Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej
Katedra Automatyki i Inżynierii Biomedycznej

Autoreferat rozprawy doktorskiej

Analiza i synteza systemu przetwarzania sygnałów opisujących gesty i elementy subkodu mimicznego w zastosowaniu do komunikacji z osobami niesłyszącymi

mgr inż. Wojciech Kozioł

Promotor: prof. zw. dr hab. inż. Wiesław Wajs

Kraków 2014

1. Wprowadzenie

Przezwyciężenie hermetyczności nauk informatycznych w stosunku do innych dyscyplin badawczych jest potrzebą chwili. Dotyczy to zwłaszcza nauk humanistycznych, gdzie zwykło się ograniczać rolę informatyki do zwykłego narzędzia do rozwiązywania problemów technicznych, matematycznych, statystycznych, gromadzenia danych opisowych czy wyszukiwania artykułów naukowych, a często również sprowadzających rolę komputera do zwykłej maszyny do pisania. Postęp na tym polu jest ciągle niewystarczający, a niezrozumienie dla pozycji poznawczych, jak też możliwości i oczekiwań badawczych pozornie czasem odległych dziedzin, bywa źródłem nieporozumień, szczególnie dotkliwych dla tych nauk w konsekwencji niewykorzystania nowoczesnych narzędzi analitycznych i inspiracji płynących z rozwoju technologii informatycznej. Współczesna epistemologia podkreśla integralność poznania w nauce. Niektóre dyscypliny dzielą więc pewne sztuczne i dogmatyczne podziały. Przykładem ukazującym, że taka koegzystencja jest możliwa jest zbliżające się metodologicznie do nauk ścisłych językoznawstwo. Lingwiści już dawno dostrzegli przydatność techniki komputerowej. Z wykorzystaniem takich narzędzi próbowano realizować koncepcje generowania tekstu zaproponowaną przez Noama Chomskiego. Skutecznie rozwinęły się badania nad frekwencją jednostek i stylem, a ostatnio – analizy korpusu tekstów (tzw. językoznawstwo korpusowe) jako skuteczne narzędzie badania tekstu. Nie pomyślano jednak o informatyce jako o narzędziu wspomagającym komunikację interpersonalną.

Zainteresowanie językiem jako hierarchicznie zorganizowaną strukturą znaków (zdań) prostych i złożonych poddanych rygorom tekstotwórczym w naturalny sposób kieruje uwagę na poziom strukturalizacji wypowiedzi i opisu syntaktycznego języka. Przydatność paradygmatu informatycznego dla stworzenia formalnego opisu funkcjonowania języka jako systemu znaków jest bezdyskusyjna. Otwiera to także pole badań porównawczych w zakresie analiz strukturalnych różnych języków i automatyzacji procesów przekładu. Dowodzi tego najlepiej bujny rozwój translatoryki komputerowej, upowszechniającej się w zasobach Internetu, zob. kolejne wersje coraz doskonalszych translatorów typu Translatica, Google Translator, Bing Translator i wiele innych.

Sprawność i skuteczność takiego informatycznego narzędzia objawić się winna także w przypadku przekładu pomiędzy językami o odmiennych podstawach substancjalnych, jakim jest naturalny język foniczny i język migowy. Wydaje się to założeniem oczywistym i w pełni uzasadnionym na gruncie lingwistyki. Jednak w odniesieniu do komunikacji pomiędzy dwoma językami o tak różnej strukturze, ograniczenia i niedostatki warsztatu naukowego obu dyscyplin stają się bardziej widoczne i wymagają próby nowego podejścia.

Problematyka tłumaczenia tekstów z polskiego języka pisanego na polski język migowy i odwrotnie – z polskiego języka migowego na polski język mówiony jest wielce skomplikowanym, zagadnieniem, a także zadaniem, zarówno od strony językowej, jak również informatycznej. Jego złożoności nie można sprowadzić do zasobności leksyku. Potwierdzone w źródłach leksykograficznych dysproporcje rzędu kilka : kilkadziesiąt tysięcy jednostek nie mogą przesłonić faktu, że mamy, w przypadku języka migowego do czynienia z równie sprawnym co polszczyzna narzędziem komunikacji, nawet na poziomie dokonań artystycznych. Jak widać, nie ma podstaw do utożsamiania mowy gestów z kodem ograniczonym. Sytuacja ta nakazuje skupić uwagę na właściwym skonstruowaniu płaszczyzny odniesienia i podstaw ekwiwalencji przekładu.

2. Cel i zakres pracy

Celem pracy jest zbudowanie i przedstawienie zasad działania systemu wspomagającego proces komunikacji osób słyszących z osobami niesłyszącymi. W tym celu zaprojektowano i wykonano testową wersję systemu informatycznego w technologii 3D. Niedostatek pozycji leksykograficznych w postaci słowników języka migowego stworzył obiektywną konieczność uporania się z jeszcze jednym zadaniem, jakim była próba określenia i usystematyzowania zasobu gestów, w formie zunifikowanej – w postaci aplikacji komputerowej, tj. tłumacza oraz słownika języka migowego. Chodzi tutaj o opracowanie i wykonanie w miarę kompletnej i jednocześnie otwartej propozycji słownika języka migowego. Słownik taki mógłby stać się użytecznym narzędziem i źródłem wiedzy językowej dla zainteresowanych środowisk, osób i instytucji.

