MEĐUNARODNO REGIONALNO SAVETOVANJE O ELEKTRODISTRIBUTIVNIM MREŽAMA

REGIONAL CONFERENCE AND EXHIBITION ON ELECTRICITY DISTRIBUTION

Crna Gora, Herceg Novi, 5 - 8. oktobar 2004. / *Montenegro, Herceg Novi, October 5 - 8, 2004*

FRONT-END APPLICATIONS FOR SCADA SYSTEM IN JP EDB

V. Stojičić, JP Elektrodistribucija Beograd, Srbija i Crna Gora
D. Milanov, JP Elektrodistribucija Beograd, Srbija i Crna Gora

INTRODUCTION

The work on the new SCADA system in JP “Elektrodistribucija Beograd” has been going for several years now. It includes creating the digital representation of the network and its elements, information system that would contain this data, and applications that would present this data to the end user. This paper presents two front-end applications developed in our company, and the information system that supports them. Communication between various parts of the system is also explained, and the possible further developments are discussed.

DisNetView

DisNetView is the client-side viewer for one-line distribution network schematics. It is used to display 110, 35 and 10 kV networks, and the X/10 kV and X/35 kV transformer substations. Apart from displaying network schematics, it is also used do display, find and process various data on network elements.

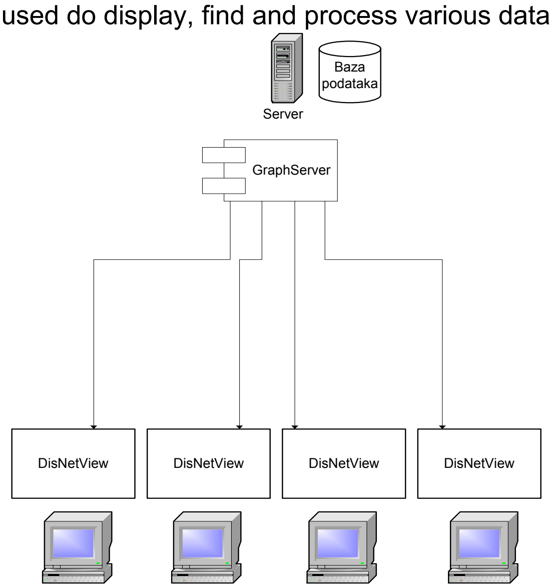

Figure 1 - Server for graphical data and clients

This data comes from various sources, MS Access and Sybase databases, through custom middle-tier servers. Graphical data is stored in MS Access databases and is delivered through a server application called GraphServer. Technical data on network elements is stored in Sybase database and is accessed through Zora information system (application server and client objects).
The graphical server reads data from database, parses it, and creates and links the resulting objects. These objects are then serialized in a data stream and delivered to clients (DisNetView application). Another server responsibility is to receive requests to change state of network elements. There are three allowed states for network elements – on, off and grounded. When a user attempts to change a network element's state, the request is sent to the GraphServer. The server processes the request, writes the new state to the database and informs all the clients that a change has been made to this element.
Since the client application cannot work without the connection to the server, we have put special attention to the communication between the clients and the server. The possible bottlenecks in this communication have been analysed and measures have been taken to prevent them. The process of informing clients that a change has happened to the network is one example. In this situation, the server has to call each client application and wait for them to return from the called function. There are two problems with this – the client can be, at this time, involved in some lengthy process or calculation; also, the client might be non-gracefully disconnected from the client. The solution for the first problem was to create a separate thread of execution in the clients that monitors calls from the server and returns the control immediately, postponing the processing of the request until the main thread is available. For the second problem, however, we found that the simple solution is the best – if the client does not respond in a certain amount of time, it is disconnected from the server.
The communication technology used in this package is DCOM – Distributed Component Object Model. Our choice to use this technology was based on several reasons. Microsoft Windows 2000 is the main supported platform, and DCOM is the technology that is preferred to CORBA for this platform. DCOM also gives a higher level of abstraction and is easier to configure and use than TCP/IP. It gives a good level of security using system authorisation and authentication. That way the company's system administrators are also charged with administering access to this package using the standard Windows security tools.
Error handling is done separately on the client and on the server. All errors and warnings are stored in log files, and the important (unrecoverable) ones are presented to the client. The information collected in the log file is critical in maintaining the integrity of the data presented to the user. This data comes from databases that are, in turn, created from AutoCad drawings. The process of updating the AutoCad drawing, and exporting it to the database can produce errors, that are best caught while parsing the drawing from the database, or even on-screen, by end users.
The network elements are displayed in two ways, as polylines and symbols. Polylines are mainly used for power lines and bays. Every element type can have its special symbol, fully customisable in shape and colour. Furthermore, each symbol has sub-symbols for every element state (on, off, grounded). A network element can be displayed using a combination of symbols, polylines and text, and each of these graphical primitives can have a different foreground and background colour. The polylines can be displayed in a variety of line types – solid, dashed, dash-dot, etc, and in desired width. Symbols can contain any combination of lines, rectangles, polygons and circles, with solid, coloured background or transparent. These options allow highly customised symbols, to suit the users' particular needs.
Designing this application, we have decided to present the most vital information using only shapes and text. Information that is not essential for the work, and that can be switched on and off, is allowed to be presented in colour, as well as in shapes and text. Thus, calculations, statistics and custom database data can be presented by colouring network elements or with texts containing data, but element states and types are presented using only shapes.
Another design decision was to display the network and substation diagrams in different windows. The user switches between 10 kV, 35 kV and 110 kV network in the main application window, while the substation diagrams are shown in an auxiliary window. The user preferences are stored separately for these two windows, to allow for maximum efficiency.

