

Wojciech KULAS

Wojskowa Akademia Techniczna, Wydział Cybernetyki

ul. gen. Witolda Urbanowicza 2, 00-908 Warszawa 46

E-mail: wojciech.kulas@wat.edu.pl

**Emulatory systemów monitorowania
zagrożeń skażeniami na potrzeby
systemu WAZkA**

1 Wprowadzenie

W artykule przedstawiono koncepcję oraz elementy realizacji podsystemu emulatorów systemów monitorowania zagrożeń na potrzeby informatycznego systemu wsparcia analiz zagrożeń skażeniami CBRN (ang. *Chemical, Biological, Radiological, Nuclear*), prognozowania ich skutków i alarmowania ludności (pod nazwą WAZkA).

Głównym celem powstania podsystemu emulatorów jest wspomaganie procesu szkolenia osób uczestniczących w działaniu Krajowego Systemu Wykrywania Skażeń i Alarmowania (KSWSiA), które jak przewiduje się, będą przyszłymi użytkownikami systemu WAZkA. Ponadto emulatory mogą być wykorzystywane jako narzędzie wspomagające testowanie i opracowywanie procedur postępowania, w szczególności w sytuacjach alarmowych.

Emulatory zostały zaproponowane jako narzędzia czysto informatyczne, uruchamiane w tym samym lub takim samym środowisku sprzętowym co budowany system WAZkA. Jako metodę emulacji wybrano symulację konstruktywną, jednak zredukowaną do prostego, zdefiniowanego dostarczania danych, zgodnie z wcześniej przygotowanym scenariuszem.

W artykule skupiono się na trzech aspektach: (a) analizie potrzeb w stosunku do podsystemu emulatorów, (b) koncepcji rozwiązania powstałych problemów w celu dostarczenia zdefiniowanych potrzeb, (c) elementach informatycznej realizacji emulatorów. Potrzeby (a) zostały przedstawione przez określenie miejsca emulatorów w systemie WAZkA oraz w postaci ogólnego modelu wymagań wynikających z zamówienia na system oraz z analizy potrzeb przyszłych użytkowników. Koncepcja (b) została przedstawiona przez reprezentatywne elementy modelu oprogramowania. Dla potrzeb artykułu zostały przedstawione tylko kluczowe elementy tego modelu. Ponadto przedstawiono przykłady projektów emulatorów w postaci uproszczonego opisu modeli symulacyjnych systemów monitorowania zagrożeń. Realizacja (c) przedstawionych koncepcji została zilustrowana kilkoma kluczowymi rezultatami, uzyskanymi w wyniku uruchomienia zaimplementowanych modeli/algorytmów symulacji, których projekty zostały przedstawione w ramach części (b).

2 Analiza potrzeb

Prezentowane emulatory dedykowane są systemowi WAZkA. Jest to system informatycznego wsparcia analiz zagrożeń skażeniami CBRN, prognozowania ich skutków i alarmowania ludności (WAZkA to akronim: Wsparcia Analiz Zagrożeń

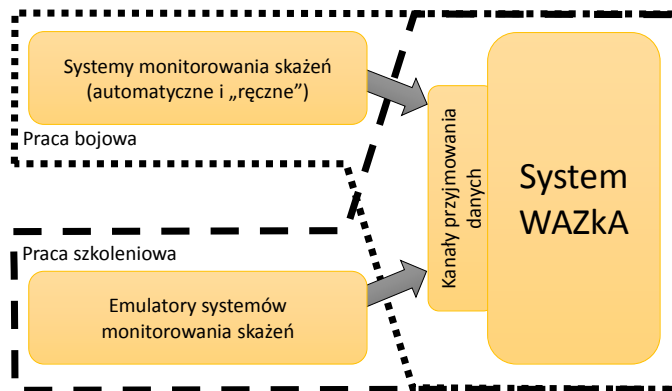
skażeniami i Alarmowania). Powstał on jako wynik realizacji projektu naukowo-wdrożeniowego. Jednym z celów powstania tego systemu jest wsparcie procesu szkolenia członków Krajowego Systemu Wykrywania Skażeń i Alarmowania.

Wykorzystanie systemu do szkolenia opiera się na koncepcji użycia dokładnie takiego samego systemu jak do pracy „bojowej”. Wymaga to jednak zmiany całego środowiska, w jakim ten system będzie pracował w trakcie szkolenia. Elementami środowiska są:

- źródła danych o skażeniach,
- odbiorcy komunikatów i ostrzeżeń,
- bazy danych.

Wymiana źródeł i odbiorców zazwyczaj sprowadza się do zastąpienia rzeczywistych elementów obsługą szkolenia (tzw. podgrywką). Takie prowadzenie szkolenia jest zazwyczaj kłopotliwe do zorganizowania i kosztowne, ponieważ potrzebna jest duża liczba członków obsługi szkolenia. Stąd pojawia się potrzeba zastąpienie części personelu (lub nawet wszystkich) przez specjalizowany podsystem informatyczny. Najprościej jest to zrobić w obszarze automatycznych lub zautomatyzowanych źródeł danych.

