

Zbigniew Łukasik
Politechnika Radomska

Waldemar Nowakowski
Z.A. KOMBUD S.A.

AKTYWNE TESTOWANIE SYSTEMÓW STEROWANIA RUCHEM KOLEJOWYM

Streszczenie: W artykule poruszono problem aktywnego testowania komputerowych systemów sterowania ruchem kolejowym przy wykorzystaniu języka TTCN-3. Czynne wykrywanie błędów w czasie działania systemów może być jedną z metod prowadzących do zapewnienia przez systemy bezpieczeństwa funkcjonalnego.

Słowa kluczowe: TTCN-3, testy aktywne, systemy srk.

1. WPROWADZENIE

Współczesne systemy sterowania ruchem kolejowym (srk) powszechnie wykorzystują technologie sieciowe tworząc struktury rozproszone. Nowego znaczenia nabiera w takim przypadku zagadnienie testowania systemów.

Dla systemów srk istnieje szereg wymagań, które muszą zostać spełnione, aby zapewnić wymagany poziom nienaruszalności bezpieczeństwa [7, 8]. Testowanie systemów jest tu głównie rozumiane w kontekście:

- walidacji (ang. *validation*), czyli testowania zgodności systemu z oczekiwaniami użytkowników,
- weryfikacji (ang. *verification*), czyli sprawdzenia zgodności systemu ze specyfikacją.

Można uznać, że takie działania prowadzą głównie do zapobiegania błędom, czyli eliminacji przyczyn ich powstawania jeszcze przed zastosowaniem systemu. Mówimy wówczas o tzw. testowaniu biernym, które nie zapewnia kontroli pracy systemu w czasie jego eksploatacji. W przypadku systemów złożonych, posiadających strukturę rozproszoną, dodatkowym przedsięwzięciem powinno być wdrożenie testowania aktywnego, prowadzącego do wykrywania błędów w czasie użytkowania systemu. W przypadku systemów automatyki transportu kolejowego, w których wymiana danych

realizowana jest z wykorzystaniem sieci komputerowych, jako język testów może posłużyć TTCN-3. Pozwala on na zbudowanie systemu testów i czynną kontrolę systemów rozproszonych.

2. JĘZYK TTCN-3

TTCN-3 (ang. *Testing and Test Control Notation*) jest językiem testów poddanym standaryzacji przez ETSI (ang. *European Telecommunications Standards Institute*). Pierwsza specyfikacja języka TTCN została zdefiniowana w roku 1992. Kolejne wersje standardu to TTCN-2 opracowany w roku 1997 i TTCN-3 obowiązujący od roku 2000. Aktualnie standard języka TTCN-3 obejmuje m.in. następujące dokumenty [4]:

- (ES 201 873-1) *Core Language (CL)*,
- (ES 201 873-2) *Tabular Presentation format (TFT)*,
- (ES 201 873-3) *Graphical Presentation Format (GFT)*,
- (ES 201 873-4) *Operational Semantics*,
- (ES 201 873-5) *Runtime Interface (TRI)*,
- (ES 201 873-6) *Control Interface (TCI)*,
- (ES 201 873-7) *The use of ASN 1*,
- (ES 201 873-8) *The IDL to TTCN-3 Mapping*,
- (ES 201 873-9) *Using XML schema with TTCN-3*,
- (ES 201 873-10) *Documentation Comment Specification*.

