A Web-Based Collection of Transient Analysis Cases

Benoît Bressac, Alain Xémard
Électricité de France
1 ave. du Général de Gaulle
92141 Clamart Cedex, France
benoit.bressac@edf.fr, alain.xemard@edf.fr

Antonín Zvácek
Laboratory of Zkušebnictví
Zkratovna, 190 11 Praha 9,
Bechovice, Czech Republic
zvacek@zku.cz

Jean Mahseredjian
IREQ / Hydro-Québec
1800 Lionel-Boulet
Varennes, Québec, Canada J3X 1S1
jean@ireq.ca

Abstract - This paper presents a web-based tool for collecting and maintaining power system cases on transient analysis. The targeted objective is to provide software validation for the Electromagnetic Transients Program (EMTP), but also to help EMTP users with practical study cases and component data compilation. Participating users can contribute to this collection through an established protocol in a web browser. This paper presents design details with usage mechanics, discusses some of the available cases and demonstrates the advantages of such a unique collection.

Keywords: software validation, data collection, web-based tool, EMTP, transient analysis

I. INTRODUCTION

One of the major difficulties in computerized analysis of power system transients is the determination of the study case. The study engineer must start by deciding on what is important to represent and which models are the most appropriate. The following step is finding data for selected components. A natural approach for starting a new study, is to search for existing similar cases in tutorials or collections of test cases.

Another major call for a collection of cases is during various software maintenance and development stages. Since the EMTP [1] is currently being recoded, it requires a major validation effort, that can be efficiently conducted by using a large collection of verified simulation results.

This paper presents the creation of a collection of cases available entirely on the Internet and using web tools. The creation of a web-based collection of cases is supported by the following considerations.

- A web-based collection can be uniquely (only one location) maintained and made available to all members of the cooperating organizations. It can be simultaneously accessed by online users. It inherits portability from web browser software.
- EMTP development and maintenance is a very long term activity. The validation process is continuously reactivated and cases are added regularly for new modeling and simulation capabilities. The validation process must be retraceable.
- A collection of study cases provides an exceptional database of knowledge on practical simulation needs. It can also evolve to represent a very broad range of possibilities related to transient analysis problems.

Establishing and maintaining a collection of cases is not a simple task, even if these cases are based on a single software environment, such as the EMTP.

The collection presented in this paper is new. It aims at the entire range of applications of EMTP. The cases are proposed by power system engineers from various specialties. There are several levels of requirements. In an ultimate case, it is needed to provide results from several sources: experimental measurements, theoretical calculations and numerical results from other simulation tools. In addition to practical cases it is also needed to create cases for stressing software features and numerical aspects.

Collection interrogation methods are as important as the test case submission methods. To create such a dual-purpose collection and to ensure its duration, several functionalities are essential. First, it is needed to provide a wide accessibility to the collection and en easy navigation method. Second, the case submission method needs to follow a rigid protocol for standardized presentations.

Web tools offer a perfect solution for navigating and searching through a collection of cases. The development of user interfaces and data sheets shown in this paper is based on a combination of HTML (Hypertext Markup Language [2]) and JavaScript [3]. This choice is mainly motivated by maximized customization needs and shown experience concludes that these languages allow an efficient software engineering cycle.

This paper starts with a brief overview of the programming possibilities offered by HTML and JavaScript. It highlights features retained for the web tool of this paper. It is followed by a presentation of the web-based tool and its underlying design methodology. The final section describes typical contents and usage experience.

II. HTML AND JAVASCRIPT CAPABILITIES

A. Overview

HTML is a language based on tags for presenting various document contents. Basic HTML tags specify such elements as paragraphs, headings and forms. An HTML file is an ASCII file (with an `htm` or `html` extension) interpreted by web browsers available on all computer platforms. Although some interpretation differences subsist between browsers and browser versions, it is feasible to create completely portable web pages. The ASCII file can be coded manually or by using a WYSIWYG editor.

JavaScript is a lightweight interpreted and standardized programming language with object oriented capabilities. JavaScript statements can be embedded directly in HTML code for performing various client-side tasks. Some HTML

editors allow inserting JavaScript lines or carry a library of JavaScript functions for various tasks. Advanced application programming, however, requires direct coding.

JavaScript uses objects to represent the web browser window and its contents. It has access to named HTML sections for interaction with document contents. The programmable access to the web browser window can avoid static HTML (page reload approach) and provides means for dynamic interaction with the user. JavaScript has event handling functions, such as `onClick`, `onMouseOut` and `onMouseOver`. There is a built-in math library. JavaScript is unrelated to Java, but can communicate with Java.

