AKADEMIA OBRONY NARODOWEJ

CENTRUM SYMULACJI
I KOMPUTEROWYCH GIER WOJENNYCH

TECHNOLOGIA PROWADZENIA SYMULACJI W SYSTEMACH SYMULACYJNYCH „ZŁOCIEŃ" i „JTLS"

Planowanie eksperymentów symulacyjnych w wybranych procesach walki

BIBLIOTEKA GŁÓWNA - ARCHIWUM
Nr ewid. 59579
II
Akademii Obrony Narodowej

WARSZAWA 2005

AKADEMIA OBRONY NARODOWEJ

CENTRUM SYMULACJI
I KOMPUTEROWYCH GIER WOJENNYCH

BIBLIOTEKA GŁÓWNA - ARCHIWUM
Nr ewid. 59579
II
Akademii Obrony Narodowej

TECHNOLOGIA PROWADZENIA SYMULACJI W SYSTEMACH SYMULACYJNYCH „ZŁOCIEŃ" i JTLS"

Planowanie eksperymentów symulacyjnych w wybranych procesach walki

płk dr inż. Ryszard WIELEBA
ppłk mgr inż. Grzegorz PIETRZAK
mjr mgr inż. Sławomir BIENIEK

Warszawa **2005**

Spis treści

SPIS TREŚCI **2**
WSTĘP **4**
1. ISTOTA SYMULACJI I EKSPERYMENTU SYMULACYJNEGO **7**
2. PLANOWANIE EKSPERYMENTU W ĆWICZENIACH CAX **10**
2.1. Faza przygotowania eksperymentu symulacyjnego 12
2.2. Faza realizacji eksperymentu symulacyjnego 14
2.3. Faza analizy i interpretacji wyników eksperymentu symulacyjnego 15
3. UWARUNKOWANIA EKSPERYMENTU SYMULACYJNEGO. **17**
3.1. Określenie celu i zakresu eksperymentu symulacyjnego 21
3.2. Ograniczenia czasowe eksperymentu symulacyjnego 22
3.3. Powtarzalność wyników eksperymentu symulacyjnego 23
4. PLANOWANIE ORGANIZACJI TWORZENIA BAZY DANYCH **24**
4.1. Zespół przygotowujący bazę danych 24
4.2. Informacje potrzebne od użytkownika 25
4.3. Wymagania dotyczące danych narzucane przez model 26
4.4. Dokumentowanie bazy danych 27
4.5. Testowanie bazy danych 27
4.5.1. Test 1 bazy danych – eksperyment elementarny *28*
4.5.2. Test 2 bazy danych – eksperyment „agregujący” *28*
4.5.3. Test 3 bazy danych – „próba generalna” *30*
5. PLANOWANIE FAZY PRZYGOTOWANIA **31**
5.1. Wybór metody i utworzenie scenariusza 32
5.2. Wprowadzanie danych i aktualizacje scenariusza 33
5.3. Weryfikacja bazy danych scenariusza 33
5.4. Testowanie statycznego i dynamicznego etapu symulacji 34
6. PLANOWANIE FAZY SYMULACJI **43**
6.1. Planowanie prostych eksperymentów symulacyjnych 44
6.2. Planowanie złożonych eksperymentów symulacyjnych 46
6.3. Wykrywanie i usuwanie błędów symulacji 51
7. PLANOWANIE GROMADZENIA DANYCH DO ANALIZY I OCENY **53**
7.1. System JTLS 53
7.1.1. GAWS *54*
7.1.1.1. Możliwości gromadzenia i składowania danych 54
7.1.1.2. Możliwości odtwarzania zapisanych danych 56
7.1.2. Post Processor System *58*
7.1.2.1. Przygotowanie do użycia PPS 59
7.1.2.2. Uaktywnianie PPS 60
7.1.2.3. Dostęp do danych PPS 60
7.1.2.4. Funkcjonalność PPS 61
7.1.3. Open Access Programs *64*
7.1.3.1. Scenario Data Client (SDC) 65
7.1.3.1.1. Tabele z danymi SDC 66

7.1.3.2. Message Processing Client (MPC) 67
7.1.3.2.1. Tabele z danymi MPC 67
7.1.3.3. Order Entry Client (OEC) 70
7.1.4. Wskazówki do uwzględnienia podczas planowania 70
7.1.4.1. GAWS 70
7.1.4.2. Post Processor System 71
7.1.4.3. Open Access Programs 71
7.1.4.4. Porównanie funkcjonalności narzędzi 72
7.2. SYSTEM „ZŁOCIEŃ" 75
7.2.1. Zobrazowanie wyników w trakcie eksperymentu 75
7.2.2. Analizy wyników ćwiczeń po ich zakończeniu 77
ZAKOŃCZENIE 80
LITERATURA 82
DODATKI 83
DANE Z PRZEBIEGU EKSPERYMENTU GROMADZONE W TABELACH SDC SYSTEMU JTLS 83
SPIS RYSUNKÓW 107
SPIS TABEL 108

Wstęp

Prezentowane opracowanie **„Planowanie eksperymentów symulacyjnych w wybranych procesach walki"** zrealizowane zostało w ramach pracy naukowo – badawczej (poz. **III.5.4.1** w „Planie zadaniowo – finansowym działalności naukowej AON na 2005 rok). Opracowanie stanowi pierwszy etap realizacji zadania badawczego pt. **„Technologia prowadzenia symulacji w systemach symulacyjnych ZŁOCIEŃ i JTLS".**

Jednym z wyznaczników nowoczesności sił zbrojnych jest stosowanie najnowszych technik symulacyjnych w różnych dziedzinach szkolenia i w badaniach naukowych. Aktualnie niewiele państw NATO posiada własne, narodowe systemy symulacyjne walki do wspomagania ćwiczeń. Najwięcej tego typu systemów powstaje w USA, gdzie najwcześniej rozpoczęto prace w tej dziedzinie i gdzie wiele poważnych firm konkuruje o zamówienia z sektora wojskowego.

Uświadomioną już prawdą jest, że szkolenie z wykorzystaniem nowoczesnych technologii (w tym CAX a także symulatorów, trenażerów, poligonów laserowych itp.) jest wielokrotnie tańsze i bardziej efektywne od szkolenia tradycyjnego. Doskonalenie umiejętności dowódców i oficerów sztabu w zakresie procedur związanych z planowaniem działań oraz w zakresie dowodzenia wojskami jest możliwe przez wykorzystanie systemów symulacji komputerowej. Przy czym systemy te zapewniają w coraz większym stopniu realizm sytuacji bojowej, środowiska geograficzno – topograficznego oraz zdarzeń i procesów zachodzących na polu walki. Podstawą komputerowych systemów są modele symulacyjne odwzorowujące procesy wzajemnego rażenia (oddziaływania) i procesy wewnętrzne: dowodzenia, zabezpieczenia (bojowego i logistycznego), wsparcia, a ponadto proces dynamiki pola walki w realnym środowisku (warunkach terenowych i meteorologicznych) w realiach czasowych. Modele symulacyjne generują zdarzenia, nowe sytuacje i zachowania wojsk zgodne z zasadami taktyki i sztuki walki. Odzwierciedlają skutki podejmowanych decyzji przez ćwiczące sztaby i stwarzają im nowe wyzwania. Ćwiczący muszą reagować na rozwój sytuacji, a każdy błąd, brak wyobraźni, czy niewłaściwa ocena działań przeciwnika, przynosi wymierne efekty. System w sposób obiektywny i bezstronny przedstawi wynik walki niezależnie od stosunku oceniającego do ćwiczącego.

Autorzy opracowania od lat specjalizują się w dziedzinie modelowania działań bojowych, w szczególności w zakresie modeli i systemów symulacyjnych walki. W obszarze ich zainteresowań mieszczą się nie tylko problemy ogólnoteoretyczne, ale także problemy

implementacji do poziomu systemu symulacyjnego (a więc technologie modelowania, projektowania i implementacji) oraz problemy efektywnej ich eksploatacji (wdrożenia, eksploatacji, pielęgnacji oraz modyfikacji). Jednocześnie uczestniczą w przygotowywaniu i przeprowadzaniu ćwiczeń dowódczo – sztabowych wspomaganych komputerowo, w tym ćwiczeń realizowanych w oparciu o systemy symulacyjne walki. W okresie ostatnich dwóch lat uczestniczyli również w trzech szkoleniach specjalistycznych, organizowanych przez producenta systemu JTLS – amerykańską firmę ROLAND, dotyczących programowania, projektowania i realizacji ćwiczeń CAX oraz istoty systemu symulacyjnego JTLS. Nabyte doświadczenia pozwalają zidentyfikować operacyjne wymagania oraz potrzeby informacyjne zespołów funkcyjnych ćwiczenia – z jednej strony, ale także funkcjonalne możliwości eksploatowanych systemów symulacyjnych – z drugiej strony.

Autorzy są pracownikami Centrum Symulacji i Komputerowych Gier Wojennych (CSiKGW), w którym przeprowadza się symulację działań militarnych (i pozamilitarnych) w oparciu o symulacyjne systemy walki, gdzie od 2005 organizowane będą ćwiczenia typu CAX. Aktualnie na wyposażeniu CSiKGW znajdują się dwa systemy symulacyjne: JTLS – system amerykańskiej produkcji, stosowany również w innych państwach NATO, który jest rozpoznawany przez autorów od początku 2003 roku oraz system „Złocień"- produkt Wojskowej Akademii Technicznej, będący aktualnie na etapie badań, a którego wdrożenie planowane jest w 2005 roku. Jedyne realne próby eksperymentalne, jakie autorzy przeprowadzić mogli, realizowane były na systemie JTLS. Zatem doświadczenia nabyte podczas realizacji eksperymentów symulacyjnych na systemie JTLS stały się podstawą weryfikacji wiedzy teoretycznej autorów i, na ile to było możliwe, oceny funkcjonalności operacyjnej i informatycznej tego systemu.

Głównym problemem badawczym pracy była konceptualizacja technologii planowania eksperymentów symulacyjnych na potrzeby ćwiczeń CAX. Przedstawione opracowanie nie dotyczy zagadnień organizowania zespołów ćwiczących i roli poszczególnych uczestników ćwiczenia, lecz problemów czysto metodycznych i technologicznych związanych z możliwościami systemów symulacyjnych. Poczynając od istoty systemów symulacyjnych w ogólności, jak i w poszczególnych fazach realizacji ćwiczenia typu CAX, autorzy starali się opracować sposób postępowania, zapewniający poprawne planowanie (projektowanie) eksperymentów symulacyjnych oraz ich późniejszą ich realizację podczas ćwiczenia CAX. Opracowanie to jest efektem dotychczasowych doświadczeń i badań autorów problematyki konceptualizacji, projektowania i implementacji systemów symulacji walki zbrojnej.

Autorzy wyrażają przekonanie, że problemy przedstawione w opracowaniu poddane zostaną głębszej analizie badawczej (w szczególności po uzyskaniu możliwości eksperymentowania na innych systemach) przez szersze grono pracowników naukowo – badawczych AON, studentów i szkolonych oficerów dowództw i sztabów. Wtedy też możliwe będzie opracowanie bardziej dojrzałej metodyki planowania eksperymentu symulacyjnego.

1. Istota symulacji i eksperymentu symulacyjnego

Symulacja jest to proces konstruowania **historii stanów**[1]. Dokładniej symulacja - to **konstruowanie w chronologicznym porządku opisów stanów tworzących historię stanów**. Taka metoda może być na przykład przeciwstawiona możliwości generowania informacji do historii stanów w porządku innym niż chronologiczny, a następnie składania lub organizowania tych informacji tak, aby utworzyć historię stanów. Historia stanów skonstruowana przy pomocy metody symulacyjnej jest w rzeczywistości historią stanów modelu, a nie systemu. Ponieważ jednak model reprezentuje system, można przyjąć, iż tak otrzymana historia jest historią stanów modelowanego systemu. Mając więc system oraz jego model można powiedzieć, że symulacja jest to zastosowanie modelu w celu chronologicznego wygenerowania historii stanów tegoż modelu, która jest uważana za historię stanów modelowanego systemu.

Natomiast modelowaniem symulacyjnym nazywa się zarówno proces budowy modelu systemu rzeczywistego jak i przeprowadzania **eksperymentów symulacyjnych** na tym modelu w celu poznania zachowania się systemu pod wpływem wewnętrznych i zewnętrznych czynników lub dokonania oceny strategii zapewniających funkcjonowanie badanego systemu. W ten sposób modelowanie symulacyjne obejmuje nie tylko budowę modelu lecz także analityczne wykorzystanie modelu do zbadania (poznania lub rozwiązania) określonego problemu.

Modelowanie symulacyjne stanowi niejako skojarzenie metody matematycznej i eksperymentalnej, bowiem budując model opisuje się formalnie mechanizm zachodzących procesów, podobnie jak przy stosowaniu metod matematycznych, natomiast sposób uzyskiwania wyników jest taki, jak w metodach eksperymentalnych z tą tylko różnicą, że proces rzeczywisty zastępuje się symulacją tego procesu na przykład na komputerze. Metoda ta jest z jednej strony swego rodzaju narzędziem pozwalającym sprawdzić praktyczne wyniki badań teoretycznych, z drugiej zaś strony - metodą teoretycznego rozwiązywania niektórych problemów wysuwanych przez praktykę. Można zatem przyjąć, że modelowanie symulacyjne zajmuje pośrednie miejsce między logicznymi i empirycznymi metodami i stanowi wiążące ogniowo miedzy teorią i praktyką.

Zbudowany **model matematyczny musi zostać przeanalizowany w celu sprawdzenia, czy może być zastosowany do opisu badanego systemu**. Jeśli model jest

[1] *Stan systemu (modelu)* – to najmniejsza liczba danych, których znajomość w danej chwili, przy znajomości wielkości wejściowych, począwszy od tej chwili – pozwala jednoznacznie określić stan i wielkości wyjściowe systemu (modelu) w przyszłości.

wystarczająco prosty, istnieje możliwość zbadania działania systemu przy pomocy metod matematycznych i otrzymania w ten sposób rozwiązania analitycznego. Niektóre metody analityczne mogą być jednak niezwykle rozbudowane, co pociągałoby za sobą olbrzymia liczbę obliczeń. Znane są także przykłady, kiedy matematyczna formuła znana jest doskonale, ale jego rozwiązania dalekie jest od prostoty.

W przypadku, kiedy model matematyczny jest znany, a związane z nim obliczenia nie są zbyt skomplikowane, zazwyczaj zaleca się, aby zamiast symulacji wybrać drogę analityczną. Jednakże, wiele systemów ma tak wysoki stopień złożoności, ze ich matematyczne modele wykluczają możliwość analitycznego rozważania. Wtedy badany model musi być analizowany symulacyjnie.

W badaniu procesów zachodzących w systemie rzeczywistym metodami symulacji wyróżnia się dwa podstawowe, równolegle przebiegające procesy:

- symulacja procesu badanego – jako realizacja eksperymentu symulacyjnego;
- obserwacja wartości określonych wielkości procesu symulowanego. Wyniki obserwacji stanowią dane do wnioskowania o wartościach interesujących nas charakterystyk.

Eksperyment symulacyjny

Symulacja procesu rzeczywistego nie dostarczy badającemu żadnych informacji o badanym systemie, jeżeli nie zostanie zorganizowana odpowiednia (zgodnie z celem badań) obserwacja procesu symulacyjnego (modelu symulacyjnego). Stan modelu symulacyjnego oraz czasy znajdowania się systemu w określonych stanach mogą być obserwowane z inną dokładnością niż są generowane w procesie obliczeniowym algorytmu symulacyjnego. Historię stanów (ciąg wartości obserwowanych) na podstawie obserwacji modelu symulacyjnego wyznacza się za pomocą funkcji obserwacji zmiany stanu:

$$\Phi^0 : X = T \times S \rightarrow \hat{Y}$$

gdzie:

X – generowany zbiór zmiennych;
S - zbiór stanów modelu;
T – dyskretne wartości czasu;
$\hat{Y}$ - oznacza zbiór wyników obserwacji określonych wielkości modelu symulacyjnego.

Badanie procesów zachodzących w systemie rzeczywistym w oparciu o ich model symulacyjny można opisać następującą funkcją:

$$G^e = (G, G^o)$$

będącą złożeniem funkcji przejścia i funkcji obserwacji modelu symulacyjnego.

Parę

$$A^e = (X^e, G^e)$$

gdzie: $X^e = X \times \hat{Y}$ nazywamy **algorytmem symulacyjnego badania systemu**.

Algorytm symulacyjnego badania systemu zapisany w języku programowania nazywa się **programem symulacyjnym systemu**. Eksperyment przeprowadzony przy wykorzystaniu algorytmu symulacyjnego badania systemu nazywamy **eksperymentem symulacyjnym**.

Celem eksperymentu symulacyjnego jest **wnioskowanie statystyczne**[2] o wartościach określonych charakterystyk systemu rzeczywistego i związkach zachodzących między nimi. Zaplanowanie i organizacja eksperymentu symulacyjnego, podobnie jak i realizacja eksperymentu na systemie rzeczywistym, należy do podmiotu badającego system.

Podczas eksperymentu symulacyjnego obserwowany jest model symulacyjny systemu, a dokładnie proces (procesy) zachodzące w badanym systemie. Wyniki obserwacji uzyskane podczas eksperymentu symulacyjnego są danymi (empirycznymi) przeznaczonymi do analizy i statystycznego wnioskowania o wartościach charakterystyk badanego systemu i związkach zachodzących między nimi.

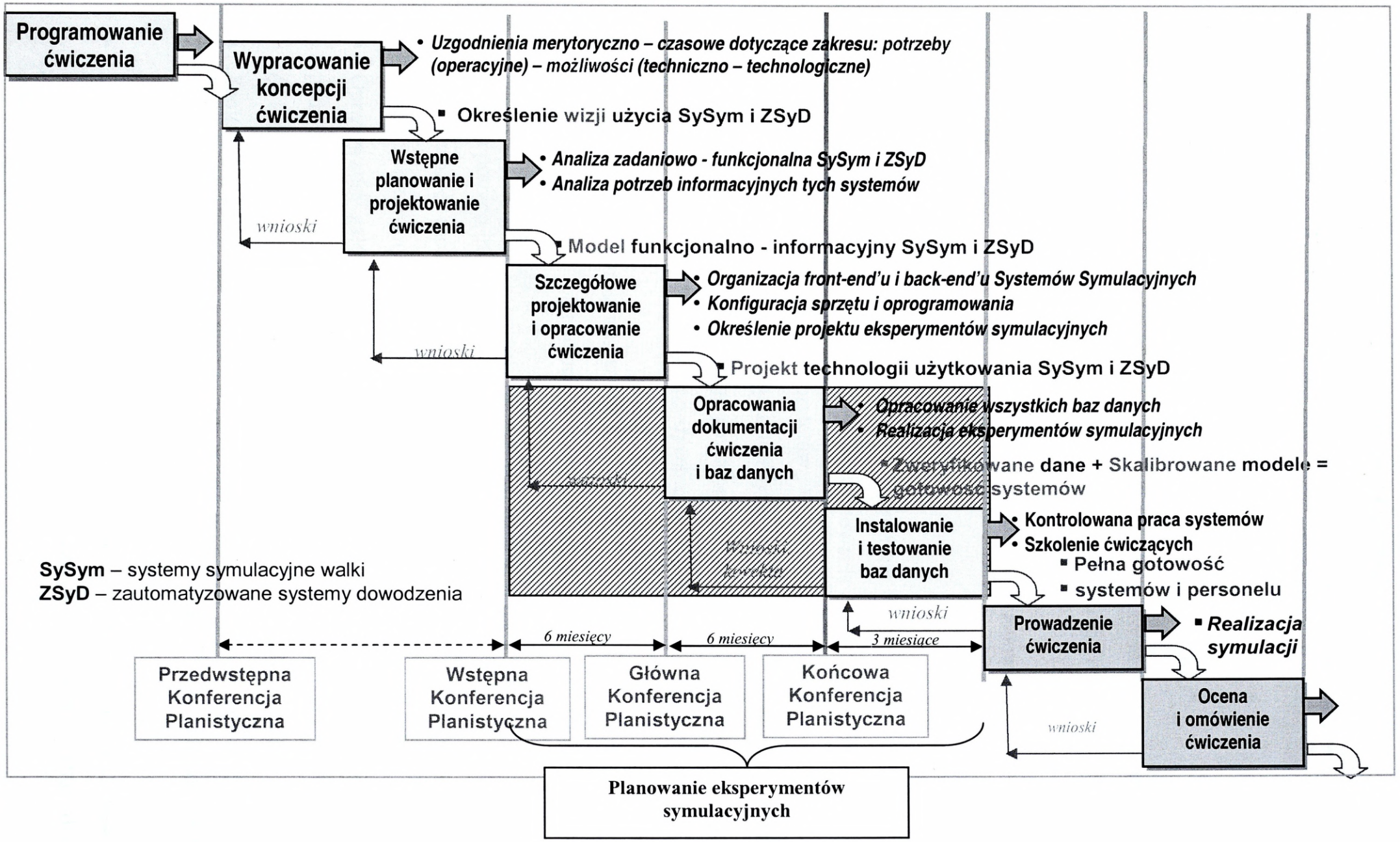
W praktyce oczekuje się, że poprzez eksperymenty symulacyjne osiągnie się tzw. poznawczy stopień wiedzy analitycznej, czyli, że nastąpi odkrycie wewnętrznych przyczyn obserwowanych zjawisk zachodzących w systemie rzeczywistym.

Jednym z ważniejszych **celów planowania eksperymentu symulacyjnego** jest redukcja liczby eksperymentów i przebiegów symulacyjnych bez zmniejszenia jakości uzyskiwanej informacji o systemie.

[2] *Wnioskowanie statystyczne* – to taki rodzaj rozumowania, w którym wniosek o charakterze probabilistycznym (czyli z określonym prawdopodobieństwem) dotyczy nie pojedynczych przedmiotów czy zdarzeń, ale odnosi się do całego zbioru pewnych elementów, nazywanego populacją (generalną). Wniosek tego typu wskazuje zwykle wartość pewnej liczby, charakteryzującej jakąś cechę rozważanego zbioru. Wartości takie nazywają się **parametrami statystycznymi**.

2. Planowanie eksperymentu w ćwiczeniach CAX

Zadania podczas realizacji eksperymentu symulacyjnego muszą być poprzedzone jego przygotowaniem zgodnie z projektem. Poniżej przedstawiono zakres zadań w procesie przygotowania i realizacji eksperymentu symulacyjnego w uwarunkowaniach czasowych przygotowania i prowadzenia ćwiczeń CAX w CSiKGW.



Rysunek 1. Planowanie eksperymentu symulacyjnego w uwarunkowaniach czasowych CAX

2.1. Faza przygotowania eksperymentu symulacyjnego

Procedura realizacji fazy przygotowania ćwiczenia typu CAX realizowana jest w trzech etapach:

ETAP 1. Przygotowanie koncepcyjno – organizacyjne

1. Wypracowanie koncepcji ćwiczenia

- Uzgodnienia merytoryczno – czasowe dotyczące zakresu wspomagania ćwiczeń. Zespolenie: potrzeb (operacyjno -koncepcyjnych) z możliwościami (techniczno – technologicznymi systemów informatycznych),
- Określenie wizji użycia systemów symulacyjnych i zautomatyzowanych systemów dowodzenia.

Pierwsze place planistyczne związane z przyszłymi eksperymentami symulacyjnymi mogą być już rozpoczęte po zakończeniu wypracowania koncepcji ćwiczenia. Wiedza na temat rodzajów działań i ich priorytetów w przyszłym ćwiczeniu CAX może być wykorzystywana do wstępnej oceny aktualnych możliwości w prowadzeniu symulacji w danym zakresie. Szczególna uwaga powinna być zwrócona na te modele symulacyjne i ich parametry, które nie były wykorzystywane, w co najmniej jednym ćwiczeniu CAX.

2. Wstępne planowanie i programowanie ćwiczenia

- Analiza zadaniowo - funkcjonalna systemów symulacyjnych i zautomatyzowanych systemów dowodzenia,
- Analiza potrzeb informacyjnych tych systemów,
- Opracowanie modelu funkcjonalno - informacyjnego systemów symulacyjnych i zautomatyzowanych systemów dowodzenia,
- Opracowanie założeń do konfiguracji systemu symulacyjnego (liczba stanowisk, podział na strony, liczba ćwiczących w ramach strony, planowanie obciążenia ćwiczących – podział dowodzenia w ramach określonego rodzaju działań,
- Zaplanowanie prostych eksperymentów symulacyjnych związanych z działaniami elementarnymi w ramach danego modelu (np. marsz, atak dla pojedynczych jednostek).
- Zaplanowanie złożonych eksperymentów symulacyjnych związanych z działaniami elementarnymi w ramach danego modelu (grupy jednostek, różne działania, oddziaływanie różnych stron).

3. Szczegółowe projektowanie i opracowanie ćwiczenia

- Opracowanie koncepcji organizacji front-end'u i back-end'u systemów symulacyjnych,
- Konceptualizacja konfiguracji sprzętu i oprogramowania,
- Określenie projektu (zakresu i treści) eksperymentów symulacyjnych,
- Opracowanie planu kampanii całego ćwiczenia,
- Dekompozycja planu całego ćwiczenia na etapy oraz następnie na działania w poszczególnych etapach łącznie z wariantowaniem działań,
- Określenie punktów synchronizacji stanów dla działań OPFOR i MSEL,
- Określenie testów poprawności i kompletności danych z punktu widzenia każdego ćwiczącego,
- Określenie punktów i zakresu danych, które będą gromadzone na potrzeby fazy analizy.

ETAP 2. Przygotowanie rzeczowo - technologiczne

4. Projekt technologii użytkowania systemów symulacyjnych i zautomatyzowanych systemów dowodzenia

- Opracowanie dokumentacji (organizacyjnej, operacyjnej) ćwiczenia
- Opracowanie wszystkich baz danych i baz dokumentów formalnych (informatyczna),
- Realizacja eksperymentów symulacyjnych (łącznie z usuwaniem błędów typu statycznego i dynamicznego),
- Analiza operacyjna wyników i kalibracja modeli zgodnie z ocenami ekspertów operacyjnych.
- Opracowanie wszystkich rozkazów (kontroler oraz wszyscy ćwiczący) do realizacji sytuacji wyjściowej ćwiczenia.
- Opracowanie rozkazów z wariantami prawdopodobnych działań ćwiczących
- Opracowanie wariantów działań w postaci rozkazów do synchronizacji działań OPFOR i MSEL z działaniami ćwiczących w ramach planu kampanii. Rozkazy opracowane będą modyfikowane (np. czas wykonania) w rzeczywistym ćwiczeniu odpowiednio do działań ćwiczących i aktualnego stanu symulacji.

Produktem dwóch pierwszych etapów fazy przygotowania ćwiczenia typu CAX są: **zweryfikowane dane + skalibrowane modele = gotowość użytkowa systemów symulacyjnych**

ETAP 3. Przygotowanie operacyjne

5. Instalowanie i testowanie baz danych

- Instalacja i konfiguracja sprzętu i oprogramowania zgodnie z koncepcją,
- Konfiguracja systemu symulacyjnego łącznie z narzędziami do gromadzenia danych dla potrzeb etapu analizy,
- Zorganizowanie i przeszkolenie *back* –end'u systemów symulacyjnych,
- Testowanie funkcji realizowanych przez systemy i zespoły *back* –end'u do poziomu kontrolowanej pracy systemów,
- Szkolenie ćwiczących,
- Testowanie wszystkich rozkazów (kontroler oraz wszyscy ćwiczący) do realizacji sytuacji wyjściowej ćwiczenia,
- Testowanie rozkazów z wariantami prawdopodobnych działań ćwiczących,
- Testowanie wariantów działań w postaci rozkazów do synchronizacji działań OPFOR i MSEL z działaniami ćwiczących w ramach planu kampanii. Rozkazy opracowane będą modyfikowane (np. czas wykonania) w rzeczywistym ćwiczeniu odpowiednio do działań ćwiczących i aktualnego stanu symulacji,
- Testowanie gromadzenia danych dla potrzeb fazy analizy zarówno, co do zakresu, jak i możliwości ich gromadzenia przy pełnym obciążeniu.

Produktem trzeciego etapu fazy przygotowania ćwiczenia typu CAX jest: **pełna gotowość operacyjna systemów i personelu.**

2.2. Faza realizacji eksperymentu symulacyjnego

Faza **prowadzenia ćwiczenia** polega na realizacji eksperymentów symulacyjnych zgodnie z opracowanymi danymi i decyzjami szkolonych oraz korektą kierownictwa ćwiczenia.
Zadania realizowane w tej fazie to:

- uruchomienie procesów symulacji,
- osiągnięcie początkowego stanu symulacji – sytuacja wyjściowej w ćwiczeniu,
- organizacja, nadzór i pomoc ćwiczącym zespołom w pracy na SySym (*front-end*) - wprowadzanie decyzji ćwiczących,

- realizacja wszystkich zadań *back-end*'u (zaplecza technicznego SySym, administratora, koordynatora),
- sterowanie przebiegiem gry zgodnie z projektem ćwiczenia i decyzjami bieżącymi kierownika ćwiczenia,
- realizacja zadań zespołów nie uczestniczących bezpośrednio w ćwiczeniu,
- utrzymanie sprawności technicznej struktury telekomunikacyjnej ćwiczenia,
- dokumentowanie multimedialne ćwiczenia.

Produktem fazy prowadzenia ćwiczenia typu CAX są:

1. bezawaryjne sterowanie pracą systemów symulacyjnych
2. osiągniecie stanu początkowego symulacji – sytuacji wyjściowej dla ćwiczących
3. przesłane do systemu symulacyjnego rozkazy ćwiczących i kontrolera
4. przesłane do ćwiczących komunikaty z systemu symulacyjnego oraz raporty
5. synchronizacja działań ćwiczących z głównymi punktami milowymi ćwiczenia
6. wprowadzenie działań podgrywki ukierunkowanych na osiągniecie założonych stanów symulacji w celu realizacji poszczególnych cząstkowych celów szkoleniowych
7. uzyskanie określonych w koncepcji ćwiczenia wyników symulacji
8. zapis najważniejszych decyzji podjętych w trakcie realizacji ćwiczenia – dziennik symulacji
9. informacje, spostrzeżenia i wnioski z codziennych AAR
10. zgromadzenie i archiwizacja danych dla fazy analizy.

2.3. Faza analizy i interpretacji wyników eksperymentu symulacyjnego

Realizacja fazy oceny i omówienia ćwiczenia polega na przeprowadzeniu następujących przedsięwzięć:

- analiza i ocena organizacji *back-end*'u oraz *front-end*'u systemów,
- analiza wyników eksperymentów symulacyjnych na potrzeby projektowanych ćwiczeń i dla celów naukowo-badawczych,
- analiza i ocena faktycznych potrzeb informacyjnych ćwiczących,
- dostarczenie odpowiednich danych niezbędnych do przeprowadzenia pełnej analizy (adekwatności modeli, poprawności powziętych decyzji) na potrzeby: AAR (ang.

After Action Review) i FER (ang. *Final Exercise Report*), poprzez odpowiednią instalację i konfigurację niezbędnych narzędzi,

- archiwizacja kompletnych danych (rzeczowych i procesualnych) w postaci elektronicznej z przeprowadzonej gry symulacyjnej,
- odtworzenie konfiguracji sprzętu i sprawności oprogramowania na potrzeby prac projektowych i testowych,
- opracowanie kompletnej multimedialnej dokumentacji.

Produktem fazy oceny i omówienia ćwiczenia typu CAX są:

- opracowane wyniki symulacji i ćwiczeń na potrzeby AAR i FER,
- zarchiwizowane bazy (danych i dokumentów) z przeprowadzonego ćwiczenia.

3. Uwarunkowania eksperymentu symulacyjnego.

Eksperyment symulacyjny w rozpatrywanych zastosowaniach (modelowaniu procesów walki) może przybierać różne formy warunkowane jego rozmiarem i zakresem. Jego skrajne formy to:

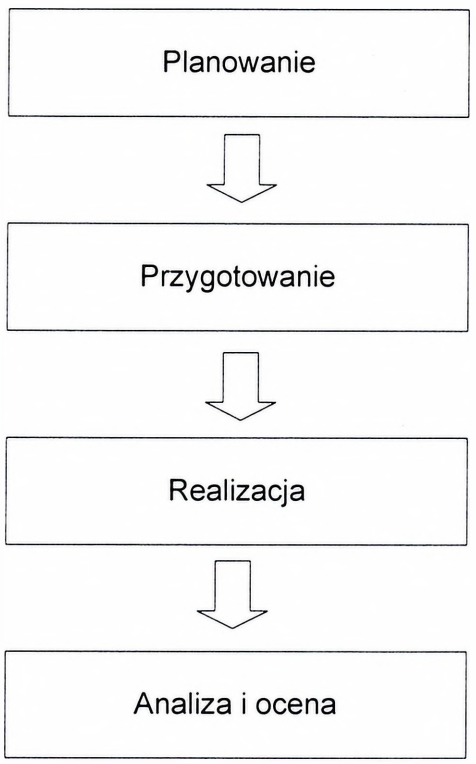
- eksperyment modelujący procesy walki na potrzeby ćwiczeń CAX,
- eksperyment cząstkowy na potrzeby weryfikacji pojedynczego parametru modelowego.

Formy pośrednie eksperymentów będą wynikały z agregowania wyników eksperymentów cząstkowych w coraz większe eksperymenty na podstawie dekompozycji zadań.

Pomimo tak różnych skrajnych postaci eksperymentu symulacyjnego można wyróżnić w nim kilka faz, jakie zawsze nim występują. Fazami takimi będą:

- planowanie,
- przygotowanie,
- realizacja,
- analiza i ocena.

Oczywiście w eksperymentach o różnym zakresie złożoność problemów do rozwiązania będzie znacznie się różniła. Różne będą też zasoby niezbędne do ich przeprowadzenia, jak również potrzebny do tego czas.



Rysunek 2. Fazy występujące w każdym eksperymencie symulacyjnym
/źródło: opracowanie własne/

Eksperyment modelujący procesy walki na potrzeby ćwiczeń CAX będzie przedsięwzięciem znacznie bardziej złożonym i wymagającym dodatkowej fazy, która określi cele, jakie mają zostać osiągnięte w wyniku jego przeprowadzenia. Te z kolei cele będą przekładały się na cele szczątkowe, które będą definiowały zakres eksperymentów, które należy przeprowadzić, aby przygotować całe przedsięwzięcie. Faza taka w przypadku zastosowania podejścia stosowanego przy przedsięwzięciach, które określa się mianem projektu, jest określana zwykle jako faza definicji.

Zastosowanie podejścia do eksperymentu symulacji modelującego procesy walki na potrzeby ćwiczeń CAX jako do projektu powala na wykorzystanie istniejących zasad i technik stosowanych przy zarządzaniu złożonymi przedsięwzięciami. Wykorzystanie ich posiada tą zaletę, że można skorzystać z doświadczeń zgromadzonych w postaci metodyk zarządzania projektami.