Aspekt praktyczny, a zarazem walor ogólnospołeczny pracy, wynika z respektowania idei przeciwdziałania społecznemu wykluczeniu osób upośledzonych, pokonywania uprzedzeń i barier komunikacyjnych przez wspomaganie procesu komunikacji w konkretnych sytuacjach życiowych, np. w urzędach, bankach, przychodniach zdrowia, szpitalach, dworcach, a także i w miejscu pracy. Bariery te są przyczyną izolowania się osób głuchoniemych od reszty społeczeństwa, nieuczestniczenia w kulturze, życiu obywatelskim itd., zatem narzędzie to może wydatnie poprawić jakość ich życia, dostarczając użytecznego wsparcia podczas załatwiania wielu spraw bytowych. Studia socjologiczne i oficjalne dokumenty państwowych instytucji zajmujących się tymi problemami[1] dowodzą, że osoby niesłyszące chcą aktywnie uczestniczyć w życiu społecznym i podejmować pracę w zakładach przemysłowych, instytucjach państwowych, nie mają jednak po temu dostatecznego wspomagania. Taką rolę mogą z powodzeniem spełnić współczesne środki techniki komputerowej. Niniejsza rozprawa doktorska wykorzystuje w związku z tym nowoczesne osiągnięcia IT do zbudowania systemu informatycznego, wspomagającego komunikację ludzi słyszących z osobami niesłyszącymi, a więc tym samym ułatwiającego im funkcjonowanie w społeczeństwie. Kwestia ta jest niebagatelna z punktu widzenia całości społeczeństwa, bowiem szacuje się, że w Polsce żyje około 100 tysięcy osób, które mają problemy ze słuchem[2]. Spora część spośród nich to ludzie w wieku produkcyjnym, zdolni do podjęcia pracy zarobkowej. Szczególnie boleśnie sytuacja ta dotyka ludzi młodych, chętnie korzystających z dobrodziejstw współczesnej cywilizacji i techniki. Dowodem na to, że dla państwa sprawa osób niesłyszących jest ważna, było podjęcie dodatkowych zobowiązań wobec interesującej nas grupy społecznej, wyrażonych w Ustawie Sejmu RP z dnia 19 sierpnia 2011 r. o języku migowym i innych środkach komunikowania[3].

Realizacja systemu tłumaczącego niesie ze sobą wiele korzystnych możliwości zastosowań:

- porozumiewanie się z osobami niesłyszącymi w razie wystąpienia barier komunikacyjnych, np. w przypadku braku kontaktu bezpośredniego;
- wzrost samodzielności i operatywności osób niesłyszących w miejscu pracy;
- tworzenie interaktywnych instrukcji obsługi stanowisk pracy, maszyn i procesów technologicznych, przeznaczonych do samodzielnej aktywacji;
- komunikowanie się na poziomie zawodowym (specjalistycznym) z przełożonym, instruktorem w miejscu pracy;
- komunikowanie się w warunkach niewspółmierności kompetencji językowych

1 np. PFRON

2 http://www.tea.org.pl/userfiles/file/Seminaria/niepelnosprawnosc_sluchowa_mczajkowska-kisil.pdf

3 Pełny tekst z poprawkami i uzupełnieniami jest dostępny pod adresem: http://isap.sejm.gov.pl/DetailsServlet?id=WDU20112091243.

i komunikacyjnych, np. relacja petent – urzędnik przy warunkach dysfunkcjonalności pisma jednej strony i nieznajomości języka migowego drugiej strony i in.

Problem, rozwiązany w ramach niniejszej rozprawy doktorskiej jest wieloaspektowy i skomplikowany. Skala związanych z nim trudności znajduje wyraz choćby w tym, że mimo podejmowanych wysiłków jak dotychczas nie wdrożono automatycznego translatora z języka polskiego do języka migowego. Należałoby tu przede wszystkim wskazać na:

- interdyscyplinarność problematyki – do realizacji zadania konieczna jest wiedza z zakresu zaawansowanej gramatyki języka polskiego, gramatyki języka migowego, programowania z wykorzystaniem najnowszych narzędzi informatycznych, animacji komputerowej 3D, w tym analiza i projektowanie mimiki twarzy;
- brak ekwiwalencji na poziomie pojedynczych znaków – słów. Konieczność tłumaczenia sytuacyjno-opisowego, która wynika z nierównomiernego rozkładu zasobności leksykalnych, por. słownik ogólny języka polskiego zawierający kilkaset tysięcy jednostek leksykalnych ze słownikiem języka migowego, który zawiera ok. 5000 leksemów[4];
- wysokie koszty profesjonalnej aparatury umożliwiającej rejestrowanie przestrzennych koordynatów kodu migowego;
- niewystarczająca integracja świata nauki z gospodarką (obszarem wdrożeń).

Z naukowego punktu widzenia praca przynosi także próbę opisu przestrzeni i ruchu oraz mimiki twarzy jako substancji i formy kodu językowego. Otwiera perspektywy systematycznych badań nad tym obszarem zagadnień z wszechstronnym wykorzystaniem narzędzi informatycznych, objaśnienia istoty zjawiska semantyzacji przestrzeni w komunikacji migowej, wyznaczenia cech dystynktywnych formy znaku ideograficznego tego typu zarówno w odniesieniu partykularnym – systemu polskiego, jak też ogólnym – systemu komunikacji gestycznej jako takiej. Domenę poznawczą i zarazem zaplecze teoretyczne prezentowanego opracowania wyznacza informatyka. Warsztat naukowy pracy korzysta więc przede wszystkim z wielu użytecznych narzędzi programistycznych, bazodanowych i graficznych, z dobrym skutkiem używanych w rozwijających się coraz bardziej komputerowych badaniach języków naturalnych i pracach nad automatycznym przekładem struktur komunikacji (wypowiedzeń i tekstów). W związku z powyższym główną tezę pracy można ująć następująco:

Możliwe jest uzyskanie „substytutu gramatyki" języka migowego poprzez realizację znaczników semantycznych w bazie danych języka polskiego i zastosowanie struktur grafowych w języku Prolog.

Zakładany cel pracy wraz z uzasadnieniem tezy implikował następujące zadania badawcze:

- opracowanie i zaprojektowanie architektury systemu;
- zaprojektowanie i utworzenie bazy danych przechowującej gesty języka migowego wraz ze znakami subkodu mimicznego twarzy;
- utworzenie aplikacji narzędziowych do wprowadzania i edycji danych językowych,

4 Konfrontacja dostępnych źródeł leksykograficznych i internetowych, jak też fakt twórczej aktywności rozmaitych ośrodków w Polsce wskazuje, że jest to liczba raczej zaniżona. Można oczekiwać, wobec wzrastającego zainteresowania językiem migowym, znacznego przyrostu liczbowego nowych znaków w najbliższych latach.

