

Metoda generacji obrazu VR w symulatorze pola walki

Andrzej SZYMAŃSKI, Wojciech SATAŁA

Centrum Symulacji i Wizualizacji Komputerowej, Wydział Cybernetyki, ul Kaliskiego 2,
00-908 Warszawa.

STRESZCZENIE: Artykuł opisuje zagadnienie wizualizacji wirtualnego pola walki na podstawie danych wygenerowanych przez ośrodki symulacji konstruktywnej. Autorzy proponują wykorzystanie modułu sprzęgającego tłumaczącego informację o dużym stopniu agregacji na postać możliwą do wykorzystania w systemach wirtualnej rzeczywistości. Ponadto określają koncepcję funkcjonowania generatora obrazu jako pasywnego elementu rozproszonego środowiska symulacyjnego.

Wstęp

Opracowanie przedstawia zagadnienia związane z problemem wizualizacji wirtualnego pola walki. Pojęcie *pole walki* lub *wirtualne pole walki* określa kompleksową i dynamicznie zmieniającą się przestrzeń informacyjną, zawierającą różnorodne dane: geograficzne, wojskowe, atmosferyczne, abstrakcyjne, itd. Ze względu na skończony czas aktualności danych oraz wymogi reżimu czasu rzeczywistego, wizualizacja tej złożonej przestrzeni staje się możliwa tylko dzięki zastosowaniu odpowiednich algorytmów programistycznych oraz szybkich stacji graficznych. System wizualizacji wirtualnego pola walki powinien być wynikiem kompromisu pomiędzy wiernością odtworzenia obszaru działań bojowych oraz efektywnością pracy całego systemu graficznego. Na podstawie nieustannie napływających danych i przy wykorzystaniu bogatego zbioru baz danych o terenie i obiektach uczestniczących w symulacji, system prezentuje bieżącą sytuację taktyczną i operacyjną. Metody prezentacji uwzględniają zarówno mapy 2D, jak również

wirtualne obrazy 3D. Na mapy 2D nakłada się dodatkowe informacje służące szybszemu i prawidłowemu zrozumieniu bieżącej sytuacji. Mogą to być siatki, kontury, drogi, obiekty wojskowe, takie jak: czołgi, samoloty, śmigłowce, budynki i wiele innych.

Prezentacje o charakterze 3D, oprócz tego, że pozwolą dowódcom prowadzonych ćwiczeń na uzmysłowienie i właściwe wyobrażenie pozycji ich wojsk, udostępnią możliwość wirtualnego poruszania się po terenie i inspekcję wybranych lub wszystkich fragmentów obszaru działań. Udostępnienie funkcji pełnej interakcji użytkownika z systemem, pozwoli uczestnikowi gry wojennej na wybór sposobu poruszania się po terenie. W zależności od intencji, dowódca może oglądać wojsko z "lotu ptaka" lub przemierzać teren obiektem naziemnym. Mechanizmy interakcji udostępnią metody selektywnego pozyskiwanie informacji o oddziałach, pododdziałach lub poszczególnych obiektach wojskowych. Uporządkowane przedstawienie sytuacji taktycznej w przestrzeni 3D będzie możliwe dzięki implementacji funkcji agregacji informacji i zastosowaniu standardowych lub abstrakcyjnych symboli wojskowych.

Niniejsze opracowanie ma na celu przedstawienie metod służących generowaniu wirtualnego pola walki w rozproszonym symulatorze pola walki.

1. Założenia ogólne systemu wizualizacji

Wizualizacja w czasie rzeczywistym tak charakterystycznego, a zarazem złożonego zagadnienia, jakim jest konflikt zbrojny, wymaga wcześniejszego przygotowania odpowiednich baz danych modeli dwu- i trójwymiarowych. Wśród modeli 3D należy wyróżnić:

- model terenu, na którym prowadzone są działania zbrojne,
- modele pojazdów wojskowych, czołgów, transporterów, śmigłowców, itd.,
- modele rzeczywistych lub abstrakcyjnych symboli militarnych,
- modele elementów środowiska naturalnego i sztucznego: drzewa, lasy, obiekty wodne, słupy trakcji elektrycznej, wieże triangulacyjne, itd.

Bazę obiektów dwuwymiarowych powinny stanowić:

- mapa cyfrowa obszaru adekwatna do jej trójwymiarowego odpowiednika,
- znaki taktyczne nanoszone na mapę w czasie planowania działania i wykonywania zadania,
- inne symbole pomocnicze.

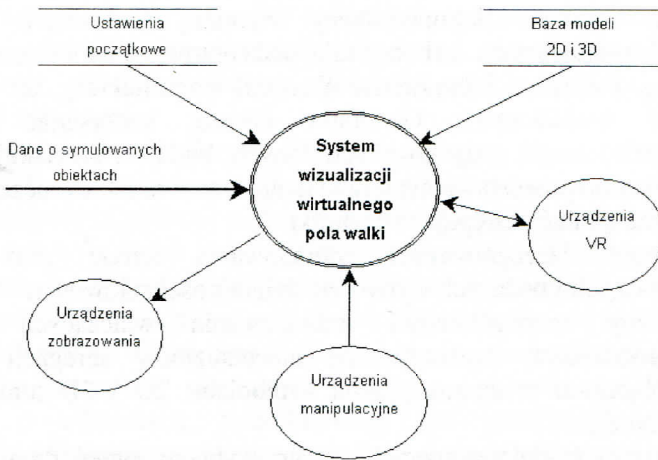
Ponieważ system interaktywnej wizualizacji jest tylko jednym z elementów kompleksowego symulatora sztabowego, należy zadbać o odpowiedni interfejs pomiędzy nim a ośrodkami symulacji konstruktywnej lub rzeczywistej.

Współczesne techniki komputerowej wymiany informacji stosowane w sieciowych symulatorach wykorzystują zaakceptowane na świecie protokoły, np. DIS (*ang. Distributed Interactive Systems*) i architektury, np. HLA (*ang. High Level Architecture*). Generator obrazu, korzystając ze zbioru zgromadzonych i ciągle pozyskiwanych danych, buduje i prezentuje wirtualne pola walki. Sposób przedstawienia terenu działań oraz obiektów uczestniczących powinien uwzględniać następujące czynniki:

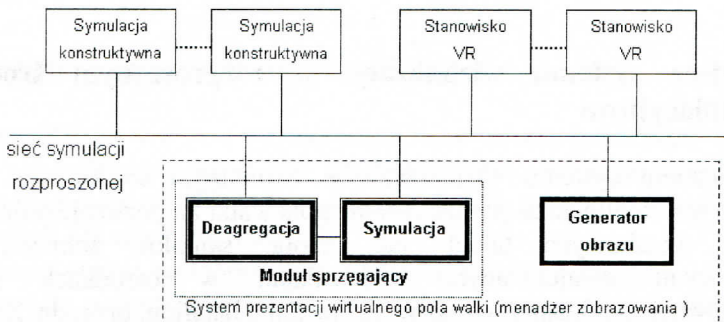
- poziomy szczegółowości zobrazowania terenu oraz obiektów naturalnych i będących wytworem działalności człowieka,
- poziomy szczegółowości zobrazowania walczących oddziałów i pododdziałów (wykorzystanie mechanizmów agregacji informacji w połączeniu z abstrakcyjnymi symbolami 2D i 3D umieszczonymi w scenie),
- wizualizację efektów specjalnych, np. wybuchy, ogień, dym, mgła, itd.
- możliwość prezentacji dodatkowych informacji o terenie i wybranych oddziałach, pododdziałach, obiektach wojskowych,
- udostępnienie kilku modeli ruchu obserwatora na potrzeby eksploracji wirtualnej sceny.