Przy takim podejściu eksperyment będzie składał się z następujących etapów:

- **definiowanie** – na tym etapie określone zostaną: cele, uzasadnienie przeprowadzenia całego eksperymentu, produkty tzn. efekty, jakie mają zostać osiągnięte, specyfikacje produktów, zadania, zespoły biorące udział oraz zakresy odpowiedzialności;

- **planowanie** – w tym etapie ustala się: działania, jakie należy podjąć, harmonogramy, dostępne zasoby, określa się ryzyko i szanse powodzenia, plany awaryjne, sposób zapewnienia jakości, dobór ludzi i ich przydział do zespołów;

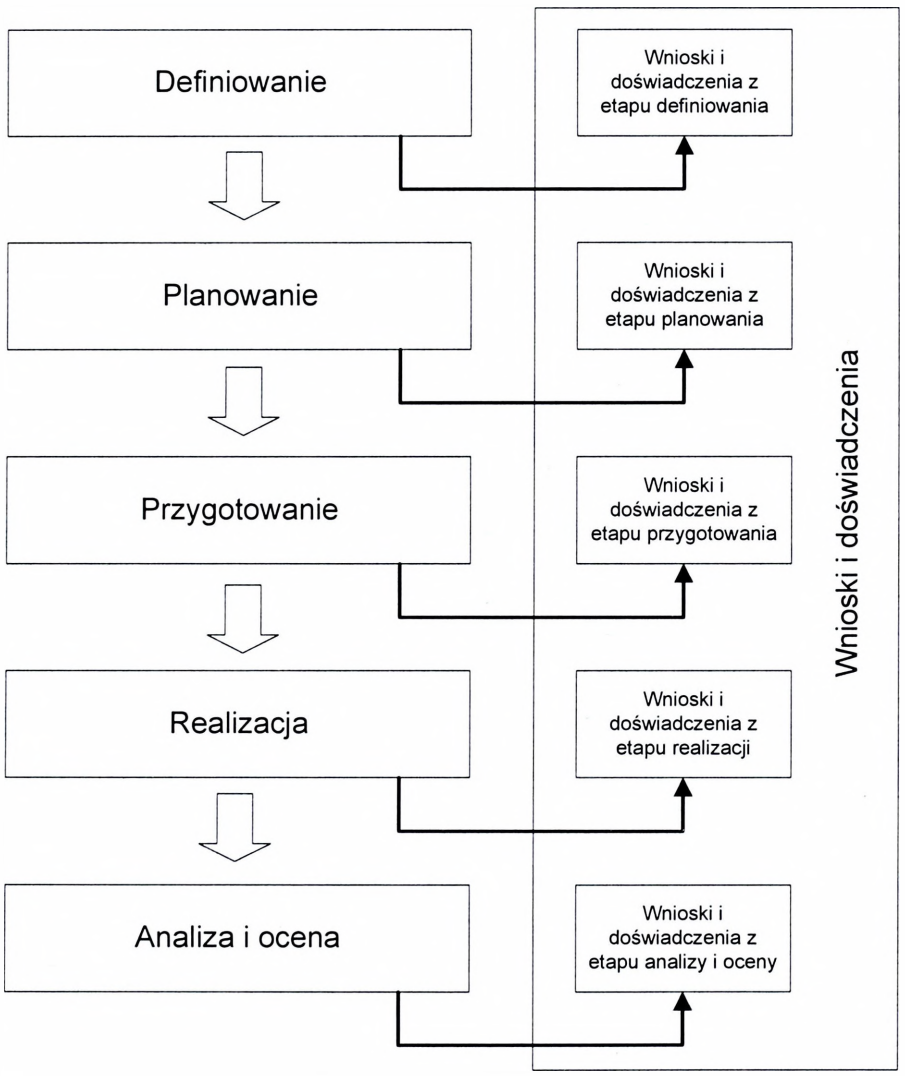
- **przygotowanie** – w tym etapie przeprowadzane są działania niezbędne do pomyślnego przeprowadzenia fazy realizacji, kontrolowane są odchylenia od harmonogramu, weryfikowane są wskaźniki jakościowe otrzymywanych rezultatów, realizowane są przygotowane plany awaryjne. W tej fazie przeprowadzane będą eksperymenty cząstkowe wypracowujące produkty pośrednie, których agregowanie doprowadzi do gotowości do przeprowadzenia fazy realizacji.

- **realizacja** – w tym etapie przeprowadzane są działania związane bezpośrednio z samym procesem modelowania procesów walki w określonym na etapie definicji zakresie;

- **analiza i ocena** – ta faza służy ocenie uzyskanych wyników. Faza ta będzie znacząco różniła się w przypadku eksperymentu symulacji modelującego procesy walki na

potrzeby ćwiczeń CAX i eksperymentu cząstkowego. W pierwszym przypadku jej głównym celem jest ocena wyników uzyskanych przez ćwiczących. Celem dodatkowym będzie oczywiście ocena procesu przeprowadzenia całości eksperymentu – dla eksperymentu cząstkowego będzie to natomiast podstawowy cel.

Kolejnym ważnym pozytywnym aspektem zastosowania takiego podejścia jest fakt, że metodyki zarządzania przedsięwzięciami jako projektami kładą duży nacisk na gromadzenie i dokumentowanie wniosków i doświadczeń, określanych jako „nauczki", zebranych w trakcie ich przeprowadzania tak, aby można było z nich skorzystać w przyszłości.

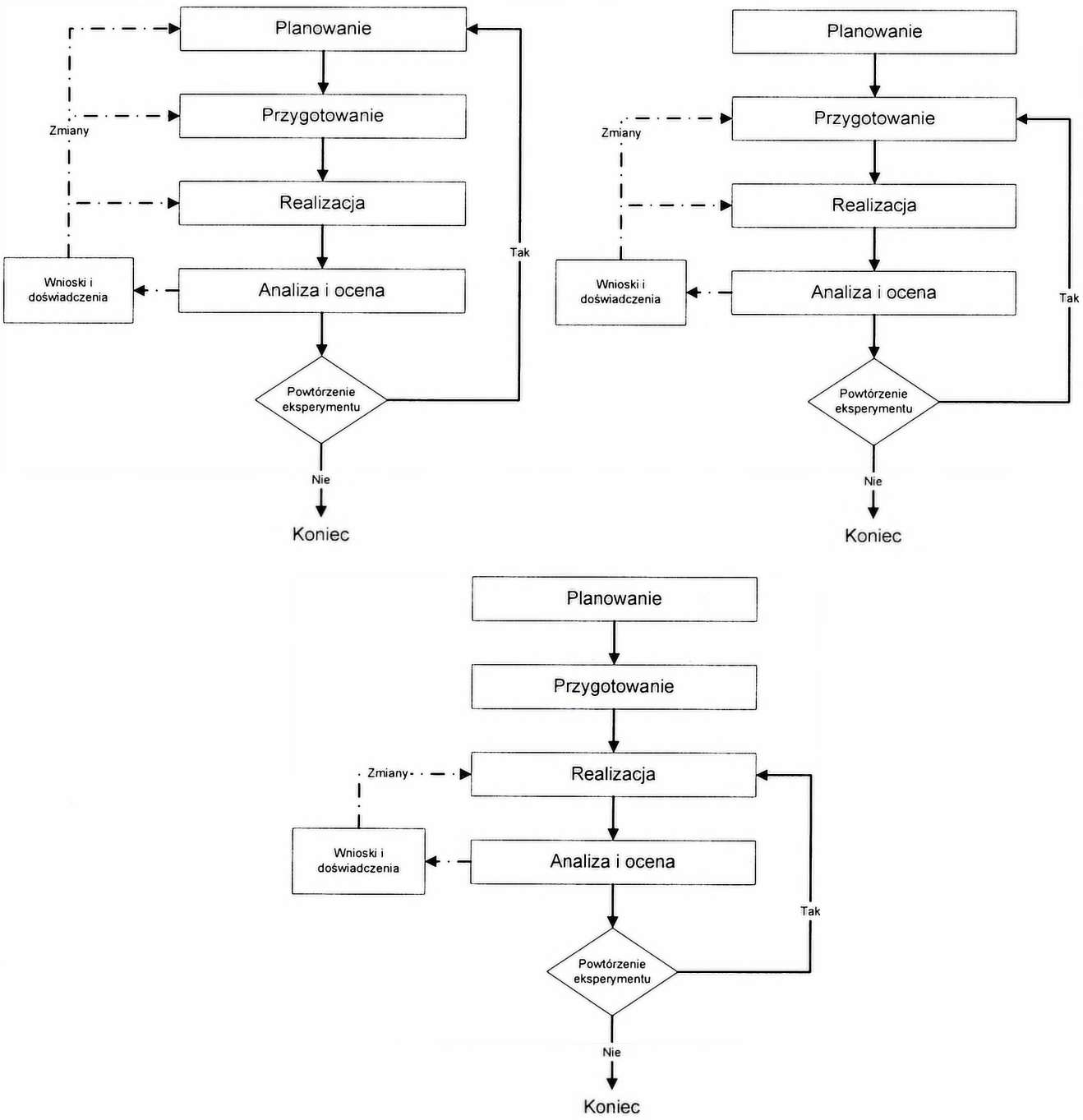


Rysunek 3. Fazy eksperymentu symulacji modelującego procesy walki na potrzeby ćwiczeń CAX
/źródło: opracowanie własne/

Istotne różnice pomiędzy eksperymentem symulacji modelującym procesy walki na potrzeby ćwiczeń CAX a eksperymentem cząstkowym z punktu widzenia planowania to:

- **iteracyjność**;

Iteracyjność w eksperymentach cząstkowych należy przyjąć za ich immanentną cechę, natomiast w przypadku symulacji działań bojowych na potrzeby ćwiczenia CAX nie będzie ona występowała. Trzeba również wziąć pod uwagę to, że kolejne iteracje nie koniecznie będą wyglądały tak samo. Wiąże się to przede wszystkim z procesem gromadzenia doświadczeń.



Rysunek 4. Iteracyjność w eksperymentach cząstkowych
/źródło: opracowanie własne/

- **wykorzystanie wniosków i doświadczeń**;

W przypadku eksperymentu cząstkowego wnioski i doświadczenia mogą być wykorzystywane niemal natychmiast do poprawienia efektów otrzymywanych w trakcie

eksperymentu lub efektywnego przeprowadzania samego eksperymentu. W przypadku eksperymentu na potrzeby ćwiczeń CAX zebrane doświadczenia mogą zostać wykorzystane dopiero przy następnym ćwiczeniu.

- **występowanie fazy definicji**;

Faza definicji dla eksperymentu cząstkowego nie będzie występowała, ponieważ wszystko co jest potrzebne do zaplanowania eksperymentu zostanie (a przynajmniej powinno zostać) zdefiniowane na poziomie fazy definicji i planowania dla całego eksperymentu na potrzeby CAX.

- **stosunek wielkości faz planowania i przygotowania do fazy realizacji**;

Fazy planowania i przygotowania na poziomie przygotowywania eksperymentu symulacyjnego na potrzeby CAX będą co najmniej równe fazie realizacji – z reguły jednak będą one dłuższe, a w przypadku fazy przygotowania znacznie dłuższe od fazy realizacji. W przypadku eksperymentów cząstkowych to faza realizacji będzie najbardziej czasochłonna.

3.1. Określenie celu i zakresu eksperymentu symulacyjnego

Planowanie etapu realizacji eksperymentu symulacyjnego obejmuje swym zakresem wszystkie zadania od zainicjowania eksperymentu symulacyjnego (określenia jego celu, zakresu i konieczności przeprowadzenia) poprzez jego przygotowanie, realizację i analizę uzyskanych wyników.

Na etapie planowania ważnym elementem jest określenie zasobów do realizacji eksperymentu symulacyjnego. Oprócz zasobów sprzętowych i programowych najważniejszym zasobem są zasoby ludzkie. Odpowiednio dobrany zespół specjalistów jest gwarantem sprawności przeprowadzenia eksperymentu, wiarygodności uzyskanych wyników oraz analiz.

Zespół prowadzący eksperymenty symulacyjne powinien się składać, z osób o różnych specjalnościach. Tworzenie zespołów opartych o specjalistów opartych jednej tylko specjalności zawsze prowadzi do braków i błędów eksperymentu symulacyjnego w zakresie pominiętych specjalności. Większość centrów symulacji posiada, co najmniej trzy różne jakościowo zespoły lub ich odpowiedniki:

- zespół operacyjny (ZO);

- zespół analiz (ZA);
- zespół symulacji (ZS).

Tabela 1 przedstawia najważniejsze elementy, które powinny zostać uwzględnione na etapie inicjowania, planowania eksperymentu symulacyjnego oraz zakres odpowiedzialności poszczególnych zespołów.

Tabela 1. Zadania i odpowiedzialności zespołów w planowaniu eksperymentu symulacyjnego

Lp.	Zadanie	Zespoły realizujące	Uwagi
1	Określenie celu i zakresu	ZO	
2	Przydział zasobów	ZO, ZA, ZS	Zasoby sprzętowe, ludzkie.
2	Założenie operacyjne	ZO	W przypadku złożonych eksperymentów rozbicie na etapy
3	Wybór scenariusza	ZO, ZS	
4	Dokumentacja podstawowych zmiennych i parametrów	ZS, ZO	
5	Konfiguracja sprzętowo-programowa	ZS	Server symulacji, stacje robocze, terminale
6	Konfiguracja systemu symulacyjnego	ZS, ZO	CEP, GENIS GIAC, MPP, IMT, drukarki
7	Przygotowanie rozkazów do eksperymentu	ZO, ZS	
8	Punkty zbierania danych	ZO	Czasowe, zdarzeniowe.
9	Określenie narzędzi do zbierania danych	ZA, ZS	
10	Ustalenie wymagań do zobrazowania wyników symulacji dla danego typu eksperymentu symulacyjnego	ZO, ZA	Ewentualne opracowanie narzędzi
11	Przeprowadzenie założonej liczby eksperymentów symulacyjnych	ZS, ZA	Łącznie z zebraniem danych dla fazy analizy
12	Opracowanie wyników symulacji	ZA	
13	Analiza wyników symulacji oraz ocena uzyskanych wyników	ZO	
14	Identyfikacja przyczyn, które mogły spowodować błędy	ZO, ZA, ZS	
15	Powtórzenie eksperymentu symulacyjnego	ZO, ZA, ZS	W przypadku nie zrealizowania celów eksperymentu symulacyjnego.

3.2. Ograniczenia czasowe eksperymentu symulacyjnego

Eksperyment symulacyjny często jest jednym z wielu eksperymentów symulacyjnych realizowanych w ramach jakiegoś większego przedsięwzięcia np. przygotowania bazy danych do ćwiczenia CAX. Nawet realizacja samodzielnych eksperymentów symulacyjnych powinna

kończyć się określonymi wnioskami, co do zmian w określonych bazach danych scenariuszy lub zaleceń dotyczących wykorzystania możliwości systemu symulacyjnego (lub wskazania ograniczeń w wykorzystywanych modelach symulacyjnych). W takich przypadkach, eksperyment (także pod względem czasowym) stanowi część całego przedsięwzięcia – harmonogramu jego realizacji. Innym źródłem ograniczeń czasowych jest dostępność zasobów do realizacji eksperymentu symulacyjnego – przede wszystkim zasobów ludzkich.

Ze względu na ograniczenia czasowe eksperymentu należy dążyć do wypracowania określonej organizacji oraz techniki prowadzenia eksperymentów symulacyjnych.

3.3. Powtarzalność wyników eksperymentu symulacyjnego

Wymagalność powtarzalności warunków przeprowadzenia eksperymentu symulacyjnego jest powodowana wieloma czynnikami. Pierwszym z nich jest zapewnienie warunków, które umożliwią wykonanie kolejnej iteracji tego samego eksperymentu symulacyjnego przy tych samych warunkach, z tymi samymi parametrami symulacji, zgodnie z takimi samymi zestawami rozkazów kontrolera i ćwiczących. Oczywiście świadoma zmiana parametrów i rozkazów w kolejnym eksperymencie symulacyjnym zmienia opisane wyżej warunki realizacji eksperymentu symulacyjnego. Jednak zachowuje wszystkie pozostałe warunki i parametry, których nie zmieniono. Zapewnienie powtarzalności warunków przeprowadzenia systemu symulacyjnego wymaga określonych działań zarówno organizacyjnych, jak i technologicznych. Z punktu widzenia technologii prowadzenia symulacji należy zapewnić archiwizację wszystkich istotnych elementów składowych eksperymentu symulacyjnego:

- bazy danych scenariusza eksperymentu symulacyjnego,
- rozkazów kontrolera (łącznie z rozkazami początkowymi oraz rozkazami SET),
- rozkazów ćwiczących,
- harmonogramem wysyłania rozkazów (w postaci grup rozkazów),
- konfiguracji eksperymentu symulacyjnego (sprzętowej, programowej, systemu symulacyjnego oraz narzędzi gromadzenia danych na potrzeby etapu analizy).

Zagadnienie zapamiętywania rozkazów kontrolera i ćwiczących w grupy rozkazów przedstawione zostało w dalszej części pracy.

4. Planowanie organizacji tworzenia bazy danych

Tworzenie bazy danych na potrzeby przeprowadzenia gry wojennej jest złożonym procesem. Potrzebny jest plan zarządzania tym przedsięwzięciem dla zapewnienia, że wymagania dotyczące planów i analiz sponsora są spełnione przez bazę wykorzystywaną przez model.

Krytycznym jest, żeby plan przygotowywania bazy danych jasno ustalał, co zamierza osiągnąć sponsor w wyniku przeprowadzenia gry wojennej. Jasne określenie tego celu jest szczególnie ważne w przypadku, kiedy sponsorzy bazy danych i zespół przygotowujący bazę danych są z różnych organizacji. Wymagania sponsora kształtują gromadzenie, agregację i wprowadzanie informacji do bazy danych. Cele sponsora mogą również służyć jako wytyczne, jeśli założenia dotyczące wartości danych są potrzebne w miarę kontynuowania procesu przygotowania bazy danych.

Plan przygotowania bazy danych musi również uwzględniać, na kiedy baza ma być gotowa oraz zawierać okresowe przeglądy postępu prac dokonywane przez uczestników zespołu przygotowującego bazę danych. Każdemu takiemu przeglądowi (ew. z wyjątkiem pierwszego) powinien towarzyszyć test bazy danych. W idealnym przypadku przygotowujący bazę i jej użytkownik powinni być ulokowani blisko siebie. Jeżeli nie jest to możliwe, reprezentanci zespołu przygotowującego bazę danych umieszczeni w siedzibie jej użytkownika, mogą poprawić koordynację.

4.1. Zespół przygotowujący bazę danych

Najlepszy zespół do budowy bazy danych składa się z analityków z różnych dziedzin, którzy wywodzą się z personelu analizy technicznej różnych organizacji lub pracowali w organizacjach posiadających doświadczenie w pracy z modelem – np. personel J-3 lub J-4, który wykorzystywał model. Pomimo tego, że programy do przygotowania danych są zaprojektowane pod kątem wykorzystywania ich przez personel operacyjny, eksperci techniczni, którzy rozumieją, w jaki sposób działa model, również powinni znaleźć się w zespole.

Ogólnie rzecz biorąc, tworzenie bazy danych będzie wymagało ekspertów z zakresu terenu, operacji powietrznych (w przypadku systemu „Złocień" dotyczy to tylko lotnictwa

wojsk lądowych), sił lądowych, logistyki i rozpoznania. Każdy członek zespołu jest odpowiedzialny za specyficzny obszar od chwili, kiedy wymagania na potrzeby symulacji zostały zidentyfikowane do chwili, kiedy baza zostanie przetestowana i dostarczona użytkownikowi. Ważnym jest, aby w formowanym zespole umieścić ludzi z doświadczeniem w modelowaniu, symulacji i prowadzeniu gier wojennych, ponieważ ich doświadczenie pozwala im na spojrzenie na dane modelu w ich najczystszym znaczeniu.

Wysiłki zespołu powinny być ukierunkowane przez plan przygotowania bazy danych, który zarysowuje, kto jest odpowiedzialny za co oraz kiedy określone czynności muszą być zakończone.

W celu uniknięcia błędów, wprowadzanie danych najlepiej jest przeprowadzać w postaci uporządkowanego (zorganizowanego) procesu. Jako generalną zasadę należy przyjąć, że jedna osoba jest odpowiedzialna za wprowadzanie danych, dokładność danych jest poprawiana, jeżeli wspomagają ją eksperci dziedzinowi, którzy gromadzą dane. Zaleca się, aby za zapewnienie jakości wprowadzania danych odpowiedzialni byli eksperci z poszczególnych dziedzin wchodzący w skład zespołu oraz przeprowadzenie końcowego zatwierdzenia danych przez zespół zapewnienia jakości składający się z ekspertów z poszczególnych dziedzin, którzy wchodzili w skład zespołu przygotowującego bazę danych oraz ekspertów operacyjnych z organizacji sponsora. Wprowadzanie danych, zapewnienie jakości i testowanie bazy danych jest procesem iteracyjnym.

Skład zespołu przygotowującego bazę danych powinien pozostać niezmieniony w trakcie całego procesu przygotowywania bazy danych. W idealnym przypadku, że względu na krzywe uczenia się występujące w tym złożonym procesie, ten sam zespół powinien być wykorzystany do opracowania wielu baz danych.

4.2. Informacje potrzebne od użytkownika

Informacje potrzebne od użytkownika zawierają ogólny scenariusz oraz siły lądowe, powietrzne i morskie, których działania będą symulowane. Potrzebne są również szczegółowe informacje w celu osiągnięcia założonego poziomu modelowania sił w przypadku systemu JTLS (np. batalionu lub brygady), typy statków powietrznych oraz systemów walki wojsk lądowych, których użycie zakłada się oraz zawartość wstępnie umiejscowionych składów

zaopatrzenia, które mają być reprezentowane. Wstępne spotkania koordynacyjne powinny powiązać ekspertów z poszczególnych dziedzin z ich odpowiednikami w sztabach użytkowników bazy danych.

W przypadku systemu JTLS funkcja weryfikacji SIP (czyli SVP) powinna być często wykorzystywana.

System „Złocień" na obecnym etapie rozwoju nie oferuje funkcjonalności analogicznej do SVP, co może powodować duże trudności w weryfikacji scenariusza i opóźnić jego realizację. Część błędów, które są wykrywane przez JTLS w trakcie weryfikacji scenariusza, w systemie „Złocień" jest wykrywana podczas ładowania scenariusza do symulatora, inne natomiast uwidocznią się dopiero w trakcie symulacji.

4.3. Wymagania dotyczące danych narzucane przez model

Osoba przygotowująca bazę danych musi rozumieć w jaki sposób działa model, ponieważ to determinuje sposób organizacji struktur danych i/lub ich agregacji dla poszczególnych pozycji w bazie danych. Często rzeczywiste parametry, jak na przykład wskaźniki zużycia paliwa, które zostały zgromadzone w warunkach testowania sprzętu, mogą być różne od tych, które są używane przez model. Jeżeli zostaną wprowadzone bezpośrednio do bazy, rzeczywiste wskaźniki mogą prowadzić do osiągnięcia nieprawidłowych wyników. Aby zapobiec temu, rzeczywiste wartości często muszą być dostosowywane tak, aby spełniać założenia scenariusza lub wymagania modelu, wówczas uzyskiwane są realistyczne wyniki.

Członkowie zespołu muszą także rozumieć wzajemne oddziaływanie na siebie elementów danych wewnątrz modelu. W przypadku systemu JTLS jest to przedstawione w dokumentacji "*JTLS Analyst's Guide*", w przypadku systemu „Złocień" dokumentacja taka nie jest dostępna na obecnym etapie rozwoju systemu. Na przykład symulowana prędkość jednostki lądowej w JTLS bierze pod uwagę zarówno ogólną średnią prędkość jednostki, jak i reprezentację terenu (na bazie siatki heksagonalnej), po którym porusza się jednostka. W systemie „Złocień" odwzorowanie przemieszczania się jednostki wykorzystuje reprezentację zarówno siatki prostokątów odwzorowujących teren, jak i model sieci kolejowo-drogowej.

Przygotowywanie bazy danych jest długotrwałym procesem i w związku z tym, jeżeli pojawi się nowa wersja modelu lub zmienią się cele scenariusza, członkowie zespołu muszą

być przygotowani do dołączenia nowych elementów danych do całkowicie lub częściowo ukończonej bazy danych.

4.4. Dokumentowanie bazy danych

Gdy dane zostaną zebrane i wprowadzone, zarówno ich źródła jak i założenia i kalkulacje dotyczące tych danych muszą być udokumentowane, aby później można było je wykorzystać. Ta dokumentacja jest ważna, ponieważ ustanawia ona poziom bazowy, względem którego baza może być modyfikowana na podstawie doświadczenia zdobytego przy jej tworzeniu, źródeł, założeń lub scenariusza wykorzystanego do przygotowania jej zmiany. Jeżeli baza jest niejawna, powinien odpowiadać jej dokument niejawny lub oddzielny niejawny załącznik.

4.5. Testowanie bazy danych

Baza danych musi zostać gruntownie przetestowana zanim zostanie dostarczona użytkownikowi. Testowanie powinno gwarantować, że symulacja działa prawidłowo zarówno z technicznego jak i funkcjonalnego (modelowanych działań) punktu widzenia.

Testowanie powinno być procesem iteracyjnym składającym się typowo z trzech faz. Każdy test jest kulminacją głównej części procesu budowania bazy danych. Test składa się z przeglądu przez zespół zapewnienia jakości: założeń modelowych i przyjętego poziomu abstrakcji opracowanych przez zespół przygotowania bazy danych, przeglądu zebranych czy też wygenerowanych dotychczas danych oraz, jeżeli te dwie części zostały zaakceptowane, rzeczywiste uruchomienie modelu przez kilka dni.

Warto zauważyć, że przeprowadzenie testu bazy danych wymaga wyszkolonych i doświadczonych graczy. Zespół graczy składający się członków zespołu tworzącego bazę danych oraz członków zespołu zapewnienia jakości użytkownika bazy danych, powinien być zorganizowany w sposób, który pozwoli na sprostanie tym wymaganiom.

Poniżej zostaną omówione trzy etapy testowania bazy danych, jakie zalecane są na podstawie doświadczeń dla bazy przygotowywanej dla systemu JTLS. W ogólnym zarysie takie podejście może zostać zastosowane również do systemu „Złocień" ze względu na uporządkowanie od strony wyraźnie postawionych celów. Szczegóły postępowania są jednak przedstawione tylko dla systemu JTLS ze względu NATO, że w przypadku systemu

„Złocień" brak jest jeszcze doświadczeń, które pozwoliłyby na dokładne opracowanie takich kroków.

4.5.1. Test 1 bazy danych – eksperyment elementarny

Pierwsza faza powinna być ukierunkowana na weryfikację danych dla specyficznego obszaru funkcjonalnego. Przynajmniej jedna jednostka każdego prototypu (TUP, SUP, HUP), jaki będzie występował w bazie danych, powinna być dostępna dla testerów w celu przeprowadzenia pierwszego testu bazy danych. Przynajmniej jeden obiekt każdej kategorii i podkategorii, jakie będą zawarte w bazie powinny być dostępne dla tego testu. Pierwszy test powinien sprawdzić specyficzne funkcje w konfiguracjach „jeden na jeden".
Powinny zostać przeprowadzone testy zarówno ofensywnych i defensywnych, jak i wspomagających misji lotniczych. Jednostki przeciwstawnych sił powinny zostać przeniesione w odizolowane lokalizacje w celu nawiązania walki wg modelu Lanchestera. Powinny zostać przeprowadzone małe operacje desantu morskiego, ostrzał artyleryjski wybrzeża, wystrzelenie rakiet, typowe zadania małych jednostek wydzielonych (ang. *HRU*) oraz działania zaopatrzenia. Wszystkie te działania powinny zostać przećwiczone.

W tej fazie wszystkie działania powinny być zazwyczaj przeprowadzane w izolacji, lub w ostateczności przy bardzo uważnie kontrolowanych ustawieniach, tak aby umożliwić zespołowi testowania bazy danych oszacowanie, czy dane są poprawne na najniższym najprostszym poziomie.

Pierwszy test bazy danych powinien z prawie całkowitą pewnością ujawnić problemy z bazą danych, które muszą być usunięte przed przejściem do następnej fazy.

4.5.2. Test 2 bazy danych – eksperyment „agregujący"

Drugi test bazy danych powinien zostać zaplanowany w celu przetestowania interakcji w modelu na dużo wyższym stopniu skomplikowania niż test pierwszy. W czasie jego przeprowadzania w bazie danych powinna znajdować się większość jednostek i obiektów. Wszystkie prototypy i podkategorie obiektów powinny być zdefiniowane. Lista broni, która może być przedmiotem niszczenia powinna być kompletna. Zestawy uzbrojenia dla samolotów oraz ich dozwolone użycie powinny być na tyle kompletne, na ile to możliwe.

W trakcie testu 2 bazy danych powinny zostać stoczone bitwy lądowe z udziałem wielu jednostek, w których uczestniczą jednostki wszystkich walczących stron, ogniem wspierającym i zróżnicowanymi zasadami użycia sił. Powinna zostać przeprowadzona mała kampania powietrzna, w trakcie której zastosowane będzie bliskie wsparcie z powietrza bitwy lądowej, głęboka izolacja lotnicza, obrona powietrzna, obezwładnienie obrony powietrznej i odpowiedź obrony powietrznej. Wyniki zawarte w pozycjach tablicy ACP IFF ARRAY powinny być przeanalizowane całkowicie dokładnie podczas tej fazy, jak również powinny zostać przeanalizowane przypadki bratobójczej walki. Są to obszary, na które użytkownicy są często bardzo wyczuleni.

Jeżeli operacje skryte (ang. *covert operations*) lub inne operacje specjalne, które mają być modelowane, powinny zostać przećwiczone podczas tej fazy. Zdolność małych jednostek wydzielonych (ang. *HRU*) do przeprowadzania skrytych misji bez wykrycia jest szczególnym obszarem zainteresowania.

Jeżeli plan kampanii zawiera znaczące operacje desantu morskiego, próby tych operacji powinny rozpocząć się w tej fazie.

Przewidywane operacje marynarki wojennej, w tym działania minowe, ostrzał wybrzeża, walka między okrętami z użyciem rakiet i artylerii okrętowej, morskie i powietrzne patrolowanie akwenów oraz inne operacje powietrzne sił morskich, wszystkie te działania powinny zostać przeprowadzone w środowisku rodzajów sił zbrojnych połączonych sił koalicyjnych, do tego stopnia w jakim są one zawarte w planie kampanii.

Powinno zostać sprawdzone funkcjonowanie systemu automatycznego uzupełniania zaopatrzenia i/lub możliwość zarządzania przez graczy niezbędnymi działaniami zaopatrywania. Gracze logistyczni często zakładają, że automatyczne generowanie zapotrzebowań uzupełnienia zaopatrzenia wyręcza ich w zupełności i nie mają już nic do zrobienia. Test 2 bazy danych jest dobrą okazją do zademonstrowania, że nawet przy automatycznym generowaniu zapotrzebowań uzupełnienia zaopatrzenia, zarządzanie sytuacjami wyjątkowymi jest zadaniem wymagającym zaangażowanie i stanowiącym wyzwanie.

Dokładność TPFDD i wystarczalność zdarzeń LOGIN powinna być dokładnie zbadana podczas tego testu.

Powinny zostać przećwiczone również funkcje rozpoznania. Sprawdzone powinny zostać patrole małych jednostek wydzielonych (ang. *HRU*), misje zwiadowcze oraz rezultaty zdarzeń zewnętrznych – wszystko to powinno zostać sprawdzone.

4.5.3. Test 3 bazy danych – „próba generalna"

Test 3 bazy danych powinien być próbą planu kampanii w takim kształcie, w jakim jest znany w czasie przeprowadzania testu. Próba powinna dotyczyć całej kampanii, w tym również przeprowadzenie działań innych niż walka, jak również wszystkie prawdopodobne zmiany stron przez frakcje oraz spowodowane tym reakcje. Czasami plan kampanii nie jest ukończony przed przeprowadzeniem testu 3 bazy danych. Jeżeli coś takiego będzie miało miejsce, wówczas zespół graczy powinien przewidzieć i oszacować (na podstawie intencji dowódcy) racjonalny plan kampanii i zaimplementować go.

Test 3 bazy danych powinien zakończyć się zatwierdzeniem bazy danych. Zazwyczaj tak się nie dzieje, ponieważ forsowny test ujawni nieprzewidziane braki i niedociągnięcia w bazie danych. Jeżeli tak się stanie, wówczas błędy muszą zostać poprawione i należy przeprowadzić kolejny test bazy danych i tak aż do zakończenia testu sukcesem, nawet jeżeli przeprowadzane testy będą miały charakter nieformalny.

Niezbędnym jest aby baza danych, która ma być uruchomiona w trakcie ćwiczenia, była testowana przynajmniej przez kilka dni przed rozpoczęciem ćwiczeń. Pominięcie tego elementu z pewnością zagrozi pomyślnemu przeprowadzeniu ćwiczenia.

5. Planowanie fazy przygotowania

Faza przygotowania eksperymentu symulacyjnego w stosunku do fazy przygotowania ćwiczenia typu CAX w większości przypadków będzie się zasadniczo różnić. Różnica wynika głównie ze skali przedsięwzięć, jakimi są ćwiczenie CAX oraz eksperyment symulacyjny. Eksperymenty symulacyjne będą budowane w oparciu o już istniejące bazy danych scenariuszy CAX. Nie można jednak wykluczyć tworzenia dedykowanych modelowych, prototypowych baz danych scenariuszy dla złożonych eksperymentów symulacyjnych. W centrach symulacji często tworzona jest tzw. „metabaza scenariuszy", w której gromadzone są doświadczenia z wielu ćwiczeń CAX oraz eksperymentów symulacyjnych. Zgromadzone doświadczenia, to parametry umożliwiające prowadzenie symulacji oraz uzyskiwanie wyników zbliżonych do rzeczywistego teatru działań wojennych. W celu doskonalenia parametrów zawartych w „metabazie" powinna zostać utworzona wspomniana wyżej modelowa baza danych scenariusza – w tym przypadku eksperymentu symulacyjnego. Na takiej bazie modelowej możliwe jest prowadzenie wielu eksperymentów symulacyjnych.

W zależności od skali przygotowywanego eksperymentu symulacyjnego oraz złożoności systemu symulacyjnego zespoły przygotowujące bazy danych scenariuszy będą realizowały różne zadania. W specyficznych warunkach pewne etapy będą mogły zostać pominięte lub będą realizowane w mniejszym zakresie. Poszczególne systemy symulacyjne wykonane w określonej technologii będą umożliwiały realizację poszczególnych zadań za pomocą mniej lub bardziej złożonych i rozproszonych środowisk programowych. W większości przypadków można wyróżnić następujące etapy w przygotowaniu bazy danych scenariusza eksperymentu symulacyjnego.

Etapy przygotowania bazy danych scenariusza:

1. wybór metody i utworzenie scenariusza;
2. wprowadzanie danych i aktualizacje scenariusza;
3. weryfikacja bazy danych scenariusza;
4. konfiguracja systemu symulacyjnego;
5. testowanie potencjalnej możliwości uruchomienia scenariusza;
6. testowanie statycznego i dynamicznego etapu symulacji;
7. archiwizacja plików danych, plików konfiguracyjnych oraz dokumentacji.

5.1. Wybór metody i utworzenie scenariusza

Wybór metody tworzenia bazy danych scenariusza zależy od wielu czynników, ale do najważniejszych należą: posiadanie istniejących (własnych) baz danych scenariuszy, stopień podobieństwa (zgodności) nowego scenariusza z już istniejącymi, doświadczenie zespołu przygotowującego bazę danych oraz od czasu na przygotowanie bazy danych scenariusza.

W systemie JTLS istnieją różne klasyfikacje metod tworzenia nowego scenariusza. W rzeczywistości (z technologicznego punktu widzenia) istnieją dwie podstawowe metody tworzenia scenariusza:

- tworzenie nowego scenariusza z predefiniowanej bazy EMPTY;
- modyfikacja istniejącego scenariusza bazy danych.

Pierwsza metoda tworzenia scenariusza nie wymaga żadnych plików, ponieważ wykorzystywana jest predefiniowana baza EMPTY. Baza EMPTY nie zawiera danych (prototypów, jednostek, obiektów, itd.). Metoda jest rzadko stosowana i nie zalecana przez producenta systemu, zespół tworzący bazę danych będzie musiał zmierzyć się nie tylko z bardzo dużą ilością danych do wprowadzenia, ale i także z całym złożonym modelem danych systemu JTLS.

Druga metoda związana jest z wykorzystaniem już istniejącego scenariusza.

W tej metodzie należy wyróżnić dwa przypadki istniejących bazy danych:

- niepełny scenariusz „BLANK";
- pełne scenariusze - przykładowo standardowych bazy SDBV26, Ratest26, lub baz danych z rzeczywistych ćwiczeń: Canon Cloud, Capable Warrior.

