

EKSPERTOWE INTELIGENTNE ŚRODOWISKO WSPOMAGANIA PROCESU DORADCZO-DECYZYJNEGO

Zbigniew BUCHALSKI

Streszczenie: W artykule przedstawiono pewną koncepcję systemu ekspertowego o nazwie LOGISER wspomagającego organizację pracy firmy spedycyjnej. Podano podstawowe założenia budowy tego systemu oraz opis funkcjonowania systemu ekspertowego. Wiedza zgromadzona w bazie wiedzy systemu ekspertowego reprezentowana jest przez reguły i fakty. Pozyskiwanie wiedzy do bazy wiedzy odbywa się na bieżąco w trakcie pracy tego systemu. Zrealizowana została implementacja komputerowa zaprezentowanego systemu.

Słowa kluczowe: nowoczesne technologie informatyczne, komputerowe wspomaganie procesu doradczo-decyzyjnego, usługi logistyczne.

1. Wstęp

W ciągu ostatnich lat zaobserwować można zarówno na rynku światowym, jak i polskim gwałtowny wzrost zapotrzebowania na specjalistyczne systemy komputerowe wspomagające ludzką działalność, czyli tzw. systemy ekspertowe [1, 2, 3, 4, 5]. Systemy ekspertowe można obecnie spotkać prawie w każdej dziedzinie ludzkiej działalności począwszy od medycyny poprzez technikę aż do podejmowania skomplikowanych decyzji finansowych. Zastosowanie systemu ekspertowego może przynieść znaczne korzyści finansowe firmie lub instytucji, która pokusi się o jego wdrożenie [6, 7, 8, 9].

Szczególnie zauważalne jest zastosowanie systemów ekspertowych na prężnie rozwijającym się rynku usług logistycznych. Logistyka to proces polegający na planowaniu, realizowaniu i kontrolowaniu sprawnego, a przede wszystkim opłacalnego przepływu towarów. Obejmuje ona zakresem swoich działań m.in. obsługę klienta, przyjmowanie i realizację zamówień, transport wewnętrzny i zewnętrzny oraz przepływ i gromadzenie informacji [10, 11, 12, 13].

Za podstawową i w obecnych czasach nieodzowną część działań logistycznych przyjmuje się spedycję. Firmy spedycyjne muszą rozwiązać w krótkim czasie wiele trudnych problemów transportowych, od wyniku których uzależniona jest ich ostateczna sytuacja finansowa. Problemy te dotyczą m.in. obsługi zamówień, przydziału odpowiedniego środka transportu do przewozu, obsługi klientów, powiększania bazy transportowej o nowe pojazdy, itp. Ze względu na złożoność problemów znalezienie właściwego rozwiązania nie należy do zadań łatwych. Dlatego też w tak prężnie rozwijającej się branży, gdzie szybkie działanie a przede wszystkim trafne podejmowanie decyzji staje się niekiedy jedyną drogą do sukcesu, zastosowanie systemów ekspertowych staje się wręcz niezbędne.

W niniejszym artykule przedstawiono pewną koncepcję systemu ekspertowego o nazwie LOGISER wspomagającego organizację pracy firmy spedycyjnej. Podano podstawowe założenia budowy i opis działania systemu oraz zaprezentowano jego implementację komputerową. System LOGISER poddano badaniom testującym mającym na celu wyeliminowanie wszystkich nieścisłości powstałych podczas jego budowy.

2. Budowa systemu LOGISER

Podstawowym celem budowy systemu LOGISER było to, aby mógł on być pomocnym narzędziem wspomagającym pracę dyspozytora firmy transportowej. Przeprowadzona została dogłębna analiza tematu w celu właściwego zapoznania się z kluczowymi zagadnieniami dotyczącymi usług logistycznych i spedycji. W efekcie tych prac zostały sformułowane podstawowe założenia dotyczące budowy systemu LOGISER.

Praca dyspozytora firmy transportowej wymaga umiejętności związanych z dobrą koordynacją nad dostępnymi w firmie środkami i wymaganiami stawianymi przez klientów oraz podejmowania działań, by te wymagania zaspokoić. W celu rozpoznania problemów spedycyjnych, jak i efektywnego sposobu ich rozwiązania, została przeprowadzona analiza tematu, której wynikiem było opracowanie podstawowych założeń budowy systemu LOGISER. Założenia te są następujące:

- system ma przydzielać pojazd i kierowcę do danego zlecenia przewozu,
- obliczać koszty wykonywanych usług,
- czuwać nad terminowością wykonywanych zleceń,
- przypominać o terminach badań technicznych oraz ubezpieczeń pojazdów, jak również przechowywać informacje o naprawach serwisowych pojazdów,
- dbać o godzinowy wymiar pracy kierowców, odpowiednio przydzielać terminy wykonywanych przez nich zleceń przewozu,
- informować o potrzebie zakupu dodatkowych środków (np. pojazdów) umożliwiających rozwój firmy,
- przeprowadzać dodatkowe szkolenie, uprawniające do kierowania nowo zakupionymi pojazdami,
- dbać o ustaloną przez firmę politykę promocji dla stałych klientów,
- przypominać o terminach badań lekarskich (kontrolnych/okresowych) kierowców.

