

Automatic simulation of IED measurements for substation data integration studies

Y. Wu, *Student Member, IEEE*, and M. Kezunovic, *Fellow, IEEE*

Abstract—With the deregulation and restructuring of utility industry, many substation automation applications are being developed. These applications need to be tested for their accuracy and reliability. In cases when physical testing in substations is difficult, it is desirable to have a software tool to simulate the Intelligent Electronic Device (IED) measurements in substations. This paper presents the design of such a software application that can be used to generate substation IED measurements in fault and/or switching scenarios. The software structure and input/output interfaces are provided, as well as their implementation details. Two examples of simulation results are included in the paper. This software application serves as a handy tool for substation automation related studies.

Index Terms—Alternative Transient Program (ATP), Energy Management Systems (EMS), Intelligent Electronic Device (IED), Supervisory Control and Data Acquisition (SCADA), substation switching, transients simulation

I. INTRODUCTION

WITH the deregulation and restructuring of utility industry, it is desirable to introduce more automated functions and use more advanced information technology to reduce the operation costs. Some potential Supervisory Control and Data Acquisition (SCADA) and Energy Management System (EMS) applications of substation automation systems are being developed, such as automated execution of control and restoration sequences [1]; on-line monitoring and failure analysis of circuit breakers (CBs) [2]; enhancement of the traditional state estimation with local substation data [3]; automated analysis of faults [4]. Some new technologies, such as Petri Nets, are also being used to improve substation automation functions [5]-[6].

Many of these new functions need to be tested for their reliability and accuracy before they will be physically implemented in real substations. In most cases, it is difficult to test these applications in a real substation environment, because such experiments are expensive and time-consuming. Thus a software substation model that can fully simulate the

This work was supported in part by PSec Project T-17: “Enhanced Reliability of Power System Operation Using Advanced Algorithms and IEDs for On-Line Monitoring”.

Y. Wu and M. Kezunovic are with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843-3128, USA (e-mails: wuyang@ee.tamu.edu, kezunov@ee.tamu.edu).

IED measurements coming from a substation is highly helpful.

This paper introduces a novel design of software application utilizing the Alternative Transient Program (ATP) that can automatically generate substation IED measurements in fault or switching scenarios according to the user’s definitions. The user can define faults, switching sequences and failure possibility of circuit breakers (CBs). The software is also capable of initiating breaker failure schemes in case any CB fails. The analog and digital data gathered can then be regarded as coming from the “virtual” IEDs in a substation, and be used for different types of substation automation studies.

Section II describes the design of data simulation (DataSim) software application, including the introduction of input and output files as well as the implementation of each software modules. Section III lists some examples of simulation results. Conclusions are presented in Section IV.

II. THE DESIGN OF DATA SIMULATION

A. Overview

The Alternative Transients Program (ATP) is the most widely used electromagnetic transients program [7]. In the design of DataSim, ATP was used as the simulation tool.

Fig. 1 shows the overall software structure of DataSim. The following describes how DataSim works.

- 1) Before simulation, two files need to be prepared – the Sequence/Topology Information File, and the Template ATP File.

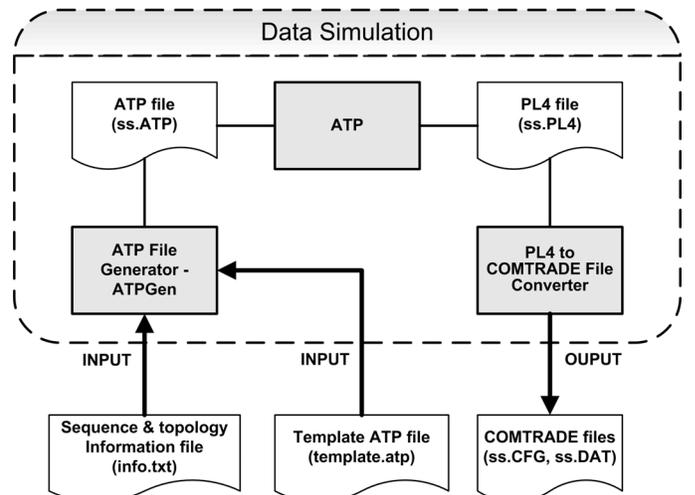


Fig. 1. Data simulation software structure.