System zachowujący się tak jak system rzeczywisty, jednak wykonany w innej technologii i w innym środowisku, nazywany jest emulatorem. Emulator zachowuje się tak jak system rzeczywisty tylko z określonego punktu widzenia, zależnego od celu wykorzystania emulatora. Ponieważ emulator „widziany” jest tylko przez system WAZkA, musi on być zgodny z systemem zastępowanym na poziomie interfejsu. Miejsce podsystemu emulatorów względem systemu WAZkA przedstawia rysunek 1. Linia kropkową zaznaczono konfigurację do standardowej pracy „bojowej”, a linią przerywaną konfigurację do pracy szkoleniowej.



Rys. 1. Miejsce podsystemu emulatorów względem systemu WAZkA

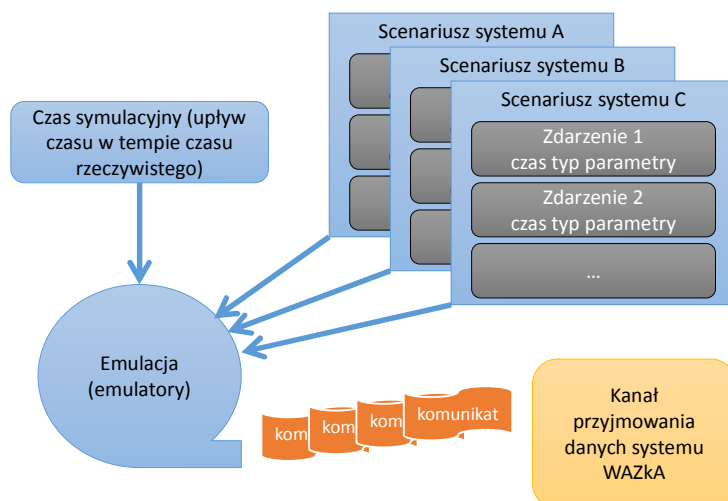
Fig. 1. Place of the emulator subsystem relative to the WAZkA system

Model zachowania się emulatora jest zależny od wymaganego stopnia adekwatności. W opisywanym zastosowaniu emulatorzy nie muszą się zachowywać identycznie jak rzeczywiste zjawiska i systemy je śledzące. Nie muszą i wręcz nie powinny dostarczać

danych najbardziej prawdopodobnych. Wystarczy, że dostarczają dane prawdopodobne. Dla celów dydaktycznych znacznie bardziej wartościowe jest szkolenie w podejmowaniu decyzji w nowych warunkach, a nie tylko w typowych. Z tych przyczyn wynika wniosek, że modele zachowania się emulatorów mogą być dość uproszczone, co znacznie ułatwia i przyspiesza ich wykonanie, a co za tym idzie, obniża koszty.

Najbardziej oczywiste zastosowanie emulatorów to „udawanie” automatycznych źródeł danych o skażeniach. Jednak możliwe jest również ograniczone zastosowanie do źródeł nieautomatyzowanych, np. w zakresie przesyłania niesformalizowanych informacji pocztą elektroniczną.

Naturalnym sposobem wykonania emulatorów jest zastosowanie tych samych technik i środowiska, co wykonanie całego systemu WAZkA. Konsekwentnie jako metodę emulacji najwygodniej jest przyjąć metodę dyskretnej symulacji komputerowej. Schemat działania takiego rozwiązania przedstawiony został na rysunku 2.



Rys. 2. Schemat działania podsystemu emulatorów

Fig. 2. The operating scheme of the emulator subsystem

Wszystkie emulatory, działające zgodnie z upływem czasu symulacyjnego, sterowane są danymi zawartymi w scenariuszach. Scenariusze są wcześniej przygotowane przez scenarzystę, analityka lub kierownika ćwiczeń. Emulatory, w wyniku interpretacji opisu zdarzeń zawartych w scenariuszach, generują ciąg komunikatów, które dostarczane są standardowym kanałem komunikacyjnym do systemu WAZkA.

Powyższe rozważania można podsumować następującymi syntetycznymi wnioskami:

- ponieważ system WAZkA jest systemem czysto informatycznym – podsystem emulatorów też będzie czysto informatyczny;
- z powyższego wynika, że dobrym rozwiązaniem może być zastosowanie technik symulacji komputerowej;

- rzeczywiste zachowanie się emulowanych systemów (nie zjawisk) ma charakter zdarzeniowy, więc taka postać symulacji komputerowej jest preferowana;
- stopień adekwatności nie musi być duży, wobec czego można przyjąć bardzo uproszczone modele emulowanych systemów i zjawisk;
- należy przyjąć budowanie emulatorów od najprostszych (celem jest obniżanie ryzyka wytworzenia całego systemu, a nie podsystemu emulatorów);
- podsystem emulatorów powinien być całkowicie przezroczysty dla systemu – powinien powstać jako niezależne oprogramowanie integrujące się z systemem przez te same interfejsy co rzeczywiste otoczenie;
- emulatory są tylko producentem danych, a nie konsumentem; można rozważyć przyszłe rozszerzenie o emulatory konsumentów danych;
- symulacja powinna odbywać się na bazie przygotowanego scenariusza, modyfikacja scenariusza w trakcie emulacji jest opcją do rozważenia.

