

Roman KONIECZNY

## NIKTÓRE ASPEKTY ROZWOJU MODUŁU MAPA DLA POTRZEB SYMULATORA RUCHU POCIĄGÓW

Streszczenie. Moduł programowy MAPA jest składnikiem symulatora ruchu pociągów, którego zadaniem jest wizualizacja wyników symulacji na mapie topologicznej wybranego rejonu sieci. Prace związane z realizacją komputerowego symulatora ruchu pociągów prowadzone są w Instytucie Transportu Politechniki Śląskiej od roku 1986 - na zlecenie Instytutu Informatyki Uniwersytetu Warszawskiego w ramach programu RP.I.09 "Rozwój języków, metod oraz podstaw formalnych oprogramowania". Artykuł zawiera skrócone omówienie całości oprogramowania symulatora ze szczególnym uwzględnieniem aktualnej wersji modułu MAPA. Dalsza jego część zawiera wybrane sugestie dotyczące dalszego usprawnienia oraz rozwoju tego modułu.

### 1. Wprowadzenie

Oprogramowanie komputerowego symulatora ruchu pociągów powstało na bazie pracy poświęconej aplikacji języka LOGLAN (na komputer IBM PC), wykonanej przez zespół pod kierunkiem autora w Instytucie Transportu Politechniki Śląskiej na zlecenie Instytutu Informatyki Uniwersytetu Warszawskiego w ramach programu badawczego RP.I.09 "Rozwój języków, metod oraz podstaw formalnych oprogramowania".

Rozwój oprogramowania realizowano kolejnymi 1-rocznymi etapami:

- 1986 - opracowanie koncepcji komputerowego makromodelu ruchu pociągów, rozważania na temat wyboru języka symulacyjnego, wybór języka LOGLAN [1];
- 1987 - badanie walorów funkcjonalnych i użytkowych LOGLANu, opracowanie wstępnej wersji symulatora ruchu pociągów [2];
- 1988 - rozszerzenie symulatora ruchu pociągów, opracowanie modułu MAPA w języku LOGLAN [3];
- 1989 - przeprogramowanie modułu MAPA z języka LOGLAN na język Turbo PASCAL, zorganizowanie pracy makromodelu w wariancie dwukomputerowym [4];
- 1990 - przeprogramowanie symulatora ruchu pociągów z języka LOGLAN na język Turbo PASCAL, zintegrowanie oprogramowania na bazie Turbo PASCALA (wariant jednokomputerowy).

Konieczność przeprogramowania modułów MAPA i SYMULATOR z języka LOGLAN na

język Turbo PASCAL wyniknęła z prozaicznej przyczyny - braku efektywnej implementacji kompilatora LOGLANu. Język programowania LOGLAN w swych zasobach posiada bowiem niezbędne środki do symulacji systemów dyskretnych zdarzeń oraz programowania obiektowego [5,6,7,8], jednakże brak kompilatora zmusił zespół do podjęcia decyzji o zmianie języka programowania. Język Turbo PASCAL wybrano ze względu na duże podobieństwo składniowe do LOGLAN-u oraz możliwość rozłącznej kompilacji modułów. Należy w tym miejscu, zwrócić uwagę na szczególnie dużą użyteczność LOGLAN-u do opracowania projektów programów symulacyjnych różnych systemów, które następnie mogą być implementowane w dowolnym języku programowania, posiadającym efektywny kompilator oraz odpowiednie środowisko.

Dla realizacji całości oprogramowania makromodelu na bazie języka Turbo PASCAL w zupełności wystarczy jeden komputer. Wariant dwukomputerowy, aczkolwiek bardziej interesujący technicznie, posiada ponadto pewne wady, takie jak:

- zbyt duża redundancja danych (część danych z konieczności musi być dublowana w pamięci drugiego komputera),
- rozproszenie funkcji operatora programu symulującego (zbędna konieczność korzystania z dwóch klawiatur i dwóch ekranów),
- strata czasu na komunikację międzykomputerową,
- występujące niekiedy kłopoty techniczne w przypadku łączenia różnych klonów IBM PC na etapie promocyjnym.

Wymienione wady (brak kompilatora LOGLAN-u oraz niedogodności pracy dwukomputerowej) spowodowały, że na początku roku 1990 zdecydowano się zrealizować całość oprogramowania makromodelu w wariantcie jednokomputerowym na bazie języka Turbo PASCAL (wersja 5.0), gdzie moduł MAPA będzie zintegrowany z symulatorem w jednym programie.

Przeprogramowanie symulatora z jednego języka na drugi nie może być jednak celem samym w sobie, dlatego już teraz (nie wnikając w korzyści wynikłe z integracji modułów na jednym komputerze) należy przeanalizować zagadnienie dalszego usprawnienia i rozwoju tego oprogramowania.

