

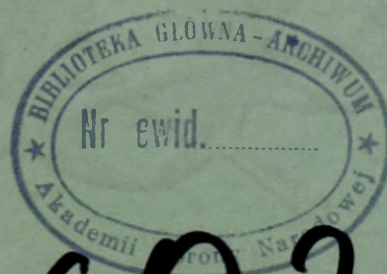


AKADEMIA  
OBRONY  
NARODOWEJ

AON 5582/2003

Krzysztof KALIŃSKI  
Grzegorz PIETRZAK  
Paweł BORYN

SYMULACYJNY SYSTEM  
WALKI JTLS



56824

WARSZAWA

2003

AKADEMIA OBRONY NARODOWEJ

---

CENTRUM SYMULACJI I KOMPUTEROWYCH GIER  
WOJENNYCH

AON 5582/03



Krzysztof KALIŃSKI  
Grzegorz PIETRZAK  
Paweł BORYN

## Symulacyjny system walki JTLS

Autorzy:

Pracę wykonano pod kierownictwem merytorycznym  
ppłk. dr inż. Janusza WOCIAŁA

mjr dr inż. Kaliński: rozdz. 1, 2, 3.1, 3.5, 4.1

mjr mgr inż. Pietrzak: rozdz. 3.3, 3.6, 4.3, 4.5, 4.6, 5

kpt. mgr inż. Boryn: rozdz. 3.4

współautorstwo mjr. dr. inż. Kalińskiego i mjr. mgr.

inż. Pietrzaka: rozdz. 3.2, 4.2, 4.4, 4.7

## WSTĘP

Skrypt poświęcony jest symulacyjnemu systemowi walki Joint Theater Level Simulation<sup>1</sup> (JTLS) rozwijanemu od lat przez firmę: ROLANDS & ASSOCIATES Corporation<sup>2</sup>.

Opracowanie zawiera aspekty zarówno technologiczne: wymagania sprzętowo-programowe, opis modułów programowych, główne przepływy i przetwarzanie informacji w systemie, definicje bazy danych systemu jak i merytoryczne: modele formalne ze szczególnym uwzględnieniem modelu walki Lanchestera, reprezentację terenu, systemy walki.

Pracę ujęto w pięciu rozdziałach. Pierwszy rozdział zawiera ogólne właściwości i uwarunkowania systemu. Drugi opisuje pokrótce cztery warianty wykorzystania systemu JTLS w czasie ćwiczeń. W rozdziale trzecim znalazły się wybrane zagadnienia cząstkowe związane z systemem. Czwarty rozdział poświęcony jest przygotowaniu systemu JTLS do działania oraz niektórym zagadnieniom związanym z jego użytkowaniem w trakcie ćwiczeń. W piątym rozdziale opisano pokrótce dodatkowe narzędzia dla faz przygotowania i omówienia ćwiczeń zaimplementowane przez inne firmy niż ROLANDS. Wykorzystano w nim materiały pozyskane od por. Janusza Maziarza z Zespołu Informatyki 2KZ.

W przyszłości system JTLS ma być podstawowym systemem symulacyjnym wspierającym ćwiczenia typu CAX (ćwiczenia z wykorzystaniem komputerowych systemów symulacyjnych działania wojsk) przeprowadzane w Centrum Symulacji i Komputerowych Gier Wojennych (CSiKGW) AON. Ponieważ CSiKGW nie posiada jeszcze instalacji JTLS poniższe opracowanie powstało przede wszystkim na bazie dostępnej dla autorów, dokumentacji technicznej. Podjęto się tego trudu, aby przybliżyć możliwości i ograniczenia symulacyjnego systemu walki JTLS.

Opracowanie wykonali oficerowie z Zakładu Symulacji CSiKGW.

---

<sup>1</sup> Najczęściej tłumaczone jako: System Symulacyjny Działań Połączonych

<sup>2</sup> Strona internetowa firmy: [www.rolands.com](http://www.rolands.com)

# 1. OGÓLNA CHARAKTERYSTYKA SYSTEMU JTLS

System Symulacyjny Działań Połączonych (JTLS) jest interaktywnym systemem komputerowym umożliwiającym prowadzenie wielostronnej gry wojennej. Modeluje działania sił: lądowych, morskich, powietrznych, a także sił specjalnych i wywiadu w układzie koalicyjnym. JTLS pierwotnie został zaprojektowany jako narzędzie do użycia w rozwoju i analizie połączonych i koalicyjnych planów operacyjnych. Często jest używany jako model do szkolenia dowództw i sztabów.

System został tak zaprojektowany aby mógł być użyty bez modyfikacji jako:

1. Narzędzie analityczne w procesie planowania.
2. System wspomagający proces edukacyjny.
3. System wspomagania ćwiczeń i szkoleń.
4. Podstawowe narzędzie do badania rezultatów walki.

Podstawowe cechy systemu to:

- szczebel strategiczno – operacyjny;
- szerokie spektrum symulacji konfliktów;
- możliwość symulacji działań przed wybuchem konfliktu;
- możliwość symulacji operacji połączonych;
- możliwość prowadzenia ćwiczeń wielostronnych z udziałem stron koalicji;
- możliwość zobrazowania terenu na dużych obszarach (2000 x 2000 mil morskich).

W JTLS główna uwaga skupiona jest na konwencjonalnych operacjach połączonych i koalicyjnych na poziomie strategiczno - operacyjnym zdefiniowanym przez Połączone Kolegium Zadaniowe. JTLS odzwierciedla działania w wymiarze powietrznym, lądowym, morskim, desantowym i sił specjalnych.

## 1. Operacje Lądowe:

1.1. Szczebel od brygady wzwyż;

1.2. Odzworowanie: Przemieszczenia wojsk, Manewru, Artylerii, Wojsk Inżynieryjnych i Lotnictwa Wojsk Lądowych.

## 2. Operacje Powietrzne:

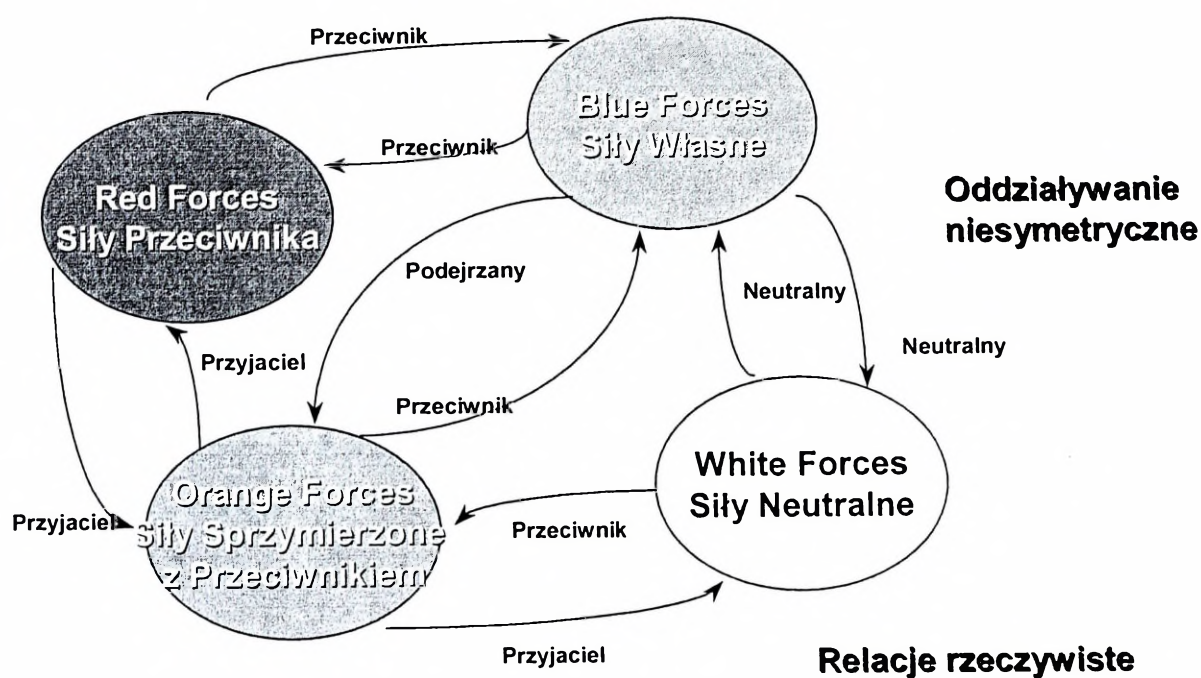
2.1. Możliwość prowadzenia pojedynczych misji lub jako część większych operacji;

2.2. Odzworowanie: Ataku powietrznego, Walki powietrznej, Transportu, Zrzutu, Tankowania w powietrzu, Osłona, Rozpoznania, Rozpoznania satelitarnego, AWACS, Patrolowania, Obrony powietrznej, WRE, Poszukiwania i Ratownictwa (SAR).

3. Operacje morskie:
  - 3.1. Możliwość prowadzenia przez okręty nawodne i podwodne pojedynczych działań lub jako część większej formacji;
  - 3.2. Odwzorowanie: Transportu morskiego, Ataku podwodnego, Walki z okrętami podwodnymi, Wsparcia artylerii okrętowej;
  - 3.3. Działania desantowe.
4. Logistyka:
  - 4.1. Uzupełnianie zapasów jednostek;
  - 4.2. Konwoje z zaopatrzeniem;
  - 4.3. Transport zaopatrzenia przy pomocy barek rzecznych i linii kolejowych;
  - 4.4. Naprawa i obsługa sprzętu i uzbrojenia.
5. Rozpoznanie:
  - 5.1. Możliwość dokładnego oceny sytuacji przez każdą ze stron konfliktu;
  - 5.2. Wykorzystanie rozpoznania strategicznego i operacyjnego.
6. Działania sił specjalnych:
  - 6.1. Zbieranie informacji rozpoznawczych;
  - 6.2. Działania bezpośrednie.
7. Inne formy działań:
  - 7.1. Odwzorowanie oddziaływania: Systemów łączności i informatyki, Taktycznych pocisków balistycznych, relacji cywilno – wojskowych.

JTLS przedstawia wielostronne koalicje sił lądowych, morskich, lotniczych i sił specjalnych. Model może przedstawiać ponadto, w obszarze zainteresowania, siły cywilne i niewojskowe. W systemie może być do 10 stron konfliktu, a każda ze stron może być później podzielona na dowolną liczbę frakcji. Lojalność frakcji zmienia się dynamicznie w czasie symulacji, a relacje pomiędzy stronami nie muszą być symetryczne, a także mogą się zmieniać w czasie symulacji (patrz rysunek 1). Fakt, że strona B jest Przyjacielem strony A nic nie mówi o relacji przeciwnej. Strona A może uważać stronę B za Przyjaciela, Neutralną, Podejrzaną lub nawet Wroga.

Nazwy stron, frakcji i kolor użyty do wyświetlenia sił strony są zdefiniowane w inicjującej bazie danych.



Rysunek 1. Relacje pomiędzy stronami w JTLS.  
/źródło: prezentacja z konferencji z dnia 22.08.2003/

Możliwe relacje pomiędzy stronami to: Przyjaciel, Neutralny, Podejrzany i Wróg. Na początku symulacji, relacje pomiędzy stronami określają inicjujące bazy danych. W trakcie symulacji relacje te mogą być zmienione jedynie poprzez:

- wysłanie rozkazu *CHANGE SIDE RELATIONSHIP* (który określa stronę i nowe relacje);
- wysłanie jednostki do zaatakowania innej jednostki. Jeżeli jednostka lądowa zostanie wysłana do ataku na obcą jednostkę, relacje automatycznie zostaną ustawione na Wróg dla wszystkich jednostek atakowanej strony.

## 2. WARIANTY UŻYCIA SYSTEMU JTLS

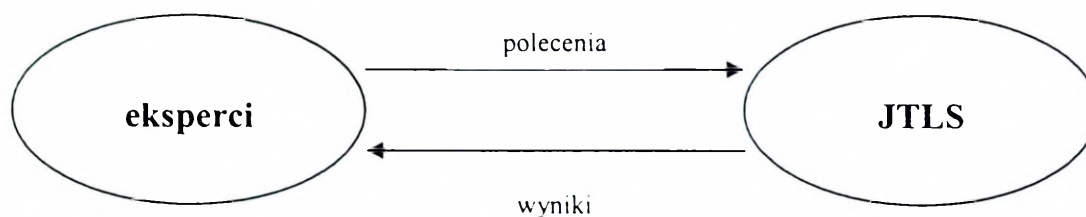
Wariant i użycie procesów symulacji systemu JTLS jest ściśle związane z celem i zasięgiem symulacji. Merytoryczne cele ćwiczenia lub cele analityczno-badawcze wymuszają wybór wariantu użycia modelu. Zwykle używa się jednego z czterech głównych wariantów:

1. Seminarium lub wariant analityczny (*Seminar wargame or analytical design*);
2. Wariant otwartego (pełnego) wsparcia (*Open support design*);
3. Wariant ukrytego (ograniczonego) wsparcia (*Hidden support design*);
4. Wariant rozdzielonych danych wyjściowych (*Distributed output design*).

Ad.1. **Wariant seminarium gry wojennej** wymaga małej liczby ćwiczących do operowania modelem i raportowania rezultatów do kluczowych decydentów.

- Doświadczona komórka wprowadza rozkazy aby określić część planu kampanii. Następnie symulacja toczy się na najwyższej szybkości w trybie przetwarzania (*batch mode*), dopóki nie zostanie osiągnięty wcześniej określony punkt lub nie upłynie określony czas.
- Ćwicząca komórka dostarcza rezultaty operacji do kierownika ćwiczenia, aby umożliwić mu podjęcie stosownej decyzji.
- Kiedy kluczowe decyzje zapadną, cykl jest kontynuowany z nowymi rozkazami wprowadzonymi do modelu.

Ten wariant użycia modelu wymaga tylko kilku graczy, ale muszą oni być dobrze wyszkoleni we wszystkich aspektach modelu oraz w wielu obszarach prowadzenia działań zbrojnych.



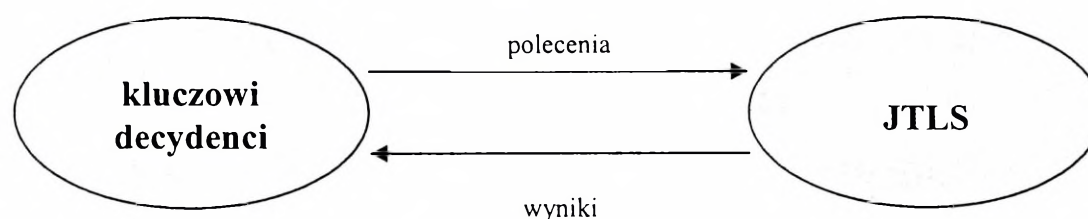
**Rysunek 2. Wariant seminarium.**

/źródło: opracowanie własne/

Ad.2. **Wariant otwartego (pełnego) wsparcia** dopuszcza kluczowych ćwiczących członków-decydentów do funkcji symulacyjnych. Decydenci ci i/lub ich reprezentanci mogą wprowadzać rozkazy bezpośrednio do symulacji i otrzymywać dane z modelu. W tym wariantcie model jest całkowicie otwarty (dostępny) dla ćwiczących.

**Zaletą** tego projektu jest nie wymaganie licznego personelu do wprowadzania danych wejściowych.

**Wady.** Ćwiczący mogą być zbyt zaangażowani w szczegóły symulacji w porównaniu z celami treningu i ich procesem decyzyjnym. Dodatkowo uproszczenia dokonane w procesie modelowania lub błędy wprowadzone przez graczy mogą zmniejszyć pożądany poziom realizmu.



**Rysunek 3. Wariant otwartego wsparcia.**

/źródło: opracowanie własne/

Ad. 3. **Wariant ukrytego (ograniczonego) wsparcia** jest próbą ominięcia ograniczenia realizmu poprzez umieszczenie reagującej komórki pomiędzy systemem symulacji, a ćwiczącymi.

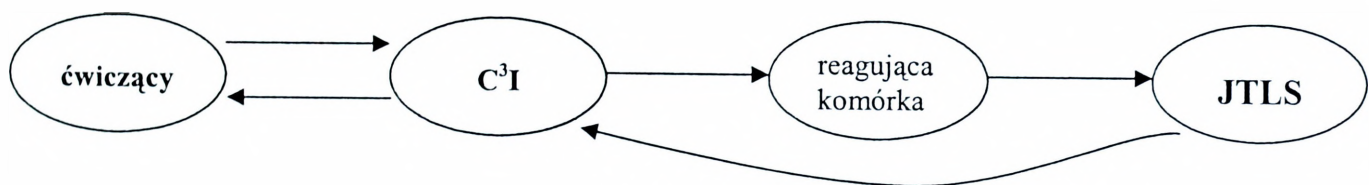
- Ćwiczący wydają rozkazy operacji, w ich standardowej postaci, ignorując ograniczenia modelu lub jego specyficzne wymagania.
- Odpowiednia komórka reagująca przetwarza te rozkazy operacji na wytyczne symulacji.
- W czasie symulacji reagująca komórka monitoruje raporty tworzone przez model i może monitorować wyniki symulacji ponad organicznymi systemami dowodzenia i kierowania.
- Ta komórka przegląda wszystkie wyniki, aby upewnić się, że błędy są poprawione zanim dane są przesłane do ćwiczących.

W ten sposób system symulacji jest ukryty przed ćwiczącymi. Wariant ten wymaga personelu do operowania stacjami roboczymi JTLS i dodatkowego personelu do utworzenia i utrzymywania odpowiednich, reagujących komórek.



**Rysunek 4. Wariant ukrytego wsparcia.**  
 /źródło: opracowanie własne/

Ad. 4. **Wariant rozdzielonych danych wyjściowych** próbuje połączyć aspekty wariantu otwartego i ukrytego wsparcia z oddziaływaniem modelu z wybranym systemem C<sup>3</sup>I. Ten wariant zezwala kluczowym decydentom przesyłać rozkazy i monitorować wyniki przez ich organiczne systemy dowodzenia i kierowania. Ich rozkazy są przetwarzane w dyrektywy symulacji poprzez reagujące komórki (tak jak w wariacie ukrytego wsparcia). Jednakże dane wyjściowe z modelu przesyłane są bezpośrednio do ćwiczących poprzez ich systemy C<sup>3</sup>I.



**Rysunek 5. Wariant rozdzielonych danych wyjściowych.**  
 /źródło: opracowanie własne/

### **3. SYSTEM JTLS**

#### **3.1. Podstawowa konfiguracja sprzętowa i jej warianty**

##### **Platforma programowa**

1. System operacyjny zgodny ze standardem POSIX:
  - Sun Sparc/Ultra stations;
  - Hewlett-Packard workstations;
  - Systemy UNIX.
2. X-Windows.
3. Biblioteki Motif Toolkit.
4. Dla kodów źródłowych wymagany jest translator SIMSCRIPT II.5 do C na docelową platformę i kompilator C.
5. Do tworzenia baz danych wymagany jest ORACLE.

##### **Platforma sprzętowa**

Poniższy sprzęt został przetestowany w trakcie testu<sup>3</sup> w Joint Warfighting Center. Testowano bazę danych składającą się z około 1500 jednostek, 5000 obiektów, 8 stron i 20 frakcji. W teście przyspieszenie czasu symulacji wynosiło 4 do 1.

- 1) Combat Events Program (CEP) - wymagany - 1
  - a) Ultra 60;
  - b) Od 512 Mbytes do 1 Gbytes RAM;
  - c) 9.1 Gbytes pamięci dyskowej;
  - d) 4 mm streamer;
  - e) port Ethernet;
  - f) 21-calowy monitor kolorowy.
- 2) G-Data Sever (GENIS) i Controller Workstation - wymagany - 1
  - a) Ultra 60;
  - b) Od 512 Mbytes do 1 Gbytes RAM;
  - c) 9.1 Gbytes pamięci dyskowej;
  - d) port Ethernet;
  - e) 21- calowy monitor kolorowy.
- 3) Map Server - opcjonalny- 1 wymagany gdy są wyświetlane mapy cyfrowe
  - a) Ultra 60;
  - b) 128 Mbytes RAM;

---

<sup>3</sup> JTLS Ececutive Overview v.2.5.

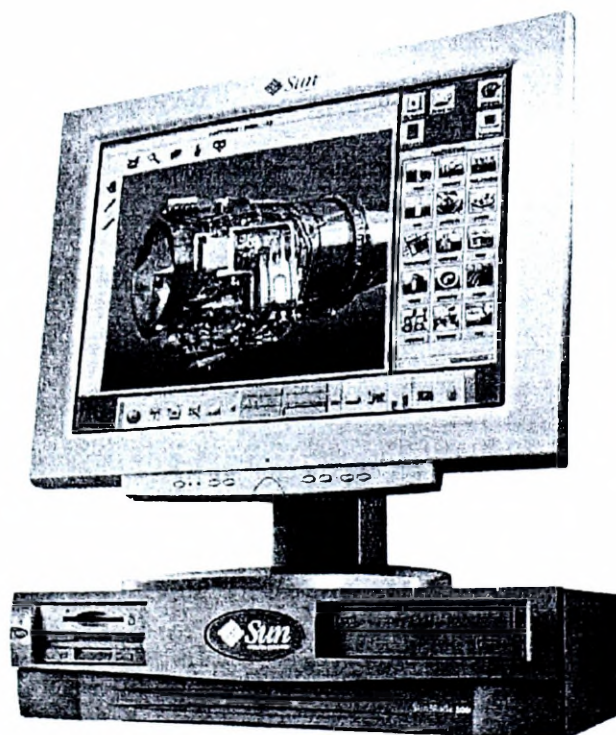
- c) 9.1 Gbytes pamięci dyskowej (około 110 Mbytes wymagany dla mapy)
  - d) CD ROM;
  - e) port Ethernet;
  - f) 21- calowy monitor kolorowy.
- 4) Player Workstation - wymagany - minimum 1 na stronę
- a) Ultra 10;
  - b) 128 Mbytes RAM;
  - c) 9.1 Gbytes pamięci dyskowej;
  - d) port Ethernet;
  - e) 21- calowy monitor kolorowy;
  - f) trzy klawiszowa mysz dla nowego IMT.
- 5) Laser Printer - opcjonalny - minimum 1. Zalecana jedna dla obszaru roboczego.
- 6) Miscellaneous – kable Ethernet i połączenia.

Mniejsze bazy danych, takie jakie mogą być użyte do treningów, testowania, eksperymentów i demonstracji mogą działać na słabszej platformie. Np. oprogramowanie CEP i GENIS może być uruchomione jednocześnie na jednym komputerze Ultra 60.

Wymagania na stację roboczą spełniają posiadane w CSiKGW komputery Sun Blade 100 i Sun Blade 1000.

#### Sun Blade 100:

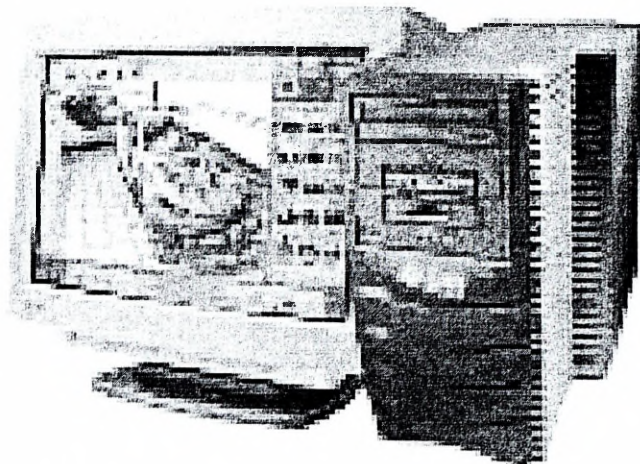
- obudowa klasy Desktop;
- ATA66 15-Gbyte HDD;
- CD-ROM;
- 1.44-MB FDD;
- czytnik kart Smart card;
- procesor 500-MHz UltraSPARC-II;
- pamięć RAM – 128 MB;
- 3 gniazda PCI;
- 1 x serial port;
- 1 x parallel port;
- karta grafiki ATI Rage XL on-board 8 MB SGRAM;
- karta sieciowa 10-megabit/100-megabit;
- 2 x port IEEE 1394;



- 4 x USB ( w tym mysz i klawiatura).

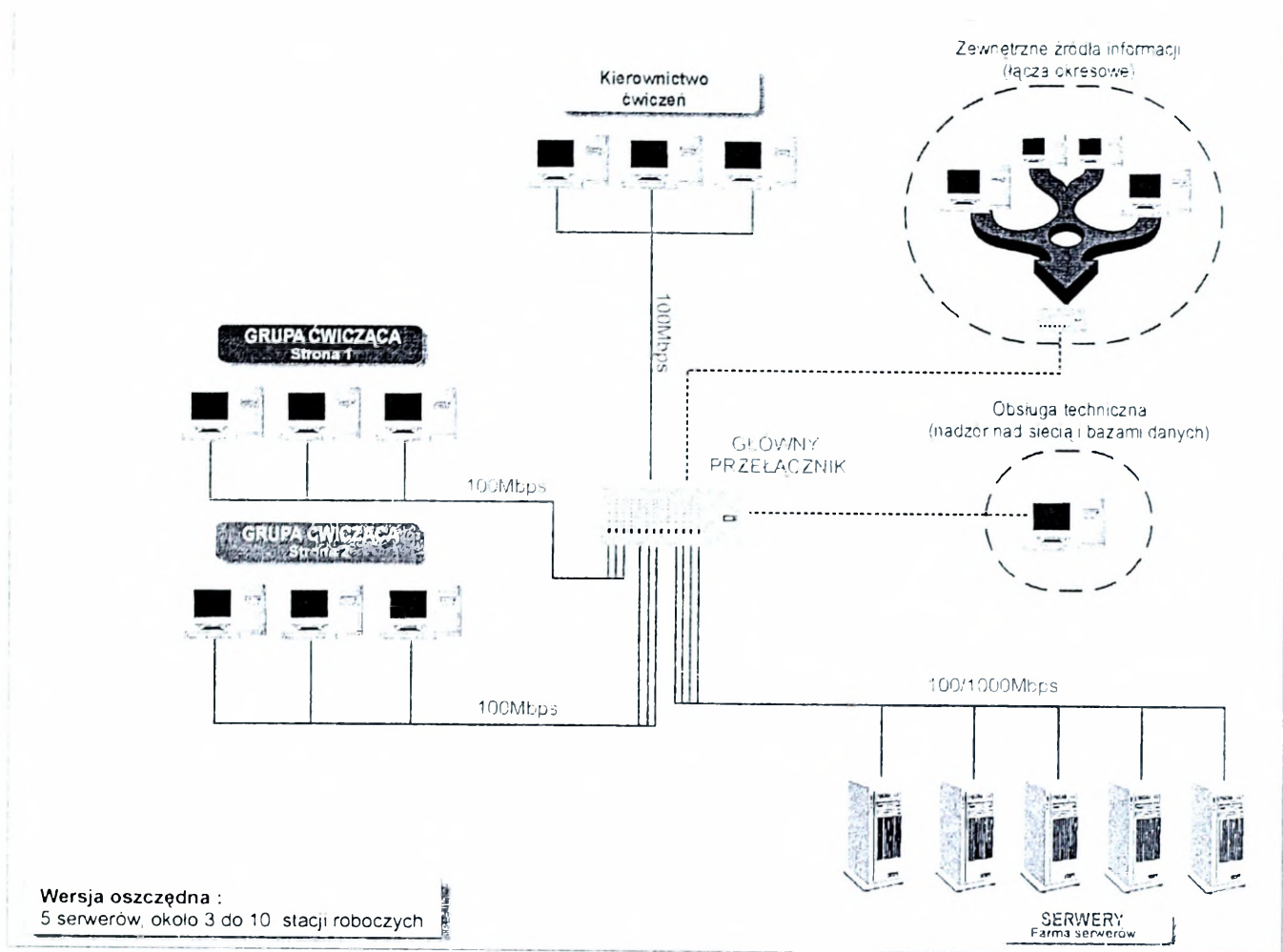
#### Sun Blade 1000:

- 2 x Processor UltraSPARC-III 750 MHz;
- pamięć 1GB;
- 4 x 64-bit złącza (PCI);
- napęd FDD 3.5 cala;
- czytnik Smart card;
- napęd DVD-ROM ;
- streamer;
- karta sieciowa Ethernet 10BASE-T/100BASE-T;
- dysk HDD 36 BB Ultra wide SCSI;
- 4 x złącze universal serial bus (USB);
- 2 x IEEE 139;
- 1 x port równoległy;
- 2 x port szeregowy RS-232C/RS423.