Została opracowana struktura bazy danych systemu, którą kontroluje użytkownik systemu. Baza danych systemu LOGISER powinna przechowywać następujące dane:

- zbiór zleceń przewozu towarów,
- zbiór klientów oraz dane potrzebne do wystawienia faktury VAT,
- zbiór pojazdów, którymi dysponuje firma oraz zestawów doczepianych do pojazdów,
- zbiór danych o zatrudnionych kierowcach,
- informacje o parametrach systemu (cena paliwa, polityka promocji, koszty pośrednie, związane na przykład z koniecznością rezerwacji noclegów dla kierowców).

Budowa systemu LOGISER została oparta głównie na opracowaniu reguł rządzących przeprowadzaniem ekspertyzy oraz samym mechanizmem, dla którego opracowane reguły byłyby czytelne. Zastosowanie w niej znalazł proces iteracyjnego wytwarzania oprogramowania, w którym krok po kroku zmniejsza się liczba elementów oprogramowania integrowanych ze sobą w funkcjonalną całość, sterowaną za pomocą opracowanych interfejsów. Struktura systemu podzielona została na elementy, w których oddzielnie zawarto model warstwy biznesowej, warstwy klienta (GUI od ang. Graphical User Interface), warstwy komunikacji z bazą danych oraz mechanizm wnioskowania i zadawania pytań. Dzięki temu składowe oprogramowania łatwiej poddają się rekonfiguracji.

System LOGISER generuje ekspertyzy z dziedziny logistyki oraz przechowuje w bazie

danych informacje o klientach, zleceniach i środkach posiadanych przez firmę spedycyjną. Instalowany jest bezpośrednio z folderu instalacyjnego wraz z niezbędnymi do jego działania elementami na którąkolwiek platformę systemową. Elementy instalowane wraz z programem to: serwer bazodanowy PostgreSQL w wersji 9.1, plik opracowanej bazy danych i reguł oraz środowisko uruchomieniowe języka JAVA (JRE). Platforma programistyczna, jaka została użyta do budowy systemu LOGISER to NetBeans IDE w wersji 7.1.1, która wspiera programowanie w języku JAVA, ułatwiając między innymi szybkie tworzenie interfejsu graficznego użytkownika.

3. Implementacja komputerowa systemu LOGISER

Implementacja komputerowa systemu ekspertowego o nazwie LOGISER polegała na budowie czterech jego podstawowych elementów składowych. Elementami tymi są:

- baza danych,
- warstwa biznesowa systemu z utrwalanymi klasami, odpowiadającymi tabelom bazy danych,
- warstwa integracji mapująca obiekty tych klas na encje,
- interfejs graficzny użytkownika, w którym ukryte zostały funkcje spełniające założenia projektowe.

3.1. Baza danych

Skonstruowanie bazy danych systemu LOGISER wymagało zgromadzenia wiedzy z zakresu logistyki i zaimplementowanie jej w strukturę tabel. Proces ten niejednokrotnie wymagał udoskonalania i rozszerzania zakresu przechowywanej wiedzy. Ostatecznie została opracowana kompozycja odpowiednich tabel bazy danych, adekwatnych do utrwalanych klas systemu.

Klasa systemu LOGISER o nazwie **Klient**, której obiekty przechowywane są w bazie danych, zawiera podstawowe informacje o klientach firmy spedycyjnej, zlecających przewóz towarów środkami posiadanych przez firmę. Kolejną klasą, której obiekty przechowywane są w bazie danych systemu jest **Kierowca**. Obiekty tej klasy magazynują informacje o kierowcach zatrudnionych w firmie spedycyjnej. Klasa o nazwie **Parametry** pozwala użytkownikowi na określenie i przechowywanie informacji niezbędnych do obliczania kosztów związanych z wykonywanymi przewozami.

Obiekty kolejnej klasy systemu LOGISER o nazwie **Reguly** zawierają informacje o regułach podejmowania decyzji podczas procesu wnioskowania systemu. Klasa o nazwie **Reguly** zawiera takie atrybuty jak:

- *ID_reguly* – numer niezbędny do identyfikacji konkretnego obiektu,
- *Nazwa_reguly* – klasyfikująca regułę do odpowiedniego typu zastosowania,
- *Warunek* – przesłanka, zdanie logiczne którego spełnienie powoduje konstatację konkluzji,
- *Konkluzja* – czyli działanie następujące w przypadku spełnienia warunku.

Obiekty klasy o nazwie **Zestaw** przechowują informację o przyczepach i naczepach, czyli ogólnie o zestawach dopinanych do pojazdów. Obiekty klasy o nazwie **Pojazd** zawierają informację o środkach mechanicznych posiadanych przez firmę spedycyjną, służących do realizacji zleceń. Podczas generowania ekspertyzy obiekty tej klasy są przyporządkowywane do konkretnych zleceń.

