

Wybrane problemy konstruowania gry wojennej

Jan ZYCH

2 Brygada Lotnictwa Taktycznego w Poznaniu,
ul. Kościuszki 92-98, 60-967 Poznań-9

STRESZCZENIE: W artykule omówiono wybrane wątki konstruowania środowiska symulacyjnego, będącego integralną częścią komputerowej, interaktywnej gry wojennej GAMBIT. Na tle ogólnego schematu funkcjonowania gry opisane zostały mechanizmy synchronizacji procesów realizowanych przez serwer oraz sposób komunikowania się użytkowników z serwerem.

1. Wprowadzenie

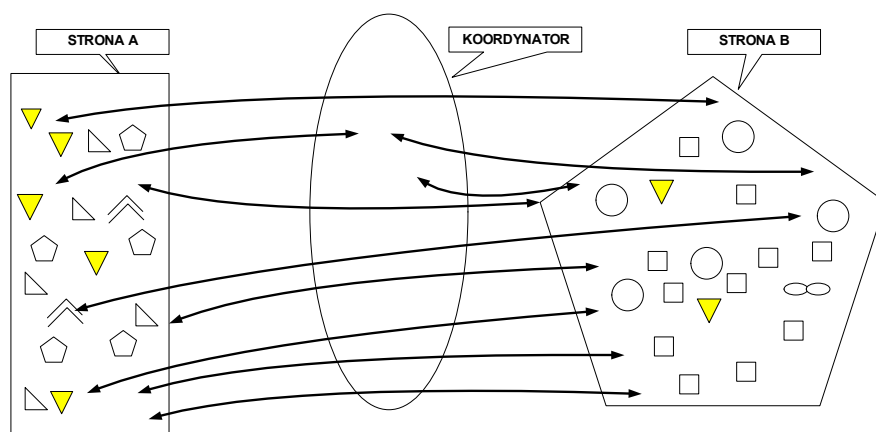
Gra wojenna GAMBIT daje możliwość ćwiczenia procesów podejmowania decyzji, występujących w realnych sytuacjach, z możliwością obserwowania skutków tych procesów. Rozgrywana jest na podkładach mapowych w kilku skalach, z uwzględnieniem wyniosłości terenu. Wymaganą platformą sprzętową jest komputer klasy IBM PC, a środowiskiem systemowym - MS WINDOWS. Znaki taktyczne wykorzystywane w grze są zgodne z obowiązującą w SZ RP notacją NATO, a zobrazowanie sytuacji operacyjno-taktycznej na ekranach monitorów jest zbliżone do topograficznych map papierowych.

Głównym przesłaniem artykułu jest prezentacja idei autorskich rozwiązań programowych, wykorzystanych podczas konstruowania gry wojennej.

Gra GAMBIT umożliwia symulację ćwiczeń walki grupowej o ugrupowaniach niejednorodnych, z uwzględnieniem występowania

w ćwiczeniu kierownika ćwiczeń- koordynatora (rys. 1). Uczestnicy gry komunikują się poprzez sieć (również przez Internet).

W grze określono dwie walczące ze sobą strony „A”/”X”/”CZERWONI” i „B”/”Y”/”NIEBIESCY”. Każda ze stron dysponuje zasobami środków walki, przydzielonymi przez koordynatora - mogą być to faktyczne zasoby którymi dysponuje określona jednostka organizacyjna. W wyniku decyzji (posunięć) strony mogą uzyskiwać korzyści lub ponosić straty (w sensie przyjętego kryterium), np. w wyniku rażenia lotniczym środkiem bojowym stacji radiolokacyjnej, sprawność tej stacji zmniejszy się odpowiednio do siły rażenia tego środka oraz odporności stacji radiolokacyjnej na ten środek rażenia.



