

Rozdział 5

Pakiet MMNET

Paweł CHRZANOWSKI

Realizując przyjętą koncepcję szkieletowego systemu doradczego, opracowano w środowisku R pakiet MMNET. Uznano, że technika programowania obiektowego w języku R będzie najlepiej nadawała się do opracowania kodów źródłowych tego systemu. Ponadto technika obiektowa umożliwia łatwy i elastyczny rozwój oprogramowania pod względem funkcjonalności jak też przez wielu niezależnych programistów.

5.1. Opis ogólny

Przyjęto, iż pakiet MMNET będzie zawierał zbiór klas, metod oraz procedur wraz z dokumentacją i przykładami. Wyodrębniono następujące funkcjonalności pakietu MMNET:

- definiowanie i edycja stwierdzeń prostych i stwierdzeń złożonych,
- definiowanie, edycja i zarządzanie zbiorem stwierdzeń (tezauresem),
- definiowanie, edycja i zarządzanie modelem wieloskalowej sieci stwierdzeń,
- definiowanie i edycja warstwy modelu,
- łączenie warstw poprzez definiowanie węzłów wspólnych,
- obliczanie sieci w zależności od jej typu (np. Bayesa, Markova, itp.),
- agregacja wyników pomiędzy warstwami modelu w oparciu o przyjęte algorytmy,
- wnioskowanie z zastosowaniem danych zewnętrznych,
- interfejs użytkownika.

Dla potrzeb realizacji niektórych funkcjonalności wykorzystano istniejące i szeroko dostępne pakiety środowiska R, które służą między innymi do wymiany danych, realizacji procesu wnioskowania oraz do działania opracowanego graficznego interfejsu użytkownika (GUI).

Przyjęto, że podstawą formą zapisu wiedzy są stwierdzenia proste, które mogą być elementami stwierdzeń złożonych. Zbiór stwierdzeń prostych oraz stwierdzeń złożonych stanowi słownik stwierdzeń tzw. tezaurus. Zarządzanie stwierdzeniami realizowane jest poprzez opracowany przez autora plik wymiany danych w formacie XML. Format ten znacznie ułatwia możliwość podglądu jak i edycję tezaury w dowolnym programie do obsługi plików tekstowych.

W budowanym modelu tezaurus jest elementem bazowym. Model, podobnie jak tezaurus jest zapisywany do formatu XML, gdzie w modelu zawarty jest jedynie stosowany fragment tezaury bazowego oraz opis struktury modelu. Struktura modelu zawiera definicję warstw,

gdzie każda warstwa to odrębna sieć zbudowana na zbiorze stwierdzeń tezaurya bazowego. Szczególnym przypadkiem modelu wielowarstwowego jest model jednowarstwowy.

Pakiet MMNET umożliwia wykonywanie obliczeń zdefiniowanego modelu dla każdej warstwy osobno oraz agregację uzyskanych wyników w poszczególnych warstwach. Wyniki agregacji zamieszczane są w tzw. warstwie zerowej.

Proces wnioskowania może być realizowany poprzez wprowadzanie danych z użyciem graficznego interfejsu użytkownika bądź poprzez zbiór danych zewnętrznych zapisanych w pliku tekstowym.

5.2. Wymiana danych w formacie XML

Dla potrzeb archiwizacji oraz wymiany danych zdefiniowano format pliku oparty na standardzie XML (<http://www.w3.org/XML/>). Powodem zastosowania standardu XML była jego prostota implementacji, przejrzystość pliku źródłowego, dzięki czemu możliwy jest jego podgląd oraz edytowanie z użyciem dowolnego edytora tekstu. Szczególną zaletą przyjęcia standardu XML jest możliwość rozszerzania formatu o nowe elementy wraz z rozwojem systemu MMNET.

