보호계전기 정정 협조 전문가 시스템

°이 승재 김 기화 조용익 — 명지대학교 전기공학과
윤 상현 — 프로컴 시스템
윤 만철 이상옥 — 한전 기술 연구원

AN EXPERT SYSTEM FOR SETTING AND COORDINATION OF PROTECTIVE RELAYS

Seung Jae Lee Ki Hwa Kim Young Ik Cho — Myong Ji University
Sang Hyun Yoon — PROCOM Systems
Man Chul Yoon Sang Ok Lee — Research Center, KEPCO

Abstract - This paper describes enhancements made on PROSET, an expert system which can perform setting and coordination of protective relays used in the ultra-high voltage transmission systems. Enhancements include the friendly and convenient environment for rulebase management and system manipulation, expanded setting capability, and faster processing speed which have been achieved through adoption of the new rule representation, rule order independent IE, rulebase editor, local database generator, interface to PSS/E fault program, RB expansion, etc.

Keywords: expert system, protection system, computer-aided design, transmission systems

INTRODUCTION

As the modern power system becomes more complex and larger and the voltage level continues to go up, the impact of a fault becomes higher. The protective relays are responsible for sensing the fault and tripping the breakers to isolate the fault. The effect of the fault can be minimized by proper coordination of those relays. Security and reliability of the power system are mainly governed by the level of accuracy and coordination of protection system which consists of various relays, circuit breakers and other auxiliary devices.

Recognizing the importance of the well-coordinated protection systems, computer programs to set the protective relays and to secure their coordination have been developed [1,2,3] and they have been very helpful to the relay engineers. However, in view of the fact that engineer's intuition and empirical knowledge play a very important role in the setting process, most of them, we believe, have not been effective in acheiving the best practical power due to the lack of ability of conventional languages in the symbolic processing.

A new approach applying the expert system techniques has been proposed by authors [4]. It is believed to have shown a good capability of incorporating the empirical knowledge, accomplishing the higher practicality. Although this work has been extended [5], it still needs improvement on various parts. This paper describes enhancements made to this system. The developed system named PROSET, runs on PC-AT/386 under DOS and has achieved a faster processing speed, more friendly user environment and interface to the fault program, etc. through introduction of various designs.

REVIEW OF PROSET [4,5]

PROSET is an expert system which can perform setting of various types of relays such as distance relays, overcurrent relays, reclosing relays, etc., currently used in KEPCO 154 KV transmission systems. It has adopted the frame representation to achieve the efficient data management and the production rule for representation of setting knowledge. PROSET has been implemented in PROLOG that runs on PC under MS-DOS and has a modular structure consisting of database and four modules (setting, evaluation, adjust, output) with stand-alone execution capability as shown in Fig. 1. A brief explanation of each module is given in the following.

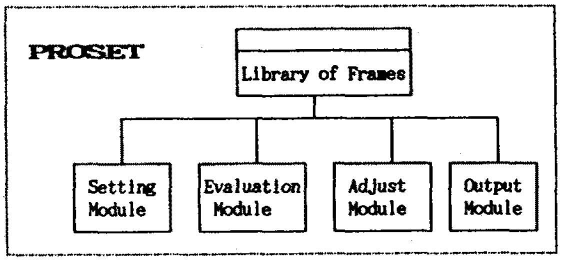

Fig.1 Structure of PROSET

Library of frames (LF) is a database storing all necessary data represented in frame which provides a strong structuredness. Four kinds of frames - relay, line, relay panel, fault - constitutes LF and Access to LF is easily obtained through the predefined predicate Get_Attribute_Value owing to the uniform data structure. Setting module performs the preliminary setting by applying the heuristic setting rules adopted by the relay engineers. It has a capability of handling various relays comprising the 154 KV T/L protection panels such as MZS from MITSUBISHI, CXS from TOSHIBA, GCX from GE.
The function of evaluation module is to check whether the preliminary settings contain any miscoordination problems such as zone 2 or zone 3 overlapping in their reaches. Checking is performed based on diagnosis rules whose condition part contains type information of the primary and backup relay pair and conclusion part has a simple but accurate coordination conditions. If any miscoordination exists, then adjust module attempts to find the possible solution to correct settings of the miscoordinated pair of relays using the heuristic correction rules. Correction is made based on the reach

and time delay change of the backup distance relay. Once setting job is done, output module can be used to show the results to the user in a very friendly way using the graphics. Also it can give the tabular presentation of the data frames in the library of frames.