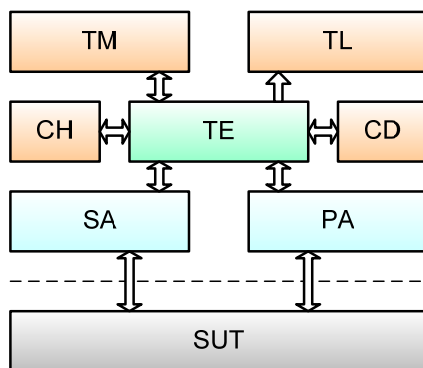
Powstanie języka TTCN było związane z potrzebą testowania protokołów telekomunikacyjnych. Rozwój tego standardu zmierza jednak w kierunku opracowania uniwersalnego języka testów. Język TTCN-3 pozwala obecnie na testowanie różnych istotnych cech systemów tj. wydajność (ang. *performance*), własności czasu rzeczywistego (ang. *real-time*), testowanie współpracy (ang. *interoperability*), wykrywanie ataków sieciowych [2]. Znalazł on zastosowanie w wielu systemach telekomunikacyjnych np.: SIP (ang. *Session Initiation Protocol*), WiMAX (ang. *Worldwide Interoperability for Microwave Access*), DSRC (ang. *Dedicated Short-Range Communications*). Aktualnie trwają prace, w ramach projektu TT-Medal, mające na celu wykorzystanie języka TTCN-3 do testowania systemów sterowania ruchem kolejowym i opracowanie standardowych testów dla komputerowych systemów nastawczych [1, 3, 5].

Architektura systemu testów, w przypadku środowiska TTCN-3, jest zestandaryzowana i obejmuje następujące elementy (ang. *entities*):

- TE – *TTCN-3 Executable* – odpowiada za wykonanie testów,
- TM – *Test Management* – odpowiada za zarządzanie systemem testów,
- TL – *Test Logging* – rejestruje zdarzenia wykonania testów i ich wyniki,
- CD – *Coding/Decoding* – odpowiada za kodowanie/dekodowanie danych wymienianych w komunikacji z SUT,
- CH – *Component Handling* – odpowiada za synchronizację różnych elementów systemu testów umieszczonych na kilku terminalach,
- SA – *SUT Adaptor* – dostosowuje abstrakcyjny interfejs systemu testów (ATSI) do interfejsu rzeczywistego,

- PA – *Platform Adaptor* – dostosowuje TE do konkretnej platformy uruchomieniowej (np. realizacja *timer-ów* i ich współdziałanie z TE),
- SUT – *System Under Test* – testowany system.

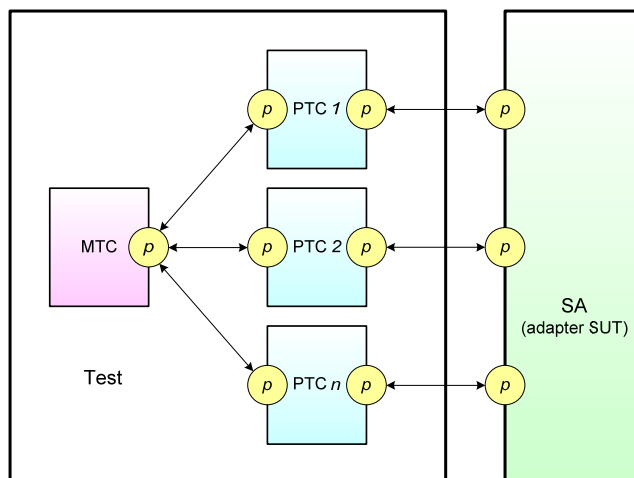
Architektura systemu testów została przedstawiona na rys 1.



Rys. 1. Podstawowe elementy systemu testów TTCN-3

TTCN-3 jest bardzo zbliżony do innych języków programowania. Podstawową jednostką zapisu w języku TTCN-3 jest moduł. Składa się on z części definiującej i sterującej. Część definiująca zawiera definicje komponentów testu, typów danych, stałych, szablonów, portów komunikacyjnych oraz funkcji. Część sterująca zawiera zapis wywołań poszczególnych testów oraz może zawierać definicje lokalnych zmiennych. Język TTCN-3 dopuszcza definiowanie struktur danych również w innych formatach tj. ASN.1, IDL, XML. Wyniki działania testów mogą być prezentowane w formie tekstowej, tabelarycznej lub graficznej.

TTCN-3 pozwala na definiowanie abstrakcyjnego systemu testowego, który zbudowany jest z komponentów testowych TC (ang. *Test Component*) [9]. Istnieje przy tym możliwość zdefiniowania jednego głównego komponentu MTC (ang. *Master Test Component*), jak również wielu komponentów równoległych PTC (ang. *Parallel Test Component*), współdziałających z komponentem MTC (rys. 2).