Rys.1. Schemat ćwiczeń z koordynatorem walczących ze sobą stron (o ugrupowaniach niejednorodnych)

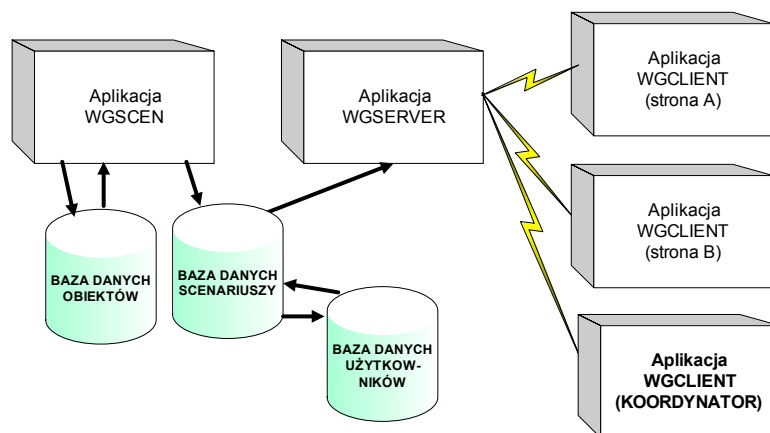
Gra jest interaktywna: strony mogą ingerować w przebieg symulowanych procesów, obserwując jednocześnie skutki podejmowanych decyzji (własnych, przeciwnika, koordynatora).

2. Mechanizmy synchronizacji danych

Aplikacje składające się na interaktywną komputerową grę symulacyjną GAMBIT tworzą zbiór programów wzajemnie powiązanych (rys. 2). Za pomocą programu WGSCEN kreowane są:

- baza danych scenariuszy,

- baza danych o obiektach (informacje o statkach powietrznych, stacjach radiolokacyjnych, zestawach raketowych, a także o określonych elementach wojsk lądowych i marynarki wojennej),
- baza danych o użytkownikach wraz z kontami dostępowymi;



Rys. 2. Komputerowa gra wojenna jako wzajemnie powiązane aplikacje

Ponadto istnieje możliwość:

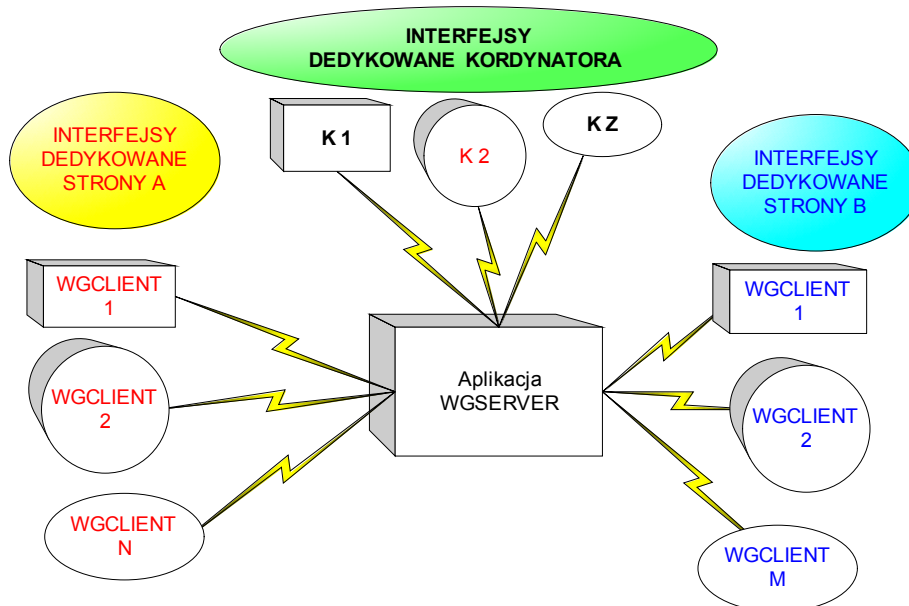
- modyfikowania wcześniej utworzonych scenariuszy,
- wczytania opracowanych podkładów mapowych.

Informacje z bazy danych scenariuszy oraz wymuszenia ze strony użytkowników (pochodzące z aplikacji WGCLIENT) stanowią dane wejściowe do programu WGSERWER, którego zadaniem jest realizacja procesów symulujących grę. WGSERWER dostarcza pełną informację o stanie symulacji. Są przy tym uwzględniane ograniczenia związane z wartościami parametrów taktyczno-technicznych obiektów.