- 2) These two files are then passed to the ATP File Generator (ATPGen) as input parameters to create an ATP file that corresponds to a certain substation switching sequence.
- 3) The ATP file can be used to initiate simulation using the ATP program.
- 4) The data simulation results are stored in a PL4 file, which is created by the ATP program. This PL4 file contains simulated data coming from Current Transformers (CTs), Voltage Transformers (VTs) and circuit breaker status measurements in the substation ATP model.
- 5) The PL4 file can then be converted to Common Format for Transient Data Exchange (COMTRADE) files (.CFG, .DAT). COMTRADE is a common format for data files and exchange medium used for the interchange of various types of fault, test, or simulation data for electrical power systems [8].
- 6) In a physical substation, transient data may come from Digital Fault Recorders (DFRs), in the format of a single pair of COMTRADE files (.CFG, .DAT). Alternatively, transient data may come from separate IEDs. In this case, each IED has its own pair of COMTRADE files. Accordingly, in DataSim the PL4 file can be converted to a single pair of COMTRADE files (.CFG, .DAT) or a group of COMTRADE files.

B. Input data

1) *Template ATP File*: In order to execute DataSim, first a template ATP model of the entire substation needs to be created using ATP's graphic tool ATPDraw [9]. Since ATP does not support the reclosable switch model, a custom circuit breaker group model was created using Transient Analysis of Control Systems (TACS) techniques [10]. Its internal logic checks the current going through the CB and opens it only when the current reduces to zero after the trip signal is issued. In the models of CBs and disconnecting isolators (DISs), a pair of very small impedances was added in series with each of the switches for the numerical stability purpose. A fault model, which consists of four impedances and four switches, was also created. By using different impedance values and properly setting the switching times, all types of phase-to-ground and phase-to-phase faults can be applied onto the transmission lines. Three custom group model symbols (CB, DIS and fault) were created to make the models more compact in appearance. These symbols and their underneath structures are shown in Table I.

A sample substation ATP model shown in Fig. 2 was created using the three group models. The substation consists of 3 buses, 3 transmission lines, 1 transformer, 7 CBs, 14 disconnecting isolators, 15 CT, and 6 VTs. Two voltage levels are involved: 220kV and 132kV. The layout of devices and their names are shown in Fig. 2(a). Its corresponding ATP model is shown in Fig. 2(b). All switching devices were set not to operate during the simulation so that the simulation result of the template substation ATP model corresponds to the steady state situation. This ATP model was saved as a 40.4kB ATP

