



POLITECHNIKA POZNAŃSKA

SZTUCZNA INTELIGENCJA - LABORATORIUM
SPRAWOZDANIE KOŃCOWE

**Regułowy system doradczy
wspomagający zakup smartfona z
uwzględnieniem upodobań i
innych cech użytkownika**

SiL-L6-Z5

Anna Lehnhardt
Maciej Sobkowiak
Rafał Ewiak

24 czerwca 2020

Spis treści

| | | |
|----------|---|----------|
| 1 | Opis zdania | 2 |
| 2 | Założenia realizacyjne | 2 |
| 2.1 | Metody, strategie oraz algorytmy wykorzystywane do rozwiązania zadania | 2 |
| 2.2 | Języki programowania, narzędzia informatyczne i środowiska używane do implementacji systemu | 3 |
| 3 | Podział prac | 3 |
| 4 | Opis implementacji | 3 |
| 4.1 | System regułowy | 3 |
| 4.2 | Interfejs użytkownika | 5 |
| 5 | Użytkowanie i testowanie systemu | 6 |
| 6 | Literatura | 8 |
| 7 | Tekst programu | 8 |
| 7.1 | Wybór preferencji | 8 |
| 7.2 | Główna logika systemu regułowego | 11 |

1 Opis zdania

Stworzony na potrzeby zadania semestralnego system doradczy przeznaczony jest do przedstawienia proponowanych urządzeń mobilnych zwracając uwagę na ich specyfikację. Urządzenia wyświetlają się w tabeli gdzie w pierwszej kolumnie zawarte są nazwy modeli urządzeń, a w drugiej graficzna interpretacja rekomendacji danego telefonu. Poprzez prosty i przejrzysty interfejs dokładnie widać, które z urządzeń jest najbardziej polecane do preferencji użytkownika. Głównym przeznaczeniem systemu jest wspomaganie użytkownika posiadającego swoje preferencje dotyczące przeznaczenia smartfonu, lecz nie mającego wiedzy o jego podzespołach.

2 Założenia realizacyjne

2.1 Metody, strategie oraz algorytmy wykorzystywane do rozwiązania zadania

Dane wejściowe: wybrane przez użytkownika dane telefonów, wgrana baza telefonów

Dane wyjściowe: Spis telefonów, wraz z informacją w jakim stopniu odpowiada on wymaganiom.

Algorytm:

1. Do aplikacji wgrana jest baza telefonów
2. Użytkownik wybiera właściwości telefonu
3. Wybrane właściwości trafiają do systemu regułowego
4. System regułowy porównuje wybrane właściwości z cechami telefonów w bazie
5. System regułowy oblicza stopień podobieństwa.
 - (a) Obliczany jest stopień podobieństwa dla każdej z cech telefonu na podstawie wybranych właściwości
 - (b) Jeśli kilka reguł odnosi się do jednej cechy to stopień podobieństwa tej cechy jest ich średnią
 - (c) Stopnie podobieństwa dla każdej z cech są sumowane i wyliczana jest z nich średnia.
6. Użytkownik otrzymuje spis telefonów, wraz z informacją w jakim stopniu odpowiada on wymaganiom

2.2 Języki programowania, narzędzia informatyczne i środowiska używane do implementacji systemu

Do implementacji logiki predykcyjnej zadania został użyty język *CLIPS 6.3*, natomiast do stworzenia interfejsu graficznego język *Java* z *OpenJDK* w wersji 12. W celu połączenia systemu regułowego z interfejsem graficznym użyta została biblioteka *CLIPS JNI 6.4*.