2. Miejsce systemu wizualizacji w rozproszonym środowisku symulacyjnym

Zadaniem systemu wizualizacji w wirtualnym środowisku konfliktu zbrojnego jest inteligentne przedstawienie pola walki na poziomie pojedynczych obiektów wojskowych takich jak: czołg, samolot, żołnierz. Jednak podstawowymi symulowanymi jednostkami w ośrodkach symulacji konstruktywnej są oddziały wojskowe: kompania, batalion, brygada. Z założenia symulatory takie oferują wiele walorów treningowych i dydaktycznych, ale tylko dowódcom wyższych szczebli. Istnieje jednak wiele sytuacji, w których "wgląd" na pole walki jest bardzo pożądany, a czasami nawet nieodzowny. Zintegrowany system wirtualnego pola walki powinien oferować możliwość inspekcji pola bitwy oraz interakcji z załogami poszczególnych obiektów. Powinien także pozwalać na symulowane przeprowadzenie mniejszych operacji wojskowych, w których udział bierze tylko kilka pododdziałów, ale które pozostają w integracji ze swoimi oddziałami macierzystymi. Oznacza to, że warunkiem koniecznym wykonania takich ćwiczeń jest zachowanie spójności danych [1]. Efekty walki pododdziału muszą w określony sposób wpływać na stan oddziału macierzystego, czyli muszą się propagować w górę hierarchii oddziałów wojskowych.



Rys. 2.1. Schemat kontekstowy środowiska podsystemu zobrazowania.



Rys. 2.2. Architektura sieci symulowanego konfliktu w środowisku VR

Rys. 2.1 przedstawia podsystem wizualizacji w syntetycznym środowisku wirtualnego konfliktu zbrojnego. Jest on odrębnym i pomocniczym elementem składowym całego systemu. Jego działanie polega na pobieraniu i inteligentnym zobrazowaniu danych. Jest to zadanie wystarczające dla mocnych stacji graficznych, więc niepożądanym wydaje się obciążenie go dodatkowymi obliczeniami, które muszą wystąpić w postaci interfejsu pomiędzy agregatami symulującymi działanie oddziałów a podsystemem zobrazowania. Należy więc powołać i zaimplementować specjalne moduły sprzęgające, przygotowujące dane o każdym jednostkowym obiekcie. Moduły takie, realizujące proces deagregacji, realizować powinny następujące czynności:

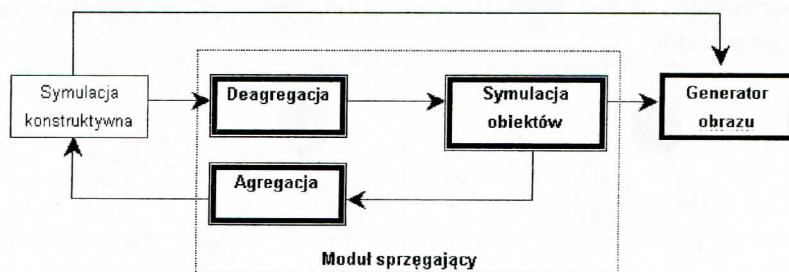
- powoływać obiekty wojskowe (czołg, transporter, BWP, itd.) przynależne do danego oddziału i rozmieszczać je w terenie zgodnie z zasadami sztuki wojennej, uwzględniając:
 - doktrynę wojenną strony walczącej,
 - kontekst bieżącej sytuacji na polu walki (oddział w ataku, w obronie, w marszu),
 - właściwości terenu, a w szczególności obecność takich elementów jak: lasy, zbiorniki wodne, bagna, tereny zurbanizowane,
- symulować i kontrolować pozycję i status danej jednostki w środowisku wirtualnym,
- symulować występowanie efektów walki typu: wybuchy, dym, płomienie, zniszczenia jednostek,
- usuwać jednostki z wirtualnego pola walki na żądanie.

Architektura sieci przedstawiona na rys. 2.2 przewiduje zastosowanie modułowego systemu prezentacji. Zapewnia to dużą elastyczność prezentacji wirtualnego pola walki. Funkcje transformacji danych pomiędzy agregatami i generatorem obrazu realizuje jeden lub kilka modułów pośredniczących.

3. Metody współpracy menadżera zobrazowania z otoczeniem

Ponieważ symulacja konstruktywna ma charakter zdarzeniowy, natomiast zadaniem podsystemu zobrazowania jest prezentacja pola walki w czasie rzeczywistym, należy wskazać metodę zarządzania symulacją pozwalającą na zachowanie jedności czasu i integralność danych.

3.1. Metoda z wykorzystaniem sprzężenia zwrotnego



Rys. 3.1. Diagram przepływu informacji w reżimie czasu rzeczywistego.

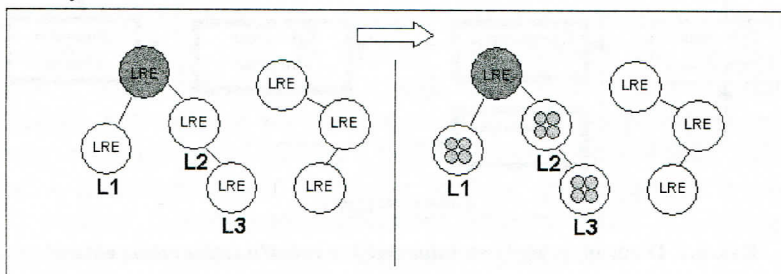
Idealnym rozwiązaniem byłoby opracowanie mechanizmów pozwalających na kontrolowane przejście symulacji konstruktywnej w tryb działania czasu rzeczywistego. Jeżeli wystąpią określone uwarunkowania, następuje deagregacja oddziału do pojedynczych obiektów i przejście kontroli nad obiektami przez moduł symulacji. Od tego momentu symulacja konstruktywna uaktualnia jedynie informacje o swojej jednostce na podstawie danych uzyskanych z modułu sprzęgającego.

Zadaniem modułu musi być więc monitorowanie akcji i statusów poszczególnych obiektów oraz uśrednianie wyników walki toczącej się w wirtualnym świecie. Dodatkowym wymaganiem jest zapewnienie wzajemnego oddziaływania pomiędzy jednostkami-agregatami a jednostkami-obiektami. Przykładem może być prowadzenie ognia przez agregaty do pojedynczych obiektów i odwrotnie. Rys. 3.1 przedstawia schemat przepływu danych w omawianym rozwiązaniu. W tym wypadku symulacja konstruktywna musi uwzględnić dane napływające z wirtualnego pola walki, co jest warunkiem koniecznym zachowania integralności danych, a dodatkowo daje możliwość interakcji ze stanowiskami wirtualnej rzeczywistości. Niezbędne jest więc jej wyposażenie w interfejs pomiędzy procesami symulacji konstruktywnej i wirtualnej rzeczywistości.