STRUCTURE OF ENHANCED SYSTEM

PROSET has been in test use by KEPCO relay engineers and it has attained a very favorable response by showing a successful performance. However, it still need further works for a better practical use, which have been identified through discussions with relay engineers and development experience like follows:

- Efficient and more friendly database management to expedite the data-in process and run-time speed
- Rulebase management environment which requires little knowledge on PROLOG language to add or change rules
- RB expansion to cover rules for protection panels of 345 KV T/L, Transformers, Bus, etc.
- Well-organized step report and setting report for permanant keeping
- Interface between PC and IBM mainframe computer for fault data transfer
- Enhancement of parameter calculation part for easier expansion

Most of those identified requirements have been realized through various designs such as local database, new rule representation and inference engine, rulebase editor, communication link, etc., resulting in the new system structure shown in Fig. 2 Details of newly added parts to PROSET are described in the following sections.

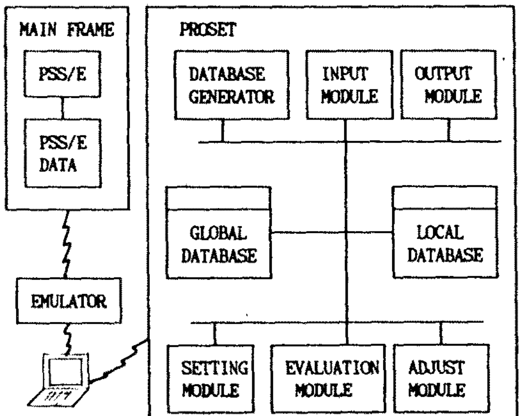

Fig.2 Structure of Enhanced PROSET

DESCRIPTION OF ENHANCEMENTS

DB Generator

Since LF, PROSET's database, is supposed to contain all relevant information on the system, protection panels, relays, faults, etc., it grows very rapidly as the system does. Therefore the data access to LF would take some time in a large system and consequently the setting process would consume quite amount of time even in case of setting single or a couple of panels (which is often the case in practice), for which in fact, only the small portion of the database is used.

Speed-up of data access has been accomplished by introduction of DB generator which has a capability of generating the local database containing the reduced set of data necessary for setting panels of concern. DB generator identifies the area beyond which settings of panels to be set are expected to have some effect as far as the protection is concerned. This study assumes the area's boundary specified by buses which are two-buses away from the panel location in the forward direction and one-bus away in the backward direction for the preliminary setting process as illustrated in Fig. 3, while four-bus-wide area in both direction is used for coordination checking.

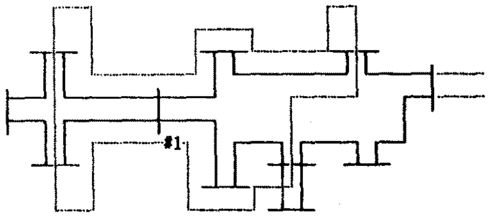

Fig. 3 Area affecting Panel #1 Setting

Then the small database called "local database" is constructed with the information associated with this area. Instead of dealing with the large database, the new system uses this local database achieving reduction of data retrieval time. Note that further saving is attained owing to the use of the working frame containing the frequently-accessed information such as CT, PT,ratios, relay specifications during the run time [4].

Rule-Base Management

Modification of the rulebase requires user's knowledge on PROLOG language since PROLOG syntax is used to represent the setting knowledge. Two-level structured-RB (one containing mainly judgmental knowledge and the other containing mainly the procedural knowledge) sometimes causes ambiguity between 1st- and 2nd-level rules and decreases the readability although it gives the conceptual correspondence to the existing form of setting knowledge [7]. If more than two rules can be applied to the given situation, the adopted PROLOG's built-in inference mechanism selects and executes the first appearing rule in the rulebase. Due to this order-dependent conflict resolution scheme, insertion of a rule requires a careful inspection of relative rule ordering and thus the RB maintenance becomes tough as the RB grows. Thus it is strongly needed to have an easy environment for the rulebase management, so that the user can easily make modification of RB. For this, a new representation of the setting knowledge and the corresponding inference engine have been developed and the rulebase editor is introduced. Details are explained in the following.