Klasa **Zlecenie** to klasa, której obiekty zawierają podstawowe informacje o zleceniach

przewozu, złożonych przez klientów do firmy spedycyjnej. Tworzenie obiektów klasy **Zlecenie** może się odbywać w dwojaki sposób. Jednym z nich jest pominięcie mechanizmów wnioskowania i własnoręczne określenie wszystkich wymaganych informacji, drugim zaś sposobem jest praca kreatora, który wykorzystuje mechanizm wnioskowania. Przy drugim sposobie niektóre z atrybutów uzupełniane są przez użytkownika, inne są uzupełniane przez system w trakcie generowania ekspertyzy. Ekspertyza w tym przypadku, polega na przyporządkowaniu odpowiednich środków i osób do wykonania danego zlecenia, wyznaczeniu odpowiednich terminów oraz redukcji bądź eliminacji pustych przebiegów.

Implementacja struktury bazy danych została zrealizowana za pomocą programu pgAdmin III. Jest to system zarządzania bazami danych PostgreSQL, umożliwiający ich tworzenie, dodawanie nowych tabel, ustawianie ograniczeń oraz konstruowanie zapytań, dzięki którym możliwe jest wykonywanie operacji na danych. Za jego pomocą można również tworzyć kopie zapasowe oraz przywracać je w razie utracenia danych. Ponadto jest on dostępny również w polskiej wersji językowej, dzięki czemu zrozumienie jego funkcjonalności nie sprawia żadnego problemu. W jego oknie głównym prezentowane jest drzewo obsługiwanych obiektów, wśród których znajduje się baza danych systemu LOGISER.

3.2. Warstwa biznesowa systemu LOGISER

Implementacja warstwy biznesowej systemu LOGISER jest adekwatna do skonstruowanej bazy danych. Stosowany w nim opis tabel i kolumn nawiązuje do terminologii obiektowej. Implementacja ta polegała na zbudowaniu stosownych klas, odpowiadających tabelom bazy danych wraz z atrybutami, które z kolei nawiązują do kolumn tych tabel. To właśnie instancje tych klas, przy wykorzystaniu logiki warstwy integracji, zapisywane są w bazie danych.

Aby klasy warstwy biznesowej mogły być odwzorowywane na tabele bazy danych, konieczna była pewna ich modyfikacja. Przekształcenie klas na encje (@Entity) wiązało się z uzupełnieniem ich kodu o właściwe adnotacje, które rzutowałyby na zawarte w nich atrybuty. Klasy musiały również implementować interfejs serializacji obiektów, dzięki któremu skutecznie przekształcane są do postaci szeregowej (strumień bajtów). Wśród metod tych klas można wyróżnić metody getNazwaAtrybutu(), zwracające wartości atrybutów danego obiektu oraz metody setNazwaAtrybutu(), przypisujące wartości do atrybutów. Oprócz tego klasy dysponują zdefiniowanymi na potrzeby typu @Entity metodami hashCode(), equals(Object object) oraz toString(), odpowiedzialnymi za identyfikację obiektu, porównanie go z innymi obiektami oraz modyfikację obiektu do łańcucha znaków, np. by wyświetlić go w polu tekstowym.

3.3. Warstwa integracji systemu LOGISER

W celu pobierania i umieszczania obiektów klas modelu biznesowego w bazie danych została zaimplementowana warstwa integracji systemu LOGISER. Pozwala ona oddzielić skomplikowaną logikę komunikacji ze źródłem danych od warstwy biznesowej. Dzięki takim narzędziom, jak środowisko programistyczne NetBeans tworzenie tej warstwy jest bardzo proste i w przypadku, gdy posiadamy już utworzoną bazę danych, sprowadza się do wykonania kontrolerów warstwy integrującej warstwę biznesową z bazą danych. Czynność tą możemy podzielić na trzy etapy: dodanie i skonfigurowanie modułu utrwalania danych

typu Persistence Unit w technologii TopLink, przekształcenie klas danych na typ @Entity, co wiąże się z dodaniem odpowiednich adnotacji i przededefiniowaniem niektórych metod oraz uzupełnienie projektu aplikacji o klasy typu Controller, dla każdej z utrwalanych klas modelu obiektowego.

Tworzenie i konfiguracja modułu Persistence Unit są przeprowadzane przez kreator. Podobnie jak ma to miejsce podczas tworzenia dowolnej klasy programu należy określić nazwę oraz tzw. package, czyli miejsce zapisu modułu, ustalić jego bibliotekę (TopLink) oraz połączenie z odpowiednią bazą danych (logis_ekspert_db). Metody klas typu Controller pozwalają na dodawanie, modyfikację, usuwanie oraz wyszukiwanie obiektów klas biz-nesowych w bazie danych.

Za pomocą klas typu Controller zwracana jest lista wszystkich instancji danej klasy, zapisanych w bazie. Korzystanie z wszystkich tych klas osobno dla poszczególnych przypadków jest dość uciążliwe. Dlatego też w celu skutecznego komunikacji stworzono klasę SystemRulles. Wiąże ona funkcjonalności udostępniane przez klasy kontrolerów. Jej instancja tworzona jest podczas startu systemu i przekazywana jako argument.

3.4. Interfejs graficzny użytkownika

Projekt interfejsu graficznego (GUI) zakładał wykorzystanie komponentów z biblioteki Swing języka JAVA. W oparciu o te założenia, cały interfejs bazuje na klasie JFrame, która stanowi okna systemu wraz z odpowiednio zastosowanymi kontrolkami, takimi jak: przyciski (JButton), tabele (JTable), pola i obszary tekstowe (JTextField) i (JTextArea) czy listy rozwijane (JComboBox). Wszystkie wymienione elementy GUI wpływają na czytelny, prosty i prawidłowy odczyt danych wprowadzanych przez użytkownika, a także na przejrzystą prezentację wyników pracy systemu.