Najistotniejszymi funkcjami programu WGSERWER są:

- przesyłanie komunikatów do strony „X”/”A” lub/i „Y”/”B” z poziomu „Koordynatora”;
- symulacja z możliwością ZATRZYMANIA/WZNOWIENIA;
- możliwość symulacji w trybie przyspieszenia (z możliwością przeskoku do dowolnego momentu czasowego);
- możliwość cofnięcia gry do dowolnego momentu czasowego;
- podgląd wybranych parametrów obiektów z danego scenariusza;
- tworzenie pliku historii z rozegranym epizodem.

Komunikowanie się serwera WGSERVER z klientem WGCLIENT odbywa się za pomocą datagramów, a wykorzystywany jest do tego aplikacyjny protokół transmisji, który bazuje na stosie protokołów TCP/IP (*Transmission Control Protocol / Internet Protocol*).



Rys. 3. Komunikowanie się programów i interfejsy dedykowane

Przetwarzanie WGSERVER realizowane jest w jednym miejscu (na określonym komputerze). W ten sposób jest ułatwiona synchronizacja procesów symulacji.

W grze przyjęto, iż żądania uaktywnienia procesów tworzą kolejkę i są realizowane zgodnie z kolejnością napływania, według reguły FIFO. Okres odświeżania zobrazowania jest stały. W programie WGSERVER monitorowany jest upływ czasu od ostatniego uaktualnienia sytuacji. Gdy czas ten zbliża się do okresu odświeżania, system generuje komunikat WM_TIMER i wstawia go na koniec kolejki żądań uaktywnienia procesów. Aplikacja WGSERVER aktywizuje ten proces dopiero po zrealizowaniu wszystkich wcześniejszych żądań i wywołuje procedurę odświeżania sytuacji na monitorach uczestników gry. Zatem czas między kolejnymi uaktualnieniami sytuacji na monitorach może

odbiegać od założonej wartości.¹ W tej sytuacji przy obliczaniu aktualnej pozycji obiektów graficznych na tle mapy wykorzystuje się funkcję pomiaru czasu, która pozwala wyznaczyć rzeczywisty czas jaki upłynął od poprzedniego uruchomienia procedury odświeżania sytuacji. Ten upływ czasu jest zobrazowywany w dodatkowym polu.

3. Komunikacja między procesami

Aplikacja WGCLIENT zapewnia użytkownikowi dedykowane interfejsy (rys. 3) umożliwiające:

- wpływanie na przebieg symulowanych procesów (w zależności od delegowanych kompetencji);
- wybór skali mapy i interesującego jej fragmentu;
- kształtowanie sytuacji operacyjno-taktycznej poprzez wykorzystywanie tzw. rejestrów decyzyjnych skojarzonych z obiektami (np. statkiem powietrznym, zestawem raketowym, stacją radiolokacyjną);
- włączanie i wyłączenie dodatkowych elementów graficznych poszczególnych interfejsów (za pomocą tzw. wyzwalaczy).

Po uruchomieniu tej aplikacji (w trybie pełnoekranowym) użytkownik na ekranie swojego monitora uzyskuje efekt ruchu dla obiektów występujących w danym scenariuszu i przetwarzanych przez WGSERVER (rys. 4).