The combination of JavaScript and HTML allows programming dynamic HTML web pages and provides all the necessary ingredients for creating the collection tool described in this paper. The following sections describe form and window manipulation features most useful for the purposes of the collection tool.

B. Form creation

Typical input form elements offered by HTML include "text box" for entering text, "push button" for selecting an action, "radio button" for making a selection among a group of options and "check boxes". These elements can be initialized, to provide, for example, default data or instructions on how to fill out the form. HTML tables can be used to align or group related forms.

C. Form verification

A common approach for validating completed forms is to use a Common Gateway Interface (CGI) application on the server-side. For the purposes of the web-based collection it has been decided to validate forms with JavaScript. This provides fast simultaneous (while data is being entered) validation and allows using the collection tool off-line (while disconnected from the web).

JavaScript verification functions are invoked when the user completes a form and clicks a button or moves out of the form. A verification function can capture form data and verify a preprogrammed set of rules, such as empty and non-optional fields, data type or an invalid combination of selections. Invalid inputs are displayed to users through alert messages.

D. Generating HTML code

JavaScript can modify the browser window properties and contents. It create a new HTML document on the client-side, which can be used, to view a final validation on entered data, to summarize a submission or even to create case dependent forms. A web page can be dynamically created and opened by invoking a preset event handler.

E. Example

The simple example of this section is used to illustrate some of the JavaScript/HTML functionalities described above. It also provides a quick appreciation on programming language level and syntax.

The web page example of this section is created using the code lines of Figure 1 and shown in Figure 2. It shows a one-line textbox used for numeric data input and a pushbutton for performing a multiplication (by two) action. The resulting window appears on top and provides a button (`OK`) for closing itself.

As shown in Figure 1, an HTML page must start with an opening tag **`<html>`** and must end with a closing tag **`</html>`**. The head section includes a title tag and a JavaScript function named `multiply` referenced in the callback of the pushbutton created below. Bold characters are used in this presentation to indicate language keywords, built-in functions and built-in methods. The **`<body>`** tag starts the visible document part where a heading is followed by two input forms, one for entering data (text form) and another one of type button (pushbutton). The button object's **`onClick`** field identifies the JavaScript callback function. Other fields, such as **`style`**, are used in HTML tags to control appearance.

When `multiply` is called it starts by retrieving data from the text form and testing its contents. When the user enters nonnumeric data or does not enter any data, it is needed to send an alert message and to reset the contents of the text form. When valid data is found, it is needed to perform the multiplication and to create a new window (using the open method) for showing the answer. The **`write`** method of the document object allows to dynamically generate a web page. This powerful feature is very useful in the creation and submission of test case forms.

It is apparent from Figure 1, that JavaScript unlike Java and C, is an untyped language. This means that a JavaScript variable (declared with `var` keyword) can hold a value of any data type. This lack of typing permits, among other things, to conveniently append a number to a string, which is very useful for creating HTML code lines with the **`write`** method.

```
<html>
<head>
<title>Example</title>
<script language="JavaScript">
 // function called after clicking
 function multiply(win,form){
 var data = form.data.value;  //retrieve data
 // test if data is valid
 if ((isNaN(data)) || (data == "")  ) {
   //Invalid data, send an error
   alert("You must enter a number");
   //Reset the form
   form.data.value="";                }
 else{
  var result = data*2;  //find result

  //open new window of desired size
  var newwin =
  win.open("","answer",
                "width=250,height=100,top=400");
  var d = newwin.document;

  //start writing HTML code for new window
  d.write("<html>");
  d.write("<head>");
  d.write("<title>Multiplication result</title>");
  d.write("</head>");
```

```
  d.write("<body>"); //start the body section
  d.write(   "<h1>"+form.data.value+
             " * 2 = "+result+"</h1><p>");

  var formstr="<form><input type='button'"+
  " value='ok'" +
  " style='font-size: 12pt; font-weight: bold'"+
  " onclick='self.close()'></form>";
  d.write(formstr); //create the ok pusbutton

  d.write("</body>");  //end the body section
  d.write("</html>");  //end the HTML code
  d.close();     }
 }
</script>
</head>
<body>
<!--  Create a Header -->
<h4>Example with HTML and JavaScript</h4>
<!--  Create a form -->
<form name="myform">
  <p>
  <input type="text" size="16"
         style="font-size: 12pt" name="data">
  <input type="button"
     name="operation" value=" * 2"
     style="font-size: 12pt; font-weight: bold"
     onClick="multiply(top, this.form);"></p>
</form>
</body>
</html>
```