Po uruchomieniu systemu ukazuje się okno główne, wyposażone w menu rozwijane oraz przyciski w górnej jego części, a także tabelę przedstawiającą utworzone i przechowywane w bazie danych zlecenia. Tabela zaopatrzona jest w opcję automatycznego sortowania wierszy, za pomocą której możliwe jest chronologiczne ułożenie zleceń od najpóźniej do najwcześniej utworzonego, bądź odwrotnie. Oczywiście chronologiczne ułożenie zleceń jest tylko przykładem a sortowanie odbywa się dla każdej kolumny wyświetlanej w tabeli.

W trakcie działania kreatora liczenia kosztów zlecenia spełniana jest również polityka promocji dla stałych klientów. Pozostałe założenia, polegające na przypominaniu o terminach zleceń, badań technicznych, lekarskich, naprawach serwisowych oraz opłat za ubezpieczenie, realizowane są za pomocą mechanizmu wnioskującego.

Kolumny te nie prezentują jednak wszystkich informacji, jakie zawierają obiekty klasy zlecenie. Dla lepszej czytelności, tylko najważniejsze informacje o zleceniu są w niej zamieszczone. W celu prezentacji wszystkich danych należy zaznaczyć w tabeli interesujące nas zlecenie, a następnie kliknąć w przycisk *Edytuj zlecenie*.

Kliknięcie w przycisk *Edytuj zlecenie* otwiera kolejne okno programu, prezentujące wszystkie dane zlecenia. Jest ono również odpowiedzialne za wprowadzanie ręcznych zmian w polach zlecenia, które w tym wypadku są wypełnione. Dokładnie to samo okno odpowiedzialne jest za dodawanie nowych zleceń. Tym razem okno otwiera się po kliknięciu przycisku *Dodaj zlecenie*, znajdujące się w nim pola tekstowe nie są wypełnione a listy rozwijane (odnośnie wyboru klienta zlecającego, kierowcy, pojazdu oraz rodzaju przewożonego towaru) posiadają zaznaczone pierwsze elementy, które się na nich znajdują. Okno to, uruchamiane z poziomu przycisków *Dodaj zlecenie* oraz *Edytuj zlecenie*, pomija

mechanizmy wnioskujące systemu i służy tylko do ręcznego wprowadzania czy też korekcji danych przez użytkownika.

Informacje wprowadzane do panelu podlegają kontroli poprawności w momencie kliknięcia przycisku *Zapisz*. Gdy proces sprawdzania przebiegnie pomyślnie, obiekt klasy *Zlecenie* zapisywany jest do bazy danych. W przypadku gdy proces ten wykryje nieprawidłowości, użytkownik jest o tym powiadomiony za pomocą okna zawierającego informacje o źródle błędu i sposobie jego korekcji. Rezygnacja z zapisu zlecenia, zamknięcie bieżącego okna *Dodawanie/edycja zleceń* oraz przywrócenie okna głównego następuje w momencie użycia przycisku *Anuluj*.

Okno główne programu jest wyposażone w kolejne cztery przyciski, które uruchamiają funkcjonalności, tym razem wykorzystujące mechanizmy wnioskujące i reguły bazy wiedzy. Przyciskami tymi są: *Wnioskowanie Kreator zleceń*, *Policz koszt zlecenia*, *Przypomnij terminy* oraz *Puste przebiegi*. Zaczynając od lewej strony okna głównego klikamy na przycisk *Wnioskowanie Kreator zleceń*, przez co zostanie utworzone nowe zlecenie. Uruchamia się okno informacyjne o nazwie *Ekspert* oraz pierwszy panel z pytaniem o nazwę zlecenia. Po wprowadzeniu nazwy nowotworzonego zlecenia klikamy na przycisk *OK* i przechodzimy do kolejnych kroków dotyczących rodzaju, wartości i kubatury przewożonego towaru.

Po zadaniu pytań użytkownikowi systemu *Kreator zleceń* zgromadził odpowiednią wiedzę, by mogło rozpocząć się wnioskowanie. Jako fakt stwierdzony po stronie tworzonego zlecenia, jak i pojazdu, który ma zostać do niego przyporządkowany należy uznać kubaturę oraz ciężar ładunku. Rozpoczyna się poszukiwanie pojazdów, spełniających postawione kryteria i tworzona jest lista tych, które pod względem ładowności wstępnie klasyfikują się do przyporządkowania do zlecenia. Tworzone jest również zdanie logiczne, na podstawie zgromadzonych informacji, stanowiące o warunkach, jakie musi spełniać pojazd, by zostać przyporządkowany do zlecenia.

