Power System Modeling for a Fully Transparent Distribution Management System

Monika Ruh, Göran Andersson
Swiss Federal Institute of Technology
Zürich, Switzerland
ruh@eeh.ee.ethz.ch, andersson@eeh.ee.ethz.ch

Andreas Borer
Rittmeyer AG
Baar, Switzerland
andreas.borer@rittmeyer.com

Abstract—**Fully integrated distribution management systems (DMSs) not only require a transparent data architecture. They also need efficient procedures connecting base system functions with advanced application functions to ensure data consistency. This paper will present a new algorithm which enables power system modeling for a fully transparent DMS.**

Keywords:* *Distribution management system, distribution grid, power system modeling, transparent data architecture.

1 INTRODUCTION

Due to the advent of distributed generation, the liberalization of power markets and the enaction of new extensive environmental laws, distribution grid operators and municipal utilities are assigned with new complex tasks and have to fulfill more stringent requirements concerning

- Security of supply and reliability
- Power quality
- Economy
- Environmental impact

To meet all these partly diverging requirements, not only economically planned and sufficiently dimensioned distribution grids are essential, but the distribution management system (DMS) with which the distribution grid will be controlled and monitored is almost as important [1].

This paper is organized as follows. Section 2 summarizes the results of a carried out state-of-the-art survey. The necessary conditions for a DMS with a transparent data architecture are described in Section 3. In Section 4 the concept for a fully transparent DMS is presented by introducing the so called transparency matrix. Section 5 describes the principle idea and procedure of the condensing algorithm. The final Section 6 contains the conclusions. A subsequent appendix explains the mathematical background of the presented transparency matrix.

2 STATE-OF-THE-ART OF EXISTING DMSs

To find out how DMSs should be modified and expanded to better suit future conditions, a state-of-the-art survey of existing and on the market available DMSs has been carried out. Especially, well-known on the market available DMSs have been analyzed by product trainings, demos, brochures, fact sheets or booklets.

The major conclusion of this state-of-the-art survey was that there is a need for a fully integrated DMS with a transparent data architecture. State-of-the-art DMSs are almost exclusively custom-made solutions. They require a high degree of project implementation work which could strongly affect the quality of the finished systems. The following tasks are part of the implementation work:

- Linking data objects and graphical elements
- Assigning data to reports
- Connecting substations' process variables and data objects of the control system

This type of work is often error-prone and has to be redone after every small configuration change. Thus, configuration changes are cumbersome and time-consuming.

According to [2], a state-of-the-art DMS having the full functionality of base system functions and applications has the disadvantage that overlaps between advanced application data models occur. Thus, such a DMS requires a data architecture achieving almost seamless integration. However, one problem still remains: If there is a configuration change, every single advanced application data model has to be updated accordingly.

3 DMS WITH A FULLY TRANSPARENT DATA ARCHITECTURE

In order to avoid the shortcomings mentioned in the previous section, a project with the aim to develop a future DMS with a fully transparent data architecture was started. Such a distribution control system with a complete transparent data structure allows the adaption to the specific costumer installation with minimum effort and simultaneously guaranteeing maximum consistency. Thus, errors done by project implementation work or by commissioning can be avoided as far as possible.

Basis of a transparent system architecture has to be a standardized and object-oriented design. Moreover, there has to be one unique data file or one unique data object list in which all data objects of the control system can be stored. As pictured in Fig. 1, not only a data model belongs to a data object, but also

- Visualization elements
- Operating elements
- Alarm elements

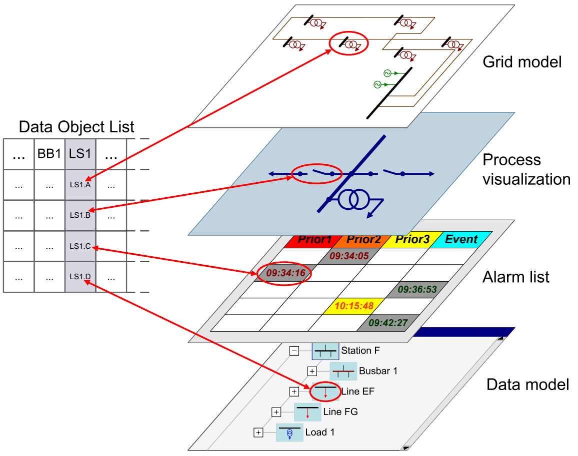


Fig. 1. Data architecture of a transparent distribution management system.

- Log and archiving elements
- Grid model

The grid model is very important because it establishes the basis of all kind of application-oriented algorithms. Based on the grid model, advanced applications like power flow calculations, short-circuit calculations and outage management functions can be executed.

4 THE TRANSPARENCY MATRIX

In this section the so called *transparency matrix* will be elaborated. This matrix plays a crucial role in the DMS that will overcome the shortcomings discussed above.

The importance of the grid model gave the authors the idea of introducing the so called transparency matrix: All data objects of the control system are stored in a matrix by placing one data object in each column. In every element of the column, one attribute of the data object is stored. To unambiguously address all data objects and their attributes, the first column comprises the name of the attributes and the first row comprises the identifier of the data objects as pictured in Fig. 2.

4.1 Structure of the Transparency Matrix

To describe the structure of the transparency matrix in detail, the distribution grid pictured in Fig. 3 is taken as an example: It contains the medium voltage (MV) source substation and six distribution substations connected by seven lines. Each distribution substation holds a distribution transformer connected with loads on the low voltage level. The transparency matrix mapped in Fig. 2 stores all data objects of the MV source substation and the three distribution substations laying within the (dark-red) dashed frame of Fig. 3.

