ZESZYTY NAUKOWE
POLITECHNIKI ŚLĄSKIEJ 2003
TRANSPORT z.51, nr kol. 1608

III INTERNATIONAL CONFERENCE
TRANSPORT SYSTEMS TELEMATICS TST`03

*railway traffic, control software,
reliability, software safety*

Andrzej LEWIŃSKI[1]
Katarzyna TRZASKA[2]

THE SOFTWARE SAFETY OF RAILWAY CONTROL SYSTEM APPLIED TO CROSS LEVEL PROTECTION

This paper presents a problem of building the safety software of railway control system using theory of software correctness. The correctness of software has an important effect to performance the fail-safe systems, especially SIL4 (*safety integrity levels 4*). For example Authors present the railway control system applied to cross level protection.

BEZPIECZEŃSTWO OPROGRAMOWANIA SYSTEMU STEROWANIA RUCHEM KOLEJOWYM NA PRZYKŁADZIE SYSTEMU SAMOCZYNNEJ SYGNALIZACJI PRZEJAZDOWEJ

W referacie przedstawiono problemy budowy bezpiecznego oprogramowania aplikacji kolejowych przy wykorzystaniu teorii poprawności oprogramowania. Poprawne oprogramowanie ma bardzo istotny wpływ na działanie systemów uwarunkowanych bezpieczeństwem, zwłaszcza 4 poziomu bezpieczeństwa (SIL4). Jako przykład autorzy przedstawili uproszczony system samoczynnej sygnalizacji przejazdowej.

1. SSP (AUTOMATIC LEVEL CROSSING SIGNALS) SYSTEM SAFETY CONDITIONS

The schematic model of automatic level crossing signals is presented on Fig.1. It presents layout of train detection equipment and equipment affected by the system.

Symbols shown on the drawing:

- Circuits influencing the system at train passage (detectors): Cz1, Cz4 (switch-on detectors for appropriate travel direction), Cz3, Cz6 (switch-on detectors for inappropriate travel direction), Cz2, Cz5 (switch-off detector for appropriate or inappropriate travel direction),

[1] Faculty of Transport, Radom University of Technology, 26-600 Radom, lewinski@kiux.man.radom.pl
[2] Chair of Automatic Control in Transport, Faculty of Transport, Silesian University of Technology, Krasińskiego 8, 40-019 Katowice, Poland, ktrzaska@polsl.katowice.pl

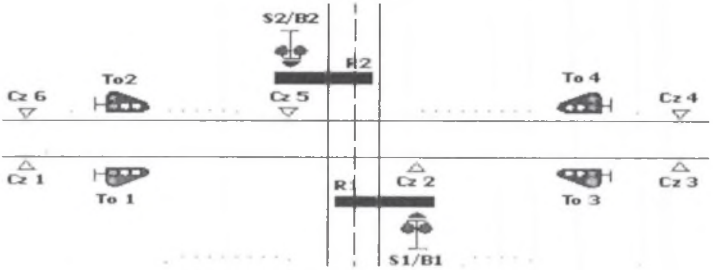


Fig.1. Schematic model of an automatic level crossing signal

- Circuits affected (controlled) by the system: S1, S2 (road light signals for warning the road users against a hazard caused by passing train), B1, B2 (acoustic signals for warning the road users against a hazard caused by passing train), R1, R2 (barrier drives for protection of level crossing) and To1, To2, To3, To4 (crossing warning disks for the train drivers).

The system has to ensure switching on of warnings at the moment of train's approach to the level crossing according to the following principles:

- At the moment when a railway vehicle enters the detector influence zone Cz1/Cz4 (depending of direction), signal of this detector is transferred to the control system, causing switching on of warning for the driver on the crossing warning disks (which enables the train passage with a normal speed), road light and acoustic signals. After the preliminary warning, the lowering of barrier rods begins, closing the road,
- At the moment of train's arrival to the level crossing, when the vehicle enters the influence zone of the switch-on detector Cz2/Cz5 (depending of direction) and its last axle leaves the output zone of this detector, the warning equipment is switched off (light and acoustic signals, crossing warning disks, gate barriers).
- Detectors Cz3/Cz6 are designed for detection of railway vehicle travelling in an incorrect direction of movement (they are applied because of a possibility of bidirectional movement). These detectors should not cause switching on of warnings by the vehicles travelling on an appropriate tracks.
- Warning should be sustained in the case when two railway vehicles travel on the same track one after another
- Warning should be sustained in the case when several railway vehicles travel on both tracks in the zone of influence of level crossing signalling devices.