## 2. Obecna implementacja modułu MAPA

Moduł MAPA realizuje dwie podstawowe funkcje:

- wyświetlanie obrazu stałego (topologii sieci),
- wyświetlanie obrazu zmiennego (wizualizacja wyników symulacji).

Obraz stały jest zbiorem elementów graficznych, których parametry nie są zmieniane podczas trwania symulacji. Do zbioru tego należą wszelkie napisy, numery oraz zasadniczy szkielet topologii sieci. Obraz zmienny dotyczy pewnych elementów ujętych na mapie, których stan zmienia się w czasie trwania symulacji (np. zmiana stanu semafora, połączenia, relacji, przesuw numeru pociągu).

Dane wejściowe do modułu MAPA przygotowywane są przy użyciu pomocniczego programu PREMAPA (napisanego w języku Turbo PASCAL wersja 3.0). Program PREMAPA, najogólniej rzecz ujmując, przetwarza źródłowy zbiór rekordów opisujących symbole obecne na mapie, na zbiór symboli uproszczonych (w większości wektorów), podlegających wyświetlaniu.

Ogólnie proces ten można zapisać następująco:

$$Z = \langle L^*, T^+, P^+, S^+, F^-, G^-, R^-, M^*, N^- \rangle \xrightarrow[\text{wektoryzacja}]{\text{PREMAPA}}$$

$$Z' = \langle L', P', S', G', R', M', N' \rangle \xrightarrow[\text{grupowanie i sortowanie}]{\text{PREMAPA}}$$

$$\langle L'', P'', S'', G'', R'', M'', N'' \rangle = Z''$$

gdzie:

- Z - plik wejściowy (źródłowy).
- Z' - plik pośredni.
- Z'' - plik wyjściowy (wynikowy).
- L - zbiór odcinków elementarnych.
- T - zbiór tekstów.
- P - zbiór połączeń.
- S - zbiór semaforów.
- F - zbiór semaforów fikcyjnych.
- G - zbiór opisów głowic.
- R - zbiór relacji.
- M - zbiór mikrorelacji.
- N - zbiór napisów i miejsc wyróżnionych na mapie;

\* - oznacza wektor z definicji.

+ - podlega wektoryzacji.

- - nie podlega wektoryzacji.

W szczególności zachodzą następujące relacje:

$$(\forall T^* \in Z) \rightarrow L_T \wedge (\forall P^+ \in Z) \rightarrow L_P + P^+ \wedge (\forall S^+ \vee F^- \in Z) \rightarrow L_S + S^+$$

gdzie  $L$ ,  $L_T$ ,  $L_P$  są odpowiednimi podzbiorami odcinków elementarnych powstałymi z  $T^*$ ,  $P^+$ ,  $S^+$  po procesie wektoryzacji,

znak ' (prim) oznacza stadium nieuporządkowane (zbiór Z'),

znak '' (bis) oznacza stadium po uporządkowaniu (plik wyjściowy Z'').

Po zakończeniu przetwarzania otrzymuje się podzbiór wynikowy:

$$L'' = L'' + L_T'' + L_P'' + L_S''$$

Jest to uporządkowany według współrzędnej X zbiór wektorów. Pogrupowanie i posortowanie rekordów wynikowych ma na celu umożliwienie szybszego wyszukania elementów każdego nowego obrazu na ekranie przy przesuwie mapy.

Rekordy źródłowe (wejściowe dla programu PREMAPA) wpisywane są do pamięci komputera dowolnym edytorem tekstu. Program PREMAPA przetwarza zbiór Z na Z''. Moduł MAPA dokonuje wizualizacji zbioru Z''.

Programy PREMAPA i MAPA zostały szczegółowo opisane w pracach [3] i

[4]. Jako ciekawostkę można zacytować tabelę nr 1, podającą efekty uzyskane po przeniesieniu wersji programu MAPA z języka LOGLAN na Turbo PASCAL wersja 5.0. Porównania dokonano na komputerze IBM PC/XT 4.77 MHz.

Tablica 1

Porównanie czasów wykonania niektórych czynności modułu MAPA

Lp.	Czynność	Czas trwania LOGLAN [s]	Czas trwania PASCAL [s]	Przyspie- szenie
1	Ładowanie pliku z danymi topologicznymi mapy	480	17	28
2	Przeskalowanie całościowe mapy	180	10	18
3	Obliczenie obrazu dla pierwszej ramki ("rozruch") programu	40	4	10
4	Przesuw mapy o 100 punktów	60	4	15
5	Wyszukiwanie (przypadek najgorszy)	195	8	24
6	Wyszukiwanie (typowy przypadek)	122	6	20

Zródło: Opracowanie własne

Proces budowy dowolnej mapy (nie tylko kolejowej) na przedstawionej tutaj zasadzie jest prosty i skuteczny. Jest to niewątpliwą zaletą obu tych programów (MAPA i PREMAPA). Natomiast do wad tego rozwiązania zaliczyć można:

- brak natychmiastowej wizualizacji na ekranie nowo wprowadzonego symbolu.
- brak własnego edytora symboli.
- zbyt duża zajętość pamięci (głównie przez podzbiór L").
- niezadowalająca szybkość budowy obrazu mimo zastosowania algorytmu o koszcie logarytmicznym (wynika to z przyjętej koncepcji - wstępnej wektoryzacji symboli).
- konieczność pracy dwuetapowej (potrzeba korzystania z pomocniczego programu PREMAPA).

