Software development for incremental integration of GIS and power network analysis system

Milorad Ljeskovac*, Imre Lendak**, Igor Tartalja*
* University of Belgrade, School of Electrical Engineering, Belgrade, Serbia
** University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia
mljeskovac@gmail.com, lendak@uns.ac.rs, tartalja@rcub.bg.ac.rs

Abstract—The paper discusses an automation procedure for integration of Geographic Information System (GIS) and Power Network Analysis System (PNAS). The purpose of PNAS is to provide analysis and optimization of electrical power distribution system. Its functionality is based on a detailed knowledge of electrical power system model. Automation procedure problems are identified and concepts of an implemented software solution are presented. The core of the solution is the software for implementation of an internal model based on IEC 61970-301 CIM standard. Second part of the solution, which this paper is focused on, is a software layer that detects difference between two relevant GIS states just before successive synchronizations with PNAS. Using “divide and conquer” strategy, the detected difference is split into groups of connected elements. Those groups are used for update of PNAS to make it consistent with GIS. The paper also explains an algorithm for detection of the difference between two states and a procedure for formation of groups, as well as the conditions and limitations of automation procedure for incremental integration.

I. INTRODUCTION

Manual integration of Geographic Information System (GIS) and Power Network Analysis System (PNAS) cannot keep the pace with dynamics of changes inside Power System (PS). Frequent changes inside PS are accumulating inside GIS and there is a need for their periodic transfer into PNAS. The transfer has to be efficient and reliable. Even though GIS and PNAS are often found inside PS, and there is a need for their integration, commercial GIS systems do not offer support for automated integration with PNAS system.

Fully automatic integration of these two systems is hard to achieve. A detailed list of all the problems, which have to be resolved for their integration, can be found in [1]. While fully automatic solution is yet to come, there is a need for partial automatization of integration through simplification of its procedure and reduction of volume and need for manual data entry. This solution has to fit seamlessly into already existing business processes where GIS is the central source of data. This paper presents a solution that fulfils those requirements. Although the solution does not guarantee a full automatic integration, its usage is far more efficient and reliable than manual integration. It significantly reduces the volume of data that has to be manually synchronized.

The remaining part of the paper is organized in the following sections. Problem explains the role of GIS and PNAS systems inside PS, the need for their integration and the problems that occur on the way. Incremental integration describes the procedure for detection of difference between two relevant GIS states just before successive synchronizations of two systems, and presents the way of splitting of the detected difference into smaller groups, which are used for PNAS update. Discussion of results evaluates the solution and gives a short overview of its usage by two clients. The Conclusion summarizes the most important experiences and provides directions for further research and development.

II. PROBLEM STATEMENT

In this section, the roles of GIS and PNAS inside PS will be described. General integration problems of these two systems will be presented and a concrete problem, which this paper addresses, will be explained.

The role of GIS is management and visualization of spatial data. Data are objects in space, which represent inventory of one PS and include substations, conducting equipment (inside or outside of substations) and conducting lines. It is widely used in versatile fields. In PS it represents the central point for many technical and administrative business processes.

PNAS is specific for PS. It helps to manage and maintain the system. Its role is to increase the overall efficiency of the system by anticipating and preventing the problems that lead to disconnection of customers. To be able to fulfil its basic functionality it needs a detailed knowledge of PS electrical model. It also needs all information about equipment status changes and measurement results inside processing units in a relatively short time interval after the change has occurred.

What is common for both systems is that they are modeling PS, but in different ways and for different purposes. It would be ideal for both systems to work on one, shared data model. In the absence of the shared model, the problem is usually resolved through system integration, which is the subject of this paper.

The paper describes a developed procedure for incremental integration. The integration implies a transfer of changes from spatial GIS model into electrical PNAS model. The procedure also checks the validity of data that represent transferred changes. Frequent synchronizations indirectly contribute to the overall validity of GIS data. GIS elements are manually supplemented with the data that represent their electrical characteristics. Supplemented data are not part of GIS data model, and GIS has limited abilities for checking their completeness and validity. Connections among GIS elements are defined by spatial connectivity rules that exist among
those objects. These rules are not sufficient to guarantee a full and precise electrical connectivity of elements. Other business processes do not depend on supplemented electrical properties, and the problems with these data stay unnoticed all the way down to the moment of integration with PNAS. Validity and proper connectivity of exported electrical data is a precondition for successful synchronization. It has to be provided by proper planning and discipline during manual GIS data entry. During synchronization, discovered errors in the exported data have to be fixed first in GIS, and then the whole export has to be done again.