W pierwszym przypadku wykorzystywana jest baza danych scenariusza „BLANK", która jest pełnym scenariuszem JTLS – jeżeli chodzi o liczbę plików, ale jest niepełnym scenariuszem, jeżeli chodzi o ilość danych. Baza BLANK zawiera większość danych modelowych, wraz z prototypami, kategoriami obiektów, itp. Nie zawiera natomiast danych specyficznych dla danego ćwiczenia tzn. jednostek, przydzielonych do nich obiektów, stron, frakcji, itp. Metoda ta jest wykorzystywana w przypadku, gdy nowy scenariusz znacznie odbiega od istniejących, ponieważ wykorzystanie jakiegokolwiek innego scenariusza, przykładowo `SDBV26`, wiązałoby się z koniecznością usuwania danych specyficznych dla danego ćwiczenia (przede wszystkim stron, frakcji, jednostek i obiektów).

W drugim przypadku wykorzystuje się istniejący pełny scenariusz. Metoda ta jest najczęściej stosowania w przypadku dużego podobieństwa nowego scenariusza do scenariusza istniejącego: podobne lub dokładnie takie same jednostki, prototypy, obiekty.

W tej metodzie zespół tworzący bazę danych musi zadecydować, który z dostępnych scenariuszy będzie wymagał mniej prac z dostosowaniem do wymogów nowego scenariusza.

Niezależnie od metody tworzenia nowego scenariusza, tworzenie z bazy EMPTY, czy tworzenie za pomocą polecenia `copyscenario`, należy na serwerze Oracle utworzyć użytkownika o nazwie zgodnej z nazwą nowego scenariusza. Operacja ta jest niezbędna, aby po utworzeniu scenariusza, w postaci plików ASCII, można było wykonać pierwszą operację `Load` (załadowanie danych, wczytanie danych - import) plików do bazy w formacie Oracle. Pierwsza operacja `Load` powoduje utworzenie niezbędnych struktur i obiektów w bazie Oracle takich jak tabele, indeksy itp.

W celu utworzenia nowego scenariusza należy z linii poleceń wydać polecenie:

`copyscenario <istniejący_scenariusz> <nazwa_nowego_scenariusza>`.

5.2. Wprowadzanie danych i aktualizacje scenariusza

Bazy danych współczesnych systemów symulacyjnych są coraz bardziej złożone. Wynika to przede wszystkim ze wzrostu mocy obliczeniowej komputerów, która z kolei umożliwia realizację bardziej złożonych modeli symulacyjnych. Wzrost możliwości obliczeniowych oraz dążenie do maksymalnego odwzorowania rzeczywistego pola walki w współczesnych systemach symulacyjnych muszą nieść ze sobą także zwiększenie wymagań na bazy danych systemów symulacyjnych i to nie tylko ze względu na ich wielkość, ale także na ich złożoność.

Mówiąc o bazie danych mówimy o całości wszystkich zasobów danych związanych ze scenariuszem ćwiczenia. Bazy danych scenariuszy składają się głównie z trzech różnych rodzajów danych:

- dane o terenie;
- dane dotyczące działań bojowych (strony walki, struktury organizacyjne zadania, itp.);
- dane parametryczne.

5.3. Weryfikacja bazy danych scenariusza

Wprowadzenie niezbędnych informacji do bazy danych otwiera możliwość do kolejnego etapu, jakim jest weryfikacja bazy danych scenariusza. Weryfikacja scenariusza nie jest procesem jednorazowym, zwłaszcza w sytuacji, gdy nowy scenariusz powstaje przez modyfikację jednego z wcześniejszych scenariuszy.

Programy wykorzystywane do wprowadzania danych umożliwiają wstępną weryfikację wprowadzanych danych. Pierwszy rodzaj kontroli dotyczy poszczególnych pól: sprawdzanie zakresu wartości, typu, wartości, sprawdzenie czy pole jest wymaganie, czy opcjonalne, itp. Następnie jest wykonywana weryfikacja na poziomie rekordu (spójność relacyjna między tabelami bazy danych).

Wyżej wymieniona weryfikacja danych, na etapie wprowadzania, jest wykonywana, w przypadku korzystania ze standardowych i podstawowych narzędzi wprowadzania danych o scenariuszu dla danego systemu symulacyjnego.

Weryfikacji danych scenariusza na znacznie wyższym poziomie dokonują specjalnie dedykowane programy do weryfikacji całej bazy danych scenariusza. Weryfikacja polega na sprawdzeniu zależności, powiązań, oraz spójności danych (łącznie z wartościami zmiennych) w ramach określonych zakresów danych (odpowiadających zastosowanym w danym systemie symulacyjnym modelom poszczególnych działań). Zwykle wykryte błędy i niezgodności w scenariuszu dzielone są na dwie grupy: błędy krytyczne oraz ostrzeżenia. Brak błędów krytycznych na etapie weryfikacji umożliwia realizacje kolejnych etapów, jakimi są konfiguracja gry oraz jej uruchomienie.

5.4. Testowanie statycznego i dynamicznego etapu symulacji

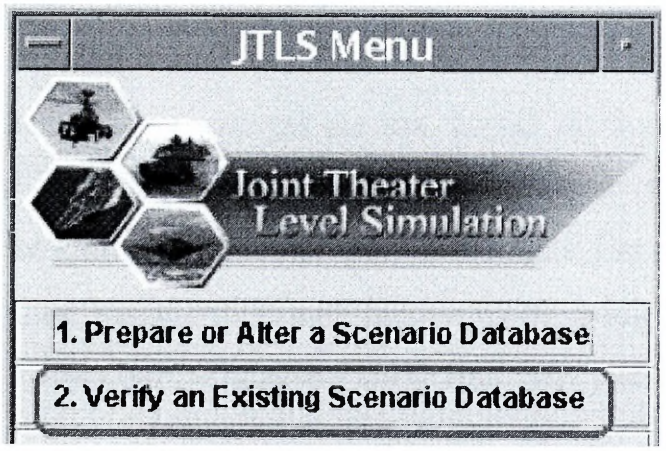
Tworzenie bazy danych nowego scenariusza nie kończy się na etapie usunięcia wszystkich błędów i ostrzeżeń z etapu weryfikacji. Wspomniana wyżej weryfikacja często określana mianem weryfikacji statycznej zapewnia tylko potencjalne uruchomienie systemu symulacyjnego. W pełni przetestowany scenariusz wymaga oprócz testowania statycznego także testowania dynamicznego. Jeżeli testowanie dynamiczne nie zostanie wykonane w fazie przygotowania danych, to niestety będzie musiało nastąpić w trakcie rzeczywistego ćwiczenia. W praktyce wszystkie testy, które wykonywane są po raz pierwszy wskazują na duże różnice między tym, co było w założeniach, a tym, co udało się osiągnąć. Realia ćwiczeń wspomaganych komputerowo wskazują, że weryfikacja dynamiczna powinna być wykonana, w możliwie jak najszerszym zakresie.

Programy weryfikacji scenariusza

Programy weryfikacji scenariusza dokonują weryfikacji bazy danych scenariusza na wyższym poziomie niż programy, które umożliwiły ich wprowadzanie. Najczęściej wykryte błędy i niezgodności w scenariuszu dzielone są na dwie grupy: błędy krytyczne oraz

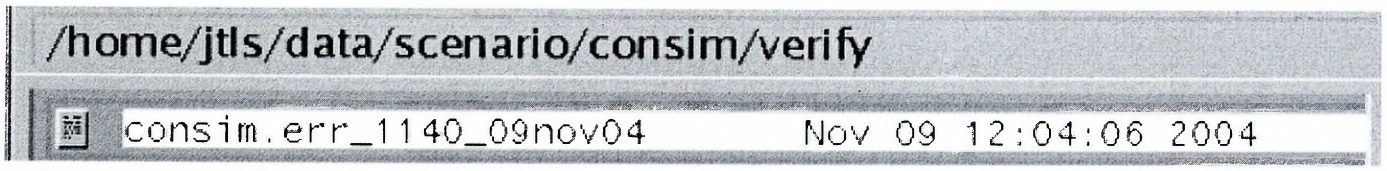
ostrzeżenia. Dodatkowo w ramach każdej z głównych grup tworzone są dodatkowe podgrupy umożliwiające grupowanie błędów według określonych typów. Grupowanie błędów ułatwia podział pracy dla zespołów zajmujących się aktualizacją i usuwaniem błędów w scenariuszu.

W celu weryfikacji scenariusza należy z menu systemu JTLS wybrać punkt menu „Verify an Existing Scenario Database". Danymi wejściowymi dla weryfikacji scenariusza są zbiory danych scenariusza znajdujące się w katalogu jtls/data/scenario/<nazwa_scenariusza>.



Rysunek 5. Weryfikacja scenariusza

Po weryfikacji scenariusza w katalogu jtls/data/scenario/nazwa_scenariusza/verify zostanie utworzony plik zawierający informacje o błędach oraz ostrzeżeniach o możliwych nieprawidłowościach w bazie danych scenariusza.



Rysunek 6. Lokalizacja pliku z błędami i ostrzeżeniami

Wystąpienie błędów (ang. *errors*) wiąże się z dużym prawdopodobieństwem poważnych błędów na etapie uruchamiania symulacji włącznie z tzw. upadkiem systemu (ang. *crash*). W związku z powyższym należy dążyć do sytuacji, aby liczba błędów, w poszczególnych głównych typach danych, była równa zeru.

```
                              CWICZEBNE
                              ---------

Finished Model Parameter Verification:          Errors:      0  Warnings:     23
Finished Prototype  Verification:               Errors:      0  Warnings:    171
Finished Lethality  Verification:               Errors:      0  Warnings:    308
Finished Unit       Verification:               Errors:      0  Warnings:    170
Finished Target     Verification:               Errors:      0  Warnings:      0
Finished TPFDD      Verification:               Errors:      0  Warnings:      0
Finished Event      Verification:               Errors:      0  Warnings:      0
Finished HRU        Verification:               Errors:      0  Warnings:      0

Completed Terrain  Verification:                Errors:      0  Warnings:      0
Completed verification:                  Total  Errors:      0  Warnings:    672
```

Rysunek 7. Nagłówek raportu błędów i ostrzeżeń

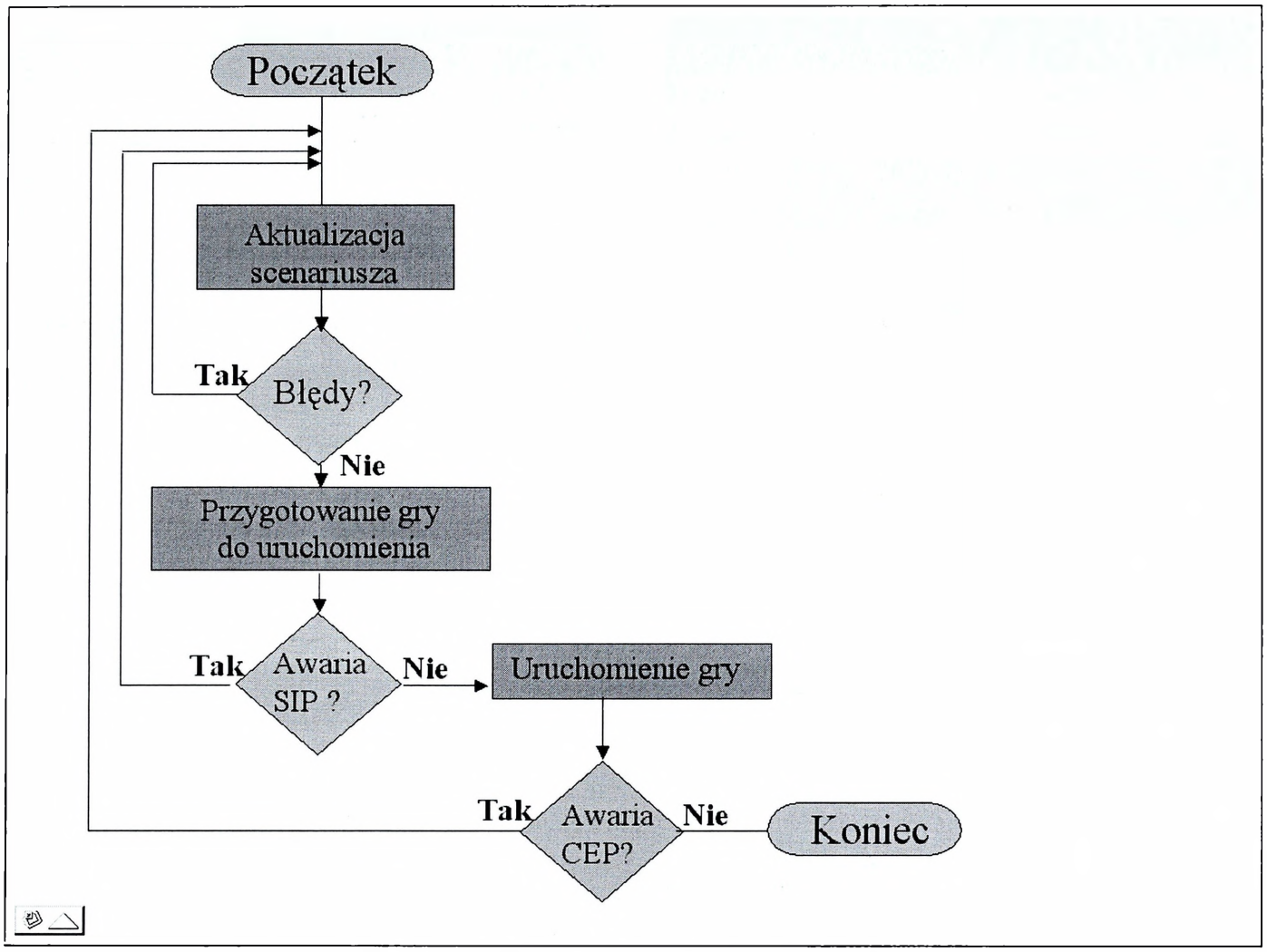
Wystąpienie ostrzeżeń (ang. *warnings*) nie ma, aż tak dużego wpływu na potencjalną możliwość uruchamiania symulacji, jak wystąpienie błędów w scenariuszu. Ostrzeżenie, sygnalizuje pewną nieprawidłowość, która w przypadku, gdy jest niezamierzona, powinna być usunięta. Ostrzeżenia nie mogą być jednak lekceważone, ponieważ mogą prowadzić do upadku programu (ang. *crash*) SIP. Zadaniem programu SIP jest przygotowanie struktury oraz plików danych dla fazy symulacji na podstawie danych opracowanych na etapie przygotowania scenariusza.

Programy i metody wspomagające fazę testowania

Tworzenie bazy danych nowego scenariusza nie kończy się na etapie usunięcia wszystkich błędów i ostrzeżeń z etapu weryfikacji. Wspomniana wyżej weryfikacja często określana mianem weryfikacji statycznej zapewnia tylko potencjalne uruchomienie systemu symulacyjnego. W pełni przetestowany scenariusz wymaga oprócz testowania statycznego także testowania dynamicznego.

W Centrum Symulacji i Komputerowych Gier Wojennych AON do wypracowano własną procedurę przygotowania nowego scenariusza. Z założenia jest to procedura dedykowana dla zespołu, który nie ma wcześniejszych doświadczeń w tworzeniu scenariuszy dla systemu JTLS.

Oprócz pierwszego założenia (zachowanie możliwości uruchomienia symulacji) drugim istotnym założeniem w opracowaniu bazy danych nowego scenariusza jest wprowadzania małych zmian oraz wykonywanie pełnego cyklu przygotowania gry, aż do uruchomienia symulacji.

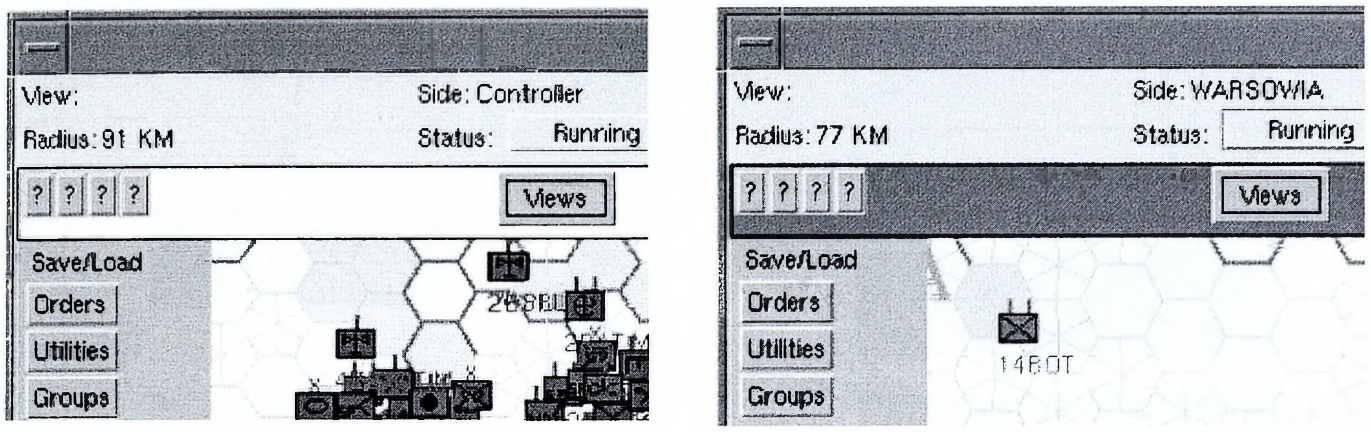


Rysunek 8. Procedura testowania scenariusza
/źródło: opracowanie własne/

Opisana procedura umożliwia szybkie i skuteczne usuwanie błędów na początkowym i statycznym etapie uruchomienia gry. Dokładnie, sprawdzamy w niej możliwość uruchomienia głównych procesów CEP i GENIS oraz programów interfejsu użytkownika. Dodatkowo sprawdzamy, jak zachuje się symulacja po wydaniu rozkazu Game Speed z wartością większą od zera. Zwykle po tym etapie dalsze testowanie polega na wydaniu rozkazów testowych przez kontrolera oraz graczy każdej ze stron. Wydawanie rozkazów i reakcja systemu symulacyjnego rozpoczynają drugi etap testowania scenariusza – etap testowania dynamicznego.

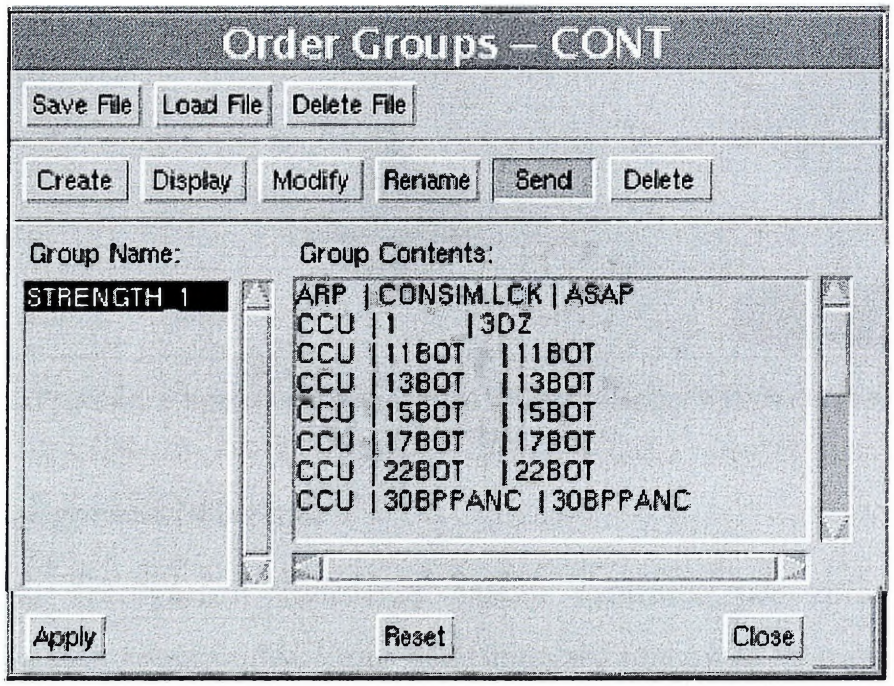
W celu uniknięcia zbędnego i przede wszystkim czasochłonnego wprowadzania rozkazów omawiana procedura została udoskonalona przez wykorzystanie możliwości systemu JTLS do zapisywania, grupowania i wysyłania wcześniej zapisanych rozkazów i grup rozkazów.

Tuż po rozpoczęciu symulacji można dla każdej ze stron oraz dodatkowo dla strony kontrolera wydać grupy rozkazów.

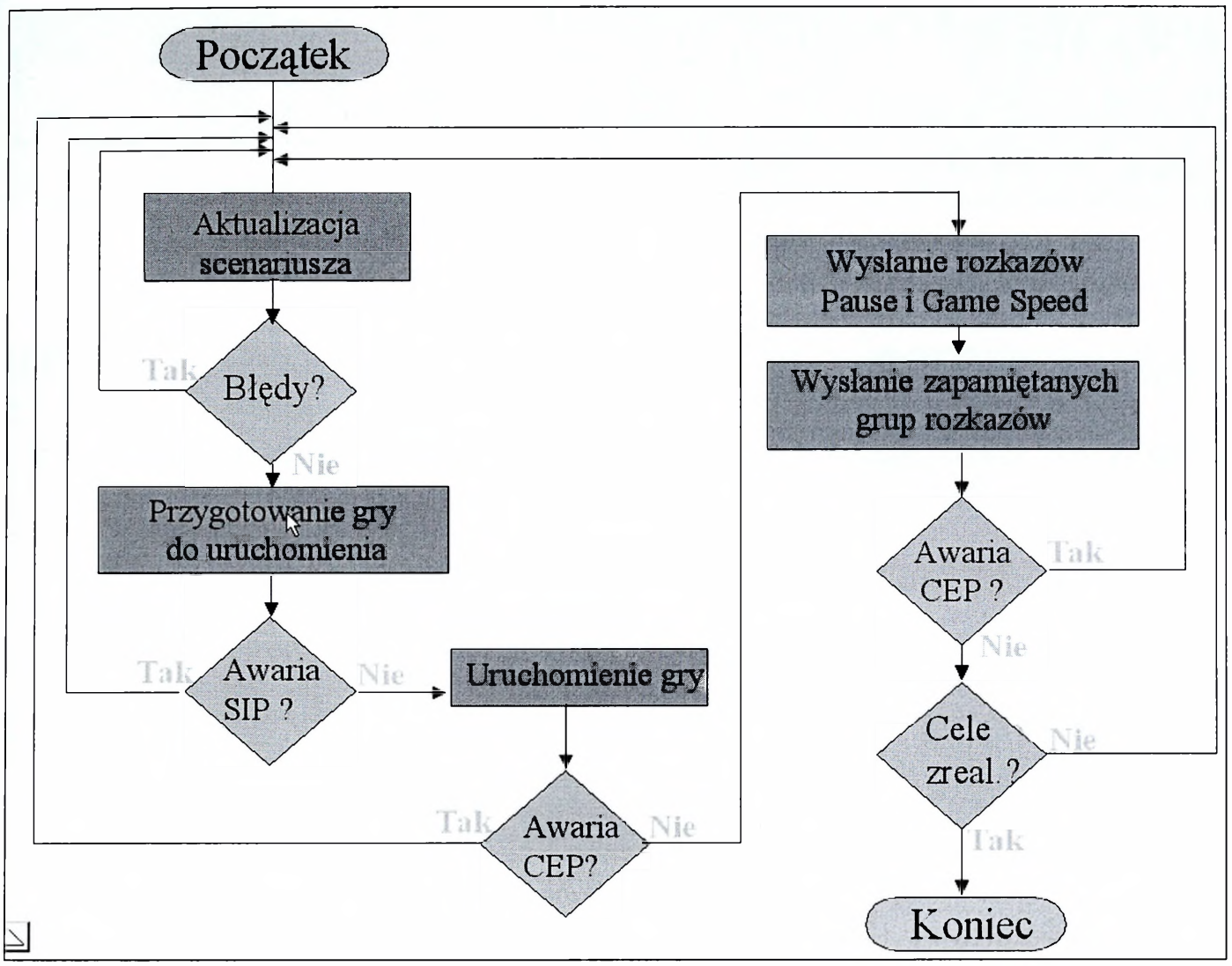


Rysunek 9. Dostęp do grup rozkazów dla każdej ze stron

Przyśpieszenie wydawaniu rozkazów polegające na wydawaniu całych grup rozkazów znacznie przyśpiesza tą fazę testów bazy danych scenariusza.



Rysunek 10. Wysyłanie grup rozkazów

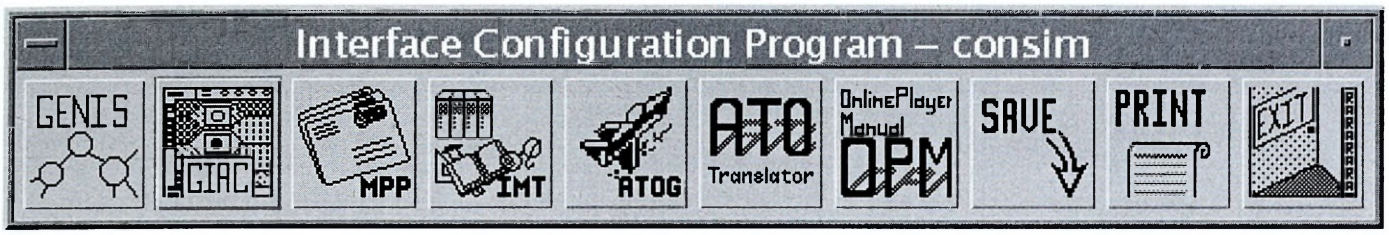


Rysunek 11. Procedura testowania scenariusza uwzględniająca wysyłanie grup rozkazów
/źródło: opracowanie własne/

Programy do konfiguracji ćwiczenia

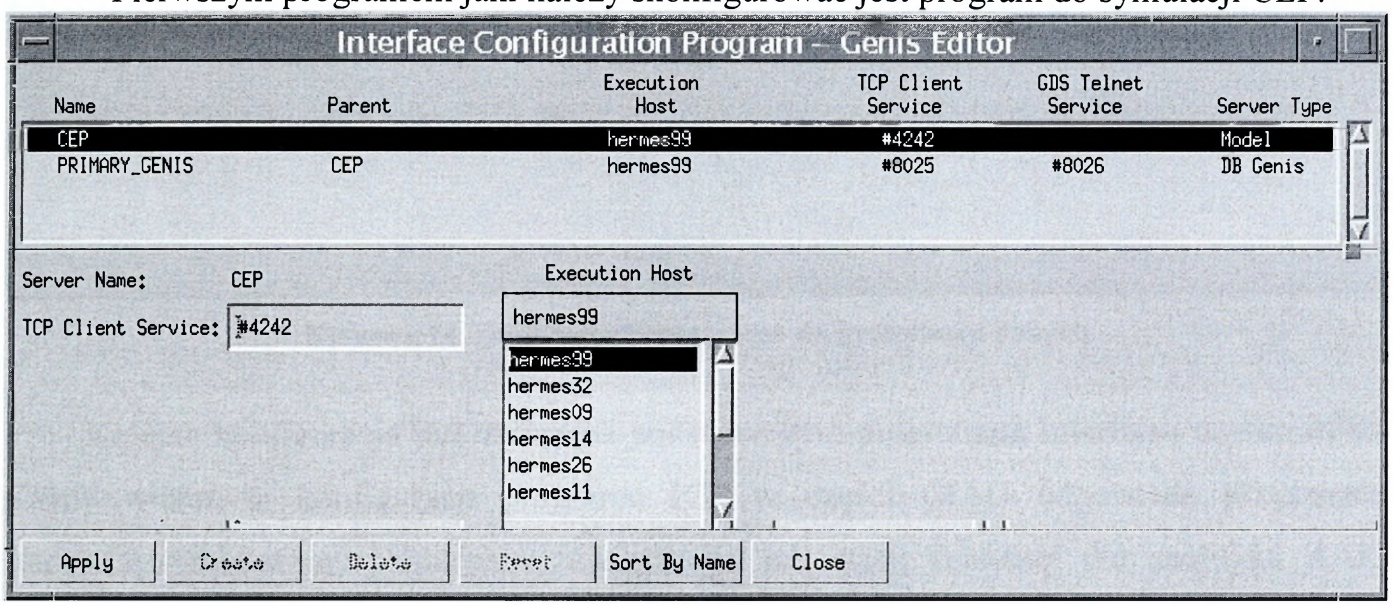
Programy do konfiguracji ćwiczenia należy rozumieć jak zbiór programów do konfiguracji wszystkich programów, które będą musiały pracować w trakcje symulacji. Do grupy programów, które muszą zostać uruchomione w trakcie symulacji należy przede wszystkim zaliczyć główny program symulacyjny, programy do przekazywania wyników symulacji, programy interfejsu użytkownika, ale także wszelkie programy, które w trakcie symulacji zbierają dane o wynikach symulacji dla fazy analizy i opracowania wyników ćwiczenia.

W systemie JTLS do konfiguracji głównych procesów i programów interfejsu wykorzystywany jest zintegrowany „Program do konfiguracji interfejsu" (*ang. Interface Configuration Program*). W programie Rysunek jego głównego menu należy dokonać wyboru procesu lub programu, którego konfiguracja będzie zmieniana, dokonać jego konfiguracji, i zapamiętać konfigurację.



Rysunek 12. Program do konfiguracji programów interfejsowych

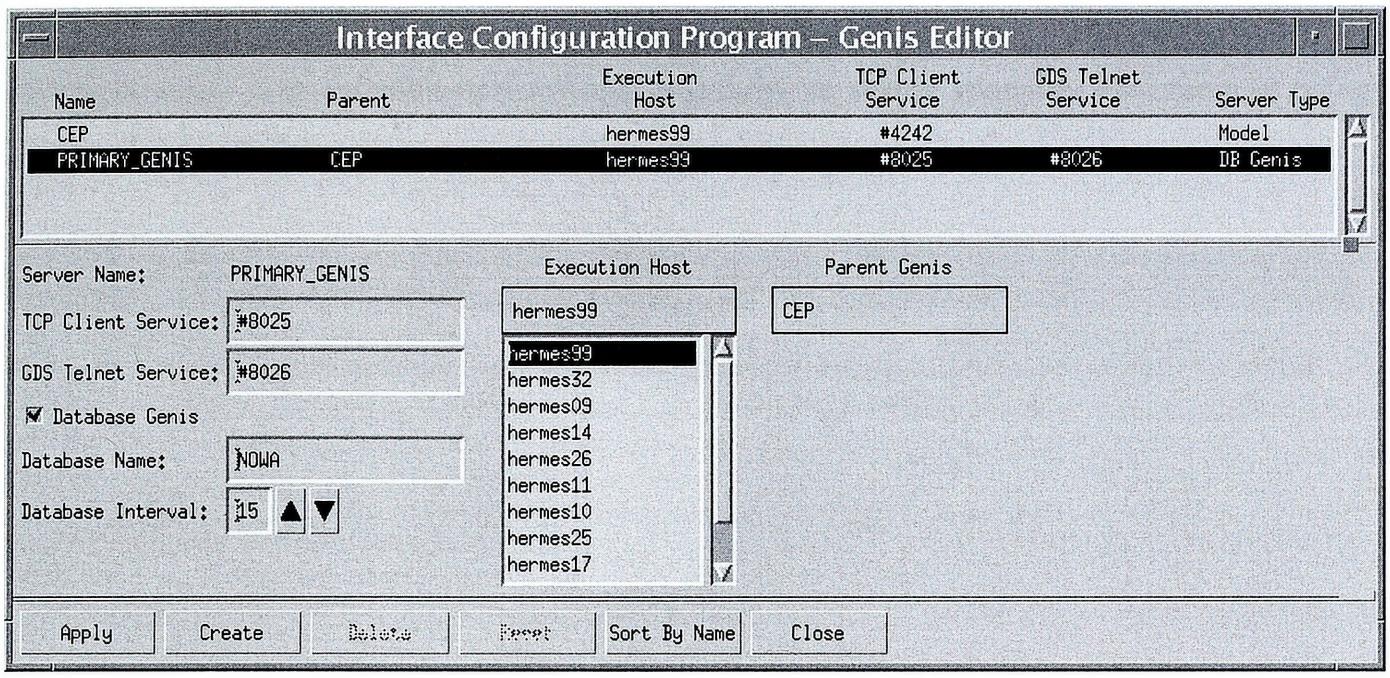
Pierwszym programem jaki należy skonfigurować jest program do symulacji CEP.



Rysunek 13. Konfiguracja programu symulacyjnego

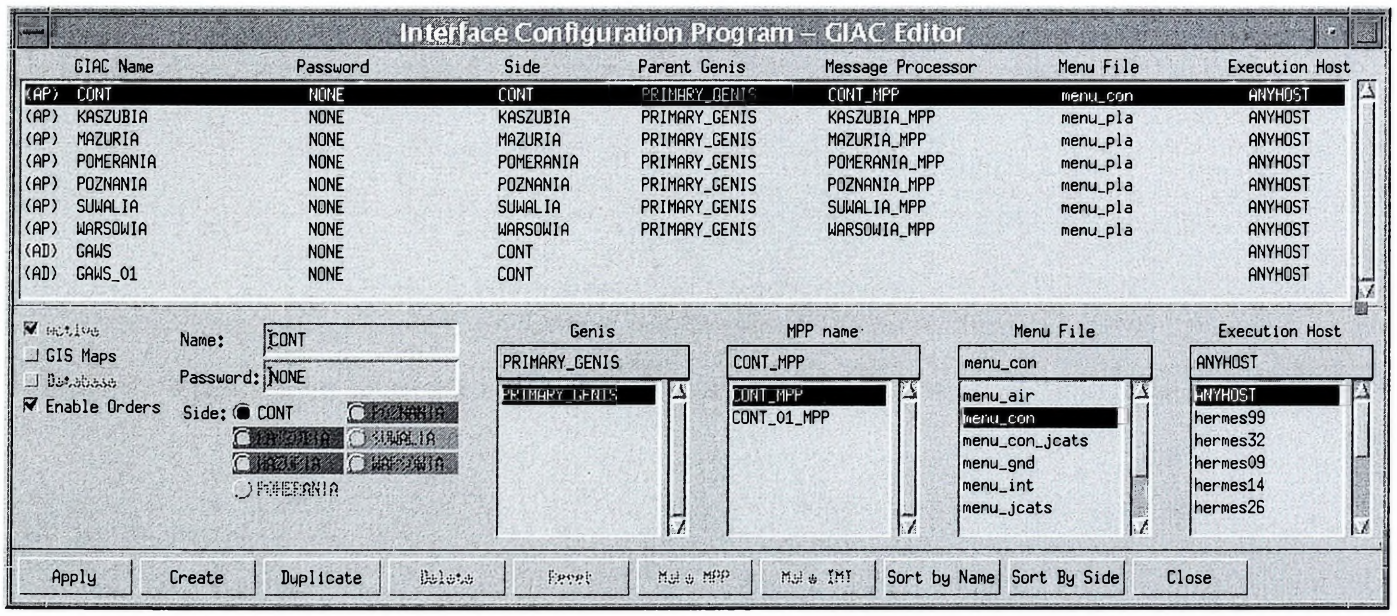
Konfiguracja głównego programu symulacyjnego na etapie programu IPC sprowadza się do wskazania serwera, na którym program symulacyjny zostanie uruchomiony. W tym przypadku ważnym elementem przed rozpoczęciem symulacji jest dokładne sprawdzenie, na jakim serwerze uruchamiany jest program symulacyjny (jeżeli to nie zostanie wykonane, to może się okazać, że główna symulacja rozpocznie się np. na starszej konfiguracji wykorzystywanej w celach testowych).

Drugi program, który należy skonfigurować, to program do dystrybucji danych GENIS. W sztandarowej konfiguracji procesu odpowiadającego za przekazywanie wyników do programów interfejsowych i odbiór rozkazów wykorzystywany jest jeden proces `primary GENIS`. W przypadku, gdy testy wydajności wykażą, że główny GENIS nie nadąża za procesem symulacji należy zdefiniować na innym komputerze dodatkowy GENIS – tzw. `secondary GENIS`. Procesów `secondary GENIS` może być więcej zwłaszcza w przypadku realizacji symulacji w konfiguracji rozległej.



Rysunek 14. Konfiguracja programu do dystrybucji danych

Kolejna konfiguracja jest związana podstawowym programem interfejsu użytkownika. Każdy wiersz w konfiguracji programu ICP w części GIAC odpowiada przyszłemu stanowisku roboczemu: kontrolera, ćwiczącego lub stacji roboczej dla analityka AAR. W konfiguracji bardzo pomocnym jest dokument określający wszystkie wymagania uczestników ćwiczenia łącznie z nazwą ćwiczącego, hasłem oraz menu (które będzie dostępne dla danego gracza w czasie ćwiczenia).