Zostają uruchomione reguły, dotyczące pojazdów i sprawdzane są zdania zapisane w części warunkowej *IF* poszczególnych reguł. Jeżeli podobne zdanie zostanie napotkane i warunek zostanie spełniony, wówczas wykonywana jest część zawarta w konkluzji reguły i określony tam pojazd zostaje przyporządkowany do zlecenia. W przeciwnym wypadku, w oknie informacyjnym, zostaje wyświetlona informacja o braku możliwości przewozu. Informacja ta sugeruje, iż konieczna jest odnowa taboru, czyli zakup nowego środka transportu lub nie została jeszcze zdefiniowana odpowiednia reguła, którą posiadany pojazd ma przyporządkować do tego typu zlecenia.

Kolejnym etapem jest ustalenie klienta, co następuje po wybraniu go z listy rozwijanej zdefiniowanych klientów oraz wypełnieniu informacji adresowych, miejsca załadunku oraz dostawy. Zostaje wyświetlone kolejne pytanie o podanie odległości z miejsca załadunku do miejsca dostawy. Po udzieleniu odpowiedzi zlecenie jest finalizowane. Zostają określone daty zlecenia, załadunku oraz dostawy. Wyznaczanie dat załadunków i dostaw jest ściśle związane z redukcją lub nawet całkowitą eliminacją pustych przebiegów.

Prezentowany system nie jest wyposażony w mechanizmy wyznaczania trasy, czy też orientacji w terenie. Dlatego też po takiej analizie zawsze następuje wygenerowanie informacji dla użytkownika, który potwierdzi bądź zaneguje podążanie pojazdu w tym samym kierunku dla realizacji dwóch zleceń lub czy ładunek z kolejnego zlecenia można zabrać drogą powrotną. Reasumując, jeżeli dane zlecenia da się połączyć z innym zleceniem i pojazd zdąży w tym samym kierunku, wyznaczone są podobne daty załadunku i dostawy obu zleceń. Jeżeli kod pocztowy miejsca dostawy jednego zlecenia wskazuje na okolice miejsca załadunku drugiego zlecenia, to termin załadunku zlecenia drugiego

wyznaczany jest po terminie dostawy zlecenia pierwszego i po wyliczonym czasie przejazdu z punktu dostawy zlecenia pierwszego do punktu załadunku zlecenia drugiego.

Tworzenie zleceń poprzez *Kreator zleceń* zawsze jest związane z eliminacją pustych przebiegów i nie ma potrzeby uruchamiania mechanizmu sprawdzania pustych przebiegów przy użyciu przycisku w oknie głównym systemu dla tak utworzonego zlecenia. Przycisk ten przeznaczony jest dla sprawdzenia zleceń utworzonych indywidualnie przez panel dodawania nowego zlecenia uruchamiany przyciskiem, znajdującym się w oknie głównym.

Po finalizacji procesu wnioskowania uaktywnił się przycisk *Zatwierdź*. W oknie informacyjnym *Ekspert* można obejrzeć przebieg wnioskowania, skontrolować poprawność udzielanych odpowiedzi i w przypadku nie stwierdzenia żadnych uchybień, zatwierdzić proces odpowiednim przyciskiem. Po kliknięciu na przycisk *Zatwierdź* nie następuje jednak bezpośredni zapis zlecenia do bazy danych. W celu lepszej wizualizacji, wszystkie dane utworzonego przez *Kreator zleceń* zlecenia, jeszcze raz wyświetlane są w panelu *Dodawanie/Edycja zlecenia*. Możliwe jest ponowne, dokładniejsze przeanalizowanie danych wygenerowanych przez *Kreator zleceń*. Zostaje jeszcze wygenerowana informacja dla użytkownika, iż koszt usługi nie został jeszcze obliczony. By to uczynić, najpierw należy zapisać zlecenie w bazie danych, za pomocą przycisku *Zapisz* w oknie *Dodawanie/edycja zlecenia*.

Po użyciu przycisku *Zapisz* zapisane w bazie danych zlecenie prezentowane jest już w tabeli okna głównego. Nadal jednak widoczne jest, że koszt zlecenia wynosi 0 zł. Inicjowanie liczenia kosztów dla żadanego zlecenia wygląda następująco. Należy wybrać interesujące nas zlecenie, w tabeli okna głównego, a następnie kliknąć w przycisk *Policz koszt zlecenia*. Tutaj, gdy koszt zlecenia wynosi 0 zł, od razu uruchomi się kreator naliczania kosztów za usługę transportową. W przypadku gdy koszt został już wprowadzony, np. po-przez okno *Dodawania/Edycji zlecenia*, kreator najpierw pyta się użytkownika, czy jest pewien, że chce zamienić aktualną kwotę na kwotę wyliczoną przez mechanizm naliczania kosztów.

Kolejnym przyciskiem, który uruchamia funkcjonalności, związane z mechanizmem wnioskującym, jest przycisk *Przypomnij terminy*. Podczas wnioskowania przypomnienia o terminach nie uwzględniane są jednak żadne fakty na bieżąco uzyskiwane w trakcie dialogu z użytkownikiem, jak to miało miejsce w poprzednich przypadkach. Fakty, uwzględniane przez mechanizm, są już bowiem zapisane w bazie danych w postaci dat różnych wydarzeń, związanych z działalnością firmy transportowej. Użycie przycisku *Przypomnij terminy*, powoduje wyświetlenie zbliżających się terminów w oknie informacyjnym *Ekspert*.