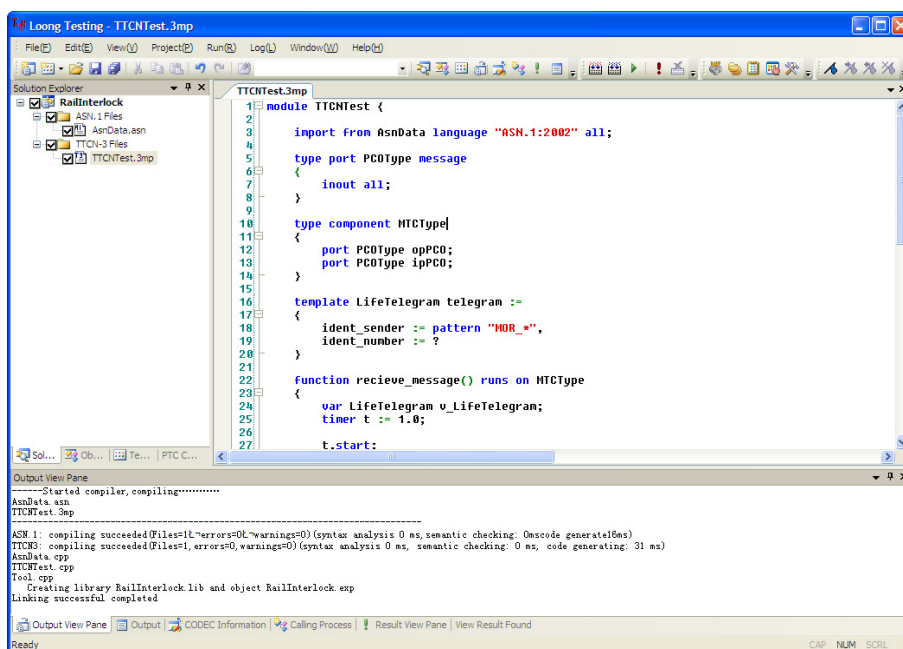


Rys. 2. Abstrakcyjny system testów TTCN-3

Z komponentem związane są porty „p” poprzez które komunikuje się on z systemem testowanym SUT lub innymi komponentami. Komponenty mogą być tworzone statycznie lub w wyniku dynamicznego kreowania.

3. AKTYWNE TESTY SYSTEMU SRK W JĘZYKU TTCN-3

W celu zbadania możliwości wykorzystania języka TTCN-3 jako narzędzia do aktywnego testowania komputerowych systemów srk wykonano prosty eksperyment w środowisku „Loong Testing” opracowanym przez USTC (ang. *University of Science and Technology of China*) (rys. 3).



Rys. 3. Środowisko „Loong Testing” języka TTCN-3

„Loong Testing” jest aplikacją posiadającą parser i kompilator modułów zapisanych w języku TTCN-3. Umożliwia definiowanie struktur danych nie tylko z wykorzystaniem TTCN-3, ale również w notacji ASN.1 (ang. *Abstract Syntax Notation One*). Wynikiem kompilacji testów są moduły programowe w języku C++ i odpowiadające im biblioteki DLL [6]. Istnieje dzięki temu możliwość bezpośredniego sprawdzenia poprawności działania testów zdefiniowanych w TTCN-3 i prezentacji wyników w formie tekstowej.

W ramach przeprowadzonego eksperymentu zbudowano prosty test w języku TTCN-3, w którym zasymulowano sprawdzanie aktywności komputerowego systemu nastawczego. Strukturę telegramu „*LifeTelegram*” w notacji ASN.1 przedstawiono na rys. 4.

```

AsnData DEFINITIONS ::=
BEGIN
    LifeTelegram ::= SEQUENCE {

```

```

        -- Interlocking data
        ident-sender  IA5String (SIZE(1..8)),
        ident-number  INTEGER (1..65535)
    }
END