3 Podział prac

| Autor | Podzadanie |
|------------------|---|
| Anna Lehnhardt | Implementacja systemu regułowego |
| Maciej Sobkowiak | Stworzenie bazy danych telefonów |
| | Współudział w implementacji systemu regułowego |
| Rafał Ewiak | Stworzenie interfejsu graficznego |
| | Wstrzykiwanie reguł do programu clipsowego w aplikacji interfejsu graficznego |

Tabela 1: Podział prac

4 Opis implementacji

4.1 System regułowy

System nie zakłada zwrócenia jednego, najbardziej pasującego wyniku do preferencji użytkownika, lecz procent dopasowania dla każdego z telefonów w kolejności od najbardziej zgodnego z cechami użytkownika. Zdefiniowana w systemie baza danych składająca się z 30 różnych telefonów charakteryzujących się następującymi atrybutami:

- nazwa
- system
- cena

- rozdzielczość przedniego aparatu
- rozdzielczość tylnego aparatu
- pamięć RAM
- przekątna ekranu
- dual-sim
- wodoodporność
- pojemność baterii
- pamięć urządzenia

Na podstawie powyższych atrybutów opierają się zdefiniowane reguły oraz fakty w systemie, np.

```
(rule (if preferred-system is android)
      (then best-system is android and best-system is
            ios with certainty 0))
(rule (if preferred-system is ios)
      (then best-system is ios and best-system is
            android with certainty 0))
(rule (if preferred-system is unknown)
      (then best-system is android with certainty 50 and
            best-system is ios with certainty 50))
```

Reguły mogą dotyczyć jednej lub kilku atrybutów, np.

```
(rule (if preferred-multiple-apps is yes)
      (then best-memory-size is big with certainty 65 and
            best-memory-size is small with certainty 35 and
            best-ram-size is big with certainty 80 and
            best-ram-size is small with certainty 20))
```

Fakty w systemie:

```
(deffacts PHONES::the-phone-list
  (phone (name "Xiaomi Redmi Note 8 Pro") (price 999)
        (system android) (front-camera 20) (back-camera 64)
        (ram 6) (screen-size 6.53) (dual-sim yes) (ip yes)
        (battery 4500) (memory 64))
  .
  .
)
```

Powyższy fakt *PHONES::the-phone-list* składa się z danych o charakterze wejściowym mającym na celu wprowadzenie dostępnych telefonów do systemu. Natomiast pierwsza reguła określa w jakim stopniu wybrany przez użytkownika system operacyjny oddziałowuje na ostateczny wynik. Druga reguła określa w jaki sposób oddziałowuje zaznaczenie przez użytkownika atrybutu, określającego czy telefon ma być odpowiedni do korzystania z wielu aplikacji na raz.

```
(defrule PHONES::check-phones-price
  ?ph <- (phone (name ?name)
               (price $? ?price $?))
  =>
  (if (> ?price 1000) then
    (if (> ?price 2000) then
      (modify ?ph (price-final big))
      else (modify ?ph (price-final medium)))
    else (modify ?ph (price-final small)))
)
```

Powyższa reguła odczytuje zadeklarowany przez użytkownika przedział cenowy i na tej podstawie określa wartość, która będzie mogła być zinterpretowana w prosty sposób przez pozostałe reguły.

```
(assert (attribute (name phone) (value ?name)
  (certainty (/ (+ (+ (+ (+ (+ (+ (+ (+ (+ ?certainty -1
?certainty -2) ?certainty -3) ?certainty -4) ?certainty -5)
?certainty -6) ?certainty -7) ?certainty -8)
?certainty -9) ?certainty -10) 10 ))))
```

Powyższa funkcja przypisuje telefonom średnią, ostateczną wartość dopasowania wybranej pozycji do wyborów użytkownika.

4.2 Interfejs użytkownika

Aplikacja interfejsu użytkownika posługuje się napisanym w języku *CLIPS* programem odpytując go, w celu uzyskania listy wynikowej zawierającej smartfony najbardziej pasujące do wybranych przez użytkownika preferencji.

System regułowy wczytywany jest z pliku o rozszerzeniu *clp* do wirtualnego środowiska utworzonego w *javie*.

```
clips = new Environment();
try {
  clips.loadFromResource("/net/sf/clipsrules/jni/examples" +
    "/chooseyourphone/resources/phones.clp");
```

```

} catch (Exception e) {
    e.printStackTrace();
    System.exit(1);
}

```

Po wybraniu przez użytkownika interesującej go cechy smartfona do tak utworzonego środowiska wstrzykiwana jest odpowiednia reguła.