The transparency matrix mapped in Fig. 2 shows three typical attribute rows. There is a graphical data row, in which the graphical coordinates of a specific view can be stored. A specific attribute concerning the data model is the data type of the data object. This attribute is stored in the type data row. The graphical data attribute and the type data attribute are static attributes. Besides these static attributes there exist more dynamic attributes like for instance the state data attribute which is gathered by the basic SCADA system and then stored in the state data row.

The most important part of the transparency matrix is the block matrix in the upper part highlighted with a (orange and red) dashed frame: This block matrix is a special incidence matrix, which has to represent the grid model or rather the grid topology such that it can later be used for load flow or other topological calculations. In its rows the incidence matrix contains the nodes of the distribution network. Therefore, busbars, which are the real nodes of the power grid, are presented twice in the transparency matrix:

- Once as a data object with all its attributes stored in a transparency matrix column
- and once as a grid node in a row of the incidence matrix.

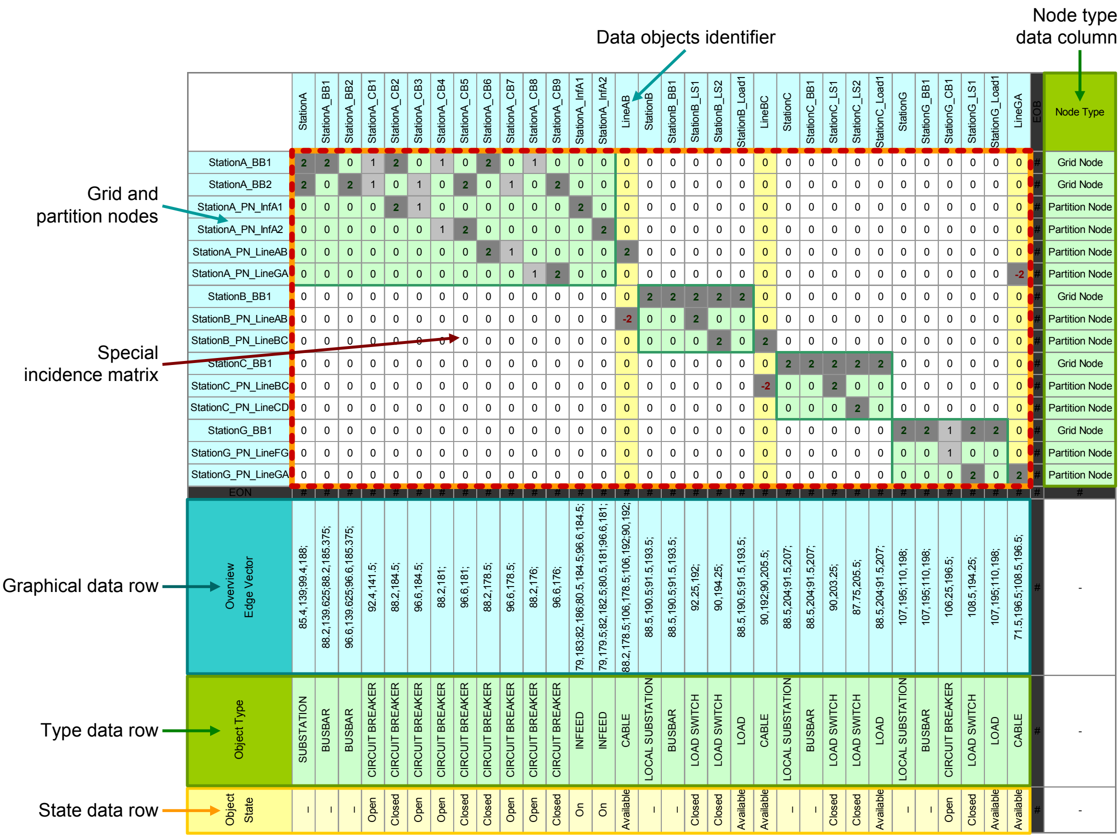
This could be verified by inspection of the transparency matrix shown in Fig. 2. Besides the grid nodes, the rows of the incidence matrix contain so called partition nodes whose mathematical derivation will be explained in Appendix A.

The incidence matrix presented here is characterized mainly by three special properties:

1. *Incidence orientation:*
   The incidence matrix is a partly unoriented and partly oriented incidence matrix: There are directed branches as well as undirected branches. A directed branch is an electrically long connection whose physical model is given by the line equation. Hence, transmission lines and cables are represented as directed branches in the incidence matrix. Otherwise, an undirected branch is an electrically short connection between two nodes. Consequently, switchable and coupling devices are represented as undirected branches in the incidence matrix.
2. *Connectivity integers:*
   The incidence matrix uses not only one integer and its additive inverse to express connectivity between different data objects. As can be seen in Fig. 2, the integer 2 and its additive inverse -2 are used besides the integer 1. The reason for that will be explained in Subsection 4.2.
3. *Half branches:*
   Since the transparency matrix has to store also data objects which do not have two-terminal characteristic, half edges[1] or better half branches are needed. Such a half branch is connected to only one node, why in the column storing it only one integer expressing connectivity appears. Data

---

[1] Half edges are used in graph theory when it is necessary to have edges with only one end. The definition and concept of half-edges is for instance described in [3].