Dwukierunkowa interakcja pomiędzy obiektami wysokiej (*ang. High Resolution Entity - HRE*) i niskiej rozdzielczości (*ang. Low Resolution Entity - LRE*) wymaga zastosowania jednego z następujących sposobów deagregacji informacji:

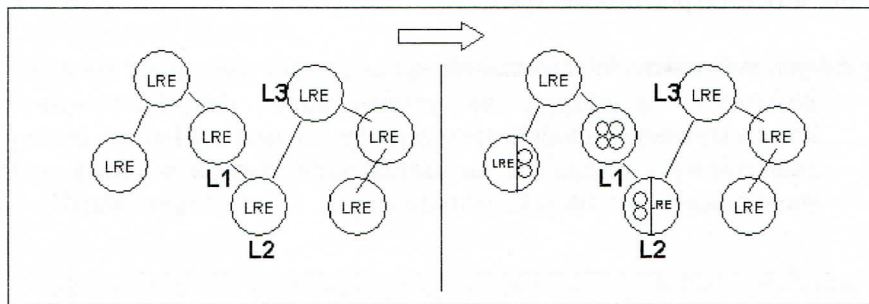
1. Deagregacja pełna (*ang. full disaggregation*).
2. Deagregacja częściowa (*ang. partial disaggregation*).
3. Deagregacja obszarowa (*ang. playboxes*).

Ad.1. Polega na operacji deagregowania wszystkich jednostek wchodzących w skład danego oddziału. Uaktywniana przeważnie w momencie kontaktu jednostek LRE z jednostkami HRE (rys.3.2). Zapewnia, że wszystkie jednostki współpracują na tym samym poziomie rozdzielczości. Nie jest to jednak metoda efektywna z powodu łańcuchowej operacji "rozbijania" kolejnych jednostek. Zarządzanie dużą liczbą pojedynczych obiektów może powodować znaczne obciążenie systemu.



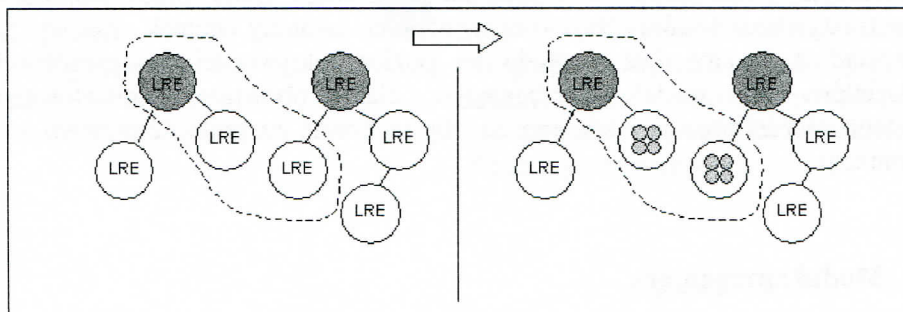
Rys. 3.2. Deagregacja pełna.

Ad.2. W przeciwieństwie do pełnej deagregacji, wykonuje operacje tylko na pewnej części jednostek LRE, zgodnie z odpowiednim algorytmem. Algorytm powinien być dobrany w taki sposób, aby deagregacja częściowa nie przerodziła się w pełną. Na rys. 3.3 tylko część jednostki L2 jest reprezentowana przez pojedyncze obiekty, które pozostają w kontakcie z obiektami L1. Pozostała część L2 pozostaje w stanie niskiej rozdzielczości i oddziały z L3.



Rys. 3.3. Deagregacja częściowa.

Ad.3. Polega na wyznaczeniu obszaru (rys.3.4), w którym mogą znajdować się tylko jednostki HRE, bez względu na to, z jakiego powodu tam się znalazły. Warunkiem koniecznym rozpoczęcia procesu deagregacji jest więc przekroczenie wyznaczonej granicy przez jednostkę LRE. Z chwilą opuszczenia obszaru wysokiej rozdzielczości następuje proces agregacji. Z reguły obszary wysokiej rozdzielczości pozostają statyczne, jeśli chodzi o granice i lokalizację, ale istnieje możliwość implementacji obszarów dynamicznych. Przykładem może być takie rozwiązanie, które jako obszar wysokiej rozdzielczości traktuje fragment pola walki prezentowany przez system wizualizacji.



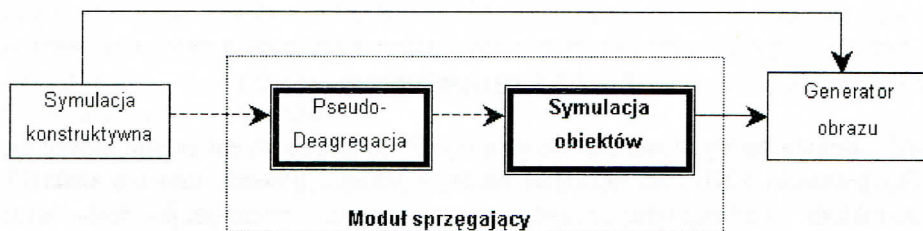
Rys. 3.4. Deagregacja obszarowa.

3.2. Metoda bez sprzężenia zwrotnego

Jeżeli symulacja konstruktywna nie obsługuje informacji zwrotnej lub działania w trybie czasu rzeczywistego, zadanie deagregacji oraz zachowania integralności danych spoczywa na module sprzęgającym. W tym wypadku mówimy o procesie pseudo-deagregacji. Przepływ informacji odbywa się w jedną stronę, co przedstawia rys.3.5.

Przy danych uwarunkowaniach możliwe są następujące tryby pracy systemu:

- krokowy - polegający na przekazywaniu danych z symulacji konstruktywnej do modułu sprzęgającego co ustalony kwant czasowy,
- zdarzeniowy - polegający na uaktualnianiu danych w chwili zmiany stanu procesu symulacji konstruktywnej.



Rys. 3.5. Diagram przepływu informacji w systemie bez sprzężenia zwrotnego.

Zdarzeniowy charakter pracy systemu symulacji konstruktywnej uniemożliwia przedstawienie pola walki jako symulacji wirtualnej. Jedynym rozwiązaniem jest statyczna prezentacja bieżącej sytuacji taktycznej na wirtualnym polu walki. W przypadku krokowego trybu pracy symulacja konstruktywna uaktualnia dane o agregatach co ustalony odcinek czasowy, na przykład 2 minuty, bez względu na poziom aktywności poszczególnych oddziałów. Na module sprzęgającym ciąży obowiązek zniwelowania rozbieżności informacji w taki sposób, aby zachować integralność prowadzonej symulacji.

4. Moduł sprzęgający

Podstawowym zadaniem modułu sprzęgającego działającego w systemie bez sprzężenia zwrotnego jest przetworzenie informacji zakodowanej w pakietach pochodzących od symulacji konstruktywnej na dane zrozumiałe dla

generatora obrazu przy zachowaniu odpowiedniej częstotliwości uaktualniania. Przyjmuje się założenie, że generator obrazu, oprócz właściwych zadań wizualizacyjnych, nie realizuje innych funkcji. Wyjątek mogą stanowić proste obliczenia wyznaczania pośrednich położenia obiektów pomiędzy kolejnymi uaktualnieniami, nadchodzącymi co określony kwant czasu. Wobec powyższego moduł sprzęgający jest odpowiedzialny za:

- pozyskiwanie danych z sieci,
- deagregację informacji,
- symulację obiektów,
- tworzenie pakietów danych o obiektach i ich transfer do generatora obrazu.

4.1. Dane wejściowe

Dla zapewnienia poprawnej pracy modułu sprzęgającego, niezbędne jest zagwarantowanie systematycznej dystrybucji pakietów danych opisujących stan poszczególnych agregatów. Pożądane przez moduł dane powinny zawierać:

- identyfikator I strony (armii, grupy ćwiczącej),
- identyfikator IA agregatu,
- położenie P środka oddziału - współrzędne x, y, z ,
- kształt K agregatu - odpowiedni do bieżącego zadania,
- wymiary W agregatu: szerokość, długość, wysokość,
- zorientowanie O w przestrzeni - współrzędne Ψ, Θ, Φ ,
- prędkość V poruszania się,
- zdolność prowadzenia ognia F ,
- ukompletowanie U ,
- liczbę obiektów wojskowych N (obiektów wysokiej rozdzielczości).

