

## INTEGRATED ENVIRONMENT FOR DATA MANAGEMENT IN EARTH SURFACE POTENTIALS MEASUREMENTS OF A GROUNDING SYSTEM

*Hrvoje Džapo, Roberto Giannini*

Faculty of Electrical Engineering and Computing, Department of Electronic Systems and Information Processing, Zagreb, CROATIA

**Abstract** – Process of validating substation grounding system includes extensive measurements of earth surface potentials and complex measuring results analysis. In order to automate and improve validation process, a customized automated measuring system needed to be developed. It consists of an intelligent measuring device and integrated program environment for measurement data management and analysis on personal computer. This paper describes measurement methodology concepts along with implementation issues of an integrated program environment as a part of such a system. Measuring project concept has been elaborated in details as a foundation for building an integrated program environment which conforms to methodology concepts. Implementation details of some specific program modules are also presented in order to clarify and extend methodology concepts.

**Keywords:** Earth Surface Potentials, Automated Instrumentation, Object-Oriented Programming

### 1. INTRODUCTION

Standard safety check procedures of power installation systems include validation of a grounding network system. Performing such checking periodically is necessary to verify whether grounding system characteristics have been degraded over a longer periods of time. Properties of a grounding system are tested by conducting extensive measurements of earth surface potentials under test conditions [1-2]. The goal is to ensure that under fault conditions touch and step voltages do not exceed permissible values.

Measuring method is divided into two parts: surface *potential map determination* and measuring of *touch and step* voltages. The first method consists of surface potential measurements with respect to remote earth. Analysis of measuring results yields potential distribution  $\varphi(x,y)$  from which a gradient in every point can be easily derived. Such an information is useful for effective planning of a second step, touch and step voltage measurements, taking into account only areas with high surface potential gradient values. Surface potential gradient has a defined vector oriented in direction of maximum potential change for every point  $(x,y)$ , enabling easy determination of correct

orientation during step voltage measurements, perpendicular to the direction of equipotential lines.

Surface potentials measurements start by determining the position and measuring of a *Grounding Potential Rise (GPR)* with respect to remote earth, latter used as a local reference point. Measured potentials are recalculated during analysis phase to correct values with respect to remote earth. For accurate determination of potential distribution map  $\varphi(x,y)$ , measurements need to be performed over rectangular grid requiring all measuring positions to be planned in advance. This includes a great number of measuring points, from few hundreds to few thousands points typically, depending on the measuring area size. Visualization of measuring results analysis yields equipotential lines distribution and this step is referred to as *Equipotential* part of a measurement process in a workflow model shown in Fig. 1.

The complexity of measurement workflow model and the amount of acquired measuring information imply the need for an elaborated methodology for project definition, preparation of measuring documentation, measuring plan

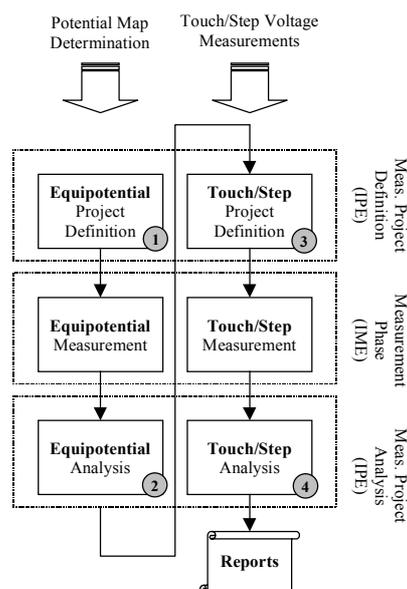


Fig. 1: Typical earth surface potentials measurement project workflow

realization, automation of measuring results analysis, generating reports and archiving information in database-like system. Space relationships are important in such a system, because the set of measuring points needs to conform to real world considerations in project definition phase. In the analysis phase with stated means, equipotential analysis results can be connected with touch and step measuring project plan. Existing database systems do not offer enough flexibility to accommodate all needs for automation of measurement process, especially in the part of specific interactive data entry and visualization of post-measurement analysis, as well as in measuring device hardware control capabilities. On the other side, off-the-shelf *Computer-Aided Design (CAD)* applications are designed for completely different purposes and cannot be easily customized for highly specific needs of a presented workflow model.