Elementy struktury techniczno-organizacyjnej CSiKGW można podzielić na:

1. Stanowiska pracy – dla kierownictwa ćwiczenia i graczy podzielonych na 2 do 10 stron ćwiczących
2. Serwery – minimum jeden, zalecane 3 do 4
  - *Combat Events Program*, główny serwer ze scenariuszami działań
  - G-Data Server (GENIS) i *Controller Workstation*, baza danych symulacji
  - Map Server, dla map cyfrowych
  - Serwery pomocnicze: wydruku, DHCP, DNS, Poczta el.
3. Przełączniki 3 warstwy CISCO WS-C6509 i WS-6506
4. Sieć światłowodową (*Multi Mode Fiber Optic*)

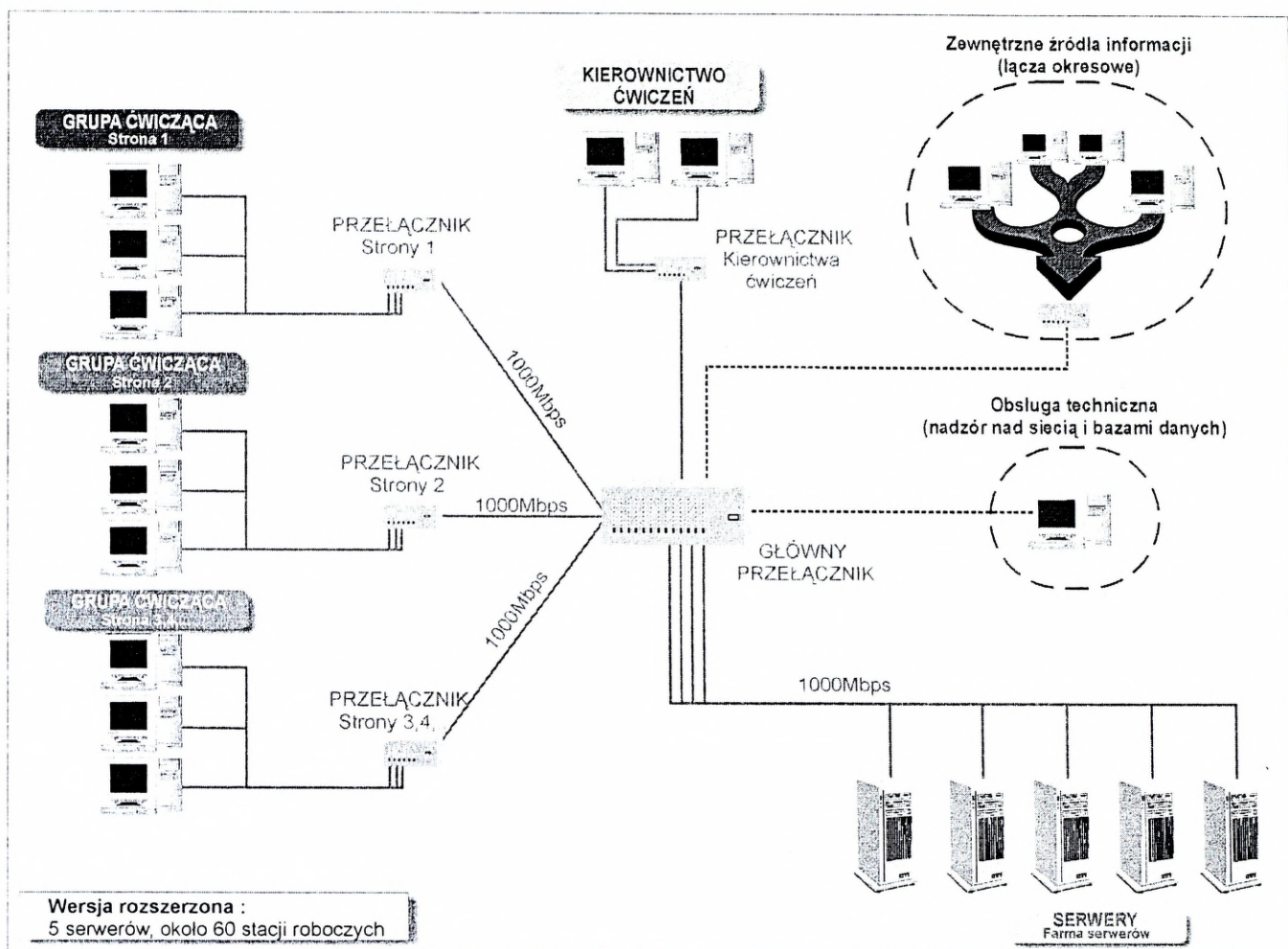


**Rysunek 6, Struktura techniczno-organizacyjna w wersji oszczędnej.**

/źródło: wystąpienie J. Kucharskiego na konferencji z dnia 22.08.2003/

Rysunek 6 przedstawia strukturę techniczno-organizacyjną w wersji oszczędnej składającej się z 2 stron i kierownictwa. Stanowiska pracy poszczególnych stron łączone są głównym przełącznikiem. Niezbędne serwery zgrupowane zostały w farmę serwerów. Stacji roboczych w wersji minimalnej jest trzy, po jednym dla każdej ze stron i dla kierownictwa ćwiczenia. Zewnętrznymi źródłami informacji są jednostki współpracujące – wydzielone, zewnętrzne stanowiska.

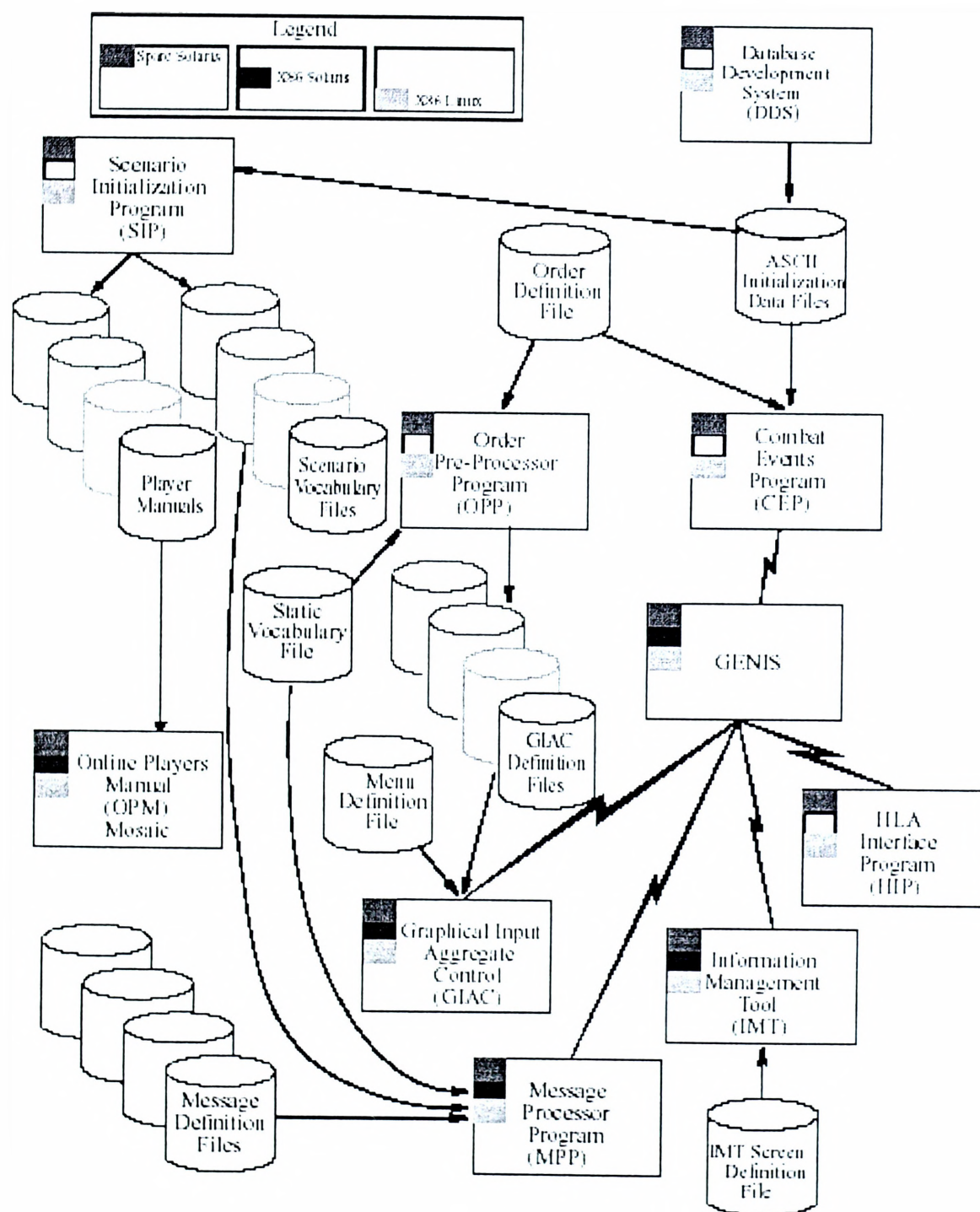
Rysunek 7 przedstawia strukturę techniczno-organizacyjną w wersji rozszerzonej składającej się z 6 stron i kierownictwa. Do głównego przełącznika podłączone są mniejsze przełączniki, co daje możliwość podłączenia nawet 400 stacji roboczych.



**Rysunek 7. Struktura techniczno-organizacyjna w wersji rozszerzonej.**  
 /źródło: wystąpienie J. Kucharskiego na konferencji z dnia 22.08.2003/

### 3.2. Logiczno programowa struktura systemu

System JTLS składa się z 6 głównych programów i licznych, mniejszych programów wspomagających (rysunek 8). Programy mogą być zarówno programami całkowicie autonomicznymi: posiadać własne interfejsy użytkownika oraz własne zbiory konfiguracyjne, jak i samodzielny bibliotekami DLL umożliwiającymi wykonanie określonej funkcji (bez korzystania z interfejsu użytkownika) – przykładowo program weryfikacji rozkazu.



Rysunek 8. Struktura programowa JTLS.  
/źródło: JTLS Executive Overview v.2.5/

Ze względu na funkcje oraz czas wykorzystania programów w całym cyklu pracy systemu JTLS można dokonać podziału programów na niżej wymienione grupy:

1. Programy tworzenia scenariusza:

- system tworzenia i modyfikacji bazy danych (*Database Development System – DDS*);
- program weryfikacji scenariusza (*Scenario Verification Program – SVP*);
- program modyfikacji terenu (*Terrain Modification Utility – TMU*);
- program modyfikacji współczynników równań Lanchester'a (*Lanchester Development Tool - LDT*);
- program obsługi graficznej bazy danych (*Graphical Database Program - GDP*);
- program konfiguracji trybu pracy systemu (*High Level Architecture Interface Program*);
- program weryfikacji rozkazu (*Order Verification Tool - OVT*).

2. Programy inicjujące oraz konfiguracyjne:

- program inicjujący scenariusz (*Scenario Initialization Program – SIP*);
- program weryfikacji scenariusza (*Scenario Verification Program – SVP*);
- program konfiguracji interfejsu (*Interface Configuration Program – ICP*);
- program logowania (*Interface Login Program – ICP Login*).

3. Program odwzorowania działań (*Combat Events Program - CEP*).

4. Podstawowe programy użytkowe:

- program sterujący graficznymi danymi wejściowymi (*Graphical Input Aggregate Control – GIAC*);
- narzędzia zarządzania informacją (*Information Management Tool – IMT*);
- podręcznik gracza (*Online Player's Manual – OPM*);
- program przetwarzania wiadomości (*Message Processor Program – MPP*);
- generator rozkazów zadań lotniczych (*Air Tasking Order Generator – ATOG*);
- tłumacz rozkazów zadań lotniczych (*Air Tasking Order Translator – ATOT*).

### **3.2.1. Programy tworzenia scenariusza**

Narzędzia tworzenia scenariusza i wspomagające są używane do:

- dostępu i czytania plików danych inicjujących ASCII;
- dostarczania informacji o zgodności i poprawności;
- działania na danych i zapisywanie nowych, uaktualnionych plików danych ASCII.

Dopóki pliki ASCII są wykorzystywane do przygotowania ćwiczeń, dobrze wyszkolony personel może mieć dostęp do plików inicjujących w jakimkolwiek edytorze.

Dostęp do plików danych, w takim przypadku, zalecany jest tylko dla doświadczonego personelu zapoznanego z formatem plików danych.

#### **System rozwoju bazy danych** (*Database Development System – DDS*)

DDS jest podstawowym narzędziem rozwoju i modyfikacji baz danych. Jest używany do stworzenia nowej bazy danych, modyfikacji istniejącej bazy danych, wyszukiwania informacji. Np. DDS może być użyty do uzyskania z bazy danych listy jednostek określonego typu. DDS jest aplikacją serwera ORACLE zbudowana przy użyciu narzędzi ORACLE Forms/Reports Developer 6i. Pliki danych ASCII określające inicjującą bazę danych dla wybranego scenariusza są wczytywane i wypełniają zbiór tabel ORACLE. Pliki te są tak zaprojektowane aby były kompatybilne ze strukturą baz danych JTLS. Użytkownik DDS uzyskuje dostęp do tych tabel aby modyfikować lub przeszukiwać zawarte w niej dane. Kiedy modyfikacja jest zakończona, dane są ładowane do stworzenia nowego zbioru plików inicjujących dla scenariusza. DDS jest uruchamiane poprzez wybór pierwszej opcji: „Tworzenie lub zmienianie baz danych scenariusza” z głównego menu JTLS. Z podmenu wybiera się : „Menu systemu rozwoju bazy danych”.

#### **Program weryfikacji scenariusza** (*Scenario Verification Program - SVP*)

SVP jest częścią rodziny narzędzi przedstawionych jako SIP (*Scenario Initialization Program*). Zadaniem SVP jest weryfikacja czy dane wprowadzone dla zadanego scenariusza są kompatybilne poprzez sprawdzenie konsekwencji pomiędzy zmianami. SIP wczytuje pliki bazy danych scenariusza, które są używane przez CEP (*The Combat Events Program*) i utrzymuje dane w takich samych strukturach używanych przez model w czasie wykonywania symulacji. To czyni wszystkie sprawdzenia zawartości wydajnymi i dokładnymi. To zapewnia np., że jednostki morskie są umieszczone w rastrach morskich, a jednostki lądowe na lądzie. Wynikami działania programu są błędy i braki konsekwencji, które zostały zauważone w danych. SVP jest uruchamiany poprzez wybór drugiej opcji: „Weryfikacja istniejącej bazy danych scenariusza” z głównego menu JTLS.

#### **Program modyfikacji terenu** (*Terrain Modification Utility – TMU*)

Ten program narzędziowy jest używany do modyfikacji plików terenu opartych na sześciokątnym rastrze. Program wyświetla wybraną przez użytkownika bazę danych terenu nałożoną na mapę wektorową. Użytkownik może zmienić skalę i powiększyć określony obszar na mapie oraz wyświetlić siatkę rastrów. Można wtedy przeglądać i modyfikować pojedyncze atrybuty rastra (typ terenu, typ przeszkody, wysokość).

### **Program modyfikacji współczynników równań Lanchester'a** (*Lanchester Development Tool - LDT*)

Ten program wspomaga rozwój równań Lanchester'a, które są używane do obliczenia rezultatów walki lądowej w JTLS. Użytkownik wprowadza spodziewany wynik bitwy o określonym czasie trwania, a LDT generuje proponowaną tabelę współczynników Lanchester'a (z dokumentacji nie można się jednak domyślić o jakie współczynniki chodzi autorom dokumentacji). LDT ma także funkcję weryfikacji, która pozwala użytkownikowi zobaczyć w postaci tabeli rezultaty długoterminowej bitwy pomiędzy dwoma lub więcej jednostkami.

### **Program obsługi graficznej bazy danych** (*Graphical Database Program – GDP*)

Ten program pozwala na graficzne rozmieszczenie jednostek i obiektów oraz na tworzenie lub modyfikację granic państw i obszarów operacji. Program wczytuje bieżące informacje scenariusza z bazy danych DDS i wyświetla je w GIAC (Graphical Input Aggregate Control).

### **Program konwersji położenia** (*Convert Location Program - XCONVERT*)

XCONVERT jest narzędziem akceptującym następujące koordynaty położenia:

- szerokość i długość geograficzną wyrażoną w postaci tekstowej i dziesiętnej;
- współrzędne rastra sześciokątnego;
- Uniwersalna Siatka Merkatora (UTM);
- Wojskowa siatka współrzędnych.

Kiedy zostanie wprowadzona poprawna lokalizacja program przekształca ją w pozostałe trzy systemy lokalizacji.

### **Program konfiguracji trybu pracy systemu** (*High Level Architecture Interface Program*)

Kiedy JTLS pracuje w trybie *High Level Architecture* (HLA), używany jest HIP (*HLA Interface Program*) do rozprowadzania danych symulacji poprzez *Run Time Infrastructure* (RTI) do innych składników HLA. Jeżeli jest uruchomiony tryb HLA z użyciem zarządzania czasem, jest używany dodatkowy program zwany krokomierzem do regulacji szybkości symulacji.

### **Program weryfikacji rozkazu** (*Order Verification Tool - OVT*)

OVT jest zaprojektowany do przyjmowania informacji dystrybuowanych przez GENIS z programu CEP (*Combat Events Program*). Używa ich do weryfikacji formatu i zawartości rozkazów graczy i kierownika ćwiczenia, które zostały zbudowane przy pomocy innych programów niż GIAC, przed wysłaniem ich do modelu. Wszystkie rozkazy po sprawdzeniu, że powinny być zrealizowane przez GIAC są także realizowane przez OVT.

OVT dostarcza graficznego interfejsu (GUI), który pozwala operatorowi zweryfikować istniejące pliki rozkazów GIAC oraz pozwala innym programom połączyć się i zweryfikować rozkazy poprzez odpowiedni protokół. Sprawdzanie rozkazów wspomagane jest przez funkcjonalną bibliotekę OVT (OVT-FL), która składa się z minimalnego podzbioru kodu niezbędnego do przeprowadzenia weryfikacji rozkazu. Ta biblioteka kodów obiektów jest także dostępna dla innych programów, które chcą przeprowadzić własną weryfikację rozkazu zamiast użyć OVT.

### 3.2.2. Programy inicjujące oraz konfiguracyjne

Programy te są używane do przygotowania JTLS do uruchomienia z określonym scenariuszem.

#### **Program inicjujący scenariusz** (*Scenario Initialization Program - SIP*)

SIP jest jednym z programów, który musi być uruchomiony do przygotowania scenariusza jaki nie był uruchamiany poprzednio w grze. Pozostałe funkcje programu SIP:

1. Ustala czy wszystkie pliki danych definiujących scenariusz istnieją;
2. Tworzy poprawną strukturę katalogów do przechowywania plików danych inicjujących grę;
3. Kopiuje wszystkie pliki danych inicjujących scenariusz do katalogu symulacji;
4. Tworzy pliki danych niezbędne przez inne procesy JTLS;
5. Weryfikuje wybór pliku powierzchni mapy do symulacji.

SIP musi zostać powtórnie uruchomiony dla scenariusza:

1. Jeżeli zmieniły się dane inicjujące;
2. Jeżeli zmieniły się definicje w słowniku statycznym;
3. Po instalacji nowej wersji GIAC;
4. Jeżeli zmienił się plik definicji rozkazu.

#### **Program konfiguracji interfejsu** (*Interface Configuration Program – ICP*)

ICP jest interaktywnym programem, który pozwala użytkownikowi zdefiniować specyfikację każdego procesu jaki może być uruchomiony dla szczególnego scenariusza: OPMs (*Online Player Manuals*), ATOG/Ts (*Air Tasking Order Generator and Translator Programs*), IMTs (*Information Management Tools*), MPPs (*Message Processor Programs*), GIACs, GENISes i CEP. Program używa graficznego interfejsu (GUI) by pozwolić użytkownikowi edytować domyślne (określone gdy ruszyło SIP) konfiguracje procesów. Dane wymagane do zdefiniowania procesów w konfiguracji symulacji są różne dla typów procesu. Definicja procesu zawiera nazwę, hasło, procesy nadrzędne, wykonawczego hosta,

stronę sił itd. ICP zwykle jest uruchamiany przed startem symulacji, aby dopasować konfigurację do potrzeb ćwiczenia. Jednakże może być także uruchomiony gdy JTLS działa.

#### **Program logowania** (*Interface Login Program - ICPLogin*)

Program zaprojektowany do ułatwienia procesów uruchamiania symulacji (np. GIAC). Wszystkie procesy symulacji normalnie są uruchamiane przy użyciu tego programu. Program ICPLogin jest uruchamiany przez wybór opcji w menu głównym JTLS. Dlatego też aby uruchomić proces użytkownik po prostu klika na właściwym klawiszu. Po wyborze procesu program ICPLogin czyta plik danych, który został utworzony przez ICP, tworzy wymagany plik konfiguracyjny i inicjuje start.

### **3.2.3. Program zdarzeń walki**

Program zdarzeń walki (*Combat Events Program – CEP*) jest głównym programem systemu JTLS, jest modelem walki. Określa wszystkie akcje i interakcje pomiędzy siłami lotniczymi, lądowymi i morskimi zdefiniowanymi i modelowanymi dla określonego, uruchamianego scenariusza. CEP tworzy, obsługuje i raportuje bieżący status modelowanego środowiska działań wojennych. CEP może modelować do 10 stron lub koalicji w każdym scenariuszu.

Każda strona może określić relacje (przyjacielski, wrogą, podejrzany, neutralny) z każdą inną przedstawioną stroną. Tylko jeden program może działać na każdym pojedynczym komputerze lub w sieci w czasie działania JTLS.

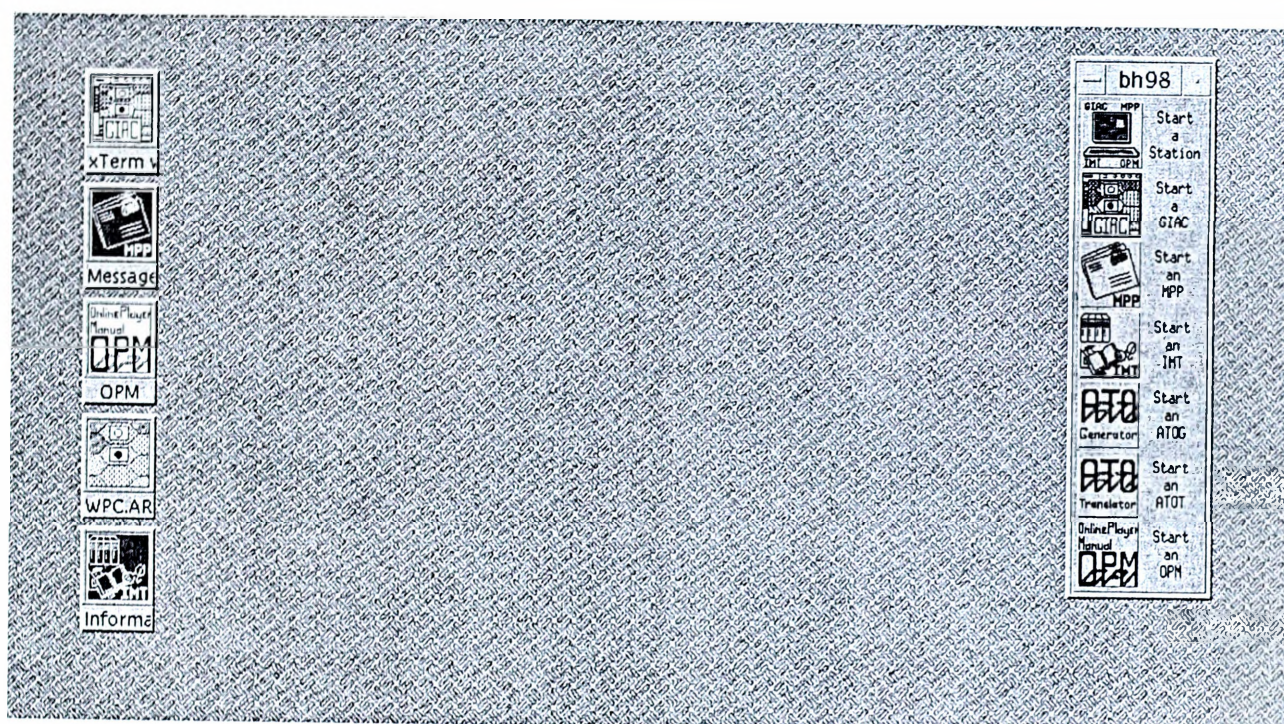
CEP komunikuje się poprzez protokół TCP/IP jedynie z pojedynczym procesem Primary GENIS. Primary GENIS otrzymuje inicjujące dane i okresowe uaktualnienia z CEP. Ten, na odwrót, komunikuje się z GIACs, MPPs, IMTs i Secondary GENISes, które są do niego przydzielone. Każdy GENIS obsługuje swoją własną bazę danych symulacji i komunikuje się z procesami bezpośrednio podległymi w strukturze drzewa informacji.

Dane wejściowe gracza do symulacji są wprowadzane w formularzu rozkazów do GIAC i przesyłane do CEP do przetworzenia przez jeden lub więcej GENIS. Gracze otrzymują informacje o grze z CEP w formie: uaktualnień graficznych GIAC, wiadomości MPP i uaktualnień tabelarycznych danych IMT.

### 3.2.4 Podstawowe programy użytkowe

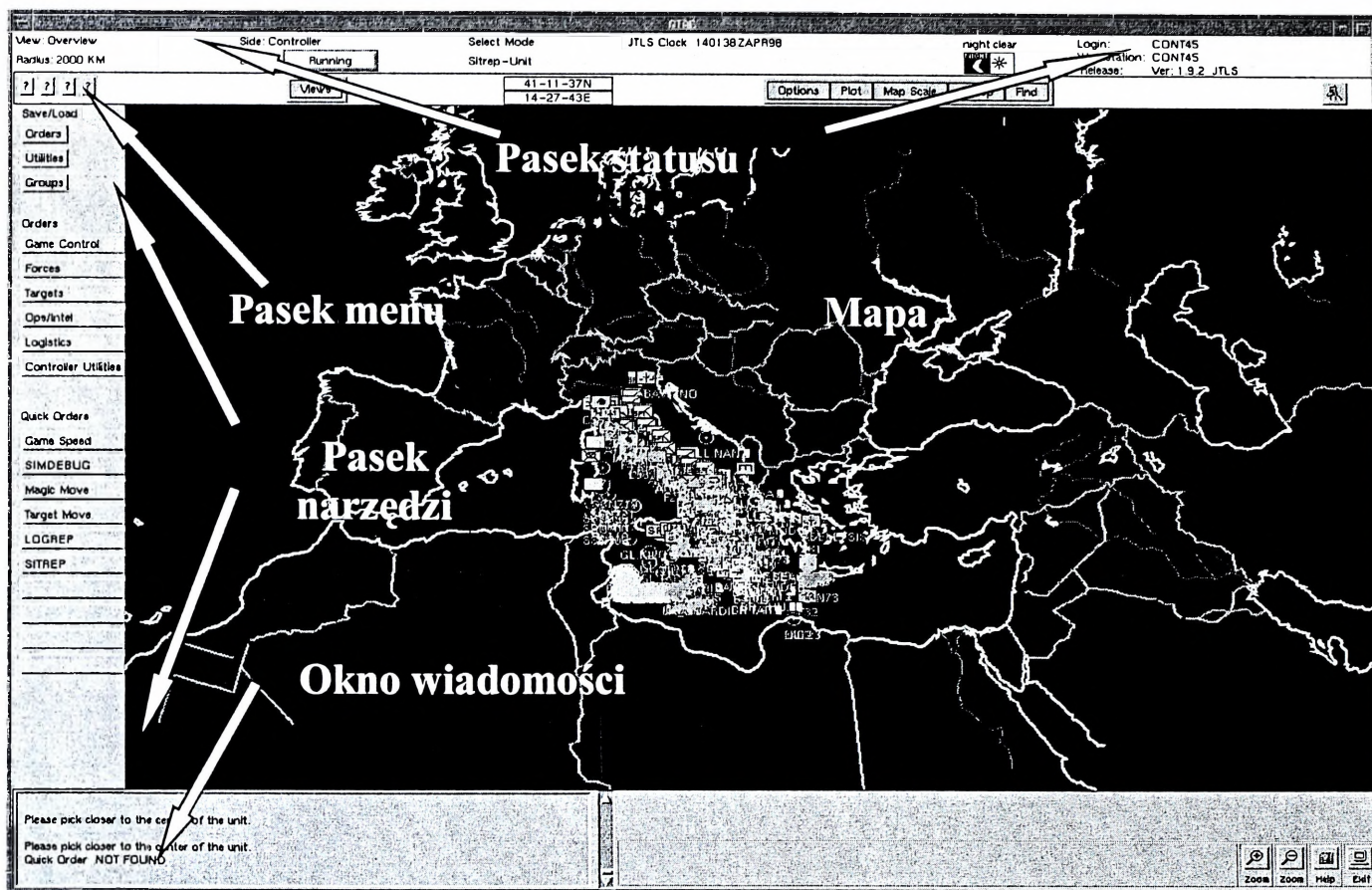
Aplikacje GIAC, IMT, OPM, MPP używane są przez wszystkich użytkowników biorących udział w ćwiczeniu, natomiast aplikacje ATOG i ATOT tylko przez użytkowników związanych z kierowaniem siłami lotniczymi.