1. Rule Representation

Consider a rule to set a zone 2 unit of distance relay which is located on the end line. The old expression for this rule would be

```
level-1: identification(Rule,Explanation) :-
        /* if  */
              class_is("DZR") and
              function_is("zone3"),!,
        /* then */
              Rule = "zone3p" and
        Explanation = "set at smaller of primary plus
                       125% forward longest times Kapp
                       and load ability impedance"
```

```
level-2: zone3p("R1") :-
        /* if  */
            unit_is("reactance_type") and
            on_end_line,!,
        /* then */
            load_ability_impedance(Value1),
            line_X_225pct(Value2),
            smaller_of(Value1,Value2,X1),
            calculate_Zr_and_PctTap(X1).
```

Note that two conditions in level-2 have different characteristics: the first one affects the way of numerical calculation (i.e., line reactance to be used in reach calculation) and the second one changes the way to set causing the discrepancy between the rule explanation of the level-1 and the action of the level-2. Therefore it is desirable, we believe, to put the judgmental information which decides the way to set into the single level and the numerical calculation part which can hardly be called "knowledge" in the conventional procedural form. By doing so, a clear distinction between the judgmental knowledge and numerical processing as well as the improved readability of the rule can be obtained. For this the rule representation has been modified into the following form:

RULE(ref,conditions[C1,C2,...],conclusion[A]).

This representation has three fields - reference, condition, conclusion. 'ref' in the reference field denotes the rule reference number which is introduced for tracing of fired rules during the reasoning process.

The condition field contains a list of conditions integrated by conjunction, which are to be expressed using the reserved words as explained in Sec. 3. The conclusion part contains the purely descriptive expression of the pertinent setting method represented in a single string. Note that this representation has removed the explanation part appeared in the old one since it can be easily reconstructed from the rule itself and it gives enhanced readability of the rule. Following the new representation, the example rule can be expressed as follows:

```
rule(6, condition( [class("DZR"),
                    function("PhaseZone3&CRtrip"),
                    located("on_end_line"))],
      conclusion("set at smaller of 225% line impedance
                  and load ability impedance"))
```

Execution of the conclusion is carried out by the separately defined procedure which has a basic form of "EXECUTE(A) :- procedure1, procedure 2,...". Thus the procedural part corresponding to the example is defined in the following EXECUTE procedure.

```
EXECUTE("set at smaller of 225% line impedance and load
        ability impedance")) :-
            unit_is("reactance_type"),
            line_reactance(X),
            Value1 = 2.25 * X,
            load_ability_impedance(Value2),
            smaller_of(Value1,Value2,X1),
            calculate_Zr(X1,Xr),
            calculate_parameters(Xr).
```

2. Inference Engine

Suppose one want to set the zone 2 unit of a distance relay which is located on the end-line. The following two pertinent rules can be found:

```
rule(11,conditions([class("DZR"),function("PhaseZone2")]),
       conclusion("set at larger of 125% line impedance
          and line impedance plus 25% forward
          shortest times Kapp"))
rule(12,conditions([class("DZR"),function("PhaseZone2"),
                    location("on end line")]),
       conclusion("set at 125% line impedance"))
```

The old IE based on PROLOG's pattern matching and backtracking will apply the rule 11 due to its precedency over rule 13 although rule 13 is the proper one in this case. This problem caused by dependency on rule ordering has been resolved by introducing the IE adopting the specificity-based conflict resolution scheme, which is to select the most specific rule among the set of candidate rules. The newly devised inference procedure consists of three steps - (1) Conflict Set Generation (2) Conflict Resolution (3) Rule Execution - which is the typical forward chaining inference mechanism. The first step produces "Conflict Set" which contains rules with their condition parts satisfied. In the second step, the rule with the most conditions is selected as the most proper rule (this is called "specificity ordering"). The last step performs the rule firing which executes the setting procedure containing the numerical processing corresponding to the conclusion part of the selected rule.