Dodatkowo, oprócz opisu stanu, agregaty powinny dystrybuować informacje na temat prowadzonych akcji bieżących, typu: walka w obronie, walka w ataku, przełamanie, pościg, wycofanie, marsz, przegrupowanie, itd. Pozwoliłoby to na wzbogacenie wirtualnego pola walki audiowizualnymi efektami specjalnymi. Protokół DIS oferuje takie możliwości poprzez udostępnienie pakietów niosących informacje o podejmowanych akcjach przez agregaty.

4.2. Dane wyjściowe

Generowane przez moduł dane wyjściowe stanowią jedyne źródło informacji o obiektach wysokiej rozdzielczości dla generatora obrazu. Dane te muszą być kompletne, a przede wszystkim odnawiane z odpowiednią

częstotliwością. Można wyróżnić trzy kategorie danych udostępnianych przez moduł sprzęgający:

- dane o stanie poszczególnych obiektów,
- dane o procesie prowadzenia ognia,
- dane powołujące i usuwające objekty ze sceny VR.

4.2.1. Informacje o stanie obiektów

- identyfikator i strony,
- identyfikator j jednostki,
- położenie p środka obiektu – współrzędne x, y, z ,
- zorientowanie o w przestrzeni - współrzędne Ψ, Θ, Φ ,
- prędkość v ,
- parametry dr dla algorytmu Dead Reckoning realizującego wyznaczanie położen pośrednich: przyspieszenie liniowe, prędkość kątowna, rodzaj stosowanego algorytmu,
- zdolność bojowa jednostki: zapas paliwa zp , zapas amunicji za , poziom zniszczenia d ,
- informacje c o stanie ruchomych części obiektu, np. lufa czołgu,
- pożądane informacje dodatkowe x : rodzaj jednostki, rok produkcji, itp.

4.2.2. Informacje o procesie prowadzenia ognia

- identyfikator jednostki prowadzącej ogień,
- identyfikator jednostki przeznaczenia,
- rodzaj amunicji,
- prędkość lotu pocisku,
- zasięg działania,
- obszar rażenia.

4.2.3. Informacje o powołaniu lub usunięciu obiektu

- identyfikator obiektu,
- identyfikator agregatu,
- rodzaj obiektu.

4.3. Przetwarzanie danych w module sprzęgającym

Zadania modułu sprzęgającego można przedstawić za pomocą grafu zależnych od siebie atrybutów, reprezentujących zarówno jednostki wysokiej jak i niskiej rozdzielczości oraz zbioru funkcji mapujących, zapewniających integralność danych [4].

Niech $Model^A$ będzie modelem obiektu niskiej rozdzielczości składającego się ze zbioru obiektów wysokiej rozdzielczości $Model^B$. $Model^M$ definiuje wszystkie atrybuty Atr danego obiektu, relacje pomiędzy nimi Rel oraz możliwe interakcje Int obiektu z otoczeniem:

$$\begin{aligned} Model^M &= \langle Atr^M, Rel^M, Int^M \rangle, \\ Atr^M &= Atr^A \cup Atr^B, \\ Rel^M &= Rel^A \cup Rel^B \cup Rel^{AB}, \end{aligned}$$

gdzie :

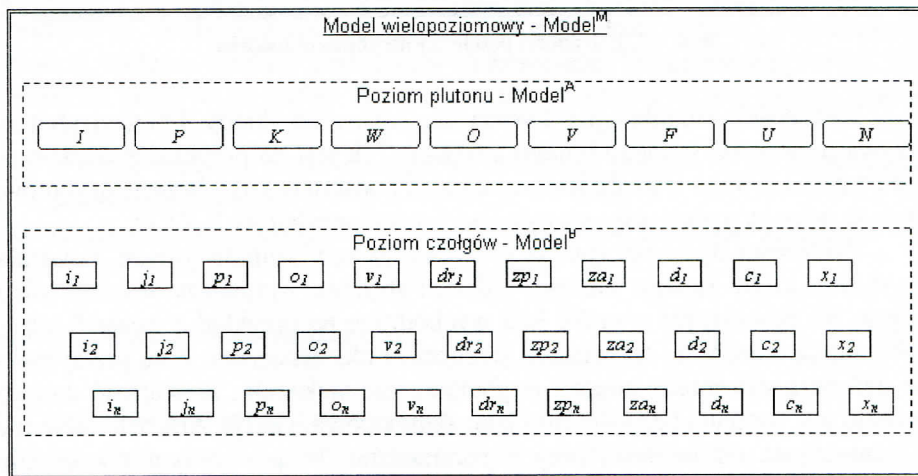
Atr^A - atrybuty obiektu niskiej rozdzielczości,

Atr^B - atrybuty obiektów wysokiej rozdzielczości,

Rel^A - relacje pomiędzy atrybutami obiektu niskiej rozdzielczości,

Rel^B - relacje pomiędzy atrybutami obiektów wysokiej rozdzielczości,

Rel^{AB} - relacje pomiędzy atrybutami należącymi do obiektów różnej rozdzielczości.

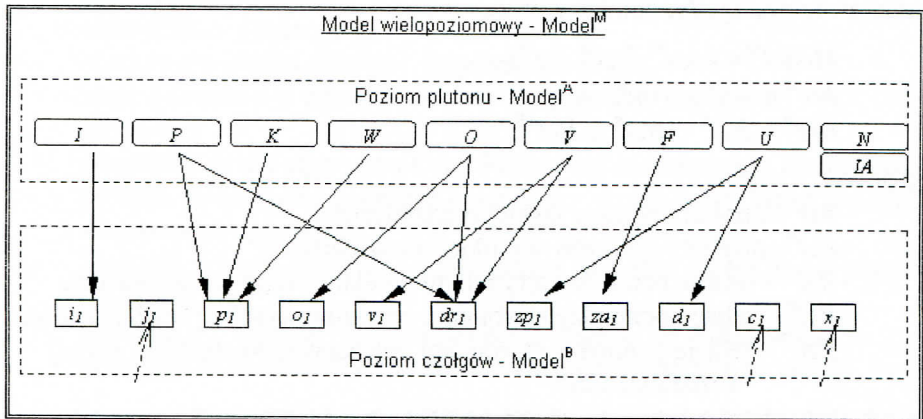


Rys. 4.1. Węzły grafu wielopoziomowego modelu pluton-czołg.

W dalszej części przedstawiony zostanie sposób konstruowania grafu dla zdefiniowanego modelu $Model^M$. Dla ustalenia uwagi można przyjąć, że $Model^A$ odpowiada plutonowi czołgów (poziom plutonu), zaś $Model^B$ – poszczególnym czołgom (poziom czołgu). Model plutonu na każdym ze swoich dwóch

poziomów posiada atrybuty zgodne z wymienionymi w punkcie 4.2.1. Pierwszym krokiem jest przypisanie węzłów do zdefiniowanych atrybutów modelu, co przedstawia rys.4.1. W kroku drugim należy ustalić zależności pomiędzy atrybutami modelu poprzez naniesienie łuków łączących poszczególne węzły (rys.4.2). W systemie bez sprzężenia zwrotnego zależności te są jednokierunkowe i określają zasady zmiany wartości atrybutów $Model^B$ na skutek zmian wartości atrybutów $Model^A$. Zależności pomiędzy atrybutami mogą należeć do jednej z dwóch kategorii:

- Kumulacyjnych,
- Dystrybucyjnych.