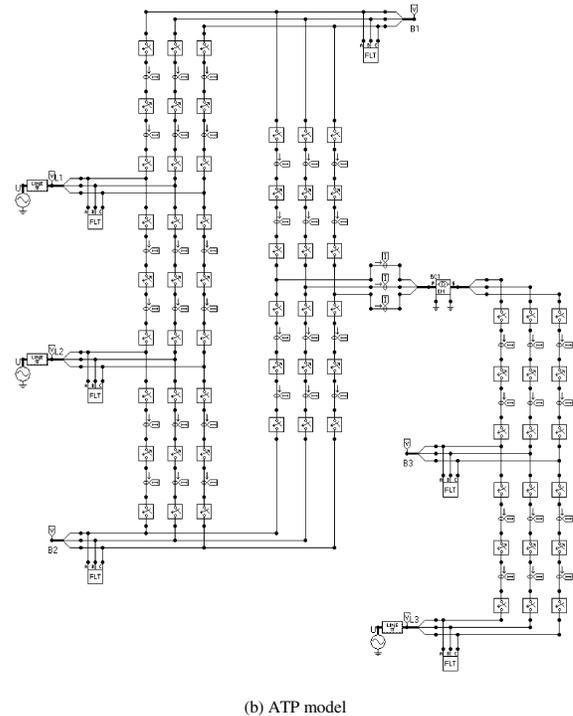
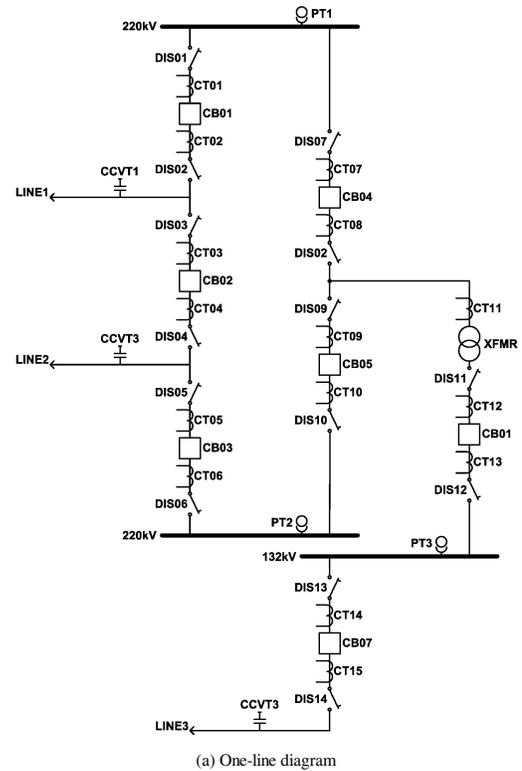
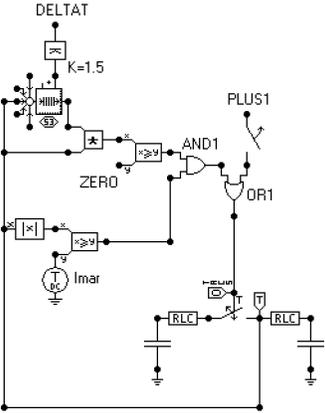
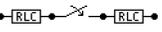
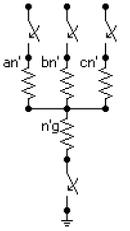


Fig. 2 The sample substation model.

file and it was used as one of the input files for DataSim.

2) *Sequence/Topology Information File*: This file is created by the user. The purpose of this file is to provide necessary information to DataSim so that it can modify the template ATP file to reflect the switching scenario to be simulated. The sequence/topology information file consists of four sections:

TABLE I
CUSTOM ATP GROUP MODELS

Symbol	Meaning	Diagram
	Circuit breaker	
	Disconnecting Isolator	
	Phase-to-Ground or Phase-to-Phase Fault	

- Topology and Switch Initial Statuses
- Fault Models Parameters
- Switching Sequence Descriptions
- Simulation Parameters

Section a) mostly defines the name and type of each device, its from/to nodes in the ATP model, initial status, corresponding current transformers, backup circuit breakers, and failure possibilities. Section b) defines the times and types of fault occurrences and value of fault resistances. According to the type of fault, one to four fault resistance values need to be designated by the user ($R_1 \sim R_4$), as shown in Fig. 3. The DataSim software then internally calculates the proper values to be set in the ATP fault models. Table II shows the relationship between the user-input fault resistance values and the corresponding ATP fault model values. Section c) contains the switching sequence of switches, including their names and operation times. Section d) consists of several simulation parameters needed for the ATP simulation and the PL4 to COMTRADE file conversion.

C. Output data

The output data of DataSim are the analog and digital signals measured by the CT/VTs and contact signal probes in the ATP model. The raw output data file format is PL4. However, the PL4 file can be converted into COMTRADE format files. These output files are named following the IEEE File Naming Convention [11]. An example of the output file name is