	StationA	StationA_BB1	StationA_BB2	StationA_CB1	StationA_CB2	StationA_CB3	StationA_CB4	StationA_CB5	StationA_CB6	StationA_CB7	StationA_CB8	StationA_CB9	StationA_InfA1	StationA_InfA2	LineAB	StationB	StationB_BB1	StationB_LS1	StationB_LS2	StationB_Load1	LineBC	StationC	StationC_BB1	StationC_LS1	StationC_LS2	StationC_Load1	StationG	StationG_BB1	StationG_CB1	StationG_LS1	StationG_Load1	LineGA	EOB	Node Type
StationA_BB1	2	2	0	1	2	0	1	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	#	Grid Node
StationA_BB2	2	0	2	1	0	1	0	2	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	#	Grid Node
StationA_PN_InfA1	0	0	0	0	2	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	#	Partition Node
StationA_PN_InfA2	0	0	0	0	0	0	1	2	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	#	Partition Node
StationA_PN_LineAB	0	0	0	0	0	0	0	0	2	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	#	Partition Node
StationA_PN_LineGA	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2	#	Partition Node
StationB_BB1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	#	Grid Node
StationB_PN_LineAB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	#	Partition Node
StationB_PN_LineBC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2	0	0	0	0	0	0	0	0	0	0	0	#	Partition Node
StationC_BB1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	0	0	0	0	0	0	#	Grid Node
StationC_PN_LineBC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2	0	0	2	0	0	0	0	0	0	0	0	#	Partition Node
StationC_PN_LineCD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	#	Partition Node
StationG_BB1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	1	2	2	0	#	Grid Node
StationG_PN_LineFG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	#	Partition Node
StationG_PN_LineGA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2	#	Partition Node
EON	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
Overview Edge Vector	85.4,139;99.4,188;	88.2,139.625;88.2,185.375;	96.6,139.625;96.6,185.375;	92.4,141.5;	88.2,184.5;	96.6,184.5;	88.2,181;	96.6,181;	88.2,178.5;	96.6,178.5;	88.2,176;	96.6,176;	79,183;82,186;80.5,184.5;96.6,184.5;	79,179.5;82,182.5;80.5,181;96.6,181;	88.2,178.5;106,178.5;106,192;90,192;	88.5,190.5;91.5,193.5;	88.5,190.5;91.5,193.5;	92.25,192;	90,194.25;	88.5,190.5;91.5,193.5;	90,192;90,205.5;	88.5,204;91.5,207;	88.5,204;91.5,207;	90,203.25;	87.75,205.5;	88.5,204;91.5,207;	107,195;110,198;	107,195;110,198;	106.25,196.5;	108.5,194.25;	107,195;110,198;	71.5,196.5;108.5,196.5;	#	-
Object Type	SUBSTATION	BUSBAR	BUSBAR	CIRCUIT BREAKER	CIRCUIT BREAKER	CIRCUIT BREAKER	CIRCUIT BREAKER	CIRCUIT BREAKER	CIRCUIT BREAKER	CIRCUIT BREAKER	CIRCUIT BREAKER	CIRCUIT BREAKER	INFEED	INFEED	CABLE	LOCAL SUBSTATION	BUSBAR	LOAD SWITCH	LOAD SWITCH	LOAD	CABLE	LOCAL SUBSTATION	BUSBAR	LOAD SWITCH	LOAD SWITCH	LOAD	LOCAL SUBSTATION	BUSBAR	CIRCUIT BREAKER	LOAD SWITCH	LOAD	CABLE	#	-
Object State	–	–	–	Open	Closed	Open	Open	Closed	Closed	Open	Open	Closed	On	On	Available	–	–	Closed	Closed	Available	Available	–	–	Closed	Closed	Available	–	–	Open	Closed	Available	Available	#	-

Fig. 2. Transparency matrix of part of the test distribution grid which is surrounded with a dark-red dashed frame.

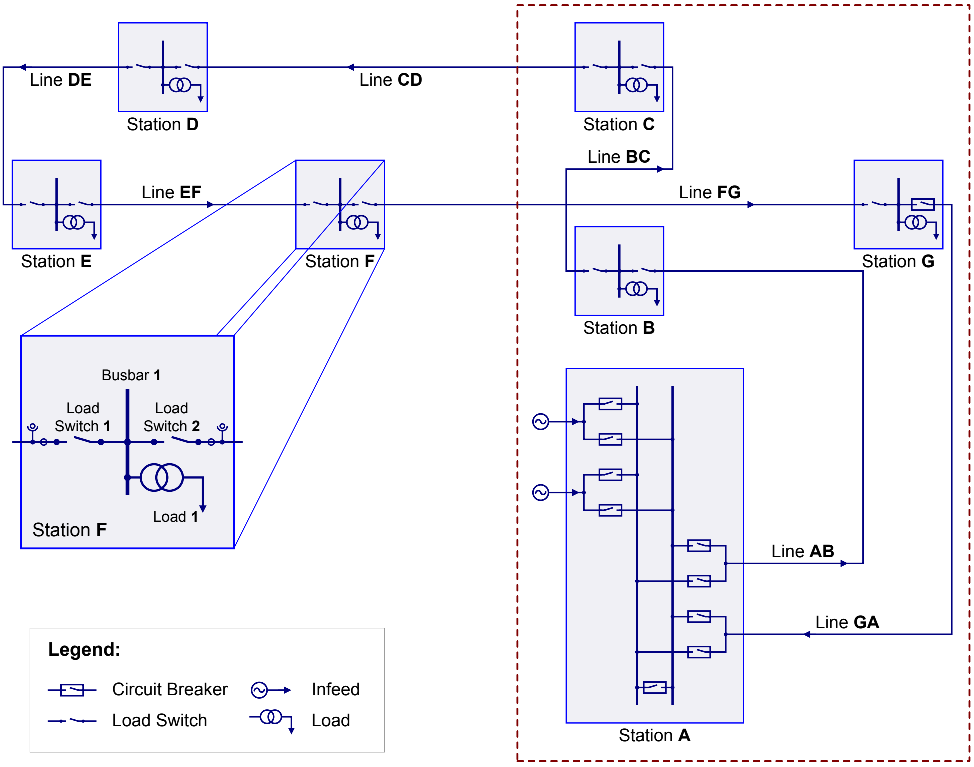


Fig. 3. The test distribution grid with seven substations and seven lines.

objects like busbars, infeeds or also virtual links to meta data objects, e.g. substations, are stored as half undirected branches in the transparency matrix.