### 3. Zagadnienie rozwoju modułu MAPA

Ponieważ budowa map komputerowych jest zagadnieniem ważnym (a istniejące pakiety programowe typu AutoCAD czy OrcAD są drogie i nie zawsze można je bezpośrednio użyć w procesie symulacyjnym, względnie przy wizualizacji stanów innych procesów), dlatego aspekty rozwoju modułu MAPA wydają się być ciągle aktualne.

Podstawowymi usprawnieniami modułu wizualizacji mapy mogą być:

- utworzenie zintegrowanego środowiska programowego służącego do budowy i wizualizacji mapy,
  - rozbudowanie procedur wyszukiwawczych,
  - stworzenie edytora fontów graficznych,
  - przyjęcie koncepcji bieżącej wektoryzacji symboli przy budowie obrazu.
- Koncepcja bieżącej wektoryzacji symboli umożliwia znaczną redukcję danych. Ponadto niepotrzebne już będą pliki pośrednie (czyli użycie programu PREMAPA w tym zakresie).

Problem reprezentacji danych można zilustrować dla wersji mapy [3,4] w postaci pascalowskiego rekordu z wariantami:

```

type typ_rekordu_zr = (L,T,P,S,F,G,R,M,N);
   rekord_zr = record
       case rek: typ_rekordu_zr of
           L: (X1,Y1,X2,Y2,STYL: integer);
           T: (XP,YP,WYS,SZER,ODSTEP,STYL,RA: integer;
              TEKST: string(80));
           P: (CNRPOL,X1,Y1,X2,Y2: integer);
           S,F: (CNRSEM,XP,YP,XR,YR,RSEM: integer);
           G: (CNRGLO,LREL,SLMrel,XP,YP: integer);
           R: (CNRGLO,NRREL,P1,P2,LMrel: integer;
              Mikrorel: array [1..20] of integer);
           M: (CNRGLO,NRMikrorel,X1,Y1,X2,Y2: integer);
           N: (XP,YP: integer; nazwa: string(12));
       end;

```

Należy tutaj zwrócić uwagę, że w rzeczywistej deklaracji dla programu komputerowego wszystkie identyfikatory (nazwy pól) rekordu musiałyby być różne. Stosowanie rekordów z wariantami w języku Turbo PASCAL nie jest zbyt efektywną techniką ze względu na nieoszczędną rezerwację pamięci przez kompilator.

Rekordy źródłowe dla modułu MAPA można zorganizować według następujących założeń:

- zakłada się "nieograniczony" zbiór danych (tj. nie mieszczący się w pamięci wewnętrznej RAM),
- zakłada się rekordy o zmiennej długości,
- zakłada się stosowanie kompresji danych,
- zakłada się operowanie na pamięci RAM podczas rysowania obrazu.

Ogólnie budowa rekordu źródłowego może być następująca:

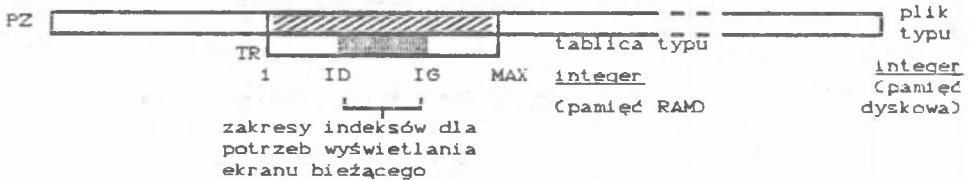


Rekord jest zbiorem liczb typu integer. Typ rekordu jest liczbą ujemną o następujących znaczeniach:

- 10000  $\leq$  Typ < 0 podstawowy (niezależny) rekord źródłowy,
- 20000  $\leq$  Typ < -10000 rekord złożony (struktura),
- 30000  $\leq$  Typ < -20000 rekord składnikowy rekordu złożonego.

Jak widać z powyższego założenia, istnieje możliwość rozróżnienia 10000 typów podstawowych rekordów źródłowych oraz 10000 typów rekordów złożonych. Wyczerpuje to praktycznie potrzeby występujące przy budowie map. Należy również dodać, że typ pierwotny integer umożliwia narysowanie na papierze milimetrowym mapy o wymiarach 32 x 32 metry. Jest to raczej tylko ciekawostka, gdyż praktyczne potrzeby nie idą nigdy tak daleko. Z punktu widzenia wyszukiwania danych rekord złożony jest grupą rekordów składnikowych (bez żadnych innych konsekwencji).