Figure 1 Simple JavaScript/HTML coding example

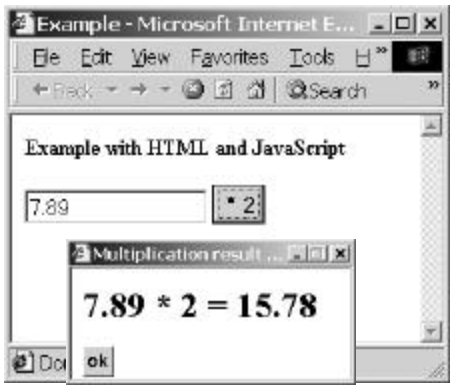

Figure 2 Web page created using the code of Figure 1

III. WEB-BASED TOOL

The previous section has shown that JavaScript and HTML provide a powerful, yet easy to program environment, for the deployment of a large test case database. In addition to previously stated design principles, a very important criteria is simplicity of usage and maintenance. Navigation is based on a coherent set of hyperlinks. Pages must provide meaningful information allowing to make quick selections.

The web-based collection starts by showing two main entry points: test case database (collection) links and test case submission links.

A. Collection categories

The collection is categorized through the following sections:

- Component type. This category is arranged to test component models such as transformer, switch, nonlinear branch, synchronous machine, line and cable models and all other available models of EMTP.
- Stressed aspect. In this section are placed cases designed for specific testing of software options, solution methods and numerical performance issues.
- Study type. This section regroups practical study cases from various fields of interest: power electronics, switching transients, insulation coordination. It also holds subsections of voltage level or study frequency range. The insulation coordination cases, for example, are grouped into temporary, slow-front or very-fast-front overvoltages.

A search tool is developed using Perl. It allows retrieving cases by keywords.

A partial view on a set of insulation coordination cases is given in Figure 3.

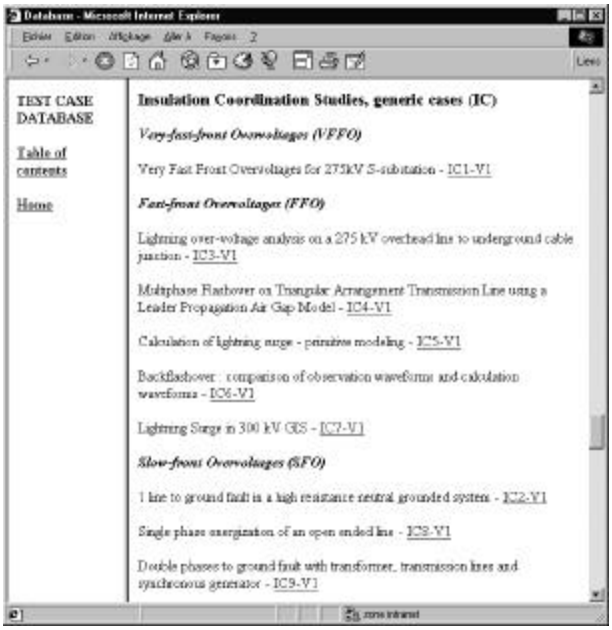

Figure 3 Partial view for a collection of cases on insulation coordination

B. Electronic forms

The case collecting protocol works as follows. Each test case possesses three electronic data sheets: one for detailed test case description ("Description form"), one for showing expected simulation results ("Results form") and the last one ("Bug report form") for reporting bugs or doubts on simulation results. To create these electronic data sheets, users must fill JavaScript/HTML based forms. A correctly completed form automatically generates an HTML data sheet. The EMTP input file and all other related files and schematics are packaged with the HTML files and sent to the webmaster through email. It is foreseeable to use a

dedicated ftp site for collecting larger cases. The webmaster must proceed through a final visual validation and reject or integrate the submitted test case into the collection. Programming form validation methods can become very complex and tedious, that is why the webmaster approach has been retained.

Each form possesses a pushbutton (`Example`) allowing to automatically set values into the different fields in order to view an example of completed form. There are required and optional fields. All fields are checked by JavaScript functions before generating the corresponding HTML data sheet.

C. Description form

The "Description form" is the test case identification form. A partial view for a typical example is shown in Figure 4. The first group of form fields is related to the presentation of the test case and its source: author's name, e-mail and affiliation, test case title, descriptive keywords, reference tag and date. Following fields present various test case attributes. Among them are the names of the data file, schematic diagram or image file, available pdf documents and EMTP software version. Even if detailed information can be found in the schematic diagram or ASCII data file version of the test case, it is needed to provide a sufficiently good summary on test case contents to allow rapid browsing and searching. An important field is the physical phenomenon description field. Another field provides a list of test case components.