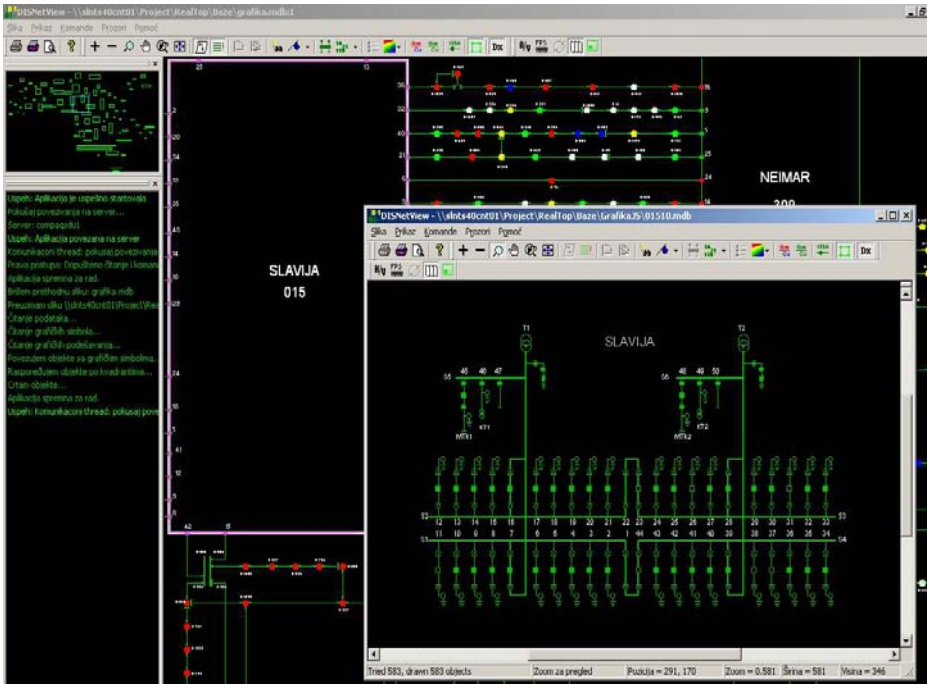

Figure 2 - Main window with 10 kV network and separate window displaying a 10 kV substation of a 35/10 station

The application implements all the standard ways to present the diagram and manipulate the on-screen presentation. The diagram can be panned or scrolled, zoomed in and out. A simple “undo” (or “go back”) function is also implemented. A special map window is also displayed. It presents the small-sized picture of the network, and allows the user to quickly move around the diagram. This is very useful, since the largest diagram, that of the Belgrade’s 10 kV network, contains more than 20000 objects, spread on a very wide area. Another technique used to improve the user experience while browsing the diagram is the use of bookmarks. The bookmarks can be placed anywhere on the network, on any zoom level, and on any network voltage. When a user selects a bookmark, the appropriate diagram is loaded (if it is not already presented) and the bookmarked spot is centred on the screen.
The diagrams can be displayed using the full-screen mode. The toolbars, menus and frames are hidden, and the scrolling is done by moving the mouse pointer to the edges of the screen. User functions are accessible using the context menu.
Searching through objects is done using a search form. The user specifies the text to search for, and optionally the type of required network object. All the results are displayed, and the selected result is centred on the screen.