3 Koncepcja rozwiązania

Ostateczne wymagania dotyczące podsystemu emulatorów wynikają z postawionego przed projektem zadania i sformułowanych wymagań na oprogramowanie. Schematycznie wymagania na oprogramowanie systemu WAZkA przedstawiono na rysunku 3.

Na tle diagramu przypadków użycia całego systemu WAZkA (schematycznie przedstawionych funkcjonalności systemu) został wyróżniony i powiększony fragment wymagań skojarzonych z emulacją systemów monitorowania. Widać na nim, że z emulacji korzystają *Operator emulatorów* i *Kierownik szkolenia*. Sama emulacja wykorzystywana jest do procesu szkolenia, a z kolei ona wykorzystuje standardowe funkcjonalności wymiany informacji systemu WAZkA z otoczeniem. Tak określone wymagania, rozszerzone o specyfikacje przypadków użycia i wymagania uzupełniające, stanowią specyfikację wymagań na oprogramowanie, które jest podstawą do wykonania projektu i zaimplementowania go w oprogramowaniu.

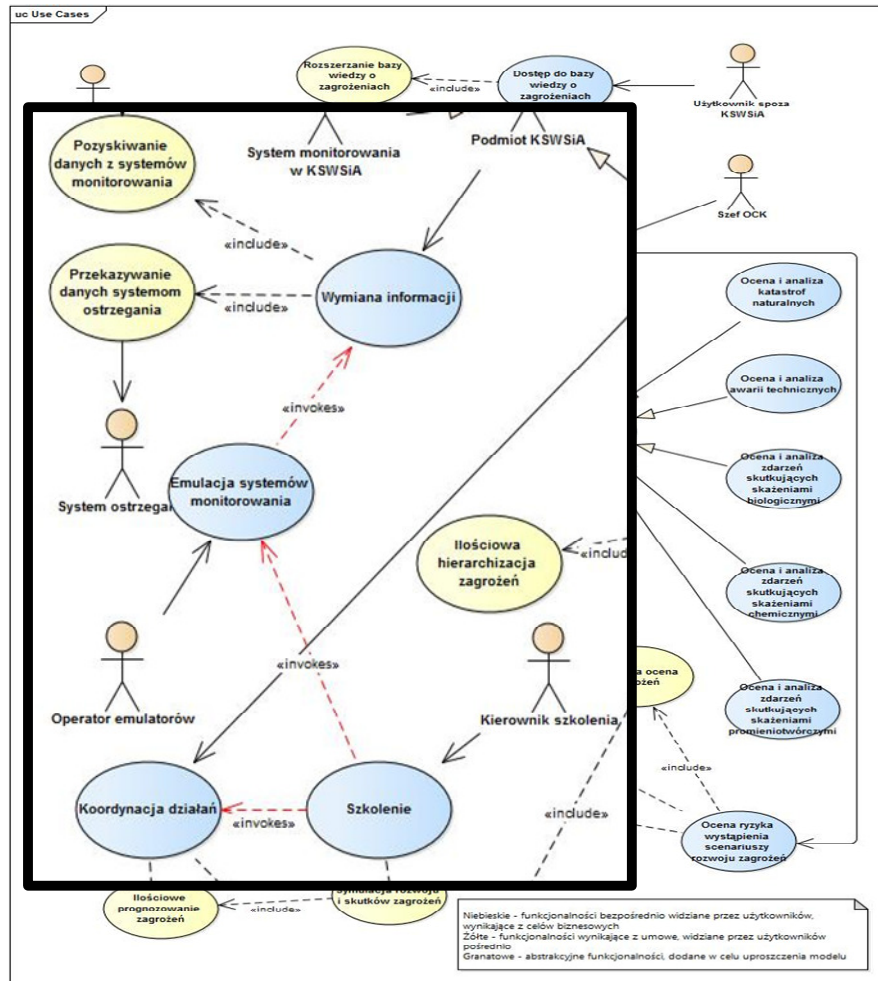
Wykonana na podstawie wymagań analiza oczekiwanego zachowania się podsystemu emulatorów doprowadziła do opracowania warstwowej koncepcji oprogramowania. Została ona przedstawiona na rysunku 4.

