Conference and Exhibition on (IEE Conf. Publ No. 482)*. June, 5: 328-328.

[3] Pressman, R. S. 1997. *Software Engineering*. 4th edition: McGRAW-Hill.

[4] IEEE Recommended Practice for Software Requirements Specifications. 1993. IEEE Standard 830-1993.