Ideę wyszukiwania danych można zilustrować następującym uproszczonym schematem:



Przyjmuje się, że PZ będzie posortowanym wg kluczy współrzędnych X i Y plikiem źródłowym o dostępie swobodnym o rozmiarze dążącym do "nieskończoności"; oraz że TR będzie tablicą roboczą w pamięci RAM o rozmiarze (1..MAX) gdzie wielkość MAX jest możliwie duża, zależna tylko od sprzętu i od implementacji kompilatora). Tablica TR stanowi kopię pewnego fragmentu pliku PZ; ID i IG są bieżącymi indeksami (dolnym i górnym) w tablicy TR, określającymi zakres indeksowy dla potrzeb wyświetlania ekranu bieżącego. Przyjąć także należy, że MAX jest dużo większe od różnicy IG-ID.

Przy budowie obrazu wyróżnić należy następujące sytuacje:

- IG = MAX (dojście "w prawo" do końca tablicy) - wtedy należy uzupełnić tablicę TR o dane z pliku PZ; można to uczynić np. przez przepisanie "górną" połowę tablicy TR na jej "dolną" połowę, a "górną" połowę uzupełnić danymi przeczytanymi z PZ;
- ID = 1 (dojście w "lewo" do początku tablicy) - wtedy należy "dolną"

połowę tablicy TR przepisać na "górną" a "dolną" uzupełnić odpowiednimi danymi z PZ.

W przypadku wyszukiwania rekordu symbolu o współrzędnych poza tablicą TR należy wykonać zwrot do pliku PZ, odnaleźć szukany rekord i skopiować (wraz z nim) całe jego "połówkowe lewe" i "połówkowe prawe" otoczenie z PZ do TR.

Przyjęte rozwiązanie zapewnia płynne "chodzenie" po dużej mapie bez potrzeby kłopotliwego dzielenia jej na sektory lub strony. W przypadku wyrywkowego przeglądania bardzo dużej mapy (dużej w sensie liczby zamieszczonych symboli a nie np. rozpiętości między największym i najmniejszym X czy Y) można chwilowo zrezygnować z posługiwania się tablicą TR, a dane odczytywać bezpośrednio z pliku PZ. Po zakończeniu wyrywkowego przeglądania mapy, program - korzystając z chwili przerwy - może skopiować wybrany (domniemany) fragment mapy z pliku PZ do tablicy TR. Można przyjąć zasadę, że tablica TR stosowana będzie tylko do przyspieszenia dokładnego oglądu mapy w wybranych "lokalnych" rejonach. W przypadku dużych map pracujących na bieżąco tablicę TR można powiększyć w komputerze na poprzez zastosowanie rozszerzenia pamięci RAM.

Praktycznie dla potrzeb map dużego rejonu sieci kolejowej (np. Śląskiej DOKP) dane zawarte w pliku PZ całkowicie mieszczą się w tablicy TR.

Na rys.1. pokazano problem optymalnego rozmieszczenia rekordów w pliku PZ.

Przy rozpatrywaniu kwestii optymalnego rozmieszczenia rekordów symboli w pliku PZ należy zwrócić uwagę, że dla wybranego skupiska symboli na mapie mogą wystąpić następujące sytuacje:

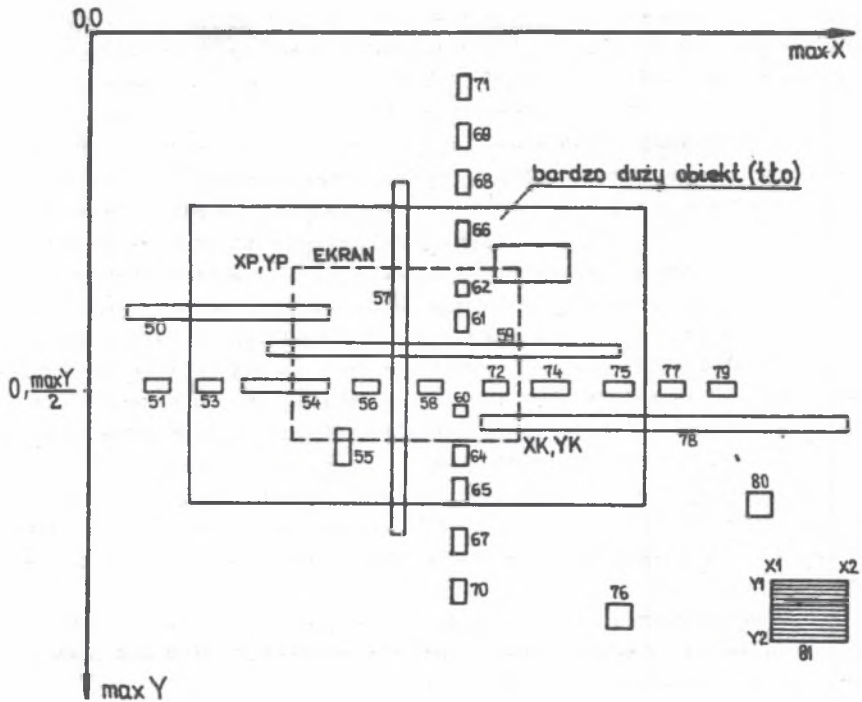
- symbol należy całkowicie do ekranu bieżącego (np. 61),
- symbol należy częściowo do ekranu (np. 50,73),
- symbol "przechodzi" przez ekran (np. 57,59),
- symbol "zakrywa" ekran - jest tak duży, że tworzy tło dla innych symboli (np. 52),
- symbol nakłada się na inny symbol (np. 57,59),
- skupisko symboli jest "rozciągnięte" wzdłuż osi X lub Y.