Po włączeniu stacji roboczej, na której zainstalowany został system JTLS, w odpowiednim oknie użytkownik uzyskuje dostęp do wszystkich aplikacji użytkowych (patrz rysunek 9).



Rysunek 9. Ekran użytkownika.

Po kliknięciu na odpowiedniej ikonie uruchamiane są przypisane aplikacje. Rysunek 10 przedstawia uruchomioną aplikację GIAC.



Rysunek 10. Okno aplikacji GIAC.

### Program sterujący graficznymi danymi wejściowymi (*Graphical Input Aggregate Control GIAC*)

GIAC jest interfejsem pomiędzy ćwiczącym, a CEP (*Combat Events Program*) opartym o X-Windows. Funkcjonalność aplikacji GIAC wyraża się następująco:

- graficzne zobrazowanie terenu (mapy);
- dostarczanie niezbędnych informacji o terenie działań (współrzędne, odległości);
- wyświetlanie ogólnych informacji o wojskach ćwiczących;
- prowadzenie połączonych działań;
- wydawanie rozkazów;
- wykonywanie przemieszczeń i przegrupowań;
- przydział celów, oddziałów, uprawnień.

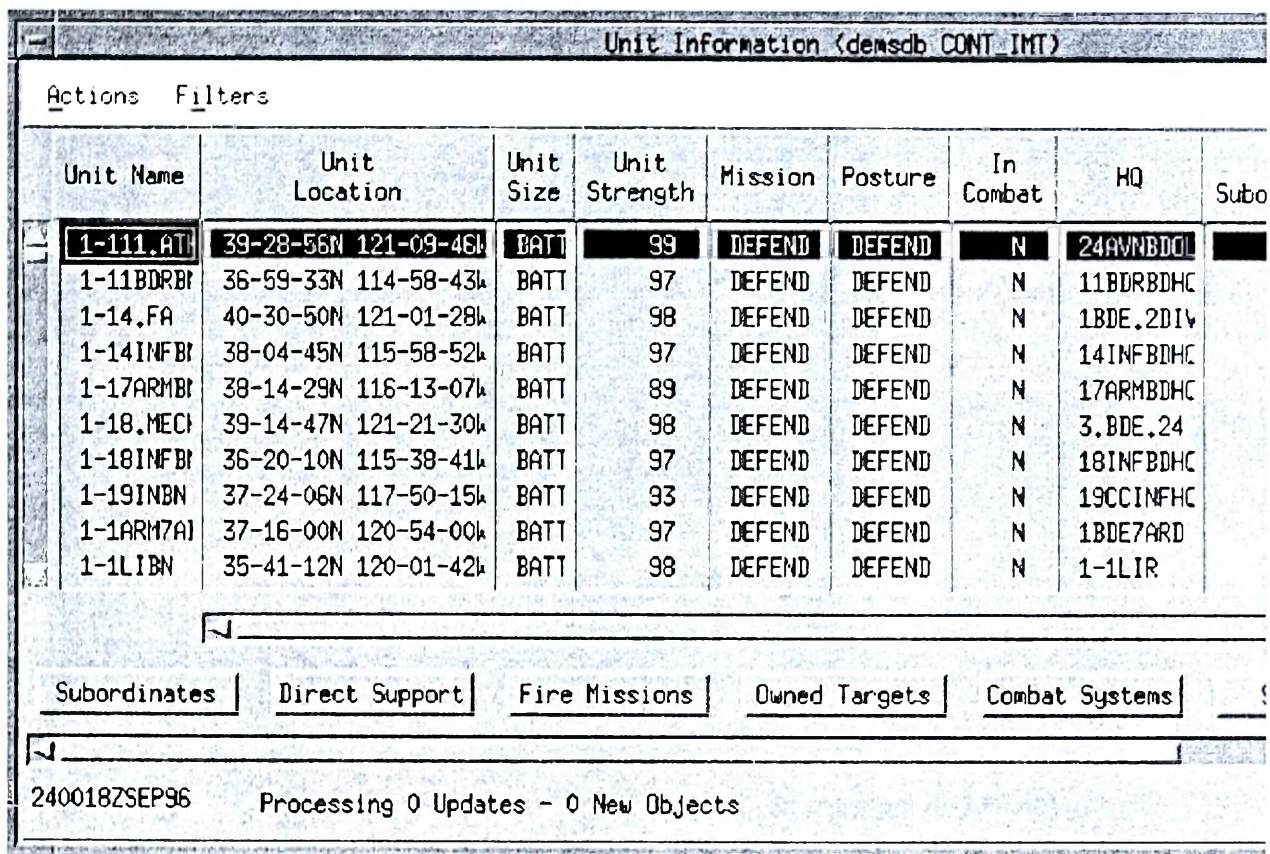
Przy pomocy tej aplikacji można uzyskać podstawowe informacje takie jak: czas operacyjny, który upłynął od rozpoczęcia ćwiczenia, pora doby (noc, dzień), współrzędne geograficzne położenia jednostek. Możliwe jest także wydawanie rozkazów dla wszystkich podległych wojsk.

### Narzędzia zarządzania informacją (*Information Management Tool – IMT*)

Aplikacja IMT pozwala na uzyskanie szczegółowych informacji na temat wybranych oddziałów, grup oddziałów lub innych elementów (np. logistycznych). Podstawową funkcją realizowaną przez aplikację jest sprawdzanie aktualnego statusu wojsk własnych i uzyskanie dostępnych informacji o przeciwniku. W aplikacji można wyświetlić informacje tylko o wybranych pododdziałach (oddziałach) – filtrowanie.

Program posiada około 40 różnych tabel i menu.

Przykładowe zobrazowanie funkcjonowania aplikacji IMT przedstawia rysunek 11.



The screenshot shows a window titled "Unit Information (densdb CONT IMT)". It contains a table with columns: Unit Name, Unit Location, Unit Size, Unit Strength, Mission, Posture, In Combat, HQ, and Subo. The first row is highlighted. Below the table are several menu buttons: Subordinates, Direct Support, Fire Missions, Owned Targets, and Combat Systems. At the bottom, there is a status bar showing "240018ZSEP96 Processing 0 Updates - 0 New Objects".

Unit Name	Unit Location	Unit Size	Unit Strength	Mission	Posture	In Combat	HQ	Subo
1-111.AT	39-28-56N 121-09-48E	BATT	99	DEFEND	DEFEND	N	24AVNBDDC	
1-11BDRBF	36-59-33N 114-58-43E	BATT	97	DEFEND	DEFEND	N	11BDRBDHC	
1-14.FA	40-30-50N 121-01-28E	BATT	98	DEFEND	DEFEND	N	1BDE.2DIV	
1-14INFBI	38-04-45N 115-58-52E	BATT	97	DEFEND	DEFEND	N	14INFBDHC	
1-17ARMBI	38-14-29N 116-13-07E	BATT	89	DEFEND	DEFEND	N	17ARMBDHC	
1-18.MECI	39-14-47N 121-21-30E	BATT	98	DEFEND	DEFEND	N	3.BDE.24	
1-18INFBI	36-20-10N 115-38-41E	BATT	97	DEFEND	DEFEND	N	18INFBDHC	
1-19INBN	37-24-06N 117-50-15E	BATT	93	DEFEND	DEFEND	N	19CCINFHC	
1-1ARM7AI	37-16-00N 120-54-00E	BATT	97	DEFEND	DEFEND	N	1BDE7ARD	
1-1LIBN	35-41-12N 120-01-42E	BATT	98	DEFEND	DEFEND	N	1-1LIR	

Rysunek 11. Przykładowa tabela w aplikacji IMT.

### Program przetwarzania wiadomości (*Message Processor Program*)

Aplikacja ta umożliwia komunikowanie się użytkowników między sobą, a także komunikowanie się z aplikacją CEP (*Combat Event Program*).

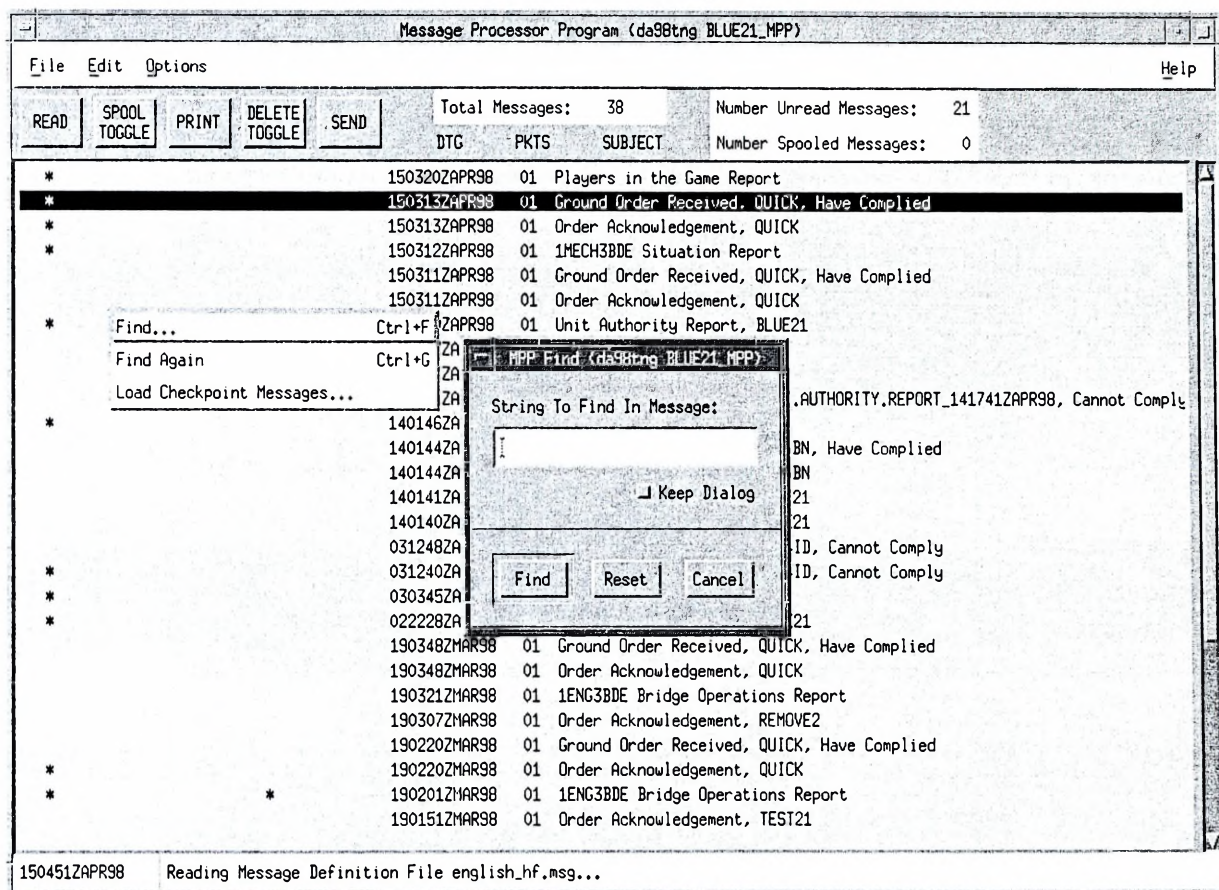
Funkcjonalność aplikacji:

- przesyłanie wiadomości z/do aplikacji CEP;
- odbieranie wiadomości e-mail i ich drukowanie;
- do czasu osiągnięcia gotowości aplikacji MPP do pracy wiadomości są kolejgowane przez aplikację CEP;

- wiadomości mogą być przesyłane do wszystkich (*broadcast*), jak i do określonej grupy użytkowników.

Poprzez aplikację MPP ćwiczący (dowódca) wydający rozkaz otrzymuje wiadomość (od aplikacji CEP) o błędach popełnionych przy wydawaniu rozkazu (np. niewłaściwie współrzędne położenia, nazwy jednostek).

widok interfejsu aplikacji przedstawia rysunek 12.



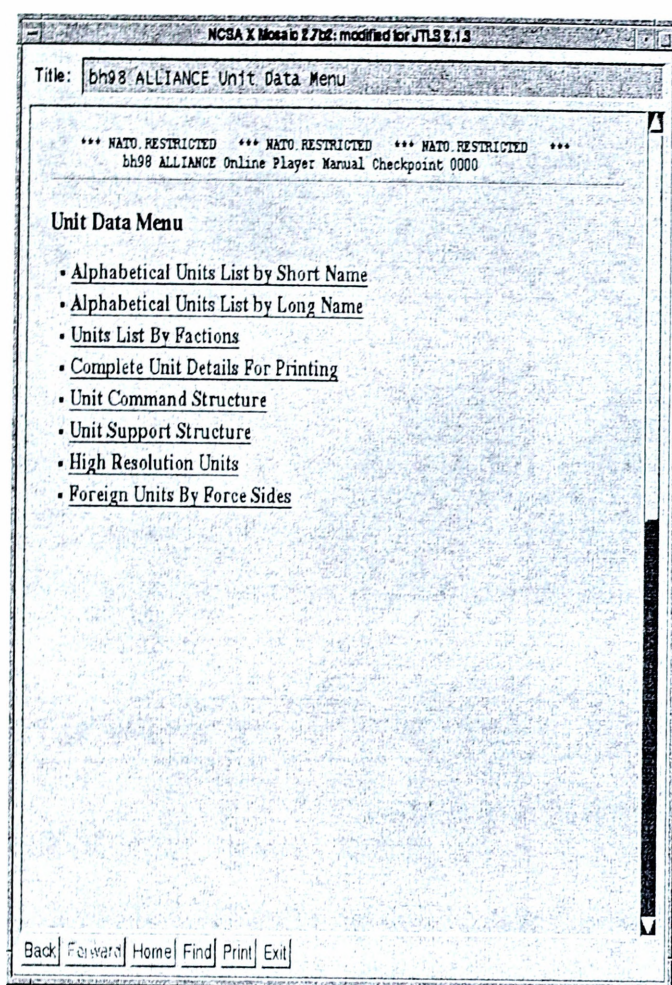
Rysunek 12. Przykładowe okno aplikacji MPP.

### Podręcznik gracza (*Online Player's Manual*)

Aplikacja OPM spełnia następujące funkcje:

- umożliwia dostęp do plików tekstowych zawierających wszystkie dane inicjujące scenariusz;
- pozwala zrozumieć zdolności bojowe i potencjał każdej jednostki biorącej udział w ćwiczeniach;
- dane są generowane na podstawie bazy danych przygotowanej dla konkretnego ćwiczenia;
- informacje są wyświetlane poprzez przeglądarkę HTML (np. MOZILLA);
- dane zawarte w plikach tekstowych są statyczne i opisują bazę danych, z której zostały wygenerowane.

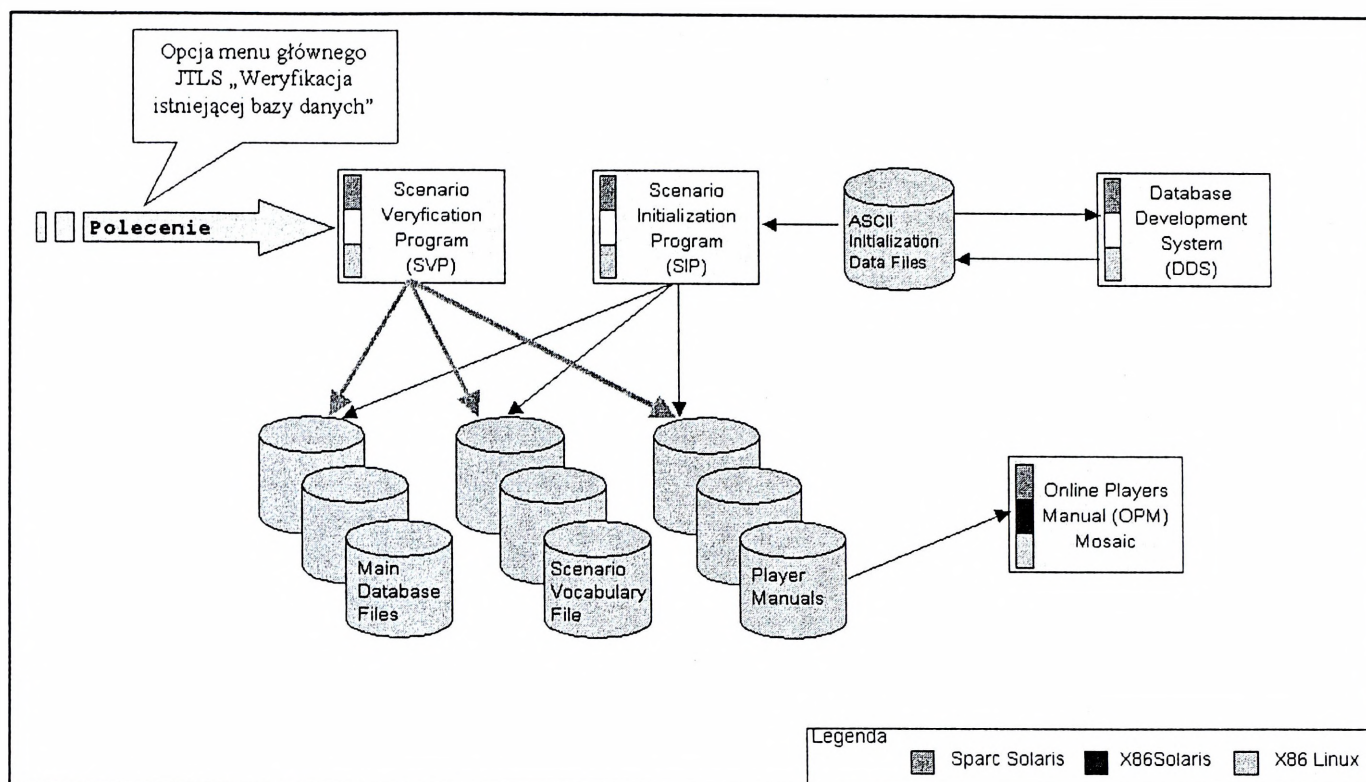
Interfejs aplikacji przedstawia rysunek 13.



Rysunek 13. Przykładowe okno aplikacji OPM.

### 3.3. Podstawowe przepływy informacji

Dane wejściowe do systemu JTLS przygotowywane są w postaci plików w formacie ASCII. Format ASCII został przyjęty głównie w celu umożliwienia przygotowania danych za pomocą własnych narzędzi spoza systemu JTLS. Dane z plików ASCII są importowane przez program tworzenia i rozwoju bazy danych (*Database Development System*) w skrócie DDS. Program DDS po zaimportowaniu danych umożliwia ich aktualizację, przeglądanie (wraz z wyszukiwaniem danych). Ważną funkcją programu DDS jest export danych do formatu ASCII, który jest wykonywany w celu archiwizacji danych lub w celu tworzenia nowych baz danych na podstawie istniejącej bazy danych. Po dokonaniu wymaganych aktualizacji i eksporcie danych do formatu ASCII, dane mogą zostać wczytane do systemu JTLS przez program inicjalizacji scenariusza (*Scenerio Initialization Program*) w skrócie SIP. Program SIP tworzy docelowe zbiory systemu JTLS (bazy danych ORACLE).



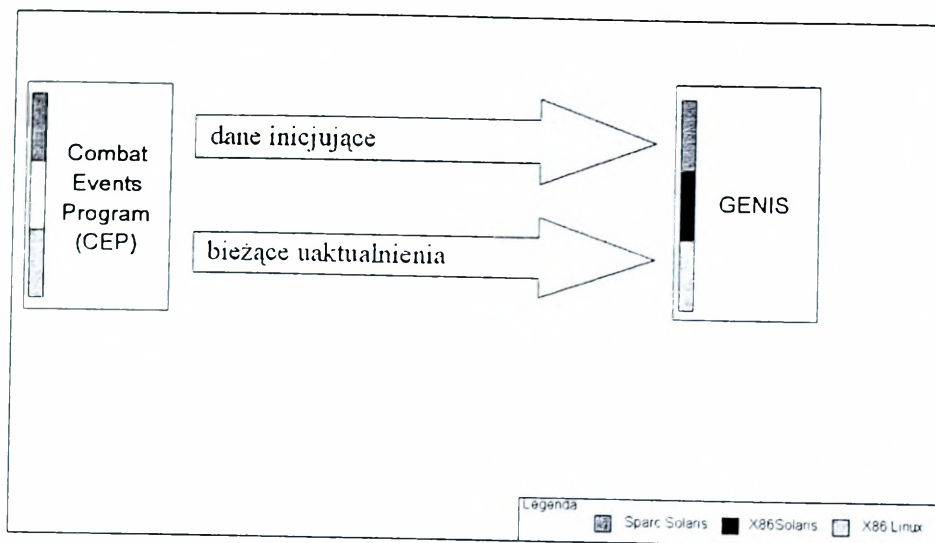
**Rysunek 14. Przepływ informacji w fazie inicjalizacji systemu.**

/źródło: opracowanie własne/

Po utworzeniu baz danych następnym krokiem jest uruchomienie programu weryfikacji scenariusza (*Scenario Verification Program*) w skrócie SVP. Program SVP nie jest uruchamiany automatycznie (choć jest częścią programu SIP). Uruchomienie programu SIP następuje po wyborze opcji menu głównego systemu JTLS „Weryfikacja istniejącej bazy danych”. Program SVP dokonuje weryfikacji danych scenariusza. W przypadku wystąpienia błędów, umożliwia wydruk raportów dotyczących błędów. Proces weryfikacji oraz poprawiania błędów może mieć charakter iteracyjny, który zakończy się w chwili, gdy w czasie weryfikacji danych nie zostaną wykryte błędy scenariusza. Po zakończeniu weryfikacji danych tworzone są statyczne (w trakcie trwania symulacji) zbiory instrukcji dla ćwiczących (*Player Manuals*). Zbiory *Player Manual* są wykorzystywane przez program umożliwiający wyświetlanie danych wykorzystanych przy tworzeniu scenariusza za pomocą przeglądarki internetowej (umożliwiającej łatwe przeglądanie danych oraz ich przeszukiwanie).

### Faza symulacji (przetwarzania)

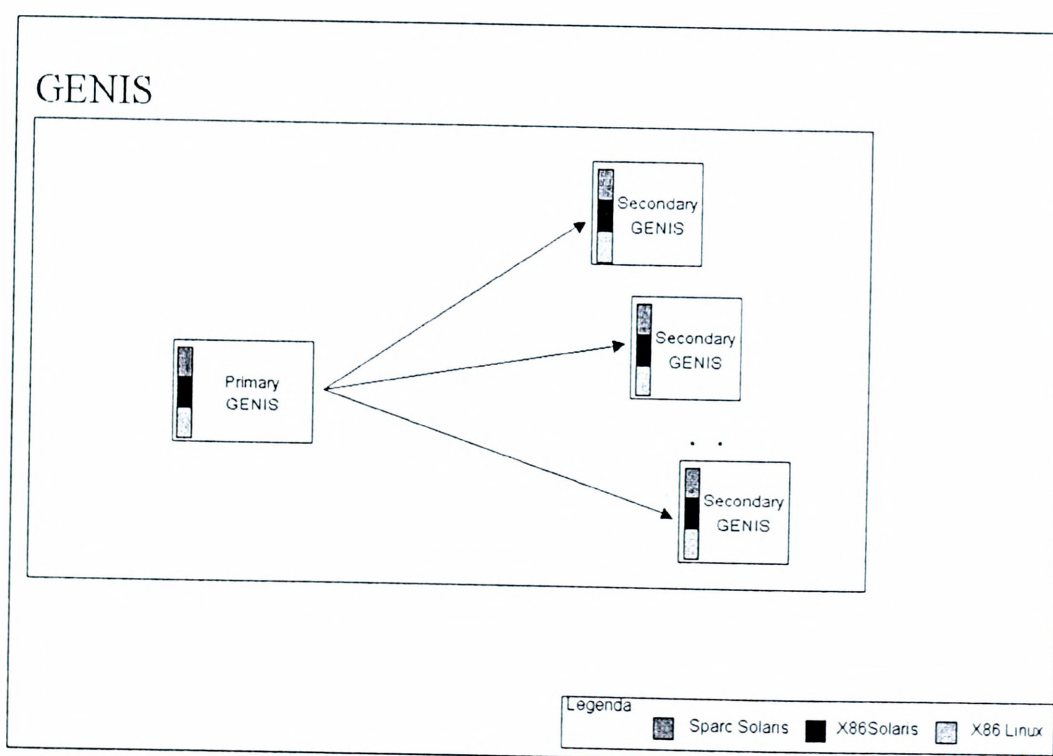
Główny program systemu JTLS (*Combat Events Program*) w skrócie CEP na podstawie danych inicjujących scenariusz przesyła dane inicjujące do procesu GENIS. Program CEP komunikuje się tylko z jednym głównym procesem GENIS przez protokół TCP/IP. W trakcie trwania symulacji do procesu GENIS przesyłane są bieżące uaktualnienia.



**Rysunek 15. Wymiana informacji pomiędzy CEP a GENIS.**

/źródło: opracowanie własne/

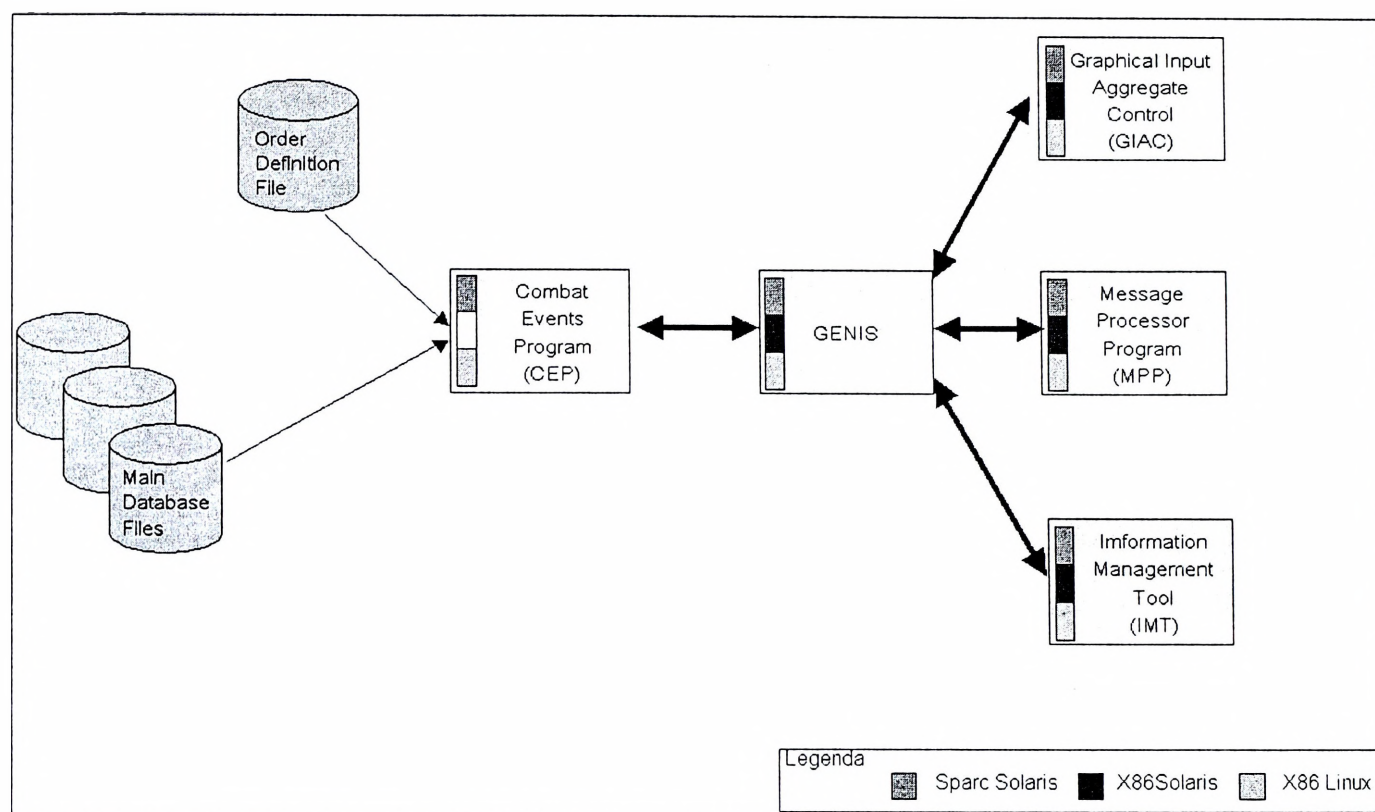
W systemie JTLS określenie procesu GENIS dotyczy tak naprawdę procesu primary GENIS oraz współpracujących z nim procesów secondary GENIS. Liczba procesów secondary GENIS jest równa liczbie stacji roboczych dla uczestników ćwiczenia (uczestnikiem ćwiczenia może być zarówno gracz jak i kierownik ćwiczenia). Wynika to z faktu, że pojedynczy proces secondary GENIS jest odpowiedzialny za obsługę programów interfejsu użytkownika: GIAC, MPP oraz IMT.