When the data sheet is successfully created, it contains hypertext links to all files needed to recreate the test case. A partial view to a "Description form" data sheet for a ferroresonance case is given in Figure 5.

Figure 4 Partial view for a "Description form" example

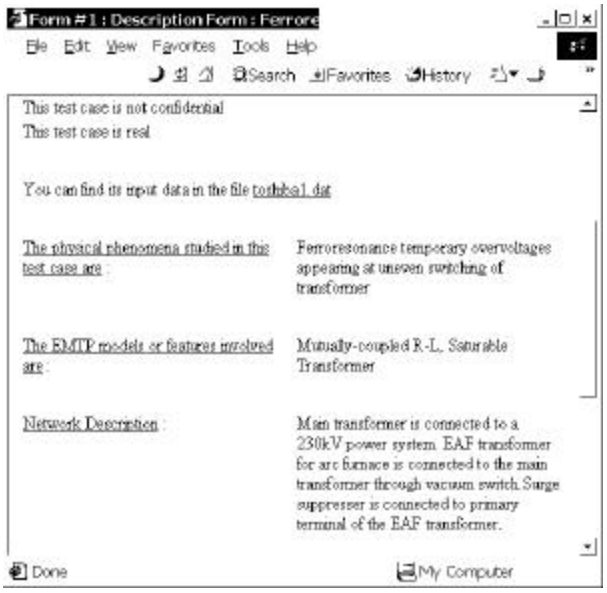

Figure 5 Partial view to a "Description form" data sheet

D. Results form

This web page contains several groups of fields. There is a textbox for entering expected results based on physical phenomenon understanding, experimental results, previous observations or theoretical analysis. Checkboxes allow to enter the provenance of available results.

Another group of fields is used to describe and submit simulation waveforms and/or related data. It is followed by sections on result comparison, other validation issues and finally a section for concluding remarks.

E. Bug report form

The last form is the bug (or problem) report form. For each software related incident the user must provide a summary and a description on the incident with an indication on incident severity for the case: critical, high, medium and low. There is also a field for indicating exactly how to reproduce the problem in the software.

IV. CONTENT EXAMPLES

Presently the collection contains test cases submitted by EDF, the laboratory of Zkusebnictví, Hydro-Québec and CRIEPI. Validation parts are reserved for the exclusive use of developers.

Due to the importance of the subject, a significant part of the collection is related to Slow Front Overvoltages (SFO) [4]. These overvoltages originate mostly from sudden changes in the power system configuration that cause a voltage step or current injection to be applied to the system. Some of those sudden changes are caused by unpredictable external events, such as faults, while some others are due to intentional switching operations. The aim of this part of the collection is to propose practical cases simulating typical events that originate switching surges. About forty cases proposed by the laboratory of

Zkusebnictví deal with this subject. Each case is given electronic data sheets, input files and related schematic diagrams. These cases involve a broad range of EMTP models, including overhead transmission lines, underground cables, transformers, surge arresters, breakers and simple lumped linear models.

Due to the probabilistic nature of event initiation, several cases must apply statistical simulation options in order to describe switching overvoltages in probabilistic terms.

A. Energizations

Several cases deal with energization:

- energization of transmission line and cables [5] involves voltage surge propagation and reflection phenomena;
- energization of capacitor banks showing related overvoltage and inrush current conditions
- transformer energization [7];
- energization of transformers terminated with line or cable [8].

Some cases are designed to account for many factors affecting the magnitude of overvoltages:

- network parameters, most particularly for source representation;
- energization instant related to the power frequency;
- the non-simultaneous closing of phases;
- line compensation with shunt reactors;
- presence of trapped charge.

Test cases also include devices for the reduction of transient overvoltages. Switching overvoltages with or without the installation of such devices are compared.

Typical study examples are: energization of an open 200km (220kV) compensated line, reclosing for an open 5km (220kV) underground cable into trapped charge, energization of an unloaded 220/60kV transformer and energization of a 90kV capacitor bank.

Typical wave shapes relevant to the opening and reclosing of a no-load and shunt compensated 220kV line, are shown in Figure 6. The opening operation is followed by oscillations between the shunt reactors and the line capacitances. Due to a compensation factor of 86.4%, the frequency of these oscillations is lower than the power system frequency. The reclosing instant determines the voltage across each breaker pole and thus the reclosing overvoltages.

B. Interruption of currents

Different kinds of current interruptions can lead to transient overvoltages, specially across the contacts of circuit breakers (TRV). The evolution of dielectric strength between the separating contacts determines the mechanisms of circuit interruption including reignition and restrikes. Test cases available in this section are designed for studying current interruption problems:

- opening of reactors and no-load transformers [9];
- opening of unloaded compensated and uncompensated lines with trapped charge;
- deenergization of capacitor banks with or without damping circuits [6] [10].