Integration uses internal data model based on IEC 61970-301 [2] CIM standard. CIM is described by its UML model and defines dictionary and basic ontology needed for data exchange in the context of electrical power transmission and distribution inside a PS. It is used for derivation of other specifications like XML and RDF schemas, which are needed for software development in integrations of different applications.

CIM fundamentals are given in the above-mentioned document [2], which focuses on needs in energy transmission where related applications include energy management system, SCADA usage, planning and optimizations. IEC 61970-501 [3] and IEC 61970-552 [4] define CIM/RDF and CIM/XML format for power transmission. IEC 61968-13 [5] defines CIM/RDF format for power distribution. IEC 61968 [6] series of standards extend IEC 61970-301 to meet the needs of energy distribution where related applications among others include distribution management system, outage management system, consumption planning and metering, inventory and corporate resources control, and GIS usage.

III. INCREMENTAL INTEGRATION

GIS data export and assumptions, which the presented solution is based upon, will be explained first. After that, the initial state for procedure of incremental integration and detection of difference between two relevant GIS states will be defined and explained. Problems that emerge during update of PNAS will be analyzed and a solution how to solve them, based on splitting detected differences during manual GIS data entry. During synchronization, discovered errors in the exported data have to be fixed first in GIS, and then the whole export has to be done again.

Integration uses internal data model based on IEC 61970-301 [2] CIM standard. CIM is described by its UML model and defines dictionary and basic ontology needed for data exchange in the context of electrical power transmission and distribution inside a PS. It is used for derivation of other specifications like XML and RDF schemas, which are needed for software development in integrations of different applications.

CIM fundamentals are given in the above-mentioned document [2], which focuses on needs in energy transmission where related applications include energy management system, SCADA usage, planning and optimizations. IEC 61970-501 [3] and IEC 61970-552 [4] define CIM/RDF and CIM/XML format for power transmission. IEC 61968-13 [5] defines CIM/RDF format for power distribution. IEC 61968 [6] series of standards extend IEC 61970-301 to meet the needs of energy distribution where related applications among others include distribution management system, outage management system, consumption planning and metering, inventory and corporate resources control, and GIS usage.

Before the synchronization between GIS and PNAS can start, we need to align those two systems first. It can be done manually or by some bulk export of data from GIS to PNAS. It is important that after the initial alignment of systems all relevant feeders’ files be exported from GIS. This is definition of initial state, which is needed for start of incremental integration. After each successful synchronization, which represents an increment in integration, states of those systems will be aligned again.

Fig. 2 illustrates the role that feeders play in detection of differences between two relevant GIS states at the moment of synchronization with PNAS. Changes inside GIS affect only certain number of feeders. For incremental integration, it is enough to export only those feeders that have been changed. Current state of the changed GIS is described by exported all feeders (changed and unchanged from previous synchronization). All unchanged feeders from previous synchronization describe previous GIS state, which corresponds to the...
current state of PNAS. After successful integration, all GIS feeders, with changed feeders included, are kept for the next synchronization where they will play the role of previous GIS state.

XML parser uses a defined XML schema to validate all feeders that belong to one relevant GIS state and, after that creates set of CIM/XML objects, which correspond to simple elements of conducting and measurement equipment, bays and substations. The set can be represented as an array of subsets, each containing objects of the same type because comparison has sense only with objects of the same type. Object can be accessed through its unique identifier, provided by GIS, which is the central source of all data. After all feeders are parsed, created objects are linked together according to the relations that are kept inside CIM/XML attributes. Containment relations (substation-bay, bay-conducting and measurement equipment) and CIM topology, which then can be used for finding of conducting and measurement equipment neighbors, are created this way. The same procedure is repeated for both compared states. For each object, the procedure sets an attribute for identification of the state.