obróbki gestów języka migowego i projektowania mimiki twarzy;
- przygotowanie modelu 3D – awatara pokazującego gesty;
- realizację zadań związanych z rejestracją gestów w studio motion capture: zaprojektowanie i rozmieszczenie układu markerów na ciele aktora pokazującego gesty, uszycie specjalnej kamizelki na potrzeby rejestracji gestów, wyposażenie awatara w układ szkieletowy oraz markery. Wymagane było również sprzęgnięcie układu szkieletowego z markerami;
- utworzenie skryptów umożliwiających import surowych danych, interpolację brakujących fragmentów sygnału opisującego gesty języka migowego, wygładzanie sygnału, przekształcenie sygnału i jego eksport do głównej bazy danych;
- utworzenie serwisu danych wyciągającego odpowiednie dane z bazy wiedzy językowej (bazy danych języka polskiego);
- utworzenie serwera translacji – reguł dokonujących analizy głębokiej tekstu na podstawie danych otrzymanych od serwisu translacji, reguł ustalania ekwiwalencji tekstu do języka migowego, opracowanie odpowiednich struktur danych opisujących struktury głębokie zawarte w tekście;
- utworzenie aplikacji głównej spełniającej zadania interakcji z użytkownikiem i wizualizacji gestów;
- realizację łącza danych pomiędzy serwisem danych językowych, serwerem translacji i aplikacją główną;
- dokonanie testów w celu weryfikacji poprawności generowania ekwiwalentnych struktur języka migowego w postaci sekwencji gestów.

Jak pokazała praktyka badań wszystkie te zadania tworzyły układ wzajemnie uzupełniających się i warunkujących elementów. Należało je zatem wykonywać równolegle. Przyniosły one autorowi wiele doświadczeń wykorzystanych w innowacyjny sposób.

3. Zawartość pracy

Praca niniejsza ma charakter interdyscyplinarny, co znajduje wyraz zarówno w jej kompozycji, zawartości, jak też w przestrzeganiu zasady komplementarności opisu. Jest to kwestia istotna, pozwalająca na uniknięcie zbędnych powtórzeń.

Całość pracy, zgodnie z przyjętym celem i założeniami opisu, rozpada się na cztery zasadnicze części. Pierwszą z nich stanowi obszerny wstęp stanowiący teoretyczne i metodologiczne zaplecze podejmowanych badań. Starano się w nim zachować należyte proporcje obu korespondujących dyscyplin naukowych: informatyki i językoznawstwa. Uwagi na ten temat zawierają podrozdziały wstępne: Ogólne wprowadzenie, uzasadnienie podejmowanych badań, sprecyzowanie celu i zakresu pracy. W dalszej kolejności dokonano przeglądu literatury naukowej polskiej i obcojęzycznej oraz przedstawiono metodologiczne podstawy opracowania.

Podstawy teoretyczne opracowania wyłożono w części drugiej. Daje ona ogólny obraz struktury komunikacji językowej i cech wyróżniających język osób niesłyszących na tle naturalnego języka fonicznego. Przede wszystkim zwrócono tu uwagę na stworzenie założeń analizy syntaktycznej tekstu uwzględniającej właściwości konotacyjne leksemów, zwłaszcza pełniącego zdaniotwórczą rolę czasownika, strukturą fraz nominalnych i przymiotnikowych, a także elementami pozostającymi w relacji współrzędnej (połączenia szeregowe) oraz członów stojących nie wykazujących cech akomodacji składniowej, czyli okoliczników. W trakcie analiz rozwiązano też problemy wynikające z polemiczności jednostek i homonimii. Starano się w tym celu maksymalnie wykorzystać wymagania

semantyczne ograniczające łączliwość jednostek w tekście tzw. znaczniki semantyczne, obecne w semantyce generatywnej. Skorzystano w tym względzie z inspiracji płynących z najważniejszych opracowań lingwistycznych na gruncie polskim (prace K. Polańskiego, R. Grzegorczykowej, I. Bobrowskiego i in.). Największym wyzwanie w procesie translacji okazał się brak wiarygodnego opisu gramatyki języka migowego. Obiektywnie stanowi to ogromne utrudnienie w ustaleniu relacji pomiędzy językiem migowym a fonicznym i samych możliwościach przekładu. Z naukowego punktu widzenia najbardziej zachęcające na tym polu są nowatorskie badania komunikacji migowej prowadzone w Stanach Zjednoczonych. Osiągnięte tam wyniki nie są jednak w pełni wiarygodne i nie dają możliwości przeniesienia na grunt polski.