Przykładowo użytkownika interesuje smartfon z systemem Android. Po zaznaczeniu tej opcji w aplikacji *Action Listener* który ją obsługuje wywoła funkcję:

```

item = PreferredSystemNames[preferredSystem.getSelectedIndex()];
if (item.equals("Android")) {
    clips.assertString("(attribute (name preferred-system) " +
        "(value android))");
} else if (item.equals("iOS")) {
    clips.assertString("(attribute (name preferred-system) " +
        "(value ios))");
} else {
    clips.assertString("(attribute (name preferred-system) " +
        "(value unknown))");
}

```

Funkcja ta zmieni wartość zmiennej odpowiadającej za preferowany system w wirtualnym środowisku systemu regułowego.

5 Użytkowanie i testowanie systemu

W celu uruchomienia programu należy wejść do folderu z projektem z poziomu konsoli i wpisać komendę:

```
java -jar .\ChooseYourPhone.jar
```

posługując się Javą w wersji 12.

Graficzny interfejs użytkownika składa się z 11 umożliwiających wybór preferencji, podzielonych na dwa typy.

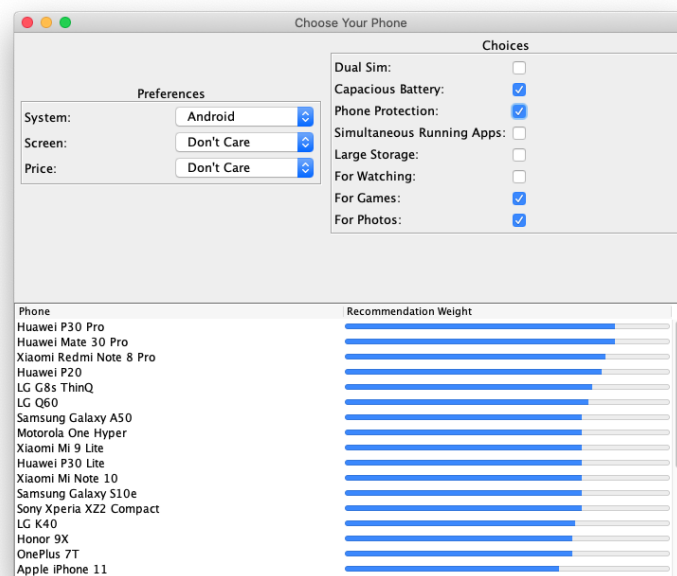
Pola typu *ComboBox* pozwalają na określenie najważniejszych z punktu widzenia użytkownika cech smartfonu, czyli

- systemu operacyjnego
 - Android
 - iOS
- wielkości ekranu

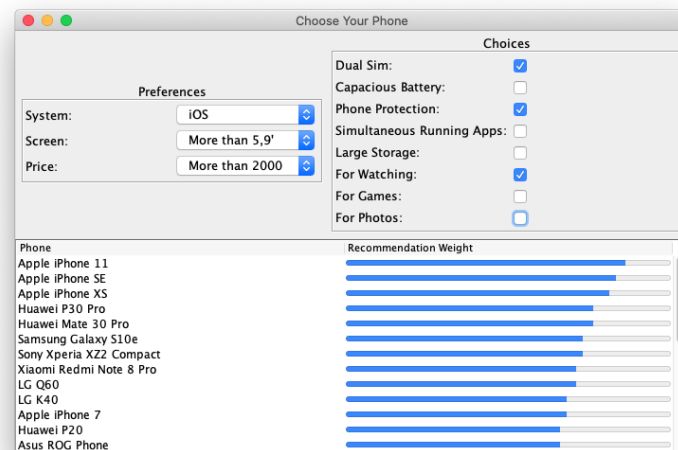
- mniej niż 5,5”
- więcej niż 5,5’
- ceny
 - poniżej 1000 zł
 - między 1000 zł i 2000 zł
 - powyżej 2000 zł

Natomiast pola typu *CheckBox* umożliwiają szczegółowe ustalenie przeznaczenia urządzenia.

Użytkownik systemu zaznacza możliwe opcje na podstawie własnych upodobań. Przykładowo, jeśli użytkownik poszukuje telefonu z systemem android, który będzie wykorzystywał głównie do oglądania filmów i robienia zdjęć, ale również zależy mu na dużej pojemności baterii i wodoodporności powinien wybrać odpowiednio opisane opcje. Poniżej przedstawione zostały przykładowe 2 zestawy danych wraz z wynikiem działania systemu.