**Rysunek 16. Współpraca Primary GENIS z Secondary GENIS.**

/źródło: opracowanie własne/

Główny program systemu CEP korzysta ze zbioru danych zawierającego dane na temat scenariusza (*Order Definition File*) oraz głównych zbiorów danych systemu. Dane uzyskane w wyniku symulacji przesyłane są następnie do programu GENIS, który z kolei przesyła je do programów interfejsu użytkownika systemu JTLS: programu GIAC, programu MPP oraz programu IMT.



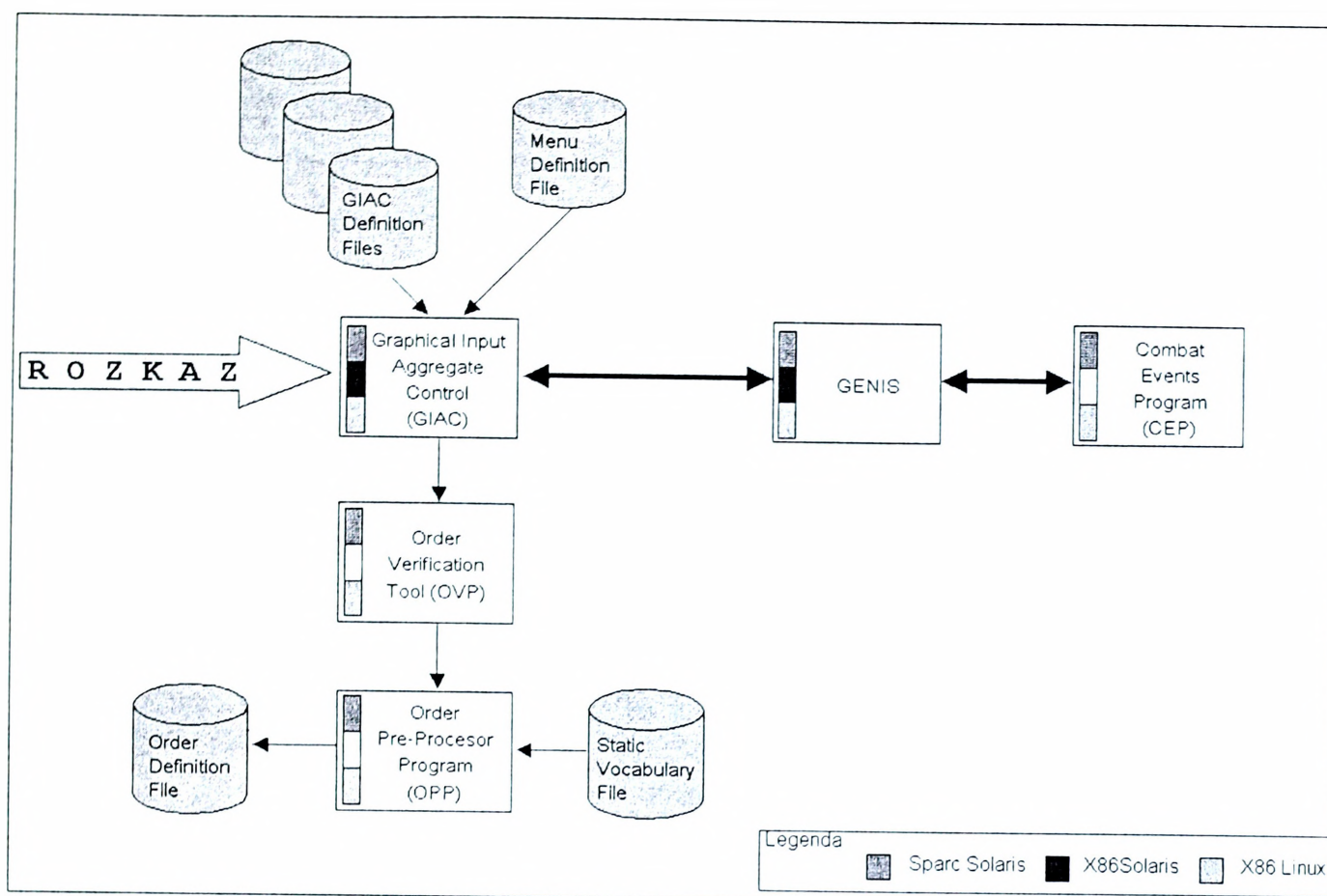
Rysunek 17. Współpraca programu CEP z programem GENIS.

/źródło: opracowanie własne/

### Faza prezentacji wyników symulacji

W fazie prezentacji wyników program GENIS komunikuje się z trzema programami interfejsu użytkownika: GIAC, MPP oraz IMT.

Program GIAC jest głównym programem interfejsu użytkownika systemu. Program umożliwia między innymi wprowadzanie rozkazów. Po wprowadzeniu rozkazu następuje jego weryfikacja (semantyczno-syntaktyczna) oraz przesłanie do programu weryfikacji rozkazu (*Order Verification Program*). Następnie rozkaz jest przesyłany do programu *Order Pre-Processor*, który jest odpowiedzialny za sprawdzenie poprawności zasadności wydania rozkazu w określonej w danej chwili sytuacji taktyczno-operacyjnej.

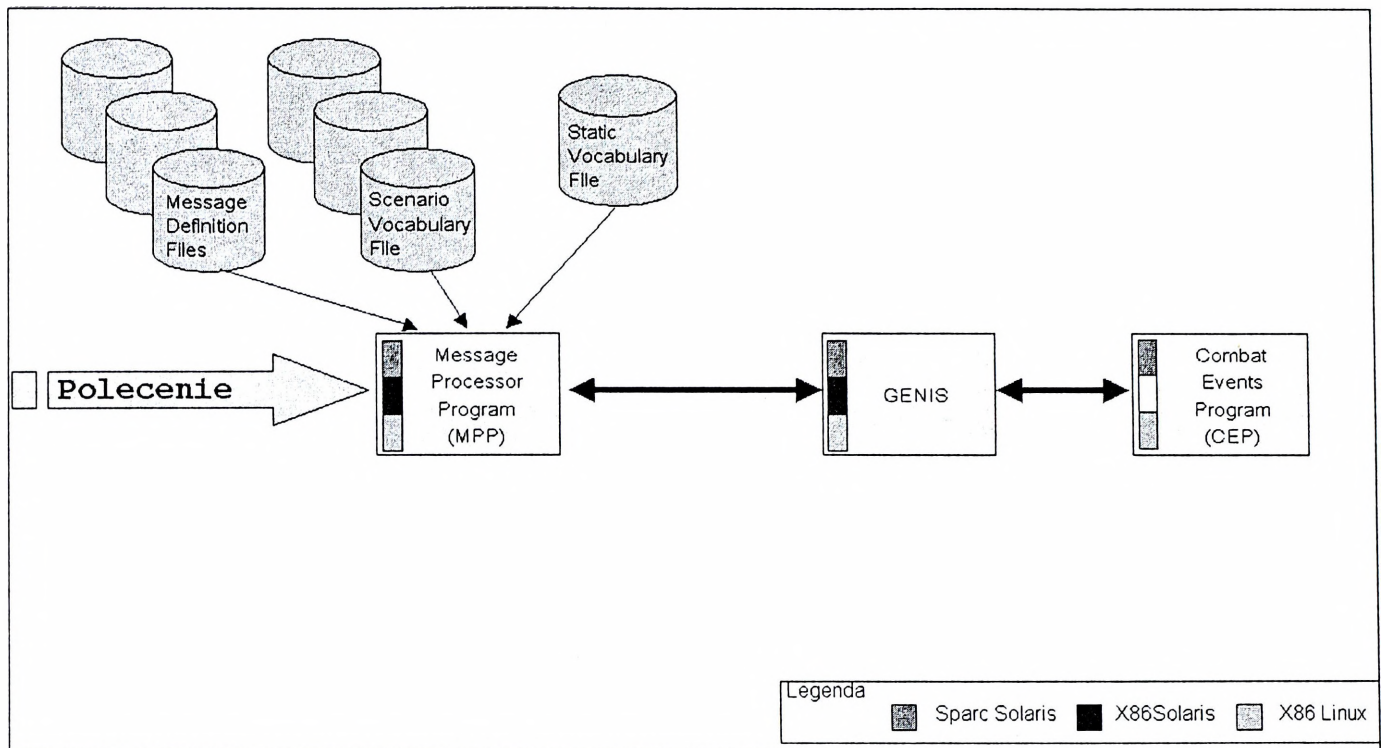


Rysunek 18. Przepływ informacji dla programu GIAC.

/źródło: opracowanie własne/

Po zakończeniu weryfikacji rozkazu zostaje on zapisany w zbiorze definicji rozkazów (*Order Definition File*). Program GIAC przesyła komunikat do systemu GENIS, który z kolei komunikuje się z programem CEP.

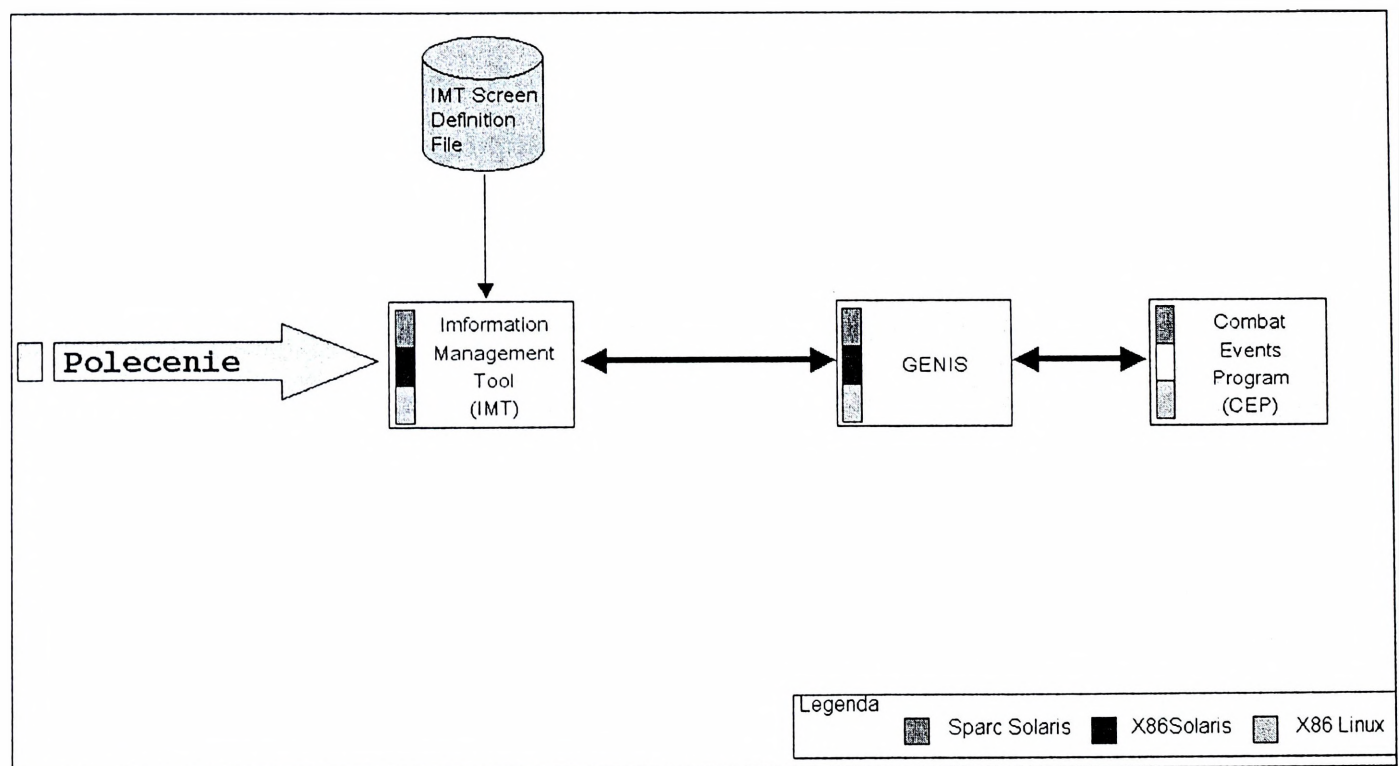
Program MPP służy do wyświetlania wiadomości w postaci tabelarycznej (zagregowanej). Ćwiczący może dokonać zmiany ustawień oraz skonfigurować program MPP stosownie do własnych potrzeb, które są zapisywane do zbiorów definicji wiadomości (*Message Definition Files*). Program MPP korzysta także ze zbiorów o charakterze słownikowym głównie w celu ograniczenia ilości przesyłanych danych (przesyłane są tylko kody zamiast pełnych danych opisowych – które są statycznie, zapisane w słownikach i niezmiennie w trakcie symulacji). Po wprowadzeniu polecenia przez ćwiczącego program MPP komunikuje się z programem GENIS, a ten z kolei z głównym programem systemu CEP.



**Rysunek 19. Przepływ informacji dla programu MPP.**

/źródło: opracowanie własne/

Program IMT podobnie jak program MPP może zostać skonfigurowany dla potrzeb ćwiczącego. Ustawienia konfiguracyjne zapisywane są do zbiorów definicyjnych programu IMT (*IMT Screen Definition File*). Po wprowadzeniu polecenia do programu IMT komunikuje się on z programem GENIS, a program GENIS z programem CEP.

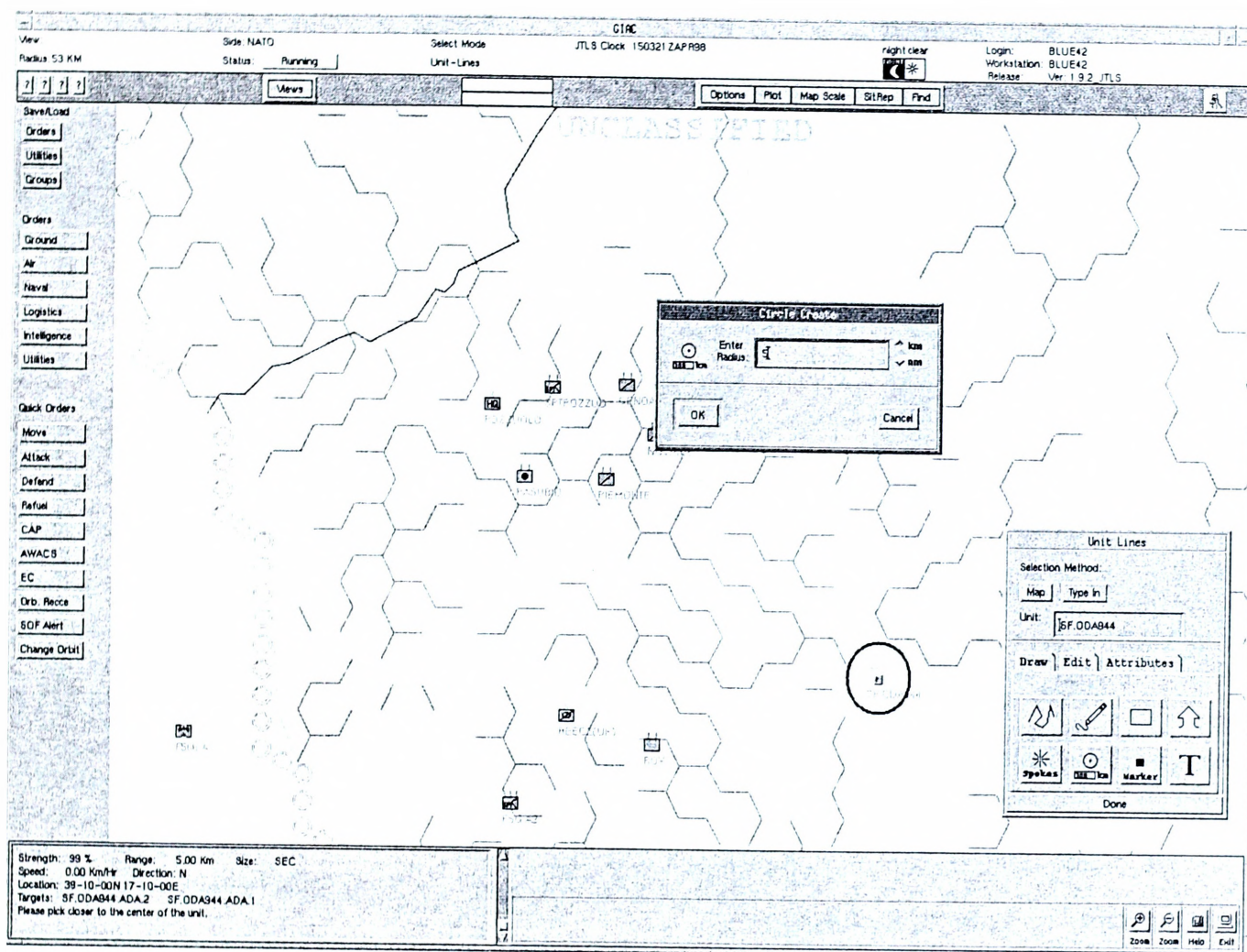


**Rysunek 20. Przepływ informacji dla programu IMT.**

/źródło: opracowanie własne/

### 3.4. Odwzorowanie terenu

W systemie JTLS teren odwzorowywany jest poprzez aplikacje GIAC (*Graphical Input Aggregate Control*). Aplikacja ta wyświetla mapę obszaru na którym toczą się działania bojowe (ćwiczenia). Patrz rysunek 21.

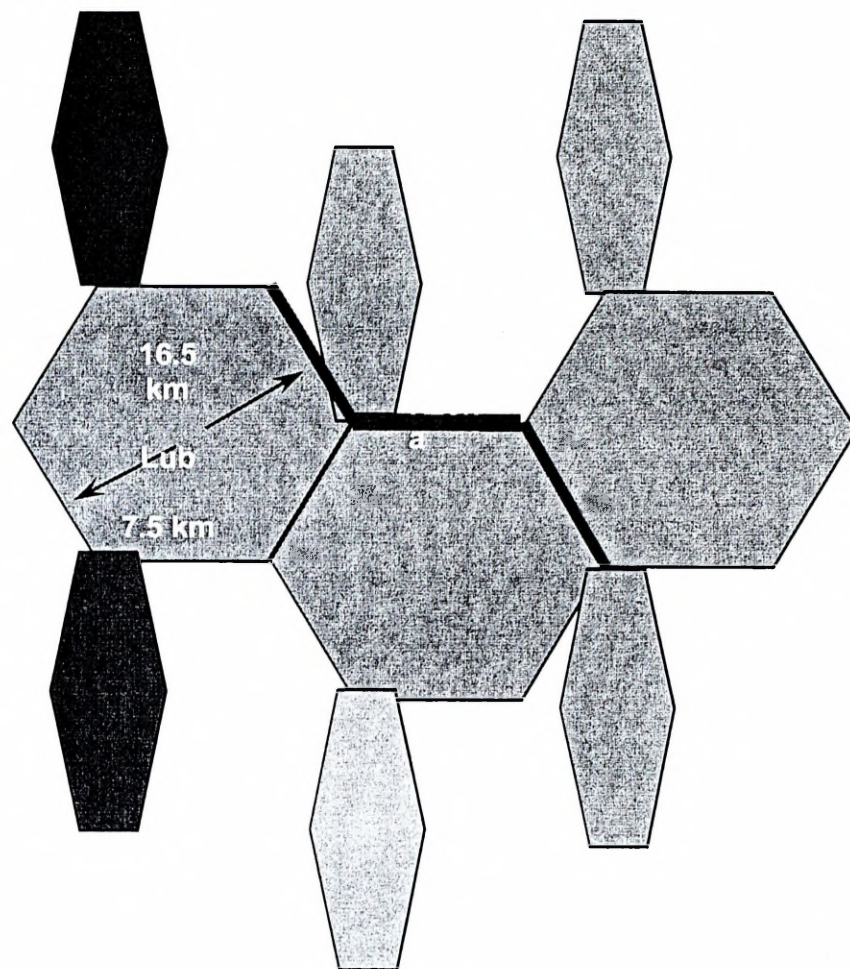


Rysunek 21. Przykładowe zobrazowanie w GIAC obszaru działań bojowych.

Mapy cyfrowe opracowane przez Defense Mapping Agency's ADRG pozwalają na użycie JTLS na całym świecie. W celu przygotowania ćwiczenia mapy poddaje się edycji dostosowując ich zawartość do warunków konkretnego ćwiczenia. Możliwa jest np. zmiana przejezdności terenu, zmiana typu tereny (np. z zalesionego na pustynny), ustawienie dodatkowych przeszkód terenowych jak np. zapory przeciwczołgowe itp.

Do edycji terenu stosuje się zestaw aplikacji TPU (*Terrain Modification Utility*). Zasady edycji mapy i użycia tego narzędzia zawarte są w instrukcji „*The JTLS Database Development System User's Guide*”.

Maksymalny obszar, który może obejmować prowadzone ćwiczenia wynosi 3400 km x 3400 km (2000 x 2000 mil morskich). Mapa poddawana edycji podzielona jest na sześciokątne klastry o wymiarach 16,5 km lub 7,5 km (patrz rysunek 22).



**Rysunek 22. Sześciokątne klastry w JTLS.**

/źródło: prezentacja z konferencji z dnia 22.08.2003/

Ukształtowania terenu, które mogą wystąpić w systemie JTLS:

- teren otwarty - kolor biały lub szary;
- lasy - kolor zielony;
- teren górzysty - kolor brązowy;
- miasta - kolor rdzy (ang. rust);
- wody - kolor niebieski.

Ponadto w systemie mogą wystąpić następujące typy terenu:

- “Open” “Good Road - Poor Road”
- “Forest Good Road - Forest Poor Road - Forest”
- “Desert Good Road - Desert Poor Road - Desert”
- “City - Rubbled City”
- “Mountain Good Road - Poor Mountain- Mountain”
- “Ocean”

System umożliwi także zdefiniowanie do 26 przeszkód terenowych, które mają wpływ na szybkość przemieszczania się wojsk i na przebieg walki (ćwiczenia). Typy przeszkód terenowych przedstawia tabela 1.

BARRIER NAME	BARRIER INDEX
BARRIER OPEN TERRAIN	1
BARRIER RIVER	2
BARRIER WADI	3
BARRIER TANK DITCH	4
BARRIER IMPASSABLE	5
BARRIER NAVIGABLE RIVER	6
BARRIER SHORE	7
BARRIER SHORE LCAC	8
BARRIER PROBLEM TERRAIN	9
BARRIER LIMITING TERRAIN	10
BARRIER DIFFICULT TERRAIN	11
BARRIER HEAVILY BRIDGED RIVER	12
BARRIER BRIDGED RIVER	13
BARRIER LIGHTLY BRIDGED RIVER	14
BARRIER HEAVILY BRIDGED WADI	15
BARRIER BRIDGED WADI	16
BARRIER LIGHTLY BRIDGED WADI	17
BARRIER HEAVILY BRIDGED TANK DITCH	18
BARRIER BRIDGED TANK DITCH	19
BARRIER LIGHTLY BRIDGED TANK DITCH	20
BARRIER RESTRICTED AREA	21
BARRIER MODERATELY RESTRICTED AREA	22
BARRIER SEVERELY RESTRICTED AREA	23
BARRIER HEAVILY BRIDGED NAVIGABLE RIVER	24
BARRIER BRIDGED NAVIGABLE RIVER	25
BARRIER LIGHTLY BRIDGED NAVIGABLE RIVER	26

Tabela 1. Zestawienie przeszkód terenowych.

/źródło: JTLS analyst's guide v.2.2/

### Przemieszczanie oddziałów, pododdziałów w terenie

Przemieszczenia oddziałów (pododdziałów) dzieli się na:

- „**taktyczne**”- zainicjowany poprzez wydanie rozkazu do natarcia, opóźnienia lub wycofania;
- „**administracyjne**”- bez styczności z przeciwnikiem, zainicjowane pozostałymi rozkazami do zmiany lokalizacji jednostki.

Zazwyczaj przemieszczenie administracyjne jest szybsze (w zależności od inicjujących baz danych) od przemieszczenia taktycznego, jednakże jednostki przemieszczające się administracyjnie posiadają dwa ograniczenia:

1. Nie mogą prowadzić ognia artyleryjskiego.
2. Jeżeli natkną się na obcą jednostkę, która zacznie prowadzić do niej ogień bezpośredni (nie artyleryjski), przejdą do obrony zanim ćwiczący nie podejmie innej decyzji.

Jednostki przemieszczające się taktycznie nie podlegają tym ograniczeniom.



Rozkaz do przemieszczenia wydaje się używając odpowiedniej formatki z menu aplikacji GIAC (patrz rysunek 25), używając mapy lub wpisując odpowiednie dane do pól tekstowych.

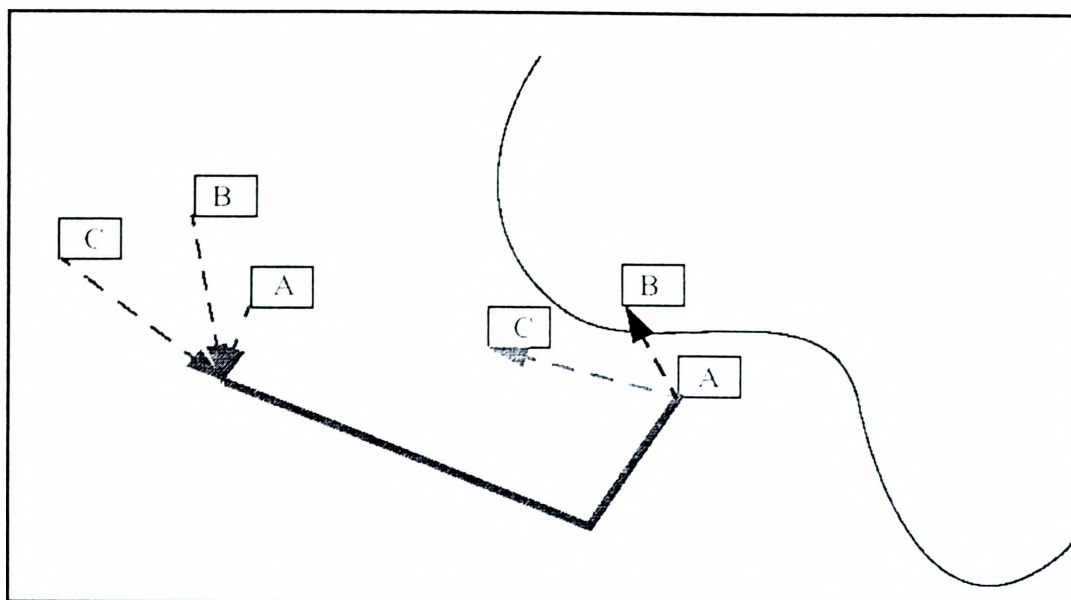
Rysunek 25. Formatka rozkazu przemieszczenia.

Parametr długości marszu na dobę (*MAX MOVE HOURS PER DAY*) z *Tactical Unit Prototypes* (TUP) – jest określony dla każdej jednostki podczas jej definiowania na podstawie posiadanego przez jednostkę wyposażenia. Określa on ilość godzin w ciągu doby jak długo jednostka może się przemieszczać. Jeżeli dana jednostka w ciągu ostatnich 24 godzin wykorzystała ten limit (*MAX MOVE HOURS PER DAY*), to mimo wydania rozkazu przemieszczenia dla tej jednostki pozostaje ona w tym samym miejscu, do czasu odtworzenia gotowości do przemieszczenia.