Rys. 4.2. Zależności pomiędzy atrybutami modelu.

Zależności kumulacyjne tworzą się wówczas, kiedy kilka atrybutów wpływa na wartość jednego innego atrybutu. Tak jest na przykład z węzłem p_j grafu z rys. 4.2. Jego wartość jest ustalana na podstawie kilku innych atrybutów. Funkcja mapująca musi więc uwzględnić wartość atrybutów P, K i W .

Zależności dystrybucyjne są przeciwieństwem kumulacyjnych. Powstają w sytuacji, kiedy zmiana wartości jednego atrybutu wpływa na wartość kilku innych. Na powyższym rysunku łuki wychodzące na przykład z węzła P lub V reprezentują zależności tej właśnie kategorii. Łuki oznaczone linią przerywaną nie definiują zależności między atrybutami, ale wskazują, że wartość danego atrybutu jest ustalana bezpośrednio jako skutek pewnej akcji. Atrybuty takie nie są uzależnione od wartości innych parametrów. W powyższym przykładzie bezwzględnie ustalone są dane dotyczące ruchomych części czołgu, informacji dodatkowych oraz identyfikator obiektu.

Ostatnim krokiem jest budowa funkcji mapujących, które dokonują transformacji danych pomiędzy atrybutami połączonymi łukami grafu. Dla przykładu funkcja X , na podstawie położenia, kształtu i wymiarów plutonu, wyznacza położenie n czołgów w plutonie w danej chwili t .

$$X: p_i(t) = f(P(t), K(t), W(t)), \quad 1 \leq i \leq n$$

Funkcja Y „rozdziela” siłę ognia plutonu pomiędzy wszystkie aktywne czołgi:

$$Y: z_{a_i}(t) = g(F(t)), \quad 1 \leq i \leq n;$$

$$F(t) = z_{a_1}(t) + z_{a_2}(t) + \dots + z_{a_n}(t)$$

Takie funkcje należy zdefiniować dla każdego atrybutu w *Modelu*^B. Zmiana dowolnego argumentu danej funkcji wymusza wywołanie funkcji i ewentualną zmianę wartości atrybutów zależnych.

4.3.1. Wyznaczanie położenia obiektów w agregacji

Zadanie wyznaczanie położenia obiektów realizowane jest na podstawie współrzędnych punktu centralnego, zorientowania, wymiarów oraz kształtu agregatu, z uwzględnieniem następujących czynników:

- liczby obiektów podległych,
- minimalnej odległości pomiędzy obiektami,
- ukształtowania terenu,
- naturalnych i sztucznych przeszkód terenowych.

Algorytm rozmieszczenia musi uwzględniać poziom agregacji jednostki i realizować funkcje demontowania w sposób rekurencyjny. Przykładowo, batalion czołgów zostanie podzielony na kompanie, kompanie na plutony, a te z kolei na pojedyncze czołgi. Rozmieszczenie obiektów można zrealizować na dwa sposoby:

- stosując matematyczne metody wyznaczania położenia k punktów wokół punktu centralnego [2],
- wykorzystując predefiniowane rozkłady obiektów wewnątrz agregatów.

4.3.1.1. Metody matematyczne

Problem zdefiniowany jest następująco:

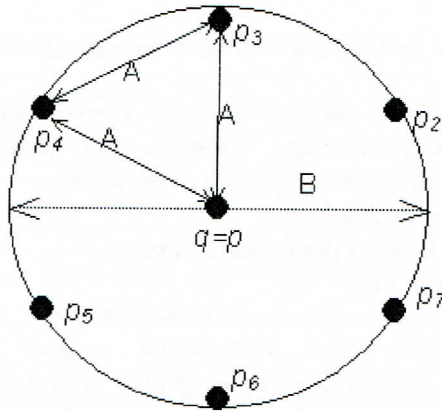
„Mając dany punkt $q \in R^c$, reprezentujący jeden komponent stanu agregatu w danym momencie czasowym t , np. położenie $P^a(t)$, liczbę naturalną $k > 1$ i liczby rzeczywiste A i B , takie że $A > 0$ i $B > A$, znaleźć k punktów $p_1, p_2, \dots, p_k \in R^c$ takich, aby spełnione były następujące warunki:

$$1) \quad \frac{1}{k} \sum_{j=1}^k p_j = q,$$

2) $1 < i < j \leq k, A \leq d(p_i, p_j) \leq B$ dla wszystkich wartości i, j ,

przy czym d oznacza odległość pomiędzy punktami, A - odległość minimalną pomiędzy punktami, B - odległość maksymalną.

Jest to problem o tyle interesujący, że w zależności od wartości danych wejściowych, nie zawsze możliwe jest uzyskanie rozwiązania. Jeżeli różnica wartości A i B będzie mała, natomiast liczba k stosunkowo duża, może się okazać, że nie dla wszystkich punktów znajdzie się miejsce w wyznaczonym okręgu. Przy wartościach B i A spełniających zależność $B=2*A$, w wyznaczonym obszarze można rozmieścić jedynie siedem punktów (rys.4.3.). Jeżeli wartość $k > 7$ zadanie nie ma rozwiązania.

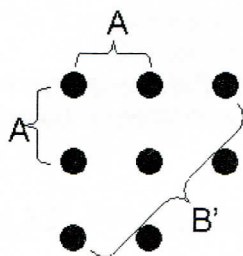


Rys.4.3. Rozwiązanie zadania rozmieszczenia siedmiu punktów dla $B=2*A$.

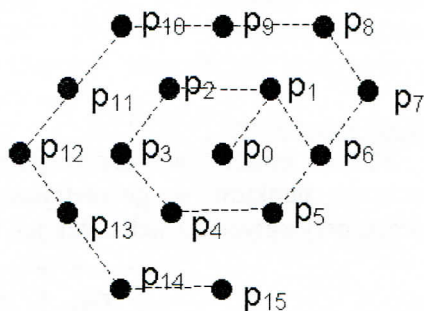
Zakładamy, że istnieje taka wartość B' , gdzie $A \leq B' \leq B$, dla której spełniony jest warunek $1 \leq i < j \leq k, A \leq d(p_i, p_j) \leq B'$, dla wszystkich możliwych wartości i, j . Zadanie sprowadza się do określenia względnie małej liczby B' , przy ustalonych A i k , dla której istnieje rozwiązanie. Jeśli $B' \leq B$, możemy obliczyć położenia punktów, w przeciwnym przypadku - zadanie nie ma rozwiązania. Poniżej zaprezentowane zostały trzy metody znajdowania wartości B' .

A. Metoda siatki

Polega na rozmieszczeniu punktów w macierzy o wymiarach $\sqrt{k} \times \sqrt{k}$ (rys.4.4). Odległość pomiędzy dwoma sąsiednimi punktami w jednym wierszu i w jednej kolumnie wynosi A . Odległość pomiędzy punktami leżącymi na krańcach przekątnej macierzy, będąca maksymalną odległością pomiędzy dowolnymi dwoma punktami wynosi $B' = (A \times \sqrt{k}) \cdot \sqrt{2}$.