Rysunek 15. Konfiguracja głównego programu graficznego GIAC

Na etapie przygotowania scenariusza jednym z istotnych elementów jest wybór narzędzi, które będą wykorzystywane do gromadzenia danych w fazie analizy. Na wybór określonych narzędzi będą miały wpływ: zakres gromadzonych danych, nakład pracy niezbędny do

przygotowania oraz konfiguracji wspomnianych narzędzi oraz wymagania w zakresie szczegółowości oraz sposobu opracowania wyników ze strony sponsora ćwiczenia oraz jego głównych uczestników.

6. Planowanie fazy symulacji

Planowanie fazy symulacji zależy głównie od dwóch czynników:

- złożoność eksperymentu symulacyjnego;
- jakość bazy danych scenariusza, na którym dokonujemy eksperymentów symulacyjnych.

Na procedurę prowadzenia symulacji składa się kilka podstawowych grup operacji. Pierwszą grupą są operacje: uruchamianie, zatrzymywanie gry oraz restart gry.

W drugiej grupie znajdują się operacje związane z wydawaniem rozkazów mające na celu sprawne kierowanie procesem symulacji. W trzeciej grupie są operacje związane ze wsparciem pracy ćwiczących (operatorów – w eksperymencie symulacyjnym będą nimi specjaliści i analitycy). W czwartej grupie znajdują się zadania związane z wykrywaniem błędów symulacji. W ostatniej grupie znajdują się zadania związane z archiwizowaniem wyników procesu symulacji oraz gromadzeniem danych dla etapu analizy.

Restarty z różnych punktów kontrolnych, zmiany prędkości gry w różnych fazach eksperymentu symulacyjnego dostosowane do potrzeb i wymagań poszczególnych ćwiczących (np. testowanie określonej grupy rozkazów, czy synchronizowanie działań), to typowy warsztat pracy kontrolera, dzięki któremu w znacznym stopniu można skrócić czas opracowania eksperymentu symulacyjnego. W eksperymencie symulacyjnym obowiązuje zasada im więcej testów tym lepiej. Niezależnie od zgłębiania wiedzy teoretycznej na temat wartości określonego parametru i jego wpływu na wyniki symulacji, przetestowanie symulacji przy różnych ustawieniach jego wartości może być cennym źródłem informacji.

Na początkowym etapie eksperymentu symulacyjnego kontroler uruchamia symulację i ustawia jej zatrzymanie w chwili, w której ćwiczący będzie mógł ocenić wyniki symulacji. W przypadku złożonych eksperymentów symulacyjnych kolejne uruchomienie mogą rozpocząć się z różnych punktów kontrolnych. Dodatkowo kontroler może znacznie przyśpieszyć symulacje w pewnych okresach czasowych, w których na danym etapie są nie istotne dla ćwiczących. Następnie zatrzymać ją i uruchomić ponownie na prędkości wymaganej przez ćwiczących.

W celu zatrzymywania i uruchamiania (na różnych prędkościach) kontroler wykorzystuje rozkazy `PAUSE` oraz `GAME SPEED`.

Ponieważ dla różnych uczestników ćwiczenia wymagania, co do punktów startu lub restartu (z punktu kontrolnego); punktów zatrzymania, przyśpieszania symulacji oraz jej zwalniania symulacji mogą być różne kontroler powinien opracować różne warianty starowania symulacją.

Symulacja jest zadaniem, które weryfikuje wszystkie efekty pracy kontrolera oraz ćwiczących. W złożonych eksperymentach symulacyjnych możliwe jest także przeprowadzenie oddzielnych symulacji dla poszczególnych ćwiczących, analityków lub stron ćwiczenia. W tym rozwiązaniu kontroler dostosowuje prędkość gry oraz wszystkie swoje działania do wymagań określonego ćwiczącego. Ćwiczący może sprawdzić wyniki działań swoich poszczególnych grup rozkazów i dokonać obserwacji na najmniejszym elementarnym poziomie. Po takim sprawdzeniu i upewnieniu się, że rozkazy zostały wprowadzone poprawnie ze wszystkimi wymaganymi opcjami oraz weryfikacji ich wyników w symulacji ćwiczący jest przygotowany na analizę wyników interakcji działań innych ćwiczących.

Po dokonaniu testów dla poszczególnych ćwiczących należy wykonać testy pełne ze wszystkimi działaniami kontrolera oraz ćwiczących.

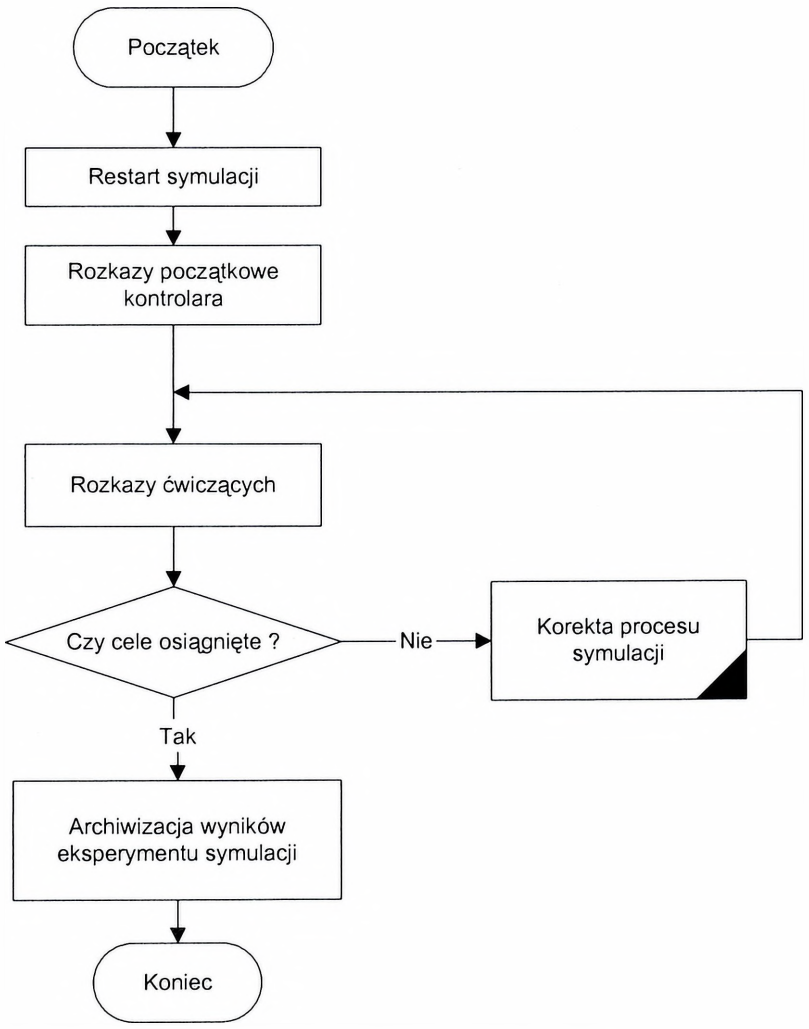
W celu zapewnienia powtarzalności wyników symulacji należy dbać o to, aby w kolejnych testach zapamiętywać do grup rozkazów eksperymentu symulacyjnego tylko te rozkazy i te w nich zmiany, które zostały zaakceptowane. Bardzo często zarówno kontroler jak i ćwiczący wydaje dodatkowe rozkazy lub modyfikuje istniejące w celu przetestowania pewnych ustawień, które po ocenie negatywnej nie powinny być zapisywanie do docelowych rozkazów eksperymentu symulacyjnego. Zapisanie przypadkowych zmian w docelowe grupy rozkazów będzie miało negatywny wpływ na przyszłe testy.

6.1. Planowanie prostych eksperymentów symulacyjnych

W przypadku prostych eksperymentów symulacyjnych procedura realizacji eksperymentu symulacyjnego jest uproszczona. Eksperyment symulacyjny realizowany jest w jednym etapie. W celu uproszczenia procedury testowania zaleca się dekompozycję wszystkich działań w etapie na mniejsze grupy działań. Rozbicie może być dokonywane zarówno ze względu na rodzaj działań, skład jednostek biorących udział w danym działaniu, jak i czas ich realizacji zadań. Procedura ma charakter iteracyjny i kończy się z chwilą, gdy cele eksperymentu symulacyjnego zostaną osiągnięte. Każde kolejne uruchomienie symulacji

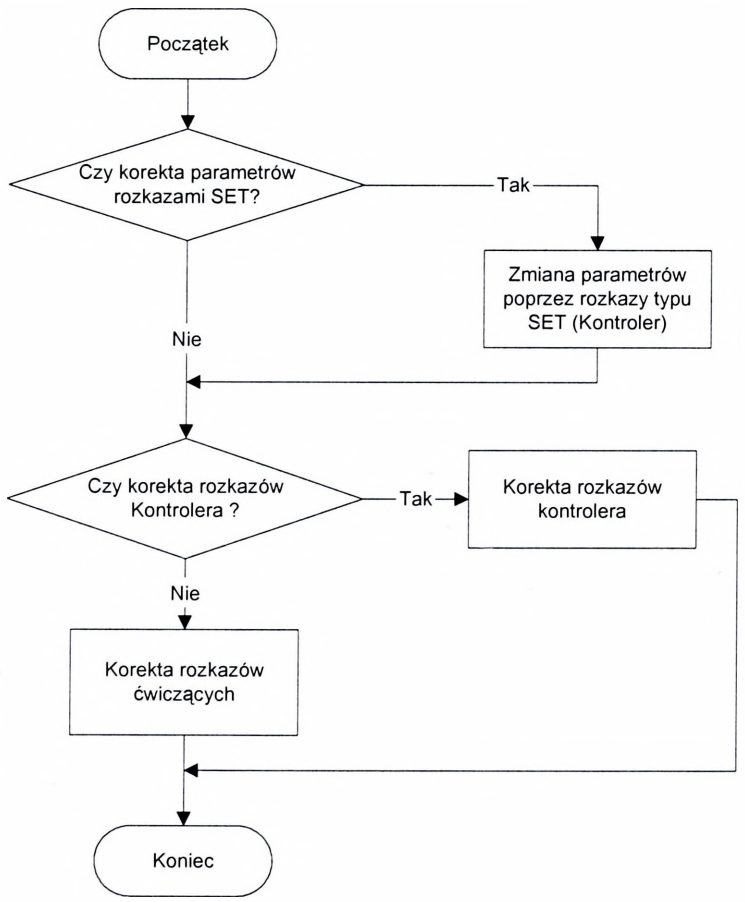
eksperymentu symulacyjnego rozpoczyna się od restartu symulacji (rozpoczęcia symulacji od momentu przyjętego, jako zerowy).

Po restarcie kontroler planuje pierwszy przyszły punkt zatrzymanie symulacji za pomocą rozkazu PAUSE. Pierwszy punkt zatrzymania symulacji najczęściej jest wykorzystywany w celu wysłania rozkazów początkowych, ustalających pewne parametry dla całego procesu symulacji. Do rozkazów początkowych kontrolera zaliczmy rozkazy przekazania dowodzenia, wysłania rozkazów rozpoznania początkowego itp. Po wysłaniu rozkazów początkowych kontrolera powinny zostać wysłane rozkazy ćwiczących. Przed wysłaniem rozkazów ćwiczących należy ustawić prędkość gry na różną od zera (rozkazy ćwiczących wymagają statusu „running" symulacji). Po wydaniu rozkazów ćwiczących należy ustawić kolejny punkt przerwania – istotny z punktu widzenia operacyjnego oraz zwiększyć prędkość symulacji. Po dokonaniu oceny wyników symulacji – przy pierwszych uruchomieniach najczęściej zgrubnych np. czy jednostki wykonują postawione im rozkazy, po jakim czasie osiągają poszczególne cele, położenia itp. Należy przystąpić do korekty procesu symulacji. Rysunek 16 przedstawia procedurę realizacji prostego eksperymentu symulacyjnego.



Rysunek 16. Procedura realizacji prostego eksperymentu symulacyjnego
/źródło: opracowanie własne/

Korekta procesu symulacji prostych eksperymentów symulacyjnych polega na modyfikacji rozkazów kontrolera lub ćwiczących. Jedna iteracja procesu symulacji składa się z następujących części: restart, wysłanie rozkazów, symulacja, ocena, korekta. Nie zaleca się dokonywania jednocześnie zmian rozkazów kontrolera i rozkazów ćwiczących z uwagi na późniejsze problemy w ocenie, która zmiana miała istotny wpływ na wyniki symulacji. Rozkazy kontrolera można podzielić na dwie grupy: rozkazy początkowe oraz rozkazy danego etapu gry (przy prostych eksperymentach występuje tylko jeden etap). Z analogicznych względów nie należy jednocześnie wprowadzać zmian w dwóch wyżej wymienionych grupach rozkazów kontrolera. Rysunek 17 przedstawia procedurę korekty prostego eksperymentu symulacyjnego.



Rysunek 17. Procedura korekty parametrów w prostym eksperymencie symulacyjnym
/źródło: opracowanie własne/

6.2. Planowanie złożonych eksperymentów symulacyjnych

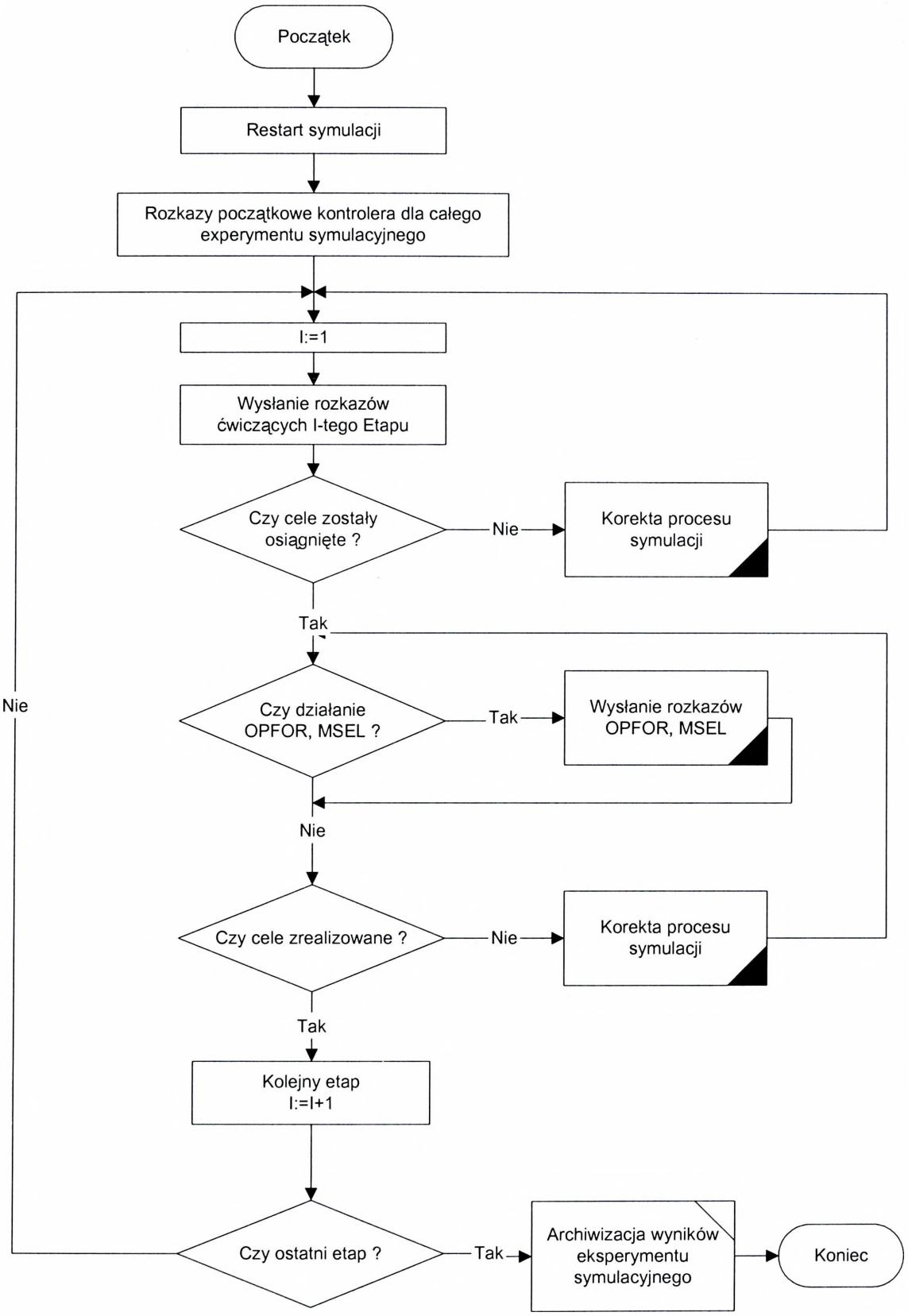
Złożone eksperymenty symulacyjne, to eksperymenty składające się z więcej niż jednego etapu symulacji (epizodu operacyjnego). Złożone eksperymenty symulacyjne wymagają dekompozycji całego eksperymentu symulacyjnego na kolejne etapy. Etap może składać się zarówno z działania ćwiczących, które spowoduje osiągnięcie pewnego stanu symulacji

(punktu milowego) oraz zsynchronizowanego z nim działania OPFOR lub działań kontrolera realizujących MSEL.

Cały eksperyment symulacyjny jest zakończony dopiero wtedy, gdy zostaną osiągnięte cele wszystkich etapów. Pojedynczy etap jest analogiczny do eksperymentu prostego. Z tą różnicą, że po osiągnięciu wyników etapu może wystąpić etap symulacji działania OPFOR lub MSEL, który oprócz fazy realizacji powinien zawierać w sobie fazę synchronizacji z wynikami etapu, dla którego został zaprojektowany.

Synchronizacja jest konieczna nawet wtedy, gdy w etapie nie wprowadzane są żadne nowe rozkazy OPFOR lub MSEL– efekty MSEL mogą być wprowadzone bezpośrednio do bazy danych (z czasami wprowadzenia do symulacji ustalonymi na etapie budowy scenariusza). W tym momencie powinna nastąpić korekta ich parametrów czasowych – synchronizacja z aktualnymi wynikami symulacji.

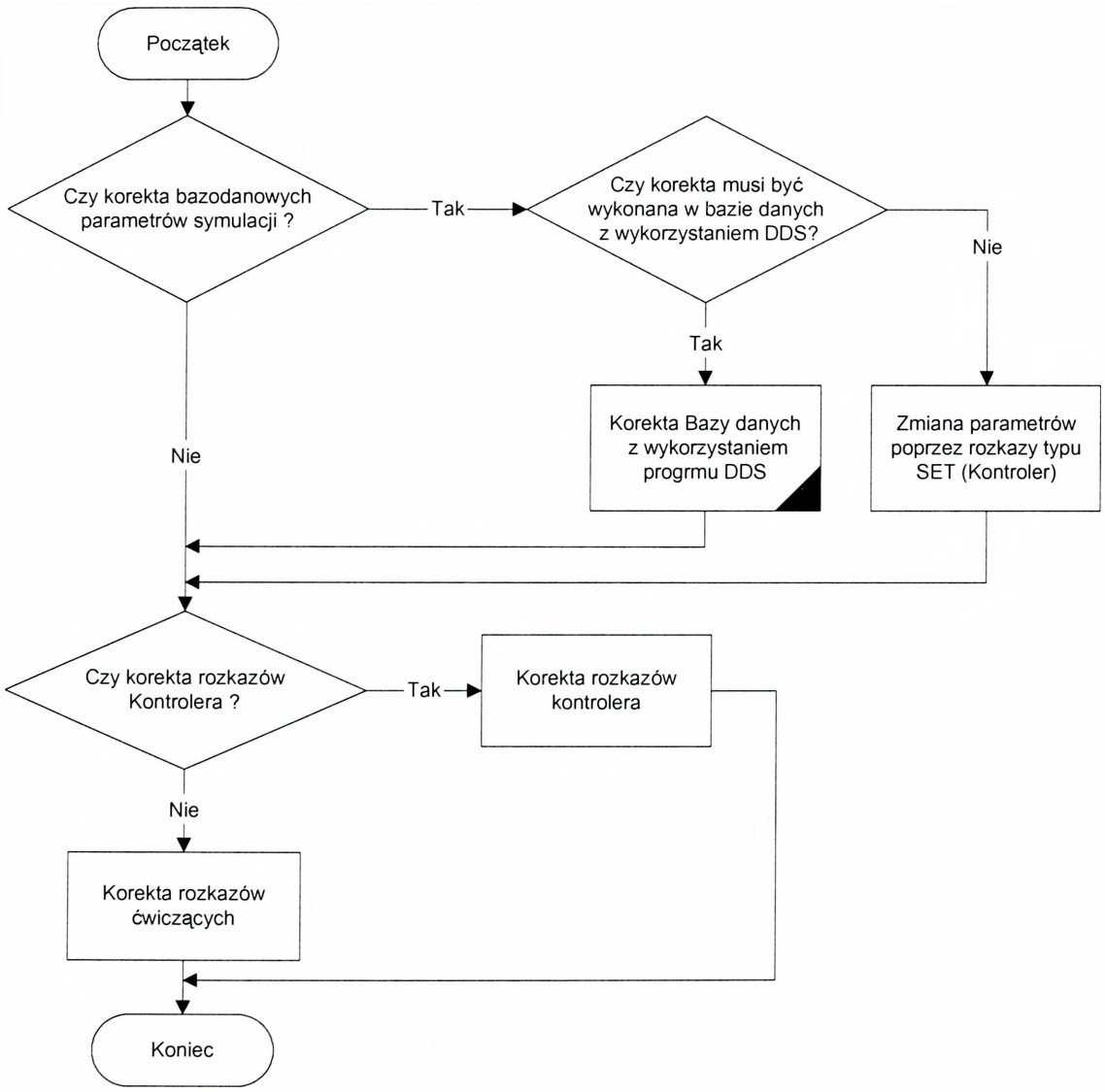
Rozkazy do synchronizacji działań OPFOR lub MSEL powinny zostać zgrupowanie w jedną grupę i zapamiętane. Mechanizm zgrupowania i zapamiętania umożliwi w trakcie rzeczywistego ćwiczenia CAX, dla którego opracowywany jest eksperyment symulacyjny szybkie, dynamiczne i dokładnie wprowadzenie zaplanowanych działań OPFOR lub MSEL. Rysunek 18 przedstawia procedurę realizacji złożonego eksperymentu symulacyjnego.



Rysunek 18. Procedura realizacji złożonego eksperymentu symulacyjnego
/źródło: opracowanie własne/

Korekta procesu symulacji może wystąpić zarówno po rozkazach ćwiczących etapu jak i po rozkazach ćwiczących następującego po nim działania (rozkazy) OPFOR lub MSEL. Etap uznaje się za zakończony wtedy, gdy są spełnione zarówno cele etapu, jaki i działań OPFOR lub MSEL. Stan po symulacji działania OPFOR lub MSEL staje się stanem wejściowym symulacji dla kolejnego etapu.

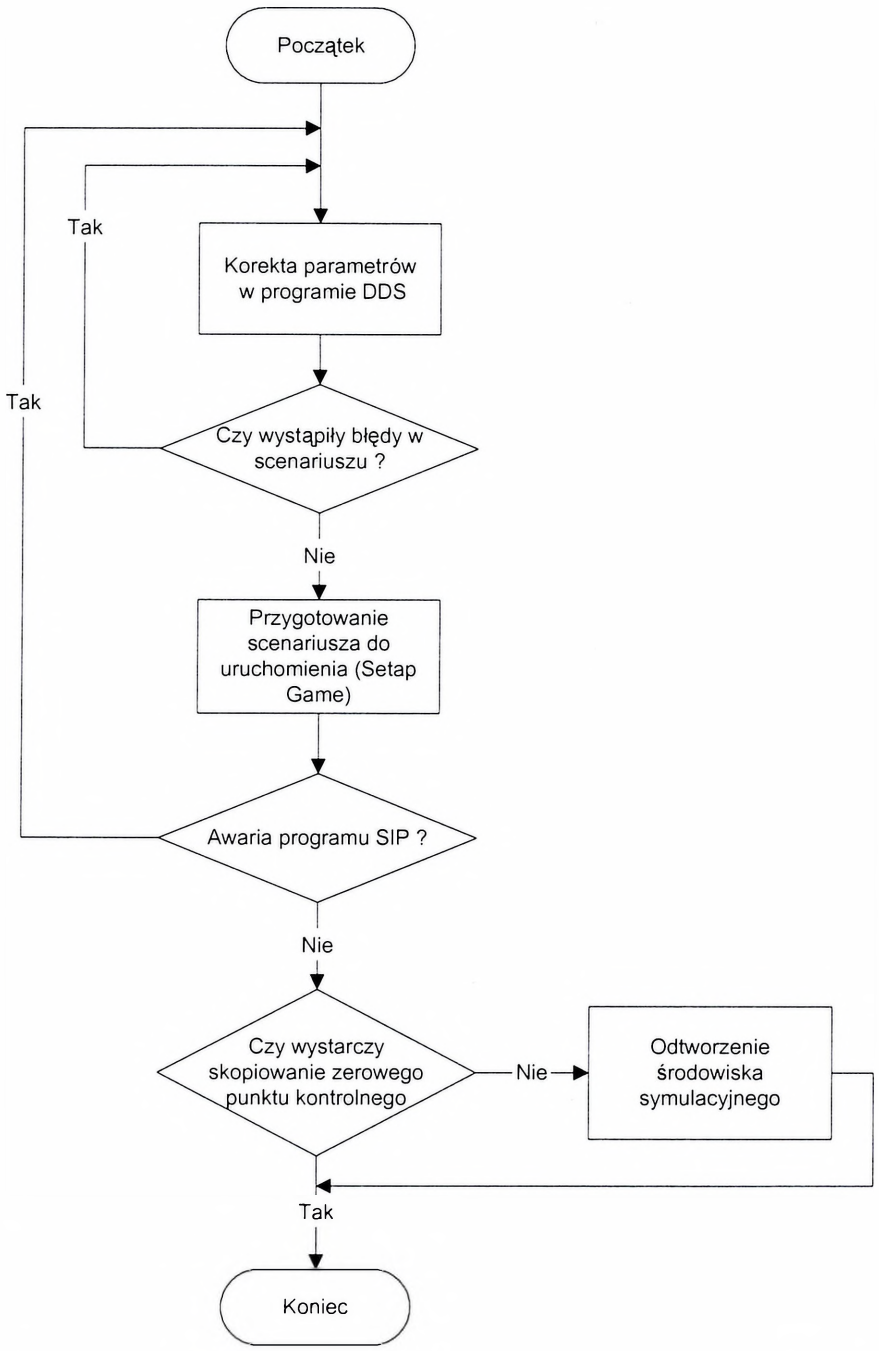
Korekta procesu symulacji podobnie, jak sam eksperyment symulacyjny może mieć charakter prosty i złożony. Kryterium podziału stanowi przede wszystkim potrzeba dokonania zmian w parametrach bazodanowych symulacji stanowiących środowisko otoczenie procesu symulacji. Dodatkowo sama modyfikacja parametrów bazodanowych może być prosta – wykonana za pomocą rozkazów kontrolera typu SET lub złożona – bardziej pracochłonna (z reguły przerywająca płynny proces opracowywania eksperymentu symulacyjnego) – wykonana za pomocą zewnętrznego programu DDS. Rysunek 19 przedstawia procedurę realizacji prostego eksperymentu symulacyjnego.



Rysunek 19. Procedura korekty parametrów w prostym eksperymencie symulacyjnym
/źródło: opracowanie własne/

Decyzja dokonania korekty parametrów bazodanowych w programie DDS może być powodowana wieloma względami. Pierwszym jest fakt, że nie wszystkie parametry bazodanowe mogą być zmienione za pomocą rozkazów typu SET. Dodatkowo zmiana parametrów bezpośrednio w procesie symulacji może spowodować awarię i zatrzymanie procesu symulacji. Po poprawieniu parametrów w programie DDS istnieje możliwość

weryfikacji scenariusza za pomocą programu SIP, który kompleksowo sprawdza całą bazę danych scenariusza. Rysunek 20 przedstawiono procedurę korekty parametrów z wykorzystaniem programu DDS (Database Development System).



Rysunek 20. Procedura korekty parametrów w programie DDS
/źródło: opracowanie własne/

Procedura dalszego postępowania po dokonaniu zmian scenariusza w programie DDS zależy od skali zmian. Zmiana niektórych parametrów może pociągać za sobą powtórzenie procedury przygotowania danych do symulacji (setup game), co z kolei może prowadzić do potrzeby ponownej konfiguracji całego eksperymentu symulacyjnego (np. po zmianie liczby lub nazwy stron). Ponowne przygotowanie danych do symulacji może spowodować, że część rozkazów może zawierać dane, które w nowym procesie symulacji już nie będą istniały (np. usunięcie systemu walki, kategorii zaopatrzenia, jednostki itp.). W wyżej wymienionych

sytuacjach wystąpi potrzeba odtworzenia procesu symulacji tzn. zapamiętanych grup rozkazów zarówno kontrolera jak i ćwiczących. Odtworzenie grup rozkazów jest istotnym elementem, ponieważ wypracowanie i wprowadzenie np. kilkuset rozkazów w złożonym eksperymencie symulacyjnym jest procesem bardzo czasochłonnym. Odtworzenie środowiska symulacyjnego z punktu widzenia zespołu operacyjnego daje możliwość dalszego płynnego opracowywania kolejnych etapów eksperymentu symulacyjnego.

6.3. Wykrywanie i usuwanie błędów symulacji

W czasie przygotowywania eksperymentu symulacyjnego jego nieodłączną częścią są awarie systemu symulacyjnego. Przyczyny awarii są różnorodne od błędów w scenariuszu bazy danych, błędów po zmianie parametrów rozkazami typu SET, błędów w działaniu ćwiczących, aż do błędów w samym procesie symulacji wynikłych z wad samego oprogramowania.

W czasie prowadzenia eksperymentu symulacyjnego awaria systemu nie jest tak ważna, jak w czasie ćwiczenia CAX, ponieważ nie wiąże się z nią ewentualna utrata wyników symulacji oraz niedozwolonymi i niezaplanowanymi przerwami czasowymi. W czasie eksperymentu symulacyjnego kontroler może w ciągu kilku minut ponownie restartować symulację, wydać zapamiętane rozkazy i zatrzymać symulację tuż przed wystąpieniem błędu. Po zatrzymaniu symulacji można dokonać następnie analizy, jak była przyczyna danego błędu. Z punktu sprawności przeprowadzenia eksperymentu symulacyjnego awarie są wadą. Jednak z punktu widzenia zespołu prowadzącego i zabezpieczającego awarie systemu mają także pozytywne strony. Doświadczenie zdobyte w czasie eksperymentów symulacyjnych daje namiastkę problemów, jakie mogą się pojawić w rzeczywistym ćwiczeniu CAX, gdy warunki laboratoryjne zmienią się na „bojowe" (rzeczywiste ćwiczenie i egzamin dla zespołu koordynatorów i kontrolerów technicznych). W trakcje prowadzenia eksperymentów symulacyjnych należy „ćwiczyć", jak najszybsze i najsprawniejsze usuwanie błędów zarówno od strony organizacyjnej, doboru specjalistów, jak technologicznej opanowania narzędzi, biegłości w posługiwaniu się nimi, jak i zrozumienia działania poszczególnych modelów wykorzystywanych w systemie symulacyjnym. Najgorszym podejściem jest traktowanie czasu usuwania błędów w eksperymencie symulacyjnym, jako czynnika mało istotnego ze względu na wspomniany laboratoryjny charakter przedsięwzięcia. W rzeczywistym ćwiczeniu CAX identyfikacja przyczyny błędu i jego usunięcie będzie się odbywała w bardzo ograniczonym czasie.

Wczesne wykrycie przyszłych błędów w trakcie symulacji jest bardzo istotnym elementem wsparcia technicznego ćwiczeń wspomaganych komputerowo. Odpowiednio wcześnie wykryty błąd może zostać poddany analizie, która pozwoli zlokalizować przyczynę błędu i następnie w uzgodnieniu między kontrolerem gry i kierownictwem ćwiczenia podjąć decyzję, w jaki sposób uniknąć wspomnianego błędu, aby miało to jak najmniejsze skutki uboczne dla ćwiczących.

Wczesne wykrywanie błędów umożliwia mechanizm przyśpieszania symulacji w połączeniu z mechanizmem do składowania wyników symulacji w tzw. punktach kontrolnych. W celu przeprowadzenia wczesnego wykrywania błędów należy tylko przenieść, skopiować bieżące dane z symulacji w inne miejsce i ustawić prędkość gry na kilkadziesiąt razy większą niż zastosowana w ćwiczeniu. Podstawowym problemem w usunięciu przyszłego błędu jest bardzo ograniczony czas na podjęcie decyzji: jak go usunąć łącznie z przeprowadzeniem jego usunięcia (w środowisku testowym). Na szanse realizacji powyższego zadania w znacznym stopniu ma wpływ posiadanie drugiego zdublowanego zespołu złożonego (minimum) z kontrolera oraz analityka modelu. Ten zespół powinien niezależnie i niejako w tle realizować procedury wykrywania przyszłych błędów oraz wypracowywać metody ich usuwania.

7. Planowanie gromadzenia danych do analizy i oceny

Proces gromadzenia danych na potrzeby analizy i oceny w trakcie eksperymentu symulacyjnego może być w ramach całości eksperymentu traktowany dwojako:

1. jako element wspomagający eksperyment – dane zgromadzone dzięki niemu posłużą do oceny wyników eksperymentu i dzięki nim zostanie dokonana ocena uzyskanych rezultatów, a co za tym idzie podjęta zostanie decyzja odnośnie kolejnych działań i kontynuowania iteracyjnego wypracowywania danych potrzebnych do przeprowadzenia eksperymentu, ustalania i korygowania warunków rozpoczęcia symulacji oraz ocena poprawności zachowania się modelu w interakcji z graczami.

2. jako element główny eksperymentu – w tym przypadku celem jest ocena samego procesu zbierania informacji o zachowaniu się środowiska symulacyjnego, poprawności działania wybranych narzędzi, które mogą zostać wykorzystane oraz zasady ich doboru w zależności od celu eksperymentu, którego elementem wspomagającym będzie gromadzenie danych.

W ogólnym przypadku, zakładając, że skład zespołu biorącego udział w planowaniu i przygotowywaniu eksperymentu symulacyjnego nie zmienia się lub też jego działania są dobrze udokumentowane, dzięki czemu doświadczenia są kumulowane, przypadek drugi może występować stosunkowo rzadko. Planujący eksperyment lub ciąg eksperymentów mogą stosować takie podejście w przypadkach, kiedy zmienia się coś w zasadzie działania wykorzystywanych narzędzi lub wręcz ich zestaw, nastąpiły istotne zmiany w budowie wykorzystywanego systemu symulacyjnego (np. po podniesieniu wersji) lub też zmianie uległa platforma sprzętowa i/lub systemowa, która może warunkować duże zmiany jakościowe (np. wydajność).

Przypadkiem, kiedy drugie podejście może być stosowane dość często, będzie proces planowania fazy analizy i oceny przy często zmieniających się (pomiędzy różnymi ćwiczeniami) podejściach do tej fazy oraz sposobu oceny. Uwidoczni się to szczególnie w przypadku planowania wykorzystania narzędzia o rozbudowanych możliwościach zarówno gromadzenia danych jak i konfiguracji ich zakresu oraz częstotliwości.

7.1. System JTLS

Planowanie procesu gromadzenia danych na potrzeby analizy i oceny w przypadku symulacji prowadzonej w oparciu o system JTLS jest silnie determinowane zestawem

narzędzi wspomagających ten proces dostarczanych wraz z systemem. W zestawie tym znajdują się następujące narzędzia:

- GAWS (ang. *GIAC Analyst Workstation, GIAC – Graphical Input Aggregate Control*),
- PPS (ang. *Post Processor System*),
- Programy otwartego dostępu (ang. *Open Access Programs*).