Rysunek 1: Przykładowy zestaw danych nr 1



Rysunek 2: Przykładowy zestaw danych nr 2

6 Literatura

<http://clipsrules.sourceforge.net/documentation/v630/ug.pdf>

<https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>

https://kcir.pwr.edu.pl/witold/ai/CLIPS_tutorial/

7 Tekst programu

7.1 Wybór preferencji

```
item = PreferredSystemNames[preferredSystem.getSelectedIndex()];

if (item.equals("Android")) {
    clips.assertString("(attribute (name preferred-system)
    (value android))");
} else if (item.equals("iOS")) {
    clips.assertString("(attribute (name preferred-system)
    (value ios))");
} else {
    clips.assertString("(attribute (name preferred-system)
    (value unknown))");
}
```

```

item = PreferredScreenNames[PreferredScreen.getSelectedIndex()];
if (item.equals("Less than 5,9")) {
    clips.assertString("(attribute (name preferred-screen-size)
        (value small))");
} else if (item.equals("More than 5,9")) {
    clips.assertString("(attribute (name preferred-screen-size)
        (value big))");
} else {
    clips.assertString("(attribute (name preferred-screen-size)
        (value unknown))");
}

item = PreferredPriceNames[PreferredPrice.getSelectedIndex()];
if (item.equals("Less than 1000")) {
    clips.assertString("(attribute (name preferred-price)
        (value small))");
} else if (item.equals("1000 to 2000")) {
    clips.assertString("(attribute (name preferred-price)
        (value medium))");
} else if (item.equals("More than 2000")) {
    clips.assertString("(attribute (name preferred-price)
        (value big))");
} else {
    clips.assertString("(attribute (name preferred-price)
        (value unknown))");
}

/*checkbox rules*/
if (preferredDualSim.isSelected()) {
    clips.assertString("(attribute (name preferred-dual-sim)
        (value yes))");
} else {
    clips.assertString("(attribute (name preferred-dual-sim)
        (value unknown))");
}

if (PreferredForGames.isSelected()) {
    clips.assertString("(attribute (name preferred-games)
        (value yes))");
} else {

```

```

        clips.assertString("(attribute (name preferred-games)
(value no))");
    }

    if (PreferredForPhotos.isSelected()) {
        clips.assertString("(attribute (name preferred-photos)
(value yes))");
    } else {
        clips.assertString("(attribute (name preferred-photos)
(value no))");
    }

    if (PreferredBatteryCapacity.isSelected()) {
        clips.assertString("(attribute (name preferred-battery)
(value yes))");
    } else {
        clips.assertString("(attribute (name preferred-battery)
(value no))");
    }

    if (PreferredDifficultConditions.isSelected()) {
        clips.assertString("(attribute (name preferred-ip)
(value yes))");
    } else {
        clips.assertString("(attribute (name preferred-ip)
(value no))");
    }

    if (PreferredMultipleApps.isSelected()) {
        clips.assertString("(attribute
(name preferred-multiple-apps)
(value yes))");
    } else {
        clips.assertString("(attribute
(name preferred-multiple-apps)
(value no))");
    }

    if (PreferredBigMemory.isSelected()) {
        clips.assertString("(attribute (name preferred-big-memory)

```

```

        (value yes))");
    } else {
        clips.assertString("(attribute (name preferred-big-memory)
        (value no))");
    }

    if (PreferredForWatching.isSelected()) {
        clips.assertString("(attribute (name preferred-movies)
        (value yes))");
    } else {
        clips.assertString("(attribute (name preferred-movies)
        (value no))");
    }

    Runnable runThread =
        () -> {
            try {
                clips.run();
            } catch (CLIPSEException e) {
                e.printStackTrace();
            }

            SwingUtilities.invokeLater(
                () -> {
                    try {
                        updatePhones();
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            ));
};

```

7.2 Główna logika systemu regułowego

```

(deftemplate MAIN::attribute
  (slot name)
  (slot value)
  (slot certainty (default 100.0)))

(defrule MAIN::start
  (declare (salience 10000))