Rys. 4.4. Metoda siatki dla $k=8$



Rys. 4.5. Metoda trójkątów dla $k=16$

B. Metoda trójkątów

Kolejne punkty umieszcza się na wierzchołkach trójkątów równobocznych wzajemnie do siebie przyległych. Pierwszy punkt p_1 wyznacza się arbitralnie i może on odpowiadać punktowi q . Następnymi sześć punktów rozmieszcza się na pierwszej, odległej o A , orbicie punktu p_1 , zachowując przy tym żadaną odległość pomiędzy punktami. Kolejne 12 punktów znajdzie się na drugiej orbicie odległej o $2 \times A$, itd. Sposób rozmieszczania 16 punktów na kolejnych orbitach przedstawia rys.4.5.

Dowolne dwa punktu oraz kolejne orbity są oddalone od siebie o A . Wyznaczenie B' sprowadza się do wyznaczenia liczby potrzebnych orbit dla ustalonego k . Niech b oznacza liczbę orbit niezbędnych do rozmieszczenia wszystkich punktów. Na b orbitach można rozłożyć $\sum_{i=1}^{b} 6i$ punktów. Pierwszy punkt nie leży na orbicie, więc sumę punktów należy zwiększyć o jeden. Obliczenie b polega więc na rozwiązaniu następującej nierówności:

$$1 + \sum_{i=1}^b 6i \geq k, \quad k > 1$$

$$1 + \frac{6(b)(b+1)}{2} = 1 + 3b + 3b^2 \geq k$$

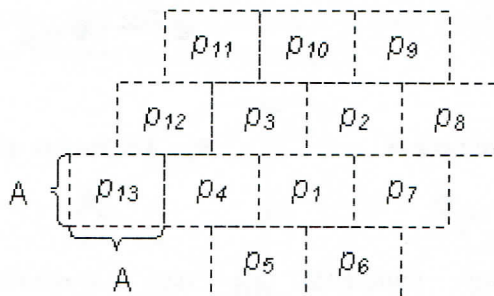
$$b = \left\lceil \frac{-3 + \sqrt{12k - 3}}{6} \right\rceil$$

Nawiasy klamrowe oznaczają zaokrąglenie w górę do liczby całkowitej. Z tego wynika, że dla ustalonego k oraz A , maksymalną odległość B' pomiędzy dwoma punktami można zapisać następująco:

$$B' = 2A \left[\frac{-3 + \sqrt{12k - 3}}{6} \right]$$

C. Metoda kwadratów

Metoda kwadratów – bardzo podobna do metody trójkątów – polega na umieszczaniu punktów w geometrycznych środkach kwadratów o bokach długości A , przyległych do siebie, tak jak na rys.4.6.



Rys. 4.6. Metoda kwadratów dla $k=14$.

Obliczenia są analogiczne jak w metodzie poprzedniej, z tą różnicą, że średnica heksagonu zbudowanego z kwadratów, liczona między ich środkami, wynosi $bA\sqrt{5}$, gdzie b oznacza liczbę warstw nakładanych wokół kwadratu ze środkiem w punkcie p_1 . Tak więc dla ustalonego k , odległość B' pomiędzy dwoma punktami jest nie większa niż $A\sqrt{5} \left[\frac{-3 + \sqrt{12k - 3}}{6} \right]$

4.3.1.2. Rozkłady predefiniowane

Metoda ta polega na wyznaczaniu współrzędnych kolejnych punktów według ściśle określonego geometrycznego wzorca rozkładu. Wzorce rozkładu wybierane są w zależności od charakteru prowadzonych działań przez dany oddział i uwzględniają "ułożenie" oddziału występujące podczas takich akcji jak:

- prowadzenie ataku,
- prowadzenie obrony,
- marsz,
- odpoczynek.

Zdefiniowane modele rozmieszczenia mogą być zapisane w postaci reguł matematycznych lub w postaci siatki punktów. Funkcje rozmieszczania obiektów wyznaczają współrzędne kolejnych punktów na podstawie wzorca oraz danych wejściowych uwzględniających:

- odległości pomiędzy obiektami,
- szerokość ugrupowania,
- głębokość ugrupowania.

4.3.2. Przeszkody terenowe

Przedstawione metody deagregacji nie uwzględniają różnego typu obiektów terenowych i przeszkód, takich jak zbiorniki wodne, drzewa, domy. Wobec tego funkcje deagregacji należy wzbogacić o metody sprawdzające dostępność danego fragmentu terenu. Najpopularniejsze z nich to:

- wyznaczenie na etapie modelowania niedostępnych fragmentów terenu, na przykład w postaci prostokątów lub okręgów obejmujących przeszkodę, i umieszczenie ich opisu w dedykowanej bazie danych,
- przypisanie bitów dostępności poszczególnym wielokątom składowym na etapie modelowania terenu,
- bieżące analizowanie właściwości terenu na podstawie charakterystycznego wyróżnika, na przykład identyfikatora nałożonej w danym miejscu tekstury, identyfikatora wykorzystanego materiału, identyfikatora koloru, co także wymaga odpowiedniego przygotowania bazy danych o terenie.

W przypadku braku możliwości ustawienia obiektu w wyznaczonym miejscu, oblicza się współrzędne kolejnego punktu zgodnie z przyjętą metodą i powtarza procedurę sprawdzającą dostępność terenu.

Opisane metody dotyczą rozmieszczania obiektów w pierwszej fazie deagregacji albo w pierwszym kroku symulacji wirtualnej. Wyznaczanie położenia w kolejnych stanach symulacji odbywa się na zasadzie przesuwania po terenie obiektów zgodnie z nadchodzącymi aktualnymi danymi o położeniu agregatu z uwzględnieniem obecności przeszkód terenowych oraz kształtu i wymiarów agregatu.

4.3.3. Powoływanie i usuwanie obiektów

Zadanie polega na przygotowaniu pakietu zawierającego żądanie powołania lub usunięcia obiektu ze środowiska wirtualnego, skierowane do generatora obrazu. Rekord danych musi zawierać:

- identyfikator strony,
- identyfikator jednostki macierzystej - agregatu.
- identyfikator obiektu,
- rodzaj obiektu.

Powołanie obiektu nie jest jednoznaczne z jego włączeniem do sceny VR. Żądanie usunięcia obiektu powoduje wyłączenie obiektu ze sceny oraz kasowanie jego instancji.

4.3.4. Generowanie efektów walki

Efekty specjalne generowane w środowisku wirtualnego konfliktu zbrojnego zdecydowanie wpływają na atrakcyjność zobrazowania i podnoszą realizm bieżącej sytuacji. Efekty specjalne można podzielić na dwie grupy:

- efekty środowiskowe,
- efekty związane z prowadzonymi działaniami wojennymi.

Efekty środowiskowe typu mgła, oświetlenie, chmury, itd., nie zależą od prowadzonej symulacji. Ustalane są jednokrotnie, podczas inicjalizacji generatora obrazu, na podstawie przyjętych założeń początkowych. W czasie ćwiczenia mogą być zmieniane dowolną ilość razy po ingerencji kierownika ćwiczenia. Efekty walki, w przeciwieństwie do efektów środowiskowych, wynikają bezpośrednio z bieżącego kontekstu symulacji i muszą odpowiadać charakterowi prowadzonych działań. Należą do nich: wybuchy, wystrzały, dym, ogień, ślady zniszczeń terenu i obiektów. Za tworzenie pakietów opisujących kolejne instancje efektów odpowiada część symulacyjna w module sprzęgającym. Przechwytuje pakiety o agregatach i wyłuskuje informacje o rodzaju prowadzonych operacji. Wymagane dane wejściowe dla procedury tworzenia efektów specjalnych to:

- identyfikator typu akcji: obrona, natarcie, marsz,
- liczba jednostek w agregacie,
- zdolność prowadzenia ognia.