Rekordy w pliku PZ można posortować według różnych kluczy:

- kolejności wprowadzania,
- odległości środka geometrycznego symbolu lub współrzędnej  $(X_1, Y_1)$  obrysu prostokątnego symbolu od początku układu współrzędnych lub np. od punktu

$$\left(0, \frac{\max Y}{2}\right).$$

Posortowanie pliku PZ ma na celu przyspieszenie wyszukiwania rekordów przy wyświetlaniu ekranu bieżącego. Analizując problem dokładniej stwierdzić można, że występują sytuacje nieoptymalnego wyszukiwania. Przykładowo dla wielu symboli rozciągniętych wzdłuż osi Y, ale posiadających tę samą lub zbliżoną współrzędną X środka geometrycznego, nastąpi niepotrzebne sprawdzenie obecności symbolu na ekranie, choć z góry wiadomo, że pewne grupy symboli i tak będą nieobecne (np. ciąg symboli 71 (Góra) do 70 (dół) -



Rys. 1. Ilustracja problemu optymalnego rozmieszczenia rekordów w pliku PZ  
 Fig. 1. An illustration of optimal lay-out of records in PZ file

zob. rys. 1D).

Proste posortowanie np. według współrzędnej  $X_1$  może czasem grozić niebezpieczeństwem zgubienia (tj. niewyświetlenia) symbolu. Przykładowo: obiekty wcześniej ułożone w pliku PZ (74,75,77) nie należą już do ekranu bieżącego, ale "późniejszy" 78 należy. W algorytmie wyszukiującym, aby nie było takich przeoczeń symboli, należy przyjąć pewną strefę penetracji pliku PZ na tzw. obszar pozaekranowy. W zależności od charakteru mapy wyszukiwanie to może być bardziej lub mniej efektywne. Można też wprowadzić pewien limit maksymalnych rozmiarów pojedynczego symbolu. Założenie takie znakomicie upraszcza wyszukiwanie rekordów, niemniej jest ono nieco sztywne



i kłopotliwe przy wprowadzaniu symboli.

Dobranie właściwego klucza sortowania rekordów w pliku PZ jest kłopotliwe i nie zawsze daje efektywny algorytm wyszukiwania. Można zatem przyjąć inną koncepcję i założyć, że plik PZ nie będzie nigdy sortowany (łatwiej to dopisywanie nowych rekordów), natomiast wystąpią pomocnicze pliki (lub tablice) indeksowe PIX1, PIY1, PIX2 i PIY2, zawierające posortowane indeksy (odsyłacze) do rekordów symboli odpowiednio według współrzędnej  $X1, Y1, X2, Y2$  obrysu prostokątnego symbolu. Ten bardziej wyrafinowany algorytm umożliwi przyspieszenie wyszukiwania odpowiednich rekordów celem wyświetlenia na ekran według atrybutu kolejności wprowadzania do pliku PZ. Pozwoli to na uniknięcie niejednoznaczności przy częściowym lub całkowitym nakładaniu się na siebie obiektów. Zatem odtworzenie dokładnego obrazu mapy będzie zgodne z intencjami budującego mapę i jednocześnie stosunkowo szybkie, przy zmniejszonej zajętości pamięci (pliki lub tablice indeksowe mogą być tworzone na bieżąco).

Można przyjąć, że rekord źródłowy dla takiego wariantu będzie miał następującą budowę :

Typ	X1	Y1	X2	Y2	K	LS	inne parametry
-----	----	----	----	----	---	----	-------------------

gdzie: K - kolejność wprowadzenia do pliku PZ,

LS - liczba słów w rekordzie (zmienna pomocnicza),

pozostałe oznaczenia, jak wyżej.