Considering facts stated above, the only practical way to accommodate all specific demands essential for automation of earth surface potentials measurement process was to develop highly customized measuring system. This system includes both *Integrated Program Environment (IPE)* based on personal computer and compatible microcontroller-based *Intelligent Measuring Equipment (IME)* [3]. Integrated program environment unifies, among others, the functions of database system, CAD interface, numerical analysis and visualization software and supervisory module for intelligent measuring equipment.

Pieces of information are organized by grouping them together into *Measuring Projects*. Each one contains all relevant data connected with single power installation grounding system along with project definitions for both *Equipotential* and *Touch/Step* parts of measurement.

## 2. MEASURING PROJECT CONCEPT

Measuring project is divided into two basic parts: *Template* and *Measurement* results (as denoted in Fig. 2). Project *Template* defines the plan of how measurements should be performed. *Measurement* part stores measuring results retrieved from *IME* and serves for post-measurement analysis and visualization. Both parts contents can be represented to users either graphically (*View*) or tabular (*Table*).

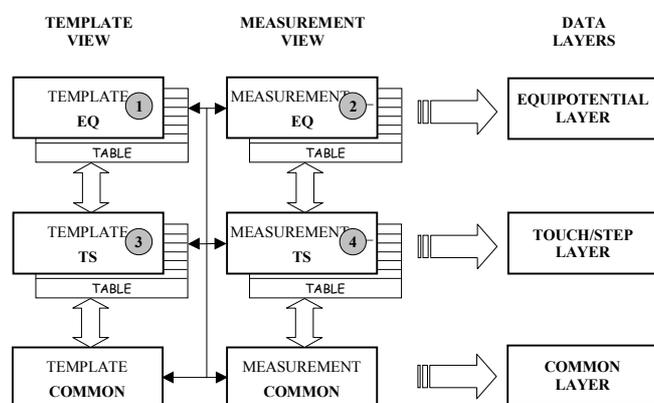


Fig. 2: Measuring Project structure

Measuring project parts are internally divided into three information subsections: *Common*, *Equipotential* and *Touch/Step Layers*. *Common Layer* contains general information about power installation system (e.g. positions of all buildings) and set of auxiliary objects (measuring area limits, map orientation, miscellaneous reference points etc.). *Equipotential* and *Touch/Step Layers* contain information about measuring task definitions (as the part of *Template*) and measuring results (as the part of *Measurement*) of surface potential map or touch and step measurements, respectively. Although all three information layers are defined separately and independently, they are usually visualized together in the same *View* interface for interoperability issues and display of space relationships. Such an approach helps to establish easily the link between e.g. surface potential gradient analysis and the step and touch voltage measurement project definition.

### 2.1. Common Layer

Project definition starts in the *Common Layer* by entering general information about the power installation. Database entry is achieved through graphic user interface, which is also used for database browsing and entry editing. *Common Layer* is internally divided into more sublayers: *Plain Objects*, *Raster Background*, *Measuring Area*, *Measuring Analysis* and *Info Layer*.

*Plain Objects Layer* contains power installation definition as a set of vector entities. Two different classes of objects are employed for completeness of a definition: *Transparent* and *Constraint* type. *Transparent* objects serve to display parts of power installation (e.g. buildings contours) and they do not interfere with measuring point definitions in any way except for helping to establish relative space references. However, *Constraint* type objects are used to define *restrictive areas* of power installations where measuring points should not be placed (e.g. asphalt surfaces, physical obstacles etc.). *Transparent* and *Constraint* objects are both defined as graphical primitives (lines, curves, rectangles, polygons etc.) which can be freely combined into complex object groups.

*Raster Background Layer* enables the loading of a scanned power installation plan or aerial (satellite) photo into *Common Layer* background. This helps the placement of *Plain Objects Layer* entities according to real distribution of power installation objects.