Dodatkowo korzysta się z danych o obiektach wysokiej rozdzielczości wyznaczonych przez funkcje mapujące, a są to: identyfikatory j , położenie p , zapas amunicji za i poziom zniszczenia d . Działanie procedury opiera się na następującym algorytmie:

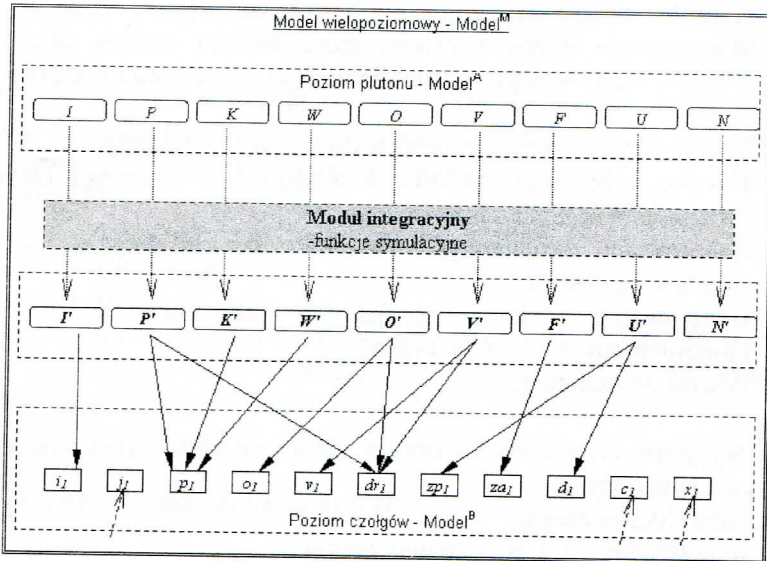
1. Kompletowanie danych.
2. Wyznaczenie liczby strzałów, które zostaną oddane przez oddział w czasie do kolejnego uaktualnienia (okres uaktualniania danych T jest stały).
3. Losowe wyznaczenie momentów czasowych strzelania.
4. Losowe wybieranie obiektów strzelających w kolejnych chwilach po uwzględnieniu kondycji jednostek.
5. Generowanie skojarzonych pakietów danych dotyczących oddania strzału oraz eksplozji.
6. Generowanie efektów specjalnych dla obiektów zniszczonych i unieruchomionych: dym, ogień.
7. Powrót do punktu 1.

Pakiety wysyłane do generatora obrazu opisujące efekty specjalne zawierają następujące pola danych:

- identyfikator efektu,
- współrzędne x,y,z, wystąpienia efektu,
- zorientowanie H,P,R efektu,
- wymiary,
- czas trwania.

4.3.5. Funkcje symulacyjne

Funkcje mapujące wyznaczają kolejne pola rekordów opisujących stany powołanych obiektów. Należy zauważyć, że częstotliwość napływania pakietów danych od agregatów jest zbyt mała, aby zaspokoić wymagania podsystemu zobrazowania. Wobec tego wielopoziomowy model $Model^M$ należy wyposażać w dodatkowe funkcje symulacyjne, których zadaniem będzie zagwarantowanie integralności danych opisujących wszystkie atrybuty modelu. Punktami krytycznymi decydującymi o trafności doboru funkcji symulacyjnych są momenty czasowe, w których następuje aktualizacja danych o agregatach. W chwili nadejścia nowych informacji nie może się okazać, że atrybuty poziomu plutonu i atrybuty poziomu czołgów pozostają w niezgodności. Dobrze zbudowany model musi zapewniać integralność danych wewnętrznych. W tym celu poziomy przedstawione w $Modelu^M$ zostaną połączone modułem integracyjnym wyposażonym w zbiór funkcji symulacyjnych zapewniających odpowiednią częstotliwość uaktualniania danych o agregatach oraz dbających o integralność danych.

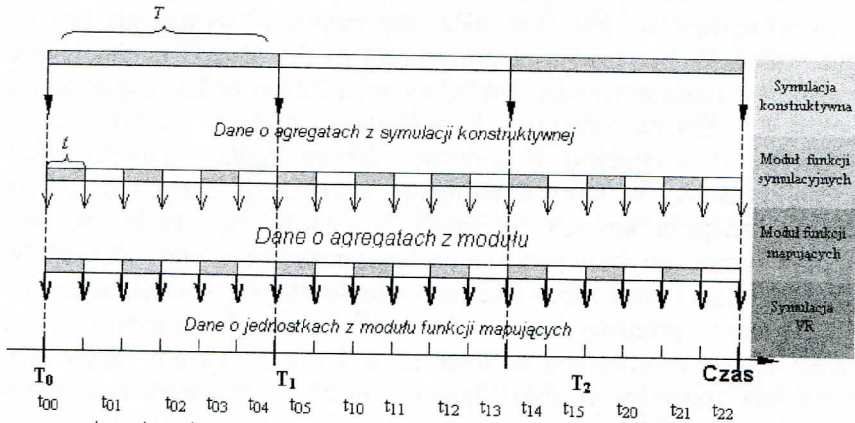


Rys. 4.7. Model wielopoziomowy z modulem integracyjnym.

Rozbudowany model przedstawia rys. 4.7. Zasadę działania modułu integracyjnego z uwzględnieniem ograniczeń czasowych tłumaczy rys.4.8. Na rysunku przyjęto następujące oznaczenia:

T - okres uaktualniania danych z symulacji konstruktywnej,

t - żądany okres uaktualniania danych dla generatora obrazu.

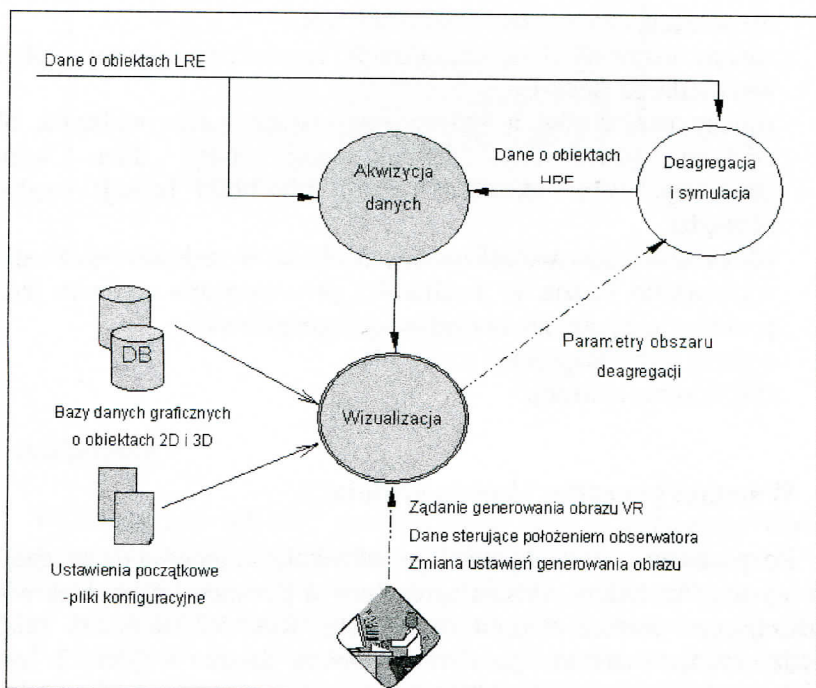


Rys. 4.8. Warstwy modelu wielopoziomowego – uwarunkowania czasowe.