The signalling system has 2 statuses of correct operation waiting status (when none of the signal switching on causes exists) and warning status (when the detectors detect at least one train). Other conditions of the device pertain to emergency situations defined in the software safety requirements (item 2)

All device statuses correct by default have been defined and written in a permanent manner (with a possibility of change) into the input status table *TabWejsc* (this applies to the detectors and traffic situation) and output table *TabWyjsc* (applies to the statuses of warning devices). The variables in the tables are of logic type and assume always one of two values true or false. Other statuses are treated as malfunction conditions and generate an appropriate reaction of output devices.

2. SOFTWARE MODEL

2.1. DEFINITION OF INPUTS AND OUTPUTS

Fig.2. presents correct operating statuses of level crossing signal together with corresponding statuses of warning equipment. Subsequent elements of the vector correspond to the following variables.

TabWejsc:(Tor1Ruch, Tor1Kier, Cz1, Cz2, Cz3, Tor2Ruch, Tor2Kier, Cz4, Cz5, Cz6),
TabWyjsc:(To1b, To1p, To2b, To2p, To3b, To3p, To4b, To4p, Sygnaly)

Variables *ToriRuch* inform whether there is movement on the track in question, variables *ToriKier* determine whether the direction of movement is appropriate or not, other variables *Czi* are responsible for information about occupancy of the zone of an and-nd vehicle presence detector.

Variables *Toib* and *Toip* indicate appropriate level crossing warning disk and appropriate light chambers (b - white, p - orange). Variable *Sygnaly* informs about switching on the warnings on the level crossing.

We may remark, that the table treats as correct the statuses where two neighbor detectors (e.g. Cz1 and Cz2) assume true value. This corresponds to two situations (e.g. Cz1 and Cz2):

A correct sequence: train left the detector 1 and then entered detector 2, incorrect sequence: both detectors indicate occupancy.

In order to discern between correct and incorrect sequence and to guarantee the appropriate reaction of the system, as well as recognizing the direction of vehicle movement, each of the detectors was divided into two zones entrance and output. Besides that, an additional variable was introduced of array type *Licznik [and]*, that changes its status in the course of vehicle passing each zone of the detector. An incorrect traveling sequence through detectors Cz1 and Cz2 is recognized using variable *Licznik [1] and Licznik [2]*.

```
const
max = 64;
T = True;
F = False;
   {table of possible input statuses}
   TabWejsc: array [1..max, 0..9] of Boolean=
                ((F,F,F,F,F,F,F,F,F,F),
                 (F,F,F,F,F,F,T,F,F,F),
                 (F,F,F,F,F,T,F,F,F,F),
                 (F,F,F,F,F,T,F,F,F,T),
                 (F,F,F,F,F,T,T,F,F,F),
                                :
                                :


   {assignment of appropriate outputs to the
    input statuses}
   TabWyjsc: array[1..max, 0..8] of Boolean =
                 ((F,F,F,F,F,F,F,F,F),
                 (F,F,F,F,F,F,F,F,F),
                 (F,F,F,F,F,F,F,F,F),
                 (F,F,T,F,F,F,F,F,T),
                 (F,F,F,F,F,F,F,F,F),
                               :
                               :
```

Fig.2. Methods of writing the correct operating statuses of signaling with corresponding warning device statuses

2.2. INTERLOCKS IN THE SYSTEM

Specific elements of the vector *Licznik [and]* depend of each other as it is shown in the table 1. Symbol "M" describes conditions likely to occur in the system in correct operation, symbol "U" indicates malfunction statuses. Symbol „M/$_{Czi}$" applies to a situation where the status may occur upon condition that the variable describing the detector status is *Czi=True,* and e. the detector has operated beforehand.