*Measuring Area Layer* contains the definition of *Measuring Area Limits* for *Equipotential* and *Touch/Step Layers* measuring points.

*Measuring Analysis Layer* serves as a placeholder for visualizing equipotential and gradient analysis results of a surface potential distribution map (equipotential lines, areas and gradients).

*Info Layer* contains all additional information about power installation which do not fit previous categorization such as plan orientation (compass), visible discrete coordinate grid, title block, plain text comments etc.

### 2.2. Equipotential and Touch/Step Layers

*Equipotential* and *Touch/Step Layers* contain two independent sets of measuring points, one for *Equipotential*

part of a project definition, and the other for *Touch and Step*. Basic data object in those two layers is the *Measuring Point*. It includes data properties (such as space coordinates, order number, measuring type, ground type, comments and measured data values), graphic definition symbol (different for every measuring point type) and method procedures of interaction with other measuring objects and *Constraint*-type objects from *Common* layer. Each measuring point has unique order number (identifier) which also suggests preferred procedure for carrying out measurements. Order numbers are consecutive and represent *Measuring Points* places on the list. Order numbers of individual points might change when new *Measuring Points* are added to a layer or some existing removed from it. They are updated during such manipulations by means of embedded automatic renumbering procedures. Measuring point type can be either *Equipotential*, *Touch* or *Step*. Information about type is important to operator for choosing the right electrodes during measurement and for a measuring device to automatically select correct input analogue channel.

To define directives rather than single measuring points themselves, more complex data objects are derived from *Measuring Points*. Such approach is useful when e.g. large areas have to be covered with regular rectangular grid in which case manual approach of placement is inefficient. Complex data object structures are called *Measuring Point Collections* and represent algorithmic directives for the automatic layout of measuring points. Those structures consist of *Member Measuring Points* which are dependant entities loosely connected by means of selected autoplacement algorithms. There are two basic types of *Measuring Point Collections*: *Measuring Curves* and *Autoplacements*. Member measuring points of *Measuring Curves* lay on defined curves (lines, arcs, Bezier curves etc.). Parameters of *Measuring Curves* include adjustable distance between adjacent points and various curve parameters. *Autoplacements* are used to cover areas with measuring points using certain rules: measuring curves or single measuring points represent basic structural elements that can be used for building *Autoplacements*. When *Measuring Point Collection* is being translated, all member points are also translated accordingly. If any of *Measuring Point Collection* parameters is changed (e.g. endpoints of *Measuring Curve* or bounding rectangle of *Autoplacement*), attached set of member points is updated which can result in adding or removing of some member points. Automatic renumbering algorithms also take this into account to achieve consecutive order numbering on a complete project measuring point list.

*Equipotential* and *Touch/Step Layers* interact with *Constraint Objects* from a *Common Layer* when placement of individual points or point collections is performed. Program environment ensures that no measuring point is placed over an area occupied by *Constraint* object. However, overlapping between *Measuring Point Collections* and *Constraint Objects* is possible. This is done by automatically checking for every member point whether it falls into restricted area and deciding whether to show the point or not. Automatic numbering procedures take this

aspect into account as well, so the point order numbers on the list are not discontinued at any time.

### 3. INTEGRATED ENVIRONMENT STRUCTURE

Measuring project concept, along with the complete automated earth surface potentials measurement workflow as described, is not easily feasible by means of standard program equipment (existing database systems, CAD packages or specialized software). Integrated program environment, along with compatible *Intelligent Measuring Equipment*, had to be developed in order to achieve the goal of efficient automation of the measurement and validation process. Integrated program environment structure, founded on an object-oriented programming approach [4], is shown in Fig. 3.

Integrated environment consists of application core and surrounding special-purpose modules, among which the most important is the *Project Management Module* linked tightly to *CAD Engine*, *Communication Module* and *Report Engine*. Graphic user interface is based on *Operating System (OS) User Interface Services* (Win32 windowing system, user input, controls etc.), while external application link is implemented by using standard inter-application data exchange protocols (such as *Object Linking and Embedding (OLE)*, *Automation* and *Data Access Objects (DAO)*). They provide means of connecting application with widely used database systems, spreadsheet applications and text processors.