Warunek przynależności obiektu do ekranu bieżącego zilustrowano na rys. 2.

$$\forall (X1 \vee X2 \in [XP, XK]) \wedge (Y2 \in [YP, \omega]) \vee \forall (Y1 \vee Y2 \in [YP, YK]) \wedge (X2 \in [XP, \omega]) \vee \forall (X1 \in [0, XP]) \wedge (Y1 \in [0, YP]) \wedge (X2 \in [XK, \omega]) \wedge (Y2 \in [YK, \omega]) \rightarrow \text{wyświetlenie.}$$

Obraz na ekranie budowany jest na podstawie przeliczenia:

$$X \in [XP, XK] \rightarrow X_{\text{ekr}} \in [0, \text{Getmax}X]$$

$$Y \in [YP, YK] \rightarrow Y_{\text{ekr}} \in [0, \text{Getmax}Y]$$

gdzie  $\text{Getmax}X$  i  $\text{Getmax}Y$  oznaczają rozdzielczość ekranu (liczbę punktów <pixeli>) odpowiednio dla  $X$  i  $Y$ . Przesuw obrazu polega na nadaniu nowych wartości  $XP \rightarrow XP'$ ,  $YP \rightarrow YP'$ ,  $XK \rightarrow XK'$  i  $YK \rightarrow YK'$  bez zmiany wartości różnic  $XK - XP$  i  $YK - YP$ . Przeskalowanie obrazu polega na zmianie wartości różnic  $XK - XP$  i  $YK - YP$ .

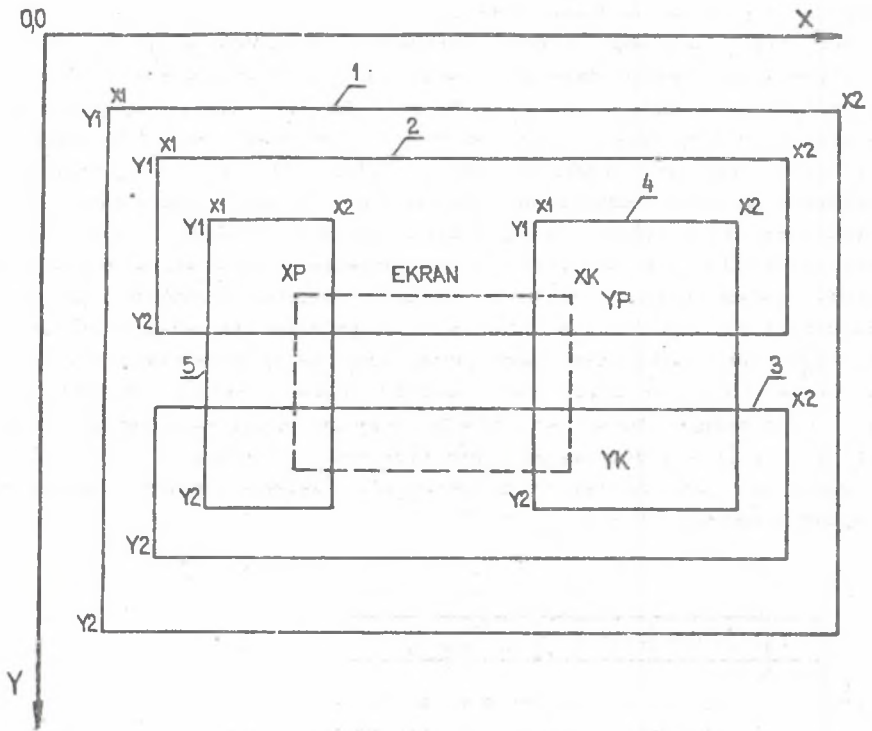
Po przeskalowaniu nastąpi przeliczenie obrazu:

$$X' \in [XP', XK'] \rightarrow X_{\text{ekr}} \in [0, \text{Getmax}X]$$

$$Y' \in [YP', YK'] \rightarrow Y_{\text{ekr}} \in [0, \text{Getmax}Y]$$

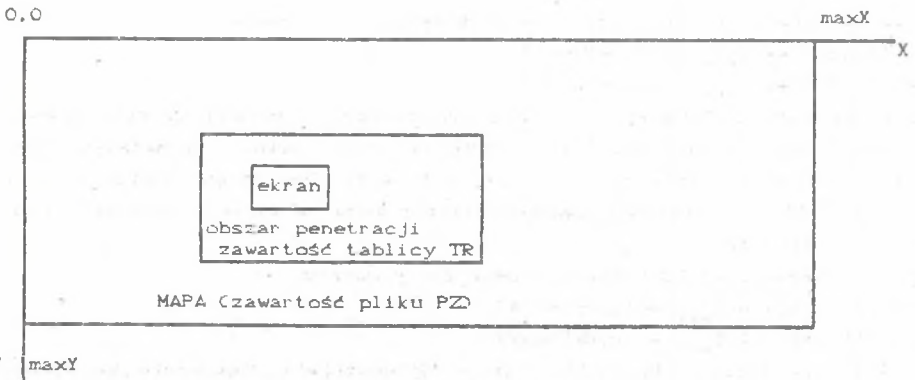
Powiększenie różnicy  $XK - XP$  i  $YK - YP$  umożliwia ulokowanie na ekranie więcej szczegółów mapy przy jednoczesnym "zmniejszeniu" obrazu.