Figure 3 - Search form

One of the most frequently used features is browsing through various diagrams, from network to substation and back, and across voltage levels. Certain objects are used as “gateways” to other diagrams. For example, the user can get from the 10 kV network diagram to the 10 kV substation of a 110/10 kV

station by clicking on the symbol for the 110/10 kV station on the network diagram. Traversing from the 10 kV to the 35 kV side of a 35/10 kV station can be done by clicking on the symbol for the transformer.
To change the state of a network object, the user has to select the object, and then to change its state. The request is sent to the server, and after receiving the information from the server that the state is changed, the resulting change is displayed on screen in all the active client applications.
There are several custom functions implemented in the client application. For all of these functions, the application calls various application servers to send the request and receive results. These functions are, at this moment, coloring of the elements using desired criteria, displaying load diagrams, access to the data on the cables, and access to the IPS SDU database. There are numerous other functions planned, and they will be implemented in accordance with the priorities in the company.

Figure 4 - Load diagram displayed over the substation scheme

The further development of this application is dictated by user requests. At this moment, the application is being actively used in several sectors of our company. As more and more people use the application, the requests to customize its various aspects are multiplying. Some of these are implemented using external application, and some require rewriting the code of the client application. The idea is to standardize the process of adding new functionality by using external applications and libraries, so that the code of the application is not changed too often.

Raster DisNet

Application Raster DisNet provides searching and displaying objects of 1KV dispatching network. There are several ways to find an object or a part of the network:

- By choosing a section from city map
- Through address
- Through section name
- Through number of station

After specifying the search criterion, the program will load the part of the city that contains the object (part of the network) and surrounding sections.
When the part of the city is loaded, the network image is displayed, by default. Also, the application loads one of the background layers:

- The ortoplan (aerial photo)
- City images
- Network images

Figure 5 - The background layers – ortoplan, city images and scaned network sections

The application also combines the network image with the ortoplan and shows transparent city images or network images on the ortoplan.
Application incorporates several functions, including zooming, panning, using the Magnifying Glass and using the Pan Window. The Pan Window is a small window that displays a scaled view of a bitmap, which is also being displayed in the Main Control's window at a size that would require scrolling. The Pan Window maintains the image's aspect ratio. A coloured Pan Rectangle is displayed in the Pan Window, to indicate the portion of the image currently being displayed in the main window. When a user clicks inside the Pan Window and moves the mouse, while holding down the button, the Pan Rectangle will move with the mouse pointer.
Application provides a "magnifying glass" as a means of "zooming in" on the image loaded into the Main Control. As the magnifying glass is moved over an image, the zoomed-in view is also displayed.

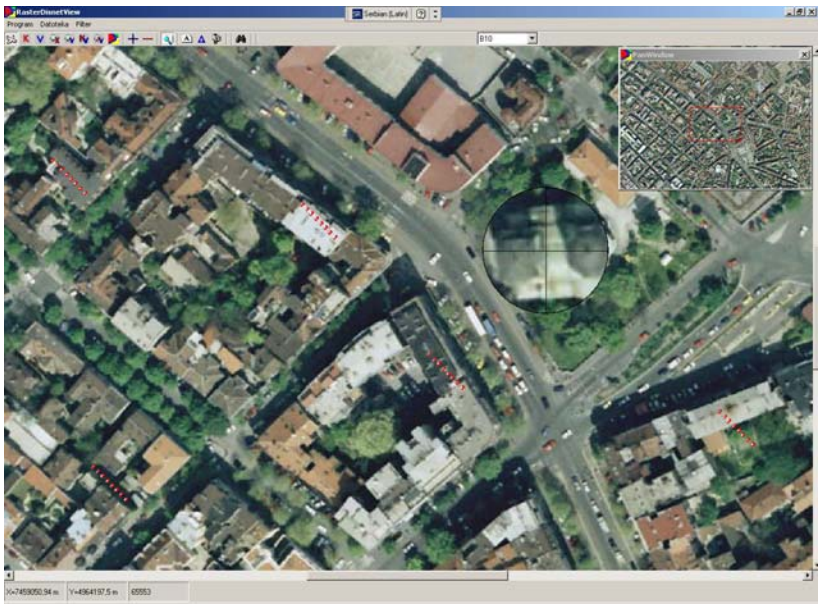

Figure 6 - Using the Magnifying Glass and Pan Window

Vector objects are associated with a bitmap as an overlay of the displayed image. Clicking on the object will give you information about it.
With the basic brightness and contrast functions you can change the intensity or contrast of the bitmap, or you can adjust the gamma correction. Also you can get some effects like sharpen, blur, contour filter etc.