Narzędzia te różnią się od siebie bardzo silnie zarówno pod względem zasad działania, postaci uzyskiwanych wyników, jak i możliwości ich wykorzystania oraz wpływu sposobu przeprowadzania eksperymentu na możliwość ich wykorzystania. Poniżej zostanie przedstawiona ich ogólna charakterystyka, która stanowić będzie podstawę do planowania wykorzystania ich w poszczególnych planowanych eksperymentach symulacyjnych.

7.1.1. GAWS

Proces GENIS odpowiedzialny w systemie za dystrybucję informacji przekazywanych mu przez CEP (ang. *Combat Events Program*), może zostać skonfigurowany tak, aby cyklicznie zapisywać informacje o bieżącej sytuacji w grze w odstępach z ustalonym interwałem. Informacje te są zapisywane w plikach na twardym dysku.
GAWS (ang. *GIAC Analyst Workstation, GIAC – Graphical Input Aggregate Control*) może wykorzystać zapisane dane o sytuacji otrzymane od GENIS do wyświetlenia obrazu sytuacji w wybranym czasie lub film złożony z sekwencji obrazów sytuacji wyświetlanych na podstawie zestawu danych zapisanych dla wskazanego okresu.

7.1.1.1. Możliwości gromadzenia i składowania danych

Żeby móc wykorzystać możliwości bazy danych GAWS należy przeprowadzić konfigurację procesu GENIS. Następujące parametry sterują gromadzeniem i zapisywaniem danych w bazie::

- `ddb`
- `ddb_directory <database_home>`
- `ddb_dbname <database_name>`
- `ddb_flush_interval <integer_value_in_minutes>`

- `ddb_model_data <model> <data_type>`
- `ddb_timemaster <model>`

gdzie:

ddb – informuje GENIS, że ma się połączyć z bazą danych;

ddb_directory <database_home> - pełna ścieżka do katalogu, w którym jest zapisywana baza danych;

ddb_dbname <database_name> – nazwa bazy danych, która będzie zapisywana;

ddb_flush_interval <integer_value_in_minutes> - czas (w minutach gry), jaki GENIS odczekuje zanim wysyła dane do bazy danych;

ddb_model_data <model><data_type> - specyfikuje typ danych, który powinien być zapisywany do bazy danych. Oddzielna pozycja `ddb_model_data` powinna występować w pliku konfiguracyjnym GENIS dla każdego wyspecyfikowanego typu danych.

model – jest nazwą modelu dostarczającego dane (np.: CBS, JTLS, AWSIM);

data_type – jest określeniem typu danych otrzymywanych z modelu. Poprawnymi typami są:

- `aircraft`
- `base`
- `radar`
- `sam`
- `ship`
- `track`
- `squadron`
- `helo`
- `ground`
- `hex`
- `target`
- `report`

- convoy
- route
- unlink
- uline
- uview
- intel
- farp
- depot
- formation
- air_mission
- air_unit
- air_squadron
- air_track

ddb_timemaster <model> - określenie modelu, z którego GIAC otrzymuje informacje o upływającym czasie (np.: CBS, JTLS, AWSIM).

7.1.1.2. Możliwości odtwarzania zapisanych danych

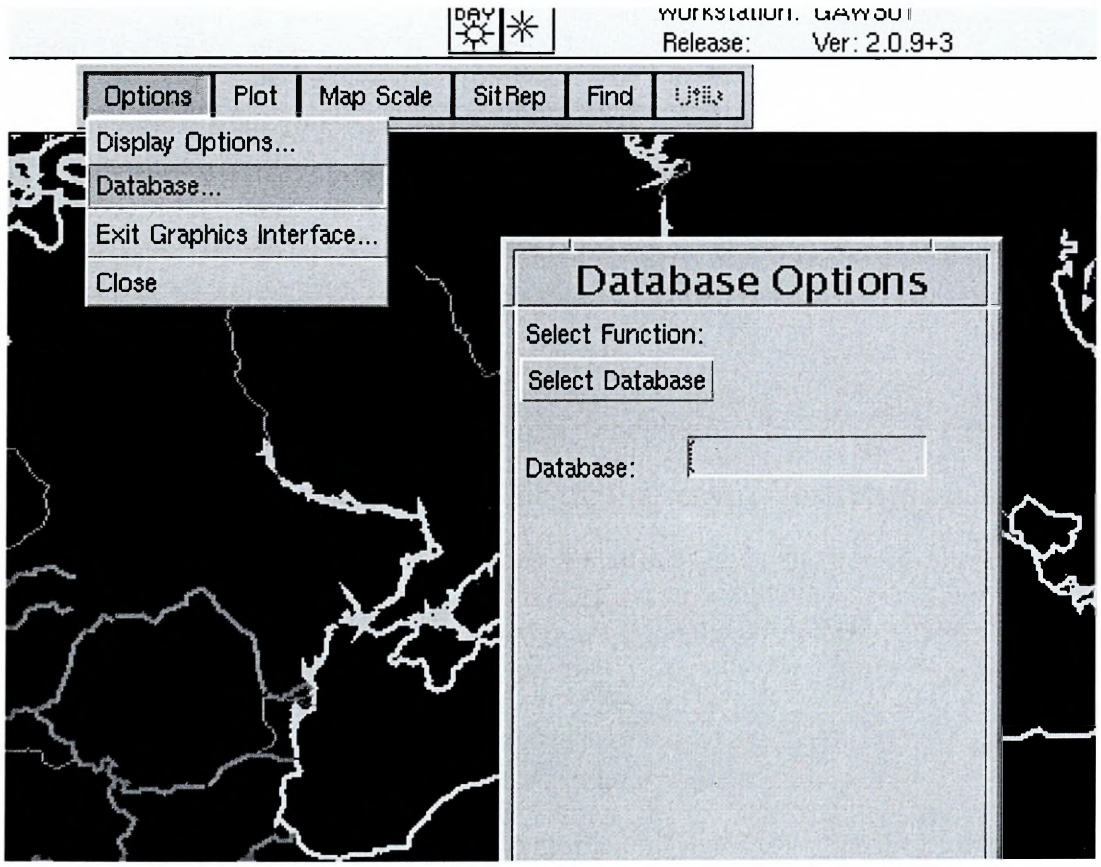
Do wyświetlania zebranych w trakcie prowadzenia CAX obrazów przekrojów czasowych (ang. *time slices*) GAWS dostarcza trybu graficznego migawki (ang. *snapshot mode*) lub filmu (ang. *movie mode*). GAWS pozwala także, na podstawie zarchiwizowanych przekrojów czasowych, na sporządzanie wydruków na papierze, jak i wytwarzanie folii transparentnych. Posiada również możliwość sporządzania graficznych i tekstowych adnotacji, która może zostać wykorzystana do opatrzenia nimi przygotowywanych materiałów do dalszej analizy.
Bazę danych GAWS stanowi repozytorium zebranych przekrojów czasowych.

Do bazy danych GAWS może być podłączone wiele stacji GAWS, umożliwiając tym samym prowadzenie równoległej analizy na wielu stacjach roboczych w wielu lokalizacjach.

GAWS umożliwia wyświetlanie zarchiwizowanych danych (zebranych i zeskładowanych w bazie danych utworzonej przez GENIS). Zarchiwizowane dane są

pobierane i wyświetlane przez GAWS pozwalając na przeprowadzenie zarówno analizy, oceny czy omówienia ćwiczeń, jak i oceny symulacji. GAWS pozwala na graficzne zobrazowanie zarchiwizowanych danych dając możliwość wyświetlenia migawki (ang. *snapshot*) działań bojowych w oparciu o jeden zadany przekrój czasowy (ang. *time slices*), lub filmu składającego się z przekrojów czasowych w zadanym zakresie (w czasie od – do).

GAWS ułatwia trenerowi lub analitykowi ustalenie kluczowych czynników leżących u podstawy sukcesu lub klęski w środowisku treningowym. Jest używana do wspomagania tworzenia scenariusza, planowania i wypróbowywania misji oraz formułowania wymagań dla nowych lub ulepszonych systemów walki. GAWS dostarcza możliwości oceny zdolności operacyjnych sił połączonych i koalicyjnych (ang. *Joint, Combined Arms, and Coalition Forces operational performance assessment*).



Rysunek 21. GIAC - menu "Options/Database" – wybór bazy danych

Baza danych pozwala na archiwizację scenariusza działań bojowych oraz jego późniejsze wydobycie i przegląd.

GAWS udostępnia trzy tryby wyświetlania zgromadzonych przekrojów czasowych:

- **Bieżący (ang. *Current*)**

Wyświetlanie ostatniego przekroju czasowego dostępnego w bazie danych. Obraz zostanie automatycznie uaktualnione, gdy do bazy zostanie dodany nowy przekrój czasowy.

- **Migawka (ang. *Snapshot*)**

  Wyświetlanie zobrazowania jednego wybranego przekroju czasowego.

- **Film (ang. *Movie*)**

  Wyświetlanie filmu złożonego z sekwencji przekrojów czasowych w zadanym zakresie (czas od - do). Szybkość sekwencji może być modyfikowana.

7.1.2. Post Processor System

PPS (ang. *Post Processor System*) wykorzystuje pojęcie punktu kontrolnego (ang. *checkpoint*), możliwości Oracle oraz właściwości implementacji Oracle w JTLS do składowania szerokiego wachlarza danych wyjściowych PPS. Do tego celu wykorzystywana jest obszar tabel Oracle zaalokowany do przechowywania scenariusza JTLS. Dodatkowa liczba nowych tabel Oracle jest definiowana dla przechowywania danych z gry JTLS dla potrzeb PPS. Są one dodawane do już istniejących tabel potrzebnych do przygotowywania scenariuszy dla JTLS przy pomocy DDS (ang. *Database Development System*). W czasie gry JTLS, kontroler ustawia znacznik, który określa czy dane dla PPS będą zbierane i gromadzone w jej trakcie (na wartość "YES" lub "NO"). Gdy znacznik ma wartość "YES", CEP (ang. *Combat Events Program*) zbiera i składuje wybrane dane dla PPS poprzez zapisywanie ich do plików w tekstowych formacie ASCII. Każdy z tych plików jest przygotowany do załadowania do jednej z tablic Oracle utworzonej dla PPS.

Na każdą tabelę w Oracle przypada jeden odpowiadający jej plik ASCII. Każdy plik jest otwierany, gdy gra startuje i zamykany, gdy tworzony jest punkt kontrolny (ang. *checkpoint*). Po zakończeniu tworzenia punktu kontrolnego otwierany jest nowy plik. Pliki są zapisywane w ten sam sposób, w jaki zapisywane są inne pliki punktu kontrolnego JTLS.

Mimo iż można zastosować wiele metod do wprowadzania danych do bazy, niżej opisana strategia jest wykorzystywana ze względu, że wczytywanie danych przy pomocy SQL*Loader (oprogramowanie z pakietu Oracle) jest najszybszą metodą ładowania dużych ilości danych do tego typu bazy (baza Oracle).

Zbiór plików wchodzących w skład punktu kontrolnego jest ładowany do tabel Oracle tylko wtedy, gdy użytkownik tak zadecyduje. Pozostawienie użytkownikowi całkowitej kontroli nad tym, jakie dane (z plików PPS punktu kontrolnego) są ładowane do tabel Oracle, posiada wiele zalet. Proces ładownia danych obwarowany jest zabezpieczeniami w celu zapobieżeniu załadowania uszkodzonych lub niespójnych danych, zmniejszenia prawdopodobieństwa przypadkowego nadpisania danych czy nieumyślnego usunięcia danych. We wszystkich tych sytuacjach generowane są ostrzeżenia.

Jedno z pól w każdym rekordzie w każdej tabeli przechowuje numer punktu kontrolnego, podczas którego dana ta została zebrana i w skład którego wchodzi. Pozwala to na stworzenie wielu możliwości odczytywania danych dla danego scenariusza, który został załadowany do JTLS.

PPS jest aplikacją, która dostarcza użytkownikom JTLS wszystkich niezbędnych funkcji do przygotowania baz danych PPS dla scenariusza JTLS. Zadaniem PPS jest wykonanie przygotowywania bazy danych oraz przeprowadzenie personelu technicznego przez proces tworzenia nowej bazy danych PPS scenariusza JTLS lub modyfikacji już istniejącej.

7.1.2.1. Przygotowanie do użycia PPS

Przed użyciem jakiejkolwiek funkcji PPS, należy upewnić się, że wszystkie pliki ASCII z danymi PPS zostały wygenerowane podczas wykonywania się gry. Generowanie potrzebnych plików ASCII z danymi PPS następuje tylko wówczas, kiedy PPS został uaktywniony podczas gry.

Kiedy JTLS jest zatrzymywany na koniec dnia ćwiczeń lub na końcu całych ćwiczeń, zwykle tworzony jest punkt kontrolny. Punkty kontrolne są również tworzone cyklicznie, co ustalony interwał, podczas gry (bez jej zatrzymywania).

Gdy PPS jest uaktywniony, częścią procedury tworzenia punktu kontrolnego jest generowanie plików ASCII dla PPS. Uaktywnienie powoduje utworzenie zestawu plików ***.pps_*** dla każdego punktu kontrolnego oraz dla sekwencji punktów kontrolnych czyli przebiegu (ang. *RUN*). W rezultacie, punkt kontrolny pozwala PPS na wypełnienie tabel bazy danych PPS danymi zebranymi i zeskładowanymi od czasu wykonania ostatniego punktu kontrolnego.

W przypadku awarii konieczne jest wykonanie tzw. awaryjnego punktu kontrolnego.

Dla każdego scenariusza JTLS nazwa i hasło użytkownika PPS w Oracle musi zostać utworzone przez administratora bazy danych Oracle. Utworzonemu użytkownikowi należy nadać również niezbędne uprawnienia. Nazwa użytkownika Oracle musi być adekwatna do scenariusza JTLS, żeby użytkownik miał dostęp do bazy danych z wykorzystaniem funkcji i możliwości PPS. Należy odwołać się do odpowiedniej dokumentacji Oracle w celu uzyskania instrukcji nt. tworzenia nowego użytkownika Oracle i nadawania mu niezbędnych uprawnień.

7.1.2.2. Uaktywnianie PPS

Jak wspomniano uprzednio, kontroler podczas gry JTLS ustawia znacznik na wartość "YES" w celu nakazania PPS rozpoczęcia gromadzenia danych. Znacznik jest ustawiany, kiedy kontroler przesyła rozkaz SET DEBUG REPORT FLAGS (Game Control->Set Parameters->Set Modeling Parameters->Set Debug and Report Flags -> Post Processor Data Flag = YES). Ustawienie znacznika gromadzenia danych PPS powoduje, że CEP zapisuje w różnych plikach PPS informacje niezbędne dla dostarczenia danych dla PPS.

7.1.2.3. Dostęp do danych PPS

JTLS jest zaprojektowany tak, aby dostarczać wsparcia dla osiągnięcia celów ćwiczenia. Jeżeli ktoś całkowicie zrozumie to proste stwierdzenie, może również w konsekwencji zaakceptować, że dane generowane przez PPS nie muszą być traktowane jako nietykalne czy wręcz "święte". Innymi słowy, dane wyjściowe z PPS mogą bardzo wiernie reprezentować to, co dzieje się w modelu i nadal nie spełniać wymogów ćwiczenia.

Przykładowo gracze (ang. *Players*) mogą nieumyślnie podejmować decyzje i wykonywać akcje niezgodne ze scenariuszem ćwiczenia. Mogą być wówczas potrzebne czynności kontrolera (ang. *Controller*) do stworzenia warunków dla zaspokojenia specyficznych wymagań kierownika ćwiczeń (ang. *Exercise Director*), lub też mogą się pojawić problemy z modelem lub platformą systemową podczas ćwiczenia.

Z takich i innych względów PPS dostarcza możliwości modyfikacji wybranych danych w poszczególnych rekordach w ściśle określonych tabelach. Ściśle określone pola, których zmiana jest dozwolona są wyróżnione dla każdej tabeli. Formularze PPS umożliwiają zmianę rekordów w sposób podobny do tego, w jaki DDS obecnie pozwala na zmiany danych inicjalizujących scenariusz.

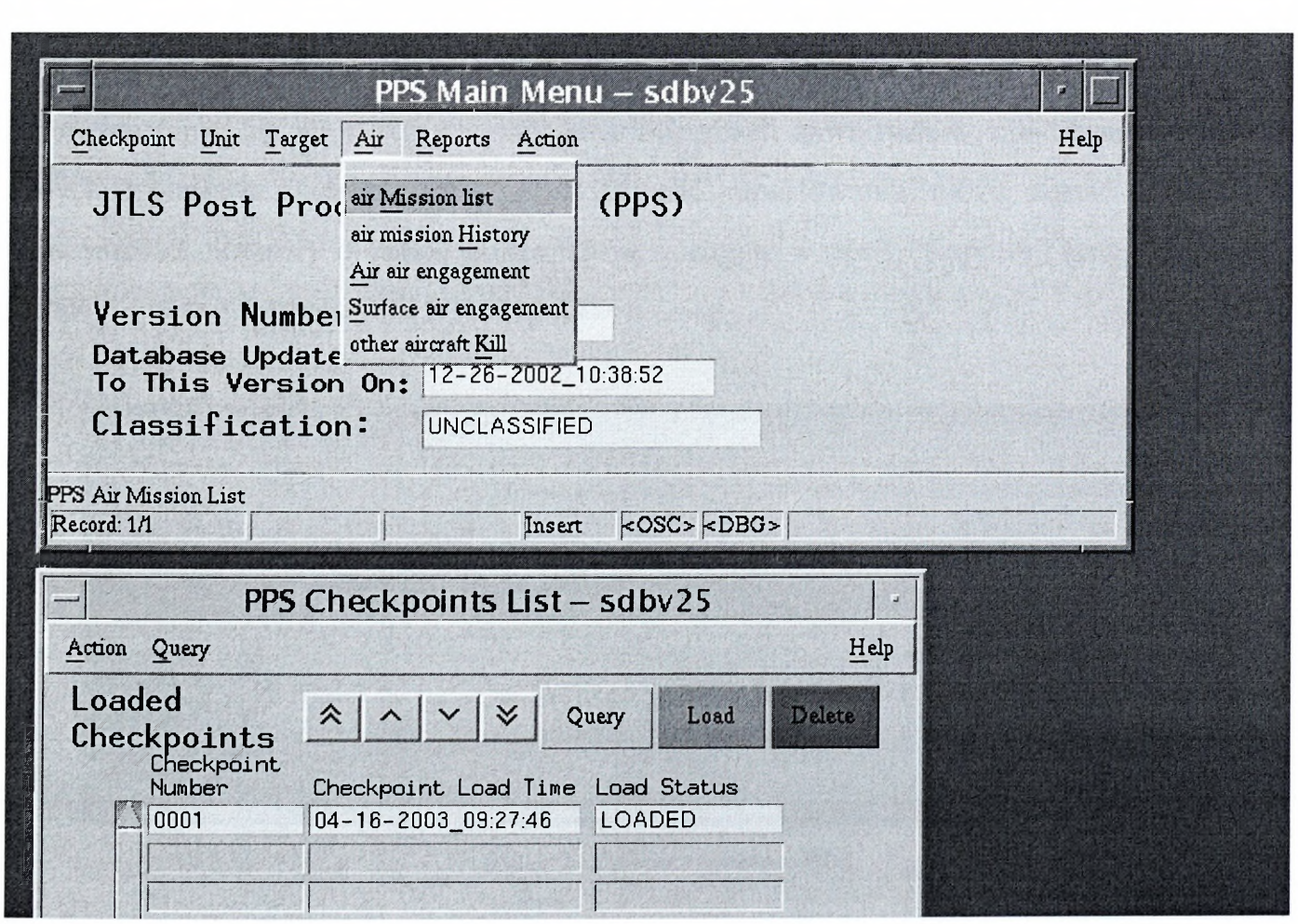
Każda zmiana dokonana przez użytkownika jest zapisywana w oddzielnym rekordzie w odpowiedniej tabeli scenariusza (np.: `PP_AIR_DATA_CHANGE_TABLE`). Pozwala to na śledzenie, podsumowywanie i archiwizację ręcznych zmian i korekt danych w bazie PPS.

7.1.2.4. Funkcjonalność PPS

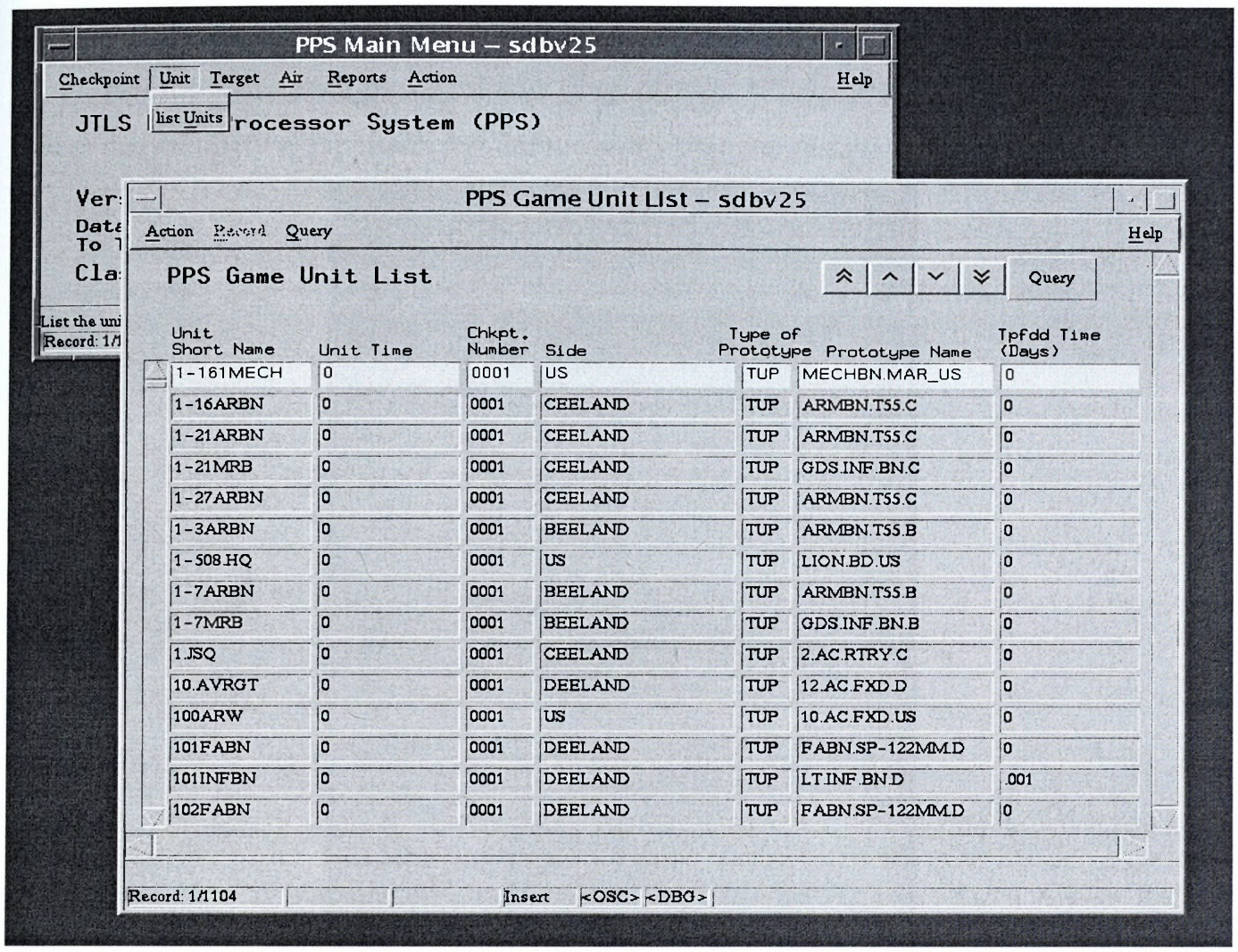
PPS jest aplikacją umożliwiającą użytkownikowi dostęp do informacji zapisanych w bazie danych PPS scenariusza JTLS. Formularze PPS zostały wytworzone przy użyciu Forms/Reports Developer 6i, który jest narzędziem do projektowania, tworzenia (implementacji) i wdrażania opartych na formularzach aplikacji typu klient-serwer pracujących w oparciu o serwer bazy danych Oracle.

Dostęp do formularzy PPS jest zorganizowany w strukturze rozwijanego hierarchicznego menu. Użytkownik obsługuje PPS poprzez używanie zestawu menu i formularzy. Formularze PPS wykorzystuje do pracy z danymi zapisanymi w bazie danych Oracle w następujący sposób:

- do odczytu rekordów z bazy;
- do wprowadzania zmian w rekordach w bazie.

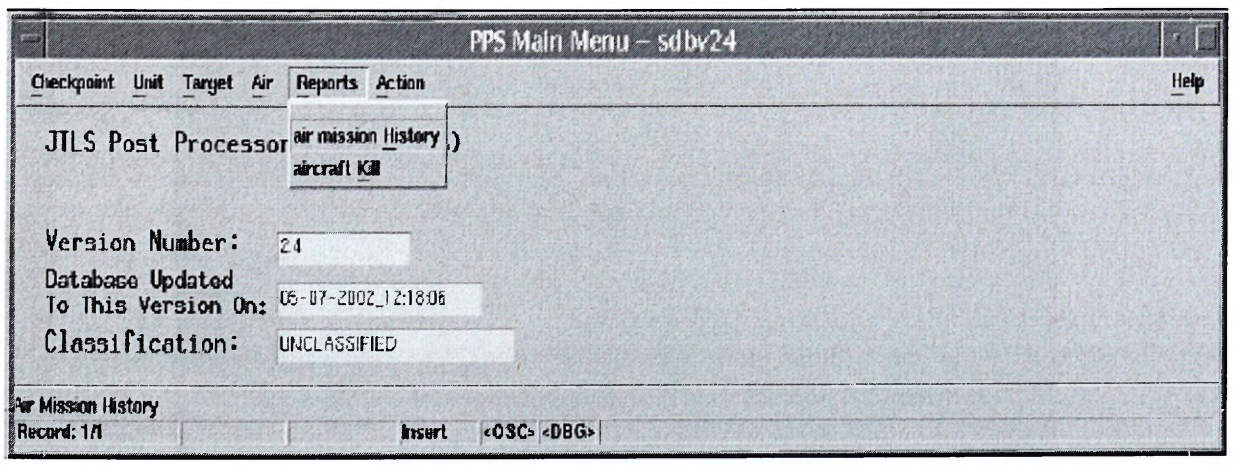


Rysunek 22. Przykładowa formatka PPS z listą załadowanych punktów kontrolnych



Rysunek 23. Przykładowa formatka PPS z listą jednostek

PPS pozwala użytkownikom PPS tworzyć szczegółowe raporty związane z lotami bojowymi (ang. *air missions*) na podstawie informacji zawartych w załadowanych do PPS danych z punktów kontrolnych. Ta możliwość raportowania może zostać rozszerzona w przyszłości. Raporty o lotach bojowych są dostępne w opcji "Raporty" (ang. *"Reports"*) formatki menu głównego PPS (rysunek 24).

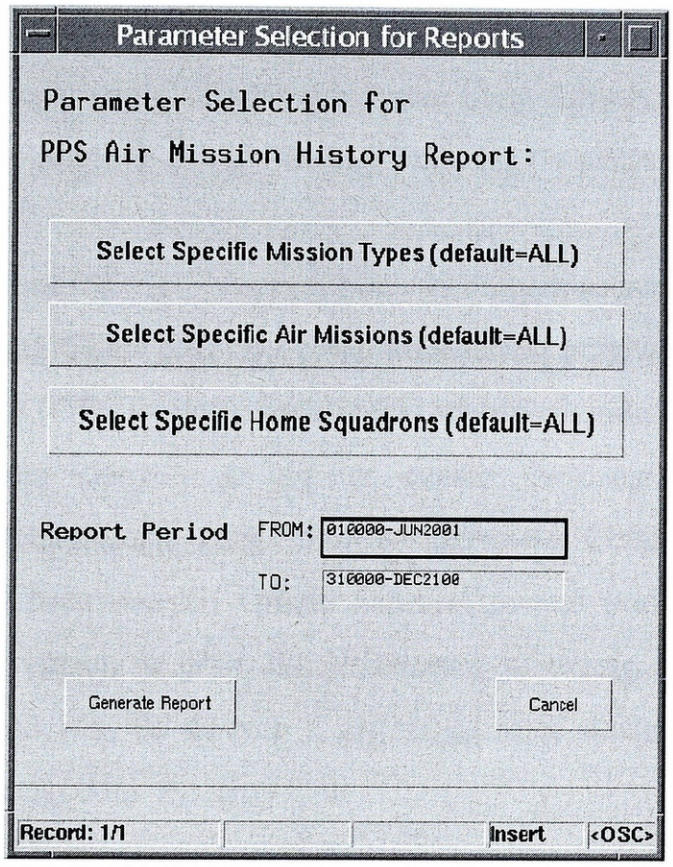


Rysunek 24. Menu raportów PPS

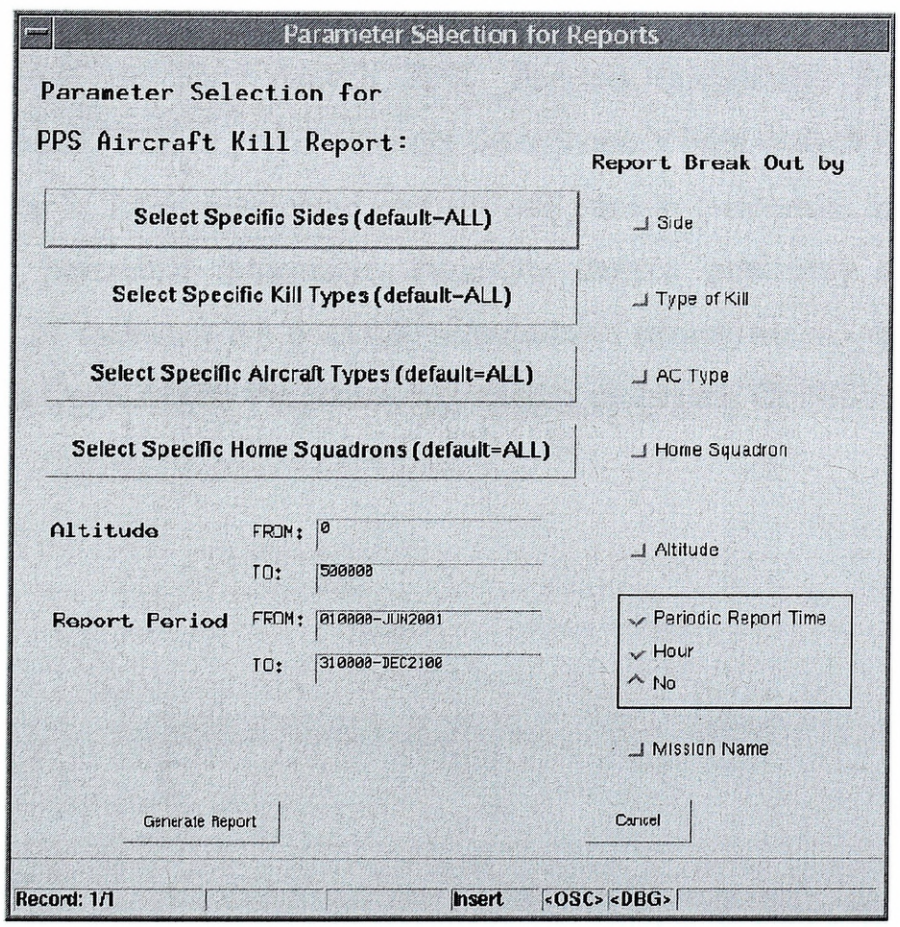
W menu raportów PPS dostępne są następujące typ raportów:

- raport o historii lotów bojowych (*ang. air mission history report*);
- raport o zestrzeleniach samolotów (*ang. aircraft kill report*).

Zakres danych ujęty w raportach jest definiowalny poprzez zestaw parametrów.



Rysunek 25. Parametry raportu o historii lotów bojowych



Rysunek 26. Parametry raportu o zestrzeleniach samolotów

7.1.3. Open Access Programs

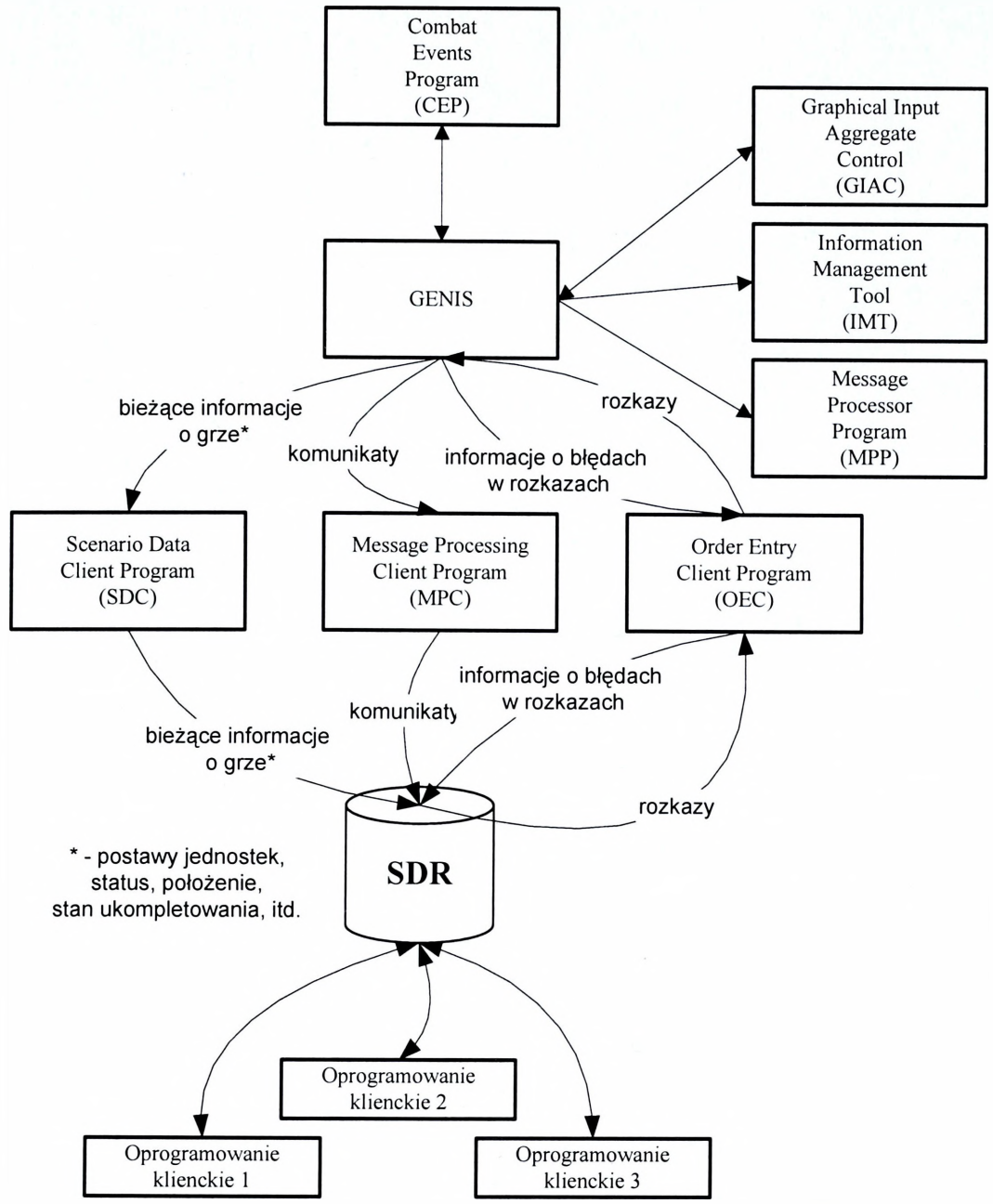
Programy otwartego dostępu (ang. *Open Access Programs*) pozwalają na dostęp do danych o grze w czasie jej trwania za pośrednictwem bazy danych, w której są składowane. W skład Open Access Programs w chwili obecnej wchodzą (w przyszłości ich liczba może się zwiększyć):

- Scenario Data Client (SDC) – program kliencki danych scenariusza;
- Message Processing Client (MPC) – program kliencki przetwarzania komunikatów;
- Order Entry Client (OEC) – program kliencki wprowadzania rozkazów.