Kluczowym komponentem opracowanej koncepcji jest komponent *emulatory*. Występuje on w tylu egzemplarzach, ile zostało opracowanych typów emulatorów. Na potrzeby systemu WAZkA zostały opracowane trzy typy:

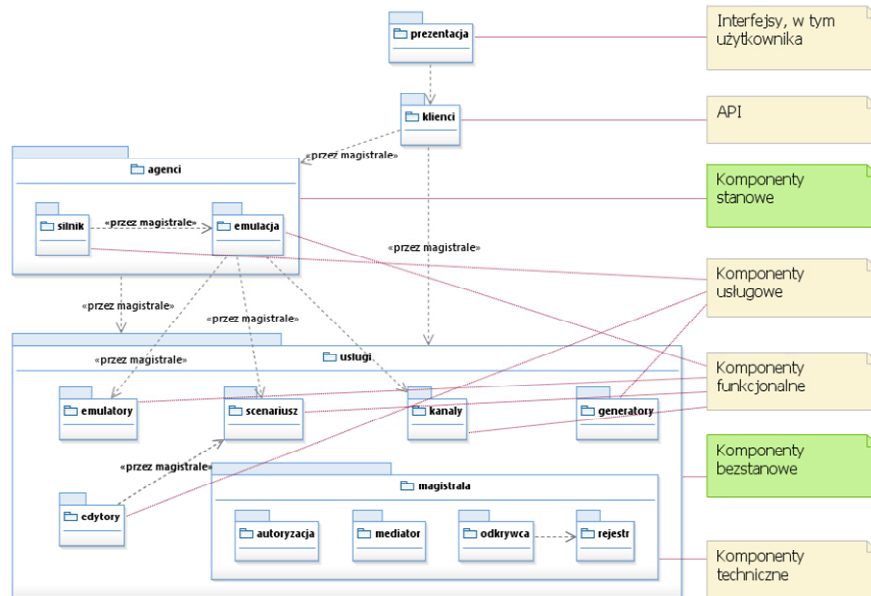
- proste generowanie danych określone w scenariuszu;
- proste, określone w scenariuszu, ustalanie stanu środowiska i automatyczne generowanie danych;
- generowanie danych z wykorzystaniem modelu zachowania się zjawiska o parametrach określonych w scenariuszu oraz automatyczne generowanie danych.

Różnią się one stopniem zaawansowania technologicznego, a co za tym idzie, skomplikowaniem wykonania. W dalszej części przedstawiono przyjęte rozwiązania dla poszczególnych typów emulatorów. Opis przedstawiony został zgodnie ze stopniem zaawansowania technologicznego, rozpoczynając od najprostszego.

Emulatory systemów monitorowania zagrożeń skażeniami
na potrzeby systemu WAZkA



Rys. 3. Diagram przypadków użycia systemu WAZkA i podsystemu emulatorów
Fig. 3. The use cases diagram of the WAZkA system and the emulator subsystem



Rys. 4. Warstwowa koncepcja oprogramowania podsystemu emulatorów

Fig. 4. Layered software concept of the emulator subsystem

Proste generowanie danych określone w scenariuszu

Ta koncepcja emulacji jest najprostsza, ale jednocześnie najbardziej elastyczna. Polega na tym, że to, co ma być przekazane w komunikatach do systemu, jest jawnie zdefiniowane w scenariuszu. Scenariusz emulacji musi zawierać:

- oczekiwany czas wysłania komunikatu,
- treść do wysłania w komunikacie.

Cała trudność w wykorzystaniu takiego emulatora polega na tym, że model rozprzestrzeniania się zjawiska i działania systemu monitorującego muszą istnieć „w głowie” osoby budującej scenariusz (scenarzysty, analityka lub kierownika ćwiczenia). W tym zakresie emulator nie wspomaga użytkownika.

Proste, określone w scenariuszu, ustalanie stanu środowiska i automatyczne generowanie danych

Tego typu scenariusz ma wbudowany model działania systemu monitorującego. Tego typu system jest zazwyczaj w pełni automatyczny i algorytm jego działania jest dość prosty. Stąd model takiego systemu jest prosty, a więc i emulator nie jest trudny do wykonania. W „głowie” scenarzysty nadal musi być model rozprzestrzeniania się zjawiska. Jednak budowanie scenariusza sprowadza się tylko do określenia wskazań, jakie mają pojawiać się na czujnikach. W związku z tym scenariusz musi zawierać:

- oczekiwany czas zmiany wartości odczytywanej przez czujnik,
- identyfikator (położenie) czujnika,
- wartość, jaka ma być odczytana z czujnika,

- parametry odczytywania czujnika przez system monitorowania:
 - częstotliwość odczytywania wartości z czujników,
 - częstotliwość wysyłania danych do systemu.

Tego typu emulator nadal nie wspomaga scenarzysty w zakresie modelowania zjawiska, ale odciąża go od organizacji procesu przekazywania danych z czujników do systemu.