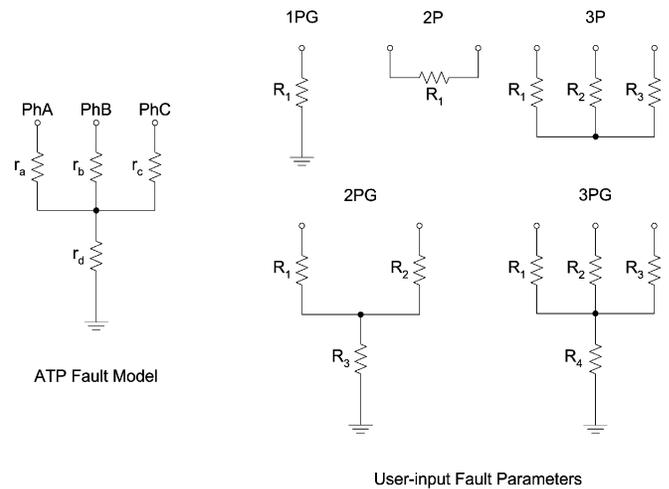


Fig. 3. Fault Resistances.

TABLE II
FAULT RESISTANCE CONVERSIONS

Fault Type	r_a	r_b	r_c	r_d
AG	$R_1/2$	∞	∞	$R_1/2$
BG	∞	$R_1/2$	∞	$R_1/2$
CG	∞	∞	$R_1/2$	$R_1/2$
AB	$R_1/2$	$R_1/2$	∞	∞
AC	$R_1/2$	∞	$R_1/2$	∞
BC	∞	$R_1/2$	$R_1/2$	∞
ABG	R_1	R_2	∞	R_3
ACG	R_1	∞	R_2	R_3
BCG	∞	R_1	R_2	R_3
ABC	R_1	R_2	R_3	∞
ABCG	R_1	R_2	R_3	R_4

illustrated in Fig. 4. Some important information of the simulation results such as simulation time and device name is directly visible from their file names.

D. Description of implementation

DataSim was developed in C++. It mainly consists of five classes. Fig. 5 shows the Universal Modeling Language (UML) diagram of DataSim and the description of all custom data-types that were used.

1) *DataFile*: DataFile is the parent class of classes PL4 and COMTRADE. It stores the data contained in an ATP simulation output file or a DFR record file. This class does not have enough member functions to complete most reading or writing tasks on such files. It only creates a framework to save the analog and digital data. Member function GetFileName() creates a proper IEEE file name.

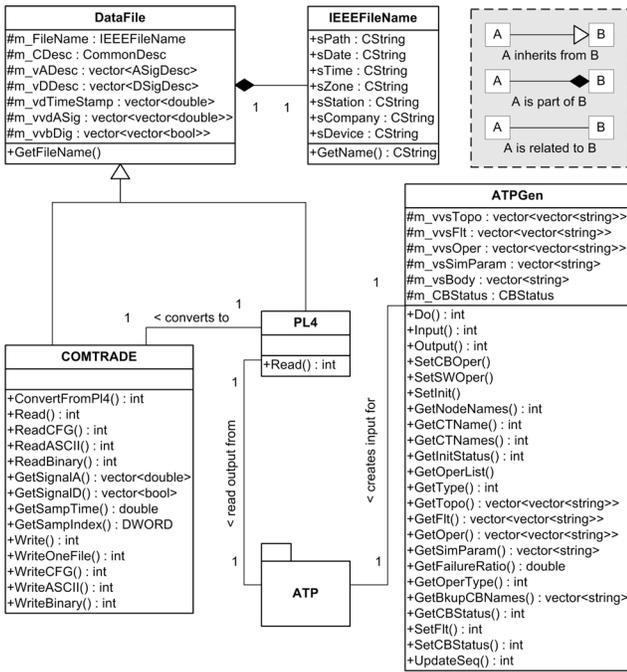
2) *IEEEFileName*: Class IEEEFileName stores the data

041023,00205000,+5D,WERC,CT03A,TAMU.dat

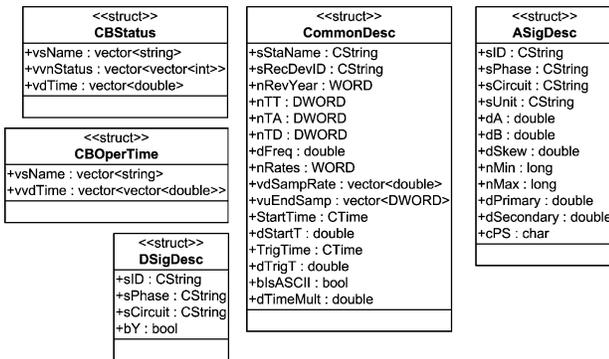
↑ ↑ ↑ ↑ ↑ ↑

date time time zone substation device company

Fig. 4. An example of the IEEE file naming convention.