Przykład: zakładając, że:

1. Zdefiniowano *MAX MOVE HOURS PER DAY* z TUP dla danej jednostki na 12 godzin.
2. W ciągu ostatnich 24 godzin jednostka wykonała jedno przemieszczenie w ciągu 12 godzin i wykorzystała limit godzin na przemieszczenie.
3. 5 godzin od czasu ostatniego przemieszczenia jednostka otrzymała kolejny rozkaz przemieszczenia.





**Rysunek 27. Przykład przemieszczenia grupy jednostek.**

/źródło: JTLS players guide v.2.2/

Przykład.

Jednostka A jest jednostką podstawową. Gruba linia przedstawia wprowadzoną trasę przemarszu (rysunek 27). Jednostka B i C są na liście jednostek. Wszystkie jednostki przemieszczają się do pierwszego punktu trasy (przerywane linie), następnie podążają po niej do końca. Podstawowa jednostka zatrzymuje się na końcu marszruty, a pozostałe jednostki z listy jednostek dokonują końcowego ruchu (przerywana linia) do lokalizacji względem położenia jednostki A identycznie jak przed przemieszczeniem. Jednostka B nie może zająć odpowiedniej lokalizacji ponieważ jest tam zbiornik wodny. W tym przypadku jednostka B zatrzyma się na końcu marszruty i będzie miała to samo położenie co jednostka A.

W JTLS zaimplementowano granice państw co pozwala ćwiczącym na modelowanie przemieszczeń jednostek i misji z uwzględnieniem suwerenności terytoriów, politycznych i ekonomicznych stref oraz operacyjnych ograniczeń takich jak obszary z zakazem lotów. Każda strona posiada pozwolenie na przekroczenie wprowadzonej granicy dla: misji lotniczych, jednostek lądowych, jednostek morskich, HRU, transportu barkami, transportu kolejowego. Ograniczenia przekroczenia granicy są narzucone albo przez inicjującą bazę danych, albo rozkazy ćwiczącego.

Każda ze stron posiada inicjujące ustawienia określające, które jednostki i misje będą respektować indywidualne granice. Na przykład misje lotnicze mogą mieć zabronione przekraczanie granicy XYZ, ale jednostki lądowe mogą nie podlegają tym ograniczeniom. W każdym rozkazie posiadających pole *Override Boundary Restriction*, ćwiczący może określić,

które granice jednostka lub misja może lub nie może przekroczyć bez względu na ustawienia inicjujące.

### 3.5. Modele formalne

W systemie JTLS wykorzystane są modele stochastyczne, probabilistyczne<sup>4</sup> i deterministyczne.

Modele stochastyczne opierają się na procesach stochastycznych, to znaczy takich, które zbudowane są z sekwencji losowo generowanych wartości – przynajmniej jeden parametr lub zmienna modelu jest zmienną losową. Wykorzystywane są do modelowania strat w skali mikro (pojedynczych elementów).

W modelach probabilistycznych<sup>5</sup> także występuje zmienna losowa, ale znany jest jej rozkład prawdopodobieństwa. Model udziela potrzebnych odpowiedzi bez konieczności poddawania go eksperymentom polegającym na doprowadzeniu do modelu przebiegów losowych.

W modelach deterministycznych nie występują zmienne losowe, zbudowane są one na ścisłych zależnościach matematycznych opartych o zasady praw statystycznych. Takie same dane wejściowe modelu dają zawsze taki sam wynik. Wykorzystywane są do modelowania strat w skali makro.

W JTLS przy pomocy modeli stochastycznych modeluje się działania lotnicze (przy użyciu współczynników wyrażonych prawdopodobieństwem zniszczenia –  $pH^6$  i prawdopodobieństwem zabicia –  $pK^7$ ) oraz działania ziemia-powietrze (przy użyciu współczynników wyrażonych prawdopodobieństwem zaangażowania –  $pE^8$  i prawdopodobieństwem zabicia).

W modelowaniu działań pocisków powietrze-ziemia, ziemia-ziemia i artylerii używa się modeli stochastycznych dla broni precyzyjnego rażenia (używa się  $pH$  i  $pK$ ), bądź probabilistycznych dla broni uderzającej w obszar (metoda oczekiwanej wartości).

W module walki lądowej użyto model walki Lanchestera. Klasyczny model Lanchestera stosowano do modelowania walki na szczeblu strategicznym do I wojny światowej. Od tamtej pory wprowadzono wiele modyfikacji modelu, tak aby mógł być stosowany do obecnych, złożonych operacji. Modyfikacje te polegały na dodawaniu kolejnych zmiennych i współczynników. Zmodyfikowany model Lanchestera,

---

<sup>4</sup> W oryginalnej dokumentacji autorzy nie posłużyli się tym pojęciem, ale użyte metody obliczeń pozwalają przypuszczać, że użyto modeli probabilistycznych.

<sup>5</sup> Por.: Spustek H.: *Wybrane zagadnienia badań operacyjnych i modelowania liniowego*. AON 2002, str. 12-14, 20-21.

<sup>6</sup> *Probability of Hit*.

<sup>7</sup> *Probability of Kill*.

<sup>8</sup> *Probability of Engage*.

uwzględniający potencjał bojowy, a więc aspekty jakościowe systemów broni wyrażony jest w postaci układu równań<sup>9</sup> (patrz wzór 1).

$$\begin{cases} \frac{dP^A(t)}{dt} = -k^B \cdot \Lambda^B(t) \cdot P^B(t) \\ \frac{dP^B(t)}{dt} = -k^A \cdot \Lambda^A(t) \cdot P^A(t) \end{cases}$$

gdzie:

$dP^A(t)$  straty stron w czasie

$k^A, k^B$  współczynniki określające zmienne modelu

$\Lambda^A(t), \Lambda^B(t)$  gęstość strumienia strzałów skutecznych określonego środka walki

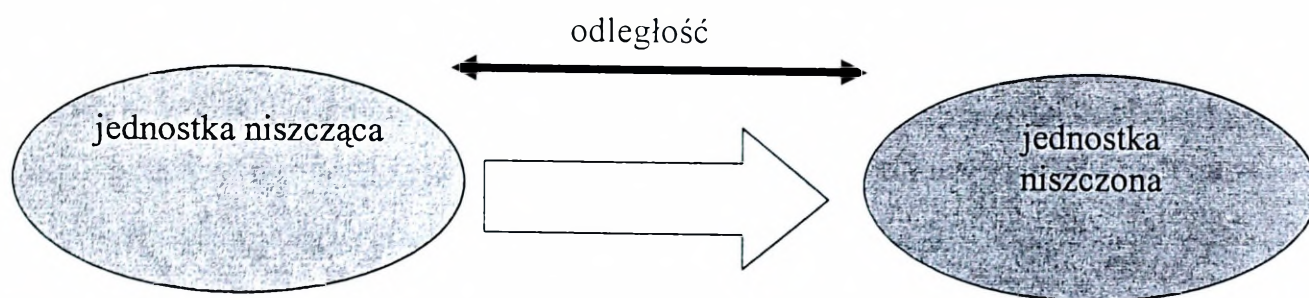
$P^A(t), P^B(t)$  potencjały bojowe stron

#### Wzór 1. Zmodyfikowane równania Lanchestera.

Wykorzystany w JTLS model walki Lanchestera określono jako: *mieszany, heterogeniczny, zatrzymywany czasem model Lanchestera*. Pojęcie *mieszany* oznacza, że zostały użyte zarówno formuły ognia bezpośredniego jak i pośredniego. *Heterogeniczny* oznacza, że pojedyncze typy systemów walki niszczą osobno określone typy systemów walki przeciwnika. *Zatrzymywany czasem* jest model, kiedy dochodzi do określonego zdarzenia – oddziaływania pomiędzy składnikami systemu. Zdarzenia te zachodzą w określonym w scenariuszu czasie.

<sup>9</sup> Ratajczak M., Wieleba R., Wocial J: *Symulacyjny model walki na szczeblu taktycznym. Synteza koncepcyjnego modelu walki*. AON 2002, str. 90.

Walkę lądową przedstawioną jako oddziaływanie jednostki niszczącej na jednostkę niszczonej przedstawia rysunek 28.



- efektywność rażenia jednostki niszczącej
- zasady zaangażowania (ROE)
- rodzaj działania bojowego

- odporność na uderzenia jednostki niszczonej
- zasady zaangażowania (ROE)
- rodzaj działania bojowego

Rysunek 28. Oddziaływanie jednostek w walce lądowej.

/źródło: opracowanie własne/

Czynniki mające wpływ na wielkość obliczanych strat w modelu Lanchestera to:

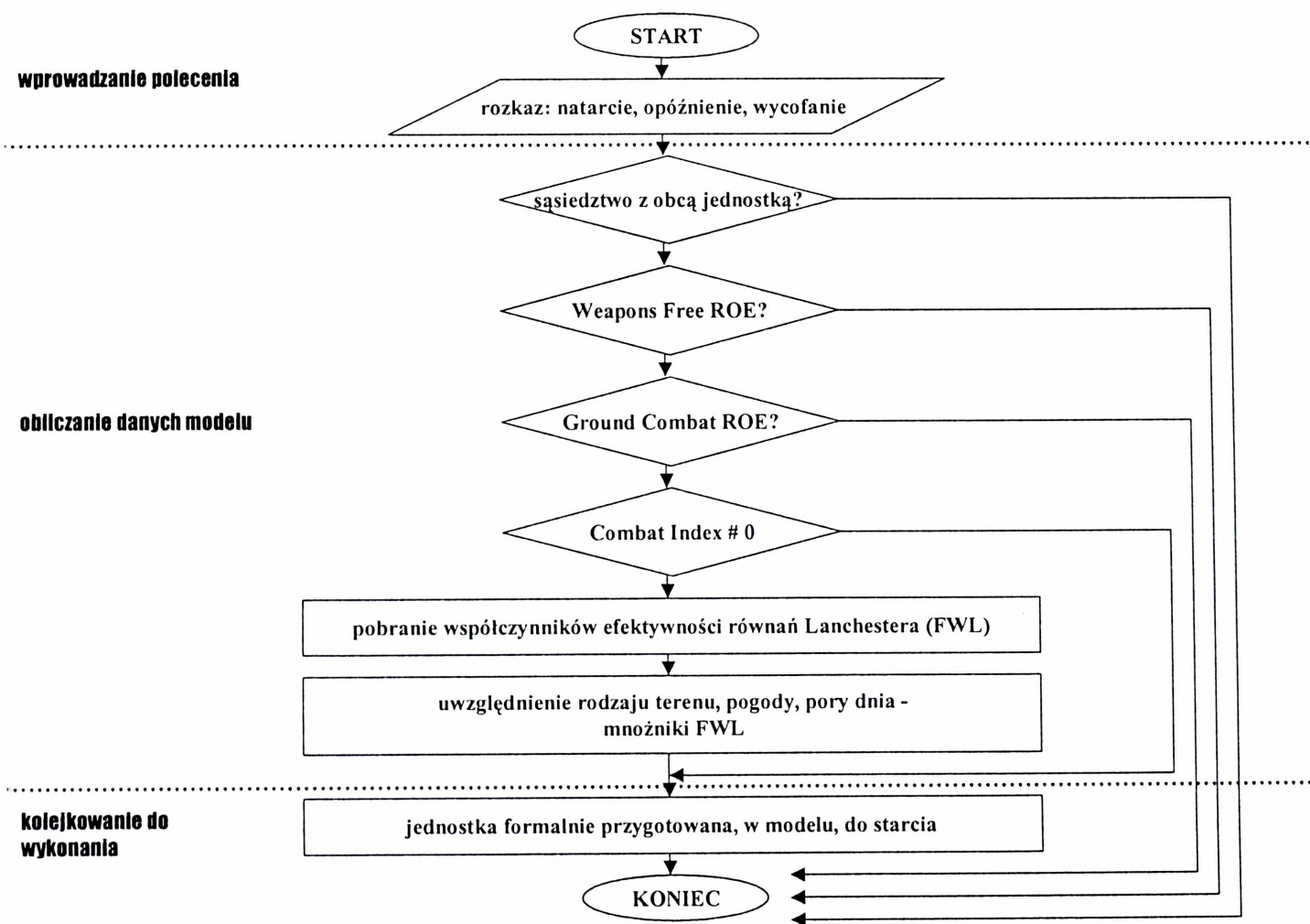
- charakterystyki walczących jednostek (efektywność rażenia jednostki niszczącej – *FIREPOWER LETHALITY PROTOTYPE*, odporność na uderzenia jednostki niszczonej – *SURVIVABILITY PROTOTYPE*, ich działania bojowych, wsparcie);
- charakterystyki systemów walki walczących jednostek (zasięg, algorytmy zniszczeń, użyta amunicja i paliwo);
- charakterystyki środowiska (pora dnia, pogoda, odległość pomiędzy jednostkami, typ terenu);
- odległość pomiędzy walczącymi jednostkami i postawa niszczonej jednostki. Dana wejściowa – *DECISIVELY ENGAGED DISTANCE*, określa odległość w jakiej mogą pojawić się zwiększone straty w jednostce niszczonej;
- poruszanie się atakującej jednostki po polu minowym założonym przez jednostkę broniącą się.

Walka lądowa rozpoczyna się, gdy jednostka przemieszcza się lub próbuje przemieścić się do nowej lokalizacji w wyniku rozkazu przemieszczenia taktycznego (patrz rysunek 29). Jeżeli znajdzie się w sąsiedztwie rastra sześciokątnego (heksogenu) zawierającego obce jednostki lub ma się znaleźć w samym rastrze zawierającym obce jednostki, JTLS sprawdza zasady zaangażowania (*Rules of Engagement – ROE*) tych

jednostek (czy *Weapons Free* posiada chociaż jedna z nich) oraz czy są w stanie zadawać sobie straty (odległość mniejsza niż *Ground Combat*). Jeżeli te warunki nie zajdą jednostka nie walczy, algorytm kończy działanie.

Jeżeli jednostka będzie walczyła JTLS określa potencjał bojowy jednostki powodującej straty poprzez użycie tablicy *COMBAT INDEX*. Jest to czterowymiarowa tablica zawierająca następujące wymiary:

1. Efektywność rażenia jednostki niszczącej (*FIRE LETHALITY PROTOTYPE*).
2. Odporność na uderzenia jednostki niszczonej (*SURVIVABILITY PROTOTYPE*).
3. Rodzaj działań bojowych jednostki niszczącej.
4. Rodzaj działań bojowych jednostki niszczonej.



Rysunek 29. Algorytm przygotowania jednostki, w modelu, do starcia.

/źródło: opracowanie własne/

Uzyskanie niezerowej wartości z tablicy *COMBAT INDEX* jest warunkiem obliczenia strat. W tym przypadku jednostka ponosząca straty ma ustawiany atrybut *unit-in-*

combat, a JTLS powiadamia o tym ćwiczącego. Straty poniesione przez jednostkę zostaną obliczone w następnym czasie obliczania strat (*Assess Combat Time*).

Tablica *COMBAT INDEX* (patrz rysunek 30), dla określonej pary walczących jednostek (jednostka niszcząca FLP:1, jednostka niszczone SP:3), uzależnia liczbę całkowitą, określającą tablicę współczynników efektywności równań Lanchestera (FWL), od rodzaju działań bojowych jednostek. Jeśli tylko jest to możliwe ta sama tablica współczynników efektywności równań Lanchestera powinna opisywać więcej niż jeden zbiór warunków walki. W przykładowej tabeli wystarczyły dwie tablice współczynników FWL (numer 1 i 2). Zero w tablicy *COMBAT INDEX* wskazuje, że w określonych warunkach walki ta jednostka nie będzie powodowała strat – sama może je ponosić.

COMBAT INDEX								
Killer FLP: 1								
Victim SP: 3								
KILLER POSTURE	VICTIM POSTURE							
	Attack	Defend	Delay	Wdraw	Moving	Air Ops	Hasty Def	Incap
Attack	1	2	2	2	2	2	1	1
Defend	2	2	0	0	0	0	2	0
Delay	1	1	0	0	0	0	1	0
Withdraw	1	0	0	0	0	0	0	0
Moving	0	0	0	0	0	0	0	0
Air Ops	0	0	0	0	0	0	0	0
Hasty Def	1	2	2	0	0	0	1	0
Incapable	0	0	0	0	0	0	0	0
Amphib	1	2	2	2	1	2	1	0
Formation	0	0	0	0	0	0	0	0

Rysunek 30. Przykładowa tablica Combat Index.

/źródło: JTLS analyst's guide v.2.2/

Współczynniki efektywności równań Lanchestera, w przykładzie dla *COMBAT INDEX=1* (rysunek 31), określają efektywność z jaką systemy walki jednostki niszczącej oddziałują na systemy walki jednostki niszczonej. Zero oznacza, że dany system walki<sup>10</sup> nie powoduje zniszczenia określonego systemu walki przeciwnika.

<sup>10</sup> Systemy walki JTLS zostały opisane w podrozdziale 3.5.1.

FWL COEFFICIENTS for COMBAT INDEX #1								
KILLER COMBAT SYSTEM	VICTIM COMBAT SYSTEM							
	1	2	3	4	5	6	7	8
1	0.01	0.005	0.007	0.006	0.001	0.0	0.002	0.0
2	0.01	0.003	0.0	0.002	0.002	0.0	0.0	0.0
3	0.0	0.003	0.01	0.003	0.0	0.0	0.003	0.0
4	0.000 1	0.01	0.011	0.004	0.001	0.0	0.002	0.0
5	0.001 0	0.000 2	0.0001	0.0002	0.0003	0.0	0.0002	0.0
6	0.0	0.004 2	0.0003	0.0004	0.0045	0.0	0.0003	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Rysunek 31. Przykładowa tablica współczynników efektywności - FWL.

/źródło: JTLS analyst's guide v.2.2/

Wartości współczynników efektywności równań Lanchestera są danymi wejściowymi. JTLS nie może być używany do obliczania efektywności z jaką jeden system walki niszczy inny. Dla czasu walki wynoszącego 1 godzinę, przy dobrej pogodzie w dzień, współczynnik efektywności równań Lanchestera dla jednostki atakującej jednostkę broniącą, dla ognia bezpośredniego wynosi od 0,0001 do 0,1. Dla ognia pośredniego, w tych samych warunkach, współczynnik wyniesie od 0,0000001 do 0,00001. Wartości spoza tego zakresu mogą spowodować błędy.

Wartości współczynników efektywności równań Lanchestera, w armii USA, są pozyskiwane z:

- US Army Concepts Analysis Agency in Bethesda, Maryland;
- US Army War College in Carlisle Barracks, Pennsylvania.

Poza stratami modelowanymi modelem Lanchestera oraz spowodowanymi przez użycie min, straty w JTLS są powodowane użyciem broni niszczącej obiekty (*Targetable Weapons*).

Broń niszcząca obiekty jest używana przez:

- systemy obrony przeciwlotniczej do zniszczenia misji powietrznych przeciwnika oraz pocisków samosterujących dalekiego zasięgu;
- misje powietrzne do niszczenia innych misji powietrznych i do atakowania naziemnych celów, zarówno jednostek, celów, transportów i samolotów na ziemi;
- artylerię do zrealizowania zadania ogniowego;

- jednostki lądowe zdolne wystrzelić pociski dalekiego zasięgu;
- okręty do pojedynków na pociski i torpedy.

Generalnie broń niszcząca obiekt może mieć kilka różnych potencjałów zniszczeń. Może być precyzyjnie sterowane lub nie. Może powodować zniszczenia punktowe lub obszarowe.

Jak już wspomniano przy opisie modeli użytych w JTLS w działaniach lotniczych oraz działaniach ziemia-powietrze używa się do obliczenia rezultatów potyczki zmiennych losowych: pK, pH i pE. Rezultaty walki są stochastyczne.

Metodologia obliczania strat użyta do obliczenia rezultatów działań pocisków powietrze-ziemia, ziemia-ziemia i artylerii zależy od tego czy jest to broń precyzyjna czy niszcząca obszar (atrybut broni: *POINT* lub *AREA WEAPON*).

Jeżeli atrybut określa, że jest to broń punktowa, do obliczenia rezultatów potyczki używa się zmiennych losowych: pK, pH, a rezultaty są stochastyczne.

Jeżeli jest to broń niszcząca obszar jest użyta metoda oczekiwanej wartości (gdzie indziej określono ją jako metodę śmiertelności obszaru – *LETHAL AREA*).

Dla systemów walki, transportów, misji lotniczych i tych celów dla których wartość zniszczeń jest liczbą całkowitą, takich jak AAA<sup>11</sup>, SAM<sup>12</sup>, SSM<sup>13</sup> i MHE<sup>14</sup>, używa się współczynników wyrażonych prawdopodobieństwem zabicia – pK dla uniknięcia problemów związanych z rezultatami ułamkowymi. Dla obiektów zaopatrzenia użyta jest metoda oczekiwanych wartości strat.

Proces użycia broni niszczącej obiekty składa się z określenia wymagań, dostarczenia zaopatrzenia, dostarczenia broni i obliczenia zniszczeń. Obliczanie rezultatów użycia broni niszczącej obiekty przeciwko celom powierzchniowym składa się z trzech kroków:

- określenia punktu uderzenia;
- określenia jakie obiekty zostały zaatakowane;
- określenia jakie były rezultaty.

Określenie punktu uderzenia zależy od punktu wystrzelenia i charakterystyki broni, szczególnie od tego czy jest to broń precyzyjnie kierowana oraz czy broń jest punktowa czy niszcząca obszar.

---

<sup>11</sup> *AntiAircraft Artillery.*

<sup>12</sup> *Surface-to-Air Missile.*

<sup>13</sup> *Surface-to-Surface Missile.*

<sup>14</sup> *Material Handling Equipment.*

Ogień artylerii może być prowadzony na określony cel, jednostkę lub zbiór współrzędnych. Zadanie dotyczące HRU (*High Resolution Units*) zawsze wymaga określenia obcego obiektu, może to być ARU (*Aggregate Resolution Units*), inna HRU lub obiekt.

### 3.5.1. Systemy walki

Następujące reguły dotyczą systemów walki w JTLS:

1. System walki (*combat system*) może być systemem ognia pośredniego, bezpośredniego lub systemem nie powodującym straty. Jeden system może być systemem tylko jednego typu.
2. Systemy ognia bezpośredniego używają algorytmów ognia celowanego (*Aimed-Fire*) i mogą zniszczyć cały system walki przeciwnika. Systemy ognia pośredniego używają algorytmów ognia niszczącego obszar (*Area Fire*), które wymagają współczynnika zadawanych strat jako część obliczeń. Wszystkie systemy przeciwnika są traktowane jako potencjalne ofiary i każdy z nich może zostać zniszczony. System nie powodujący strat nie niszczy inne systemy, ale sam może zostać zniszczony.
3. System walki może niszczyć system przeciwnika tylko wtedy gdy jest w jego zasięgu. Zasięg jest obliczany jako geograficzna odległość pomiędzy środkami jednostek. Każdy system walki, którego maksymalny efektywny zasięg jest większy lub równy odległości pomiędzy jednostkami, może powodować zniszczenia. Każdy typ systemu walki ma określony maksymalny zasięg.
4. Jednostki, którym brakuje amunicji mogą mieć niektóre systemy walki ustawione jako nieskuteczne. Współczynnik ten jest określony przez *CSP CS SUPPLY CATEGORY TO FIGHT*.
5. Jednostki, które nie mają paliwa mogą mieć niektóre z systemów walki o zmniejszonej skuteczności. Współczynnik ten jest określony przez *CSP CS NO FUEL EFFICIENCY*.
6. Jednostka walcząca z inną jednostką może używać tylko tych systemów walki, które są skierowany w stronę tej jednostki.
7. Żadna jednostka nie może stracić więcej systemów walki niż ma sprawnych. Broń w naprawie nie może być niszczone.
8. Systemy walki należące do jednostek wsparcia bezpośredniego nie są niszczone w walce wspieranej jednostki.
9. Jeżeli niszczona jednostka jest w zasięgu *DECISIVELY ENGAGED DISTANCE* z jednostką niszczącą ponosi straty jakby była w postawie ATAK.

10. Niszczona jednostka przeprowadzająca aktywne poszukiwania (*Active Search*) w postawie OBRONA lub NIEZDOLNA ponosi takie straty jakby była w postawie POSPIESZNA OBRONA.
11. Każda jednostka posiada parametr *UT EFFECTIVENESS MULTIPLIER* jako część danych inicjujących. Jeżeli ta liczba jest inna niż 1.0 jest używana jako mnożnik – może zwiększać lub zmniejszać wielkość strat.

### 3.6. Bazy danych

#### Organizacja zespołu przygotowania bazy danych

Budowa bazy danych dla symulacji jest procesem złożonym. Plan zarządzania wdrożeniem dla wspomnianego zadania powinien być wykonany ze względu na zapewnienie odzwierciedlenia wymagań sponsora wynikłych z procesu planowania i analizy.

Krytyczne znaczenie ma plan przygotowania bazy danych, który jasno precyzuje jakie są wymagania sponsora odnośnie zadań bazy danych, które powinna realizować.

Jasne sprecyzowanie wspomnianych celów jest szczególnie istotne jeżeli sponsor bazy danych oraz zespół rozwoju bazy danych należą do różnych organizacji. Wymagania sponsora precyzują sposób zbierania, agregacji oraz wprowadzania informacji do bazy danych. Cele postawione przez sponsora mogą służyć jako główny przewodnik przy podejmowaniu określonych założeń i ustaleń z wiązanych z określonymi wartościami niektórych danych, które pojawią się w trakcie procesu przygotowania bazy danych.

Plan zarządzania przygotowania bazy danych musi określać, kiedy ma nastąpić zakończenia przygotowania bazy danych oraz zawierać plan okresowych przeglądów wykonywanych przez wszystkich uczestniczących w procesie przygotowania bazy danych. Każdemu okresowemu przeglądowi (może za wyjątkiem pierwszego) powinien towarzyszyć test bazy danych. Idealnym rozwiązaniem byłoby zespołów przygotowania oraz użytkownika bazy danych razem (jak najbliżej siebie). W przypadku, gdy powyższe rozwiązanie nie jest możliwe to podniesienie koordynacji należy zapewnić poprzez reprezentantów zespołu przygotowania bazy danych ze strony użytkownika.

Stworzenie najlepszego zespołu do przygotowania bazy danych wymaga, aby składał się on z dużej liczby analityków funkcyjnych (kierunkowych), wywodzących się z oddzielnych zespołów analitycznych lub innych doświadczonych użytkowników korzystających z modelu (przykładowo personel J-3, J4, który model wykorzystuje). Chociaż programy do przygotowywania bazy danych zostały przygotowane głównie do wykorzystania przez personel operatorski, to jednak eksperci techniczni posiadający wiedzę jak model

funkcjonuje powinni także zostać włączeni. W ogólności konstrukcja bazy danych JTLS wymaga ekspertów funkcyjnych od terenu, operacji powietrznych, sił lądowych, logistyki oraz wywiadu. Każdy członek zespołu jest odpowiedzialny za określony obszar związany z wymaganiami czasu symulacji identyfikowanymi aż do czasu testowania bazy danych oraz dostarczenia jej użytkownikowi. W formowaniu zespołu ważną rzeczą jest włączenie ludzi doświadczonych w modelowaniu, symulacji oraz prowadzeniu gier ponieważ ich doświadczenie pozwala im spoglądać na wartości danych użytych w modelu w ich najczystszy sensie. Praca zespołu powinna zostać określona przez plan przygotowania bazy danych, który precyzuje kto jest za co odpowiedzialny oraz kiedy określone zadania powinny zostać zakończone. W celu uniknięcia błędów zaleca się zorganizowania wprowadzania danych jako procesu strukturalnego. Generalną zasadą jest aby jedna osoba była odpowiedzialna za wprowadzanie danych, dokładność danych może wzrosnąć jeżeli jest ona wspomagana przez eksperta funkcyjnego, któremu przedzielono dany zakres danych. Zaleca się aby zapewnienie jakości wprowadzania danych należało do zadań odpowiednich zespołów ekspertów funkcyjnych. Na zakończenie wprowadzone dane powinny uzyskać aprobatę Zespołu Zapewnienia Jakości składającego się ze ekspertów funkcjonalnych z Zespołu Przygotowania Bazy Danych oraz ekspertów operacyjnych należących do organizacji sponsora. W dokumentacji systemu JTLS przedstawiono, że wprowadzanie, zapewnienie jakości oraz testowanie bazy danych są procesem iteracyjnym.

Nie należy dokonywać zmian w składzie zespołu bazy danych w całym okresie przygotowania bazy danych. Najlepiej, aby ten sam zespół był wykorzystywany do rozwoju wielu baz danych (ze względu na czasochłonny i skondensowany wysiłek uczenia wymagany w tym złożonym procesie).