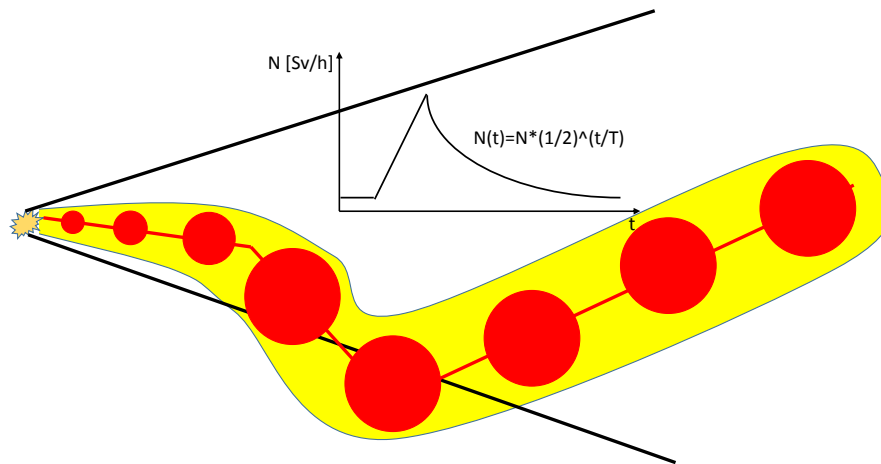
Generowanie danych z wykorzystaniem modelu zachowania się zjawiska o parametrach określonych w scenariuszu oraz automatyczne generowanie danych

W tego typu emulatorze w scenariuszu umieścić trzeba tylko parametry inicjalizacji zdarzenia i parametry działania systemu monitorowania. Scenarzysta musi wiedzieć tylko jakie, zdarzenia mają zajść, natomiast nie musi wiedzieć, jak one przebiegają. Nie jest wymagane od scenarzysty posiadanie zaawansowanej wiedzy. Również czas opracowywania scenariusza (a więc i przygotowania ćwiczeń) jest znacznie krótszy. Natomiast konstrukcja samego emulatora jest znacznie bardziej skomplikowana. Jednak uwzględniając wcześniej uzasadnione założenia, model rozprzestrzeniania się zjawiska może być uproszczony. Rysunek 5 przedstawia schematycznie przyjęty model. Rozszerzające się linie określają sektor, w którym z prawdopodobieństwem 0,7 będzie znajdowała się całość skażeń wynikających z rozprzestrzeniania się zjawiska. Sektor ten jest wyznaczany na podstawie pragmatycznej metodyki ATP 45(D). Zgodnie z tak określonym sektorem wyznaczane są położenia obłoku skażenia (przedstawione w postaci kół). Przejście obłoku przez teren powoduje jego skażenie, które jest zależne od poziomu skażenia obłoku, jego wielkości i czasu przechodzenia nad terenem. Po przejściu obłoku następuje samoistne obniżanie skażenia, zależne głównie od czasu połowicznego rozpadu środka skażającego. Przyjęto tutaj również pragmatyczną regułę, która określa, że po czasie siedmiokrotnie większym od czasu ekspozycji skażenie terenu maleje dziesięciokrotnie.

Do określania kierunku rozprzestrzeniania się obłoku wykorzystywane są dane pogodowe, ograniczone do kierunku i efektywnej prędkości wiatru. Pozostałe parametry zostały pominięte, co jest dopuszczalne dla modelu wykorzystywanego w emulacji. Oczywiście modele wykorzystywane do oceny i podejmowania decyzji muszą być znacznie bardziej zaawansowane, a co za tym idzie, bardziej adekwatne w stosunku do rzeczywistego zjawiska.

Z powyższego wynika, co powinno być zawarte w scenariuszu:

- oczekiwany czas zajścia zdarzenia,
- miejsce zajścia zdarzenia,
- wielkość zdarzenia (moc wybuchu),
- parametry pogodowe:
 - czas zmiany warunków pogodowych,
 - kierunek wiatru,
 - efektywna prędkość wiatru;
- parametry odczytywania czujnika przez system monitorowania:
 - częstotliwość odczytywania wartości z czujników,
 - częstotliwość wysyłania danych do systemu.



Rys. 5. Schemat modelu rozprzestrzeniania się zjawiska
 Fig. 5. Sketch of the phenomenon spreading model

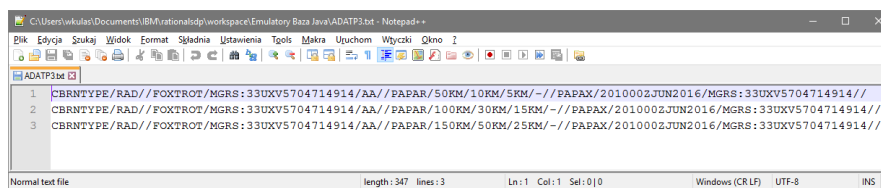
Pozostałe parametry są pochodną danych zawartych w scenariuszu. Należą do nich:

- trajektoria rozprzestrzeniania się zjawiska (wyliczana na podstawie kierunku wiatru),
- prędkość rozprzestrzeniania (wyliczana na podstawie prędkości wiatru),
- czas trwania rozprzestrzeniania (wyliczany na podstawie wielkości zdarzenia),
- prędkość rozszerzania się obłoku (wyliczana na podstawie wielkości zdarzenia),
- początkowa wielkość dawki (wyliczana na podstawie wielkości zdarzenia),
- czas połowicznego rozpadu (przyjęto typową wartość).