```

```

=>
(set-fact-duplication TRUE)
(focus CHOOSE-QUALITIES PHONES))

(defrule MAIN::combine-certainties ""
  (declare (salience 100)
            (auto-focus TRUE))
  ?rem1 <- (attribute (name ?rel) (value ?val) (certainty ?per1))
  ?rem2 <- (attribute (name ?rel) (value ?val) (certainty ?per2))
  (test (neq ?rem1 ?rem2))
=>
  (retract ?rem1)
  (modify ?rem2 (certainty (/ (+ ?per1 ?per2) 2))))

(deffacts the-phone-rules

; system
(rule (if preferred-system is android)
      (then best-system is android and best-system is ios with
certainty 0))
(rule (if preferred-system is ios)
      (then best-system is ios and best-system is android with
certainty 0))
(rule (if preferred-system is unknown)
      (then best-system is android with certainty 50 and
            best-system is ios with certainty 50))

; dual-sim
(rule (if preferred-dual-sim is yes)
      (then best-dual-sim is yes and best-dual-sim is no with
certainty 0))
(rule (if preferred-dual-sim is no)
      (then best-dual-sim is no and best-dual-sim is yes with
certainty 0))
(rule (if preferred-dual-sim is unknown)
      (then best-dual-sim is yes with certainty 50 and
            best-dual-sim is no with certainty 50))

; screen-size
(rule (if preferred-screen-size is big)

```

```

        (then best-screen-size is big and best-screen-size is
small with certainty 0))
(rule (if preferred-screen-size is small)
      (then best-screen-size is small and best-screen-size is
big with certainty 0))
(rule (if preferred-screen-size is unknown)
      (then best-screen-size is big with certainty 50 and
          best-screen-size is small with certainty 50))

; price
(rule (if preferred-price is big)
      (then best-price is big and best-price is small with
          certainty 0 and best-price is medium with certainty 0))
(rule (if preferred-price is medium)
      (then best-price is medium and best-price is small with certain
(rule (if preferred-price is small)
      (then best-price is small and best-price is big with
          certainty 0 and best-price is medium with certainty 0))
(rule (if preferred-price is unknown)
      (then best-price is small with certainty 50 and
          best-price is medium with certainty 50 and
          best-price is big with certainty 50))

; for gamers
(rule (if preferred-games is yes)
      (then best-screen-size is big with certainty 65 and
          best-screen-size is small with certainty 35 and
          best-ram-size is big with certainty 80 and
          best-ram-size is small with certainty 20 and
          best-battery is big with certainty 80 and
          best-battery is small with certainty 20 and
          best-memory-size is big with certainty 60 and
          best-memory-size is small with certainty 40))

; for photos
(rule (if preferred-photos is yes)
      (then best-front-camera is big with certainty 80 and
          best-front-camera is small with certainty 20 and
          best-back-camera is big with certainty 90 and

```

```

        best-back-camera is small with certainty 10 and
        best-memory-size is big with certainty 70 and
        best-memory-size is small with certainty 30))
(rule (if preferred-photos is no)
      (then best-front-camera is big with certainty 50 and
            best-front-camera is small with certainty 50 and
            best-back-camera is big with certainty 50 and
            best-back-camera is small with certainty 50 and
            best-memory-size is big with certainty 50 and
            best-memory-size is small with certainty 50))

; capacious battery
(rule (if preferred-battery is yes)
      (then best-battery is big and best-battery is small
            with certainty 0))
(rule (if preferred-battery is no)
      (then best-battery is big with certainty 0 and
            best-battery is small))

; difficult conditions
(rule (if preferred-ip is yes)
      (then best-ip is yes and best-ip is none with
            certainty 0))
(rule (if preferred-ip is no)
      (then best-ip is none and best-ip is yes with
            certainty 0))

; multiple apps
(rule (if preferred-multiple-apps is yes)
      (then best-memory-size is big with certainty 65 and
            best-memory-size is small with certainty 35 and
            best-ram-size is big with certainty 80 and
            best-ram-size is small with certainty 20))
(rule (if preferred-multiple-apps is no)
      (then best-ram-size is small with certainty 50 and
            best-ram-size is big with certainty 50 and
            best-memory-size is big with certainty 50 and
            best-memory-size is small with certainty 50))

; big memory