Proces przypomnienia terminów dostarcza również informacji o braku uzupełnienia wiadomości, związanej z czasem przeprowadzenia danej procedury. Nie dzieje się tak wyłącznie w zakresie informacji o terminach zawartych w poszczególnych zleceniach, gdyż uzupełnianie ich jest konsekwentnie egzekwowane przez system. W procesie wnioskowania udział biorą cztery reguły, które określają przedział czasu przed danym wydarzeniem, dla którego przypomnienie ma być sygnalizowane. Osobna reguła decyduje o przypomnieniach dla terminów dostaw, kolejna dla terminów badań lekarskich kierowców, następna dla terminów badań technicznych i ubezpieczeń pojazdów, a jeszcze inna dla terminów badań technicznych i ubezpieczeń zestawów doczepianych.

Ostatnim przyciskiem okna głównego, przy pomocy którego uruchamiane są mechanizmy wnioskujące systemu, jest przycisk *Puste przebiegi*. Również on uruchamia okno informacyjne o nazwie *Ekspert*, dla wybranego z tabeli zlecenia, czyli towarzyszy temu podobna procedura, jak w przypadku przycisku *Policz koszt zlecenia*.

Eliminacja pustych przebiegów związana jest ściśle z terminami zleceń. To za pomocą terminów określa się, czy dane zlecenia można połączyć w jeden transport, czy też mają one być wykonywane po sobie, oczywiście jeżeli w realizacji tego zlecenia bierze udział ten sam środek transportu. Gdy pojazdy są różne, dla dwóch różnych zleceń, ich terminy ustalane są niezależnie od siebie.

Do zarządzania elementami bazy danych został skonstruowany interfejs akwizycji wiedzy. Służy on do dodawania nowych reguł do systemu, modyfikacji już istniejących, oraz usuwania niepotrzebnych bądź nieaktualnych reguł. Aby uruchomić interfejs, należy w oknie głównym systemu kliknąć menu o nazwie *Zarządzanie regułami*. Uruchamia się wówczas okno podstawowe z tabelą prezentującą wszystkie zapisane w bazie wiedzy reguły. Tabela pozwala sortować poszczególne kolumny, a także dowolnie zmieniać ich rozmiar. Wiersze danych nie są jednak przeznaczone do edycji, użytkownik może jedynie przeglądać treść tabeli i zaznaczać zawarte w niej obiekty. Okno wyposażone jest w cztery przyciski. Przycisk *Powrót* zamyka okno *Zarządzanie regułami* i przywraca okno główne systemu. Przycisk *Usuń regułę* powoduje wyświetlenie zapytania o poświadczenie decyzji usunięcia i w przypadku odpowiedzi twierdzącej usuwa regułę z tabeli i z bazy wiedzy systemu. Pozostałe dwa przyciski odpowiedzialne są za dodawanie i edycję reguł. Dla dodawania reguł panel uruchamia się niewypełniony, natomiast dla edycji jest on wypełniony danymi wybranej reguły.

Sprawdzanie poprawności wprowadzonych informacji, które odbywa się po kliknięciu na przycisk *Zapisz regułę* dotyczy wyłącznie tego, czy uzupełnione zostały wszystkie pola tekstowe, wyświetlane w panelu. Ze względu na różnorodność wprowadzanych informacji, jakiegokolwiek inne sprawdzanie nie przyniosłoby wymiernych rezultatów. Oprócz tego, za dodawanie i modyfikację reguł odpowiedzialny jest inżynier systemu bądź ekspert, który dysponuje odpowiednią wiedzą, by poprawnie konstruować reguły.

Obiekty klasy *Reguła* posiadają następujące cztery atrybuty:

- numer identyfikacyjny reguły, nadawany automatycznie przez system podczas zapisu tworzonego obiektu do bazy wiedzy,
- nazwa reguły, określająca jej przyporządkowanie do typu rozwiązującego poszczególne problemy decyzyjne,
- warunek, zawierający zdanie logiczne, którego spełnienie (określenie jako prawda), egzekwuje uruchomienie działania lub też stwierdzenie konkluzji,
- konkluzja, bądź też działanie następujące po spełnieniu warunku.

Atrybuty w obiektach klasy *Reguła*, zapisanych w bazie, są łańcuchami znaków, dokładniej interpretowane są dopiero przez mechanizm wnioskujący, który wyposażony jest w metody „czytające” znaczenie wprowadzonych danych.

4. Testowanie systemu LOGISER

Testowanie systemu LOGISER było nieodłączną czynnością wykonywaną każdorazowo i wielokrotnie dla sprawdzenia poprawności działania aplikacji i wyeliminowania błędów kodu. Wiązało się ono również z oszacowaniem zgodności funkcjonalnych elementów systemu z założeniami określonymi podczas przystępowania do jego realizacji. Testy dla skontrolowania prawidłowego działania funkcji, takich jak przydzielanie pojazdu i kierowcy do zlecenia, eliminacja pustych przebiegów czy liczenie kosztów określonych jako podstawowe cele budowy tegoż systemu, były przeprowadzane dla wielu różnorodnych danych wejściowych.