```

Rys. 4. Struktura telegramu w notacji ASN.1

Telegramy mogą być zwrótnie przesyłane do systemu testów, który wykona ich analizę. Służy do tego celu lista alternatyw „alt”, zdefiniowana w funkcji „*receive_message()*” komponentu „*MTCType*”. Wymiana danych realizowana jest z użyciem portów „*opPCO*” i „*ipPCO*”. Zdarzenia wykonania testów i ich wyniki są rejestrowane w jednostce Test Logging (TL). Program testu został zamieszczony na rys. 5.

```

module TTCNTest {
    import from AsnData language "ASN.1:2002" all;
    type port PCOType message {
        inout all;
    }
    type component MTCType {
        port PCOType opPCO;
        port PCOType ipPCO;
    }
    template LifeTelegram telegram := {
        ident_sender := pattern "MOR_*",
        ident_number := ?
    }
    function recieve_message() runs on MTCType {
        var LifeTelegram v_LifeTelegram;
        timer t := 1.0;
        t.start;
        alt
        {
            ipPCO.receive(telegram)->value v_LifeTelegram {
                log(v_LifeTelegram);
                setverdict(pass)
            }
            [] ipPCO.receive {
                log("Failed telegram");
                setverdict(fail)
            }
            [] t.timeout {
                setverdict(fail)
            }
        }
    }
    testcase Test() runs on MTCType {
        connect(mtc:opPCO, mtc:ipPCO);
        // Example telegram
        telegram.ident_sender := "MOR_ROZ";
        telegram.ident_number := 1;
        opPCO.send(telegram);
        recieve_message();
    }
}

```

```

control {
    execute(Test());
}
}

```

Rys. 5. Przykładowy program testu w języku TTCN-3

4. PODSUMOWANIE

Wykorzystanie języka TTCN-3 do aktywnego testowania komputerowych systemów sterowania ruchem kolejowym może prowadzić do zwiększenia bezpieczeństwa funkcjonalnego systemów. Prace badawcze dotyczące tego zagadnienia prowadzone w ramach projektu TT-Medal znajdują się aktualnie w fazie wstępnej. Dlatego też wszelkie próby oceny możliwości wykorzystania języka TTCN-3, jako metody testowania systemów sterowania ruchem kolejowym, mogą przyczynić się do poszerzenia zakresu stosowania badanej technologii. Jednym z pośrednich celów poruszonej w pracy tematyki jest popularyzacja nowego obszaru wykorzystania języka TTCN-3 oraz ocena korzyści, jakie może przynieść wdrożenie tej metody testowania w przypadku komputerowych systemach sterowania ruchem kolejowym.

Bibliografia

1. Blom S., Ioustinova N., Van de Pol J., Rennoch A., Sidorova N.: Simulated Time for Testing Railway Interlockings with TTCN-3. 5th international workshop, FATES 2005, Edinburgh, UK, 2005.
2. Brzeziński K.M.: Język TTCN-3 a bierne wykrywanie ataków sieciowych. Poznańskie Warsztaty Telekomunikacyjne, Poznań 2006.
3. Calame J.R., Goga N., Ioustinova N., Van de Pol J.: TTCN-3 Testing of Hoorn-Kersenboogerd Railway Interlocking. In Proceedings of the Canadian Conference on Electrical and Computer Engineering, CCECE 2006, Ottawa, Canada, 2006.
4. <http://www.ttcn-3.org>
5. <http://www.tt-medal.org>
6. <http://ttcn.ustc.edu.cn>
7. PN-EN 50128:2002 - Zastosowania kolejowe. Łączność, sygnalizacja i systemy sterowania. Programy dla kolejowych systemów sterowania i zabezpieczania. PKN 2007.
8. PN-EN 50129:2007 -Zastosowania kolejowe. Systemy łączności, przetwarzania danych i sterowania ruchem –Elektroniczne systemy sterowania ruchem związane z bezpieczeństwem. PKN 2007.
9. Willcock C., Deiß T., Tobies S., Keil S., Engler F., Schulz S.: An Introduction to TTCN-3. John Wiley & Sons Ltd, New York, 2005.

ACTIVE TESTS FOR COMPUTER RAILWAY CONTROL SYSTEMS

Abstract: The article presents the problems of active tests for computer railway control systems with use TTCN-3, which is a flexible language applicable to the specification of all types of reactive system tests. Active testing of railway applications can guarantee functional safety of railway systems.

Keywords: TTCN-3, active test, computer railway control systems.