Do przechowywania danych z gry w czasie ćwiczenia wykorzystywane jest repozytorium danych scenariusza (ang. *SDR - Scenario Data Repository*). Jest ono zaimplementowane jako baza danych Oracle zaprojektowana pod kątem przechowywania aktualnej informacji o sytuacji w grze aktualizowanej w czasie rzeczywistym. Ponieważ wspiera ono otwarte standardy, do danych z gry mogą mieć dostęp aplikacje firm trzecich takie jak narzędzia raportujące lub systemy C4I.

Open Access Programs również (podobnie jak np.: GIAC) są programami klienckimi GENIS komunikującymi się z GENIS w celu odbierania bieżących danych o grze, które następnie przekazują do bazy danych SDR. Podczas inicjalizacji program ustanawia połączenie z modelem JTLS za pośrednictwem GENIS oraz z bazą danych SDR.

Każda aplikacja, która potrzebuje otrzymywać dane o ćwiczeniu, może połączyć się z SDR i uzyskać potrzebne informacje. Przepływ danych pomiędzy tymi programami obrazuje rysunek 27 (schemat nie pokazuje szczegółowo przepływu informacji nieistotnych dla opisu działania Open Access Programs, np.: między procesami GENIS i CEP).



Rysunek 27. Przepływ danych między SDR i JTLS

/źródło: opracowanie własne na podstawie JTLS Technical Coordinator's Guide/

7.1.3.1. Scenario Data Client (SDC)

Scenario Data Client (SDC) jest odpowiedzialny za przekazywanie i aktualizację Scenario Data Repository (SDR) aktualnymi informacjami o grze dostarczanymi przez GENIS. Wewnątrz SDR istnieją oddzielne tabele Oracle dla każdego typu danych otrzymywanych od GENIS (Dodatki–Dane z przebiegu eksperymentu gromadzone w tabelach SDC, Tabela 5 oraz Tabela 6 do Tabela 45).

SDC może zażądać całej informacji o scenariuszu oraz odbierać wszystkie informacje generowane przez GENIS i przekazywać te dane do SDR, lub może zostać skonfigurowany tak, żeby żądać informacji o wybranych stronach (lub nawet jednej stronie) gry.

Od chwili, kiedy początkowe rekordy inicjalizujące zostaną zapisane w tabelach SDR, SDC jest odpowiedzialny za propagowanie informacji aktualizujących oraz usunięć

otrzymywanych od GENIS tak, aby informacje zawarte w tabelach odpowiadały bieżącej sytuacji w grze.

7.1.3.1.1. *Tabele z danymi SDC*

Przeważająca większość tych tabel dziedziczy swój format bezpośrednio ze struktur danych GENIS. Na przykład struktura `G_BASE`, która jest używana przez GENIS do przekazywania informacji o bazach lotniczych, ma odpowiadającą jej tabelę `base_data` w SDC. Istnieje kilka drobnych różnic pomiędzy strukturą `G_BASE` i tabelą `base_data`:

1. Tabela `base_data` zawiera kolumnę `side_view`, ponieważ tabel może zawierać informacje o obiektach wielu stron.
2. Atrybut `SAID` ze struktury `G_BASE` odwzorowany jest w tabeli `base_data` jako dwie kolumny: jedna dla indeksu obiektu, druga dla typu obiektu.
3. Atrybut `G_TIME` ze struktury `G_BASE` odwzorowany jest w tabeli `base_data` jako dwie kolumny: jedna dla liczby sekund od wystartowania gry, druga dla zgrupowanych daty i czasu w formacie tekstowym.
4. Atrybuty ze struktury `G_BASE`, które nie są używane przez JTLS, nie mają swoich odpowiedników w postaci kolumn w tabeli `base_table`.
5. Atrybutom, które występują w strukturze `G_BASE` jako typ wyliczeniowy, odpowiadają kolumny zawierające tekst o zmiennej długości. SDC jest odpowiedzialny za translację typów wyliczeniowych GENIS definicję JTLS, a następnie wykorzystania plików słowników do określenia tekstu, jaki powinien zostać umieszczony w kolumnie.

Powyższe różnice odnoszą się do wszystkich tabel SDC w bazie danych i odpowiadających im struktur danych używanych przez GENIS.

Dodatkowo struktura `G_GRAPHIC` jest rozdzielona na dwie tabele: jedną do przechowywania danych ogólnych i drugą do przechowywania informacji o punktach – odpowiednio tabelę `graphic_data` i tabelę `graphic_location_pts`. Takie rozwiązanie zostało przyjęte ze względu na łatwość programowania i tworzenia interfejsów, ponieważ łatwiej jest wyszukiwać punkty w uporządkowanej tabeli i odczytywać je, niż z serii dokładnie nazwanych kolumn w tabeli `graphic_data`.

7.1.3.2. Message Processing Client (MPC)

Message Processing Client (MPC) jest odpowiedzialny za umieszczanie w Scenario Data Repository (SDR) komunikatów generowanych przez CEP. Powstał on na bazie programu MPP opracowanego przez NC3A, poprzez wprowadzenie modyfikacji i rozszerzenie funkcjonalności, jednak logika przetwarzania komunikatów pozostała taka sama.

Jedna instancja MPC jest uruchamiana dla scenariusza do odbierania i składowania komunikatów o całej sytuacji i wszystkich stronach biorących udział w grze. Każde program kliencki może uzyskiwać dane dla określonej strony w grze z SDR

Wewnątrz SDR istnieją oddzielne tabele Oracle do przetwarzania i składowania poszczególnych typów komunikatów. MPC może zostać skonfigurowany do odbierania wszystkich komunikatów od CEP i składowania w tabeli `general_message`.

7.1.3.2.1. *Tabele z danymi MPC*

Do prawidłowego działania MPC potrzebuje danych z tabel SDC w bazie danych SDR. Zakłada się, że SDC jest uruchamiane przed MPC dla zapewnienia przekazywania do niego bieżących informacji o grze.

Definicje tabel związanych z MPC bazują tych, jakie wykorzystuje MPP będący podstawą do jego opracowania. Do każdej tabeli zostały dodane pola w celu dostarczenia dodatkowej funkcjonalności, takiej jak np.: możliwość obsługi przez MPC komunikatów do wszystkich stron gry.

Tabela general_message jest wypełniana danymi wówczas, gdy znacznikowi `General_Message_Flag` została przypisana wartość. Tabela ta przechowuje komunikaty w postaci nieprzetworzonej, tzn. w takiej samej, w jakiej CEP przesyła je do procesu GENIS. Jeżeli komunikat jest zbyt długi żeby zmieścić się w jednym rekordzie (baza danych Oracle nakłada ograniczenie długości do 4000 znaków), jest on dzielony na dwie lub więcej części, które są numerowane sekwencyjnie. Numer ten jest wstawiany w odpowiednie pole rekordu komunikatu. Tabela 2 przedstawia strukturę tabeli `general_message`.

Tabela 2. Struktura tabeli general_message

Nazwa kolumny	Zawartość

Nazwa kolumny	Zawartość
report_id	unikalny identyfikator każdego komunikatu (zwiększany o 1 dla każdego odebranego komunikatu)
messagetype	BROADCAST lub CLIENT
receiving_sides	specyfikacja strony/stron, dla której przeznaczony jest komunikat (każdy znak reprezentuje jedną stronę: C-Kontroler, 1-strona 1, 0-strona 0, 1C-strona 1 i Kontroler, itp.)
broadcast_ground	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
broadcast_air	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
broadcast_log	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
broadcast_naval	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
broadcast_amphib	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
broadcast_intel	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
broadcast_sof	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
broadcast_cont	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
seq_nbr	kolejny numer części dla długich komunikatów
reporttime	czas gry, kiedy komunikat został wygenerowany
messagenbr	typ komunikatu wg definicji zawartych pliku definicji komunikatów (np.: english.msg)
messagestring	tekst wiadomości

Podzbiór komunikatów programu MPP systemu JTLS jest przetwarzany przez MPC. Komunikaty te są analizowane, rozkładane i przetwarzane (interpretowane) do postaci tekstowej w odróżnieniu od komunikatów przechowywanych w postaci nieprzetworzonej w tabeli general_message. Szczegółowy opis komunikatów i struktur, których są przechowywane jest opisana w dokumentacji programu MPP opracowanej przez NC3A. Listę komunikatów przetwarzanych przez MPC zawiera tabela 3. Komunikaty z oznaczeniem (B) są komunikatami rozgłoszeniowymi (ang. *broadcast messages*).

Tabela 3. Komunikaty przetwarzane przez MPC

Nazwa komunikatu JTLS	Identyfikator komunikatu JTLS
Ground Order Complied Report	2050
Ground Order Rejected Report	2100

HRU Current Activity Trouble Report	2310
Minefield Operations Status Report	2650
Engineering Task Halted Report	2700
Engineering Task Report	2710
Ground Unit Contact Report	2750
HRU Unit Contact Report	2760
CAS Request Processed	3020
Air Order Complied Report	3050
Air Order Rejected Report	3100
Mission Report	3180
Mission Capable Report	3200
Reconnaissance Report	3900
HRU Logistics Report	4315
Foreign Unit Destroyed Report	5600
BDA Area Report	5800
Naval Mining Status Report	6250
Submarine Contact Report	6600
Controller Air Combat Report (B)	7000
Target Destruction Message	7100
Unit ADA Engagement Report	7200
Controller Damage Report (B)	7300
Unit Damage Report	7500
HRU Damage Report	7700
Combat System Summary Report (B)	9000
Ground BDA Report (B)	9140
Controller BDA Report (B)	9150

W oparciu o struktury tabel związanych z MPP zostały opracowane struktury MPC. Na potrzeby zwiększonej funkcjonalności rozszerzono je o zestaw pól, które przedstawia tabela 4.

Tabela 4. Rozszerzenie struktur tabel MPP na potrzeby MPC

Nazwa kolumny	Zawartość
receiving_sides	specyfikacja strony/stron dla której przeznaczony jest komunikat (każdy znak reprezentuje jedną stronę: C-Kontroler, 1-strona 1, 0-strona 0, 1C-strona 1 i Kontroler, itp.)
messagetype	BROADCAST lub CLIENT
broadcast_ground	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
broadcast_air	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
broadcast_log	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
broadcast_naval	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
broadcast_amphib	YES lub NO (tylko dla komunikatów rozgłoszeniowych)

Nazwa kolumny	Zawartość
broadcast_intel	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
broadcast_sof	YES lub NO (tylko dla komunikatów rozgłoszeniowych)
broadcast_cont	YES lub NO (tylko dla komunikatów rozgłoszeniowych)

7.1.3.3. Order Entry Client (OEC)

Order Entry Client (OEC) sprawdza tabele bazy danych wewnątrz SDR (Scenario Data Repository) pod kątem rozkazów, które zostały w nim umieszczone w celu przesłania ich do CEP. Jeżeli rozkazy zostały umieszczone w SDR, OEC jest odpowiedzialny za ich pobranie, sformatowanie, weryfikację oraz przesłanie do JTLS. Jeżeli rozkaz zawiera błędy, OEC informuje o tym fakcie oprogramowanie klienckie, które go przesłało, za pośrednictwem komunikatów o błędach umieszczanych w tabeli bazy danych. Oprogramowanie klienckie jest odpowiedzialne za przesyłanie rozkazów oraz monitorowanie statusu ich przesłania do JTLS.

7.1.4. Wskazówki do uwzględnienia podczas planowania

7.1.4.1. GAWS

Wewnątrz katalogu GAWS, gdzie zapisywane są dane, powinna być dostępna wystarczająca ilość przestrzeni dyskowej. Jako wskazówkę można podać, że w ćwiczeniu, gdzie na GIAC kontrolera występowało 28 000 obiektów, każdy zapis informacji o sytuacji zajmował 20 Mb. Przy problemach z zapewnieniem przestrzeni dyskowej można zmodyfikować parametr ddb_flush_interval określający długość interwału pomiędzy zapisami w minutach gry i zwiększyć jego wartość.

Dla zredukowania przestrzeni zajmowanej w procesie tworzenia bazy danych, należy pominąć parametry ddb_model_data definiujące gromadzenie danych dla takich obiektów jak: samolot (*ang. aircraft*), droga lotnicza (*ang. track*) oraz szyk, ugrupowanie (*ang. formation*). Samoloty i drogi lotnicze zmieniają swoje położenie zbyt szybko, aby możliwe było zapisanie informacji pozwalających na ich zobrazowanie w użyteczny sposób, natomiast szyk nie jest wyświetlany w GIAC jako pojedyncza pozycja.

Dalszą redukcję zajętości przestrzeni dyskowej można osiągnąć poprzez pominięcie zapisywania informacji o obiektach typu konwój (*ang. convoy*), jeżeli zobrazowanie konwojów nie jest użyteczne w fazie analizy, oceny ani omówienia ćwiczeń.

Parametr `ddb_flush_interval` określa długość interwału pomiędzy zapisami w minutach gry, w związku z czym w eksperymentach przeprowadzanych przy dużym przyspieszeniu czasu symulacji w stosunku do czasu rzeczywistego pojawiają się problemy ze składowaniem danych w bazie GAWS. Dlatego też nie należy planować wykorzystania tego narzędzia w takich eksperymentach.

GENIS może działać zarówno jako GENIS dla GAWS, jak i GENIS dla normalnych klientów. Użycie automatycznej konfiguracji przy pomocy ICP w żaden sposób nie przeszkadza GENIS dla GAWS w obsługiwaniu zwykłych klientów. Wybór wariantu zależy od rozkładu obciążenia systemu.

7.1.4.2. Post Processor System

PPS gromadzi dane tylko i wyłącznie na potrzeby analizy działania sił powietrznych. W chwili obecnej nie jest możliwe wykorzystanie go do żadnych innych celów.

Gromadzenie danych rozpoczyna sie w momencie ustawienia znacznika. Następuje to po przesłaniu przez kontrolera rozkazu SET DEBUG REPORT FLAGS (`Game Control->Set Parameters->Set Modeling Parameters->Set Debug and Report Flags -> Post Processor Data Flag = YES`).

Zmiana wartości tego znacznika na wartość "`YES`" znacząco zwiększa obszar przestrzeni dyskowej, którą trzeba posiadać, aby uruchomić model i wykonywać (tzn. składować) punkty kontrolne.

Działanie PPS jest silnie powiązane z mechanizmem wykonywania punktów kontrolnych. Dane nie będą dostępne wcześniej niż po wykonaniu kolejnych punktów kontrolnych.

7.1.4.3. Open Access Programs

Wydajność SDR zależy od konfiguracji sprzętowej oraz programowej systemu, na którym jest umiejscowiona jego baza danych Oracle takich jak np.: rozmiar SGA (ang. *System Global Area*), ilość procesorów itp. Strojenie wydajności serwera Oracle jest poważnym zagadnieniem do rozważenia z punktu widzenia działania SDR i należy wziąć w tym przypadku pod uwagę wszelkie zalecenia firmy Oracle.

SDR ma charakter aplikacji łączącej cechy systemów OLTP (ang. *Online Transaction Processing*) i DSS (ang. *Decision Support System*). Poza wyrażeniami DML (ang. *Data Manipulation Language*) wykonywanymi przez klientów GENIS wykorzystujących SDR (jak SDC, MPC, OEC) składujących w czasie rzeczywistym informacje o grze, mogą pojawiać się również złożone zapytania SQL wykonywane przez aplikacje zewnętrzne (firm trzecich). Ten fakt musi być wzięty pod uwagę podczas instalacji i konfiguracji serwera bazy danych Oracle (np. powinien być odpowiednio dobrany parametr `DB_BLOCK_SIZE`).

System nie dostarcza możliwości równoległej pracy klientów SDC (uruchomienia kilku instancji jednocześnie). W danej chwili powinien być uruchomiony tylko klient dla wszystkich stron (opcja zalecana) lub dla wybranej strony, ale nie dwóch lub więcej klientów.

SDC utrzymuje całą informację o bieżącym stanie gry na podstawie informacji otrzymywanych od procesu GENIS. Gromadzenie informacji pozwalającej na prześledzenie zmian poszczególnych obiektów wymaga dodatkowej pracy nad stworzeniem tabel dla danych archiwalnych oraz implementacji mechanizmu wyzwalaczy (ang. *triggers*) na tabelach SDC, który będzie odpowiedzialny za archiwizowanie informacji.

7.1.4.4. Porównanie funkcjonalności narzędzi

1. Nakłady na przygotowanie narzędzia do gromadzenia danych

GAWS	niskie, konfiguracja GENIS tak, aby zapisywał dane w bazie
Post Procesor System	bardzo niskie, wydanie polecenia gromadzenia danych
Open Access Programs	wysokie, przygotowanie środowiska bazodanowego, konfiguracja połączeń z GENIS i bazą danych

2. Możliwość określania zakresu gromadzonych danych

GAWS	tak
Post Procesor System	nie
Open Access Programs	określenie stron, których dane mają być gromadzone aktualne informacje, ale brak możliwości ograniczania zakresu informacji w ramach strony dowolny zakres danych archiwizowanych

3. *Dodatkowe wymagania*

GAWS	przestrzeń dyskowa
Post Procesor System	przestrzeń dyskowa serwer Oracle
Open Access Programs	serwer Oracle strojenie serwera Oracle

4. *Możliwość pracy wielu użytkowników*

GAWS	tak
Post Procesor System	w ograniczonym zakresie (problem ładowania danych),
Open Access Programs	tak, ale nie bezpośrednio z JTLS, lecz bazą SDR (Oracle).

5. *Częstotliwość aktualizacji danych*

GAWS	przekroje czasowe z ustalonym interwałem
Post Procesor System	na bieżąco
Open Access Programs	na bieżąco

6. *Dostępność danych*

GAWS	natychmiast po zapisaniu przekroju czasowego
Post Procesor System	po wykonaniu punktu kontrolnego (ang. *checkpoint*),
Open Access Programs	na bieżąco

7. *Archiwizacja danych*

GAWS	tak (zapisane w bazie przekroje czasowe)
Post Procesor System	tak – na bieżąco (wraz z mechanizmem punktu kontrolnego)
Open Access Programs	nie – tylko aktualna sytuacja w grze, możliwość dodania funkcjonalności i dostosowania do własnych potrzeb

8. *Prezentacja wyników*

GAWS	graficzna, możliwość tworzenia animacji
Post Procesor System	zestawienia tabelaryczne i raporty

Open Access Programs	brak

9. *Możliwość rozbudowy funkcjonalności na własne potrzeby*

GAWS	brak
Post Procesor System	brak
Open Access Programs	bardzo szerokie – możliwość rozbudowy baz Oracle dla uzyskania archiwizacji wybranych danych, użycia własnych narzędzi obróbki danych i raportowania

7.2. System „Złocień"

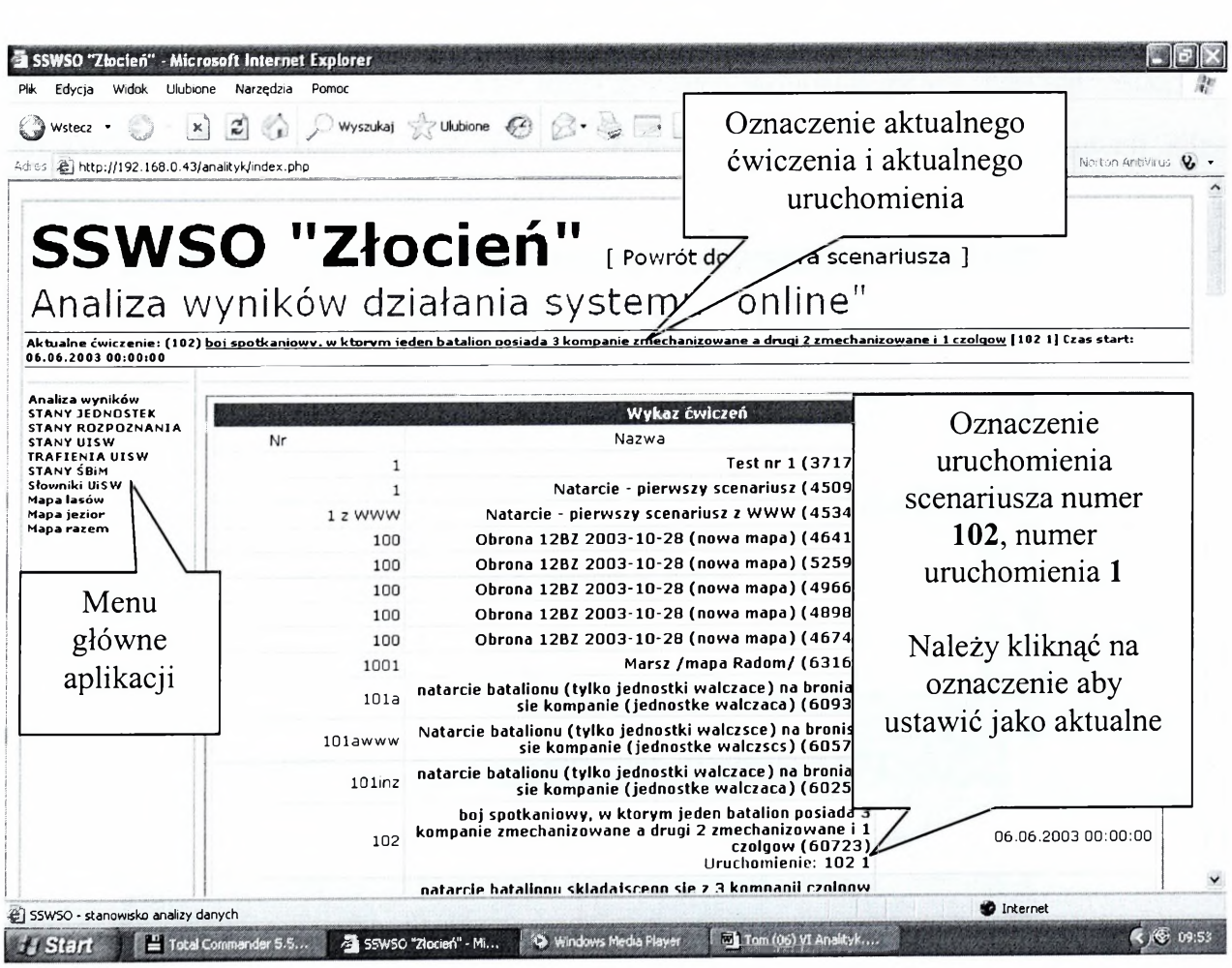
W przypadku systemu „Złocień", zakres danych potrzebnych do analizy i oceny nie jest konfigurowalny. Jedynym parametrem, który jest możliwy do zdefiniowania, jest interwał z jakim dane są zapisywane do bazy danych. Jest on podawany jako parametr uruchomienia symulatora. Dane są składowane w relacyjnej bazie danych Oracle.

Na podstawie danych zapisywanych w bazie jest możliwe:

- analizowanie informacji na bieżąco – w trakcie eksperymentu symulacyjnego;
- analizowanie zgromadzonych informacji po zakończeniu eksperymentu;
- odtwarzanie i zobrazowanie przebiegu symulacji.

7.2.1. Zobrazowanie wyników w trakcie eksperymentu

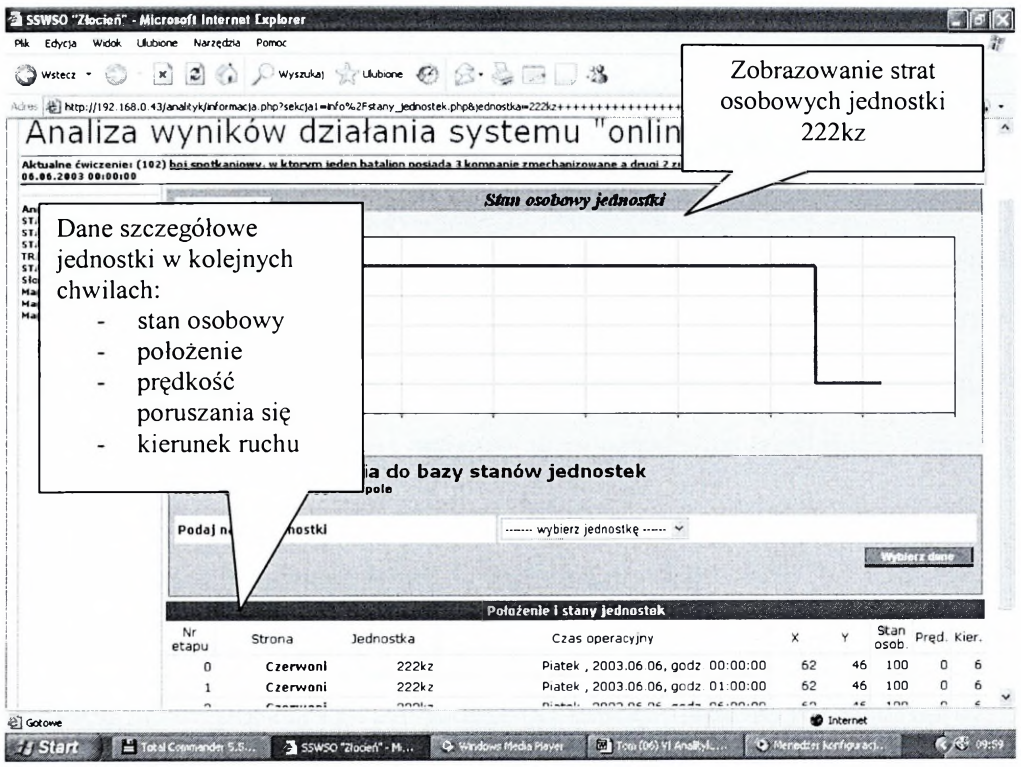
W trakcie eksperymentu dostęp do gromadzonych danych odbywa się poprzez moduł uruchamiany z wykorzystaniem przeglądarki WWW.



Rysunek 28. Okno główne aplikacji zobrazowania wyników w trakcie eksperymentu

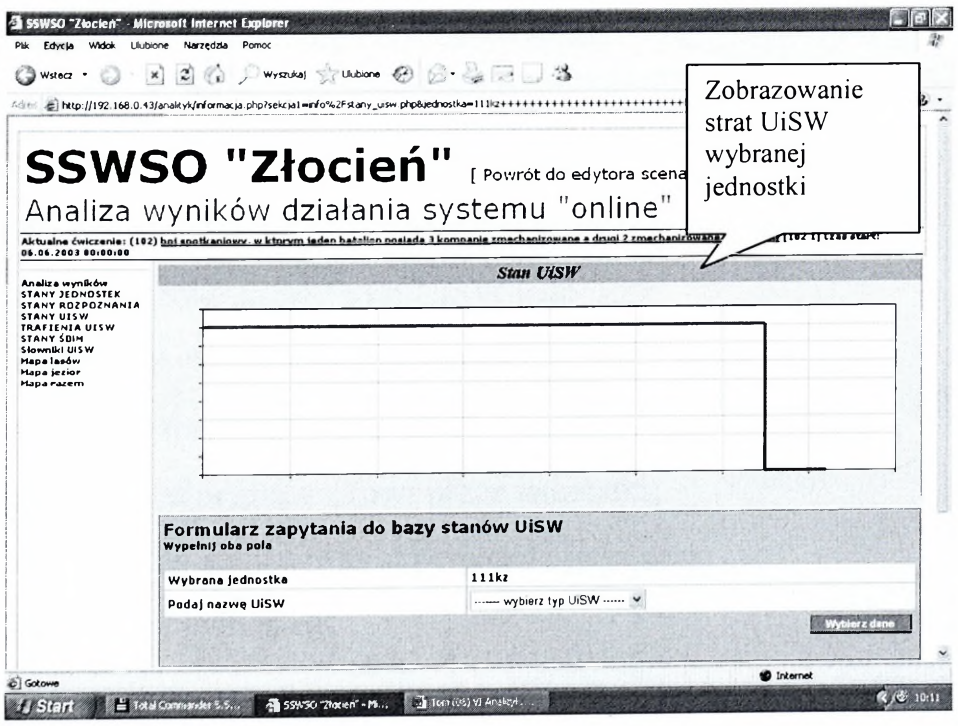
Dostępne są następujące grupy informacji:

- STANY JEDNOSTEK – stany osobowe poszczególnych jednostek biorących udział w ćwiczeniu oraz zmiana położenia jednostek;



Rysunek 29. Okno opisu stanu jednostek

- STANY ROZPOZNANIA – wykaz rozpoznanych jednostek przez inne jednostki;
- STANY UiSW – stany UiSW (straty) poszczególnych jednostek biorących udział w ćwiczeniu;



Rysunek 30. Okno stanu UiSW jednostek

- TRAFIENIA UiSW - straty zadane przez poszczególne jednostki biorące udział w ćwiczeniu;
- STANY ŚBiM – stany ŚBiM (straty) poszczególnych jednostek biorących udział w ćwiczeniu.

7.2.2. Analizy wyników ćwiczeń po ich zakończeniu

Aplikacja służąca do analizy wyników ćwiczenia po ich zakończeniu została wykonana jako skoroszyt arkusza kalkulacyjnego Microsoft Excel. Po zasileniu go danymi zebranymi w trakcie eksperymentu pozwala ona na analizę analogicznych grup informacji, jakie były dostępne w trakcie przebiegu symulacji opisane w poprzednim rozdziale, czyli:

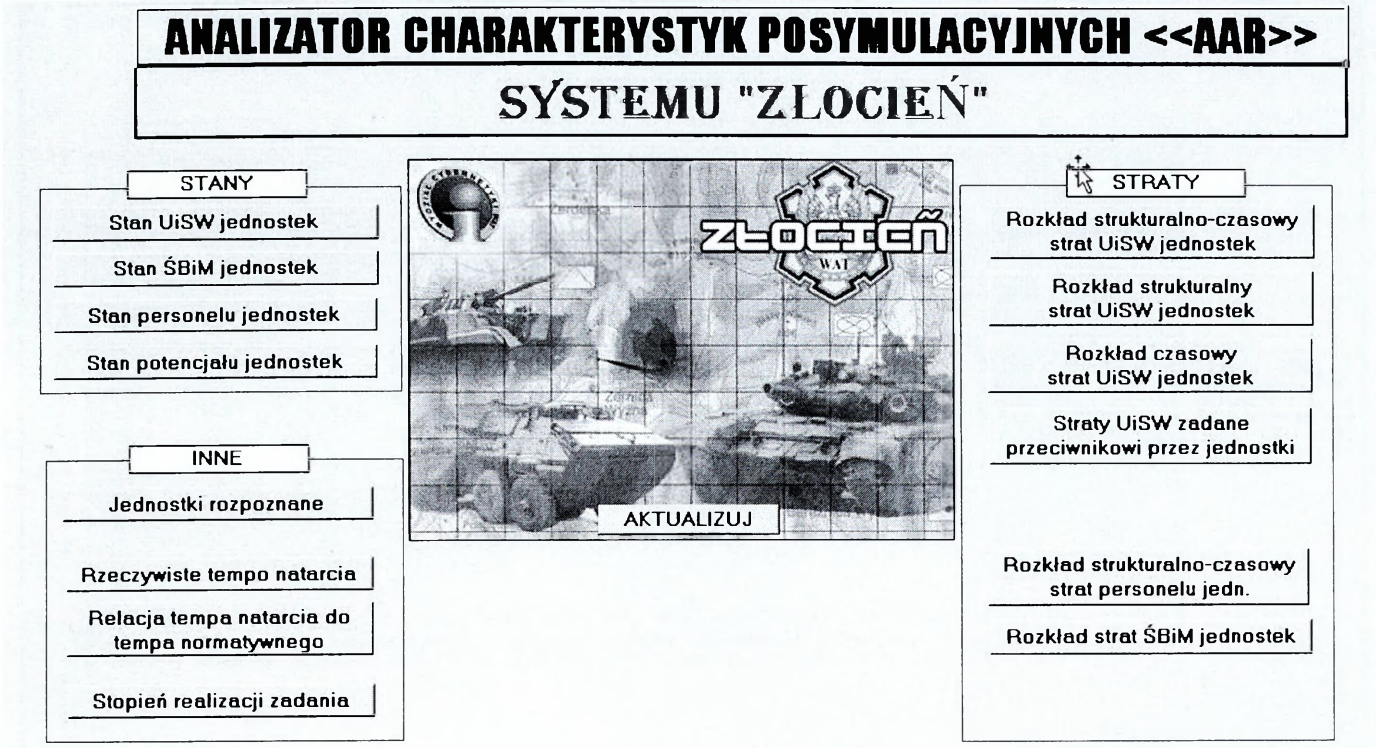
- stany osobowe poszczególnych jednostek biorących udział w ćwiczeniu;
- wykaz rozpoznanych jednostek przez inne jednostki;
- stany UiSW (straty) poszczególnych jednostek biorących udział w ćwiczeniu;
- straty zadane przez poszczególne jednostki biorące udział w ćwiczeniu;
- stany ŚBiM (straty) poszczególnych jednostek biorących udział w ćwiczeniu;

oraz dodatkowe charakterystyki

- stanu potencjału bojowego;
- rzeczywiste tempo natarcia;
- relacja tempa natarcia do tempa normatywnego;
- stopień realizacji zadania.

Dla strat możliwe są również następujące bardziej szczegółowe analizy:

- rozkład strukturalno-czasowy strat UiSW jednostek;
- rozkład strukturalny strat UiSW jednostek;
- rozkład czasowy strat UiSW jednostek;
- straty UiSW zadane przeciwnikowi przez jednostki;
- rozkład strukturalno-czasowy strat personelu jednostek;
- rozkład strat ŚBiM jednostek.



Rysunek 31. Okno programu sterującego wraz z menu aplikacji

Wybór poszczególnych opcji menu powoduje przejście do arkusza kalkulacyjnego w postaci tabeli przestawnej z możliwością generowania wykresu.