4.2 Managing Topologies with the Transparency Matrix

Each interconnected power grid has a limited but large number of possible connections. The collectivity of all possible connections is referred to as *possible topology*. In a given operation state, only a subset of the possible connections are in use since the usage of *all* possible connections would not be reasonable. The topology resulting of the actually used connections at a particular time is the so called *actual topology*. In distribution grids, one goal of power system management and control is to maintain a certain grid topology over longer periods of time. Only during disturbances or maintenance work another topology is in use and directly after their correction or their completion the switchable connections are switched back to the specific grid topology, the so called *regular topology*.

Since the DMS has to manage all these three topologies [4], the transparency matrix has to express which node and which branch are incident, but also if their incidence is actual, regular or just possible. The possible and regular topology can be stored in the incidence matrix by using integers 0, 1 and 2 for undirected branches and integers -2, -1, 0, 1 and 2 for directed branches.

Undirected branches:
The value of an unoriented incidence matrix element $[i, j]$ is defined as follows:

0: node i and branch j are **never** incident.

1: node i and branch j are incident in the *possible topology*:

Branch j is a possible connection with node i.

2: node i and branch j are incident in the *possible* and *regular topology*:

Branch j is a possible and in regular situations used connection with node i.

Directed branches:
The value of an oriented incidence matrix element $[i, j]$ is defined as follows:

0: node i and branch j are **never** incident.

-1: node i and branch j are incident in the *possible topology*:

Branch j is a possible connection entering node i.

1: node i and branch j are incident in the *possible topology*:

Branch j is a possible connection leaving node i.

-2: node i and branch j are incident in the *possible* and *regular topology*:

Branch j is a possible and in regular situations used connection entering node i.

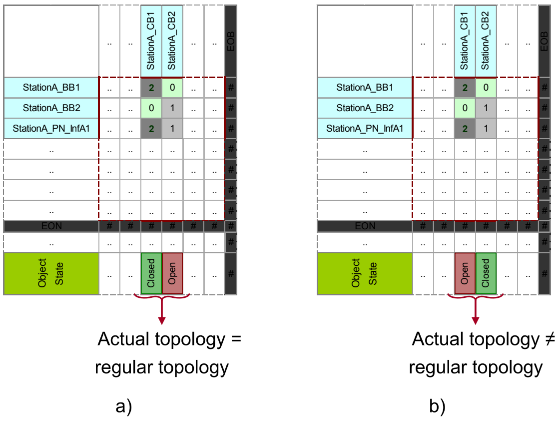


Fig. 4. a) The actual topology is the same as the regular topology due to the switch states. b) The actual topology differs from the regular topology due to the actual switch states.

2: node i and branch j are incident in the *possible* and *regular topology*:

Branch j is a possible and in regular situations used connection leaving node i.

For the determination of the actual topology, the DMS has to consider all switch states. It is assumed that the switch states are stored and updated in the state data row of the transparency matrix. Hence, the topology determination algorithm needs to check the state data row as shown in Fig. 4: The state of a specific switchable device has to be checked in case its data object column contains a **1** or **2** in the incidence matrix.

For example, the switch state **Closed** together with a **2** in the incidence matrix means that the switch is in its *regular state* like sketched in Fig. 4 a).

On the other hand, the switch state **Open** together with a **2** in the incidence matrix means that the switch is not in its *regular state*. As a consequence, the *actual topology* of the distribution grid differs from its *regular topology*. This situation is pictured in Fig. 4 b).

4.3 Consistency and Filterability of the Transparency Matrix

The main benefit of the transparency matrix is that it fulfills the six transparency criteria stated in [5]. Particularly concerning efficient procedures for tight coupling between base system functions and application functions, the fulfillment of the criteria *consistency* and *filterability* is most important.

Consistency:

The transparency matrix forces data consistency: Each program or view uploads its needed information from the transparency matrix. Changes generated in one program are saved to the transparency matrix. Hence, all other programs or views are updated automatically when uploading their required information from the transparency matrix.

Filterability:

The transparency matrix is easily filterable:

Columns of not needed data objects or rows of not needed attributes can just be ignored.

Due to the consistency and filterability of the transparency matrix it is possible to adapt the level or rather the accuracy of the grid model to the needs of the specific application function. Whereas many of the base system functions, e.g. SCADA functions, require a grid model with the highest device resolution, many of the advanced application functions need only a simplified equivalency of the grid model [2]. The most complex grid model with the highest device resolution has to contain all grid components and grid elements of the distribution grid. Hence, the transparency matrix storing all data objects of the grid, contains this most complex grid model, namely in the form of the incidence matrix. Using the filterability of the transparency matrix, efficient simplifying algorithms or filtering algorithms can be developed which reduce the complex grid model in form of the incidence matrix to a simplified grid model containing only the needed information. Each such algorithm guarantees that data it provides to its associated applications or functions is only coming from the transparency matrix being the unique data object storage.

4.4 Transparency Matrix and Standard Data Models

The transparency matrix is a conceptual model for the *internal* data architecture of a fully transparent DMS fulfilling the six transparency criteria stated in [5]. However, for *external* interaction, communication or data exchange with IT-systems of other utilities it is essential that the transparency matrix can be combined with industrial standard data models.

Such an industrial standard data model representing all main components of an electric power system is the Common Information Model (CIM) [6] standardized within IEC 61970. Since CIM and the transparency matrix have both an object-oriented data structure, the data objects stored in the columns of the transparency matrix can be standardized CIM-objects. The grid topology modelled in CIM can be transformed into the incidence matrix of the transparency matrix. Each attribute described in CIM is stored in a row of the transparency matrix.