#### 3.1. Project Management Module

*Project Management Module* implements basic tools for measuring project management in database-like system. Such an information system is not based, as usual, on related tables and records, but rather on complex object model which defines means of data storage and object interaction according to *Measuring Project* concept specifications. Great deal of proposed specifications is implemented in stand-alone custom designed class libraries where are also definitions of all objects used in *Common Layer* and

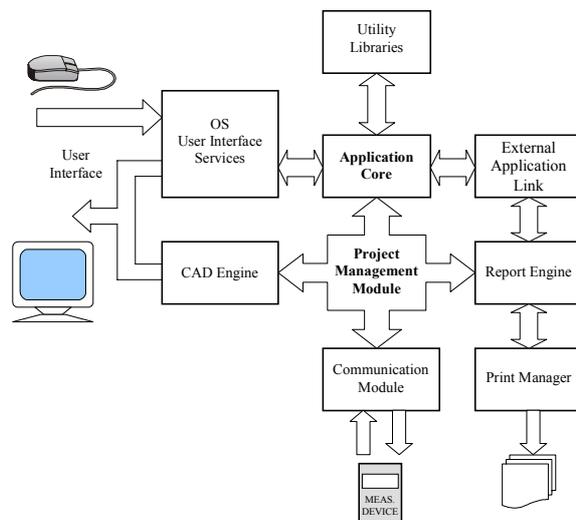


Fig. 3: Integrated Program Environment structure

*Measuring Point/Collections.* Advanced tools for graphic database entry and browsing are implemented in *CAD Engine* module while the procedures of data exchange with *Intelligent Measuring Equipment* are the part of *Communication Module*.

Some important procedures connected with measuring project management that have not been mentioned yet are *Project Splitting/Merging* and *Back Annotation* procedures. *Project Splitting* is a procedure of dividing large measuring projects into two or more parts (i.e. measuring devices) in order to achieve faster overall measurement time by performing parallel measurements. That approach enables reducing overall power installation idle time and performing measurements faster, prior to possible atmospheric conditions changes. Rain causes changes in soil resistance and, if it occurs during course of measurements, all previously measured results have to be discarded. *Project Merging* is an inverse procedure of assembling divided project parts back into original project to process and analyse measuring results. *Back Annotation* procedure enables *Template* updates for future reference. This includes cases when either new measuring points are defined during measurement phase or definitions of some existing points are changed in the memory of measuring equipment.

### 3.2. CAD Engine

*CAD Engine* module is the most complex part of integrated environment. It includes front-end drawing tools, graphic database entry, browsing and editing procedures and measuring results analysis visualisation routines. Simplified block scheme of *CAD Engine* is shown in Fig. 4.

*CAD Engine* is realized as an *ActiveX* component with a program core implementing basic vector and raster routines, coordinate transformations, embedded mathematical procedures and link with OS low-level *Graphic Driver Interface* routines (Win32 *Application-Program Interface (API)* function calls) for image rendering in target *handle device context (hDC)* [5]. Class libraries provide detailed descriptions of how every specific object should be displayed which is taken into account when rendering final image. Data part of the component contains all information layers represented as an object tree structure based upon class libraries definitions. *Event Handler* takes care of processing OS messages which includes interaction with user as well.

Numerical analysis routines are tightly connected with *CAD Engine* module, but they are placed into external libraries to speed up calculation time. The most complex problem is visualization of potential map from discrete data set of surface potential measuring points. Potential distribution function is interpolated from measuring data using Delaunay triangulation and B-spline interpolation approach [6]. Once potential distribution function is determined, component core program routines take care of visualizing equipotential lines, areas and gradient (module and direction) in *Analysis* part of the *Common Layer* according to parameters defined by user (adjustable levels and color mapping).