Comparison of two relevant GIS states is performed by finding symmetric difference \[ \Delta \] of the two sets:

\[
\Delta(m1, m2) = (m1 \setminus m2) \cup (m2 \setminus m1)
\]

Total difference, is a union of symmetric differences of all subsets, each of which contain objects of the same type. It is calculated with generic function, which, as its arguments, accepts two subsets of objects of the same type from two relevant GIS states and produces their symmetric difference. After applying generic function on all existing subsets, resulting detected difference can be represented with three sets: added, changed and removed elements. By traversing a subset from the current state and comparing it with corresponding subset of the same type from previous state, generic function detects the sets of:

1. added elements - where unique identifier exists in the current but not in previous state
2. changed elements - where unique identifier exists in both states but elements are not the same.

By traversing a subset from previous state and comparing it with corresponding subset of the same type from the current state, generic function detects the set of:

3. deleted elements - where unique identifier exists in previous but not in the current state.

Detected difference represents a list of object references from the previous (for deleted elements) or the current state (for added and changed elements). Detection of difference in case of addition, change or deletion of an element inside substation automatically adds the substation reference in the set of changed elements. Only current state objects, which represent changed elements, have internal reference to a corresponding object from the previous state. During analysis of a detected difference, referred object can provide the information about the state it belongs to. If it belongs to the previous state, it is a deleted element. If it belongs to the current state, it can be determined whether it is a new or changed element based on existence of corresponding object internal reference.

Elements in the detected difference, which are references to objects inside substation, are removed from the difference because the difference already contains the substation reference. The synchronization of changed substation is performed by its deletion and recreation. This will also synchronize changed and added elements inside the substation.

By updating PNAS with the detected difference, its state will become consistent with current GIS state. Update could be done by sending deleted, added and changed elements one by one. Taking into account that accessing PNAS initiates a CORBA request, this solution would be inefficient. On the other side, such a solution would provide the simplest diagnostics when something goes wrong because it is a strait forward action to find problematic element. Another possibility is to send the whole detected difference inside one request, which would be efficient, but if something goes wrong the diagnostics would be much more complex. Deleting non-existing element inside PNAS is not a problem and can be ignored. Updating PNAS with the detected difference that includes added and changed elements can fail in at least the following two cases:

1. emerging of a new equipment structure inside one of CIM/XML bays, which does not exist in PNAS
2. manual change inside PNAS which is not synchronized with GIS

These types of problems are too rare and their automation cannot justify increase in complexity of solution and potential introduction of dependency on the used PNAS. An error during the update needs to be found and fixed manually. As the number of elements in update request grows larger, finding the error gets harder. In order to make this process more efficient and user friendly, the solution applies “divide and conquer” strategy, and splits the detected difference of added and changed elements into isolated groups of connected elements. This reduces the scope of error search to just one group instead of the whole detected difference. Update of PNAS, after removal of all deleted elements, is done individually with each group of added and changed elements instead of the whole detected difference. Errors are fixed by reorganization of data inside GIS or PNAS. After data reorganization, the whole synchronization procedure is repeated.

Forming of groups with connected added and changed elements (only “groups” in the rest of the text) represents a procedure of exhaustion of two resulting sets produced at the end of difference detection phase: the set of added and the set of changed elements (“detected difference” in the rest of the text). Each iteration extracts one group, isolated from the rest of detected difference. Isolation assumes that the group elements do not have neighbors or that all their neighbors are unchanged. This procedure changes only two resulting sets, which at the end become completely empty. Connectivity of elements is defined by topological relations inside CIM/XML and, after their creation, does not change. Each element can provide set of all its neighbors. Each group starts by random selection of a starting element, which recursively pulls other elements from the detected difference. A group can be
trivial with just one element but it can also include all the elements from detected difference. Elements of extracted group are removed from detected difference and the procedure is recursively repeated until detected difference is not fully exhausted.

Fig. 3 illustrates topology of one detected difference with two isolated groups. Line segments represent conducting lines, squares substations and circles conducting equipment outside substation, which terminates conducting lines. A selected starting element is specified, and numbers represent order of each of its neighbors.