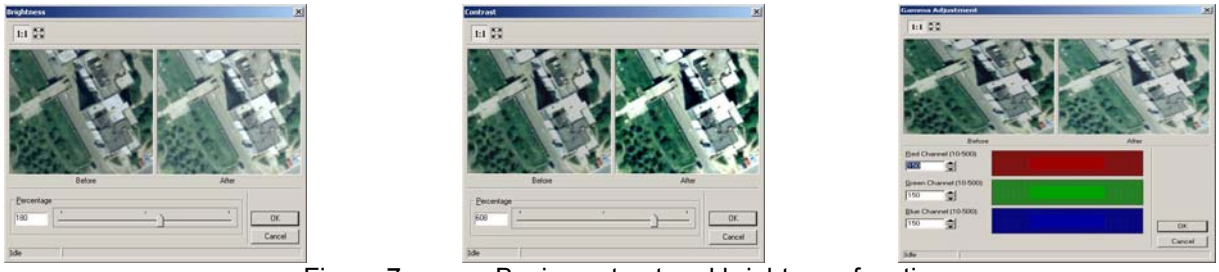

Figure 7 - Basic contrast and brightness functions

The application also provides measuring the distance between two points of the image.

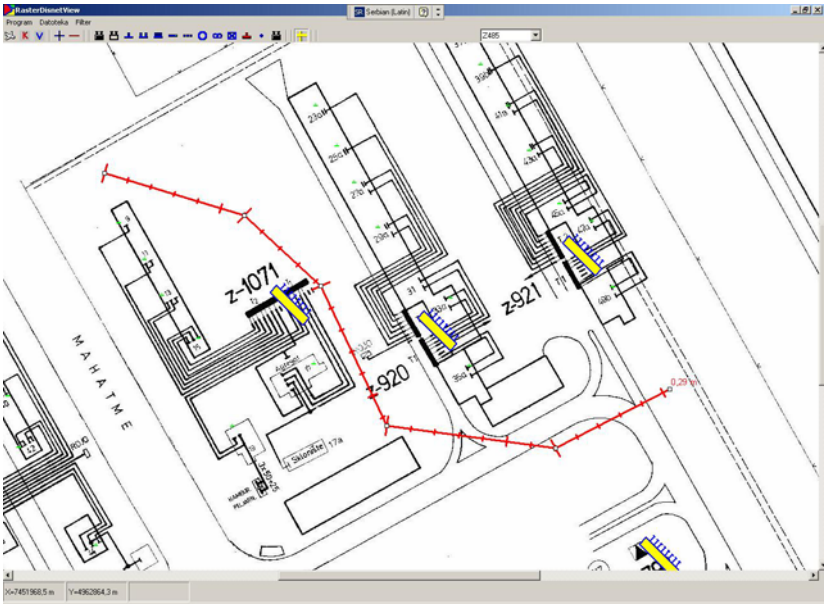

Figure 8 - Measuring the distance

In design mode, you can select tools from the toolbar and draw vector objects over the image in the window. You can then use the right mouse button to get a pop-up menu that is tailored to each object type.

The popup menu is context sensitive. It lists only the properties that can be changed on the selected object or objects. The popup menu lets you do the following:

- Undo the last user action.
- Select all vector objects in the container.
- Delete all selected objects.
- Change the default line width and style properties
- Change the default fill mode (transparent, translucent, or opaque) and the fill pattern.
- Change the polygonal fill mode, which controls which areas are filled when lines cross.
- Change the default foreground and background colours.
- Change the font selection and font characteristics.
- Scale all selected objects.
- Rotate all selected objects.
- Save object in the database.
- Save object connection with other objects