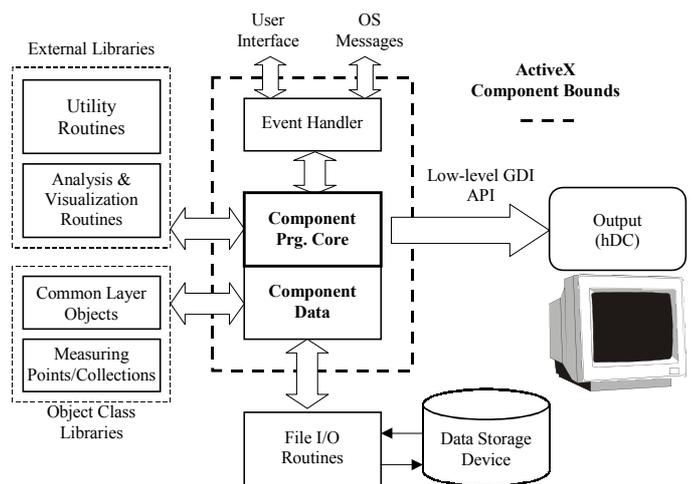


Fig. 4: Integrated environment *CAD Engine* structure

### 3.3. Communication Module

*Communication Module* converts measuring projects into representation suitable for *Intelligent Measuring Equipment* and also retrieves data after measurements are completed. Measuring results are converted back to appropriate layers (*Equipotential* or *Touch/Step*) in *Measurement* part of a project where final analysis and visualization take place.

This module supports procedures for measuring device memory backup and restoration capabilities as well, which means that it is possible to save memory contents of a measuring device prior to accidental data loss (e.g. on portable computer).

### 3.4. Report Engine

*Report Engine* is the part of integrated environment for automatic generation of all paperwork documentation that includes *Measuring Documentation* and *Reports*. *Measuring Documentation* includes all documentation which can be taken to measurement site to help operator to follow the project plan (automatically generated maps with marked measuring point positions and order numbers, tables etc). *Reports* denote final documentation based on measuring results analysis presented in predefined textual, tabular and graphical forms.

## 4. CONCLUSION

Described integrated program environment exhibits superiority over traditional manual approach. The greatest gains manifest in reducing the overall measurement time, faster and more efficient analysis of measuring results, reducing total procedure costs and more organized data management. *Project Splitting* and *Merging* procedures are particularly useful features, especially in situations when huge power installations need to be examined. *Reports Engine* and seamless integration with standard external applications provide end-users with powerful tools for generating various types of standard and custom designed reports.

## REFERENCES

- [1] CENELEC HD 637 S1, "Power installations exceeding 1 kV a.c.", 1999
- [2] ANSI/IEEE Std 81, "IEEE Guide for Measuring Earth Resistivity, Ground Impedance, and Earth Surface Potentials of a Ground System", 1983
- [3] R. Giannini, H.Džapo, "Embedded Device Concept for Earth Surface Potential Measurements of a Grounding System", In press
- [4] S. Khoshafian, R. Abnous, "Object Orientation: Concepts, Analysis & Design, Languages, Databases, Graphical User Interfaces, Standards, 2nd Edition", *John Wiley & Sons*, New York, USA, 1995.
- [5] D. Appleman, "Visual Basic 5.0 Programmer's Guide to the Win32 API", *Ziff-Davis Press*, Emeryville, California, USA, 1997.
- [6] M.H. Shultz, "Spline Analysis", *Prentice-Hall*, New York, USA, 1973.

---

**Authors:** **Hrvoje Džapo**, B.Sc.EE, Department of Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, Unska 3, HR-10000 Zagreb, CROATIA, Tel: +385 1 6129 977, Fax: +385 1 6129 652, E-mail: hrvoje.dzapo@fer.hr  
**Roberto Giannini**, M.Sc.EE, Department of Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, Unska 3, HR-10000 Zagreb, CROATIA, Tel: +385 1 6129 878, Fax: +385 1 6129 652, E-mail: roberto.giannini@fer.hr