**Bazy danych w systemie JTLS (podobnie jak i programy) można podzielić na następujące grupy:** inicjalizacyjne, konfiguracyjne oraz główne (związane bezpośrednio z realizacją procesu symulacji). Niżej wymieniono wszystkie bazy danych w systemie JTLS (na poziomie ogólnym):

- *ASCII Initialization Data Files* Inicjujące zbiory danych w formacie ASCII;
- *Scenario Vocabulary Files* Zbiory słownikowe scenariusza;
- *Order Definition File* Zbiór definicji rozkazów;
- *Static Vocabulary File* Zbiór słownika statycznego;
- *Players Manuals* Podręczniki graczy (ćwiczącego);
- *Menu Definition File (GIAC)* Zbiór definicji menu programu GIAC;
- *GIAC Definition Files* Zbiór definicji (konfiguracji) programu GIAC;

- *Screen Definition File (IMT)*      Zbiór definicji (konfiguracji) programu IMT;
- *Message Definition File (MPP)*      Zbiór definicji wiadomości programu MPP;
- *Main Database Files*                      Główne zbiory danych.

### Lista klas w systemie JTLS

W głównej bazie danych systemu JTLS występują niżej wymienione klasy przedstawione poniżej (tabela 2). Klasy zostały opracowane na podstawie opisu samych atrybutów klas. Autorzy opracowania nie dysponowali dokumentacją (brak dokumentacji *Database Development User Guide*) opisującą same klasy.

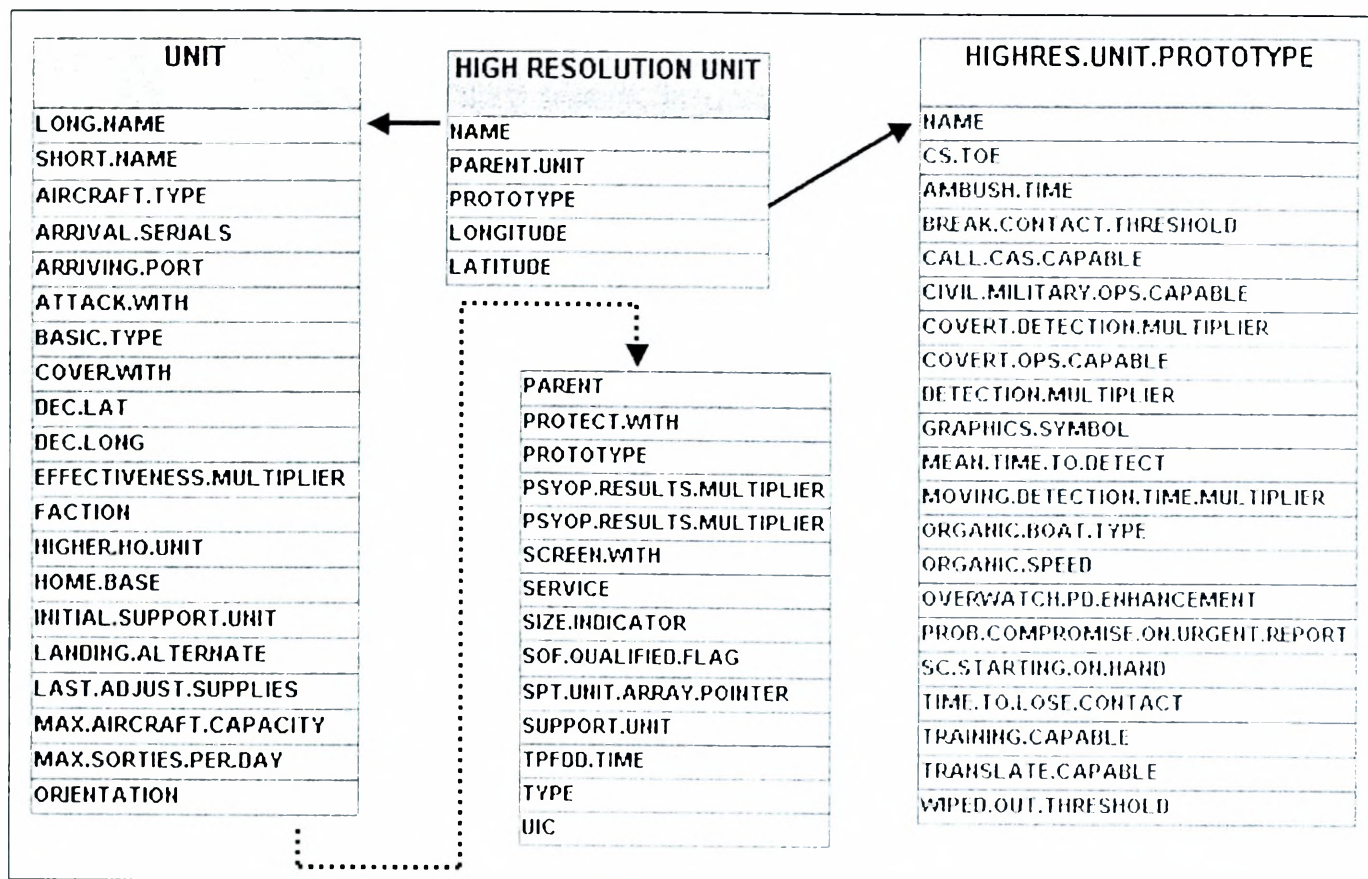
<i>AIR AIR LETHALITY</i>	Siła niszczenia powietrze-powietrze
<i>AIR CONTROL PROTOTYPE</i>	Wzorzec kierowania dla lotnictwa
<i>AIR DEFENSE</i>	Lotnictwo
<i>AIR DEFENSE CLASS, ALTITUDE ZONE</i>	Strefa wysokości dla lotnictwa
<i>AIR DEFENSE WEATHER</i>	Pogoda dla lotnictwa
<i>AIRCRAFT CLASS</i>	Samoloty
<i>AIRCRAFT SHELTER TYPE</i>	Typy schronów dla samolotów
<i>AIRCRAFT TARGET</i>	Samolot (jako cel, obiekt)
<i>ALTITUDE ZONE</i>	Strefa wysokości
<i>AREA KILL LETHALITY</i>	Siła niszczenia obszaru
<i>BARRIER TYPE</i>	Typy barier
<i>BRIDGE</i>	Most
<i>CALIBER</i>	Kaliber
<i>COMBAT SYSTEM PROTOTYPE</i>	Wzorzec dla systemu walki
<i>COMMAND CONTROL PROTOTYPE</i>	Wzorzec dla dowodzenia
<i>COMMUNICATION CENTER</i>	Węzeł łączności
<i>COMMUNICATIONS PROTOTYPE</i>	Wzorzec dla węzła łączności
<i>DEPTH ZONE</i>	Strefa głębokości
<i>FACTION COUNTRY</i>	Kraj frakcji
<i>FIRE LETHALITY PROTOTYPE</i>	Wzorzec dla siły niszczenia ognia
<i>FORCE SIDE</i>	Siły stron
<i>HIGH RESOLUTION UNIT</i>	Pododdziały HRU
<i>HIGHRES.UNIT.PROTOYPE</i>	Wzorzec dla pododdziałów HRU
<i>HUP TARGET TYPE</i>	Typy środków dla pododdziałów HRU
<i>INTEL INFO PROTOTYPE</i>	Wzorzec dla rozpoznania
<i>INTERDICTION POINT TYPE</i>	Typy punktów kontroli
<i>JAMMER TYPE</i>	Typy urządzeń zakłócających
<i>LOGIN</i>	Plan zaopatrywania
<i>MANEUVER PROTOTYPE</i>	Wzorzec możliwości manewrowych
<i>MINEFIELD.TYPE</i>	Typy pól minowych
<i>MOBILITY PROTOTYPE</i>	Wzorzec dla mobilności (przejezdności)
<i>NATIONAL BOUNDARY</i>	Granice państw
<i>PIPELINE</i>	Rurociąg
<i>PIPELINE ARC</i>	Połączenia rurociągów
<i>PIPELINE NODE</i>	Węzły rurociągów

<i>POINT KILL LETHALITY</i>	Siła niszczenia broni punktowego rażenia
<i>PUMPING STATION</i>	Stacje tankowania
<i>RAIL ARC</i>	Połączenia kolejowe
<i>RAIL NODE</i>	Węzły kolejowe
<i>RAILROAD</i>	Linie kolejowe
<i>SENSOR TYPE</i>	Typy sensorów
<i>SHIP UNIT PROTOTYPE</i>	Wzorzec dla jednostek floty
<i>SMALL BOAT</i>	Małe łodzie
<i>SUPPLY CATEGORY</i>	Kategoria zaopatrzenia
<i>SUPPLY MOVEMENT ASSET</i>	Środki transportu dla zaopatrzenia
<i>SUPPLY STORAGE AREA</i>	Obszar składowania dla zaopatrzenia
<i>SUPPLY TYPE</i>	Typy zaopatrzenia
<i>SURFACE AIR LETHALITY</i>	Siła niszczenia ziemia-powietrze
<i>SURFACE SURFACE MISSILE</i>	Pocisk ziemia ziemia
<i>SURVIVABILITY PROTOTYPE</i>	Wzorzec dla przetrwania
<i>SURVIVABILITY PROTOTYPE</i>	Wzorzec dla przetrwania
<i>TARGET</i>	Cel. Środki inżynieryjne i przeciwlotnicze przydzielone jednostce
<i>TARGETABLE WEAPON</i>	Środki niszczenia
<i>TERRAIN TYPE</i>	Typy terenów
<i>TRANSPORTATION</i>	Środki transportu
<i>TUNNEL CLASS</i>	Tunel
<i>UNIT</i>	Jednostka
<i>WEATHER CONDITION</i>	Warunki pogodowe
<i>WEATHER FRONT</i>	Fronty pogodowe

Tabela 2. Klasy baz danych.

### Przykład powiązań pomiędzy głównymi klasami danych

Klasa HRU jest powiązana przez atrybut NAME z klasą jednostki UNIT (z której może zostać wydzielona). Parametry wspólne dla wielu jednostek HRU nie są umieszczone bezpośrednio w definicji danej jednostki tylko w tzw. wzorcu. W naszym przykładzie *HIGHRES.UNIT.PROTOTYPE*. Rozwiązanie powyższe umożliwia szybszą i bardziej niezawodną aktualizację danych – potrzeba modyfikacji danych wspólnych istnieje tylko w jednym miejscu – we wzorcu.

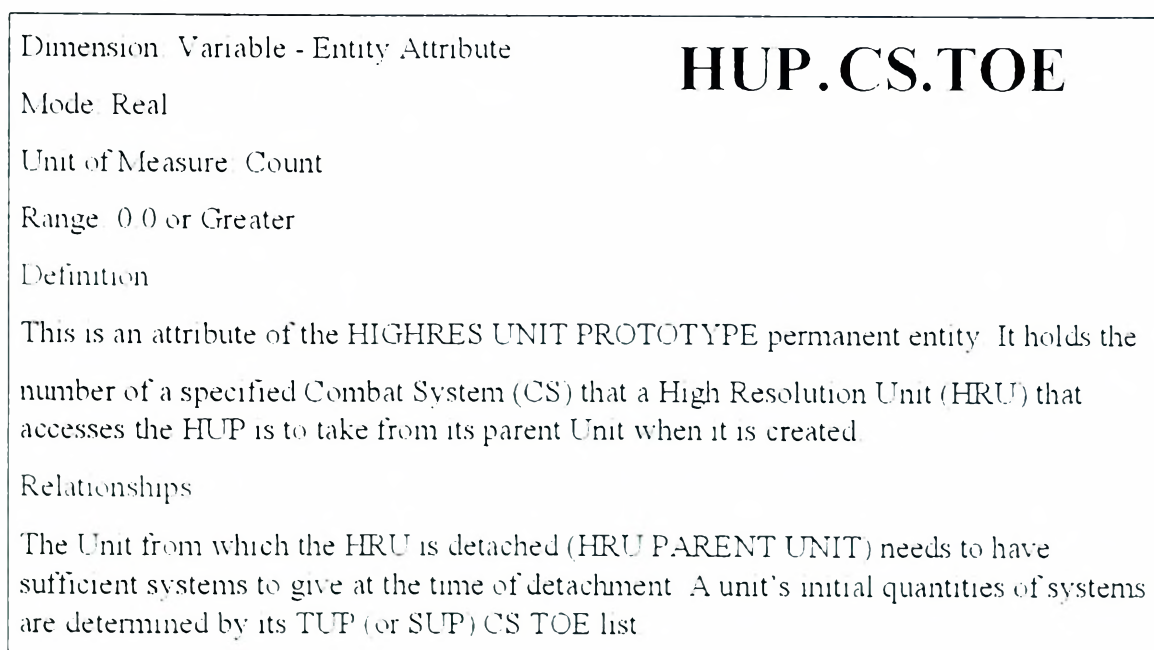


Rysunek 32. Powiązania między klasami HRU, Unit, a HIGHRES.UNIT.PROTOTYPE.

/źródło: opracowanie własne/

Przykład definicji (opisu) atrybutu w systemie JTLS

Definicja atrybutu w dokumentacji systemu JTLS składa się z następujących elementów: rozmiaru (*dimension*), typu danych (*mode*), jednostki miary (*unit of measure*), zakresu (*range*), znaczenia (*definition*) oraz powiązań z innymi atrybutami (*relationships*).



Rysunek 33. Definicja atrybutu HUP.CS.TOE.

/źródło: JTLS data requirements manual v.2.2/

## 4. PRZYGOTOWANIE SYSTEMU JTLS DO PRACY

### 4.1. Stanowiska w trakcie ćwiczenia

Wszystkie warianty wykorzystania JTLS wymagają pięć kluczowych stanowisk:

- 1) **Dyrektora ćwiczenia** (*Exercise Director*);
- 2) **Kierownika ćwiczenia** (*Senior Controller/Exercise Controller*);
- 3) **Koordynatora technicznego** (*Technical Coordinator*);
- 4) **Technika systemu** (*Computer Systems Manager*);
- 5) **Ćwiczących** (*Players*).

Ad. 1) **Dyrektor ćwiczenia** planuje i zarządza grą.

Ad. 2) **Kierownik ćwiczenia** monitoruje postęp działań. Używa narzędzi modelu aby kształtować komputerową walkę tak, aby spełnić wymagania operacji lub cele ćwiczenia określone przez dyrektora ćwiczenia.

Ad. 3) **Koordynator techniczny** uruchamia i zatrzymuje grę, monitoruje wszystkie składniki systemu komputerowego niezbędne do symulacji oraz dostarcza wsparcia technicznego dla symulacji.

Ad. 4) **Technik systemowy** konfiguruje komputery i koordynuje zmiany systemu oprogramowania; opiekuje się sprzętem.

Ad. 5) **Ćwiczący** wprowadzają rozkazy i śledzą status (stan) przydzielonych sił. Mogą rozkazywać całym siłom lub mogą mieć określone komu wolno im rozkazywać lub funkcji jakie wolno im spełniać.

W JTLS jest sześć podstawowych typów ćwiczących. Powszechnie używane typy to:

- a) **Dowódca**, który może wykonywać wszystkie funkcje gracza.
- b) **Ćwiczący lądowy**, który może wydawać tylko polecenia niezbędne do kierowania siłami lądowymi.
- c) **Ćwiczący lotniczy**, który może wydawać tylko polecenia misjom lotniczym.
- d) **Ćwiczący morski**, który ma dostęp tylko do rozkazów niezbędnych do kierowania okrętami i operacjami morsko-lądowymi modelowanymi w JTLS.
- e) **Ćwiczący wywiadu** wydający rozkazy zbierania danych wywiadowczych, przetwarzania zebranych informacji przez wbudowane organy i przesyłania informacji do zainteresowanych graczy.

- f) **Ćwiczący logistyczny**, który jest ograniczony do ustalenia poziomu zapasów, kierowania operacjami zaopatrzenia, kierowania konwojami i sieciami zaopatrzenia.

JTLS dostarcza sztabowi planującemu ćwiczenia możliwość stworzenia innych typów Ćwiczących będących połączeniem kilku lub wszystkich poniższych funkcji. Dokonuje się to poprzez tworzenie plików definiujących menu rozkazów specyficznego użytkownika dopasowanych do wymagań ćwiczenia. Np. szczególne ćwiczenia mogą wymagać użytkownika, który ma dostęp do zarówno niektórych rozkazów lotniczych jak i logistycznych.

#### 4.2. Koordynator techniczny

W JTLS koordynator techniczny (*Technical Coordinator*):

- uruchamia symulacje;
- tworzy punkty testowe (*checkpoints*);
- zatrzymuje symulacje;
- przeprowadza ponowne uruchomienia symulacji z wcześniej utworzonych punktów testowych.

Poniżej omówiono postępowanie koordynatora technicznego oraz programy:

- *Scenario Initialization Program* (SIP) tworzący scenariusz;
- *Interface Configuration Program* (ICP) definiujący konfigurację ćwiczącego;
- *Executive Program* wykorzystywany do uruchomienia symulacji.

Jeżeli koordynator techniczny uruchamia scenariusz pierwszy raz musi:

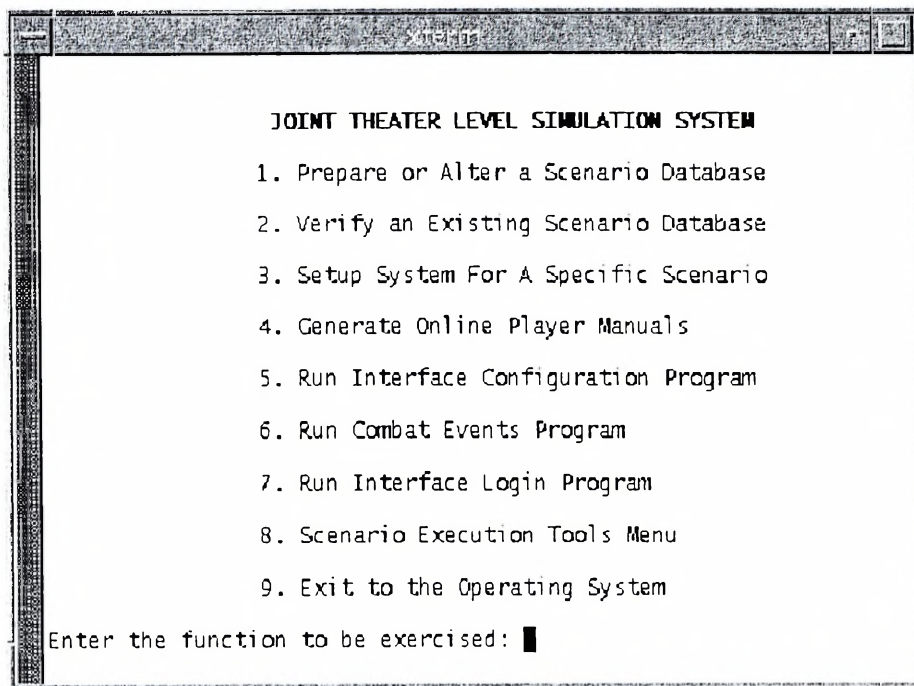
1. uruchomić program SIP (*Scenario Initialization Program*);
2. skonfigurować otoczenie ćwiczącego oraz
3. wygenerować zbiory OPM (*Online Player Manual*).

**Powyższe działania muszą być powtórzone za każdym razem, gdy zmienia się baza danych scenariusza.** Jeżeli zostały już wcześniej wykonane, a baza danych scenariusza nie zmieniła się, koordynator może uruchomić grę.

#### Inicjowanie scenariusza

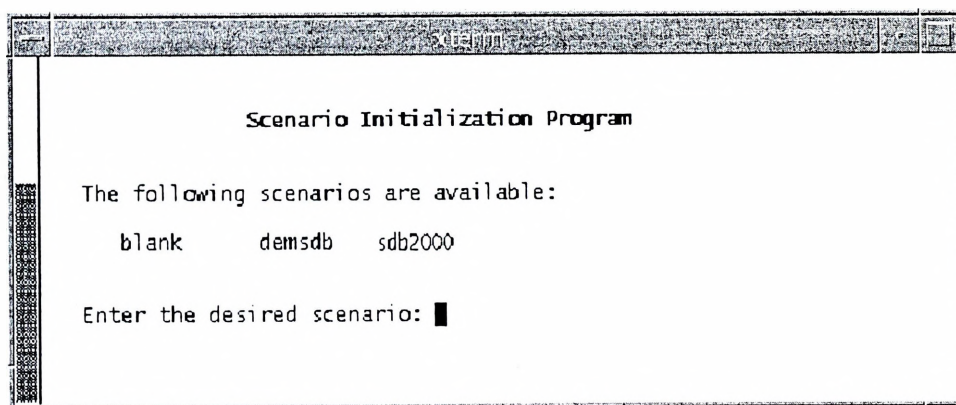
Kiedy zostanie zbudowana i zatwierdzona baza danych może zostać użyta do uruchomienia symulacji. Pierwszym krokiem przy użyciu nowej lub zmodyfikowanej bazy danych jest uruchomienie *Scenario Initialization Program* (SIP). Program ten tworzy

niezbędną strukturę katalogów oraz pliki rozkazów używane przez GIAC, a także kopiuje wymagane pliki scenariusza i map do katalogu symulacji. SIP jest uruchamiany z głównego menu *Executive Program* poprzez wybranie opcji 3 (rysunek 34).



Rysunek 34. Główne menu Executive Program.

Po wybraniu tej opcji *Executive Program* ustala nazwę scenariusza (rysunek 35).



Rysunek 35. Wybór scenariusza.

Załóżmy, że koordynator techniczny wybrał scenariusz o nazwie **demsdb**. Po wyborze scenariusza *Executive Program* sprawdza strukturę katalogów (rysunek 36).

```

$JGAME Directory
|   |-- demsdb Directory
|   |   |-- ato Directory
|   |   |-- checkpoint Directory
|   |   |-- config Directory
|   |   |-- debug Directory
|   |   |-- group Directory
|   |   |-- location Directory
|   |   |-- onlineman Directory
|   |   |-- user Directory

```

**Rysunek 36. Wymagana struktura katalogów.**

/źródło: JTLS analyst's guide v.2.2/

*Executive Program* sprawdza czy istnieje katalog \$JGAME/demsdb oraz jego podkatalogi. W tych katalogach umieszczane są pliki wyjściowe symulacji. Jeżeli brak jakiegoś katalogu jest on tworzony poprzez SIP:

- \$JGAME/demsdb/ato jest używany przez *ATO Translator* (ATOT) do zapisywania danych wyjściowych i przechowywania danych;
- \$JGAME/demsdb/checkpoint jest używany przez wszystkie programy JTLS do przechowywania danych związanych z punktami kontrolnymi. Po każdym rozkazie punktu kontrolnego CEP tworzony jest unikalny podkatalog w katalogu \$JGAME/demsdb/checkpoint;
- \$JGAME/demsdb/config jest wypełniany przez *Interface Configuration Program* (ICP) plikami indywidualnej konfiguracji dla wszystkich programów ćwiczącego;
- \$JGAME/demsdb/debug jest używany przez wszystkie programy JTLS do przechowywania plików wyjściowych debugowania;
- \$JGAME/demsdb/group zawiera niezbędne pliki do inicjalizacji kolejności w GIAC oraz jest używany przez stacje GIAC do przechowywania współdzielonych danych;
- \$JGAME/demsdb/location jest używany przez CEP do okresowego przechowywania położenia sił;
- \$JGAME/demsdb/onlineman jest używany przez *Online Player Manuals* (OPM) do przechowywania podręczników ćwiczącego;
- \$JGAME/demsdb/user jest używany przez stacje GIAC do przechowywania prywatnych danych i plików rozkazów użytkownika.

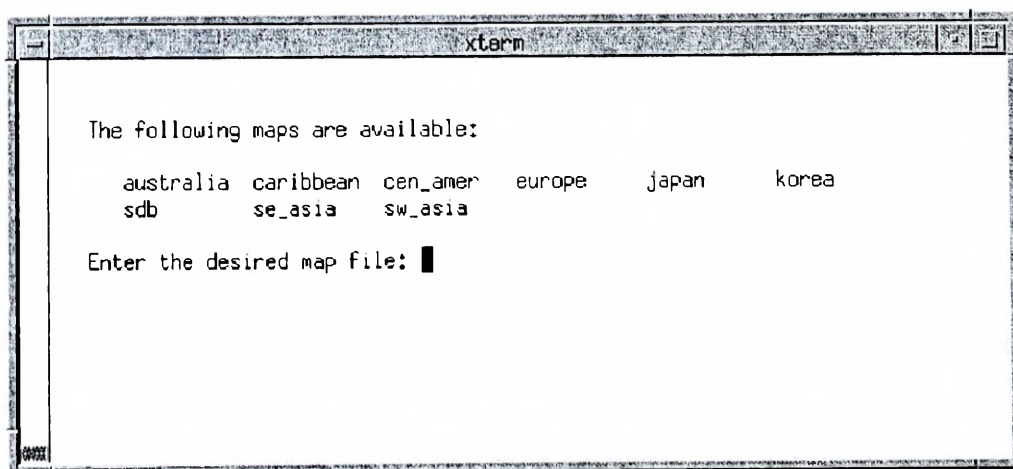
Pliki wejściowe JTLS muszą być umieszczone w odpowiednich katalogach. Program SIP sprawdza obecność plików scenariusza w odpowiednich katalogach. Jeżeli tam są koordynator techniczny jest pytany czy nowy scenariusz ma być skopiowany z katalogu \$JDATA/scenario/demsdb do katalogu symulacji. Jeżeli odpowiedź brzmi TAK pliki są kopiowane do katalogu \$JGAME/demsdb/checkpoint/0000 z zachowaniem swojej nazwy. **Program CEP szuka plików danych w katalogu \$JGAME, a nie w \$JDATA.**

Jak tylko program SIP skopiuje pliki scenariusza do katalogu \$JGAME/demsdb/checkpoint/0000 oraz stworzy wszystkie pliki interfejsu strony sprawdzane jest czy każda strona w grze posiada katalog ze swoją nazwą strony w katalogu \$JGAME/\$scenario\_name/group. Jeżeli pliki te są dostępne i została wybrana funkcja setupu, SIP zapyta koordynatora technicznego czy mają być utworzone nowe pliki *GIAC Order Definition*.

Jeżeli scenariusz jest nowy lub zmieniła się baza danych scenariusza pliki *GIAC Order Definition* muszą być wygenerowane. Pliki *Order Definition* określają jakie dane wejściowe są dozwolone dla każdego pola rozkazu na stacji GIAC. Dzięki plikom *Order Definition*, GIAC zna m.in. nazwy wszelkiej broni w scenariuszu i pozwala ćwiczącemu na wybór z listy dostępnych nazw.

Jeżeli koordynator techniczny wybierze generowanie plików *Order Definition*, SIP uruchamia *Order Pre-Processor* (OPP), aby wykonać to zadanie. Koordynator techniczny jest informowany o postępie tworzenia pliku w trakcie tej operacji. Dla dużych scenariuszy proces generowanie tych plików może trwać ponad 30 minut (w zależności od mocy procesora komputera).

Jak pliki *Order Definition* zostaną wygenerowane lub jeżeli koordynator techniczny nie polecił ich generować, wybiera mapę dla scenariusza (rysunek 37).



Rysunek 37. Wybór mapy dla scenariusza.

Po wyborze pliku mapy, przez koordynatora technicznego, SIP kopiuje niezbędną mapę do katalogu \$JGAME/demsdb/config. Po wykonaniu tego koordynator techniczny powraca do Executive Program.

### **Tworzenie podręczników ćwiczących**

Po zakończeniu pracy programu SIP, koordynator techniczny może wygenerować podręczniki ćwiczących poprzez wybranie 4 opcji z głównego menu JTLS. Generowanie tych plików nie jest konieczne. Jednakże jeżeli koordynator techniczny nie wygeneruje ich, to podręczniki nie będą dostępne dla ćwiczących lub będą zawierały nieaktualne dane. Zaleca się aby podręczniki ćwiczących były generowane, dla nowego scenariusza lub gdy nastąpiły zmiany bazy danych scenariusza.

Podręczniki ćwiczących powinny być powtórnie generowane z katalogów punktów kontrolnych jeżeli model jest uruchamiany z punktu kontrolnego. To zapewni aktualność danych.