Wykorzystanie pomocniczej tablicy TR w rozpatrywanym wariantcie może



Rys.2. Ilustracja problemu przynależności obiektu do ekranu  
 Fig.2. A problem illustration of object affiliation to screen

być zorganizowane poprzez określenie pozaekranowego obszaru penetracji mapy (rys.3), limitowanego tylko rozmiarem tej tablicy.



Rys.3. Wyjaśnienie określenia "obszar penetracji mapy"  
 Fig.3. An explaining of "map penetration area" term

Odpowiednie rekordy mogą być przepisane do TR według wskazań plików indeksowych PIX1, PIY1, PIX2 i PIY2. Gdy ekran dojdzie do któregoś z brzegów obszaru penetracji, można wtedy określić nowy zakres tego obszaru i zaktualizować tablicę TR. Przedstawione rozwiązanie również umożliwia płynne "chodzenie" po dowolnej płaskiej mapie bez względu na rozmieszczenie skupisk szczegółów.

Struktura pliku indeksowego jest prosta. Podstawowymi elementami tego pliku są pary <współrzędna: X1  $\vee$  Y1  $\vee$  X2  $\vee$  Y2, indeks (odsylacz)>. Oprócz podstawowych plików indeksowych mogą być również utworzone pomocnicze pliki indeksowe zawierające wyłącznie odsyłacze dla wybranych symboli na mapie, np. dla mapy kolejowej: semaforów, głowic, połączeń itd.

Uzupełniając rozważania dotyczące rozwoju modułu MAPA należy wspomnieć jeszcze o trzech aspektach:

- edycji symboli,
- umieszczaniu symboli,
- wyszukiwaniu symboli.

Symbole mogą być podzielone na trzy grupy:

- elementarne (np. okrag, odcinek, prostokąt... z repertuaru biblioteki graficznej Turbo PASCAL-a są spline, wielobok... z biblioteki użytkownika),
- podstawowe (składające się z symboli elementarnych, np. semafor, czy połączenie na mapie kolejowej),
- złożone (składające się z symboli podstawowych ewentualnie uzupełnione symbolami elementarnymi, np. grupa torów, stacja, głowica).

Symbol można tworzyć na bazie fontu kreskowego (wektorowego) lub punktowego (matrycowego) względnie obu naraz. Opcje budowy symboli nie powinny odbiegać od znanych w tym zakresie standardów. Powinny uwzględniać obsługę "myszy", rozwijalne menu, względnie sterowanie ikonami. Powinny dawać możliwość określania rozmiarów symbolu (powiększanie, obrót itd.) oraz dawać możliwość modyfikacji jego parametrów. Utworzone symbole powinny być gromadzone w pliku bibliotecznym.

Przy umieszczaniu symboli na mapie powinna być możliwość: wyświetlania pomocniczego fragmentu siatki współrzędnych, przesuwu symbolu, obrotu, modyfikacji, a także kopiowania względnie usuwania. Można także przewidzieć opcję automatycznego rozmieszczania symboli, a w dalszej przyszłości możliwość generowania różnych wariantów map względnie ich fragmentów.

Opcja wyszukiwania, na podstawie pomocniczych plików indeksowych, powinna umożliwiać odnalezienie symbolu dowolnego typu na mapie, a także dokonanie zestawienia (statystyki) symboli, wyszukiwania grup symboli itd.

#### 4. Uwagi końcowe

Tworząc nowe projekty rozwojowe modułu MAPA należy ciągle mieć na względzie istniejący ustawiczny duży postęp w tego typu standardowym profesjonalnym oprogramowaniu. Idee i koncepcje rozszerzenia modułu MAPA nie powinny być nigdy wtórne, a jedynie nawiązujące do najlepszych rozwiązań światowych w tym zakresie (względnie je wycinkowo wzbogacające), przy uwzględnieniu realności krajowych potrzeb oraz możliwości zastosowań.

Jak pokazały dotychczasowe doświadczenia z modułem programowym MAPA, a także inne eksperymenty programowe, przeprowadzone w Instytucie Transportu Politechniki Śląskiej, w zakresie podstawowym, tj. wyszukiwania symboli i budowy obrazu, osiągnięto rezultaty porównywalne (a nawet nieco korzystniejsze) w stosunku do takich standardów, jak AutoCAD czy OrcAD. Natomiast z punktu widzenia przyszłego użytkownika przedstawione oprogramowanie jest znacznie tańsze, nie podlega licencji i jest łatwiejsze do adaptacji pod kątem wybranych zastosowań.