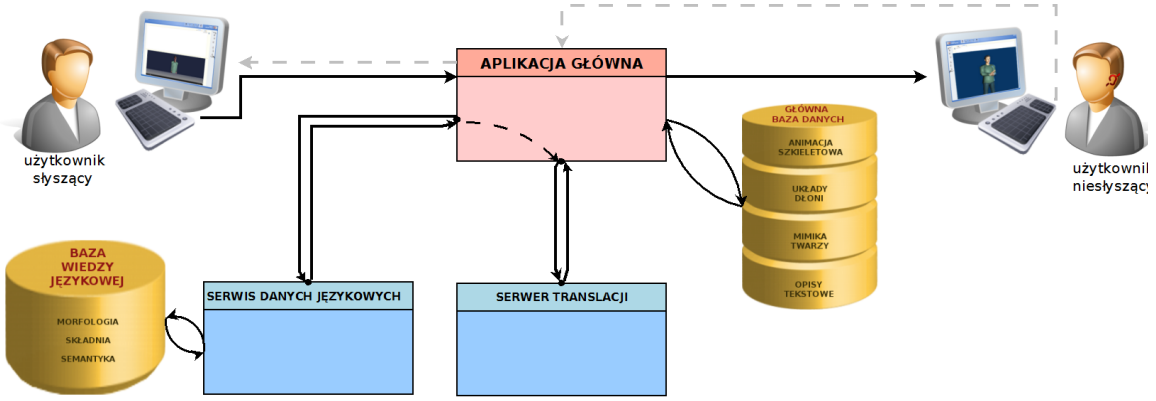
Część trzecia stanowi zasadniczy trzon pracy. Przedstawiono w niej na wymaganym poziomie szczegółowości budowę systemu tłumaczącego. W pierwszej kolejności opisano ogólną architekturę systemu tłumaczącego. W drugiej kolejności przedstawiono aplikacje narzędziowe, utworzone na potrzeby gromadzenia i obróbki danych wykorzystywanych przez system tłumaczący. Należą do nich: aplikacja do projektowania mimiki twarzy, aplikacja wykorzystywana w procesie rejestracji i obróbki gestów języka migowego w przestrzeni 3D oraz aplikacja umożliwiająca wprowadzanie i edycję danych w bazie wiedzy językowej w zakresie morfologii, składni i semantyki. Następnie omówiono proces wizualizacji gestów języka migowego. W ramach tego zadania opisano krótko wykorzystywane w systemie tłumaczącym techniki animacji, przedstawiono konstrukcję siatki awatara 3D, pokazującego gesty języka migowego oraz typy układów kostnych zastosowanych w procesie tworzenia animacji. Zreferowano również proces akwizycji gestów języka migowego oraz proces tworzenia wypowiedzi w języku migowym, na który składają się odpowiednio zsynchronizowane tory animacji rąk, układów dłoni i mimiki twarzy. W dalszej kolejności opisano bazę wiedzy językowej, w tym własności kategorialne leksemów języka fonicznego, reprezentację poszczególnych leksemów w bazie wiedzy; omówiono też kategorie leksykalno-pojęciowe języka migowego i ich przekazywanie, przedstawiono również reprezentację właściwości semantyczno-gramatycznych różnych części mowy w bazie wiedzy. W ramach budowy systemu tłumaczącego opisano także główną bazę danych systemu tłumaczącego, przechowującą dane niezbędne przy generowaniu wypowiedzeń języka migowego w technologii 3D, jak również algorytmy dokonujące tłumaczenia tekstów języka polskiego na język migowy. Obejmują one koncepcję i realizację procesu segmentacji tekstu, budowania struktur wymagań, dopasowań i ekwiwalencji oraz generowanie struktury wyjściowej tworzącej odpowiedni komunikat umożliwiający wizualizację przetłumaczonej treści. Proces analizy i syntezy tekstu zobrazowano kilkoma typowymi, a jednocześnie instruktywnymi przykładami użycia. Opisano również aplikację główną odpowiadającą za interakcje z użytkownikiem i wizualizację treści w języku migowym.

Całości dopełnia syntetyczne zakończenie, wskazujące na możliwości aplikacji osiągniętych wyników badań i ich twórczego rozwinięcia w aspekcie badań nad sztuczną inteligencją i uczącymi się systemami analizy języka.

4. Architektura systemu tłumaczącego

Już we wstępnych fazach budowy systemu uznano, że ze względu na stopień złożoności systemu powinien on posiadać budowę modułową. Rozwiązanie takie sprzyja skalowalności systemu, umożliwia łączenie różnych technik informatycznych i użycie odpowiednich paradygmatów programowania. Rozdzielenie systemu na moduły pozwala również rozmieścić je na różnych maszynach połączonych w sieci, umożliwiając rozproszenie obliczeń. W przyszłości zaś umożliwi to udostępnianie usługi tłumaczenia przez Internet. Na rys.1, poniżej przedstawiono modułową architekturę systemu

dokonującego translacji zdania języka polskiego na język migowy.



Rysunek 1: Schemat architektury systemu tłumaczącego.

W systemie wyróżnić można następujące moduły:

Moduł główny – zwany aplikacją główną lub końcową – zaznaczony kolorem czerwonym. Jest on najważniejszą częścią systemu, odpowiedzialną za interakcję z użytkownikiem. Pobiera z konsoli tekst w języku polskim i przekazuje go do *serwisu danych językowych*. Odbiera dane od serwisu danych językowych i przekazuje je do serwera translacji. Odbiera od serwera translacji przetłumaczoną treść w postaci struktury danych przechowującej listę gestów do wymigania wraz z układami dłoni i mimiką twarzy modelu. W końcu wizualizuje on gesty przy użyciu technologii 3D. Aplikacja główna spełnia zatem rolę warstwy prezentacji danych w systemie. Zaimplementowano ją w języku C#.

Serwis danych językowych – to proces odpowiedzialny za obsługę dostępu do *bazy wiedzy językowej*. Odbiera on od aplikacji głównej tekst wprowadzony przez użytkownika i parsuje go do listy słów. Wyszukuje w *bazie wiedzy językowej* wszystkie przeciążenia[5] danego wyrazu. Serwis translacji pobiera pełny zakres przeciążeń danego słowa. Jedno z przeciążeń, ustalone w drodze analizy, będzie pasujące do znaczenia danej jednostki sensu w segmencie lub zdaniu. Serwis danych językowych pobiera zestaw cech morfologicznych właściwych dla danego słowa oraz atrybuty określające łączliwość składniową i semantyczną danej jednostki z innymi jednostkami w zdaniu. Dodatkowo pobierany jest również unikalny klucz tekstowy reprezentujący nazwę migu w *głównej bazie danych*. Pole to jest kluczem zespalającym migi w *głównej bazie danych* z leksemami w *bazie wiedzy językowej*. Pobrany zestaw informacji zostaje przekształcony do postaci odpowiednich faktów języka Prolog, a następnie zwrócony do aplikacji głównej, która prześle go do serwera translacji. Serwis danych językowych zaimplementowano w języku C#.