4 Efekt końcowy

Opracowano wiele scenariuszy dla każdego z omówionych typów emulatorów. W tym punkcie przedstawiono po jednym przykładzie dla każdego.

Jako przykład prostego generowania danych, na podstawie opracowanego scenariusza, może posłużyć wysłanie trzech komunikatów, w standardzie ADatP3, o strefach skażenia. Scenariusz zawiera jawną postać wysyłanych komunikatów. Efektem końcowym jest przesłanie ich do systemu. Postać tych komunikatów przedstawia rysunek 6. Dla potrzeb artykułu wyświetlono je w postaci tekstowej, w zwykłym edytorze tekstowym.



Rys. 6. Wyniki działania emulatora prostego
 Fig. 6. Results of the simple emulator

*Emulatory systemów monitorowania zagrożeń skażeniami
na potrzeby systemu WAZka*

Do zaprezentowania wyników działania emulatora zawierającego model systemu monitorowania wykorzystano prezentację danych w postaci standardowego dla tego systemu arkusza kalkulacyjnego. Arkusz taki jest przekazywany w sposób automatyczny do systemu WAZka jako załącznik do poczty elektronicznej. Przykładowy arkusz przedstawiony jest na rysunku 7. Zawiera on informacje o czujnikach (ich nazwy i lokalizacje), odczytane wartości promieniowania i czas wykonania odczytu. Dane do tabeli zbierane są automatycznie, natomiast wartości odczytywane przez czujniki zdefiniowane są w scenariuszu.

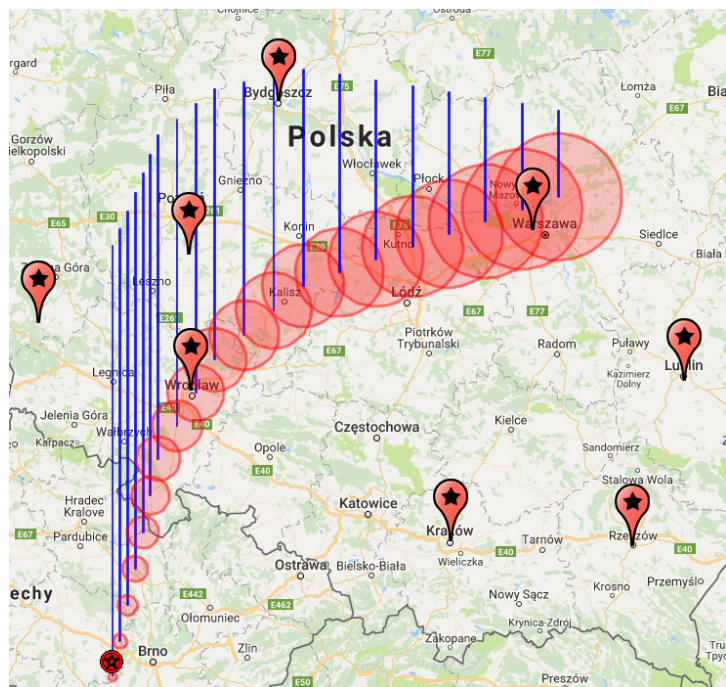
A	B	C	D	E	F	G	H	I	J	
1	Lokalizacja	Szerokość	Długość	Urządzenie	Data pomiaru	Kanał	Param	Wartość	Jednostka	Ważny
2	Warszawa	52,262043	20,87641335	Warszawa - Warszawa (PL-01)	2017-11-00 00:00:00	Dose rate (DOSE-RATE_Dose)	Dose rate (DOSE-RATE)	99	nSv/h	1
3	Zagań	51,61520004	15,31297684	Zagań - Zagań (PL-02)	2017-11-00 00:00:00	Dose rate (DOSE-RATE_Dose)	Dose rate (DOSE-RATE)	98	nSv/h	1
4	Wrocław	51,14305878	17,02443314	Wrocław - Wrocław (PL-03)	2017-11-00 00:00:00	Dose rate (DOSE-RATE_Dose)	Dose rate (DOSE-RATE)	94	nSv/h	1
5	Szczecin	53,44453049	14,52349663	Szczecin - Szczecin (PL-04)	2017-11-00 00:00:00	Dose rate (DOSE-RATE_Dose)	Dose rate (DOSE-RATE)	92	nSv/h	1
6	Kraków	50,07149887	19,95605278	Kraków - Kraków (PL-05)	2017-11-00 00:00:00	Dose rate (DOSE-RATE_Dose)	Dose rate (DOSE-RATE)	95	nSv/h	1
7	Lublin	51,21460724	22,57163239	Lublin - Lublin (PL-06)	2017-11-00 00:00:00	Dose rate (DOSE-RATE_Dose)	Dose rate (DOSE-RATE)	99	nSv/h	1
8	Gdynia	54,52026367	18,54649544	Gdynia - Gdynia (PL-07)	2017-11-00 00:00:00	Dose rate (DOSE-RATE_Dose)	Dose rate (DOSE-RATE)	93	nSv/h	1
9	Bydgoszcz	53,13835144	18,01563454	Bydgoszcz - Bydgoszcz (PL-08)	2017-11-00 00:00:00	Dose rate (DOSE-RATE_Dose)	Dose rate (DOSE-RATE)	98	nSv/h	1
10	Rzeszów	50,03532791	21,9926796	Rzeszów - Rzeszów (PL-09)	2017-11-00 00:00:00	Dose rate (DOSE-RATE_Dose)	Dose rate (DOSE-RATE)	98	nSv/h	1
11	Bartoszyce	54,24460602	20,80464554	Bartoszyce - Bartoszyce (PL-10)	2017-11-00 00:00:00	Dose rate (DOSE-RATE_Dose)	Dose rate (DOSE-RATE)	95	nSv/h	1
12	Śrem	52,09538269	17,00196838	Śrem - Śrem (PL-11)	2017-11-00 00:00:00	Dose rate (DOSE-RATE_Dose)	Dose rate (DOSE-RATE)	96	nSv/h	1
13	Świnoujście	53,90303802	14,24202251	Świnoujście - Świnoujście (PL-12)	2017-11-00 00:00:00	Dose rate (DOSE-RATE_Dose)	Dose rate (DOSE-RATE)	93	nSv/h	1
14	Ustka	54,56268311	16,8116188	Ustka - Ustka (PL-13)	2017-11-00 00:00:00	Dose rate (DOSE-RATE_Dose)	Dose rate (DOSE-RATE)	94	nSv/h	1
15										