In principle, each industrial standard that has an object-oriented data structure relating to electric power systems can be stored in the transparency matrix: The data objects stored in the columns of the transparency matrix are complying with the concerning standard and the rows of the transparency matrix contain the associated attributes. Attributes not belonging to the standard but crucial for fulfilling the transparency concept have to be stored in additional rows like for instance the graphical data rows.

4.5 Implementation of the Transparency Matrix

Even though the transparency matrix mapped in Fig. 2 just stores all data objects of a MV source substation and three distribution substations, it is already a 20×35-matrix. Consequently, for a real distribution grid with many distribution substations and several MV substations, the transparency matrix gets large dimensions. That's why the presented transparency matrix is more a conceptual model of a fully transparent data architecture than a real software solution to store object data. For the final software implementation, the transparency matrix can be mapped on a data base. Still, the transparency matrix is available: It can be generated out of the data base into the random access memory (RAM) of the DMS, where it can be used for instance for transparency checks after configuration changes or as a basis for application-oriented algorithms.

5 THE CONDENSING ALGORITHM

5.1 Motivation for the Condensing Algorithm

Many application-oriented algorithms, like power flow calculations or short circuit calculations, need as grid model a nodal admittance matrix representing the actual connections between lines and busbars in operation. The filterability of the transparency matrix described in Subsection 4.3 allows the derivation of such a nodal admittance matrix by filtering, or rather condensing, the transparency matrix into a so called *condensed incidence matrix*. This derived condensed incidence matrix represents the actual topology of the distribution grid in the desired concentrated form: Only lines and busbars in operation are considered. Finally, this derived condensed incidence matrix can be transformed into the desired nodal admittance matrix.

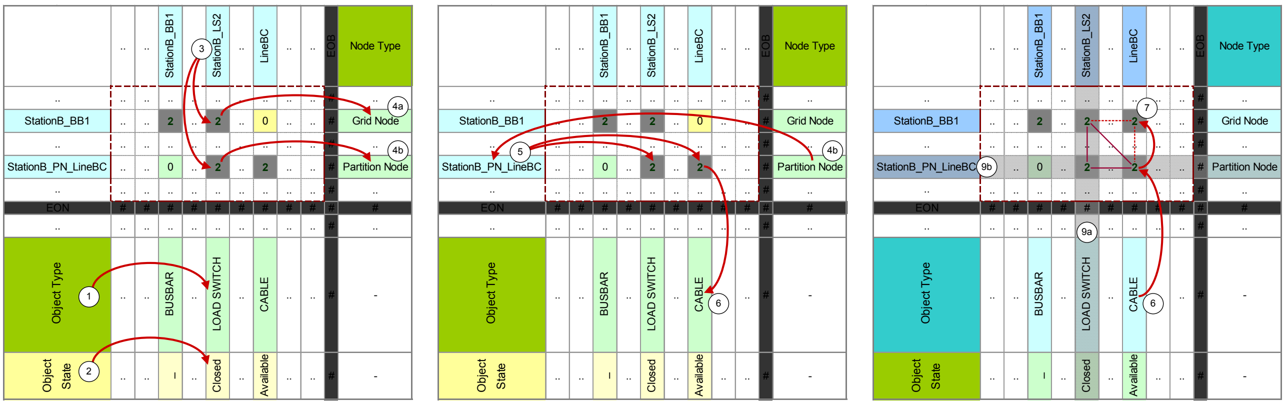
5.2 The Principle Idea of the Condensing Algorithm

The principle idea of the condensing algorithm is to establish for every closed switch a virtual connection between its adjacent grid branch, e.g. transmission line or cable, and its adjacent grid node, e.g. busbar. In the derived transparency matrix or rather in the derived incidence matrix this virtual link is achieved by replacing a zero integer with a connectivity integer expressing incidence between the relevant grid node and the relevant grid branch. Hence, the closed switch's connection between the grid branch and the grid node becomes a redundant connection, which can be filtered out.

5.3 Procedure of the Condensing Algorithm

Before explaining the procedure of the condensing algorithm in detail, three different cases have to be distinguished:

1. *Closed single switch:*
   This case is the standard one: A closed switch makes *one* short electrical connection between its adjacent grid branch and its adjacent grid node. This is also the case for a closed switch belonging to a switch pair of a double busbar, if its partner switch is opened.



a)

b)

c)

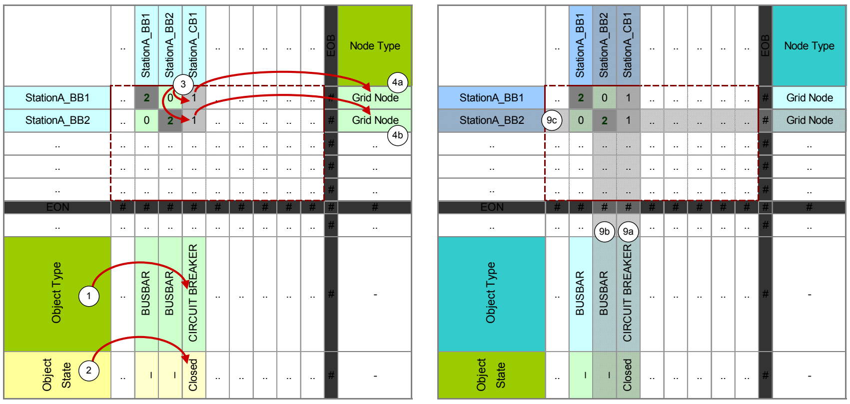
Fig. 5. Procedure of the condensing algorithm for the case of the *closed single switch*: a) Steps ① to ④ of the algorithm. b) Steps ④ to ⑥ of the algorithm. c) Steps ⑥, ⑦ and ⑨ of the algorithm.

a)

b)

c)

Fig. 6. Procedure of the condensing algorithm for the case of the *closed pair switch*: a) Steps ① to ④ of the algorithm. b) Steps ④ to ⑥ of the algorithm. c) Steps ⑥ to ⑨ of the algorithm.



a)

b)

Fig. 7. Procedure of the condensing algorithm for the case of the *closed busbar coupler*: a) Steps ① to ④ of the algorithm. b) Step ⑨ of the algorithm.