In the above example, rules 11, 12 will form the Conflict Set and the new IE will select rule 12 which has one more condition than rule 11, and then execution of its conclusion part will be carried out through the corresponding procedure EXECUTE("set at 225% line impedance"). Thus rule-order dependency has been eliminated and this makes it easier to manage the rulebase.

3. Rulebase Editor

The newly adopted rule representation and inference engine have enabled the development of the rulebase editor. Developed editor provides an easy environment of rule modification to the user with little knowledge of PROLOG through predefined predicates for most commonly used functions. It consists of two parts - condition editor and conclusion editor.

Condition Editor: Noting only a few pieces of judgmental information constitute the condition part of setting rules such as class and function of the relay [7], most commonly appearing predicates have been predefined and used as reserved words in the editor. Those words together with their possible arguments are shown in Table 1.

Table 1. Example of Reserved Words

Reserved Words	Arguments
class	DZR, DGZR, OCR, OCGR, DOCGR, SCR, RCR, UVR
function	PhaseZone1,PhaseFaultDetectionForCRstarting
	GroundFaultDetectionForSupervising, etc.
unit	forward,backward
location	on end line, not on end line
unit type	mho type, reactance type, lens type

Conclusion Editor: The conclusion part as explained earlier is supposed to contain a single-string expression of "how to set" and its execution is handled by the separately defined procedure. Thus it requires to put the appropriate expression in a purely symbolic form and to program the corresponding EXECUTE procedure. In order to facilitate this process, like the condition editor, some predicates have been predefined (Table 2). Thus with knowledge of those words and basic arithemetic operators (*,/,+,-,etc.), one can build the conclusion part without difficulty. Majority of the predefined predicates are related with the data retrieval from the working frame or the local database. Although these functions can be programmed using the basic data access predicate, get_Attribute_Value, they have been introduced to save the programming effort.

Table 2. Part of Predefined Predicates

Reserved Predicates	Function
Get_Attribute_Value	data access to the frame
Line_Impedance	impedance of relay-located line
Load_Ability_Impedance	worst load impedance seen from the relay location
Calculate_Zr	impedance seen by the relay
Get_3P_Fault_Data	3 phase fault current

4. PSS/E Interface

Since PROSET runs on PC and PSS/E (a fault program developed by PTI and currently used in KEPCO) resides on HO-NAS XL50 mainframe computer, in order to obtain the necessary fault data for setting, one has to run PSS/E on the mainframe first and identify the required data from the output data file containing a lot of comments and redundant data. In the next step, he has to manually input the needed data into the PROSET database. Consequently this whole data logging process becomes prone to errors, very time-consuming and tedious.