(a) Class models



(b) Data-types

Fig. 5. The UML model of DataSim.

fields needed in a proper IEEE file name, such as date, time, time zone, substation name, company name, device name, etc. Member function `GetFileName()` returns a proper IEEE file name.

3) *PL4*: Class *PL4* is directly derived from class *DataFile*. It only contains a single member function `Read()`, which reads a binary format *PL4* file and stores all data in the *PL4* file into member variables of class *DataFile*.

4) *COMTRADE*: Class *COMTRADE* is directly derived from class *DataFile*. A number of member functions are available for the manipulation of *COMTRADE* files. `ConvertFromPl4()` reads a *PL4* file and a topology/sequence information file, and converts corresponding data fields into proper *COMTRADE* format. `Read()`, `ReadCFG()`, `ReadASCII()` and `ReadBinary()` deal with the reading of *COMTRADE* files. `Write()`, `WriteOneFile()`, `WriteCFG()`, `WriteASCII()`, `WriteBinary()` deal with the writing of *COMTRADE* files. `GetSignalA()`, `GetSignalD()`, `GetSampTime()` and `GetSampIndex()` extract the values of certain types of data fields from the member variables in class

DataFile.

5) *ATPGen*: Class *ATPGen* deals with the creation of ATP files based on the topology/sequence information file. Member variables `m_vsTopo`, `m_vsFlt`, `m_vsOper` and `m_vsSimParam` store the data in the topology/sequence information file. `m_vsBody` stores the information in the template ATP file. `m_CBStatus` stores the circuit breaker statuses at different times. Member functions `Input()` and `Output()` deal with the input and output functions. `Do()` starts a complete ATP file generation by calling `Input()`, `UpdateSeq()` and `Output()` internally. `SetCBOper()`, `SetSWOper()`, `SetInit()`, `SetFlt()` modify the corresponding parts in the template ATP file according to the switching sequence and fault information. `SetCBStatus()` modifies the member variable `m_CBStatus` to reflect any changes in circuit breaker statuses. `GetNodeNames()`, `GetCTName()`, `GetCTNames()`, `GetInitStatus()`, `GetOperList()`, `GetType()`, `GetTopo()`, `GetFlt()`, `GetOper()`, `GetSimParam()`, `GetFailureRatio()`, `GetOperType()`, `GetBkupCBNames()`, `GetCBStatus()` provide convenient interfaces for retrieving specific data fields in either the topology/sequence information file or the protected member variables.

The member function `UpdateSeq()` is an important member function that deals with the updating of the switching sequence and launching breaker failure schemes in case there are failed circuit breakers. In order to do so, the CB statuses are constantly monitored during the simulation.

CB statuses are expressed by a 3-bit binary code, as show in Table III. Bit 0 is either CLOSED or OPEN, indicating the open/close status of a circuit breaker. Bit 1 is either ENABLED or DISABLED. When a circuit breaker is ENABLED, it accepts operation signals; otherwise all operation signals issued to this circuit breaker are blocked and the circuit breaker cannot be operated until its status is ENABLED again. Bit 2 is either HEALTHY or UNHEALTHY. When a circuit breaker is HEALTHY, it is in a normal state and can execute operations correctly; otherwise the circuit breaker has failed and needs repairing.