```

```

(rule (if preferred-big-memory is yes)
      (then best-memory-size is big and
             best-memory-size is small with certainty 0))
(rule (if preferred-big-memory is no)
      (then best-memory-size is small and
             best-memory-size is big with certainty 0))

; for watching
(rule (if preferred-movies is yes)
      (then best-screen-size is big with certainty 90 and
             best-screen-size is small with certainty 10 and
             best-memory-size is big with certainty 60 and
             best-memory-size is small with certainty 40 and
             best-battery is big with certainty 75 and
             best-battery is small with certainty 25))

(rule (if preferred-movies is no)
      (then best-battery is big with certainty 50 and
             best-battery is small with certainty 50
             best-memory-size is big with certainty 50 and
             best-memory-size is small with certainty 50))
)

(defrule PHONES::check-phones-screen
  ?ph <- (phone (name ?name)
                (screen-size $? ?scr $?))
  =>
  (if (> ?scr 5.9) then
      (modify ?ph (screen-size-final big))
    else (modify ?ph (screen-size-final small)))
)

(defrule PHONES::check-phones-price
  ?ph <- (phone (name ?name)
                (price $? ?price $?))
  =>
  (if (> ?price 1000) then
      (if (> ?price 2000) then
          (modify ?ph (price-final big))
        else (modify ?ph (price-final medium)))
  )
)

```



```

        else (modify ?ph (price-final small)))
    )

(defrule PHONES::check-battery-size
  ?ph <- (phone (name ?name)
                (battery $? ?bat $?))

  =>
  (if (> ?bat 3100) then
      (modify ?ph (battery-final big))
    else (modify ?ph (battery-final small)))
  )

(defrule PHONES::check-ram-size
  ?ph <- (phone (name ?name)
                (ram $? ?r $?))

  =>
  (if (> ?r 3) then
      (modify ?ph (ram-final big))
    else (modify ?ph (ram-final small)))
  )

(defrule PHONES::check-front-camera
  ?ph <- (phone (name ?name)
                (front-camera $? ?fc $?))

  =>
  (if (> ?fc 20) then
      (modify ?ph (front-camera-final big))
    else (modify ?ph (front-camera-final small)))
  )

(defrule PHONES::check-back-camera
  ?ph <- (phone (name ?name)
                (back-camera $? ?bc $?))

  =>
  (if (> ?bc 20) then
      (modify ?ph (back-camera-final big))
    else (modify ?ph (back-camera-final small)))
  )

(defrule PHONES::check-memory

```

```

?ph <- (phone (name ?name)
              (memory $? ?mem $?))
=>
  (if (>= ?mem 32) then
      (modify ?ph (memory-final big))
    else (modify ?ph (memory-final small)))
)

(attribute (name best-price) (value ?price-final)
           (certainty ?certainty-1))
(attribute (name best-system) (value ?c)
           (certainty ?certainty-2))
(attribute (name best-dual-sim) (value ?s)
           (certainty ?certainty-3))
(attribute (name best-ram-size) (value ?rf)
           (certainty ?certainty-4))
(attribute (name best-front-camera) (value ?fcf)
           (certainty ?certainty-5))
(attribute (name best-back-camera) (value ?bcf)
           (certainty ?certainty-6))
(attribute (name best-memory-size) (value ?memf)
           (certainty ?certainty-7))
(attribute (name best-battery) (value ?batf)
           (certainty ?certainty-8))
(attribute (name best-ip) (value ?ip)
           (certainty ?certainty-9))
(attribute (name best-screen-size) (value ?scrf)
           (certainty ?certainty-10))
=>
  (assert (attribute (name phone) (value ?name)
                    (certainty (/ (+ (+ (+ (+ (+ (+ (+ (+ (+ (+
?certainty-1 ?certainty-2)
?certainty-3) ?certainty-4)
?certainty-5) ?certainty-6)
?certainty-7) ?certainty-8)
?certainty-9) ?certainty-10) 10 )))
)

```