Some examples for such studies are: opening of compensation coil, deenergization of a 90kV grounded capacitor bank, and opening of a 200/60kV transformer terminated line.

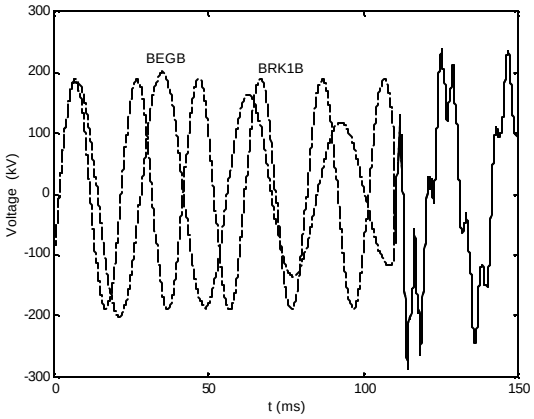

Figure 6 Opening and reclosing of a no-load and shunt compensated line, BRK1B is on the source side and BEGB is on the line side

C. Faults

Fault initiation and elimination can cause both switching and temporary overvoltages [11]. Switching overvoltages at fault initiation are strongly related to the instant of sine wave at which the fault occurs. Temporary overvoltages depend on the type of system grounding and the ratio of the zero-sequence to the positive-sequence impedances of the faulted network. Cases simulating single-phase-to-ground, 2-phases-to-ground, 3-phases-to-ground and phase-to-phase faults have been included in the collection.

V. USAGE FOR VALIDATION

A collection of practical cases is very important for the targeted software design quality and usability. The web-based tool presented in this paper is used on a continuous basis for the development of successive new EMTP versions. It has been found to become extremely useful in testing component models and program features. When a new component is added, the database of test cases is used to rapidly extract several cases containing such a component, various topological situations and realistic data possibilities are tested. Tests results are then compared with provided and verified waveforms.

The existing set of tests related to programming features and numerical behavior ("Stressed aspect" category) is continuously updated to account for new possibilities offered at each software development stage.

Experience has shown that readily available and specifically designed test cases, result into a significant effort reduction for software development and allow to achieve higher quality in fewer iterations.

VI. CONCLUSIONS

This paper has presented the design of a web-based collection of EMTP cases. Although the original intent was to design a software validation site, experience has shown that such a collection can become a very useful tool for engineers conducting transient analysis studies. The web allows to maintain the collection at a unique location accessible to all users.

Data capture forms have been programmed using JavaScript/HTML. This approach inherits portability, simplifies form programming and allows advanced customization. The initial programming effort pays off in the long term usage.

A comprehensive database of practical test cases has always been a desirable feature for EMTP users. The next step could concentrate on data portability using a standardized data format.

VII. REFERENCES

[1] Electromagnetic Transients Program (EMTP), Development Coordination Group of EMTP.

[2] C. Musciano and B. Kennedy: HTML The Definitive Guide. O'Reilly & Associates, 1992, 2rd Edition.

[3] D. Flanagan: JavaScript The Definitive Guide. O'Reilly & Associates, 1998, 3rd Edition.

[4] IEC 60071-2 standard, Insulation Coordination Group. 1996

[5] IEEE Working Group on Switching Surges: Part IV, Control and Reduction on AC Transmission Lines. IEEE Trans. on PAS, Vol. PAS-101, No. 8, August 1982.

[6] P. Vukelja, J. Mrvic, M. Sencanic, D. Hrvic, D. Radulovic: Transient Phenomena at Energization and Deenergization of Capacitor Banks. IPST'99 Proceedings pp. 131-136.

[7] M. Laasonen, R. Hirvonen : Inrush Current of Three-Winding Transformers. IPST'95 Proceedings pp. 142-147.

[8] G. Chr. Paap, A. A. Alkema, L. Van der Sluis: Overvoltages in Power Transformers Caused By No-Load Switching. IEEE Trans on Power Delivery, Vol. 10, No. 1, January 1995.

[9] L. Prikler, G. Ban, G. Banfai: EMTP Models for Simulation of Shunt Reactor Switching Transients. IPST'95 Proceedings pp. 178-183.

[10] T. E. Grebe: Technologies For Transient Voltage Control During Switching of Transmission and Distribution Capacitor Bank. IPST'95 Proceedings pp. 189-194.

[11] D. Santos, G. Cabriel: Transient Recovery Voltages When Clearing a Fault in Presence of Series Limitation Reactors. IPST'99 Proceedings pp. 105-111.