It should be noted that a DISABLED circuit breaker is not

TABLE III
CIRCUIT BREAKER STATUSES

Bit	0	1	2
Value	C = CLOSED O = OPEN	E = ENABLED D = DISABLED	H = HEALTHY U = UNHEALTHY

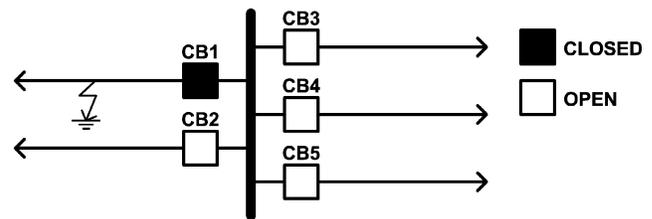


Fig. 6. A breaker failure scheme.

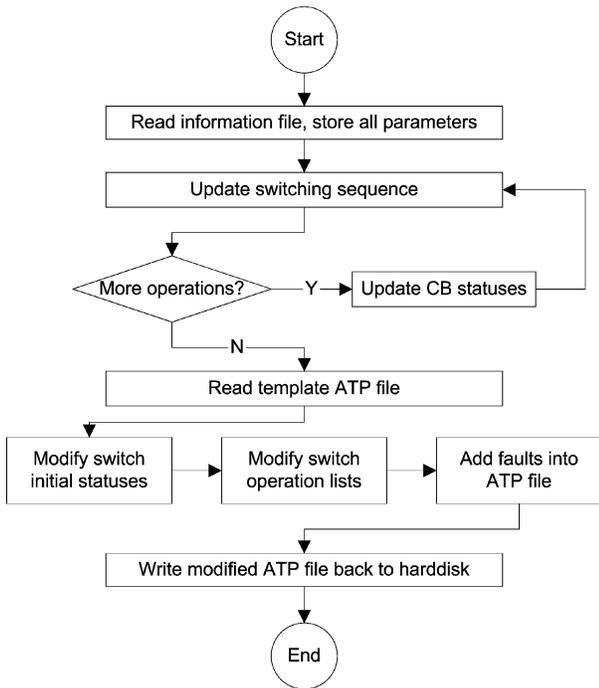


Fig. 7. The flowchart of ATP file generation.

necessarily an UNHEALTHY circuit breaker. For example, in a breaker failure scheme shown in Fig. 6, when the fault occurs and CB1 fails to trip, all other CBs need to trip in order to clear the fault. The backup breakers (CB2, CB3, CB4 and CB5) are HEALTHY but they become DISABLED after tripping, since they have to isolate the faulted transmission line and thus are not allowed to re-close.

Likewise, an ENABLED circuit breaker is not necessarily HEALTHY, because although a circuit breaker can accept operation commands, it might physically fail at the time of execution.

Fig. 7 shows the flowchart of member function Do(). By repetitively reading and updating CB statuses array, ATPGen decides whether or not a switching operation is successful and when a breaker failure scheme should be initiated.

III. SIMULATION RESULTS

DataSim has been tested in an integrated environment of substation automation project. It provides data to a number of application including automated digital protective relay data analysis (DPRA), verification of IED data stored in substation database (VSDB) and substation switching sequences verification (SSSV). Numerous test cases have been examined and the correctness of DataSim has been demonstrated. Some simulation results will be listed in this section.

A. Test case

In this test case, the following switching scenario is pre-defined: originally ($t=0.0s$) all switches in the substation are closed and no fault exists. At $t=0.1s$, a temporary phase-A-to-ground fault occurs on line 1, and it disappears at $t=0.3s$. CB01A and CB02A open at $t=0.2s$ to clear the fault. At $t=1.2s$, these two CBs re-close. Since at that time the fault has

disappeared, they remain closed until the simulation ends.

The summary of the above events is shown in Table IV.

A failure possibility of 50% is given to CB01A, which means this circuit breaker has a 50% chance to fail and not to follow commands.

B. Test results when CB fails

When CB01A fails during the simulation, the original switching sequence cannot be executed correctly. DataSim reports an updated switching sequence reflecting the action of the breaker failure scheme. The resulted events are listed in Table V; simulation results are shown in Fig. 8.