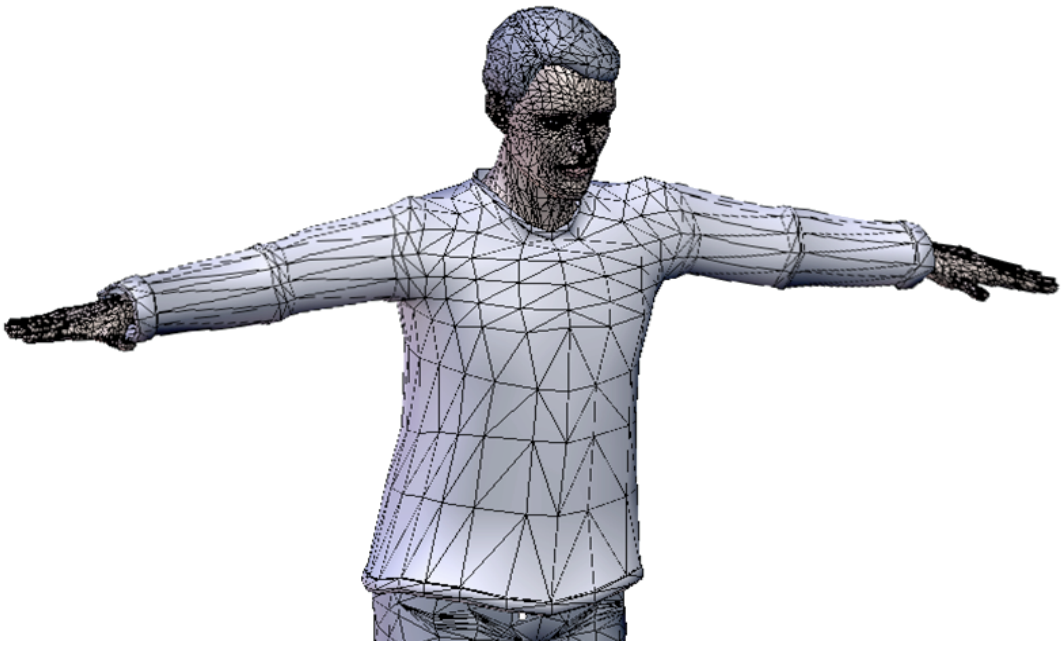
Serwer translacji – proces odpowiedzialny za analizę głęboką zdań języka polskiego i ekwiwalencję do języka migowego. Do jego zadań należy: pobranie danych językowych w postaci faktów języka Prolog wysyłanych przez aplikację główną, budowanie dynamicznej wiedzy z pobranych danych, segmentacja zdań złożonych, analiza głęboka zdań składowych, ustalenie ekwiwalencji na język migowy oraz synteza do postaci komunikatów języka migowego w formie struktury danych z przetłumaczonym komunikatem i wysłanie przetłumaczonej struktury do aplikacji głównej. Serwer translacji zaimplementowano w języku Prolog.

5 Przeciążenia oznaczają wystąpienia tego samego wyrazu w różnych znaczeniach, dla różnych części mowy oraz w postaci konglomeratów, tj. członów wielowyrazowych stanowiących w całości oddzielną jednostkę sensu.

5. Konstrukcja siatki aktora 3D

Wizualizacja gestów języka migowego powinna cechować się realizmem, dużą dokładnością i szczegółowością. Przed rozpoczęciem modelowania przejrzano zasoby sieciowe w poszukiwaniu darmowych siatek 3D. Najbardziej odpowiednim rozwiązaniem okazało się użycie oprogramowania MakeHuman[6]. Jest to oprogramowanie OpenSource umożliwiające generowanie różnego rodzaju siatek 3D dla ciała człowieka, cechujących się przy tym dużą szczegółowością. Biorąc jednak pod uwagę względy optymalizacyjne i obciążenia obliczeniowe podczas renderowania, zdecydowano się na użycie tylko niektórych części ciała wygenerowanego modelu.

Dla przekazu treści w języku migowym szczególnie istotnymi częściami ciała są dłonie i głowa. Dłonie są najważniejsze, gdyż podczas migania składane są w odpowiednie układy. Układy te muszą być czytelne, jeśli gesty mają zostać rozpoznane – niektóre migi wykonuje się tym samym ruchem i w tej samej pozycji względem ciała, różnią się tylko układem dłoni i oczywiście znaczeniem gestu. Głowa aktora jest również bardzo ważna, gdyż obrazuje ona mimikę twarzy modelu i emocje. Pozwala również czytać słowa z ruchu warg. Osoby niesłyszące, dodajmy, wspomagają swoją komunikację, czytając z ruchu warg, szczególnie podczas kontaktów z osobami słyszącymi. Przyjęto zatem rozwiązanie, w którym głowa i dłonie modelu 3D zostały wycięte z siatki wygenerowanej oprogramowaniem MakeHuman, pozostałe zaś części siatki awatara zostały domodelowane ręcznie i scalone w jedną siatkę przy użyciu oprogramowania Blender[7].
Podczas modelowania siatki należało zwrócić szczególną uwagę, by poszczególne części ciała modelu odpowiadały proporcjom aktora, którego ruchy były nagrywane. Rzetelnie przygotowana siatka pozwoli uzyskać lepszej jakości, bardziej realistyczną animację. Modelując ubiór awatara, trzeba zwrócić baczniejszą uwagę na staranniejsze opracowanie anatomicznych miejsc zgięć stawów tj. łokci i barków. Na rys. 2 pokazano gotową siatkę aktora 3D wraz z nałożoną poglądową teksturą. Widać z niej wyraźnie dużą przewagę liczbową werteksów na dłoniach i głowie modelu w stosunku do reszty ciała awatara.



Rysunek 2: Widok siatki modelu 3D wraz z nałożonymi teksturami.

Ostatecznie podział siatki ze względu na liczbę wierzchołków wygląda następująco:

6 http://www.makehuman.org/
7 http://www.blender.org/

dla głowy awatara ok 7 tys. werteksów, dla obu dłoni ok 3 tys. werteksów, pozostała część siatki złożona jest z ok 3 tys. werteksów, w sumie siatka awatara składa się z ok 11 tys. wierzchołków.

Na siatkę modelu nałożono trzy rodzaje tekstur: koloru, odbić (ang. specular), oraz map normalnych (ang. normal mapping). Mapy tekstur i odbić znacznie udoskonalają ekspresję awatara, zob. rys. poniżej. Efekt renderowania map wypukłości i odbić uzyskiwany jest przy użyciu techniki *PixelShader* i *VertextShader*. W celu zwiększenia wydajności procesu animacji przyjęto, że mapy wypukłości i mapy normalnych będą renderowane jedynie dla rąk i głowy. Jak już wspominano, na jakość przekazu informacji w języku migowym w dużej mierze wpływa wyrazistość i czytelność dłoni i twarzy animatora.