Jako podstawy element struktury formatu pliku przyjęto tezaurus (klucz: *THESAURUS*) składający się z definicji stwierdzeń (klucz: *STATEMENT*). Każde stwierdzenie zawiera między innymi typ stwierdzenia (klucz: *STMTYPE*), identyfikator stwierdzenia (klucz: *ID*), treść stwierdzenia (klucz: *CONTENT*) oraz definicję nazw wartości (klucz: *VALNAMES*). Fragment pliku zawierającego przykładową definicję tezaurya przedstawiono poniżej.

```
<?xml version="1.0"?>
<MMNET VERSION="1.0">
  <STATICPROPERTIES>
    <source value="Core ver. 0.5, Pawel Chrzanowski (KPKM)"/>
    <date value="Mon Dec 6 13:36:44 2010"/>
  </STATICPROPERTIES>
  <THESAURUS stmts="5" notice="Write notice, please" title="Thesaurus ..." idThesa="0">
    <STATEMENT>
      <STMTYPE>CStmtValMBelief_TR</STMTYPE>
      <ID>10</ID>
      <MODE>1</MODE>
      <DESCRIPTION>belief value</DESCRIPTION>
      <LAYERS>0</LAYERS>
      <SHORTNAME>none</SHORTNAME>
      <CONTENT>Stwierdzenie nr 1</CONTENT>
      <HELP>Pomoc dla stwierdzenia nr 1</HELP>
      <VSTMTID></VSTMTID>
      <VALNAMES>
        <VALNAME>yes</VALNAME>
        <VALNAME>no</VALNAME>
      </VALNAMES>
    </STATEMENT>
    <STATEMENT>
      <STMTYPE>CStmtValMBelief_TR</STMTYPE>
      <ID>20</ID>
      <MODE>1</MODE>
      <DESCRIPTION>belief value</DESCRIPTION>
      <LAYERS>0</LAYERS>
      <SHORTNAME>none</SHORTNAME>
      <CONTENT>Stwierdzenie nr 2</CONTENT>
      <HELP>Pomoc dla stwierdzenia nr 2</HELP>
      <VSTMTID></VSTMTID>
      <VALNAMES>
        <VALNAME>yes</VALNAME>
        <VALNAME>no</VALNAME>
      </VALNAMES>
    </STATEMENT>
    .....
  </THESAURUS>
</MMNET>
```

Bazując na definicji tezaury, opracowano format zapisu modelu MMNET w standardzie XML. Przyjęto, że format ten będzie zawierał tezaurus o strukturze opisanej powyżej oraz definicję modelu. W strukturze pliku, definicja modelu (klucz: *NETWORK*) zawiera między innymi: identyfikator sieci (klucz: *IDNET*), nazwę modelu (klucz: *NAME*) oraz definicję warstw (klucz: *LAYERS*). W definicji warstwy (klucz: *LAYER*) zamieszczono listę węzłów (klucz: *NODES*) w postaci identyfikatorów stwierdzeń (patrz: identyfikator stwierdzenia), definicje krawędzi (klucz: *EDGES*) wraz z parametrami (klucz: *ARC*) zawierającymi identyfikatory węzłów, zwrot krawędzi (1- od N1 do N2, 2- od N2 do N1, 3-zwrot nieokreślony) oraz przypisane do węzłów tablice wartości (klucz: *NODETABLES*). Fragment pliku zawierającego przykładową definicję modelu zamieszczono poniżej.

```
<?xml version="1.0"?>
<MMNET VERSION="1.0">
  <STATICPROPERTIES>
    <source value="Core ver. 0.5, Pawel Chrzanowski (KPKM)"/>
    <date value="Mon Dec 6 13:40:04 2010"/>
  </STATICPROPERTIES>
  <THESAURUS stmts="5" notice="Write notice, please" title="Thesaurus ..." idThesa="0">
    <STATEMENT>
      .....
    </STATEMENT>
  </THESAURUS>
  <NETWORK>
    <IDNET>1</IDNET>
    <NAME>CNetBayes</NAME>
    <NOTICE>none</NOTICE>
    <LAYERS nrLayers="2">
      <LAYER>
        <IDLAYER>1</IDLAYER>
        <NOTICE>none</NOTICE>
        <WEIGHT>1</WEIGHT>
        <NODES>10 20 70 40</NODES>
        <EDGEVAL></EDGEVAL>
        <EDGES nrEdges="4">
          <ARC N1="10" N2="20" DIRECTION="1"/>
          <ARC N1="10" N2="70" DIRECTION="1"/>
          <ARC N1="20" N2="40" DIRECTION="1"/>
          <ARC N1="70" N2="40" DIRECTION="1"/>
        </EDGES>
        <NODETABLES nrnodeTables="4">
          <NODETABLE node="10" nrows="1">
            <ROW>50 50</ROW>
          </NODETABLE>
          <NODETABLE node="20" nrows="2">
            <ROW>10 90</ROW>
            <ROW>50 50</ROW>
          </NODETABLE>
          <NODETABLE node="70" nrows="2">
            <ROW>80 20</ROW>
            <ROW>70 30</ROW>
          </NODETABLE>
          <NODETABLE node="40" nrows="4">
            <ROW>99 1</ROW>
            <ROW>90 10</ROW>
            <ROW>90 10</ROW>
            <ROW>0 100</ROW>
          </NODETABLE>
        </NODETABLES>
      </LAYER>
      <LAYER>
        .....
      </LAYER>
    </LAYERS>
  </NETWORK>
</MMNET>
```