From Fig. 8(a), it can be seen that CB01A's status remains CLOSED during the whole simulation, indicating that the CB

TABLE IV
TEST EVENTS

Device	Time (s)	Event
Line 1	0.1	AG fault occurs
CB01A	0.2	opens
CB02A	0.2	opens
Line 1	0.3	fault disappears
CB01A	1.2	re-closes
CB01A	1.2	re-closes

TABLE V
GENERATED EVENTS WHEN CB01A FAILS

Device	Time (s)	Event
Line 1	0.1	AG fault occurs
CB02A	0.2	opens
CB04A	0.28	opens
Line 1	0.3	fault disappears

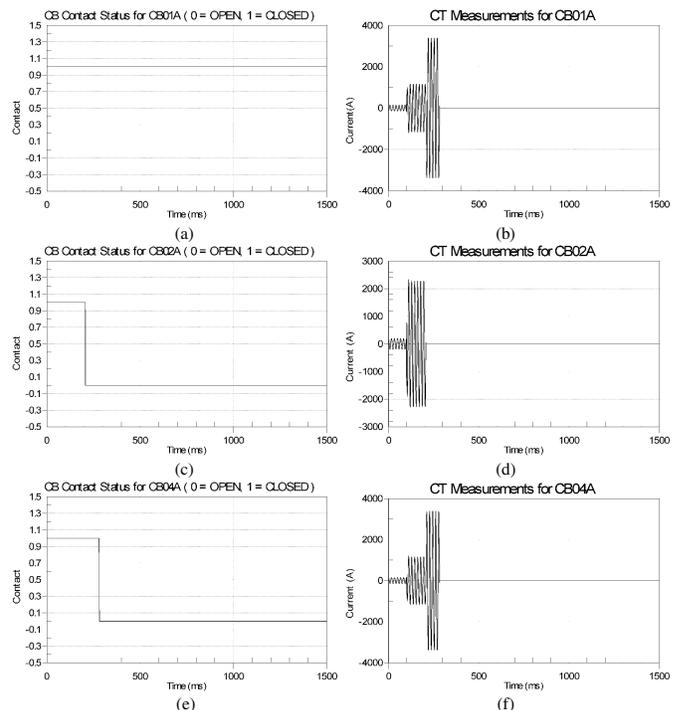


Fig. 8. Simulation results when CB01A fails.

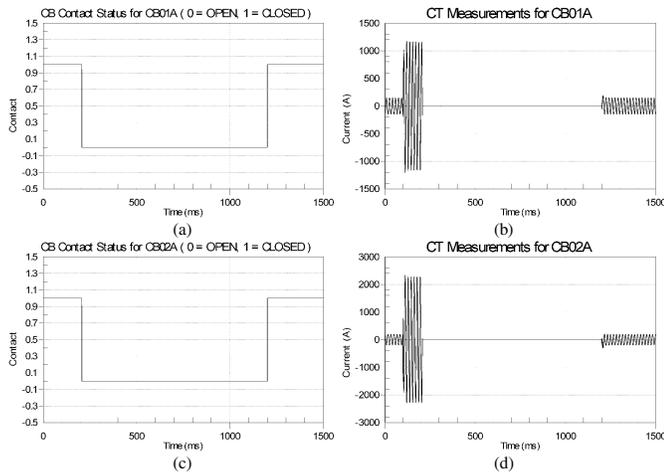


Fig. 9. Simulation results when CB01A works correctly.

fails to follow the trip signal.

Fig. 8(c) shows that CB02A correctly opens at $t=0.2s$, indicating that this CB is in a proper working condition.

Fig. 8(d) shows that breaker failure scheme detects the failure of CB01A and trips CB04A at $t=0.28s$. After that, CB01A is backed-up by CB02A and CB04A, and the faulted transmission line is successfully isolated.

Fig. 8(e)-(f) show the measurements of currents that go through CB01A, CB02A and CB04A during the simulation. They further confirm that CB01A fails to trip at 0.2 sec and CB04A trips at 0.28 sec as a backup breaker.