MENU GŁÓWNE

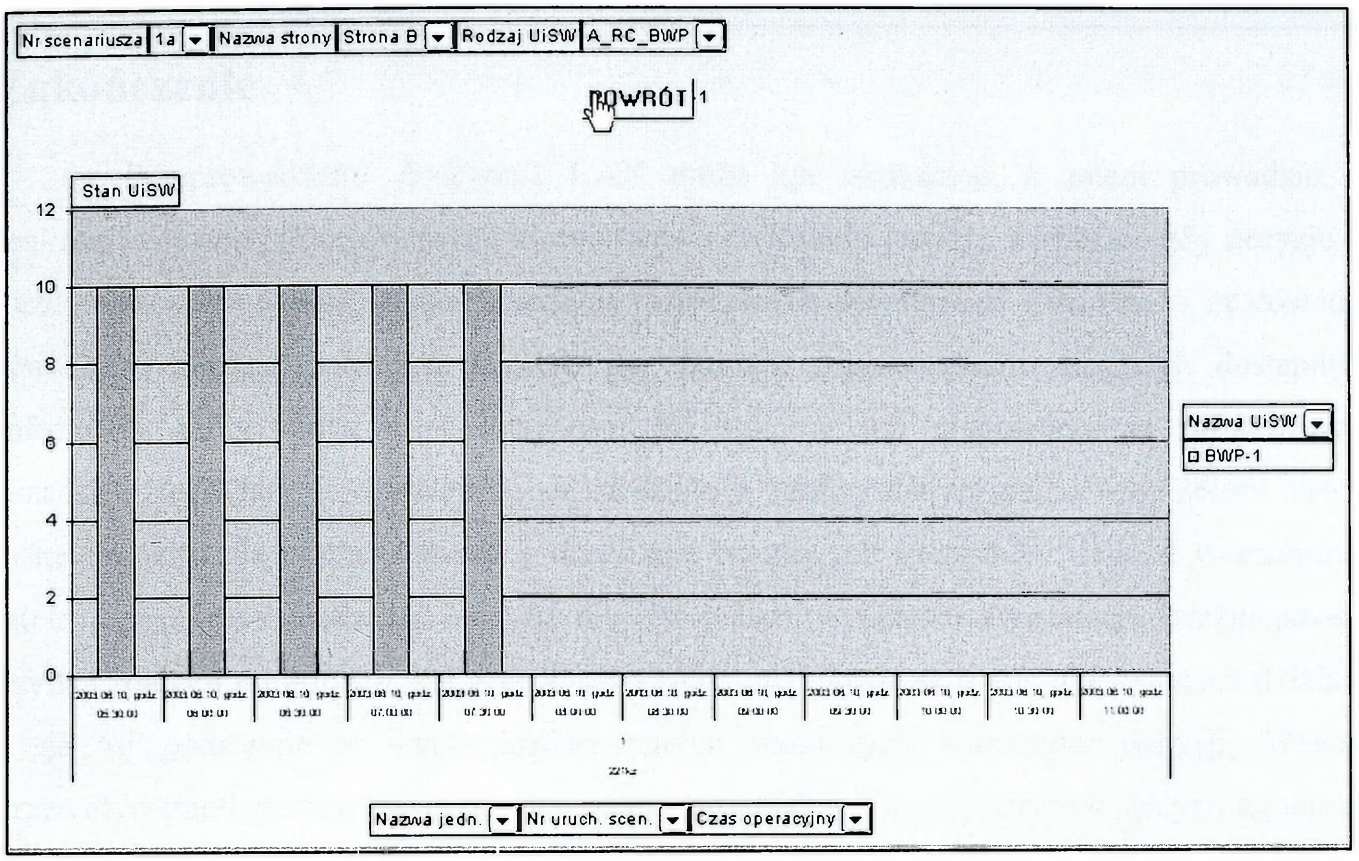
STAN UiSW JEDNOSTEK

Nr scenariusza	1a
Nazwa strony	Strona B
Rodzaj UiSW	A_RC_BWP

WYKRES

Stan UiSW			Nazwa UiSW	
Nazwa jedn.	Nr uruch. scen.	Czas operacyjny	BWP-1	Suma końcowa
221kz	1	2003.06.10, godz. 05:30:00	10	10
		2003.06.10, godz. 06:00:00	10	10
		2003.06.10, godz. 06:30:00	10	10
		2003.06.10, godz. 07:00:00	10	10
		2003.06.10, godz. 07:30:00	10	10
		2003.06.10, godz. 08:00:00	0	0
		2003.06.10, godz. 08:30:00	0	0
		2003.06.10, godz. 09:00:00	0	0
		2003.06.10, godz. 09:30:00	0	0
		2003.06.10, godz. 10:00:00	0	0
		2003.06.10, godz. 10:30:00	0	0
		2003.06.10, godz. 11:00:00	0	0
	1 Średnia		4,17	4,17
	1 Maksimum		10	0
	1 Minimum		0	0
221kz Średnia			4,17	4,17
221kz OdchStd			5,15	5,15

Rysunek 32. Przykładowe okno charakterystyk



Rysunek 33. Przykładowe okno wykresu charakterystyki

Zakończenie

Przeprowadzenie ćwiczenia CAX może być skuteczne, a zatem prowadzić do realizacji założonych celów przez kierownictwo ćwiczenia jedynie wówczas, gdy decyzje na każdym szczeblu dowodzenia są oparte na racjonalnych podstawach. Oznacza to, że dowódca (ćwiczący) wydaje polecenia dopiero po zebraniu i przebadaniu wszelkich dostępnych informacji, które mogą mieć znaczenie dla sukcesu lub niepowodzenia zamierzonych przedsięwzięć. Dane te opracowuje się za pomocą analizy decyzyjnej, a więc zbioru metod, które umożliwiają ocenę sytuacji i określenie możliwych kierunków działań (wariantów). Istotą procesu podejmowania decyzji nie jest jednak wyłącznie konstatacja teraźniejszości, czy też zdarzeń minionych, ale przede wszystkim znajomość skutków planowanych działań i – na tej podstawie – wybór najlepszego z możliwych wariantów decyzji. Właśnie przewidywanie konsekwencji jest warunkiem racjonalnego podejmowania decyzji zgodnie z pozytywistyczną zasadą: *savoir pour prèvoir afin de pouvoir* (wiedzieć, aby przewidywać, a zatem móc – A. Comte). Przyszłość nie jest jednak oglądalna bezpośrednio, wszelkie sądy o niej są tylko mniej lub bardziej spekulatywnymi hipotezami. Jednak poprzez szczegółowe studiowanie zjawisk, uważną obserwację otaczającej rzeczywistości, a zwłaszcza stosowanie dynamicznych sposobów analizy informacji np. metodą symulacji, można wyizolować przesłanki, które pozwolą – z dużą pewnością – formułować wnioski o tym, jak zachowa się obiekt poddany zamierzonym działaniom, czy też określonym bodźcom środowiskowym. Realizacja tej metody wymaga zaplanowania i przygotowania szczegółowego eksperymentu symulacyjnego, (najczęściej w postaci bazy danych i scenariusza, według którego realizowany jest proces symulacji.

Opracowanie zawiera syntezę prac badawczych przeprowadzonych przez zespół autorów nad jednym z przedsięwzięć realizowanym podczas przygotowania i prowadzenia ćwiczenia dowódczo - sztabowe typu CAX. W opracowaniu przedstawiona została koncepcja planowania eksperymentów w środowisku systemów symulacyjnych.

Problem podjęty przez autorów jest z jednej strony – stosunkowo złożony i interdyscyplinarny, a z drugiej strony – jego rozwiązanie niezwykle istotne z punktu widzenia powodzenia w realizacji ćwiczenia CAX w oparciu o posiadane systemy symulacyjne. Jego rozwiązanie wymaga wnikliwego podejścia od strony technologii informatycznych oraz stosownej wiedzy operacyjnej z dziedziny nauk wojskowych. Ze względu jednak na jednostkowe, jak dotychczas, doświadczenia autorów w eksploatacji systemów symulacyjnych (JTLS), niemożliwe okazało się opracowanie w pełni dojrzałej metodyki

planowania eksperymentów symulacyjnych dla systemu JTLS oraz ZŁOCIEŃ, gdyż oprócz wiedzy teoretycznej, która jak się okazało nie jest wystarczająca, konieczne jest uwzględnienie pragmatyki, co będzie możliwe po zrealizowaniu przynajmniej jednego ćwiczenia CAX w CSiKGW z zastosowaniem obu systemów.symulacyjnych.

Literatura

1. Bieniek S.: Narzędzia wsparcia fazy analizy i oceny ćwiczeń CAX w systemie symulacji walki JTLS, AON, Warszawa, 2004
2. GIAC Overview Release 1.7, Los Alamos National Laboratory, Los Alamos, 1997
3. GIAC Player's Guide Release 1.9, Los Alamos National Laboratory, Los Alamos, 1999
4. GIAC Model Controller's GuideRelease 1.9, Los Alamos National Laboratory, Los Alamos, 1999
5. JTLS Executive Overview v.2.6, U.S. JFCOM Joint Warfighting Center, Suffolk, 2003
6. JTLS Technical Coordinator's Guide v.2.6, U.S. JFCOM Joint Warfighting Center, Suffolk, 2003
7. JTLS PPS User's Guide v.2.6, U.S. JFCOM Joint Warfighting Center, Suffolk, 2003
8. JTLS Software Maintenance Manual v.2.6, U.S. JFCOM Joint Warfighting Center, Suffolk, 2003
9. Kaliński K., Pietrzak G., Boryn P.: Symulacyjny system walki JTLS, AON, Warszawa, 2003
10. Urbanek M., Błażejczyk W., Grzyb J., Świeboda H.: Analiza systemowa i prognozowanie w symulacji, AON, Warszawa, 2003
11. Kaliński K., Boryn P., Bieniek S., Pietrzak G.: Opracowanie eksperymentu symulacyjnego – realizacja epizodu taktycznego, AON, 2004

Dodatki

Dane z przebiegu eksperymentu gromadzone w tabelach SDC systemu JTLS

Tabela 5. Odwzorowanie struktur GENIS w tabele danych SDC

Nazwa struktury GENIS	Nazwa tabeli SDC
G_AC_MAINT	ac_maint_data
G_AIRCRAFT	aircraft_data
G_ARC	arc_data
G_BACKORDER	backorder_data
G_BASE	base_data
G_CAS_REQUEST	cas_request_data
G_CONVOY	convoy_data
G_CRUISE	cruise_data
G_DEPOT	depot_data
G_DMPI	dmpi_data
G_DMPI_TARGET_LINK	dmpi_target_link_data
G_EQUIPMENT	equipment_data
G_FACTION	faction_data
G_FARP	farp_data
G_FIRE_MISSION	fire_mission_data
G_FORMATION	formation_data
G_GRAPHIC	graphic_data
	graphic_location_pts
G_GROUND	ground_data
G_GROUND_DAMAGE	ground_damage_data
G_HEX	hex_data
G_HRU	hru_data
G_IADS	iads_data
G_INTEL	
G_MODEL_STATUS	model_status_data
G_NODE	node_data
G_PENDING_ORDER	pending_order_data
G_PIPELINE	pipeline_data
G_PORT_EVENT	port_event_data
G_RAILROAD	railroad_data
G_ROE	roe_data
G_RUNWAY_CUT	runway_cut_data
G_SHIP	ship_data

Nazwa struktury GENIS	Nazwa tabeli SDC
G_SIDE_RELATION	side_relation_data
G_SQUADRON	squadron_data
G_SUPPLY_CATEGORY	supply_category_data
G_TARGET	target_data
G_TRACK	track_data
G_UI_GROUND	ui_ground_data
G_UI_TARGET	ui_target_data
G_WEAPONLOAD	weaponload_data
G_WEATHER	weather_data

Tabela 6. Odwzorowanie atrybutów struktury G_AC_MAINT w kolumny tabeli ac_maint_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_AC_MAINT_FLAG_SAID	object_index
	object_data_type/ SDC_Object_Type
G_AC_MAINT_FLAG_OWNING_UNIT	owning_unit_index
	owning_unit_data_type/ SDC_Object_Type
G_AC_MAINT_FLAG_SIDE	side/No_Force_Side
G_AC_MAINT_FLAG_NBR_AIRCRAFT	nbr_aircraft
G_AC_MAINT_FLAG_MAINT_END_TIME	maint_end_time_seconds
	maint_end_time_text

Tabela 7. Odwzorowanie atrybutów struktury G_AIRCRAFT w kolumny tabeli aircraft_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_AIRCRAFT_ID	object_index
	object_data_type/ SDC_Object_Type
G_AIRCRAFT_OWNING_SQUADRON	owning_squadron_index
	owning_squadron_data_type/ SDC_Object_Type
G_AIRCRAFT_ASSOC_UNIT_ID	assoc_unit_id_index
	assoc_unit_id_data_type/ SDC_Object_Type
G_AIRCRAFT_REFUELEE	refuelee_index
	refuelee_data_type/ SDC_Object_Type
G_AIRCRAFT_LAT	Lat
G_AIRCRAFT_LON	Lon
G_AIRCRAFT_ALTITUDE	Altitude
G_AIRCRAFT_SPEED	Speer
G_AIRCRAFT_COURSE	Course
G_AIRCRAFT_SIDE	side/No_Force_Side
G_AIRCRAFT_MISSION	mission/Mission_Type
G_AIRCRAFT_AIRCRAFT_STATUS	aircraft_status/ Air_Mission_Posture

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_AIRCRAFT_ACTIVE	active/Yes_No_Flag
G_AIRCRAFT_GHOST	ghost/Yes_No_Flag
G_AIRCRAFT_CONCURRENT_STATUS	con_status
G_AIRCRAFT_FUEL	Fuel
G_AIRCRAFT_FUEL_PERCENT	fuel_percent
G_AIRCRAFT_NUMBER	number_aircraft
G_AIRCRAFT_NUMBER_TASKED	number_tasked
G_AIRCRAFT_NBR_AC_INTERCEPTED	nbr_ac_intercepted
G_AIRCRAFT_SYMBOL	symbol/Graph_Symbol
G_AIRCRAFT_NAME	Name
G_AIRCRAFT_AC_TYPE	ac_type
G_AIRCRAFT_EVENT	Event
G_AIRCRAFT_TIME_NEXT_CHANGE	time_next_change_seconds
	time_next_change_text
G_AIRCRAFT_ETR	etr_seconds
	etr_text
G_AIRCRAFT_TARGET_TRACK	target_track
G_AIRCRAFT_DEST1_LAT	dest1_lat
G_AIRCRAFT_DEST1_LON	dest1_lon
G_AIRCRAFT_DEST2_LAT	dest2_lat
G_AIRCRAFT_DEST2_LON	dest2_lon
G_AIRCRAFT_RADIUS	Radius
G_AIRCRAFT_MISSION_CLASS	mission_class/Mission_Type
G_AIRCRAFT_LOAD	Load
G_AIRCRAFT_FIXED_WING	fixed_wing/Fixed_Wing_Flag
G_AIRCRAFT_MODE1_SQUAWK	mode1_squawk
G_AIRCRAFT_MODE2_SQUAWK	mode2_squawk
G_AIRCRAFT_MODE3_SQUAWK	mode3_squawk
G_AIRCRAFT_NEXT_COORD_LAT	next_coord_lat
G_AIRCRAFT_NEXT_COORD_LON	next_coord_lon
G_AIRCRAFT_NEXT_COORD_TIME	next_coord_time_seconds
	next_coord_time_text

Tabela 8. Odwzorowanie atrybutów struktury G_ARC w kolumny tabeli arc_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_ARC_ID	object_index
	object_data_type/ SDC_Object_Type
G_ARC_NETWORK_ID	network_id_index
	network_id_data_type/ SDC_Object_Type
G_ARC_SOURCE_ID	source_id_index
	source_id_data_type/ SDC_Object_Type
G_ARC_SINK_ID	sink_id_index
	sink_id_data_type/ SDC_Object_Type
G_ARC_RATE	rate
G_ARC_DISTANCE	distance
G_ARC_NET_STATUS	status/Network_Status
G_ARC_NET_DIRECTION	direction/Network_Direction
G_ARC_NAME	name

Tabela 9. Odwzorowanie atrybutów struktury G_BACKORDER w kolumny tabeli backorder_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
	object_index
G_BACKORDER_FLAG_SAID	object_data_type/ SDC_Object_Type
	owning_unit_index
G_BACKORDER_FLAG_OWNING_UNIT	owning_unit_data_type/ SDC_Object_Type
	owed_to_unit_index
G_BACKORDER_FLAG_OWED_TO_UNIT	owed_to_unit_data_type/ SDC_Object_Type
G_BACKORDER_FLAG_CACHE_LAT	cache_lat
G_BACKORDER_FLAG_CACHE_LON	cache_lon
G_BACKORDER_FLAG_SIDE	side/No_Force_Side
G_BACKORDER_FLAG_PRIORITY	priority/Requisition_Type
G_BACKORDER_FLAG_REQUESTED_TIME	requested_time_seconds
	requested_time_text

Tabela 10. Odwzorowanie atrybutów struktury G_BASE w kolumny tabeli base_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_BASE_ID	object_index
	object_data_type/ SDC_Object_Type
G_BASE_HHQ	hhq_index
	hhq_data_type/ SDC_Object_Type
G_BASE_REINF	reinf_index
	reinf_data_type/ SDC_Object_Type
G_BASE_SUPPORT	support_index
	support_data_type/ SDC_Object_Type
G_BASE_RUNWAY_ID	runway_id_index
	runway_id_data_type/ SDC_Object_Type
G_BASE_LIFTING_FORMATION	lifting_formation_index
	lifting_formation_data_type/ SDC_Object_Type
G_BASE_LAT	lat
G_BASE_LON	lon
G_BASE_DEST_LAT	dest_lat
G_BASE_DEST_LON	dest_lon
G_BASE_SPEED	speed
G_BASE_AMT_AV_FUEL_ON_HAND	amt_av_fuel_on_hand
G_BASE_AMT_AV_FUEL_DUE_IN	amt_av_fuel_due_in
G_BASE_AMT_CL_IV_ON_HAND	amt_cl_iv_on_hand
G_BASE_AMT_CL_IV_DUE_IN	amt_cl_iv_due_in
G_BASE_PSYOP_RESULT_MULT	psyop_result_mult
G_BASE_AC_PARKING	ac_parking
G_BASE_AC_SHELTERS	ac_shelters

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_BASE_NBR_SUBORDS	nbr_subords
G_BASE_NBR_REINF_UNITS	nbr_reinf_units
G_BASE_NBR_TARGETS_OWNED	nbr_targets_owned
G_BASE_NBR_MHE_TARGETS_OWNED	nbr_mhe_targets_owned
G_BASE_NBR_UNITS_SUPPORTED	nbr_units_supported
G_BASE_NBR_BACKORDER_DUE_IN	nbr_backorder_due_in
G_BASE_NBR_BACKORDER_DUE_OUT	nbr_backorder_due_out
G_BASE_NBR_CONVOY_INBOUND	nbr_convoy_inbound
G_BASE_NBR_SQUADRONS_AT_BASE	nbr_squadrons_at_base
G_BASE_TOE_AC_AT_BASE	toe_ac_at_base
G_BASE_STRENGTH	strength
G_BASE_ACTIVE	active/Yes_No_Flag
G_BASE_TYPE	type/Base_Type
G_BASE_SIDE	side/No_Force_Side
G_BASE_SIZE	unit_size/Unit_Size
G_BASE_COMBAT_STATUS	in_combat/Yes_No_Flag
	in_contact/Yes_No_Flag
G_BASE_SYMBOL	symbol/Graph_Symbol
G_BASE_PERCEPTION	perception/Perception
G_BASE_POSTURE	posture/Unit_Posture
G_BASE_MISSION	mission/Unit_Posture
G_BASE_ORIENT	orientation/Orientation
G_BASE_DETECT_TIME	detect_time_seconds
	detect_time_text
G_BASE_POSTURE_TIME	posture_time_seconds
	posture_time_text
G_BASE_NAME	name
G_BASE_DETECT_SOURCE	detect_source
G_BASE_FACTION	faction
G_BASE_COMMAND_WS	command_ws
G_BASE_PROTOTYPE_NAME	prototype
G_BASE_INFO	info
G_BASE_GHOST	ghost/Yes_No_Flag
G_BASE_FACTION_ID	faction_id_index
	faction_id_data_type/ SDC_Object_Type
G_BASE_SERVICE	service/Service_Names
G_BASE_SOF_ALERT	sof_alert/Yes_No_Flag

Tabela 11. Odwzorowanie atrybutów struktury G_CAS_REQUEST w kolumny tabeli cas_request_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_CAS_REQUEST_FLAG_SAID	object_index
	object_data_type/ SDC_Object_Type
G_CAS_REQUEST_FLAG_REQUESTING_ UNIT	requesting_unit_index
	requesting_unit_data_type/ SDC_Object_Type
G_CAS_REQUEST_FLAG_SIDE	side/No_Force_Side
G_CAS_REQUEST_FLAG_NBR_AC_ REQUESTED	nbr_ac_requested
G_CAS_REQUEST_FLAG_CAS_IS_LATE	cas_is_late
G_CAS_REQUEST_FLAG_REQUESTED_ TIME	requested_time_seconds
	requested_time_text
G_CAS_REQUEST_FLAG_TYPE_ REQUESTED	type_requested

Tabela 12. Attribute/Column Names For G_CONVOY w kolumny tabeli convoy_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_CONVOY_ID	object_index
	object_data_type/ SDC_Object_Type
G_CONVOY_HOME_UNIT	home_unit_index
	home_unit_data_type/ SDC_Object_Type
G_CONVOY_ALSP_ID	alsp_id_index
	alsp_id_data_type/ SDC_Object_Type
G_CONVOY_LAT	lat
G_CONVOY_LON	lon
G_CONVOY_DEST_LAT	dest_lat
G_CONVOY_DEST_LON	dest_lon
G_CONVOY_SPEED	speed
G_CONVOY_SIDE	side/No_Force_Side
G_CONVOY_ORIENT	orientation/Orientation
G_CONVOY_SYMBOL	symbol/Graph_Symbol
G_CONVOY_CONVOY_TYPE	convoy_type/Type_Convoy
G_CONVOY_PERCEPTION	perception/Perception
G_CONVOY_STRENGTH	strength
G_CONVOY_ACTIVE	active/Yes_No_Flag
G_CONVOY_NAME	name
G_CONVOY_FACTION_ID	faction_id_index
	faction_id_data_type/ SDC_Object_Type
G_CONVOY_RECEIVING_UNIT	receiving_unit_index
	receiving_unit_data_type/ SDC_Object_Type
G_CONVOY_RAILROAD	railroad_index
	railroad_data_type/ SDC_Object_Type
G_CONVOY_ORIG_CARGO_TRUCKS	orig_cargo_trucks
G_CONVOY_ORIG_TANKER_TRUCKS	orig_tanker_trucks
G_CONVOY_ORIG_RAILCARS	orig_railcars
G_CONVOY_ORIG_BARGES	orig_barges
G_CONVOY_CURR_CARGO_TRUCKS	curr_cargo_trucks
G_CONVOY_CURR_TANKER_TRUCKS	curr_tanker_trucks
G_CONVOY_CURR_RAILCARS	curr_railcars
G_CONVOY_CURR_BARGES	curr_barges
G_CONVOY_STATUS	status
G_CONVOY_ORIG_OTHER_TRUCKS	orig_other_trucks
G_CONVOY_ORIG_HET	orig_HET
G_CONVOY_CURR_OTHER_TRUCKS	curr_other_trucks
G_CONVOY_CURR_HET	curr_HET

Tabela 13. Odwzorowanie atrybutów struktury G_CRUISE w kolumny tabeli cruise_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_CRUISE_ID	object_index
	object_data_type/ SDC_Object_Type

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_CRUISE_LAT	lat
G_CRUISE_LON	lon
G_CRUISE_ALTITUDE	altitude
G_CRUISE_SPEED	speed
G_CRUISE_COURSE	course
G_CRUISE_SIDE	side/No_Force_Side
G_CRUISE_SYMBOL	symbol/Graph_Symbol
G_CRUISE_NAME	name
G_CRUISE_ACTIVE	active/Yes_No_Flag
G_CRUISE_GHOST	ghost/Yes_No_Flag

Tabela 14. Odwzorowanie atrybutów struktury G_DEPOT w kolumny tabeli depot_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_DEPOT_ID	object_index
	object_data_type/SDC_Object_Type
G_DEPOT_HHQ	hhq_index
	hhq_data_type/SDC_Object_Type
G_DEPOT_REINF	reinf_index
	reinf_data_type/SDC_Object_Type
G_DEPOT_SUPPORT	support_index
	support_data_type/SDC_Object_Type
G_DEPOT_LIFTING_FORMATION	lifting_formation_index
	lifting_formation_data_type/ SDC_Object_Type
G_DEPOT_LAT	lat
G_DEPOT_LON	lon
G_DEPOT_DEST_LAT	dest_lat
G_DEPOT_DEST_LON	dest_lon
G_DEPOT_SPEED	speed
G_DEPOT_ARTY_RANGE	arty_range
G_DEPOT_AMT_GROUND_FUEL_HAND	amt_ground_fuel_on_hand
G_DEPOT_PSYOP_RESULT_MULT	psyop_result_mult
G_DEPOT_NBR_SUBORDS	nbr_subords
G_DEPOT_NBR_REINF_UNITS	nbr_reinf_units
G_DEPOT_NBR_TARGETS_OWNED	nbr_targets_owned
G_DEPOT_NBR_MHE_TARGETS_OWNED	nbr_mhe_targets_owned
G_DEPOT_NBR_UNITS_SUPPORTED	nbr_units_supported
G_DEPOT_NBR_BACKORDER_DUE_	nbr_backorder_due_in
G_DEPOT_NBR_BACKORDER_DUE_	nbr_backorder_due_out
G_DEPOT_NBR_CONVOY_INBOUND	nbr_convoy_inbound
G_DEPOT_TOE_CARGO_TRUCKS	toe_cargo_trucks
G_DEPOT_TOE_TANKER_TRUCKS	toe_tanker_trucks
G_DEPOT_CARGO_TRUCKS_AVAIL	cargo_trucks_avail
G_DEPOT_TANKER_TRUCKS_AVAIL	tanker_trucks_avail
G_DEPOT_NBR_CONVOYS_DISPATCHED	nbr_convoys_dispatched
G_DEPOT_TYPE	type/Unit_Type
G_DEPOT_SIDE	side/No_Force_Side
G_DEPOT_SIZE	unit_size/Unit_Size
G_DEPOT_HEADING	heading/Orientation
G_DEPOT_ORIENT	orientation/Orientation
G_DEPOT_POSTURE	posture/Unit_Posture
G_DEPOT_MISSION	mission/Unit_Posture
G_DEPOT_SYMBOL	symbol/Graph_Symbol

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_DEPOT_PERCEPTION	perception/Perception
G_DEPOT_DETECT_TIME	detect_time_seconds
	detect_time_text
G_DEPOT_POSTURE_TIME	posture_time_seconds
	posture_time_text
G_DEPOT_NAME	name
G_DEPOT_DETECT_SOURCE	detect_source
G_DEPOT_FACTION	faction
G_DEPOT_PROTOTYPE_NAME	prototype
G_DEPOT_STRENGTH	strength
G_DEPOT_STATUS	in_combat/Yes_No_Flag
	in_contact/Yes_No_Flag
G_DEPOT_ACTIVE	active/Yes_No_Flag
G_DEPOT_COMMAND_WS	command_ws
G_DEPOT_INFO	info
G_DEPOT_GHOST	ghost/Yes_No_Flag
G_DEPOT_FACTION_ID	faction_id_index
	faction_id_data_type/ SDC_Object_Type
G_DEPOT_SERVICE	service/Service_Names
G_DEPOT_SOF_ALERT	sof_alert/Yes_No_Flag

Tabela 15. Odwzorowanie atrybutów struktury G_DMPI w kolumny tabeli dmpi_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_DMPI_SAID	object_index
	object_data_type/SDC_Object_Type
G_DMPI_NUMBER	name_number
G_DMPI_DESCRIP	description
G_DMPI_CATEGORY	damage_category/Target_Category
G_DMPI_SUBCATEGORY	damage_subcategory
G_DMPI_SYMBOL	symbol/Graph_Symbol
G_DMPI_LAT	lat
G_DMPI_LON	lon
G_DMPI_PERCENT_CAPABLE	percent_capable

Tabela 16. Odwzorowanie atrybutów struktury G_DMPI_TARGET_LINK w kolumny tabeli dmpi_target_link_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_DMPI_TARGET_LINK_ID	object_index
	object_data_type/SDC_Object_Type
G_DMPI_TARGET_LINK_TARGET	target_index
	target_data_type/SDC_Object_Type
G_DMPI_TARGET_LINK_DMPI	dmpi_index
	dmpi_data_type/SDC_Object_Type

Tabela 17. Odwzorowanie atrybutów struktury G_EQUIPMENT w kolumny tabeli equipment_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_EQUIPMENT_ID	object_index
	object_data_type/SDC_Object_Type
G_EQUIPMENT_UNIT_ID	unit_said_index
	unit_said_data_type/ SDC_Object_Type
G_EQUIPMENT_AUTHORIZED	authorized
G_EQUIPMENT_ON_HAND	on_hand
G_EQUIPMENT_IN_MAINT	in_maint
G_EQUIPMENT_IN_USE	in_use
G_EQUIPMENT_OPERATIONAL	operational
G_EQUIPMENT_MAX_EVER_ OPERATIONAL	max_ever_operational
G_EQUIPMENT_VALUE_OF_ONE	value_of_one
G_EQUIPMENT_MANNED	manned
G_EQUIPMENT_EQUIP_NAME	equip_name
G_EQUIPMENT_ISSUE_CATEGORY	issue_category
G_EQUIPMENT_WHOLE_MANNED	whole_manned

Tabela 18. Odwzorowanie atrybutów struktury G_FACTION w kolumny tabeli faction_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_FACTION_ID	object_index
	object_data_type/SDC_Object_Type
G_FACTION_UNIT_SIDE	side/No_Force_Side
G_FACTION_NAME	name
G_FACTION_FACTION_TYPE	faction_type/Faction_Affiliation
G_FACTION_COUNTRY_CODE	country_code
G_FACTION_COMBAT_SYSTEM_PROTOTYPE	combat_system_prototype/ Combat_System_Prototype
G_FACTION_SUSTAINMENT_LOG_PROTOTYPE	sustainment_log_prototype/ Sustainment_Log_Prototype
G_FACTION_FIRE_LETH_PROTO	fire_lethality_prototype/ Fire_Lethality_Prototype

Tabela 19. Odwzorowanie atrybutów struktury G_FARP w kolumny tabeli farp_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_FARP_ID	object_index
	object_data_type/ SDC_Object_Type
G_FARP_HHQ	hhq_index
	hhq_data_type/ SDC_Object_Type
G_FARP_REINF	reinf_index
	reinf_data_type/ SDC_Object_Type
G_FARP_SUPPORT	support_index
	support_data_type/ SDC_Object_Type

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_FARP_LIFTING_FORMATION	lifting_formation_index
	lifting_formation_data_type/ SDC_Object_Type
G_FARP_LAT	lat
G_FARP_LON	lon
G_FARP_DEST_LAT	dest_lat
G_FARP_DEST_LON	dest_lon
G_FARP_SPEED	speed
G_FARP_ARTY_RANGE	arty_range
G_FARP_AMT_AV_FUEL_ON_HAND	amt_av_fuel_on_hand
G_FARP_PSYOP_RESULT_MULT	psyop_result_mult
G_FARP_NBR_SUBORDS	nbr_subords
G_FARP_NBR_REINF_UNITS	nbr_reinf_units
G_FARP_NBR_TARGETS_OWNED	nbr_targets_owned
G_FARP_NBR_MHE_TARGETS_OWNED	nbr_mhe_targets_owned
G_FARP_NBR_UNITS_SUPPORTED	nbr_units_supported
G_FARP_NBR_BACKORDER_DUE_IN	nbr_backorder_due_in
G_FARP_NBR_BACKORDER_DUE_OUT	nbr_backorder_due_out
G_FARP_NBR_CONVOY_INBOUND	nbr_convoy_inbound
G_FARP_NBR_SQUADRONS_AT_BASE	nbr_squadrons_at_base
G_FARP_TOE_AC_AT_BASE	toe_ac_at_base
G_FARP_SIDE	side/No_Force_Side
G_FARP_SIZE	unit_size/Unit_Size
G_FARP_ORIENT	orientation/Orientation
G_FARP_POSTURE	posture/Unit_Posture
G_FARP_MISSION	mission/Unit_Posture
G_FARP_SYMBOL	symbol/Graph_Symbol
G_FARP_PERCEPTION	perception/Perception
G_FARP_NAME	name
G_FARP_DETECT_TIME	detect_time_seconds
	detect_time_text
G_FARP_POSTURE_TIME	posture_time_seconds
	posture_time_text
G_FARP_DETECT_SOURCE	detect_source
G_FARP_FACTION	faction
G_FARP_STRENGTH	strength
G_FARP_STATUS	in_combat/Yes_No_Flag
	in_contact/Yes_No_Flag
G_FARP_ACTIVE	active/Yes_No_Flag
G_FARP_COMMAND_WS	command_ws
G_FARP_INFO	info
G_FARP_GHOST	ghost/Yes_No_Flag
G_FARP_FACTION_ID	faction_id_index
	faction_id_data_type/ SDC_Object_Type
G_FARP_COURSE	course
G_FARP_SERVICE	service/Service_Names
G_FARP_SOF_ALERT	sof_alert/Yes_No_Flag
G_FARP_PROTOTYPE_NAME	prototype

Tabela 20. Odwzorowanie atrybutów struktury G_FIRE_MISSION w kolumny tabeli fire_mission_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_FIRE_MISSION_FLAG_SAID	object_index

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	object_data_type/ SDC_Object_Type
G_FIRE_MISSION_FLAG_FIRING_UNIT	firing_unit_index
	firing_unit_data_type/ SDC_Object_Type
G_FIRE_MISSION_FLAG_AIM_POINT_TARGET	aim_point_target_index
	aim_point_target_data_type/ SDC_Object_Type
G_FIRE_MISSION_FLAG_AIM_POINT_UNIT	aim_point_unit_index
	aim_point_unit_data_type/ SDC_Object_Type
G_FIRE_MISSION_FLAG_SIDE	side/No_Force_Side
G_FIRE_MISSION_FLAG_IMPACT_LAT	impact_lat
G_FIRE_MISSION_FLAG_IMPACT_LON	impact_lon
G_FIRE_MISSION_FLAG_FIRE_AT_RANGE	fire_at_range
G_FIRE_MISSION_FLAG_FIRE_AT_BEARING	fire_at_bearing
G_FIRE_MISSION_FLAG_FIRE_MISSION_STATUS	fire_mission_status
G_FIRE_MISSION_FLAG_NBR_ROUNDS	nbr_rounds
G_FIRE_MISSION_FLAG_REQUESTED_TIME	requested_time_seconds
	requested_time_text
G_FIRE_MISSION_FLAG_AMMO_FIRED	ammo_fired
G_FIRE_MISSION_FLAG_START_NLT_TIME	start_nlt_time_seconds
	start_nlt_time_text

Tabela 21. Odwzorowanie atrybutów struktury G_FORMATION w kolumny tabeli formation_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_FORMATION_ID	object_index
	object_data_type/ SDC_Object_Type
G_FORMATION_LAT	lat
G_FORMATION_LON	lon
G_FORMATION_SIDE	side/No_Force_Side
G_FORMATION_NAME	name
G_FORMATION_MISSION	mission/Formation_Posture
G_FORMATION_NBR_UNITS_EMBARKED	nbr_units_embarked

Tabela 22. Odwzorowanie atrybutów struktury G_GRAPHIC w kolumny tabeli graphic_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_GRAPHIC_ID	object_index
	object_data_type/ SDC_Object_Type
G_GRAPHIC_ASSOC_ID	associated_object_index
	associated_object_data_type/ SDC_Object_Type
G_GRAPHIC_TYPE	type/Graphic_Type
G_GRAPHIC_COLOR	color/Graph_Color
G_GRAPHIC_LINE_TYPE	line_style/Graph_Line
G_GRAPHIC_FILL_TYPE	fill_pattern/Graph_Fill_Pattern

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_GRAPHIC_STATION_TYPE	station_type/Station_Keeping_Method
G_GRAPHIC_ACTIVE	active/Yes_No_Flag
G_GRAPHIC_NAME	name
G_GRAPHIC_LATPTS G_GRAPHIC_LONPTS	Współrzędne punktów przechowywane są w tabeli graphic_location_pts.