2. *Closed pair switch:*
   This case can only appear at double busbar infeeds: If both switches of the switch pair are closed, then *two* short electrical connections exist: One between the adjacent grid branch and the first busbar and one between the adjacent grid branch and the second busbar. Hence, the two busbars are electrically connected through both closed switches. As a consequence, both busbars have the same voltage. They can be condensed to one busbar in the nodal admittance matrix.

A special case appears if a busbar coupler at a double busbar station is closed:

3. *Closed busbar coupler:*
   The closed busbar coupler makes *one* short electrical connection between its adjacent busbars. Consequently, the two busbars have the same voltage. They can be condensed to one busbar in the nodal admittance matrix.

The procedure of the algorithm is slightly different for the three cases. Fig. 5 pictures the procedure for the case of the *closed single switch*. In Fig. 6 the procedure for the case of the *closed pair switch* is shown and in Fig. 7 the procedure for the case of the *closed busbar coupler* is sketched.

The algorithm starts with making a copy of the transparency matrix. This duplicate of the transparency matrix is called *derived transparency matrix* and will be used for the construction of the condensed transparency matrix. For each data object column of the transparency matrix the algorithm has to execute the following instructions:

① Check if the object type of the data object stored in the actual column i of the transparency matrix is a switch:
   - If yes, go to ②.
   - If no, go to the next data column.

② Check if the actual object state of the switch is closed:
   - If yes, go to ③.
   - If no, go to ⑨.

③ Examine all incidence matrix elements of the actual column. Check if the integers stored in the incidence matrix elements are connectivity integers:
   - If yes, store the number of the actual row and note the node type of the node stored in this row of the incidence matrix.
   - Then go to ④.

④ Check if one of the noted nodes is a partition node:
   - If yes, go to ⑤.
   - If no, go to ⑨.

⑤ Examine all incidence matrix elements of the row in which the found partition node is stored. Check if the integers stored in the incidence matrix elements of this row are connectivity integers:
   - If yes, store the number of the actual column and note the data type of the data object stored in this column of the transparency matrix.
   - Then go to ⑥.

⑥ Check if one of the found adjacent data objects is a cable or another switch:
   - If it is a cable, go to ⑦.
   - If it is a second closed switch, go to ⑧.
   - Otherwise go to ⑨.

⑦ Take now the derived transparency matrix. Go to its matrix element $[k, l]$ being located in the column l in which the cable is stored and in the row k in which the partition node is stored. Copy the connectivity integer stored in this matrix element to the matrix element $[j, l]$ laying in the same column l but in the row j in which the grid node is stored. Doing so, the triangle of the matrix elements $[j, i]$, $[k, i]$ and $[k, l]$ expressing connectivity is enhanced with the matrix element $[j, l]$ to a square with connectivity integers, like can be see by inspection of Fig. 5 c) and Fig. 6 c).
   - Then go to ⑨.

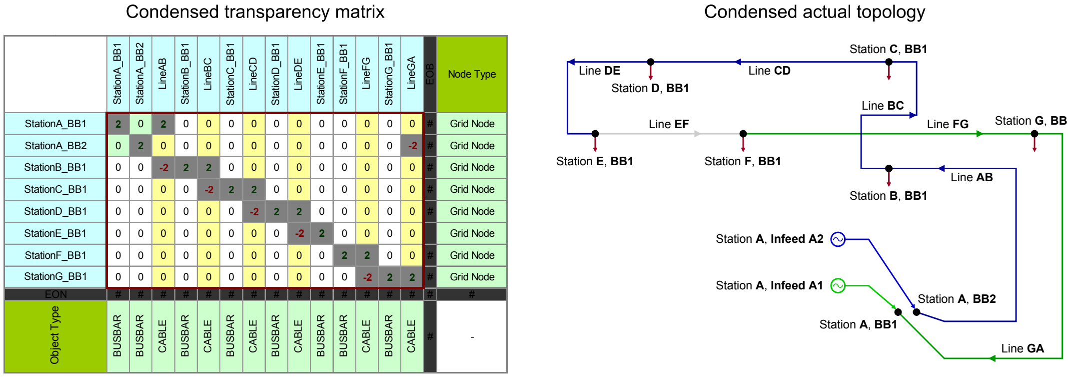
⑧ Check if the found other switch is connected to a grid node:
   - If yes, store the number of the actual row.
   - Then go to ⑨.

⑨ Mark the derived transparency matrix column in which the actual switch is stored for deleting.
   - If a partition node has been found in step ③, mark the concerning row for deleting.
   - If a second closed switch has been found in step ⑥, mark the concerning column for deleting.
   - If a second grid node has been found in step ④or in step ⑧, copy all connectivity integers appearing in one busbar's row to the corresponding matrix elements in the other busbar's row and vice versa. Then, mark the second busbar for deleting.
   - Then, go to the next column of the transparency matrix.

After all data columns of the transparency matrix have been examined, the rows and columns of the derived transparency matrix which are marked for deleting can be eliminated. The derived transparency matrix is then renamed *condensed transparency matrix*.