C. Test results when no failure occurs

When CB01A works properly during the simulation, the generated events exactly obey the pre-defined events as in Table IV. The CB statuses and CT measurements for CB01A and CB02A are shown in Fig. 9.

IV. CONCLUSIONS

A novel software application for automatic substation transient data simulation is introduced in this paper. Being an important enhancement of traditional ATP transient simulations, it greatly reduces the workload of substation data simulation tasks. It can be used as a valuable tool to test various new substation automation applications in a more realistic environment.

Many tests have been done with the data simulation application, showing that the generated switching sequences and corresponding transient data are correct. The software application has been integrated in a substation automation software package, serving a number of different automation functions as their data source.

REFERENCES

- [1] M. S. Sachdev, P. Dhakal, T. S. Sidhu, "A computer-aided technique for generating substation interlocking schemes," *IEEE Transactions on Power Delivery*, vol. 15, no. 2, pp. 538-544, April 2000.
- [2] M. Kezunovic, C. Nail, Z. Ren, D. R. Sevcik, J. Lucey, W. Cook and E. Koch, "Automated circuit breaker monitoring and analysis," in

Proceedings of 2002 IEEE Power Engineering Society Summer Meeting, vol. 1, pp. 559-564, 21-25 July 2002.

- [3] S. Jakovljevic and M. Kezunovic, "Advanced substation data collecting and processing for state estimation enhancement," in *Proceedings of 2002 IEEE Power Engineering Society Summer Meeting*, vol. 1, pp. 201-206, 21-25 July 2002.
- [4] M. Kezunovic, "Practical applications of automated fault analysis," in *Proceedings of PowerCon 2000*, vol. 2, pp. 819-824, Dec. 2000.
- [5] J. L. Pinto de Sa and J. P. Sucena Paiva, "Design and verification of concurrent switching sequences with Petri nets," *IEEE Trans. on Power Delivery*, vol. 5, no. 4, pp. 1766-1772, Oct. 1990.
- [6] K. L. Lo, H. S. Ng, D. M. Grant and J. Trecat, "Extended Petri net models for fault diagnosis for substation automation," in *IEE Proceedings of Generation, Transmission and Distribution*, vol. 146, no. 3, pp. 229-234, May 1999.
- [7] *ATP - Alternative Transient Program Rule Book*, Leuven EMTP Center, Leuven, Belgium, July 1987.
- [8] IEEE Std C37.111-1999, *IEEE standard Common Format for Transient Data Exchange (COMTRADE) for power systems*, Power Systems Relay Committee of the IEEE Power Engineering Society, USA, 1999.
- [9] L. Prikler and H. K. Høidalen. *ATPDraw Version 3.5 for Windows 9x/NT/2000/XP Users' Manual*, INTEF/EFI, Trondheim, Norway, Oct. 2002.
- [10] L. Dube and H. W. Dommel, "Simulation of Control Systems in an Electromagnetic Transients Program with TACS," in *Proceedings of IEEE Power Industry Computer Application Conference*, pp. 266-271, 1977.
- [11] *File Naming Convention for Time Sequence Data*, Final Report of IEEE Power System Relaying Committee Working Group H8, USA, 2001.

Yang Wu (S'05) received his B.S. and M.S. degrees from Xi'an Jiaotong University, Xi'an, China, both in electrical engineering, in 1999 and 2002 respectively. He has been a Ph.D. student in Texas A&M University since Aug. 2002. His research interests include protective relaying, substation automation and state estimation.

Mladen Kezunovic (S'77, M'80, SM'85, F'99) received his Dipl. Ing. Degree from the University of Sarajevo, the M.S. and Ph.D. degrees from the University of Kansas, all in electrical engineering, in 1974, 1977 and 1980, respectively. He has been with Texas A&M University since 1987 where he is the Eugene E. Webb Professor and Director of Electric Power and Power Electronics Institute. His main research interests are digital simulators and simulation methods for equipment evaluation and testing as well as application of intelligent methods to control, protection and power quality monitoring. Dr. Kezunovic is a registered professional engineer in Texas, and a Fellow of the IEEE.