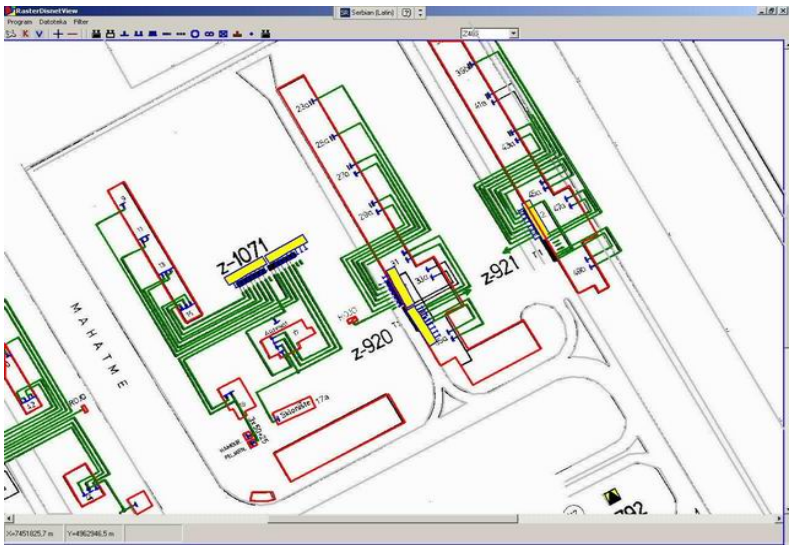

Figure 9 - Part of the network

If you use the popup menu to change a property, the value that you specify is applied to the current selection, and it becomes the default for the next object. The popup menu is also affected if more than one object is selected. In that case, when you click the right mouse button, the popup menu lets you change the properties of all the selected objects, and it lets you delete all the selected objects.
At this moment, the application supports the following types of network objects:

- Station
- Connection box
- T-junction
- Conductor
- Cable
- Open point
- Pole

You can also define the new type of network object.
Application let you lock and unlock vector objects using a key. By setting the user for a vector object, modification to the object cannot be made unless the correct user is logged on.
Saving and printing images is also possible. You can save or print:

- Section
- Region
- Screen

Technical Information System – IPS SDU EDB

The first step for building the technical information system IPS SDU EDB (“Informacioni podsistem sistema daljinskog upravljanja Elektrodistribucije Beograd”) was to create a common data model. It was described through a UML object model, in numerous structure diagrams and packages. The model is based on IEC 61968 and IEC 61970 standards. The drafts of these standards propose not only the data model, but also the software architecture, and the ways to exchange data between different software modules, using XML.

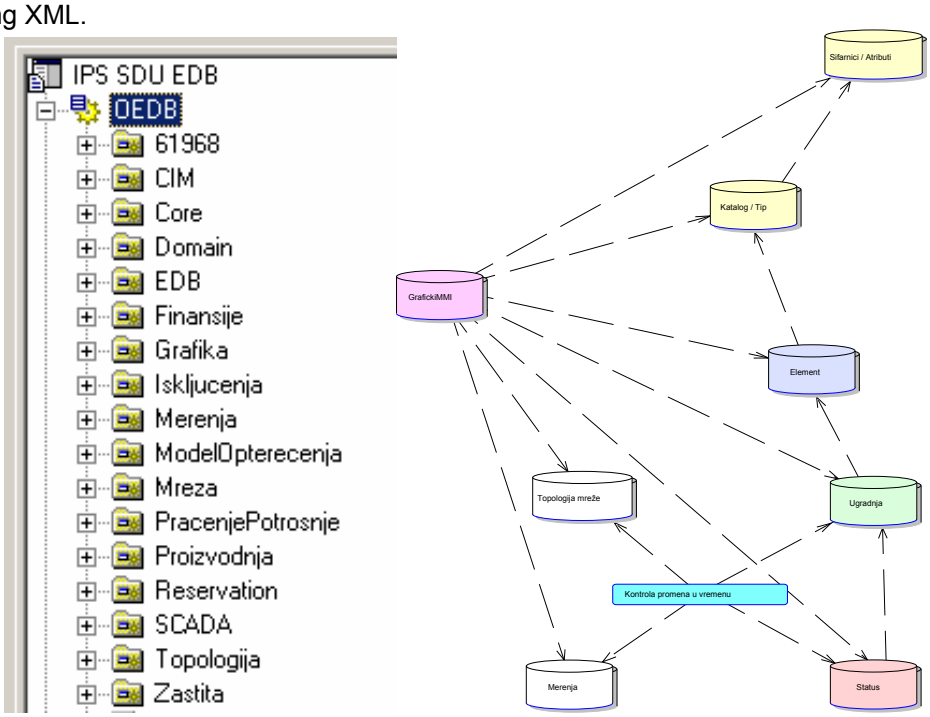

Figure 10 - Packages in the IPS SDU EDB model

The n-tier architecture imposed by IEC standards requires a specific software environment. It contains the following layers:

- an SQL database, to persist the data on all objects in the system
- a middle tier (middleware), server application that creates objects from persistent data, and provides access to them and their functions; all changes in object attributes are sent to the database, that processes and persists them in a relational model, using specially designed object-relational mapping

- the user interface (the “front end”), which contains forms and components that present the objects from the middleware, and enable the user to browse and change data, and perform custom commands
- the web services, that provide other applications with the required data from the database and the middle tier

The process of analysis and design of IPS SDU EDB has included using the following methods, standards and references:

- UML object model to present the model of the information system
- XML schemes to describe object attributes in the model
- Results of the workgroups 13 and 14 in the IEC TC 57 (International Electrotechnical Commission Technical Committee 57: Power System Control and Associated Communications)
- IEC 61968 1 to 10 series of standards, aimed to describe inter-application integration
- IEC 61970 301, 302 and 303 Energy Management System Application Program Interface - EMS API - Common Information Model - CIM
- Creating a relational database model synchronized with the object model
- Specification of protocols to exchange data between clients and middleware using CIM XML/RDF notation

Both presented applications use the IPS SDU EDB information system, through web services, .NET Remoting and COM protocols.

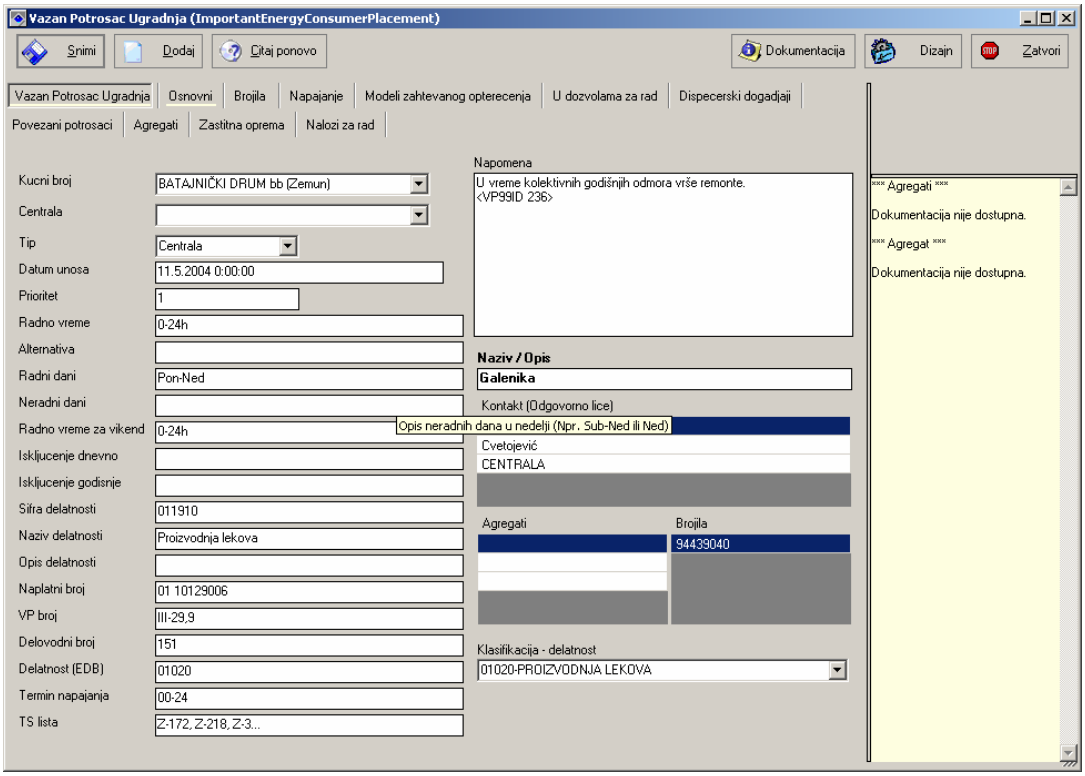

Figure 11 - An IPS SDU EDB Edit form

CONCLUSION

The information system and front-end applications for the new SCADA system in JP “Elektrodistribucija Beograd” have been deployed in the several sectors of this company. At this moment, we are working on several new features, particularly on further integration with the technical information system. Our plans also include connection with the on-line data from the communication subsystem, and displaying it to the end user, which will allow us to partially supplement the current front-end used since 1980s in the near future.