The new system provides an environment in which PSS/E can be easily accessed and its generated fault data can be automatically converted into the PROSET-required form and saved in the PROSET database. This has been achieved through the communication emulator and the data converter. Emulator provides the communication link between PC and the host computer, and data converter written in C language has a capability of interpreting the output file transmitted from the mainframe computer and generating the datafile containing fault data in a recognizable form by PROSET. Fig. 4 illustrates the PSS/E generated output and its converted form produced by data converter.

```
.....................................................................
.                    PSS/E SHORT CIRCUIT OUTPUT                     .
.           KEPCO POWER SYSTEM SIMULATE FOR RELAY SETTING           .
.    DATA CHECKED BY SYSTEM PROTECTION DIVISION ( MAXIMUM CONDITION ) .
.                                                                   .
.            *** FAULTED BUS IS : 3260 (YESAN SS 154) ***           .
.    WED, MAR 27 1991 15:48  HOME BUS IS : 3260 (YESAN SS  154)     .
.                          0 LEVELS AWAY                            .
.....................................................................

 AT BUS 3260 (YESAN SS 154) AREA 3 (PU) V+:/0.0000/0.00 (PU)VA:/0.0000/0.0
        V0:/0.5589/ 187.84  V+:/0.7780/ -0.80 V-:/0.2193/ -176.31

THEV.R,X,X/R:POSITIVE 0.00364 0.02942 8.068 NEGATIVE 0.00362 0.02904 8.028
                ZERO      0.01314 0.07341 5.588
                                          THREE PHASE FAULT
------- FROM -------AREA CKT  I/Z     /I+/     AN(I+)   /Z+/    AN(Z+)  APP X/R
 3240 (CHONANSS 154)  3   1  PU/PU  7.1278  -80.15  0.0464  79.89    5.606
 3250 (ONYANGSS 154)  3   1  PU/PU  6.5697  -80.06  0.0470  79.84    5.578
 3270 (HONGSONG 154)  3   1  PU/PU 10.0348  -84.88  0.0279  82.88    7.988
 3270 (HONGSONG 154)  3   2  PU/PU 10.0348  -84.88  0.0297  82.88    7.968
      TOTAL FAULT CURRENT (P.U.)  33.7384  -82.95

                                     ONE  PHASE FAULT
                          /I0/    AN(I0)   /Z0/    AN(Z0)   APP X/R
                         1.2705  -78.84  0.2158  -101.72    4.618
                         1.2094  -79.04  0.2976  -101.72    4.520
                         1.6014  -77.13  0.1598  -101.70    4.630
                         1.6014  -77.13  0.1598  -101.70    4.630
                                   7.4946     -81.21
```

Fig 4. PSS/E Generated Output and Converted Form

5. Miscellaneous Extensions

In the ultra-high voltage transmission systems (154 and 345 KV T/L) of KEPCO, various protection panels from various manufacturers are being used. Only a part of these panels can be handled by the old system. This limited capability has been extended by expansion of RB of setting module and corresponding procedures to cover CXS (TOSHIBA), GCX (GE), MZS (MITSUBISHI), K-DAR, L-DAR (WH), XCG (GEC) used for 154 KV T/L protection.

Documentation is another important issue to the relay engineers and thus it is essential to present setting results and intermediate setting steps in a well-organized form. Through discussions with KEPCO relay engineers, some specific forms of presentation have been established. As an example, Fig. 5 and 6 illustrate the part of the setting output and step report respectively.

```
****************************************
*         >>  SDJ#2   PANEL  <<        *
*                                      *
* REMOTE PANEL :     SOC#1             *
* CT = [ 800, 5]  PT = [154000, 110]   *
****************************************

** RELAY TYPE : MZS1D
** FUNCTION : Impedance relay for phase fault
-----------------------------------------------------------------------
unit    % tap   basic_tap   1st imp   2ry imp   timer   max_ang
01      50.00   0.25        4.35      0.50      0.00    90.0
02      14.00   0.25        16.05     1.83      0.33    90.0
SU      46.00   2.00        38.19     4.36      1.67    60.0
-----------------------------------------------------------------------

** RELAY TYPE : MZS2D
** FUNCTION : Impedance relay for carrier starting
-----------------------------------------------------------------------
unit      % tap   basic_tap   1st imp   2ry imp   timer   max_ang
forward   57.00   3.00        46.11     5.27              75.0
backward  20.00   3.00                  1.00              255.0
-----------------------------------------------------------------------

** RELAY TYPE : MZS3D
** FUNCTION : OutOfStepBlocking
-----------------------------------------------------------------------
unit      % tap   basic_tap   1st imp   2ry imp   timer   max_ang
forward   47.00   3.00        55.69     6.36              80.0
backward  40.00   3.00                  2.00              240.0
-----------------------------------------------------------------------

** RELAY TYPE : L031B
** FUNCTION : PhaseFaultDetectionForSupervising
-----------------------------------------------------------------------
  tap = 2.39
-----------------------------------------------------------------------

** RELAY TYPE : L01C
** FUNCTION : GroundFaultDetectionForSupervising
-----------------------------------------------------------------------
  tap = 1.20
-----------------------------------------------------------------------
```