5.3. Wymagania

Ze względu na dynamiczny rozwój środowiska R i nie zawsze aktualnych pakietów dodatkowych stosowanych przez pakiet MMNET podjęto decyzję o użyciu wyłącznie wersji środowiska R 2.10.1 (<http://www.r-project.org/>). Zastosowane pakiety dodatkowe niezbędne do działania systemu MMNET to:

- biblioteka GTK+ (<http://www.gtk.org/>).
- pakiety środowiska R:
 - gWidgets [5.11],
 - gWidgetsRGtk2 [5.7],
 - igraph [5.4],
 - linprog [5.2],
 - matlab [5.3],
 - gRain [5.5],
 - bnlearn [5.8],
 - BMN [5.6],
 - XML [5.10].

5.4. Budowanie pakietu

Środowisko R umożliwia budowanie pakietów w oparciu o zbiór własnych kodów źródłowych, dzięki czemu możliwe jest łatwe dystrybuowanie pakietów. W celu wygenerowania pakietu instalacyjnego dla środowiska R (w tym przypadku) w systemie Windows, należy uprzednio zainstalować odpowiednie oprogramowanie oraz przygotować system Windows. Wszelkie szczegółowe informacje dotyczące budowania i weryfikacji pakietu dla środowiska R zostały opisane w <http://win-builder.r-project.org>.

Przygotowanie systemu Windows wymaga starannej realizacji działań. W tym celu należy:

1. Zapoznać się z dokumentacją opisującą narzędzie dla Windows (<http://cran.r-project.org/doc/manuals/R-admin.html#The-Windows-toolset>) w podręczniku administratora R.
2. Przejść na stronę (<http://www.murdoch-sutherland.com/Rtools/>)
3. Pobrać i zainstalować na komputerze pakiet **Rtools.exe**, odpowiedni do wersji środowiska R. Pakiet ten zawiera:
 - zbiór narzędzi uruchamianych z linii poleceń,
 - Perl,
 - kompilator MinGW,
4. Zainstalować dodatkowo:
 - InnoSetup installer (<http://jrsoftware.org/>)
 - MikTeX (<http://www.miktex.org/>),
 - Microsoft HTML Help Workshop (<http://msdn.microsoft.com/en-us/library/ms669985.aspx>) (nie jest wymagane),

5. Zweryfikować czy w zmiennej systemowej **PATH** istnieje wpis do:
`c:/Rtools/bin;c:/Rtools/perl/bin;c:/Rtools/MinGW/bin;c:/R/bin`
 (Uwaga: ścieżki należy podać zgodnie z lokalizacją oprogramowania).
6. Dokonać wpisu w **PATH** (patrz: Panel Sterowania – > System – > Zaawansowane ustawienia systemu – > Zmienne środowiskowe)
7. Niepoprawne wpisy w **PATH** są przyczyną nieprawidłowego działania procedury generowania i weryfikacji pakietu.
8. zrestartować komputer.