Table 1

Dependecies between variable Licznik [i]

Value of variable Licznik[1]	0	1	2	3	4
Cz2=True	U/M	U	U	U	M
Value of variable Licznik[3]	0	1	2	3	4
Cz2=True	M	U	U	U	U/M

The dependencies between the variables have been implemented at several locations (modules) of the program. Additionally the system takes into account such dependencies as relationships between subsequent statuses of an and-nd vehicle detector. In the case when the sequence is not observed, the handling is that of situation deemed as emergency one.

```
                            ⋮
  Jest_Usterka := True;
    for and:=1 to max do
      if (TabWejsc[and,0] = Tor1Ruch) and
         (TabWejsc[and,1] = Tor1Kier) and
         (TabWejsc[and,2] = Cz1)      and
         (TabWejsc[and,3] = Cz2)      and
         (TabWejsc[and,4] = Cz3)      and
         (TabWejsc[and,5] = Tor2Ruch) and
         (TabWejsc[and,6] = Tor2Kier) and
         (TabWejsc[and,7] = Cz4)      and
         (TabWejsc[and,8] = Cz5)      and
         (TabWejsc[and,9] = Cz6)      then
         begin
           Form1.Wyjscia(TabWyjsc[and,0], TabWyjsc[and,1],
TabWyjsc[and,2],
                        TabWyjsc[and,3], TabWyjsc[and,4],
TabWyjsc[and,5],
                        TabWyjsc[and,6], TabWyjsc[and,7],
TabWyjsc[and,8]);
           if not Sekwencja_Czujnikow then Jest_Usterka := False;
           Break;
         end;
    if Jest_Usterka then Usterka_Czujniki
    else if ((Cz1 and Cz2) or (Cz2 and Cz3) or (Cz4 and Cz5)
    or (Cz6 and Cz5)) then Usterka_Czujniki;

    if Sekwencja_Liczniki then Usterka_Sekwencji;
                            ⋮
```

Fig.3. Fragment of code representation of the program related with malfunction handling

Fig.3. represents the fragment of program code representation corresponding to a decision block responsible for recognition of emergency situation.

2.3. EMERGENCY SITUATION HANDLING

System recognizes as malfunction the following detector statuses:

- If there's no traffic on the track and.e. *ToriRuch=False*, and any detector *Czi=True*,
- If on the track 1 and 2 simultaneously appears a sequence *Cz1, Cz2, Cz3=True* or *Cz4, Cz5, Cz6=True*,
- If sequences *Cz1, Cz3=True* and *Cz4, Cz6=True* appear simultaneously
- If detector 2 or 5 in the system signals occupancy (*Cz2 or Cz5=True*) without previous occupancy of respectively detector 1 or 3 and 4 and 6,
- If one of the following sequences appear without taking into account dependencies between variables Licznik[i}:
    - Cz1, Cz2=True,
    - Cz3, Cz2=True,
    - Cz4, Cz5=True,
    - Cz6, Cz4=True.

In addition, the system takes into account the malfunctions related with failure to keep the passing sequence by an i-nd detector. Only statuses possible to be realized have been taken into account – all others result in malfunction handling starting up.

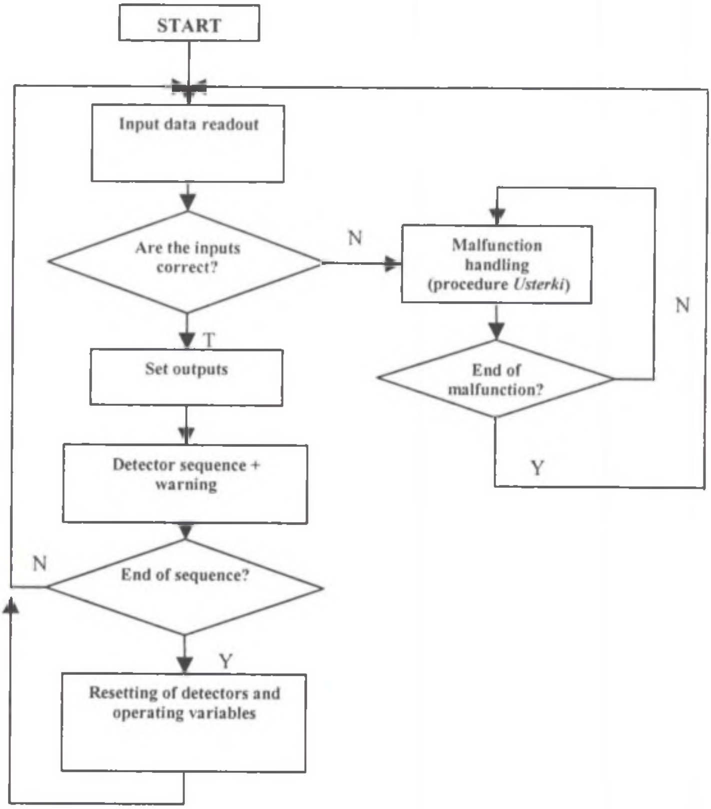
Reaction to each of these malfunctions consists in switching on the warning on the level crossing and orange lights on the crossing warning disks. If there is malfunction in a detector