5.4 The Resulting Condensed Transparency Matrix

The described condensing algorithm produces as result the condensed transparency matrix containing only grid components, that means grid nodes, e.g. busbars, and grid branches, e.g. lines. Consequently, the number of columns of the condensed incidence matrix corresponds to the sum of busbars and lines in operation. Similarly, the number of rows of the condensed incidence matrix corresponds to the number of busbars in operation. Using the test distribution grid introduced in



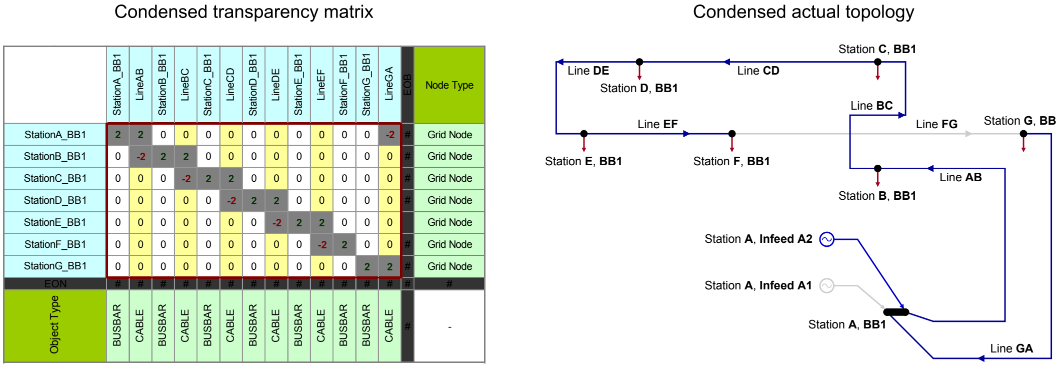
Condensed transparency matrix

	StationA_BB1	StationA_BB2	LineAB	StationB_BB1	LineBC	StationC_BB1	LineCD	StationD_BB1	LineDE	StationE_BB1	StationF_BB1	LineFG	StationG_BB1	LineGA	EOB	Node Type
StationA_BB1	2	0	2	0	0	0	0	0	0	0	0	0	0	0	#	Grid Node
StationA_BB2	0	2	0	0	0	0	0	0	0	0	0	0	0	-2	#	Grid Node
StationB_BB1	0	0	-2	2	2	0	0	0	0	0	0	0	0	0	#	Grid Node
StationC_BB1	0	0	0	0	-2	2	2	0	0	0	0	0	0	0	#	Grid Node
StationD_BB1	0	0	0	0	0	0	-2	2	2	0	0	0	0	0	#	Grid Node
StationE_BB1	0	0	0	0	0	0	0	0	-2	2	0	0	0	0		Grid Node
StationF_BB1	0	0	0	0	0	0	0	0	0	0	2	2	0	0	#	Grid Node
StationG_BB1	0	0	0	0	0	0	0	0	0	0	0	-2	2	2	#	Grid Node
EON	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
Object Type	BUSBAR	BUSBAR	CABLE	BUSBAR	CABLE	BUSBAR	CABLE	BUSBAR	CABLE	BUSBAR	BUSBAR	CABLE	BUSBAR	CABLE	#	-

Condensed actual topology

a) b)

Fig. 8. Case of *decoupled busbars* in station A: a) The condensed transparency matrix. b) The condensed actual topology.



Condensed transparency matrix

	StationA_BB1	LineAB	StationB_BB1	LineBC	StationC_BB1	LineCD	StationD_BB1	LineDE	StationE_BB1	LineEF	StationF_BB1	StationG_BB1	LineGA	EOB	Node Type
StationA_BB1	2	2	0	0	0	0	0	0	0	0	0	0	-2	#	Grid Node
StationB_BB1	0	-2	2	2	0	0	0	0	0	0	0	0	0	#	Grid Node
StationC_BB1	0	0	0	-2	2	2	0	0	0	0	0	0	0	#	Grid Node
StationD_BB1	0	0	0	0	0	-2	2	2	0	0	0	0	0	#	Grid Node
StationE_BB1	0	0	0	0	0	0	0	-2	2	2	0	0	0		Grid Node
StationF_BB1	0	0	0	0	0	0	0	0	0	-2	2	0	0	#	Grid Node
StationG_BB1	0	0	0	0	0	0	0	0	0	0	0	2	2	#	Grid Node
EON	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
Object Type	BUSBAR	CABLE	BUSBAR	CABLE	BUSBAR	CABLE	BUSBAR	CABLE	BUSBAR	CABLE	BUSBAR	BUSBAR	CABLE	#	-

Condensed actual topology

a) b)

Fig. 9. Case of *coupled busbars* in station A: a) The condensed transparency matrix. b) The condensed actual topology.

Section 4, the condensing algorithm is verified for two cases with basically different actual grid topologies:

1. *Decoupled double busbar:*
   For an actual grid topology with decoupled busbars in station A, the condensed actual topology and the condensed transparency matrix are pictured in Fig. 8.
2. *Closed pair switch:*
   For an actual grid topology with coupled busbars in station A, the condensed actual topology and the condensed transparency matrix are shown in Fig. 9. Due to the busbar coupling, the second busbar is eliminated and does not appear in the condensed transparency matrix.

For power flow calculations, the condensed incidence matrix can be transformed to the corresponding adjacency matrix. Finally, this adjacency matrix and the series admittances of all lines can be taken to compute the desired nodal admittance matrix.

5.5 The Benefit of the Condensing Algorithm

The benefit of the condensing algorithm is that it establishes a direct and efficient link between SCADA functions using the complex grid model and advanced applications requiring the nodal admittance matrix. Since the complex grid model is stored in the transparency matrix, the simplicity of the condensing algorithm proofs also the benefits of the transparency matrix: All needed information for the nodal admittance matrix is just filtered out of the transparency matrix. Hence, data consistency for advanced application functions is ensured.