Podczas implementacji warstwy integracji rozpoczęły się pierwsze testy modułowe,

mające na celu wykrycie nieprawidłowości podczas komunikacji systemu z bazą danych. Konstruowane były odpowiednie klasy testujące, jak na przykład *KierowcaJpaControllerTest*, sprawdzające poprawność działania poszczególnych metod klas kontrolerów, komunikujących się z modulem utrwalania danych w bazie. Testy te obejmowały metody, które zapisywały, modyfikowały, usuwały oraz wyszukiwały obiekty utrwalanych klas. Przygotowane zestawy danych testowych, po wykonaniu sprawdzanych metod, poddawane były inspekcji w pliku bazodanowym przy użyciu systemu zarządzania bazą danych pgAdmin III. Kolejne testy jednostkowe przeprowadzane były dla systematycznie powstających klas interfejsu graficznego użytkownika. Metody te, głównie sprawdzające poprawność wprowadzanych przez użytkownika informacji, były kontrolowane pod względem zwracania poprawnych wyników swojej pracy.

Wraz ze wzrostem ilości elementów, wchodzących w skład systemu LOGISER, coraz większą rolę zaczęły odgrywać testy integracyjne. Obejmowały one ponownie komunikację z bazą danych, która została uskuteczniiona w jednej klasie *SystemRulles*. Obiekt tej klasy, tworzony przy starcie systemu, przekazywany jest kolejnym, uruchamianym przez użytkownika modułom, zapewniając w ten sposób, bezpośredni i łatwy dostęp do danych źródłowych z każdego miejsca ciała klas, wykorzystujących informacje z bazy danych. Jedną z metod omawianej klasy jest wielokrotnie testowana i korygowana metoda zapisu do bazy danych, która odpowiedzialna jest również za prawidłowe nadawanie obiektom numerów identyfikacyjnych. Dzięki tej klasie kod programu stał się bardziej przejrzysty a integracja poszczególnych modułów łatwiejsza w realizacji.

Testami integracyjnymi objęte były również wszystkie metody inicjalizujące pola paneli interfejsu graficznego, danymi zapisanymi w bazie danych. Sprawdzana była poprawność bindowania tabel oraz list rozwijanych z danymi źródłowymi. Szczególnie intensywnie testy te przeprowadzane były dla utworzonego w późniejszej fazie budowy systemu, mechanizmu wnioskującego. Jest on ukryty w klasie, stanowiącej również interfejs graficzny, pozwalający na zadawanie pytań i prezentację wyników o nazwie *Ekspert*. Zapewnienie dobrego i niezakłóconego przepływu informacji, otrzymywanych od użytkownika tudzież informacji zapisanych w bazie danych, a przede wszystkim reguł systemu oraz odpowiednie operowanie nimi w komunikacji międzymodułowej było sprawą bardzo istotną. Prawidłowa komunikacja międzymodułowa, szczególnie w ujęciu komunikacji z mechanizmem wnioskującym, przekładała się na generowanie odpowiednich i wyczerpujących ekspertyz.

Testowanie programu, pod kątem generowanych przez niego ekspertyz, można zaliczyć do testów systemowych. Testowana była bowiem zintegrowana całość aplikacji a nie tylko pojedyncze metody czy kilka modułów komunikujących się między sobą.

We wszystkich wymienionych poziomach testów oprogramowania szczególną rolę odegrał *Debugger*, stanowiący część środowiska programistycznego *NetBeans*. Za pomocą punktów przerywania tzw. *BreakPoint* możliwe było zatrzymanie wykonania programu w dowolnym momencie oraz kontrola wartości zmiennych, na których wykonywane były operacje. Ponadto wszelkie błędy zgłaszające wyjątki, które nie zostały odpowiednio oprogramowane i skutkują niepoprawnym wykonaniem programu, wyświetlane są w oknie *Output* środowiska *NetBeans*. Dzięki formie, w jakiej są one wyświetlane (odnośnik do liniiki kodu), możliwa jest dokładna lokalizacja miejsca, w którym wystąpił wyjątek.

Podczas testowania zdolności systemu do przyporządkowywania odpowiednich środków transportu poszczególnym rodzajom towarów, został opracowany szereg danych wejściowych. Dane są przykładowymi zleceniami przewozu różnych materiałów, a także osób.

System LOGISER tworzony był głównie z myślą o firmach, zajmujących się przewozem towarów. Dopuszcza on jednak zlecenia przewozu osób. Zleceniodawca w tym przypadku chciałby przewieźć niewielką ilość osób, na wycieczkę zakładową w góry. Liczba osób to siedem. Dyspozytor, po kontakcie ze zleceniodawcą, odpowiada na pytania, które zadaje mu system.

Zaprezentowane przykłady dowodzą, że system ekspertowy LOGISER radzi sobie z takimi sytuacjami, jak wykrywanie niewykorzystanego miejsca w ładowni, czy też detekcji pustych przebiegów, dla podobnego rodzaju towarów. We wszystkich przypadkach, ustalanie terminów załadunków i dostaw następuje adekwatnie do okoliczności.