2 or 5 then the warning disks signal the malfunction at the appropriate and inappropriate direction side. Warning is switched off only after removal of malfunction (resetting of the system).

In the case of appearance of malfunction, it is detected using the *TimerPoprawnosc* procedure that is called upon cyclically every 500μs. After recognition of malfunction status (not identified yet) the procedure hands over the controls to the appropriate control subroutine responsible for classification and appropriate reaction (such as lighting up appropriate warning disks).

Other malfunctions (such as warning device failures) have not been take into account in the system. They will be taken into account as work continuation.

3. A SIMPLIFIED ALHORITHM OF SYSTEM OPERATION


START
Input data readout
Are the inputs correct?
N
T
Malfunction handling (procedure *Usterki*)
End of malfunction?
N
Y
Set outputs
Detector sequence + warning
End of sequence?
N
Y
Resetting of detectors and operating variables

4. PRINCIPLES OF SAFETY SOFTWARE DESIGNING

Implementing new computer systems instead of the present systems (such as relay-based srk systems) it is necessary to assure at least similary level of functionality or safety in relation to the reliability and operation parameters. A specific feature of these new standards is division between hardware and software. Reliability analysis at the level of hardware is based on the methods known from the reliability theory, taking into account mainly structure of links between the subsystems, devices and system elements. While analysing the software for estimation of probability of a correct preparation of software, we have to use other methods related for example with Markow proces theory.

Designing of correct programs is related wit application of special procedures and methods of programming, such as systematic, structural, specified or defensive programming [1]. An important issue is to ensure both syntax and semantic correctness, Although syntax correctness is not a special issue because of strict rules governing the programming language, the semantic correctness is not as easily identifiable. Hence the numerous methods of proving the semantic correctness, most often based upon checking whether the program's behaviour is such as indicate conditions put on the program variables. This is a very important issue n the case of control systems related with safety, such as NSRK systems (Supervision and Control of Railway Traffic). According to UIC and CENELEC each stage of safety system software, starting from specification of requirements till the approval of software, should apply appropriate analyses and methods of creation of the software [4]. These elements are important especially in light of the future Poland's membership in European Union structures, where CENELEC reports and standards are in force.

BIBLIOGRAPHY

[1] LEWIŃSKI A.: Problemy oprogramowania bezpiecznych systemów komputerowych w zastosowaniach transportu kolejowego (Programming issues concerning safety computer systems in railway transport applications), Politechnika Radomska im. K. Pułaskiego, Monography No 49, Radom 2001.

[2] DEMBIŃSKI P., MAŁUSZYŃSKI J.: Matematyczne metody definiowania języków programowania (Mathematical methods of programming language definition), WNT, Warsaw, 1981.

[3] Opracowanie CNTK, Zakład Sterowania Ruchem Kolejowym: Wymagania bezpieczeństwa dla urządzeń sterowania ruchem kolejowym (Safety requirements for railway traffic control equipment), Warsaw, Feb. 1998.

[4] CENELEC EN 50128, Railway Applications, Software for railway control and protection systems, 1997.

[5] MYERS G. J.: Projektowanie niezawodnego oprogramowania (Desing of reliable software), WNT, Warsaw 1980.

[6] DĄBROWA-BAJON M.: Podstawy sterowania ruchem kolejowym. Funkcje, wymagania, zarys techniki (Basics of railway traffic control. Functions, requirements, outline of technique), WPW 2002.

[7] KOPETZ H.: Niezawodność oprogramowania (Software reliability), WNT, Warsaw 1980.

Reviewer: Ph. D. Jerzy Mikulski