Zwiększenie wydajności renderingu uzyskano również dzięki zastosowaniu obiektu *Mesh* dostępnego w technologii DirectX. Ponieważ siatka modelu składa się z podstawowych elementów zwanych prymitywami, których boki przylegają do siebie, informacja o każdym z wierzchołków powtarzała się średnio pięciokrotnie. Użycie obiektu *Mesh* do reprezentacji siatki awatara, eliminuje zupełnie redundancję werteksów w siatce modelu, co skutkuje polepszeniem efektywności w procesie renderowania animacji.

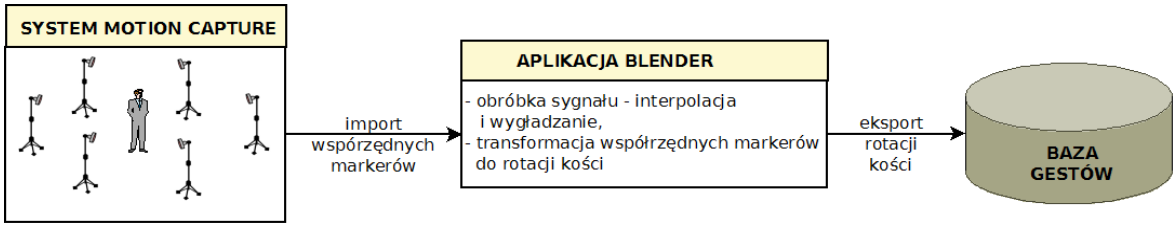
Rysunek 3: Widok awatara renderowanego z mapami wypukłości i odbić.

Sterowanie siatką awatara 3D obywa się przy użyciu systemu kości połączonych ze sobą hierarchicznie, przy czym każda z kości posiada swój własny przestrzenny układ współrzędnych, zależny od położenia układu współrzędnych rodzica. Ruch kości modelu odbywa się poprzez zmianę rotacji kości w przestrzeni w zakresie opisanym wartościami kątów Eulera ψ, θ, ϕ[8], opisujących przestrzenny zwrot wektora kości o danej długości.

Akwizycja gestów języka migowego dokonywana jest przy użyciu systemu wizyjnego motion capture. Dane zarejestrowane systemem motion capture zawierają informacje opisujące przestrzenne położenia markerów w czasie. Sterowanie awatarem

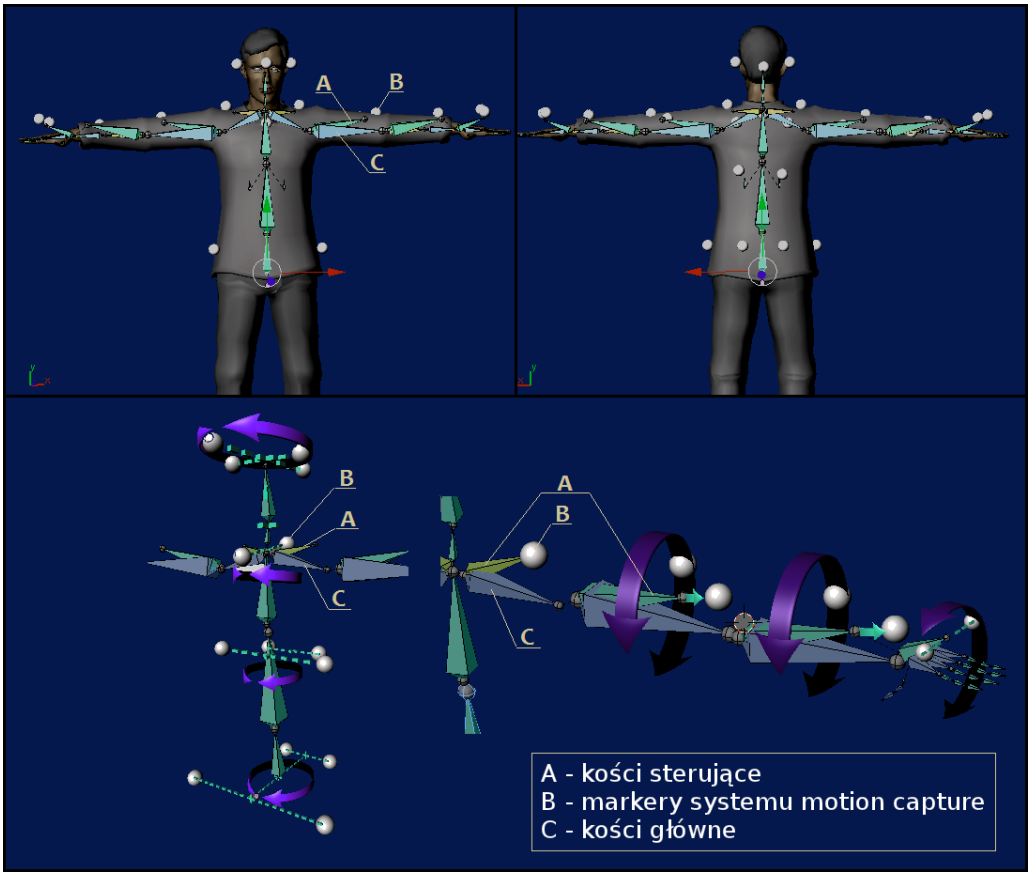
8 http://pl.wikipedia.org/wiki/K%C4%85ty_Eulera

odbywa się natomiast poprzez zmianę wartości kątów Eulera dla poszczególnych kości w układzie szkieletowym modelu względem czasu. Należało zatem sprzęgnąć ze sobą te dwa układy. Wykorzystano w tym celu aplikację Blender, która posiada moduł odwrotnej kinematyki (*Inverse kinematics solver),* dokonujący przekształcenia współrzędnych markerów na rotacje kości. Wykonano również zestaw skryptów w języku Python, realizujących następujące zadania: import współrzędnych markerów zarejestrowanych systemem motion capture do programu Blender; obróbkę zaimportowanych danych, czyli interpolację brakujących fragmentów sygnału oraz wygładzenie toru ruchu markerów; przekształcenie współrzędnych markerów do rotacji kości oraz zapisanie tak przetworzonej animacji do głównej bazy danych w postaci obiektów binarnych, opisujących ruch dla danego gestu języka migowego. Schemat blokowy akwizycji gestów języka migowego przedstawia poniższy rysunek.



Rysunek 4: Schemat blokowy pozyskiwania animacji 3D.