Rys. 7. Wyniki działania emulatora z automatycznym systemem monitorowania

Fig. 7. Results of the emulator with an automatic monitoring system

Wyniki działania emulatora z zaimplementowanym modelem rozprzestrzeniania się zjawiska przedstawione zostały na rysunku 8. Dla potrzeb artykułu przedstawione zostały one w formie graficznej. Oryginalne dane generowane są w postaci analogicznego arkusza jak w poprzednim przykładzie. Jednak forma graficzna pozwala na znacznie prostsze oszacowanie poprawności zbudowania modelu, działania jego implementacji i konstrukcji scenariusza zgodnej z intencjami scenarzysty.

Rysunek przedstawia punkt inicjalizacji zjawiska i kolejne położenia obłoku skażenia wyliczane w odstępach dziesięciominutowych (obliczenia, dla uzyskania odpowiedniego efektu, zostały wykonane przy bardzo silnym wietrze). Pionowe linie wyrowadzone z środka kół są logarymicznie proporcjonalne do poziomu skażenia obłoku. Skręcanie obłoku w kierunku Warszawy jest spowodowane zmianami warunków atmosferycznych (zmiana kierunku wiatru). Ponadto na mapie zaznaczono położenia czujników mierzących poziom skażenia. Wyniki odczytywane przez automat z tych czujników są przekazywane w formie arkusza do systemu WAZka.



Rys. 8. Wyniki działania emulatora z modelem zjawiska
 Fig. 8. Results of the emulator with the phenomenon model

5 Podsumowanie

Przedstawione środowisko emulacyjne pozwala na elastyczne budowanie i dostarczanie emulatorów. W opisanym przypadku zostało ono wykorzystane na potrzeby konkretnego systemu, ale potencjalnie można je wykorzystać w dowolnym innym systemie. Przedstawiono trzy wykonane emulatory, ale istnieje możliwość rozszerzania środowiska o inne, zarówno budowane na potrzeby konkretnego systemu, jak i uniwersalne.

Najprostszy ze zbudowanych emulatorów ma duży potencjał uniwersalności. Ponieważ nie ma w nim wbudowanego żadnego modelu, może być on dowolnie wykorzystywany. Może on stanowić punkt początkowy wykorzystania emulatorów w dowolnym systemie. Bardziej rozbudowane emulatory można budować i wdrażać stopniowo w trakcie realizacji przedsięwzięcia.

Głównym celem dostarczenia emulatorów była możliwość łatwego wykorzystania systemu do celów szkoleniowych. Jednak emulatory okazały się również przydatne w procesie wytwarzania oprogramowania. Umożliwiły one szybkie dostarczenie środowiska do testowania budowanego produktu.

Przyjęta została zasada dostarczania emulatorów od najprostszych. Wydaje się to sprzeczne z zasadami inżynierii oprogramowania, które mówią, że należy rozpoczynać od zadań najbardziej ryzykownych. Jednak należy pamiętać, że emulatory są tylko podsystemem całości, a ryzyko należy rozważać w stosunku do całego systemu. A tutaj szybkie dostarczenia środowiska do testowania jest najważniejsze i powoduje obniżenie ryzyka całego przedsięwzięcia przez umożliwienie lepszego testowania.