Następnie można przystąpić do przygotowania źródłowej wersji pakietu. W tym celu należy zdefiniować bazową strukturę folderów (tab.5.1) oraz plik *DESCRIPTION* w formacie tekstowym [5.9] zawierający podstawowe parametry użyte podczas generowania pakietu.

Tab. 5.1: Struktura folderów wersji źródłowej pakietu MMNET

| Folder | Zawartość | Uwagi |
|--------|------------------------------|---|
| główny | plik DESCRIPTION | podstawowy plik, zawiera parametry kompilacji |
| R | pliki źródłowe pakietu | *.R |
| man | pliki źródłowe dokumentacji | *.rd |
| data | pliki z przykładowymi danymi | np. tezaurs, sieć, *.XML |

Budowanie pakietu w wersji binarnej należy przeprowadzić wg następującej kolejności:

- przygotować wersję źródłową w folderze np. `c:/Moje_Pakiety/MMNET`,
- uruchomić wiersz poleceń systemu Windows,
- przejść do katalogu `c:/Moje_Pakiety/`
- wpisać polecenie: `RCMD build --binary MMNET`
- zostanie wygenerowany pakiet o nazwie ustalonej w pliku DESCRIPTION np. `MMNET_0.5.2`

Weryfikację poprawności zbudowanego pakietu należy:

- przejść do katalogu `c:/Moj_Pakiety/`
- wpisać polecenie: `RCMD check MMNET`
- zweryfikować treść pliku log zawierającego wyniki weryfikacji,
- w przypadku powstania ostrzeżeń, starać się je wyeliminować wprowadzając w wersji źródłowej ewentualne poprawki i ponownie wykonać budowanie i weryfikację pakietu.

5.5. Numerowanie wersji

Przyjęto, że numeracja wersji oprogramowania będzie składała się z trzech poziomów, gdzie każdy zapisywany jest liczbą całkowitą od zera. Poziomy te oddzielone są kropką wg formatu `< poziom1.poziom2.poziom3 >` np. 0.5.1. Pierwszy poziom oznacza numer główny wersji i dotyczy będzie znacznych zmian w oprogramowaniu (wstępna oznaczona jest cyfrą zero). Drugi poziom oznacza podwersję zawierającą zmiany, będące rozszerzeniem wersji poprzedniej. Natomiast poziom trzeci dotyczy niewielkich (kosmetycznych) zmian wersji poprzedniej np. poprawiona stabilność działania pakietu.

5.6. Instalacja i pierwsze uruchomienie

Instalację pakietu MMNET należy przeprowadzić z poziomu środowiska R wg następującej procedury:

1. Uruchomić środowisko R (w wersji GUI).
2. Zainstalować pakiet MMNET z lokalnych zasobów (*Packages – > Install package(s) from local zip files ...*) - nie wolno zmieniać nazwy pakietu.
3. Załadować pakiet MMNET z menu *Packages– >Load package*
4. W linii poleceń środowiska R wpisać polecenie uruchamiające interfejs graficzny pakietu: **MMNET_gui()**. Jeżeli dodatkowe pakiety wskazane powyżej oraz biblioteka GTK+ nie są zainstalowane, wówczas rozpocznie się automatyczna instalacja tych pakietów ze wskazanego repozytorium środowiska R. Po instalacji GTK+ należy zrestartować komputer, a następnie ponownie załadować (aktywować) bibliotekę MMNET oraz uruchomić interfejs poleceniem **MMNET_gui()**.
5. W przypadku problemów z automatycznym zainstalowaniem dodatkowych pakietów, można przeprowadzić ich ręczną instalację z dowolnego repozytorium (*Packages – > Install package(s)*).