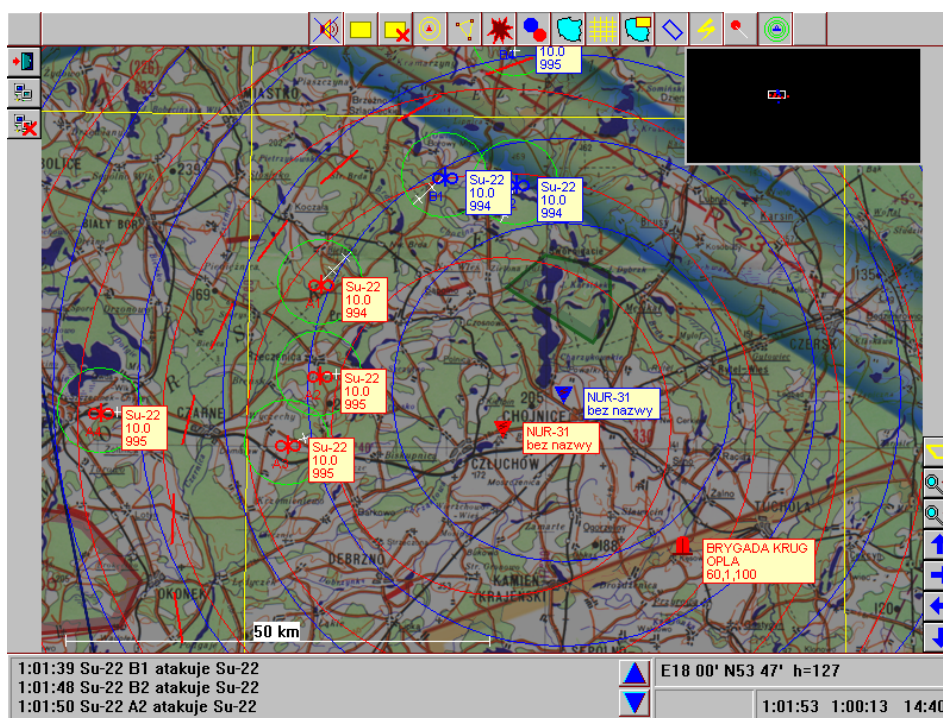
Dla jednoczesnego wykonywania wielu procesów aplikacji istotne są następujące uwarunkowania:

- wymagane jest rozróżnianie użytkownika (odbiorcy), do którego jest skierowany datagram;
- procesy są inicjowane i likwidowane dynamicznie;
- nadawca może mieć zbyt mało informacji, aby wskazać proces na innej jednostce centralnej komputera;
- przeładowanie systemu operacyjnego (np. w przypadku zbyt dużej liczby zgłoszeń) powoduje zakłócenie wszystkich procesów;

¹ W wyniku przeprowadzonych doświadczeń ustalono, iż okres odświeżania sytuacji na ekranach monitorów uczestników gry powinien być zbliżony do wartości 1 [sek]. W przeprowadzonych badaniach, przy wykorzystaniu platformy sprzętowej: AMD/700MHz/128RAM/karta graficzna 32MB, okazało się, że okres odświeżania zmienia się od 990 [ms] do 1200 [ms]. Dodatkowe przypadkowe opóźnienia mogą być spowodowane wahaniami opóźnień transmisji danych w sieci (ang. *jitter*), co jest ceną za możliwość komunikowania się za poprzez Internet.

- biorący udział w grze muszą mieć możliwość wpływu na innych uczestników (odbiorców) na podstawie realizowanych przez nich funkcji, nie wiedząc, który proces realizuje te funkcje;
- odbiorca datagramu musi mieć możliwość ustalenia jakiej funkcji życzy sobie nadawca (aby umożliwić realizację więcej niż jednej funkcji w jednym procesie).

Wymienione uwarunkowania spowodowały, że w wykonywanych procesach nie określa się końcowych odbiorców komunikatów, a wprowadzony został zbiór abstrakcyjnych punktów docelowych zwanych portami protokołów. Jednocześnie lokalny system operacyjny dostarcza mechanizm przydziału procesom portów i określania dostępu do nich.



Rys. 4. Przykładowy ekran dla jednej ze stron w czasie gry. Górna listwa zawiera wyzwalacze, po prawej stronie ekranu są strzałki do przesuwania mapy, na dole zobrazowano czasy: symulacji, operacyjny i systemowy oraz rejestr zdarzeń; po lewej stronie są „przyciski” do logowania; w górnym prawym rogu usytuowane jest okno nawigacyjne

W celu uzyskania połączenia z odległym komputerem, na którym jest uruchomiony program WGSERVER, uczestnik gry musi znać adres IP tego komputera (aby w sposób jednoznaczny zaadresować go w sieci), jak również

numer docelowego portu protokołu na tym komputerze. Zatem każdy komunikat przesyłany w trakcie rozgrywania gry, oprócz adresów IP nadawcy i odbiorcy, musi zawierać numer portu odbiorcy w komputerze, do którego został wysłany oraz numer portu nadawcy w komputerze, do którego mają być adresowane odpowiedzi (dowolny proces odbierający komunikat może wysłać odpowiedź do nadawcy).