Tabela 23. Odwzorowanie atrybutów struktury G_GROUND w kolumny tabeli ground_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_GROUND_ID	object_index
	object_data_type/ SDC_Object_Type
G_GROUND_HHQ	hhq_index
	hhq_data_type/ SDC_Object_Type
G_GROUND_REINF	reinf_index
	reinf_data_type/ SDC_Object_Type
G_GROUND_SUPPORT	support_index
	support_data_type/ SDC_Object_Type
G_GROUND_LIFTING_FORMATION	lifting_formation_index
	lifting_formation_data_type/ SDC_Object_Type
G_GROUND_LAT	lat
G_GROUND_LON	lon
G_GROUND_DEST_LAT	dest_lat
G_GROUND_DEST_LON	dest_lon
G_GROUND_SPEED	speed
G_GROUND_SPDMAX	spdmax
G_GROUND_AMT_GROUND_FUEL_HAND	amt_ground_fuel_on_hand
G_GROUND_AMT_GROUND_FUEL_DUE_IN	amt_ground_fuel_due_
G_GROUND_AMT_CL_IV_ON_HAND	amt_cl_iv_on_hand
G_GROUND_PSYOP_RESULT_MULT	psyop_result_mult
G_GROUND_THRESHOLD	threshold
G_GROUND_STATUS	in_combat/Yes_No_Flag
	in_contact/Yes_No_Flag
G_GROUND_NBR_SUBORDS	nbr_subords
G_GROUND_NBR_REINF_UNITS	nbr_reinf_units
G_GROUND_NBR_TARGETS_OWNED	nbr_targets_owned
G_GROUND_NBR_MHE_TARGETS_OWNED	nbr_mhe_targets_owned
G_GROUND_NBR_UNITS_SUPPORTED	nbr_units_supported
G_GROUND_NBR_BACKORDER_	nbr_backorder_due_in
G_GROUND_NBR_BACKORDER_OUT	nbr_backorder_due_out
G_GROUND_NBR_CONVOY_INBOUND	nbr_convoy_inbound
G_GROUND_NBR_FIRE_MISSIONS	nbr_fire_missions
G_GROUND_NBR_PENDING_ORDERS	nbr_pending_orders
G_GROUND_BRANCH	branch/Ground_Branch
G_GROUND_TYPE	type/Ground_Unit_Type
G_GROUND_SIDE	side/No_Force_Side
G_GROUND_SIZE	unit_size/Unit_Size
G_GROUND_HEADING	heading/Orientation
G_GROUND_ORIENT	orientation/Orientation

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_GROUND_POSTURE	posture/Unit_Posture
G_GROUND_MISSION	mission/Unit_Posture
G_GROUND_SYMBOL	symbol/Graph_Symbol
G_GROUND_PERCEPTION	perception/Perception
G_GROUND_NAME	name
G_GROUND_INFO	info
G_GROUND_DETECT_TIME	detect_time_seconds
	detect_time_text
G_GROUND_POSTURE_TIME	posture_time_seconds
	posture_time_text
G_GROUND_DETECT_SOURCE	detect_source
G_GROUND_FACTION	faction
G_GROUND_PROTOTYPE_NAME	prototype
G_GROUND_STRENGTH	strength
G_GROUND_ARTY_RANGE	arty_range
G_GROUND_SOF_ALERT	sof_alert/Yes_No_Flag
G_GROUND_ACTIVE	active/Yes_No_Flag
G_GROUND_COMMAND_WS	command_ws
G_GROUND_GHOST	ghost/Yes_No_Flag
G_GROUND_FACTION_ID	faction_id_index
	faction_id_data_type/ SDC_Object_Type
G_GROUND_SERVICE	service/Service_Names
G_GROUND_COURSE	course

Tabela 24. Odwzorowanie atrybutów struktury G_GROUND_DAMAGE w kolumny tabeli ground_damage_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_GROUND_DAMAGE_ID	object_index
	object_data_type/ SDC_Object_Type
G_GROUND_DAMAGE_POWER	owner_index
G_GROUND_DAMAGE_LAT	lat
G_GROUND_DAMAGE_LON	lon
G_GROUND_DAMAGE_AC_DESTROYED	nbr_systems
G_GROUND_DAMAGE_SIDE	side/No_Force_Side
G_GROUND_DAMAGE_BE_TARGET_NUMBER	system_name
G_GROUND_DAMAGE_TIME_DAMAGED	damage_time_seconds
	damage_time_text

Tabela 25. Odwzorowanie atrybutów struktury G_HEX w kolumny tabeli hex_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_TERRAIN_ID	object_index
	object_data_type/ SDC_Object_Type
G_TERRAIN_EW_INDEX	ew_index
G_TERRAIN_NS_INDEX	ns_index
G_TERRAIN_BARRIER_N	barrier_n/Barrier_Type

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_TERRAIN_BARRIER_NE	barrier_ne/Barrier_Type
G_TERRAIN_BARRIER_SE	barrier_se/Barrier_Type
G_TERRAIN_BARRIER_S	barrier_s/Barrier_Type
G_TERRAIN_BARRIER_SW	barrier_sw/Barrier_Type
G_TERRAIN_BARRIER_NW	barrier_nw/Barrier_Type
G_TERRAIN_MOBILITY	mobility/Terrain_Type
G_TERRAIN_CONTAMINATION	contamination/ Terrain_Contamination

Tabela 26. Odwzorowanie atrybutów struktury G_HRU w kolumny tabeli hru_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_HRU_ID	object_index
	object_data_type/ SDC_Object_Type
G_HRU_OWN_UNIT_ID	owning_unit_index
	owning_unit_data_type/ SDC_Object_Type
G_HRU_ASSOC_UNIT_ID	assoc_unit_index
	assoc_unit_data_type/ SDC_Object_Type
G_HRU_FACTION_ID	faction_id_index
	faction_id_data_type/ SDC_Object_Type
G_HRU_LAT	lat
G_HRU_LON	lon
G_HRU_DEST_LAT	dest_lat
G_HRU_DEST_LON	dest_lon
G_HRU_SPEED	speed
G_HRU_TEAM_RADIUS	team_radius
G_HRU_POSTURE	posture/Unit_Posture
G_HRU_MISSION_TYPE	mission/Unit_Posture
G_HRU_SIDE	side/No_Force_Side
G_HRU_ORIENT	orientation/Orientation
G_HRU_SYMBOL	symbol/Graph_Symbol
G_HRU_PERCEPTION	perception/Perception
G_HRU_NEXT_REPORT_TIME	next_report_time_seconds
	next_report_time_text
G_HRU_OFF_STATION_TIME	off_station_time_seconds
	off_station_time_text
G_HRU_POSTURE_TIME	posture_time_seconds
	posture_time_text
G_HRU_NAME	name
G_HRU_PROTOTYPE_NAME	prototype
G_HRU_SOF_COVERT	sof_covert/Yes_No_Flag
G_HRU_ACTIVE	active/Yes_No_Flag
G_HRU_GHOST	ghost/Yes_No_Flag
G_HRU_COURSE	course
G_HRU_COMBAT_STATUS	in_combat/Yes_No_Flag
	in_contact/Yes_No_Flag
G_HRU_SERVICE	service/Service_Names
G_HRU_STRENGTH	strength
G_HRU_SOF_ALERT	sof_alert/Yes_No_Flag

Tabela 27. Odwzorowanie atrybutów struktury G_IADS w kolumny tabeli iads_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_IADS_FLAG_SAID	object_index
	object_data_type/ SDC_Object_Type
G_IADS_FLAG_NAME	name

Tabela 28. Odwzorowanie atrybutów struktury G_MODEL_STATUS w kolumny tabeli model_status_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_MODEL_STATUS_MODEL	model
G_MODEL_STATUS_STAT	status/Model_Status
G_MODEL_STATUS_GAME_SPEED	game_speed
G_MODEL_STATUS_CURRENT_TIME	current_time_seconds
	current_time_text
G_MODEL_STATUS_START_TIME	start_time_seconds

Tabela 29. Odwzorowanie atrybutów struktury G_NODE w kolumny tabeli node_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_NODE_ID	object_index
	object_data_type/ SDC_Object_Type
G_NODE_NETWORK_ID	network_id_index
	network_id_data_type/ SDC_Object_Type
G_NODE_LAT	lat
G_NODE_LON	lon
G_NODE_OFFLOAD	offload
G_NODE_FLOW	flow
G_NODE_NET_STATUS	status/Network_Status
G_NODE_NAME	name

Tabela 30. Odwzorowanie atrybutów struktury G_PENDING_ORDER w kolumny tabeli pending_order_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_PENDING_ORDER_FLAG_SAID	object_index
	object_data_type/ SDC_Object_Type
G_PENDING_ORDER_FLAG_OWNING_UNIT	owning_unit_index
	owning_unit_data_type/ SDC_Object_Type
G_PENDING_ORDER_FLAG_AFFECTED_TARGET	affected_target_index
	affected_target_data_type/ SDC_Object_Type

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_PENDING_ORDER_FLAG_SIDE	side/No_Force_Side
G_PENDING_ORDER_FLAG_EXECUTION_LAT	execution_lat
G_PENDING_ORDER_FLAG_EXECUTION_LON	execution_lon
G_PENDING_ORDER_FLAG_REQUESTED_TIME	requested_time_seconds
	requested_time_text
G_PENDING_ORDER_FLAG_ORDER_TYPE	order_type

Tabela 31. Odwzorowanie atrybutów struktury G_PIPELINE w kolumny tabeli pipeline_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_PIPELINE_FLAG_SAID	object_index
	object_data_type/ SDC_Object_Type
G_PIPELINE_FLAG_OPERATING_UNIT	operating_unit_index
	operating_unit_data_type/ SDC_Object_Type
G_PIPELINE_FLAG_SIDE	side/No_Force_Side
G_PIPELINE_FLAG_NBR_PUMPING_STATIONS	nbr_pumping_stations
G_PIPELINE_FLAG_NBR_STORAGE_TANKS	nbr_storage_tanks
G_PIPELINE_FLAG_NAME	name

Tabela 32. Odwzorowanie atrybutów struktury G_PORT_EVENT w kolumny tabeli port_event_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_PORT_EVENT_FLAG_SAID	object_index
	object_data_type/ SDC_Object_Type
G_PORT_EVENT_FLAG_PORT_UNIT	port_unit_index
	port_unit_data_type/ SDC_Object_Type
G_PORT_EVENT_FLAG_AFFECTED_UNIT	affected_unit_index
	affected_unit_data_type/ SDC_Object_Type
G_PORT_EVENT_FLAG_ARRIVAL_LAT	arrival_lat
G_PORT_EVENT_FLAG_ARRIVAL_LON	arrival_lon
G_PORT_EVENT_FLAG_SIDE	side/No_Force_Side
G_PORT_EVENT_FLAG_TYPE_EVENT	type_event/Port_Events
G_PORT_EVENT_FLAG_IS_SHIPMENT_LATE	is_shipment_late
G_PORT_EVENT_FLAG_MEMBER_PRIORITY	member_priority
G_PORT_EVENT_FLAG_ARRIVAL_TIME	arrival_time_seconds
	arrival_time_text

Tabela 33. Odwzorowanie atrybutów struktury G_RAILROAD w kolumny tabeli railroad_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_RAILROAD_FLAG_SAID	object_index
	object_data_type/ SDC_Object_Type

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_RAILROAD_FLAG_NAME	name

Tabela 34. Odwzorowanie atrybutów struktury G_ROE w kolumny tabeli roe_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_ROE_FLAG_SAID	object_index
	object_data_type/ SDC_Object_Type
G_ROE_FLAG_OWNING_OBJECT	owning_object_index
	owning_object_data_type/ SDC_Object_Type
G_ROE_FLAG_MSSN_SQUADRON	mssn_squadron_index
	mssn_squadron_data_type/ SDC_Object_Type
G_ROE_FLAG_GROUND_ROE	roe_distance_ground
G_ROE_FLAG_SUR_AIR_ROE	roe_distance_surf_air
G_ROE_FLAG_AIR_AIR_ROE	roe_distance_air_air
G_ROE_FLAG_PURSUIT	shadow_distance
G_ROE_FLAG_SIDE	side/No_Force_Side
G_ROE_FLAG_AGAINST_SIDE	against_side/No_Force_Side
G_ROE_FLAG_ALTITUDE	ground_roe
G_ROE_FLAG_ROE_DISTANCE_GROUND	sur_air_roe
G_ROE_FLAG_ROE_DISTANCE_SURF_AIR	air_air_roe
G_ROE_FLAG_ROE_DISTANCE_AIR_AIR	pursuit
G_ROE_FLAG_SHADOW_DISTANCE	altitude

Tabela 35. Odwzorowanie atrybutów struktury G_RUNWAY_CUT w kolumny tabeli runway_cut_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_RUNWAY_CUT_FLAG_SAID	object_index
	object_data_type/ SDC_Object_Type
G_RUNWAY_CUT_FLAG_OWNING_TARGET	owning_target_index
	owning_target_data_type/ SDC_Object_Type
G_RUNWAY_CUT_FLAG_LENGTH	length
G_RUNWAY_CUT_FLAG_REPAIR_SCHEDULED	repair_scheduled_seconds
	repair_scheduled_text

Tabela 36. Odwzorowanie atrybutów struktury G_SHIP w kolumny tabeli ship_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_SHIP_ID	object_index
	object_data_type/ SDC_Object_Type
G_SHIP_HHQ	hhq_index

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	hhq_data_type/ SDC_Object_Type
G_SHIP_REINF	reinf_index
	reinf_data_type/ SDC_Object_Type
G_SHIP_FORMATION_ID	formation_id_index
	formation_id_data_type/ SDC_Object_Type
G_SHIP_LAT	lat
G_SHIP_LON	lon
G_SHIP_DEST_LAT	dest_lat
G_SHIP_DEST_LON	dest_lon
G_SHIP_BEARING	bearing
G_SHIP_SPEED	speed
G_SHIP_SPDMAX	spdmax
G_SHIP_AMT_AV_FUEL_ON_HAND	amt_av_fuel_on_hand
G_SHIP_PSYOP_RESULT_MULT	psyop_result_mult
G_SHIP_DEPTH	depth
G_SHIP_RANGE	range
G_SHIP_AC_CAPACITY	ac_capacity
G_SHIP_NBR_SUBORDS	nbr_subords
G_SHIP_NBR_TARGETS_OWNED	nbr_targets_owned
G_SHIP_NBR_MHE_TARGETS_OWNED	nbr_mhe_targets_owned
G_SHIP_NBR_UNITS_SUPPORTED	nbr_units_supported
G_SHIP_NBR_BACKORDER_DUE_IN	nbr_backorder_due_in
G_SHIP_NBR_BACKORDER_DUE_OUT	nbr_backorder_due_out
G_SHIP_NBR_CONVOY_INBOUND	nbr_convoy_inbound
G_SHIP_NBR_SQUADRONS_ON_SHIP	nbr_squadrons_on_ship
G_SHIP_TOE_AC_ON_SHIP	toe_ac_on_ship
G_SHIP_TYPE	type/Ship_Basic_Types
G_SHIP_SIDE	side/No_Force_Side
G_SHIP_SIZE	unit_size/Unit_Size
G_SHIP_HEADING	heading/Orientation
G_SHIP_ORIENT	orientation/Orientation
G_SHIP_POSTURE	posture/Unit_Posture
G_SHIP_SYMBOL	symbol/Graph_Symbol
G_SHIP_PERCEPTION	perception/Perception
G_SHIP_STRENGTH	strength
G_SHIP_CATAPULTS_OPERATIONAL	catapults_operational/Yes_No_Flag
G_SHIP_COMBAT_STATUS	in_combat/Yes_No_Flag
	in_contact/Yes_No_Flag
G_SHIP_MISSION	mission/Unit_Posture
G_SHIP_ACTIVE	active/Yes_No_Flag
G_SHIP_STATUS_FLAG	ship_status/Ship_On_Station
G_SHIP_SUBTYPE_FLAG	subtype/Ship_Subtypes
G_SHIP_DETECT_TIME	detect_time_seconds
	detect_time_text
G_SHIP_POSTURE_TIME	posture_time_seconds
	posture_time_text
G_SHIP_DETECT_SOURCE	detect_source
G_SHIP_NAME	name
G_SHIP_INFO	info
G_SHIP_FACTION	faction
G_SHIP_SHIP_CLASS	ship_class
G_SHIP_COMMAND_WS	command_ws
G_SHIP_GHOST	ghost/Yes_No_Flag
G_SHIP_FACTION_ID	faction_id_index

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	faction_id_data_type/ SDC_Object_Type
G_SHIP_SERVICE	service/Service_Names
G_SHIP_SOF_ALERT	sof_alert/Yes_No_Flag

Tabela 37. Odwzorowanie atrybutów struktury G_SIDE_RELATION w kolumny tabeli side_relation_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_SIDE_RELATION_ID	object_index
	object_data_type/ SDC_Object_Type
G_SIDE_RELATION_VIEW_SIDE	view_side/No_Force_Side
G_SIDE_RELATION_OTHER_SIDE	other_side/No_Force_Side
G_SIDE_RELATION_RELATIONSHIP	relationship/Relationship

Tabela 38. Odwzorowanie atrybutów struktury G_SQUADRON w kolumny tabeli squadron_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_SQUADRON_ID	object_index
	object_data_type/ SDC_Object_Type
G_SQUADRON_HHQ	hhq_index
	hhq_data_type/ SDC_Object_Type
G_SQUADRON_HOMEBASE_ID	homebase_id_index
	homebase_id_data_type/ SDC_Object_Type
G_SQUADRON_REINF	reinf_index
	reinf_data_type/ SDC_Object_Type
G_SQUADRON_SUPPORT	support_index
	support_data_type/ SDC_Object_Type
G_SQUADRON_LAT	lat
G_SQUADRON_LON	lon
G_SQUADRON_DEST_LAT	dest_lat
G_SQUADRON_DEST_LON	dest_lon
G_SQUADRON_SPEED	speed
G_SQUADRON_SPDMAX	spdmax
G_SQUADRON_RANGE	arty_range
G_SQUADRON_AMT_AV_FUEL_ON_HAND	amt_av_fuel_on_hand
G_SQUADRON_PSYOP_RESULT_MULT	psyop_result_mult
G_SQUADRON_RUNWAY_LENGTH	runway_length
G_SQUADRON_TOE_AC	toe_ac
G_SQUADRON_NBR_SUBORDS	nbr_subords
G_SQUADRON_NBR_REINF_UNITS	nbr_reinf_units
G_SQUADRON_NBR_TARGETS_OWNED	nbr_targets_owned
G_SQUADRON_NBR_MHE_TARGETS_OWNED	nbr_mhe_targets_owned
G_SQUADRON_NBR_UNITS_SUPPORTED	nbr_units_supported
G_SQUADRON_NBR_BACKORDER_DUE_IN	nbr_backorder_due_in
G_SQUADRON_NBR_BACKORDER_DUE_OUT	nbr_backorder_due_out

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_SQUADRON_NBR_CONVOY_INBOUND	nbr_convoy_inbound
G_SQUADRON_NBR_SHELTERS_OWNED	nbr_shelters_owned
G_SQUADRON_NBR_AC_AVAILABLE	nbr_ac_available
G_SQUADRON_NBR_AC_FLYING	nbr_ac_flying
G_SQUADRON_NBR_AC_ON_ALERT	nbr_ac_on_alert
G_SQUADRON_NBR_AC_IN_MAINT	nbr_ac_in_maint
G_SQUADRON_NBR_AC_WEAPONS_DELAY	nbr_ac_weapons_delay
G_SQUADRON_NBR_AC_AIRCRAFT_DELAY	nbr_ac_aircraft_delay
G_SQUADRON_NBR_AC_LIFT_UNIT_DELAY	nbr_ac_lift_unit_delay
G_SQUADRON_NBR_AC_FUEL_DELAY	nbr_ac_fuel_delay
G_SQUADRON_NBR_AC_LIFT_LOAD_DELAY	nbr_ac_lift_load_delay
G_SQUADRON_NBR_AC_RUNWAY_DELAY	nbr_ac_runway_delay
G_SQUADRON_MAX_SORTIES_PER_CYLCE	max_sorties_per_cycle
G_SQUADRON_SORTIES_THIS_CYCLE	sorties_this_cycle
G_SQUADRON_SORTIES_LAST_CYCLE	sorties_last_cycle
G_SQUADRON_SORTIES_2_CYCLES_AGO	sorties_2_cycles_ago
G_SQUADRON_SORTIES_3_CYCLES_AGO	sorties_3_cycles_ago
G_SQUADRON_BRANCH	branch/Ground_Branch
G_SQUADRON_TYPE	type/Ground_Unit_Type
G_SQUADRON_SIDE	side/No_Force_Side
G_SQUADRON_SIZE	unit_size/Unit_Size
G_SQUADRON_POSTURE	posture/Unit_Posture
G_SQUADRON_SYMBOL	symbol/Graph_Symbol
G_SQUADRON_PERCEPTION	perception/Perception
G_SQUADRON_NAME	name
G_SQUADRON_INFO	info
G_SQUADRON_AC_TYPE	ac_type
G_SQUADRON_POSTURE_TIME	posture_time_seconds
	posture_time_text
G_SQUADRON_DETECT_TIME	detect_time_seconds
	detect_time_text
G_SQUADRON_DETECT_SOURCE	detect_source
G_SQUADRON_FACTION	faction
G_SQUADRON_STRENGTH	strength
G_SQUADRON_COMBAT_STATUS	in_combat/Yes_No_Flag
	in_contact/Yes_No_Flag
G_SQUADRON_ACTIVE	active/Yes_No_Flag
G_SQUADRON_ORIGINAL	original
G_SQUADRON_MISSION	mission/Unit_Posture
G_SQUADRON_COMMAND_WS	command_ws
G_SQUADRON_PROTOTYPE_NAME	prototype
G_SQUADRON_GHOST	ghost/Yes_No_Flag
G_SQUADRON_NBR_AC_RESERVED	nbr_ac_reserved
G_SQUADRON_FACTION_ID	faction_id_index
	faction_id_data_type/ SDC_Object_Type
G_SQUADRON_COURSE	course
G_SQUADRON_SERVICE	service/Service_Names
G_SQUADRON_SOF_ALERT	sof_alert/Yes_No_Flag

Tabela 39. Odwzorowanie atrybutów struktury G_SUPPLY_CATEGORY w kolumny tabeli supply_category_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_SUPPLY_CATEGORY_FLAG_SAID	object_index
	object_data_type/ SDC_Object_Type
G_SUPPLY_CATEGORY_FLAG_OWNING_UNIT	owning_unit_index
	owning_unit_data_type/ SDC_Object_Type
G_SUPPLY_CATEGORY_FLAG_ON_HAND	on_hand
G_SUPPLY_CATEGORY_FLAG_REORDER_LEVEL	reorder_level
G_SUPPLY_CATEGORY_FLAG_STOCK_OBJECTIVE	stock_objective
G_SUPPLY_CATEGORY_FLAG_DUE_IN	due_in
G_SUPPLY_CATEGORY_FLAG_BASIC_LOAD	basic_load
G_SUPPLY_CATEGORY_FLAG_SHIPMENT_METHOD	shipment_method/Shipment_Type
G_SUPPLY_CATEGORY_FLAG_NAME	name
G_SUPPLY_CATEGORY_FLAG_SUPPORT	support_index
	support_data_type/ SDC_Object_Type

Tabela 40. Odwzorowanie atrybutów struktury G_TARGET w kolumny tabeli target_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_TARGET_ID	object_index
	object_data_type/ SDC_Object_Type
G_TARGET_OWN_UNIT_ID	own_unit_id_index
	own_unit_id_data_type/ SDC_Object_Type
G_TARGET_PIPELINE_ID	pipeline_id_index
	pipeline_id_data_type/ SDC_Object_Type
G_TARGET_ASSOC_UNIT_ID	assoc_unit_id_index
	assoc_unit_id_data_type/ SDC_Object_Type
G_TARGET_LAT	lat
G_TARGET_LON	lon
G_TARGET_CURRENT_LENGTH	current_length
G_TARGET_RANGE	range
G_TARGET_NUM_TARGETS	num_targets
G_TARGET_NUM_DESTROYED	num_destroyed
G_TARGET_NUM_BUSY	num_busy
G_TARGET_CURRENT_CUTS	current_cuts
G_TARGET_CCF	ccf
G_TARGET_DETECT_TIME	detect_time_seconds
	detect_time_text
G_TARGET_DETECT_SOURCE	detect_source
G_TARGET_FACTION	faction
G_TARGET_CATEGORY	category/Target_Category
G_TARGET_SIDE	side/No_Force_Side
G_TARGET_SYMBOL	symbol/Graph_Symbol
G_TARGET_PERCEPTION	perception/Perception
G_TARGET_CAPABILITY	capability
G_TARGET_MOBILITY_FLAG	mobility/Target_Mobility
G_TARGET_SUBCATEGORY	target_subcategory
G_TARGET_STATUS	status/Target_Status
G_TARGET_IS_TARGET_ON	is_target_on/Yes_No_Flag

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_TARGET_ACTIVE	active/Yes_No_Flag
G_TARGET_FACTION_ID	faction_id_index
	faction_id_data_type/ SDC_Object_Type
G_TARGET_THREAT_ZONE1	threat_zone1_altitude
	threat_zone1_range
G_TARGET_THREAT_ZONE2	threat_zone2_altitude
	threat_zone2_range
G_TARGET_THREAT_ZONE3	threat_zone3_altitude
	threat_zone3_range
G_TARGET_THREAT_ZONE4	threat_zone4_altitude
	threat_zone4_range
G_TARGET_THREAT_ZONE5	threat_zone5_altitude
	threat_zone5_range
G_TARGET_WEAPON_NAME	weapon_name
G_TARGET_GHOST	ghost/Yes_No_Flag
G_TARGET_WEAPON_COUNT	weapon_count
G_TARGET_ADV_WEAPON_NAME	adv_weapon_name
G_TARGET_ADV_WEAPON_COUNT	adv_weapon_count

Tabela 41. Odwzorowanie atrybutów struktury G_TRACK w kolumny tabeli track_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_TRACK_ID	object_index
	object_data_type/ SDC_Object_Type
G_TRACK_DETECTEE	detectee_index
	detectee_data_type/ SDC_Object_Type
G_TRACK_LAT	lat
G_TRACK_LON	lon
G_TRACK_ALTITUDE	altitude
G_TRACK_COURSE	course
G_TRACK_SPEED	speed
G_TRACK_SIDE	side/No_Force_Side
G_TRACK_DETECTING_SIDE	detecting_side/No_Force_Side
G_TRACK_TYPE	type/Track_Types
G_TRACK_ID_TYPE	id_type/Relationship
G_TRACK_SYMBOL	symbol/Graph_Symbol
G_TRACK_SINKING	sinking_flag/Yes_No_Flag
G_TRACK_ACTIVE	active/Yes_No_Flag
G_TRACK_NAME	name
G_TRACK_LAST_DET_TIME	last_detect_time_seconds
	last_detect_time_text
G_TRACK_NUMBER	number_aircraft
G_TRACK_AC_TYPE	ac_type
G_TRACK_MISSION	mission/Mission_Type
G_TRACK_AIRCRAFT_STATUS	aircraft_status/Air_Mission_Posture
G_TRACK_MODE1_SQUAWK	mode1_squawk
G_TRACK_MODE2_SQUAWK	mode2_squawk
G_TRACK_MODE3_SQUAWK	mode3_squawk

Tabela 42. Odwzorowanie atrybutów struktury G_UI_GROUND w kolumny tabeli ui_ground_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_UI_GROUND_SAID	object_index
	object_data_type/ SDC_Object_Type
G_UI_GROUND_LAT	lat
G_UI_GROUND_LON	lon
G_UI_GROUND_STRENGTH	strength
G_UI_GROUND_COMBAT_STATUS	in_combat/Yes_No_Flag
	in_contact/Yes_No_Flag
G_UI_GROUND_ACTIVE	active/Yes_No_Flag
G_UI_GROUND_SIDE	side/No_Force_Side
G_UI_GROUND_PERCEIVING_SIDE	perceiving_side/No_Force_Side
G_UI_GROUND_SIZE	unit_size/Unit_Size
G_UI_GROUND_POSTURE	posture/Unit_Posture
G_UI_GROUND_SYMBOL	symbol/Graph_Symbol
G_UI_GROUND_DETECT_TIME	detect_time_seconds
	detect_time_text
G_UI_GROUND_FACTION	faction
G_UI_GROUND_DETECT_SOURCE	detect_source
G_UI_GROUND_NAME	name
G_UI_GROUND_FACTION_ID	faction_id_index
	faction_id_data_type/ SDC_Object_Type

Tabela 43. Odwzorowanie atrybutów struktury G_UI_TARGET w kolumny tabeli ui_target_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_UI_TARGET_SAID	object_index
	object_data_type/ SDC_Object_Type
G_UI_TARGET_LAT	lat
G_UI_TARGET_LON	lon
G_UI_TARGET_CAPABILITY	capability
G_UI_TARGET_ACTIVE	active/Yes_No_Flag
G_UI_TARGET_SIDE	side/No_Force_Side
G_UI_TARGET_PERCEIVING_SIDE	perceiving_side/No_Force_Side
G_UI_TARGET_CATEGORY	category/Target_Category
G_UI_TARGET_SYMBOL	symbol/Graph_Symbol
G_UI_TARGET_DETECT_TIME	detect_time_seconds
	detect_time_text
G_UI_TARGET_FACTION	faction
G_UI_TARGET_DETECT_SOURCE	detect_source
G_UI_TARGET_NAME	name
G_UI_TARGET_FACTION_ID	faction_id_index
	faction_id_data_type/ SDC_Object_Type

Tabela 44. Odwzorowanie atrybutów struktury G_WEAPONLOAD w kolumny tabeli weaponload_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
G_WEAPONLOAD_FLAG_SAID	object_index
	object_data_type/SDC_Object_Type
G_WEAPONLOAD_FLAG_OWNER	owner_index
	owner_data_type/SDC_Object_Type
G_WEAPONLOAD_FLAG_TYPE	type/Weapon_Types
G_WEAPONLOAD_FLAG_INDEX	weapon_index
G_WEAPONLOAD_FLAG_LOADED	loaded
G_WEAPONLOAD_FLAG_EXPENDED	expended
G_WEAPONLOAD_FLAG_REMAINING	remaining
G_WEAPONLOAD_FLAG_NAME	name

Tabela 45. Odwzorowanie atrybutów struktury G_WEATHER w kolumny tabeli weather_data

Atrybut struktury GENIS	Nazwa kolumny SDC/ Translacja opisu
	side_view
G_WEATHER_DAY_TYPE	day_type/Day_Night
G_WEATHER_WEATHER	weather/WC_Symbol

Spis rysunków

Rysunek 1. Planowanie eksperymentu symulacyjnego w uwarunkowaniach czasowych CAX 11
Rysunek 2. Fazy występujące w każdym eksperymencie symulacyjnym 17
Rysunek 3. Fazy eksperymentu symulacji modelującego procesy walki na potrzeby ćwiczeń CAX .. 19
Rysunek 4. Iteracyjność w eksperymentach cząstkowych 20
Rysunek 5. Weryfikacja scenariusza 35
Rysunek 6. Lokalizacja pliku z błędami i ostrzeżeniami 35
Rysunek 7. Nagłówek raportu błędów i ostrzeżeń 36
Rysunek 8. Procedura testowania scenariusza 37
Rysunek 9. Dostęp do grup rozkazów dla każdej ze stron 38
Rysunek 10. Wysyłanie grup rozkazów 38
Rysunek 11. Procedura testowania scenariusza uwzględniająca wysyłanie grup rozkazów 39
Rysunek 12. Program do konfiguracji programów interfejsowych 40
Rysunek 13. Konfiguracja programu symulacyjnego 40
Rysunek 14. Konfiguracja programu do dystrybucji danych 41
Rysunek 15. Konfiguracja głównego programu graficznego GIAC 41
Rysunek 16. Procedura realizacji prostego eksperymentu symulacyjnego 45
Rysunek 17. Procedura korekty parametrów w prostym eksperymencie symulacyjnym 46
Rysunek 18. Procedura realizacji złożonego eksperymentu symulacyjnego 48
Rysunek 19. Procedura korekty parametrów w prostym eksperymencie symulacyjnym 49
Rysunek 20. Procedura korekty parametrów w programie DDS 50
Rysunek 21. GIAC - menu "Options/Database" – wybór bazy danych 57
Rysunek 22. Przykładowa formatka PPS z listą załadowanych punktów kontrolnych 61
Rysunek 23. Przykładowa formatka PPS z listą jednostek 62
Rysunek 24. Menu raportów PPS 62
Rysunek 25. Parametry raportu o historii lotów bojowych 63
Rysunek 26. Parametry raportu o zestrzeleniach samolotów 63
Rysunek 27. Przepływ danych między SDR i JTLS 65
Rysunek 28. Okno główne aplikacji zobrazowania wyników w trakcie eksperymentu 75
Rysunek 29. Okno opisu stanu jednostek 76
Rysunek 30. Okno stanu UiSW jednostek 76
Rysunek 31. Okno programu sterującego wraz z menu aplikacji 78
Rysunek 32. Przykładowe okno charakterystyk 78
Rysunek 33. Przykładowe okno wykresu charakterystyki 79

Spis tabel

TABELA 1. ZADANIA I ODPOWIEDZIALNOŚCI ZESPOŁÓW W PLANOWANIU EKSPERYMENTU SYMULACYJNEGO.....22
TABELA 2. STRUKTURA TABELI GENERAL_MESSAGE.....67
TABELA 3. KOMUNIKATY PRZETWARZANE PRZEZ MPC.....68
TABELA 4. ROZSZERZENIE STRUKTUR TABEL MPP NA POTRZEBY MPC.....69
TABELA 5. ODWZOROWANIE STRUKTUR GENIS W TABELE DANYCH SDC.....83
TABELA 6. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_AC_MAINT W KOLUMNY TABELI AC_MAINT_DATA..84
TABELA 7. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_AIRCRAFT W KOLUMNY TABELI AIRCRAFT_DATA....84
TABELA 8. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_ARC W KOLUMNY TABELI ARC_DATA.....85
TABELA 9. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_BACKORDER W KOLUMNY TABELI BACKORDER_DATA.....86
TABELA 10. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_BASE W KOLUMNY TABELI BASE_DATA.....86
TABELA 11. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_CAS_REQUEST W KOLUMNY TABELI CAS_REQUEST_DATA.....87
TABELA 12. ATTRIBUTE/COLUMN NAMES FOR G_CONVOY W KOLUMNY TABELI CONVOY_DATA.....88
TABELA 13. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_CRUISE W KOLUMNY TABELI CRUISE_DATA.....88
TABELA 14. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_DEPOT W KOLUMNY TABELI DEPOT_DATA.....89
TABELA 15. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_DMPI W KOLUMNY TABELI DMPI_DATA.....90
TABELA 16. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_DMPI_TARGET_LINK W KOLUMNY TABELI DMPI_TARGET_LINK_DATA.....90
TABELA 17. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_EQUIPMENT W KOLUMNY TABELI EQUIPMENT_DATA.....91
TABELA 18. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_FACTION W KOLUMNY TABELI FACTION_DATA.....91
TABELA 19. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_FARP W KOLUMNY TABELI FARP_DATA.....91
TABELA 20. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_FIRE_MISSION W KOLUMNY TABELI FIRE_MISSION_DATA.....92
TABELA 21. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_FORMATION W KOLUMNY TABELI FORMATION_DATA.....93
TABELA 22. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_GRAPHIC W KOLUMNY TABELI GRAPHIC_DATA.....93
TABELA 23. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_GROUND W KOLUMNY TABELI GROUND_DATA.....94
TABELA 24. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_GROUND_DAMAGE W KOLUMNY TABELI GROUND_DAMAGE_DATA.....95
TABELA 25. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_HEX W KOLUMNY TABELI HEX_DATA.....95
TABELA 26. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_HRU W KOLUMNY TABELI HRU_DATA.....96
TABELA 27. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_IADS W KOLUMNY TABELI IADS_DATA.....97
TABELA 28. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_MODEL_STATUS W KOLUMNY TABELI MODEL_STATUS_DATA.....97
TABELA 29. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_NODE W KOLUMNY TABELI NODE_DATA.....97
TABELA 30. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_PENDING_ORDER W KOLUMNY TABELI PENDING_ORDER_DATA.....97
TABELA 31. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_PIPELINE W KOLUMNY TABELI PIPELINE_DATA.....98
TABELA 32. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_PORT_EVENT W KOLUMNY TABELI PORT_EVENT_DATA.....98
TABELA 33. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_RAILROAD W KOLUMNY TABELI RAILROAD_DATA 98
TABELA 34. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_ROE W KOLUMNY TABELI ROE_DATA.....99
TABELA 35. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_RUNWAY_CUT W KOLUMNY TABELI RUNWAY_CUT_DATA.....99
TABELA 36. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_SHIP W KOLUMNY TABELI SHIP_DATA.....99
TABELA 37. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_SIDE_RELATION W KOLUMNY TABELI SIDE_RELATION_DATA.....101
TABELA 38. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_SQUADRON W KOLUMNY TABELI SQUADRON_DATA.....101
TABELA 39. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_SUPPLY_CATEGORY W KOLUMNY TABELI SUPPLY_CATEGORY_DATA.....102
TABELA 40. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_TARGET W KOLUMNY TABELI TARGET_DATA.....103
TABELA 41. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_TRACK W KOLUMNY TABELI TRACK_DATA.....104
TABELA 42. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_UI_GROUND W KOLUMNY TABELI UI_GROUND_DATA.....105

TABELA 43. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_UI_TARGET W KOLUMNY TABELI UI_TARGET_DATA 105
TABELA 44. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_WEAPONLOAD W KOLUMNY TABELI WEAPONLOAD_DATA 105
TABELA 45. ODWZOROWANIE ATRYBUTÓW STRUKTURY G_WEATHER W KOLUMNY TABELI WEATHER_DATA 106

BIBLIOTEKA GŁÓWNA - ARCHIWUM
Nr ewid. 59579
II
Akademii Obrony Narodowej

S/6427

10,90