5.7. Aktualizacja pakietu

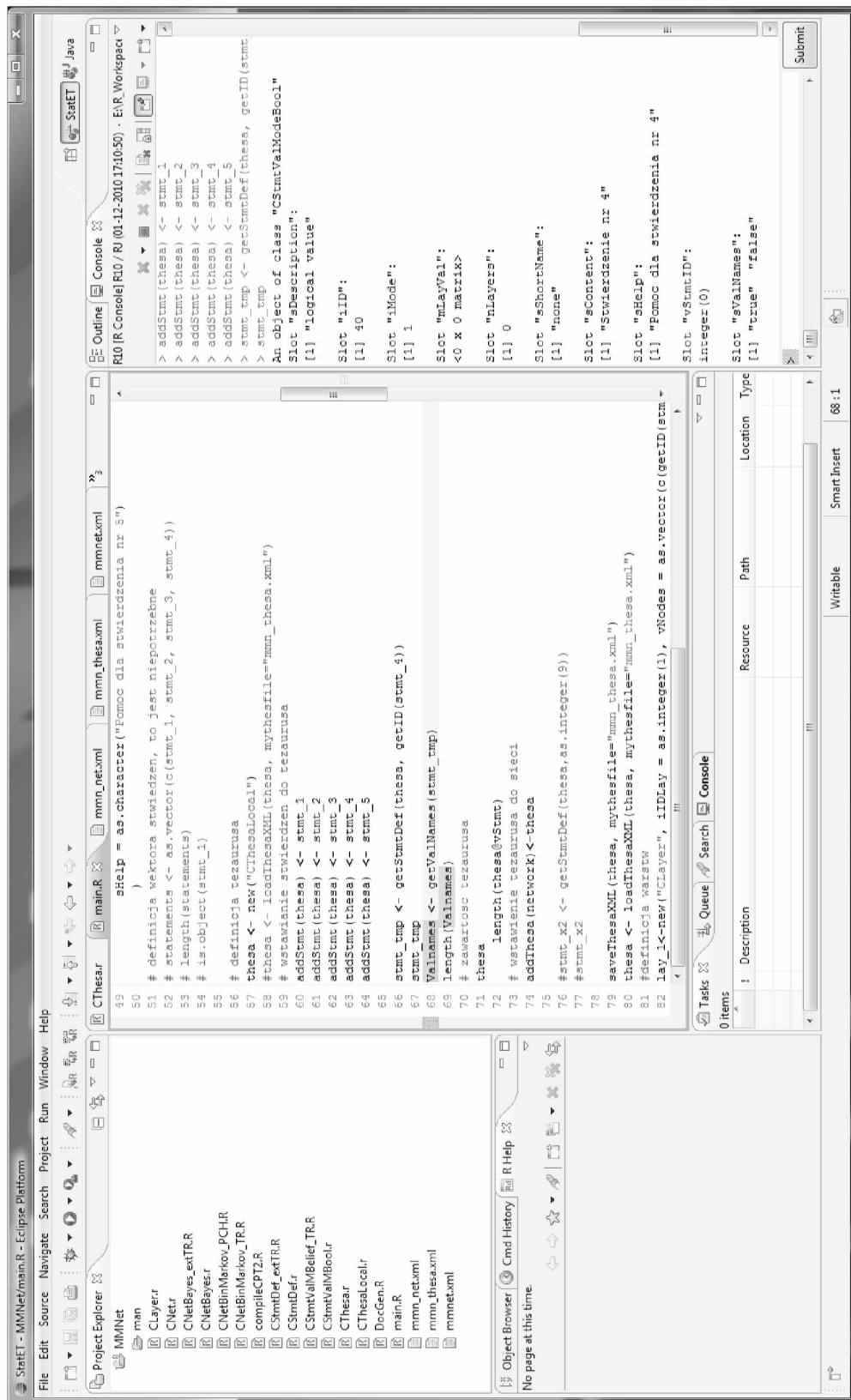
W przypadku aktualizacji pakietu MMNET, w pierwszej kolejności należy:

- Przeprowadzić deinstalację poprzedniej wersji pakietu MMNET (polecenie: *remove.packages("MMNET",lib=.libPaths())*).
- Zrestartować R (przy zamykaniu środowiska R, nie zapisywać zmiennych środowiskowych).
- Uruchomić środowisko R i zainstalować nową wersję pakietu MMNET z lokalnych zasobów (*Packages – > Install package(s) from local zip files ...*).
- Załadować pakiet MMNET z menu *Packages– >Load package*

5.8. Stosowanie pakietu MMNET bez GUI

Pakiet MMNET może być stosowany bez użycia własnego interfejsu graficznego. Opracowany interfejs graficzny jest jedynie narzędziem ułatwiającym konstruowanie modeli wielowarstwowej sieci stwierdzeń MMNET. Po instalacji pakietu nie ma konieczności uruchamiania interfejsu graficznego. Dostępne są wszystkie metody i klasy umożliwiające budowanie zbioru stwierdzeń, modelu sieci, obliczanie modelu, wnioskowanie oraz odczyt/zapis modelu czy słownika stwierdzeń z/do pliku formatu XML. Można również rozszerzać funkcjonalność pakietu MMNET czy też stosować go we własnych pakietach.

Wygodnym narzędziem znacznie ułatwiającym rozwój oprogramowania w środowisku R jest platforma Eclipse (<http://www.eclipse.org/platform>) [5.1], która umożliwia integrację ze środowiskiem R. Zaletą tej platformy jest łatwość pełnej integracji ze środowiskiem R i oferowanie typowych narzędzi programistycznych jak edycja kodu, debugowanie, system pomocy środowiska R, budowanie dokumentacji, budowanie pakietów i wiele innych. Przykład użycia platformy Eclipse zintegrowanej ze środowiskiem R przedstawiono na rysunku 5.1.



Rys. 5.1: Okno platformy Eclipse zintegrowanej ze środowiskiem R

Bibliografia

- [5.1] Eclipse platform. Dostępny w Internecie: <http://www.eclipse.org/platform>.
- [5.2] Arne Henningsen A. Linprog: Linear programming // optimization. Dostępny w Internecie: cran.r-project.org/web/packages/linprog/linprog.pdf.
- [5.3] Bengtsson H., Riedy J. R.matlab: Read and write of mat files together with r-to-matlab connectivity. Dostępny w Internecie: cran.r-project.org/web/packages/R.matlab/R.matlab.pdf.
- [5.4] Csardi G. Network analysis and visualization. Dostępny w Internecie: cran.r-project.org/web/packages/igraph/igraph.pdf.
- [5.5] Højsgaard Søren. gRain: Graphical independence networks. Dostępny w Internecie: genetics.agrsci.dk/~sorenh/public/R/gRainweb/, 2010.
- [5.6] Hoefling Holger. BMN: The pseudo-likelihood method for pairwise binary markov networks, 2010.
- [5.7] Lawrence Michael, Verzani John. gWidgetsRGtk2: Toolkit implementation of gWidgets for RGtk2. Dostępny w Internecie: <http://cran.r-project.org/web/packages/gWidgetsRGtk2/index.html>, 2010.
- [5.8] Scutari Marco. Learning bayesian networks with the bnlearn R package, 2010.
- [5.9] Team R Development Core. Writing R Extensions. Dostępny w Internecie: cran.r-project.org/doc/manuals/R-exts.pdf, Version 2.12.0 (2010-10-15).
- [5.10] Temple Lang Duncan. XML: Tools for parsing and generating XML within R and S-Plus. Dostępny w Internecie: cran.r-project.org/web/packages/XML/index.html, Version 3.2-0 (2010-10-05).
- [5.11] Verzani John. gWidgets: gWidgets API for building toolkit-independent, interactive GUIs. Dostępny w Internecie: cran.r-project.org/web/packages/gWidgets/index.html, 2010.