#### LITERATURA

- [1] - KONIECZNY R. z zespołem (praca zbiorowa): Opracowanie koncepcji makroskopowego modelu symulacyjnego ruchu pociągów. Katowice 1986 (maszynopis pracy naukowo-badawczej wykonanej w Instytucie Transportu Politechniki Śląskiej w ramach programu RP.I.09).
- [2] - KONIECZNY R. z zespołem (praca zbiorowa): Zastosowanie języka LOGLAN do modelowania dużych systemów transportowych na przykładzie modelu ruchu pociągów. Katowice 1987 (maszynopis pracy naukowo-badawczej wykonanej w Instytucie Transportu Politechniki Śląskiej w ramach programu RP.I.09).
- [3] - KONIECZNY R. z zespołem (praca zbiorowa): Zastosowanie języka LOGLAN do modelowania dużych systemów transportowych na przykładzie modelu ruchu pociągów - etap II. Katowice 1988 (maszynopis pracy naukowo-badawczej wykonanej w Instytucie Transportu Politechniki Śląskiej w ramach programu RP.I.09).
- [4] - KONIECZNY R. z zespołem (praca zbiorowa): Zastosowanie języka LOGLAN do modelowania dużych systemów transportowych na przykładzie modelu ruchu pociągów - etap III. Katowice 1989 (maszynopis pracy naukowo-badawczej wykonanej w Instytucie Transportu Politechniki Śląskiej w ramach programu RP.I.09).
- [5] - KONIECZNY R.: Charakterystyka zasobów symulacyjnych języka LOGLAN na potrzeby modelowania systemów transportowych. Zeszyt Naukowy Politechniki Śląskiej, s. Transport nr 9, Gliwice 1989 cz. I i II.
- [6] - KONIECZNY R.: Zasoby symulacyjne języka LOGLAN. Zeszyt Naukowy Politechniki Śląskiej, s. Transport nr 13, Gliwice 1989.
- [7] - KONIECZNY R.: Przykłady rozwiązania problemów symulacyjnych w języku LOGLAN. Zeszyt Naukowy Politechniki Śląskiej, s. Transport nr 13, Gliwice 1989.
- [8] - KONIECZNY R.: Zagadnienie rozszerzenia loglanowskich zasobów symulacyjnych. Zeszyt Naukowy Politechniki Śląskiej, s. Transport nr 13 Gliwice 1989.

## SOME ASPECTS OF MAPA MODULE DEVELOPMENT FOR RAILWAY TRAFFIC SIMULATION

## Summary

MAPA programme module is a part of railway traffic simulator and it vizualizes simulation data on topographic map of the selected railway system area.

Research on computer railway traffic simulator has been carried out at the Institute of Transport of the Silesian Technical University since 1986 by order of the Institute of Computer Science at Warsaw University. This is a part of RP.I.09 project ("Languages, methods and development of methods and formal basic software"). The paper contains general account of simulator's software and MAPA module present version in particular. There are also some suggestions on further rationalization and development of the module.

## EINIGE ENTWICKLUNGSASPEKTE DES MAPA-PROGRAMMODULS FÜR DEN ZUGVERKEHRSSIMULATOR

## Zusammenfassung

Das MAPA - Programmmodul stellt ein Bestandteil des Zugverkehrssimulators dar. Die Aufgabe des Moduls besteht in der Visualisierung von Simulationsergebnissen auf der topologischen Karte ausgewähltes Bahnnetzregions. Die Forschungsarbeiten, die mit der Realisierung des Zugverkehrssimulators auf dem Rechner verbunden sind, laufen im Transport-Institut der Schlesischen Technischen Universität seit 1986. Sie werden im Auftrag des Informatik-Institutes der Warschauer Universität im Rahmen des Forschungsprogramms Nr. RP.I.09: "Entwicklung von Sprachen, Methoden sowie formalen Grundlagen der Software". Der Aufsatz beinhaltet in gekürzter Fassung die Besprechung der Simulatorsoftware, unter der besonderen Berücksichtigung aktueller Version des MAPA - Moduls. Im weiteren hat er ausgewählte Suggestionen bezüglich der weiteren Entwicklung und Vervollkommnung dieses Moduls zum Inhalt.

## НЕКОТОРЫЕ ОСОБЕННОСТИ РАЗВИТИЯ МОДУЛЯ МАРА НЕОБХОДИМОЕ МОДЕЛИРОВАНИЯ ДВИЖЕНИЯ Поездов

### Резюме

Программная модуль МАРА является составляющей модели движения поездов которая имеет своей целью проширивание результатов моделирования на топологической карте района железнодорожной сети. Работы связанные с компьютерной реализацией модели движения поездов ведутся на Кафедре Транспорта Силезского Политехнического Института с 1986 года по заказу Кафедры Информатики Варшавского Университета как часть программы RP. I.09 "Развитие языков, методов и формальных основ программирования". В статье помещено сокращенное описание всех программ модели с особым учетом актуального варианта модули МАРА.

Содержит она также избранные предположения касающиеся большего развития этой модули.