5. Generator obrazu

Generator obrazu stanowi element wykonawczy systemu prezentacji. Powołanie modułu sprzęgającego, implementacja w jego ramach funkcji mapujących i symulacyjnych - wszystko to ma na celu przygotowanie odpowiedniej informacji na potrzeby generatora, który jest pasywnym elementem środowiska symulacji rozproszonej. Może przechwytywać dowolne pakiety danych, zarówno te opisujące jednostki wysokiej jak i niskiej rozdzielczości. Jedynym jego zadaniem jest pobranie, odpowiednie przetworzenie i zobrazowanie informacji w postaci wirtualnego pola walki. Przewiduje się, że w ramach zadania przetwarzania informacji, generator może realizować obliczenia wyznaczania położeń pośrednich według algorytmu *Dead-Reckoning*.

5.1. Rodzaje prezentowanych danych



Rys. 5.1. Przepływ danych w generatorze obrazu.

Generator obrazu, poprzez dowiązany interfejs graficzny, pozostawi użytkownikowi możliwość kompozycji zbioru elementów przeznaczonych do wizualizacji [3]. Lista wyboru obejmuje następujące pozycje:

- wyświetlacz Head-Up pełniący rolę radaru,
- półprzezroczyste symbole 3D agregatów pokrywające powierzchnię zajmowaną przez oddziały wraz ze znakami taktycznymi,
- modele 3D obiektów HRE,
- efekty specjalne,
- informacje dodatkowe, np. bieżący tryb obserwacji, wykresy wydajnościowe aplikacji.

5.2. Interfejs operatora

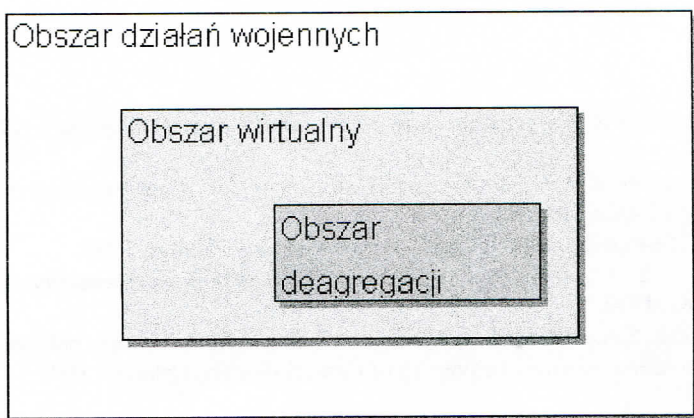
Operator kontroluje przebieg wizualizacji za pomocą poleceń i danych przesyłanych do procesu wizualizacji. Podstawowe akcje inicjowane przez operatora dotyczą:

- rozpoczęcia i zakończenia symulacji VR,
- zmiany ustawień okna wizualizacji: rozmiarów, położenia, obecności wyświetlacza Head-Up,
- zmiany parametrów generowanego obrazu: kątów widzenia, bliższej i dalszej płaszczyzny, ograniczającej, pory dnia, warunków atmosferycznych, obecności tekstur, obecności funkcji wygładzania krawędzi,
- wyboru rodzajów wizualizowanych obiektów (zakłada się, że aplikacja wyposażona będzie w możliwości prezentowania zarówno jednostek poziomego agregatu, jak i pojedynczych obiektów),
- wyboru trybu eksploracji sceny VR,
- zakończenia aplikacji.

5.3. Warunki początku i końca symulacji

Rozpoczęcie symulacji powoduje uaktywnienie procesu deagregacji, przy czym jedynym kryterium wyboru agregatów są parametry ostrośłupa widzenia wyznaczającego granice obszaru deagregacji. Rys. 5.2. tłumaczy zależność pomiędzy różnymi strefami pola walki. Obszar działań wojennych jest tłem prowadzonych przez związki taktyczne symulowanych operacji wojskowych. Tylko część tego obszaru, zwanego obszarem wirtualnym, posiada graficzną reprezentację w postaci modelu lub modeli 3D. W skrajnym przypadku modele 3D mogą jednak pokrywać 100 % teatru działań wojennych. Podzbiorem wirtualnego pola walki jest obszar deagregacji. Jeżeli pole widzenia

obserwatora, choćby w niewielkiej części pokryje dany agregat, inicjowany jest proces pełnej deagregacji obiektu. Symulacja jednostek agregatu trwa nieprzerwanie do chwili wyjścia jednostki z obszaru deagregacji i może być kontynuowana przez ustalony okres czasu stanowiący margines bezpieczeństwa, na wypadek, gdyby agregat po chwili ponownie znalazł się w obszarze deagregacji. Ponowna deagregacja mogłaby diametralnie różnie rozłożyć jednostki w terenie, co wywierałoby wrażenie utraty integralności danych. Warunkiem definitywnie kończącym proces deagregacji jest polecenie zakończenia symulacji VR wydane przez operatora. Jest ono równoważne z usunięciem ze sceny wszystkich modeli 3D związanych z obiektami HRE.



Rys. 5.2. Strefy wirtualnego obszaru działań

6. Zakończenie

W niniejszym artykule przedstawiono sposób połączenia ośrodków symulacji konstruktywnej z systemem graficznego zobrazowania pola walki. Wskazano na szereg istotnych aspektów, których uwzględnienie wydaje się konieczne przy łączeniu systemów korzystających z różnych procedur upływu czasu. Przewidziano wykorzystanie modułu sprzęgającego pełniącego rolę symulatora obiektów wojskowych wchodzących w skład poszczególnych oddziałów i związków taktycznych. Zaproponowano kilka metod wyznaczania położenia obiektów wojskowych w ramach agregatu. Określono koncepcję funkcjonowania generatora obrazu jako pasywnego elementu rozproszonego środowiska symulacyjnego. Funkcjonalne rozdzielenie modułu sprzęgającego

oraz generatora obrazu wynika z przewidywanego rozwoju symulatora pola walki, który powinien prowadzić do łączenia ośrodków symulacji konstruktywnej z symulatorami wirtualnej rzeczywistości oraz aplikacjami CGF (*ang. Computer Generated Forces*). Pojawienie się tego typu aplikacji wyeliminuje konieczność stosowania modułu sprzęgającego, natomiast w żadnym stopniu nie zachwieje pozycji generatora obrazu, dla którego źródła generowania danych są nieistotne. Włączenie do symulatora aplikacji CGF wymagać będzie utworzenia dwukierunkowych interfejsów pomiędzy ośrodkami symulacji konstruktywnej a ośrodkami symulacji w czasie rzeczywistym w celu zapewnienia wzajemnego oddziaływania i zachowania spójności danych. Jest to problem na tyle złożony i zarazem istotny, że może być podstawą wielu opracowań naukowych.

Literatura:

- [1] Franceschini R.W.: *Linking constructive and virtual simulation in DIS*, Orlando, 1994.
- [2] Franceschini R.W.: *Computational strategies for disaggregation*, University of Central Florida, Orlando
- [3] MultiGen-Paradigm, Inc.: *Tactic software module*, Dallas, 1997.
- [4] Natrajan A.: *Consistency maintenance in concurrent representation*, University of Virginia, 1997
- [5] Schricker S.A., Barbosa E.J., Smith R.G.: *Enhancing the realism of pseudo-disaggregated entities*, University of Central Florida, Orlando, 1999

Recenzent: dr hab. inż. Przemysław Rokita
Praca wpłynęła do redakcji 20.11.2001