Dla dużych systemów zazwyczaj wymagane jest, aby były dostarczane łącznie ze środowiskiem do szkolenia. Duże szkolenia wymagają zaangażowania dużych środków i czasu. Konieczne jest opracowanie scenariuszy ćwiczeń oraz wyszkolenie obsługi. Główna część obsługi to tzw. podgrywka, czyli ludzie udający źródła lub odbiorców komunikatów. W obu tych aspektach emulatory powodują obniżenie kosztów i skrócenie czasu przygotowania ćwiczeń przez:

- dostarczenie narzędzi do budowania scenariuszy,
- uproszczenie budowania scenariuszy przez dostarczenie zaimplementowanych modeli,
- dostarczenie narzędzi do automatycznego generowania zdarzeń z pominięciem obsługi.

Koncepcję emulatorów można rozszerzać. Jedną z możliwości jest dostarczenie, obok emulatorów źródeł komunikatów, również emulatorów konsumentów. Przedstawione emulatory, po ich uruchomieniu, działają zgodnie ze scenariuszem, bez możliwości ingerencji. W niektórych przypadkach może okazać się przydatne wpływanie na emulację w trakcie jej trwania. Jest to drugi z możliwych kierunków rozwoju prezentowanych emulatorów.

Literatura

1. Najgebauer A., Antkiewicz R., Chmielewski M., Kasprzyk R., Koszela J., Kulas W., Pierzchała D., Rulka J., Tarapata Z., Wantoch-Rekowski R.: Modelowanie i symulacja pola walki. *Problemy modelowania i projektowania opartych na wiedzy systemów informatycznych na potrzeby bezpieczeństwa narodowego*, s. 71-82, Wojskowa Akademia Techniczna 2014
2. Pierzchała D.: Symulacja komputerowa – od procedury do chmury. *Problemy modelowania i projektowania opartych na wiedzy systemów informatycznych na potrzeby bezpieczeństwa narodowego oraz administracji publicznej*, s. 104-118, Warszawa 2014
3. Najgebauer A.: Integracja symulatorów pola walki na potrzeby sił zbrojnych. *Niebezpieczny świat: Systemy. Informacja. Bezpieczeństwo*, s. 212-222, Akademia Obrony Narodowej 2015
4. Antkiewicz R., Chmielewski M., Dyk M., Kasprzyk R., Najgebauer A., Pierzchała D., Rulka J., Tarapata Z.: The Qualitative and Quantitative Support Method for Capability Based Planning of Armed Forces Development. *Intelligent Information and Database Systems*, s. 212-223, Springer International Publishing AG 2015
5. Tarapata Z., Dyk M., Kasprzyk R., Kulas W., Pierzchała D., Kołdej J.: Koncepcja szkoleniowej konfiguracji systemu wsparcia analiz zagrożeń skażeniami i alarmowania (WAZKA). *Inżynieria bezpieczeństwa a zagrożenia cywilizacyjne. Zagrożenia CBRNE*, s. 123-136, Częstochowa 2016
6. Antkiewicz R., Najgebauer A., Kasprzyk R., Rulka J., Chlebicki K.: Analityczno-symulacyjne metody oceny wymaganych zdolności SZ RP. XXIII Warsztaty Naukowe PTSK 2016

7. Tarapata Z., Najgebauer A., Koszela J.: Symulacja działań połączonych w szkoleniu wojsk. Materiały konferencyjne z konferencji pt. „Informatyczne zabezpieczenie procesu dowodzenia”, 03.04.2017 r.

Streszczenie

W artykule przedstawione zostały wyniki realizacji projektu dotyczącego monitorowania zagrożeń na potrzeby informatycznego systemu wsparcia analiz zagrożeń skażeniami CBRN, w zakresie podsystemu emulatorów. Głównym celem tego podsystemu jest wsparcie procesu szkolenia. Artykuł zawiera analizę potrzeb, koncepcję wykonania i wyniki przykładowego uruchomienia podsystemu. Potrzeby zostały przedstawione na podstawie analizy środowiska i wymagań postawionych przed całym systemem. Koncepcja wykonania została przedstawiona na przykładzie trzech emulatorów o różnym poziomie zaawansowania technologicznego.

Słowa kluczowe: emulator systemu, monitorowanie zagrożeń, wspomaganie decyzji

Contamination threats monitoring system emulators for the WAZkA system

Summary

This paper presents the results of the threats monitoring project for the IT system needs for supporting analysis of CBRN threats, in the scope of the emulator subsystem. The main purpose of this subsystem is to support the training process. This paper contains an analysis of needs, the implementation concept and the subsystem exemplary launch results. The needs were presented on the basis of the environment analysis and the entire system requirements. The implementation concept has been presented on the three emulator examples with different technological advancement levels.

Keywords: system emulator, threats monitoring, decisi