Następną funkcjonalnością systemu, poddaną wielu testom, była metoda naliczania kosztów za wykonanie usługi przewozu i związana z tym polityka promocji, o którą system ma dbać. Samo naliczanie kosztów nie jest zbyt skomplikowane, a reguły uaktywniane w tym procesie, związane są ze średnią prędkością pojazdu na trasie oraz liczbą godzin pracy kierowcy. Gdy czas potrzebny na wykonanie przewozu przekroczy zakładane normy, oszacowane w regułach oraz parametrach systemu, to z tego tytułu kierowcy należy się dodatkowa kwota związana z dietą bądź noclegiem, która zostaje wliczona w koszt wykonania usługi przewozu.

Dla części mechanizmu, odpowiedzialnego za przypominanie terminów przewozów, przeprowadzone testy również nie wykazały uchybień, związanych z błędnym informowaniem użytkownika o zbliżający się terminach. Podobne wyniki przyniosły testy, odpowiedzialne za sprawdzenie poprawności pozostałych paneli systemu, związanych z ręcznym wprowadzaniem informacji do bazy danych. Zarówno kontrola poprawności wprowadzanych przez użytkownika informacji, jak również zapisu danych do bazy danych nie stwierdziła żadnych uchybień, a inspekcja pliku bazodanowego potwierdziła prawidłowość zawartych w niej danych.

5. Uwagi końcowe

System LOGISER to narzędzie w pełni wspomagające pracę dyspozytora firmy spedycyjnej. Umożliwia on przechowywanie danych niezbędnych do prawidłowego funkcjonowania firmy oraz sprawne wykonywanie na nich różnych operacji. Dane te ponadto stanowią wiedzę wykorzystywaną do spełniania priorytetowej roli pełnionej przez system, jaką jest zarządzanie procesami logistycznymi. Założenia, jakie były przyjęte podczas przystępowania do budowy aplikacji, zostały spełnione w zadowalającym stopniu.

Funkcje spełniane przez system poddane były skrupulatnym testom, które utrwaliły jego zalety. Na poziomach kontroli poprawności wprowadzanych danych, komunikacji programu z bazą wiedzy oraz poprawnego generowania ekspertyz nie stwierdzono uchybień. Ze względu na podejście do problemu, które zakładało szeroki zakres zastosowań systemu LOGISER, w niektórych przypadkach posiadana przez system wiedza powinna zostać rozszerzona i sprecyzowana. Szczególnie powinno mieć to miejsce dla potrzeb firm, zajmujących się specjalistycznym transportem, np. chemikaliów lub innych płynnych substancji, które nie powinny ze sobą wchodzić w reakcje chemiczne.

Zbudowany system z pewnością można rekomendować do zastosowań praktycznych. Decyzje, jakie pomaga podejmować system, szczególnie cenne będą dla niedoświadczonych dyspozytorów i pomoże on zdobyć im doświadczenie zawodowe. Dodatkowym atutem jest czytelny i intuicyjny interfejs graficzny, wyposażony ponadto w szereg pojawiających się podpowiedzi. Praca z programem jest więc łatwa a dostarczane ekspertyzy rzetelne.

Literatura

1. Buchalski Z.: The Role of Symbolic Representation of Natural Language Sentences in Knowledge Acquisition for Expert System. Polish Journal of Environmental Studies, Vol.16, No.4A, 2007, pp.40-43.
2. Buchalski Z.: Analysis of expert system application possibilities for the needs of doctors decision in various disease diagnostics. Information Systems Architecture and Technology, J. Świątek, L. Borzemski, A. Grzech, Z. Wilimowska (eds.), Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2009, pp.49-58.
3. Chromiec J., Strzemieczna E.: Sztuczna inteligencja. Metody konstrukcji i analizy systemów eksperckich. Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1994.
4. Niederliński A.: Regułowo-modelowe systemy ekspertowe. Pracownia Komputerowa Jacka Skalmierskiego, Gliwice, 2006.
5. Rutkowski L.: Metody i techniki sztucznej inteligencji. PWN, Warszawa, 2012.
6. Buchalski Z.: Computer Advisory-Decision System for the Logistics Services Support. Polish Journal of Environmental Studies, Vol.18, No.3B, 2009, pp.53-57.
7. Radzikowski W.: Komputerowe systemy wspomaganie decyzji. PWE, Warszawa, 1990.
8. Twardowski Z.: Inteligentne systemy wspomaganie decyzji w strategicznym zarządzaniu organizacją gospodarczą. Wydawnictwo Akademii Ekonomicznej, Katowice, 2007.
9. Zieliński J.: Inteligentne systemy w zarządzaniu. Teoria i praktyka. PWN, Warszawa, 2000.
10. Ciesielski M.: Logistyka w biznesie. Polskie Wydawnictwo Ekonomiczne, Warszawa, 2006.
11. Golemska E.: Kompendium wiedzy o logistyce. PWN, Warszawa, 2010.
12. Januła E., Truś T., Gutowska Ż.: Spedycja. Wydawnictwo Difin S.A., Warszawa, 2011.
13. Sikorski P., Zembrzycki T.: Spedycja w praktyce. Polskie Wydawnictwo Transportowe, Warszawa, 2006.

Dr inż. Zbigniew BUCHALSKI
Instytut Informatyki, Automatyki i Robotyki
Politechnika Wroclawska
50-372 Wrocław, ul. Janiszewskiego 11/17
tel.: (0 71) 320 32 92
e-mail: zbigniew.buchalski@pwr.wroc.pl