Wykorzystanie systemu motion capture do akwizycji gestów języka migowego wymaga opracowania sposobu rozmieszczenia markerów na ciele animatora.



Rysunek 5: Układ kości i markerów w aplikacji Blender.

System motion capture rozpoznaje marker gdy jest on widziany przez co najmniej dwie kamery. Z uwagi na to że nie można rozmieszczać markerów w obszarach

anatomicznych zgięć stawów, rozkład markerów na ciele animatora musi być podyktowany wedle kryteriów optymalnej widoczności markerów przez kamery systemu rejestrującego. Jest to przyczyną rozbieżności pomiędzy usytuowaniem na siatce modelu układu szkieletowego i układu markerów. Problem sprzęgnięcia markerów nieznajdujących się w miejscach zgięć stawów rozwiązuje zastosowanie dodatkowych kości sterujących w aplikacji Blender, zob. rys. 5.

6. Działanie systemu tłumaczącego

Kiedy użytkownik systemu wprowadzi tekst w *aplikacji głównej* i uruchomi proces tłumaczenia, wykonywany jest następujący tok zadań. Tekst wpisany przez użytkownika zostaje wysłany z aplikacji głównej do *serwisu danych językowych*, gdzie zostaje sparsowany na poszczególne wyrazy. Następnie serwis sięga po dane dotyczące poszczególnych wyrazów do *bazy wiedzy językowej*. Serwis danych językowych przeszukuje również bazę pod względem wystąpienia konglomeratów[9] w tekście. Pobrany zestaw danych przetworzony zostaje do postaci faktów języka Prolog i przesłany do *aplikacji głównej*. *Aplikacja główna* odbiera dane i przekazuje je do *serwera translacji*. Po odebraniu danych *serwer translacji* ładuje je dynamicznie do pamięci, tworząc dla siebie dynamiczną, operacyjną bazę wiedzy. Baza ta zawiera tylko niezbędne informacje, które biorą udział w przetwarzaniu treści wprowadzonej przez użytkownika. *Dynamiczna baza wiedzy*, w odróżnieniu od statycznej bazy wiedzy językowej[10], ze względu na swój mały rozmiar nie obciąża pamięci operacyjnej i nie absorbuje procesora przeszukiwaniem dużych przestrzeni danych. Dzięki temu algorytmy działają znacznie szybciej. Uruchomienie po raz kolejny procesu translacji wymazuje z pamięci dynamiczną bazę wiedzy i tworzy ją na nowo na podstawie aktualnego tekstu, który wprowadził użytkownik.

Po zbudowaniu w pamięci dynamicznej bazy wiedzy uruchamiane są algorytmy dokonujące segmentacji zdań wielokrotnie złożonych w tekście (relacje hipotaksy i parataksy). Segmentacja ta polega na utworzeniu tablicy struktur opisujących segmenty, które powstają dla każdego przeciążenia czasownika lub imiesłowu. Dla każdego członu zdaniowego dla zdania złożonego może powstać wiele segmentów, w zależności od liczby przeciążeń danego czasownika/imiesłowu w *dynamicznej bazie wiedzy*. Struktury opisujące segmenty zawierają domyślne zakresy początku i końca segmentów, identyfikator, numer przeciążenia i pozycję czasownika/imiesłowu pełniącego rolę orzeczenia w segmencie, oraz identyfikator do struktury mieszczącej w sobie struktury elementów wymaganych i dopasowanych (zidentyfikowanych w segmencie), oraz struktury powstałe po procesie ustalania ekwiwalencji. Zakresy określające rozpiętość segmentów są używane w dalszych etapach analizy podczas wyszukiwania poszczególnych fraz i elementów wewnątrz segmentu. Po zakończeniu segmentacji następuje proces budowania *dynamicznej bazy wiedzy* w postaci struktur, opisujących poszczególne segmenty zdań. Wyróżnić można struktury wymagań składniowych i

9 Wyrazy często łączą się ze sobą w jednostki dwu lub więcej wyrazowe tworząc oddzielną jednostkę sensu, tak więc APARAT FOTOGRAFICZNY to nie APARAT ORTODONTYCZNY czy APARAT SŁUCHOWY.

10 Statyczna baza wiedzy w języku Prolog jest to baza faktów zawarta w kodzie programu języka Prolog i jest ona konsolidowana w trakcie kompilacji kodu do pliku wykonalnego. Podczas uruchamiania pliku wykonalnego lub uruchamiania kodu w interpreterze języka Prolog jest ona ładowana do pamięci operacyjnej. Jeśli załadować do niej dane językowe opisujące właściwości gramatyczne i semantyczne wyrazów języka polskiego, baza zajmie większą część pamięci operacyjnej. Również przeszukiwanie takiej bazy wiąże się z dużym kosztem czasowym, co ogranicza wydajność i możliwości algorytmów. Bardzo kłopotliwa jest również zmiana w strukturze danych takiej bazy i jej skalowalność, czego nie można powiedzieć o dynamicznej bazie wiedzy. W pierwszych fazach rozwoju systemu tłumaczącego i algorytmów używano bazy statycznej o ograniczonym, podstawowym zakresie słownictwa.

semantycznych, definiujące zasady łączliwości składniowej i semantycznej czasownikowego orzeczenia w poszczególnych segmentach z obligatoryjnymi frazami nominalnymi dla danego segmentu. Definiowane są również wymagania dotyczące wystąpienia niektórych wyrazów w segmencie[11]. Następnie uruchamiane są algorytmy wyszukania wymaganych elementów w segmentach na podstawie zdefiniowanych wcześniej struktur wymagań. Otrzymywane są w ten sposób obligatoryjne dla segmentów frazy **nominalne** w pozycji podmiotu, dopełnienia bliższego i dopełnienia dalszego. Kolejnym etapem jest porównywanie struktur definiujących wymagania ze strukturami otrzymanymi w procesie przeszukiwania (dopasowania). Jeśli dla struktury opisującej segment istnieje rozbieżność pomiędzy elementami wymaganymi a znalezionymi, struktura w takiej formie uznawana jest za niekompletną i dewiacyjną. Zostaje zatem usunięta. Wskutek takiej eliminacji pozostają tylko te przeciążenia, które spełniają warunki segmentu. System akceptuje niekompletne struktury jedynie w przypadku, gdy brakuje wypełnienia tylko dla jednego pola konotacyjnego, a w schemacie semantycznym nie występuje gniazdo rozwinięcia dla zdania podrzędnego. Dodatkowym warunkiem akceptacji takiej niekompletnej struktury jest występowanie po danym segmencie w analizowanym tekście odpowiedniego gniazda rozwinięcia zapowiadającego występowanie (przyłączenie) zdania podrzędnego.