Bez względu na to, czy podręczniki ćwiczących są generowane z inicjujących danych (z plików z katalogu ~/checkpoint/0000), czy z danych punktu kontrolnego (z plików z katalogu checkpoint/00nn), Executive Program będzie pytał koordynatora technicznego o nazwę scenariusza. Dla dużych scenariuszy proces generowania podręczników ćwiczących może trwać ponad 30 minut. Koordynator techniczny będzie otrzymywał wiadomości o generowaniu plików dla kolejnej strony. Kiedy pliki zostaną wygenerowane koordynator techniczny powraca do *Executive Program*.

### **Konfiguracja interfejsu**

Program *Interface Configuration* (ICP) jest używany do zdefiniowania i skonfigurowania stacji roboczej ćwiczącego oraz sieci komputerowej. **Program ICP musi być uruchomiony dla nowego scenariusza.** Koordynator techniczny powinien także uruchomić ICP jeżeli chce zmienić konfigurację związaną ze scenariuszem. Jeżeli żaden z tych przypadków nie zachodzi nie uruchamia się ICP.

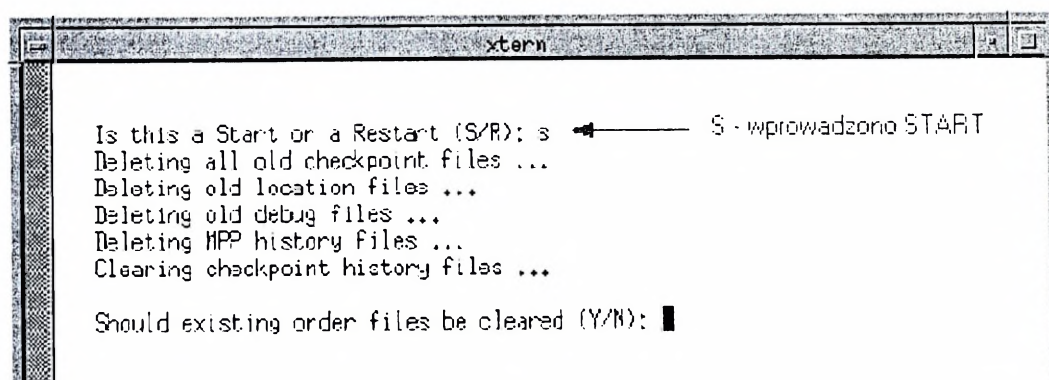
### **4.3. Start symulacji i jej powtórne uruchomienie**

Jak opisano powyżej, jeżeli scenariusz jest uruchomiony pierwszy raz lub gdy w bazie danych scenariusza wystąpiły zmiany, koordynator techniczny musi przed rozpoczęciem symulacji uruchomić program SIP, wygenerować zbiory OPM, oraz utworzyć konfigurację dla gracza. Jeżeli scenariusz był już raz uruchomiony, a baza danych nie zmieniła się, to koordynator może pominąć te czynności i uruchomić grę.

Symulacja w systemie JTLS jest uruchamiana przez wybór polecenia nr 6 w menu głównym „*Run Combat Events Program*” – uruchomienie głównego modułu CEP. Po wyborze tej opcji zostaje wyświetlona lista dostępnych scenariuszy ćwiczeń. Jeżeli wymagany scenariusz nie występuje na liście, oznacza to, że dla tego scenariusza nie uruchomiono programu SIP. W tym przypadku koordynator techniczny musi najpierw uruchomić program SIP dla danego scenariusza.

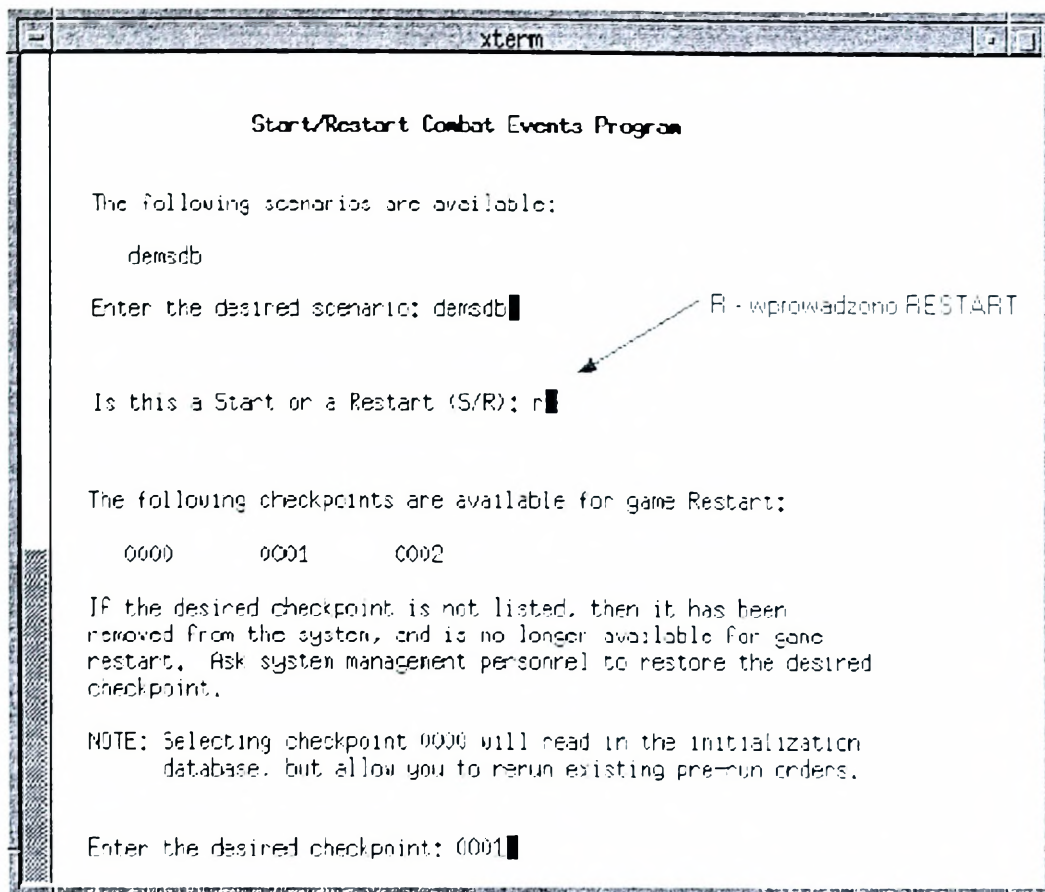
Po dokonaniu wyboru scenariusza (w naszym przykładzie o nazwie „demsdb”) program wyświetli pytanie jaki typ symulacji należy uruchomić:

1. Jeżeli wprowadzono „S” (od słowa *Start*), to zbiory punktów kontrolnych oraz związane z nimi zbiory danych z poprzedniego ćwiczenia zostaną usunięte. W przykładzie wybór opcji „S” spowoduje usunięcie wszystkich punktów kontrolnych – poprzednie zapisy plików oraz pliki debugera zostaną usunięte. Oczywiście inicjalizacyjne pliki bazy danych w katalogu ~/checkpoint/0000 nie zostaną usunięte (rysunek 38).



Rysunek 38. Polecenie uruchomienia symulacji.

2. Jeżeli wprowadzono „R” (od słowa *Restart*), to wszystkie pliki punktów kontrolnych oraz związane z nimi pliki danych z poprzednich gier zostaną, a koordynator techniczny musi wprowadzić numer punktu kontrolnego, od którego ćwiczenie ma być wznowione (rysunek 39).



Rysunek 39. Polecenie powtórnego uruchomienia symulacji.

Niezależnie od wyboru, żadne dane nie zostaną automatycznie usunięte z katalogów \$JGAME/demsdb/group oraz \$JGAME/demsdb/user za wyjątkiem pliku MPP.prew w katalogu \$JGAME/demsdb/group. Te katalogi zawierają dane związane z programem GIAC, współdzielone i prywatne dane użytkowników oraz archiwalne pliki rozkazów.

#### 4.4. Punkty kontrolne

Punkty kontrolne ASCII tworzą „zdjęcie” statusu CEP oraz zapamiętują ważne dane GIAC i MPP, które są wymagane do powtórnego uruchomienia programu CEP, oraz programów: GIAC i MPP, a także powtórnego wygenerowania podręczników ćwiczących.

Jedynie kierownik ćwiczenia może wykonywać punkty kontrolne. Kiedy jest przetwarzany punkt kontrolny, zegar symulacji zatrzymuje się, a wszystkie dane wyjściowe są zatrzymane. CEP tworzy podkatalog w katalogu \$JGAME/demsdb/checkpoint z czteropozycyjnym unikalnym numerem (0001 dla pierwszego punktu kontrolnego tworzonego dla tego scenariusza). Wszystkie dane punktu kontrolnego zostaną zapisane w tym katalogu. Następne punkty kontrolne zostaną nazwane kolejnymi numerami (po numerze 0001 zostanie nadany numer 0002 itd.). CEP informuje pierwotnego GENIS, że punkt

kontrolny jest wykonywany i zaczyna zapisywać swój bieżący stan. Primary GENIS przesyła zawiadomienie do wszystkich podwładnych procesów GENIS oraz wszystkich MPP, które są powiązane. Każdy GENIS powtarza tą procedurę. Kiedy wszystkie procesy MPP otrzymają zawiadomienie o punkcie kontrolnym zaczynają zapisywać swój bieżący stan.

Gdy następuje punkt kontrolny, procedury są automatyczne i nie wymagają interwencji żadnego ćwiczącego. Ćwiczący prawdopodobnie nie zauważą, że nastąpił punkt kontrolny dopóki nie wyślą rozkazów do CEP lub nie złączą aktywnie pracować ze swoim MPP. Jeżeli ćwiczący wysyłają rozkazy do CEP zauważą, że ich rozkazy nie są dostarczane i nie będą dostarczane jak długo CEP zapisuje swoje dane. Dodatkowo dopóki MPP zapisuje swoje ważne dane może nie reagować na akcje ćwiczącego przez kilka sekund.

W rozkazie „Punkt kontrolny” kierownik ćwiczenia decyduje, czy CEP kontynuuje działanie po punkcie kontrolnym (tzw. eksploatacyjny punkt kontrolny - *Running Checkpoint*), czy też symulacja zatrzymuje się kiedy punkt kontrolny się wykona (tzw. zatrzymujący punkt kontrolny - *Stop Checkpoint*). Jeżeli punkt kontrolny zatrzymuje symulację, proces CEP staje się po zapisaniu ważnych danych punktu kontrolnego, w przeciwnym przypadku model działa dalej.

### **Punkty kontrolne ASCII**

Punkt kontrolny ASCII jest zbiorem plików ASCII zawierających wszystkie znaczące informacje o grze. Ten typ punktu kontrolnego (ASCII) ma następujące zalety:

1. Może być przeanalizowany przez analityka do wyegzekwowania danych użytych w grze.
2. Może być zmodyfikowany przez analityka aby usunąć dane, które mogą przynosić błędy (upadki systemu).
3. Może być wczytany przez CEP.

Koordynator techniczny decyduje czy ma być wygenerowany poprzez CEP punkt kontrolny ASCII, gdy pojawią się błędy w trakcie symulacji lub gdy wymagane jest zatrzymanie symulacji.

Punkt kontrolny o numerze 0000 zawsze odsyła do inicjujących plików baz danych w katalogu \$JGAME/scenario\_name/checkpoint/0000 i jest błędem wybranie go do regeneracji podręczników ćwiczącego. Aby wygenerować podręczniki ćwiczącego z punktu kontrolnego 0000 koordynator techniczny musi wybrać opcję 4 w *Execution Program*.

#### 4.5. Zatrzymywanie symulacji

System JTLS umożliwia proces składowania danych oraz restart systemu. Proces zatrzymywania oraz odtwarzania jest możliwy z dowolnego punktu trwania symulacji dla którego wykonano punkt kontrolny.

Program CEP jest zatrzymywany na rozkaz koordynatora technicznego. W celu zatrzymania symulacji w czasie trwania ćwiczenia jest wprowadzenie rozkazu *CHECKPOINT* z zaznaczoną opcją „*Stop Checkpoint*”. Wspomniana opcja jest zabezpieczona hasłem. Po wprowadzeniu polecenia „STOP” wykonywany jest punkt kontrolny końcowy.

Alternatywną metodą zatrzymywania symulacji jest wykorzystanie rozkazu *ENTER DEBUG*. Wprowadzenie tego rozkazu powoduje przejście programu CEP w tryb debuggera. W chwili otrzymania przez CEP wspomnianego rozkazu program CEP wywołuje procedurę *SIMDEBUG.R* języka *SIMSCRIPT*, która uruchamia debbuger oraz wyświetla znak zachęty do pisania (*prompt*) na ekranie CEP. Rozkaz *ENTER DEBUG* powinien być wprowadzany tylko przez doświadczonego koordynatora technicznego lub Programistę. Wprowadzenie po znaku zachęty do pisania *SIMDEBUG* litery „Q” lub „q” (od słowa quit) spowoduje zatrzymanie programu CEP. Zatrzymanie w ten sposób programu CEP nie spowoduje wygenerowania końcowego punktu kontrolnego. W celu wznowienia pracy programu CEP należy w linii *SIMDEBUG* wprowadzić literę „C”.

#### Składowanie danych

Po zakończeniu sesji JTLS, koordynator techniczny jest odpowiedzialny za poprawne zeskładowanie (zapis) wszystkich danych od chwili rozpoczęcia symulacji. Jeżeli dane nie zostaną zapisane na oddzielny dysk lub taśmę, to wszystkie dane wyjściowe zostaną utracone w chwili wykonania startu nowego scenariusza. Dane nie zostaną stracone jeżeli dokonujemy powtórnego uruchomienia (restartu) systemu.

#### Restartowanie systemu

System JTLS może być restartowany za pomocą *Programu Executive* z dowolnego punktu kontrolnego. Koordynator techniczny może wyświetlić (aktualną, dostępną) listę punktów kontrolnych, z których można dokonać restartu systemu. Jeżeli scenariusz został poprawnie zainicjalizowany, to lista dostępnych punktów kontrolnych powinna zawierać co najmniej punkt kontrolny z etapu inicjalizacji („0000”).

Liczba oraz nazwy programów MPP oraz GENIS w czasie restartu musi być taka sama jak w czasie tworzenia punktu kontrolnego. Koordynator techniczny może zmienić inne informacje o tych procesach.

#### 4.6. Dane sterujące eksperymentem symulacji

Kierownik ćwiczenia posiada szerokie możliwości sterowania przebiegiem symulacji. Oprócz wcześniej omawianych możliwości związanych z tworzeniem punktów kontrolnych może tworzyć i modyfikować TPFDD (*Time-Phased Force and Deployment Data*) oraz LOGINs, zatrzymywać pracę systemu, modyfikować prędkość symulacji, tworzyć szczegółowe raporty związane z funkcjonowaniem modelu, zmieniać parametry symulacji, kontrolować liczbę przesyłanych wiadomości, przywracać (odtwarzać) do symulacji jednostki, zmieniać położenie oddziałów (*Magic Move*).

##### Zmiana prędkości symulacji

Stosunek prędkości zegara czasu rzeczywistego do czasu symulacji może się zmieniać w czasie symulacji. Model został zaprojektowany tak, aby czas symulacji biegł szybciej niż czas rzeczywisty. Przykładowo prędkość symulacji 4:1 oznacza, że każdej minucie czasu symulacji odpowiadają cztery minuty czasu rzeczywistego. Model umożliwia podanie także wyrażenie stosunku prędkości symulacji w ułamkach przykładowo 1,5 : 1 oraz dla prędkości wolniejszych od czasu rzeczywistego (przykładowo 0.5 : 1). Kierownik ćwiczenia może zmienić prędkość symulacji w dowolnym momencie za wyjątkiem okresu, w którym wykonywany jest punkt kontrolny. Choć w oprogramowaniu systemu JTLS nie istnieją ograniczenia prędkości symulacji, to w praktyce będą one istniały ze względu na możliwości przetwarzania określonego zestawu komputerowego.

##### Tworzenie jednostek

W czasie trwania symulacji czasami pojawiają się nieprzewidywalne wymagania nowych jednostek. Wymagania te nie zawsze da się spełnić za pomocą łączenia istniejących jednostek, zwłaszcza w przypadku okrętów wojennych. W celu spełnienia tych wymagań kierownik ćwiczenia może w czasie trwania symulacji utworzyć jednostkę dowolnego typu. Wszystkie dane opisujące jednostkę wprowadzane są w rozkazie *CREATE UNIT* wysłanym do CEP. Tworzenie nowych jednostek jest możliwe tylko za pomocą rozkazów kierownika ćwiczenia. To zdarzenie nie może być umieszczone w bazie danych (tak, jak przykładowo zdarzenie *Assess Weapon Damage*). Podobnie zdarzenie *CREATE UNIT* nie może zostać umieszczone w przyszłym harmonogramie. Jednostka jest tworzona w chwili otrzymania rozkazu, automatycznie ustawiany jest czas TPFDD na wartość 99.0. Wprowadzenie jednostki do symulacji w czasie innym niż domyślny 99.0 wymaga wprowadzenia rozkazu *TPFDD BUILD* (dla którego mają zastosowanie wszystkie typowe reguły).

## Zdarzenia zewnętrzne

Bazy danych JTLS lub kierownik ćwiczenia mogą reprezentować wywiad działający poza teatrem działań lub symulować akcje sił specjalnych za pomocą zdarzeń zewnętrznych. Zdarzenia zewnętrzne mogą być wykorzystane w celu tworzenia, przemieszczania, zniszczenia oraz odtwarzania celów.

### 4.7. Raporty

**Pojedyncze raporty** dostępne dla kierownika ćwiczenia oraz dla ćwiczących to:

Raport o artylerii przeciwlotniczej - <i>ADA Report</i>
Raport o zrzutach lotniczych - <i>Air Lift Drop Report</i>
Raport o jednostce lotniczej - <i>Air Unit Report</i>
Raport o statusie desantu wodno-ładowego - <i>Amphibious Status Report</i>
Raport o barkach - <i>Barge Assets Report</i>
Raport o intensywności walki - <i>Combat Power Report</i>
Raport o trasach ładowych - <i>Ground Route Report</i>
Raport o barierach terenowych - <i>Hex Barrier Report</i>
Raport logistyczny - <i>Logistics Report</i>
Raport o misjach - <i>Mission Report</i>
Raport o rurociągach - <i>Pipeline Report</i>
Raport o biorących udział w grze - <i>Players In Game</i>
Raport o portach - <i>Port Report</i>
Zapytanie o stan konwoju - <i>Query Convoy Status</i>
Zapytanie o możliwości transportowe - <i>Query Lift Capability</i>
Zapytanie o meldujących ćwiczących - <i>Query Reporting Players</i>
Raport o środki kolejowe - <i>Rail Assets Report</i>
Raport o poziomie rekwiracji - <i>Requisition Levels Report</i>
Raport o obszarach zniszczeń - <i>Show BDA Areas</i>
Raport o obszarach bezpośrednich poszukiwań - <i>Show Directed Search Areas</i>
Raport o logowaniach - <i>Show LOGIN</i>
Raport o przemieszczeniach - <i>Show Pushes</i>
Raport o rozmieszczeniu sił w czasie - <i>Show TPFDD</i>
Raport o relacjach strony - <i>Side Relationship Report</i>
Raport o położeniu - <i>Situation Report</i>

Raport o obiekcie - <i>Target Log Report</i>
Raport o władzy nad jednostką - <i>Unit Authority Report</i>
Hierarchia jednostki - <i>Unit Hierarchy</i>
Przewidywana pogoda - <i>Weather Prediction</i>

**Tabela 3. Pojedyncze raporty.**

Pojedyncze raporty dostępne tylko dla kierownika ćwiczenia:

1. Raport o rozmieszeniu wszystkich sił w czasie - *Show TPFDDS*;
2. Raport o zaplanowanych zewnętrznych zdarzeniach - *Scheduled External Events Report*.

Przy użyciu rozkazu *SET DEBUG REPORT FLAGS* kierownik ćwiczenia może wymusić kolejne raporty niedostępne dla ćwiczących:

- Raport o walce lotniczej - *Air-To-Air Combat Report*,
- Raport o zniszczeniach - *Controller Damage Report*,
- Raport o bitwie lądowej - *Ground Battle Report*.

Tym samym rozkazem kierownik ćwiczenia pozwala lub zabrania ćwiczącym na uzyskanie raportów.

**Raporty okresowe** nie są pojedynczą wiadomością lecz grupą 18 wiadomości. Te zbiory wiadomości są generowane bez interwencji kierownika ćwiczenia i ćwiczących co pewien czas określony przez atrybut: *FS TIME BETWEEN PERIODIC REPORTS*. Niektóre z tych raportów nie są przeznaczone dla ćwiczących.

Ćwiczący może wymusić jeden z dostępnych 14 okresowych raportów poprzez rozkaz: *REQUEST PERIODIC REPORT*. Raport taki zostanie wysłany jedynie do MPP wydającego rozkaz ćwiczącego i nie ma wpływu na automatycznie generowane raporty okresowe.

Raporty okresowe składają się z zestawu 14 raportów tworzonych standardowo podczas pracy głównego programu JTLS - CEP. Raporty te są przeznaczone dla każdego z dowódców stron (*Force Side Commander*) oraz kierownika ćwiczenia na zakończenie określonego okresu, którego długość domyślnie jest ustalana na jeden dzień przez ustawienie zmiennej początkowej *PERIODIC.SUMMARY.TIME*. Kierownik ćwiczenia może zmienić czas pomiędzy raportami okresowymi na dowolny okres za pomocą rozkazu *SET MODEL EVENTS TIMES*. W dalszej części rozdziału założymy, że okres został ustawiony na 24 godziny.

Raporty okresowe są także generowane w chwili zatrzymania systemu. Włączając w powyższe wszystkie katastroficzne anomalie programowe „upadki systemu” dla których wykonywana jest opcja tworzenia punktów kontrolnych ASCII. W przypadku normalnego trybu wygenerowania raportów okresowych wszystkie zmienne podsumowań i liczniki są zerowane oraz zliczanie dla „bieżącego okresu” jest ponownie inicjalizowane. Zmienne podsumowań oraz liczniki nie są zerowane w przypadku zatrzymania systemu rozkazem STOP. Przykładowo założmy, że okres pomiędzy raportami okresowymi został ustawiony na 24 godziny oraz polecenie STOP zostało wydane po 12 godzinach. Raporty okresowe z ostatniego 12 godzinnego okresu zostaną wygenerowane. Gdy ćwiczenie zostanie ponownie uruchomione, następny zestaw raportów okresowych zostanie wygenerowany wg oryginalnie ustawionego harmonogramu oraz będzie dotyczył całości 24 godzinnego okresu.

Raporty okresowe są przesyłane w postaci wiadomości wysyłanych do programów MPP ćwiczących oraz kierownika ćwiczenia. Oznacza to, że MPP musi być skonfigurowana do odbioru określonej grupy wiadomości (funkcjonalnie od typu ćwiczącego). Przykładowo, w celu otrzymania raportu *Own Side Aircraft Losses Summary*, ćwiczący musi być zarejestrowany do odbioru wiadomości zakresu Air. Kategorie funkcyjne wiadomości, które mogą być odbierane przez dany program MPP są konfigurowane przez uruchomienie programu *Interface Control Program (ICP)*. Ustawienia wspomniane wyżej mogą być zmieniane w czasie ćwiczenia za pomocą opcji setup-u MPP – *Message Receipt*.

Ćwiczący może także wysłać w dowolnej chwili rozkaz „*REQUEST PERIODICT REPORTS*” w celu otrzymania raportów okresowych. Raporty okresowe tworzone na polecenie ćwiczącego nie zerują zmiennych podsumowań oraz liczników, nie zmieniają także okresu, za który drukowane są raporty okresowe.

Raport okresowy	Funkcje ćwiczącego
Podsumowanie o systemie walki - <i>Combat System Summary</i>	GROUND
Zużycie amunicji i pocisków - <i>Artillery and Missile Expenditures</i>	GROUND, AIR, NAVAL
Dane o jednostce strony - <i>Own Side Unit Data</i>	GROUND
Podsumowanie strat lotniczych strony - <i>Own Side Aircraft Losses Summary</i>	AIR
Podsumowanie misji lotniczych - <i>Air Mission Summary</i>	AIR
Podsumowanie o bazach lotniczych i eskadrach - <i>Airbase and Squadron Summary</i>	AIR

<b>Raport okresowy</b>	<b>Funkcje ćwiczącego</b>
Podsumowanie o nakładach broni lotniczej - <i>Air Weapons Expenditure Summary</i>	AIR
Podsumowanie o wypadach innej strony - <i>Other Side Sortie Summary</i>	AIR, INTELLIGENCE
Podsumowanie o bazach lotniczych przeciwnika - <i>Other Side Airbase Summary</i>	AIR, INTELLIGENCE
Podsumowanie stanu zaopatrzenia i strat - <i>Supplies Status and Loss Summary</i>	AIR, LOGISTICS, GROUND
Podsumowanie o zapasach - <i>Supply Backorder Summary</i>	LOGISTICS
Podsumowanie o stratach środków transportu - <i>Transportation Class Loss Summary</i>	LOGISTICS
Podsumowanie wywiadu o jednostkach przeciwnika - <i>Other Side Unit Intelligence Summary</i>	INTELLIGENCE
Podsumowanie wywiadu o celach przeciwnika - <i>Other Side Target Intelligence Summary</i>	INTELLIGENCE

Tabela 4. Raporty okresowe.

## 5. Dodatkowe narzędzia wykorzystywane z systemem JTLS

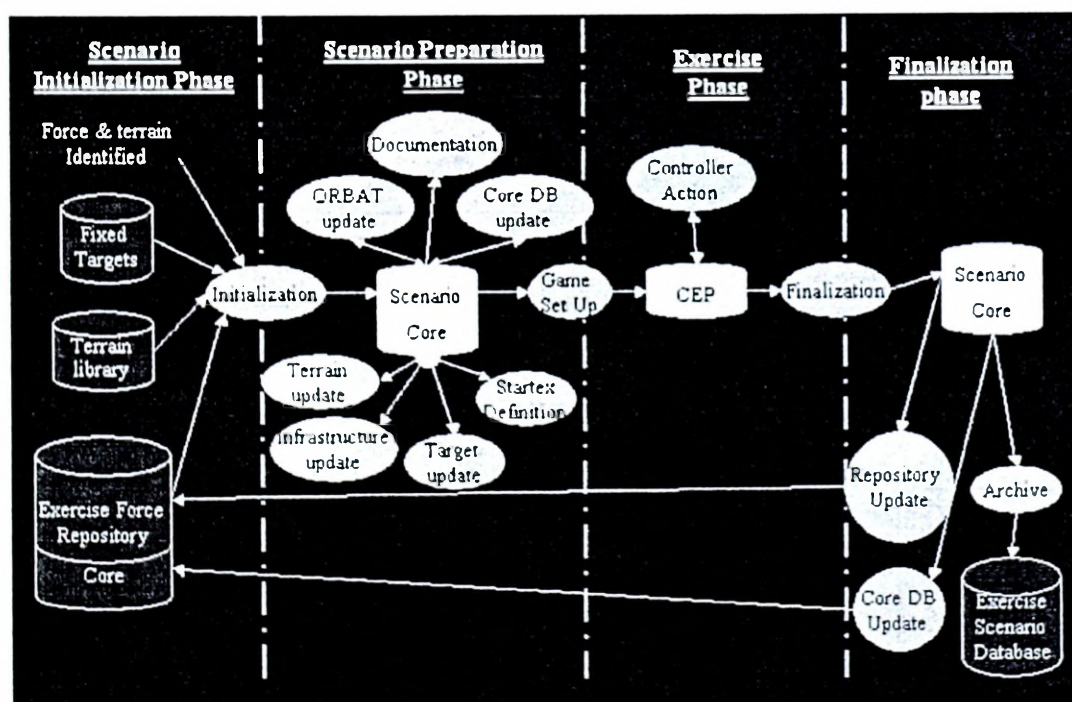
System JTLS składa się z wielu programów, które są wykorzystywane w fazie realizacji ćwiczenia. Część z wyżej wymienionych programów, która nie jest wykorzystywana w fazie realizacji jest bezpośrednio związana z jej przygotowaniem (inicjalizacja) lub jej konfiguracją. System JTLS nie oferuje narzędzi do wspomaganie fazy przygotowania oraz fazy omówienia ćwiczeń.

W związku z potrzebą wspomaganie faz przygotowania oraz omówienia ćwiczeń powstają dodatkowe narzędzia nie związane bezpośrednio z systemem JTLS (nie tworzone przez producenta systemu). Powstawanie dodatkowych narzędzi wspomagających system JTLS jest zgodne z głównym założeniem i koncepcją systemu wyrażona w jego otwartości. System JTLS, w wielu obszarach, zawiera interfejsy do wprowadzenia danych (zasilenia systemu), które mogą być przygotowane poza systemem (przykładem jest format zbiorów ASCII dla danych scenariusza).