Opisany wyżej mechanizm stanowi podstawę komunikacji z serwerem. Adresowanie IP pozwala na jednoznaczne identyfikowanie komputera biorącego udział w grze. Natomiast izolacja procesów realizowanych na tym komputerze jest uzyskiwana drogą przydziału indywidualnych portów. Serwer rezerwuje port X i tworzy gniazdo (SOCKET) w trybie nasłuchiwania (LISTEN). Wszystkie próby połączenia z serwerem na tym porcie trafiają do tego gniazda, którego zadaniem jest odebranie połączenia i wygenerowanie nowego gniazda (już z innym numerem portu). W konsekwencji „nowy klient”, który chciał się połączyć z serwerem, uzyskuje połączenie już z wygenerowanym nowym gniazdem. To nowe gniazdo pracuje w trybie połączeniowym (CONNECT) i przejmuje na siebie obsługę utworzonego połączenia. Natomiast gniazdo powiązane z portem X czeka na kolejne połączenie. Jeżeli ono nastąpi - otrzyma unikatowy (wolny w danej sesji) numer portu.

4. Uwagi końcowe

Kilkuletnia praca nad modelowaniem walki sił obrony powietrznej oraz skonstruowanie komputerowej gry symulacyjnej - **GAMBIT** pozwala autorowi sformułować kilka konkluzji natury bardziej ogólnej:

- Stosując eksperyment symulacyjny można weryfikować nowe metody i techniki prowadzenia działań bojowych oraz wykorzystania nowych środków walki, przy uwzględnieniu przeciwdziałania przeciwnika.
- Stosowanie nowoczesnych systemów informatycznych w sferze badań efektywności funkcjonowania wybranych elementów sił powietrznych może i powinno przyczynić się do identyfikacji, a następnie do wyjaśnienia i podjęcia prób rozwiązania złożonych, często bardzo kontrowersyjnych problemów występujących na współczesnym polu walki.
- Odpowiednio skonstruowane gry wojenne - systemy informatyczne – generują dodatkowe możliwości badania złożonych systemów walki w wymiarze czasowym i przestrzennym. Obserwacje i rejestrowanie zmian zachodzących w tych procesach mogą prowadzić do istotnych

usprawnień organizacyjnych, funkcjonalnych oraz informacyjnych. Wyniki uzyskane za pomocą eksperymentu symulacyjnego, mimo kontrowersyjności (przede wszystkim chodzi tu o adekwatność modeli symulacyjnych), dają dodatkową wiedzę o skomplikowanych związkach przyczynowo - skutkowych dotyczących współczesnego pola walki.

- Znoszone są ograniczenia natury humanitarnej, finansowej, terytorialnej, ekologicznej, czasowej. W tego typu systemach symulacyjnych nawet błędne decyzje nie powodują katastrofy.
- Dynamiczny rozwój systemów komputerowych i systemów informatycznych, ich powszechna dostępność oraz relatywnie niska cena w stosunku do mocy obliczeniowych prawdopodobnie z wielokrotni zapotrzebowanie na komputerowe gry symulacyjne.

Literatura:

- [1] Zych. J., *Komputerowa gra symulacyjna - wybrane elementy pola walki*. IX Konferencja Naukowa - Sterowanie i regulacja w radiolokacji i obiektach latających (Jelenia Góra 16-18 czerwca 1998).
- [2] Zych. J., *Model walki sił obrony powietrznej szczebla taktycznego*, Akademia Obrony Narodowej, Warszawa, 2002.

Recenzent: dr inż. Marek Salamon

Praca wpłynęła do redakcji 12.01.2004