Fig. 5 Part of Setting Output

```
****>> Intermediate Step Report <<****
*     PANEL NAME  :     SDJ#2        *
*     CT / PT     =     0.11         *
**************************************

>>  Relay  :          MZS1D  <<

Unit   : 01

 85% line impedance :
   Z1 = j4.5883 * 0.85 / 0.422
      = j9.2418
   Zr = j9.2418 * CT / PT = 1.0562

Unit   : 02
```

```
1) 125% line impedance :

   Zl = j4.5883 * 1.25 / 0.422
      = j13.5909

2) 100% line impedance + 25% forward shortest times Kapp :

   Zl = (j4.5883 + j1.5960 * 0.5 * 2.1705 (Kapp) * 0.5) / 0.422
      = 12.9249

** select larger value of the two
   Zl = j13.5909
   Zr = j13.5909 * CT / PT = j1.5532
Unit  : SU

1) load ability impedance :
   Zl = 38.1904(60 )

2) 100% line impedance + 125% forward longest times Kapp :

   Zl = ((1.2358+j4.5883)+(0.4540+j1.8280) * 0.5
          * 2.1705 (Kapp) * 0.5) / 0.422
      = 5.8472+j22.6122
      = 23.3560 (75.5)
   ** into relay impedance
      => 24.2377 (60 )

   ** select smaller value of the two
   Zl = 24.2377(60 )
   Zr = 24.2377 * CT / PT = 2.7700(60 )


>>  Relay :          L031B <<

 1) 120% line current limit :
    1670.0 A * 1.2 = 2004.00
 2) 50% remote bus LL fault current :
    4067.51 A * 0.5 = 2033.76

    ** select the smaller of the two :
    pick up = 2004.00 / 160.00 (CT ratio) = 12.53
```

Fig. 6 Part of Step Report

VI. Conclusions

This paper describes an expert system which performs setting and coordination of protective relays used in the ultra-high voltage transmission systems. A newly adopted rule representation and conflict resolution of specificity-ordering have achieved the rule-order independent inference scheme and a friendly and convenient rulebase editor. Communication link between PC and the mainframe computer and data converter have been used to provide an automatic fault data logging. The processing speed has been improved by adoption of the local database. With the screen-based menu-driven manipulation and well-organized intermediate step report and setting report, the relay engineer can get the best benefit for the design of protection system.

The system needs some more works in achieving a better system-wide coordination and extended setting capability for various other relay panels for protection of buses, transformers, breaker failure, etc. Another important issue to be pursued in the next step is to link to DBMS (database management system) which provides a better control for data management.

REFERENCES

[1] H.Y.Tsien. "An Automatic Digital Computer Program for setting Transmission Line directional Overcurrent Relays", IEEE Tr. on PAS. Vol. 83, Oct. 1964, pp. 1048-1053.

[2] W.M. Thorn. et al., "A Computer Program for Setting Transmission Line Relay", Proc. American Power conference, Vol.35, Chicago, May 1973, pp. 1025-1034.

[3] R.B. Gastineua, et al., "Using the Computer to Set Transmission Line Phase Distance and Ground Backup Relays." IEEE Tr. on PAS, Vol. 96, NO.2, March/April 1977, pp.478-484.

[4] Seung Jae Lee, et al.(1989). An Expert System for Protective Relay Setting of Transmission Systems. IEEE Tr. Power Delivery, Vol.5, No.2, April, 1990, pp.1202-1208,

[5] Seung Jae Lee, et al, "PROSET : An Expert System for Protective Relay Setting", IFAC Symposium, Seoul, Korea, Aug. 1989.

[6] D.D.Wolfgram, et al., Expert Systems for the Technical Proffessional. John Wiley and Sons, Inc. New York.1987

[7] F. Hays-Roth, et al., Building Expert Systems, Addison-Wesley Pub.

[8] System Protection Div., Korea Power Electric Cor., Guideline for Relay Setting.

[9] Seung Jae Lee, et al., "Rule-Based Coordination of Distance Relaying in High Voltage Transmission Systems", Journal of KIEE, Vol.4, No.1, 1991, pp.1-8.

[10] A. Barr and E.A. Feigenbaum, The Handbook of Artificial Intelligence. Vol.1, Heuristech Press, California, 1981.

[11] N.S. Lee, "Programming with P-Shell", IEEE EXPERT Summer, 1986.