W dalszej części przedstawione zostaną dwa programy program *ORBAT* wykorzystywany w fazie przygotowania ćwiczenia oraz program *CADIE* wykorzystywany w fazie omówienia ćwiczenia.

### 2.1.Orbat Editor 2.0

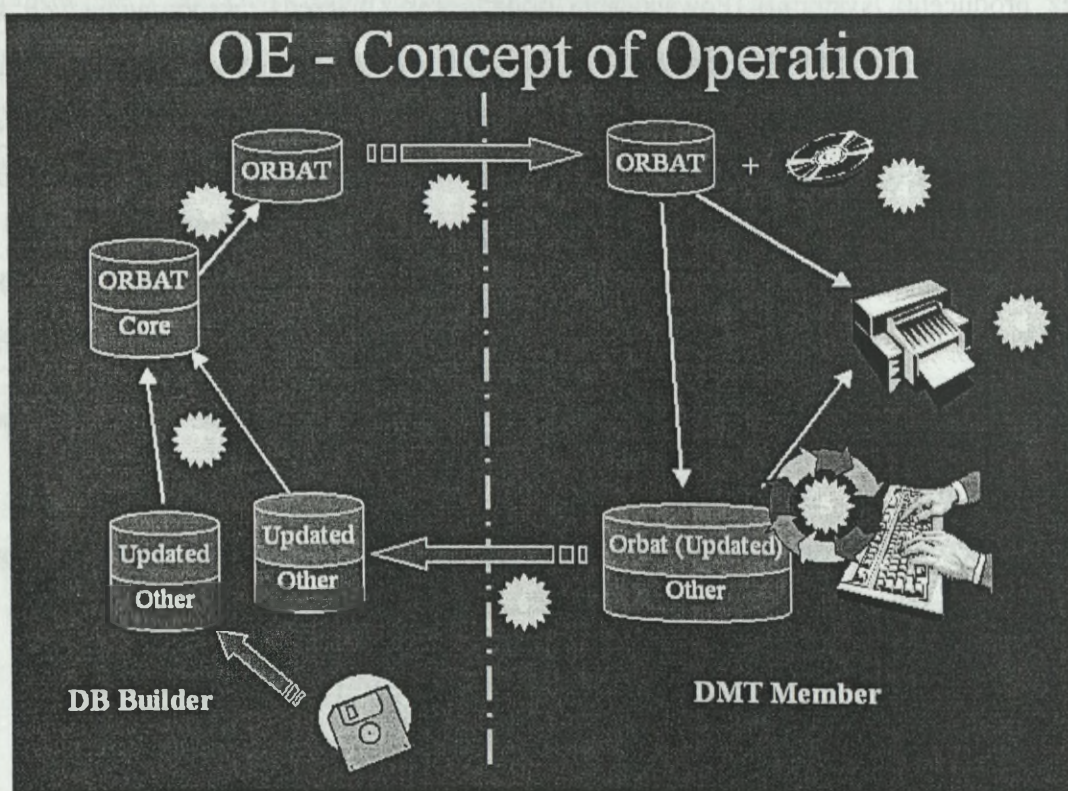
Program Orbat Editor został opracowany w celu uproszczenia, zautomatyzowania i zwiększenie efektywności procesu tworzenia bazy danych dla systemu JTLS.



Rysunek 40. Wykorzystanie Orbat Editor w fazach wypracowania danych.  
/źródło: prezentacja poświęcona programowi Orbat Editor/

W fazie planowania Komputerowej Gry Wojennej program Orbat Editor jest przesyłany do biorących udział w ćwiczeniu dowództw i sztabów. Wraz z nim przesłana jest "mini" baza danych - zawierająca jedynie ogólne informacje o strukturze jednostek i uzbrojeniu (*Order of Battle - ORBAT*). Program Orbat umożliwia aktualizację tej "mini" bazy poprzez dodawanie do niej nowych jednostek, tworzenie prototypów jednostek (TUP - *Tactical Unit Prototype*) oraz modyfikację ich właściwości.

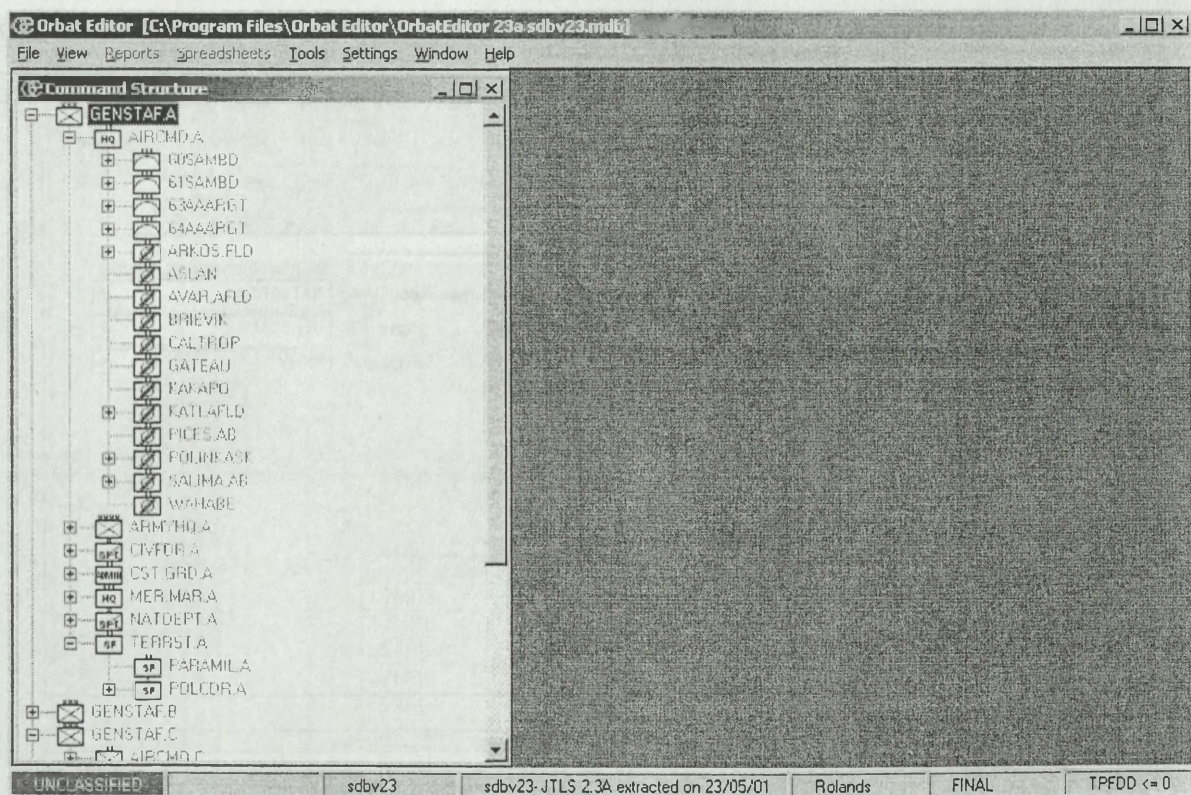
Po wprowadzeniu wszystkich niezbędnych informacji, uaktualniona baza danych przesyłana jest z powrotem do centrum przygotowania danych, gdzie po scaleniu z bazami z innych dowództw oraz pozostałymi danymi tworzy scenariusz JTLS.



Rysunek 41. Koncepcja przygotowania danych.

/źródło: prezentacja poświęcona programowi Orbat Editor/

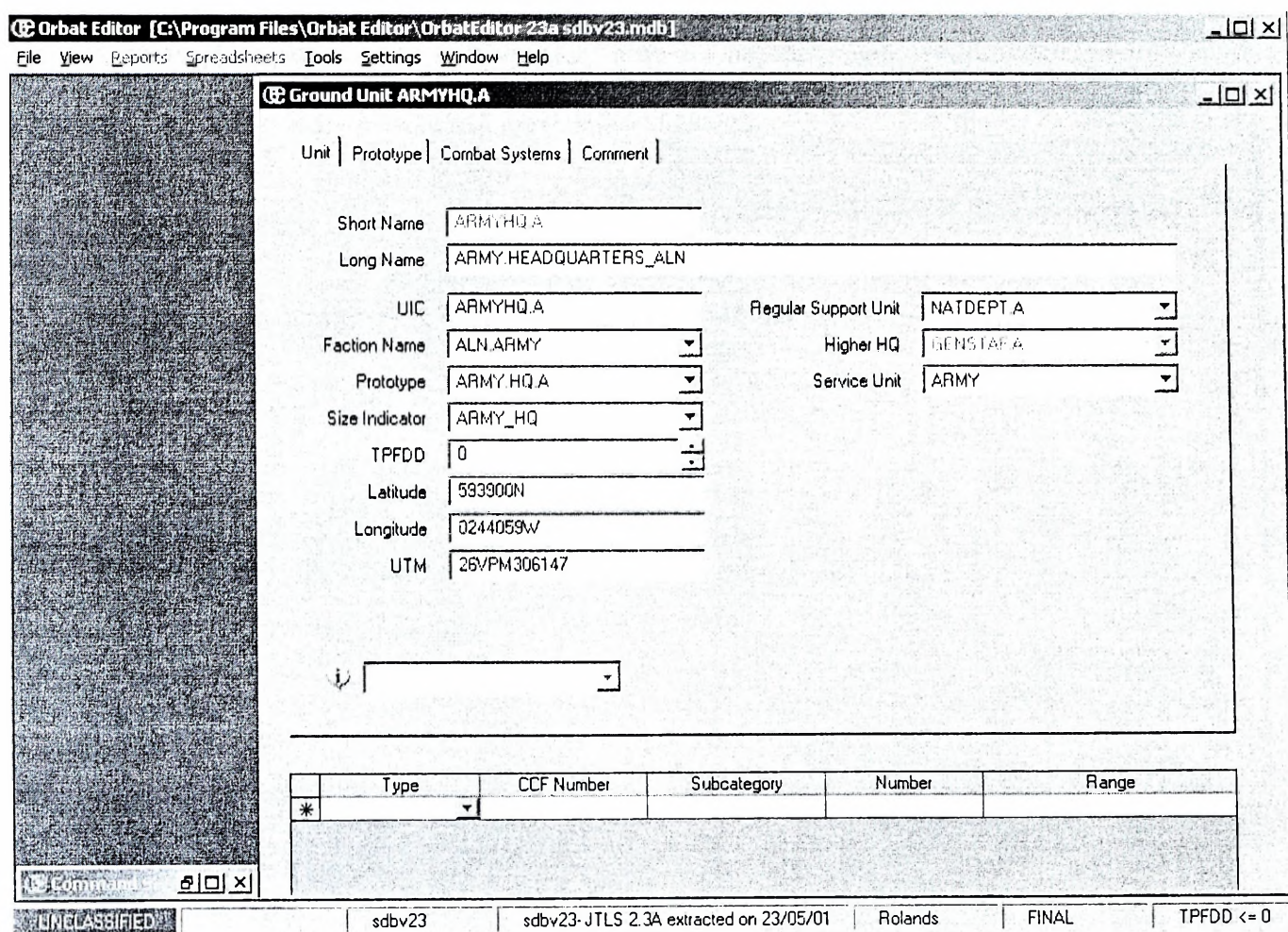
Orbat Editor używa drzewa do zobrazowania powiązań pomiędzy jednostkami, jednostkami nadrzędnymi i stronami konfliktu. Poszczególne jednostki są tu przedstawione jako węzły (gałęzie) posiadające, krótką nazwę oraz symbol graficzny - znak taktyczny jednostki.



Rysunek 42. Struktura dowodzenia.

Jeżeli najedziemy kursorem na jednostkę (węzeł) i naciśniemy prawy przycisk myszy to otworzy się menu, z którego będziemy mogli wybrać opcje *add* - dodaj, *remove* - usuń, *move* - przesuń, *copy* - kopiuj, i *rename* - zmień nazwę. Aby wyświetlić jednostki podporządkowane należy kliknąć + przy węźle, a aby ukryć –.

Po zaznaczeniu symbolu jednostki otworzy się po prawej stronie okno jednostce z czterema zakładkami. Najważniejsza informacja na zakładce **Unit** to *PrTOTYPE*. Mówi ona, z jakiego prototypu (TUP) korzysta jednostka. Wszystkie wytłuszczone pola użytkownik może modyfikować. W polach typu combo (te ze strzałką) zmieniamy dane wybierając nowe z rozwijanej listy.



Rysunek 43. Parametry jednostki lądowej.

W zakładce **Prototype** możemy zmieniać właściwości prototypu w ten sam sposób jak robiliśmy to dla jednostki w zakładce **Unit**. Należy jednak pamiętać, że wprowadzone zmiany będą się również odnosiły do innych jednostek korzystających z tego prototypu. Ich lista znajduje się w tabeli poniżej (*Unit(s) using this Prototype*).

Unit Prototype | Combat Systems | Comment |

Prototype	GDS.INF.BN.B	Capacity Storage Dry (s ton)	4033.00065
Graphic Symbol	INFANTRY	Capacity Storage Wet (gallon)	115680
Average Speed (km/day)	1400	Capacity Carry Dry (s ton)	4033.00065
Maximum Move (hour/day)	20	Capacity Carry Wet (gallon)	115680
Bridge Ops Capable	<input type="checkbox"/>	Caliber	HWY.MORTAR

Unit(s) using this Prototype

Short Name	Type
1-7MRB	GROUND
2-7MRB	GROUND
35MRBN	GROUND
3-7MRB	GROUND
45MRBN	GROUND
53MRBN	GROUND

Type	CCF Number	Subcategory	Number	Range
*				

Rysunek 44. Definicja wzorca (prototypu) jednostki.

Zakładka **Combat Systems** umożliwia zmianę sprzętu przypisanego do prototypu, a tym samym do jednostki. Wprowadzone zmiany będą się również odnosiły do innych jednostek korzystających z tego prototypu.

Unit | Prototype Combat Systems | Comment |

GDS.INF.BN.B

	Name	TOE
	CREW-WEAPONS	72
	IFV	47
	INFANTRY	482
	LAW	37
	MAW	13
	OTHER-TROOPS	168
	TRUCK-CARGO	4
	TRUCK-TANKER	5
▶	TRUCK-UTILITY	18
*		

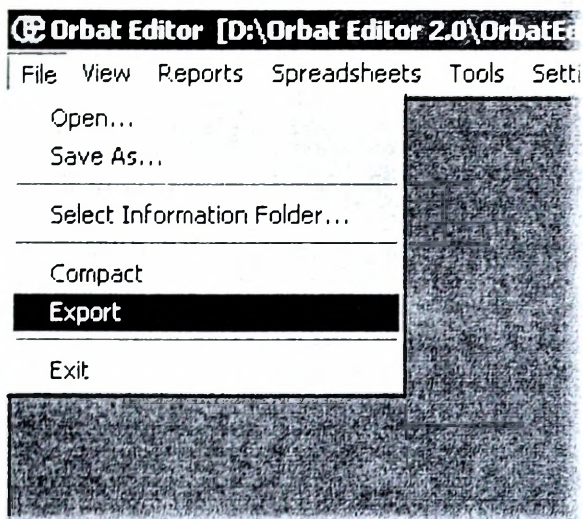
	Type	CCF Number	Subcategory	Number	Range
▶	JAMMER	53MRBN-1	?	1	15
*					

Rysunek 45. Systemy walki jednostki.

Tabela na dole to sekcja *Targets*. Zawiera ona informacje o środkach inżynieryjnych i przeciwlotniczych będących na wyposażeniu jednostki. Znajdujący się tu sprzęt przypisany jest wyłącznie tej jednostce nie jest zawarty w prototypie.

### Eksportowanie danych

Po naniesieniu wszystkich niezbędnych poprawek baza danych programu Orbat jest pakowana i wysyłana do centrum przygotowania danych - korzystając z polecenia *Export*.



Rysunek 46. Polecenie eksportu danych.

Baza danych programu Orbat jest pakowana i dzielona na pliki o rozmiarze, co najwyżej 1,44 MB aby można ją było przenieść na dyskietkach.

## 5.2. Program CADIE

Jako część programu SHAPE. NC3A opracowała program CADIE (od *CAX Disconstaff Enviroment*) dla sztabu kierującego ćwiczeniem do monitorowania i oceny sytuacji podczas ćwiczeń z wykorzystaniem system JTLS.

Pierwsza prototypowa wersja systemu była z powodzeniem wykorzystywana podczas CC02.

### Obszary zastosowania:

- wizualizacja ruchów oddziałów i ich potencjału militarnego podczas ćwiczenia;
- wizualizacja zmian stosunku sił w obszarze odpowiedzialności korpusu;
- analiza strat lotnictwa podczas każdego dnia.

### CADIE – funkcje ogólne

- przedstawia całą strukturę dowodzenia;
- wyświetla na mapie wybrane dywizje (brygady) wraz z podporządkowanymi jednostkami;
- wyświetla zmiany stosunku sił na obszarze operacyjnym korpusu.

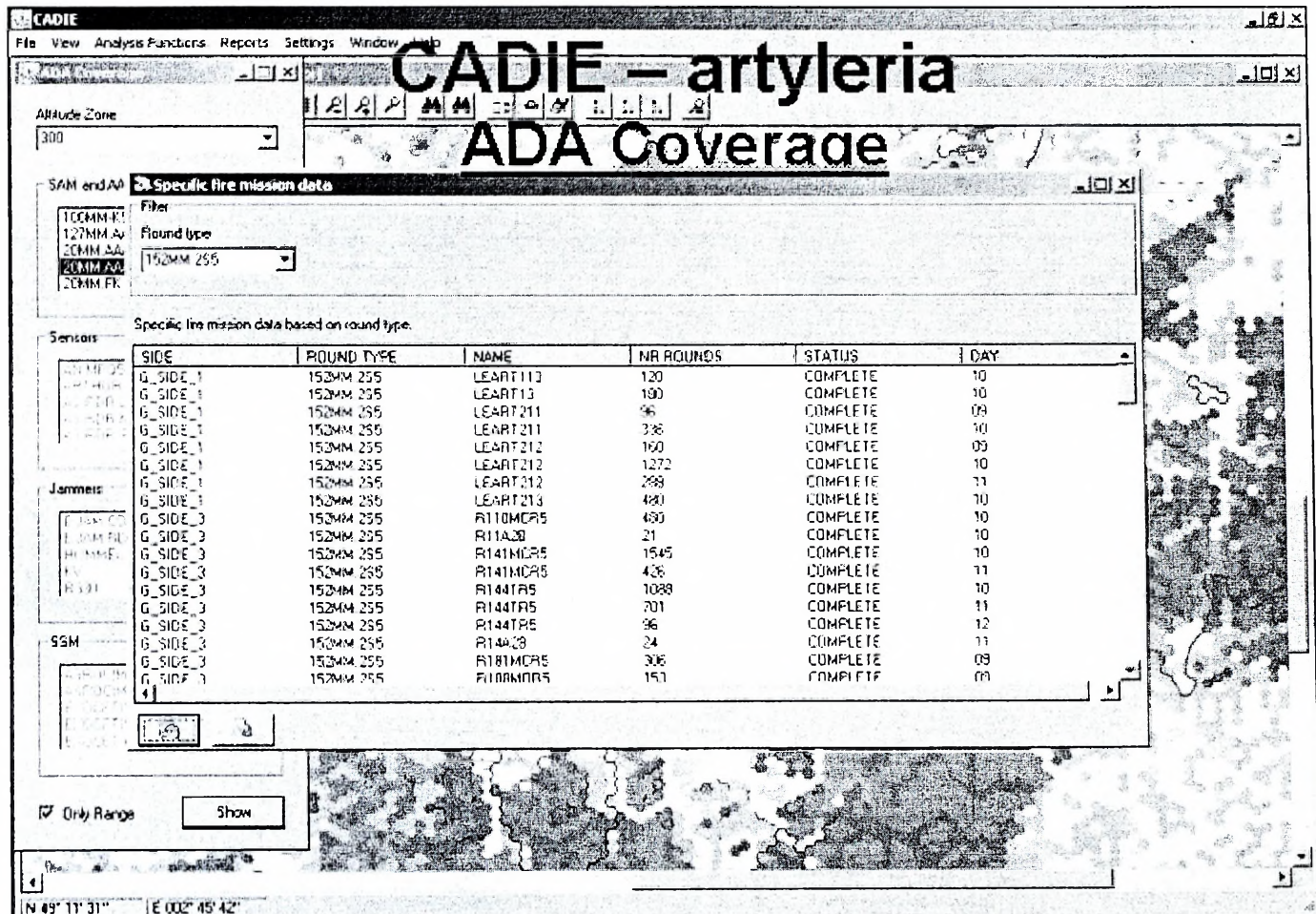
### CADIE – lotnictwo

- ilość strąconych samolotów w wyniku walk powietrznych lub ognia artylerii;
- wizualizacja na mapie miejsca strącenia;
- liczba strąconych samolotów w postaci wykresu.

### CADIE – artyleria przeciwlotnicza

- wyświetla na mapie wybrane środki artylerii przeciwlotniczej;

- wyświetla dla wybranego rodzaju amunicji, ilość wystrzelanych pocisków przez każdą jednostkę każdego dnia;
- wyświetla, dla wybranego rodzaju amunicji, całkowitą ilość wystrzelanych pocisków w ciągu kolejnych dni (bez rozbijania na jednostki, jak wyżej).



Rysunek 47. Ekran CADIE dla artylerii.

## ZAKOŃCZENIE

Opracowanie powstało na podstawie dostępnej dla autorów dokumentacji technicznej oraz materiałów i wniosków z odbywających się w CSiKGW konferencji poświęconych systemowi JTLS. Dokumentacją techniczną dostępną dla autorów był niepełny zestaw dokumentacji poświęconej JTLS v.2.2 oraz „JTLS Executive Overview v.2.5” ze stycznia 2003, pozyskanego spod adresu internetowego: [www.rolands.com](http://www.rolands.com). Poza problemami z tłumaczeniem dokumentacji, napisanej trudnym językiem, autorzy opracowania znaleźli niespójności oraz błędy w tejże dokumentacji. Przez co przedstawiony materiał z pewnością nie jest doskonały. Autorzy, z powodu braku polskojęzycznej literatury poświęconej systemowi JTLS, zaproponowali własne tłumaczenie pojęć oraz nazw własnych związanych z systemem.

Przyszłe opracowania poświęcone symulacyjnemu systemowi walki będą z pewnością powstawały na podstawie badania i rozpoznawania systemu informatycznego JTLS, a nie tylko dokumentacji technicznej.

## Spis rysunków

Rysunek 1. Relacje pomiędzy stronami w JTLS.....	6
Rysunek 2. Wariant seminarium.....	7
Rysunek 3. Wariant otwartego wsparcia. ....	8
Rysunek 4. Wariant ukrytego wsparcia. ....	9
Rysunek 5. Wariant rozdzielonych danych wyjściowych. ....	9
Rysunek 6, Struktura techniczno-organizacyjna w wersji oszczędnej. ....	13
Rysunek 7. Struktura techniczno-organizacyjna w wersji rozszerzonej. ....	14
Rysunek 8. Struktura programowa JTLS.....	15
Rysunek 9. Ekran użytkownika. ....	21
Rysunek 10. Okno aplikacji GIAC.....	22
Rysunek 11. Przykładowa tabela w aplikacji IMT.....	23
Rysunek 12. Przykładowe okno aplikacji MPP.....	24
Rysunek 13. Przykładowe okno aplikacji OPM.....	25
Rysunek 14. Przepływ informacji w fazie inicjalizacji systemu. ....	26
Rysunek 15. Wymiana informacji pomiędzy CEP a GENIS. ....	27
Rysunek 16. Współpraca Primary GENIS z Secondary GENIS. ....	27
Rysunek 17. Współpraca programu CEP z programem GENIS. ....	28
Rysunek 18. Przepływ informacji dla programu GIAC. ....	29
Rysunek 19. Przepływ informacji dla programu MPP. ....	30
Rysunek 20. Przepływ informacji dla programu IMT.....	30
Rysunek 21. Przykładowe zobrazowanie w GIAC obszaru działań bojowych.....	31
Rysunek 22. Sześciokątne klastry w JTLS.....	32
Rysunek 23. Możliwe kierunki ruchu jednostki. ....	34
Rysunek 24. Przemieszczenie jednostki z punktu A do B.....	34
Rysunek 25. Formatka rozkazu przemieszczenia. ....	35
Rysunek 26. Algorytm przemieszczania jednostki.....	36
Rysunek 27. Przykład przemieszczenia grupy jednostek. ....	37
Rysunek 28. Oddziaływanie jednostek w walce lądowej. ....	41
Rysunek 29. Algorytm przygotowania jednostki, w modelu, do starcia. ....	42
Rysunek 30. Przykładowa tablica Combat Index. ....	43
Rysunek 31. Przykładowa tablica współczynników efektywności - FWL.....	44
Rysunek 32. Powiązania między klasami HRU, Unit, a HIGHRES.UNIT.PROTOTYPE. ....	51

Rysunek 33. Definicja atrybutu HUP.CS.TOE.....	51
Rysunek 34. Główne menu Executive Program. ....	54
Rysunek 35. Wybór scenariusza.....	54
Rysunek 36. Wymagana struktura katalogów. ....	55
Rysunek 37. Wybór mapy dla scenariusza. ....	56
Rysunek 38. Polecenie uruchomienia symulacji. ....	58
Rysunek 39. Polecenie powtórnego uruchomienia symulacji. ....	59
Rysunek 40. Wykorzystanie Orbat Editor w fazach wypracowania danych.....	67
Rysunek 41. Koncepcja przygotowania danych.....	68
Rysunek 42. Struktura dowodzenia.....	69
Rysunek 43. Parametry jednostki lądowej.....	70
Rysunek 44. Definicja wzorca (prototypu) jednostki.....	71
Rysunek 45. Systemy walki jednostki.....	72
Rysunek 46. Polecenie eksportu danych.....	73
Rysunek 47. Ekran CADIE dla artylerii.....	74

## **Spis tabel**

Tabela 1. Zestawienie przeszkód terenowych.....	33
Tabela 2. Klasy baz danych.....	50
Tabela 3. Pojedyncze raporty.....	64
Tabela 4. Raporty okresowe.....	66

## Literatura

1. JTLS analyst's guide v.2.2.
2. JTLS data requirements manual v.2.2.
3. JTLS executive overview v.2.5.
4. JTLS players guide v.2.2.
5. Prezentacja poświęcona programowi Orbat Editor.
6. Prezentacje z konferencji w CSiKGW z dnia 22.08.2003.
7. Ratajczak M., Wieleba R., Wocial J: Symulacyjny model walki na szczeblu taktycznym. Synteza koncepcyjnego modelu walki. AON 2002.
8. Spustek H.: Wybrane zagadnienia badań operacyjnych i modelowania liniowego. AON 2002.
9. Strona internetowa firmy: [www.rolands.com](http://www.rolands.com).

# SPIS TREŚCI

WSTĘP .....	3
1. OGÓLNA CHARAKTERYSTYKA SYSTEMU JTLS .....	4
2. WARIANTY UŻYCIA SYSTEMU JTLS .....	7
3. SYSTEM JTLS .....	10
3.1. Podstawowa konfiguracja sprzętowa i jej warianty .....	10
3.2. Logiczno programowa struktura systemu .....	15
3.2.1. Programy tworzenia scenariusza.....	16
3.2.2. Programy inicjujące oraz konfiguracyjne .....	19
3.2.3. Program zdarzeń walki.....	20
3.2.4 Podstawowe programy użytkowe .....	21
3.3. Podstawowe przepływy informacji .....	25
3.4. Odwzorowanie terenu.....	31
3.5. Modele formalne .....	39
3.5.1. Systemy walki .....	46
3.6. Bazy danych .....	47
4. PRZYGOTOWANIE SYSTEMU JTLS DO PRACY .....	52
4.1. Stanowiska w trakcie ćwiczenia.....	52
4.2. Koordynator techniczny .....	53
4.3. Start symulacji i jej powtórne uruchomienie.....	57
4.4. Punkty kontrolne .....	59
4.5. Zatrzymywanie symulacji .....	61
4.6. Dane sterujące eksperymentem symulacji .....	62
4.7. Raporty .....	63
5. Dodatkowe narzędzia wykorzystywane z systemem JTLS .....	67
2.1.Orbat Editor 2.0.....	67
5.2. Program CADIE .....	73
ZAKOŃCZENIE .....	75
Spis rysunków.....	76
Spis tabel.....	77
Literatura.....	78