6 CONCLUSIONS

In this paper, a condensing algorithm which establishes a link between base system functions and advanced application functions has been presented. Basis of the condensing algorithm is the transparency matrix. The simplicity of the condensing algorithm proofs that the transparency matrix constitutes an optimal basis for application-oriented algorithms. In fact, without an efficient condensing algorithm linking data of the transparency matrix to the nodal admittance matrix of the grid, the transparency concept is much less useful. So, the transparency matrix and the condensing algorithm are two strong partners needed for a fully transparent DMS providing advanced application functions.

Acknowledgment

The authors want to thank Bernhard Müller, Christoph Bücheler, Mathias Hartmann and Gerhard Nigg of Rittmeyer AG for valuable and stimulating discussions.

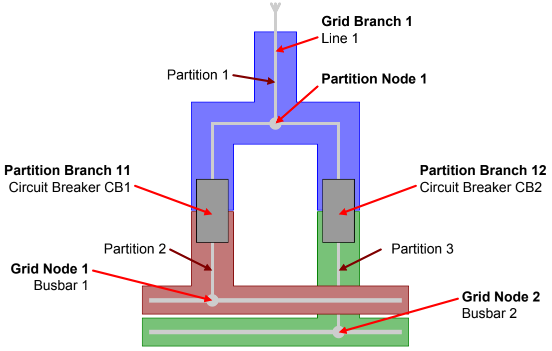


Fig. 10. Partition nodes and branches of a double busbar with an infeed with two circuit breakers.

The financial support of the project by Rittmeyer AG is gratefully acknowledged.

REFERENCES

[1] Ernst-Günther Tietze: *Netzleittechnik Teil 1: Grundlagen*, 2. Auflage, VDE-Verlag GmbH, Berlin, 2006.
[2] James Northcote-Green, Robert Wilson: *Control and Automation of Electrical Power Distribution Systems*, Taylor & Francis Group, Boca Raton, 2007.
[3] Daniel C. Slilaty: *Bias matroids with unique graphical representations*, Department of Mathematics and Statistics, Wright State University, Dayton OH, 2006.
[4] Dieter Rumpel, Ji Rong Sun: *Netzleittechnik*, Springer-Verlag, Berlin, Heidelberg, 1989.
[5] Monika Ruh, Göran Andersson, Andreas Borer: *A New Concept for a Fully Transparent Distribution Management System*, in Proc. of IEEE PowerTech Conference, Lausanne, 2007.
[6] David Becker: *Common Information Model (CIM): CIM 10 Version*, EPRI, Palo Alto, 2001.

APPENDIX A
PARTITIONS BRANCHES AND NODES

Grid branches and grid nodes[2] are almost never fix connected to each other: As shown in Fig. 10, grid nodes, e.g. busbars, and grid branches, e.g. lines, are connected or disconnected over switching devices, e.g. circuit breakers. The actual state of such a switchable device defines which grid node is connected to which line. Hence, there is a need for introducing a sort of branches and nodes between the grid nodes and grid branches so that all possible switch states, that means all states in which the power grid can be found, can be mapped on the resulting mathematical graph structure.

From a mathematical point of view, switching devices are like *active* branches. That's why for every switching device of the power grid a so called *partition branch* has to be introduced. Since branches are always connected by nodes, so called *partition nodes* have to be introduced between partition branches adjacent to a grid branch or between two adjacent partition branches. Fig. 11 shows the mathematical graph structure of a part of the test distribution grid resulting when the partition branches and the partition nodes are introduced. This view of the power grid is called the mathematical view.

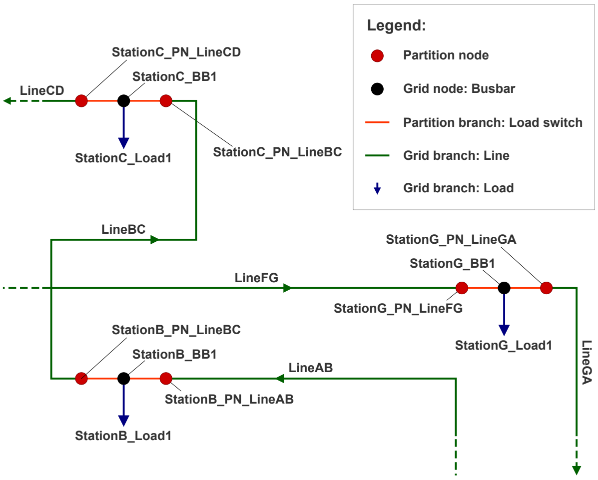


Fig. 11. Mathematical view of three local stations of the test distribution grid. Besides grid nodes and grid branches, this view contains partition nodes and partition branches.

Summarized, the following properties are valid for partition branches respectively partition nodes:

Partition Branches:

- Are connectable branches (mathematical view)
- Are real "active" grid elements or rather switching devices like
  - Circuit breakers
  - Load switches
- Belong to two different partitions
- Can have a voltage across themselves

Partitions Nodes:

- Are nodes connecting partition branches (mathematical view)
- Are real junctions between switching devices
- Are neither grid components nor grid elements
- Belong to just one partition
- Have one electric potential

As already shown in Fig. 2, the introduced partition nodes appear in the incidence matrix like normal grid nodes. The only difference is that a partition node only appears in a row and doesn't appear in a column of the transparency matrix. The cause therefor is that a partition node is just a mathematical object and not a real existing data object.

The reason why these introduced branches and nodes are named partition branches respectively partition nodes becomes obvious when studying Fig. 10: For a double busbar with an infeed, the different partitions are highlighted with colors. Partitions are location and voltage dependent parts of grid components like explained in [1]. Together with the charted partition nodes and partition branches it becomes evident that different partitions are separated by partition branches and that every partition contains one node. Therefore, the given names partition branches and partition nodes are reasonable.

---

[2] For a detailed definition of grid branches, grid nodes, partitions and grid elements the reader is referred to [1]