Algorithm for creation of groups with connected elements is given by the following description:

1. Select an arbitrary element from detected difference. It is a new starting element. After selection, remove it from the detected difference.
2. Based on topological connectivity, the search starts for all elements that are starting element’s neighbors and they belong to the detected difference. In other words, red neighbors are searched for. They are neighbors of the 1\textsuperscript{st} order. After the search, remove them from the detected difference.
3. Neighbors of the 1\textsuperscript{st} order are treated as starting elements and step 2 is recursively repeated for them. This gives the neighbors of the 2\textsuperscript{nd} order. Eventually, recursion will stop when neighbors of order \(k\) become an empty set (they either do not exist nor they are added nor they are changed). All neighbors in range \(i=1...k-1\) represent one group of connected elements extracted from the detected difference.
4. The whole procedure starting from step 1 is repeated unless the detected difference is an empty set.

Resulting groups do not depend on the elements that are selected as starting elements. Order of groups does not affect update of PNAS, but on the client’s request, the groups are arranged in such a way that first come all the groups with substations, then all the groups with conducting equipment and at the end the groups with only conducting lines. To provide this order of groups, before selection of the first starting element and after removal of all substation internal elements, the remaining elements are to be arranged according to the following order:

1. added substations
2. changed substations
3. added conducting equipment outside substation
4. changed conducting equipment outside substation
5. added conducting lines
6. changed conducting lines

The starting element is now the first element in the remaining detected difference.

The solution has been implemented for Windows platform, as one DLL component, in C++ programming language. The component is integrated in desktop application developed in PyQt programming language. The overall solution for incremental integration has 42k physical lines of code and 380 classes, out of which 194 classes belong to CIM standard realization.

IV. DISCUSSION OF RESULTS

The solution does not guarantee a full automatic integration because the update of PNAS can fail and the manual intervention is needed to resolve this problem. The most frequent reason for that are changes inside PNAS, which are not synchronized with GIS. On the other hand, the solution is quite general and does not depend on the used PNAS. The fieldwork has shown that the most important thing for a successful resolving of manual interventions is education of a customer and his focus on correct entry of GIS data, because that is how the level of automatization is really controlled. Insisting on correct entry of GIS data does not additionally complicate the existing procedures. This shows that the procedure of incremental integration simply fits into already existing business processes based on a GIS system, which is the central point of the whole software system.

Presented solution is deployed and continuously used by two clients. After setting GIS up to the level that is required by integration, the usage of the application becomes an integral part of GIS maintenance. Thanks to the extensive validation of exported feeders, the application contributed to more systematic and more uniform GIS data entry. Splitting of detected difference into groups helps to resolve remaining problems faster and simplifies troubleshooting.

Tab. 1 presents weekly success rate of the system synchronizations for two client installations within one month. The integration is performed once a day or more frequently if problems appear. Installation 1 is being used for a longer period than Installation 2 and has less frequent and more stable integrations. Installation 2 has around 3 times more feeders than Installation 1, with changes of larger range that include a significant number of feeders, which ultimately give more integration problems. On the sites of both clients, unsuccessful integrations are resolved by entering the corrections in GIS and then repeating the integration.

- S – number of successful integrations
- U – number of unsuccessful integrations
- F – number of changed feeders
Table 1. Success rate of five weekly integrations for two clients

<table>
<thead>
<tr>
<th></th>
<th>Week</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Installation</td>
<td>Feeders</td>
<td>S</td>
<td>U</td>
<td>F</td>
<td>S</td>
</tr>
<tr>
<td>1</td>
<td>500</td>
<td>1</td>
<td>1</td>
<td>27</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1400</td>
<td>3</td>
<td>1</td>
<td>76</td>
<td>8</td>
<td>3</td>
</tr>
</tbody>
</table>

V. CONCLUSION

Efficient integration of GIS and PNAS is a practical problem inside the PS domain. This paper explains the need for a solution of this problem and presents one possible way of solving the problem. Suggested solution does not achieve a full automatic integration, but has proved to be practically useful because it fits into existing business processes and increases the level of their automatization and consequently improves their efficiency and reliability. The solution implements functional requirements by using widely accepted mechanisms for integration development: incremental integration supported by standard defined exchange of data. CIM standard proved itself as a very good specification for internal data model realization, which showed to be the central point for success of the whole application.

A problem of fully automatic integration remains as a challenge for further research and development. A promising solution would be to do an export of only CIM/XML changes of GIS states and avoid the need for full state comparison. In order to make something like this possible, it is necessary to get adequate support from GIS, which needs to know how to generate difference of its two relevant states. Except CIM/XML export format, the solution should also offer other export formats supported by different GIS systems.

REFERENCES