Proces eliminacji niekompletnych segmentów nie gwarantuje, że po jego zakończeniu pozostanie tylko jeden docelowy wariant segmentu dla każdego członu wypowiedzenia złożonego. Zdarza się, że dla czasownika mającego wiele znaczeń schematy konotacyjne są zbliżone do siebie lub nawet identyczne. System wygeneruje wtedy więcej poprawnych segmentów. W takim przypadku wybierze schemat, kierując się statystyką. W praktyce sprowadza się to do wybrania pierwszej wypełnionej struktury[12]. Na tym etapie liczba segmentów przechowywanych w tablicy segmentów jest równa rzeczywistej liczbie segmentów, tj. członów zdania złożonego.

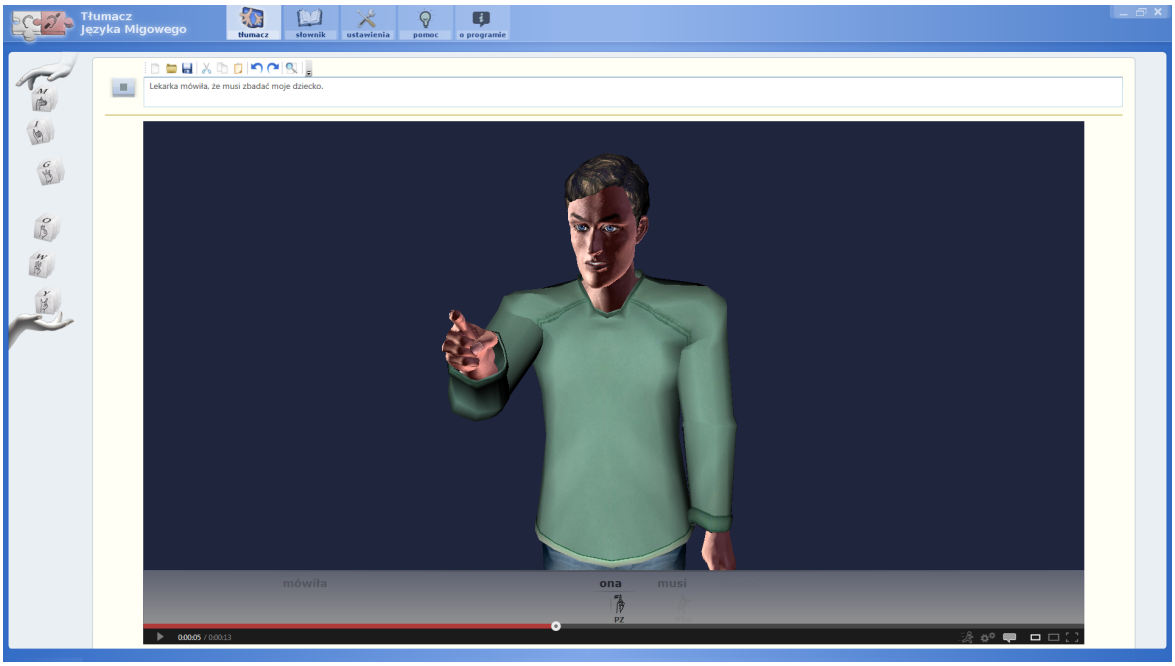
Po wybraniu odpowiedniej struktury konotacyjnej dla każdego członu zdania złożonego następuje etap wyszukiwania nieobligatoryjnych części segmentu, tzn. okoliczników oraz przydawek czyli określeń atrybutywnych należących do struktur fraz nominalnych. Wykrycie elementów obligatoryjnych ustala ostatecznie rozpiętość każdego z segmentów oraz domyka proces analizy tekstu. Dodać należy, że wszystkie elementy zidentyfikowane w procesie analizy zapisywane są w strukturze elementów zidentyfikowanych jako identyfikatory przeciążeń.

Zakończenie procesu analizy umożliwia uruchomienie procesu ustalania ekwiwalencji i syntezy komunikatu do języka migowego. Ustalanie ekwiwalencji polega na zbudowaniu struktury symetrycznej do struktury dopasowań, która mieści w sobie znalezione wyrazy w postaci identyfikatorów ich przeciążeń. Każde przeciążenie zapisane w dynamicznej bazie wiedzy posiada argument, w którym zakodowany jest wskaźnik do animacji gestu przechowywanego w głównej bazie danych. Argument ten nosi nazwę *blender_nazwa_migu*. System przejdzie zatem przez całą strukturę dopasowań i będzie budował strukturę ekwiwalencji, zamieniając identyfikatory przeciążeń na wartości argumentu *blender_nazwa_migu* tych przeciążeń. W przypadku, gdy argument *blender_nazwa_migu* nie jest ustawiony, system doda animację, dla której wyraz będzie literowany przy wykorzystaniu systemu palcowego. Ponieważ symetria struktury ekwiwalencji zachowuje informacje o frazach nominalnych, system zdolny jest poukładać wszystkie zidentyfikowane wewnątrz segmentów frazy według porządku podyktowanego

11 Przykładowo dla trzeciej osoby trybu rozkazującego orzeczenie wymaga wystąpienia modulantu NIECH, a orzeczenie wyrażane w czasie przyszłym wymaga słowa posiłkowego tworzonego od czasownika BYĆ w określonej formie, tj.: *będę*, *będziesz*, *będzie* itp.

12 Twórcy *Słownika syntaktyczno-generatywnego czasowników polskich* (Polański 1980-1992), układając kolejność schematów dla czasowników cechujących się polisemią, opisywali je według częstości (frekwencji) ich występowania w tekście.

składnią języka migowego.



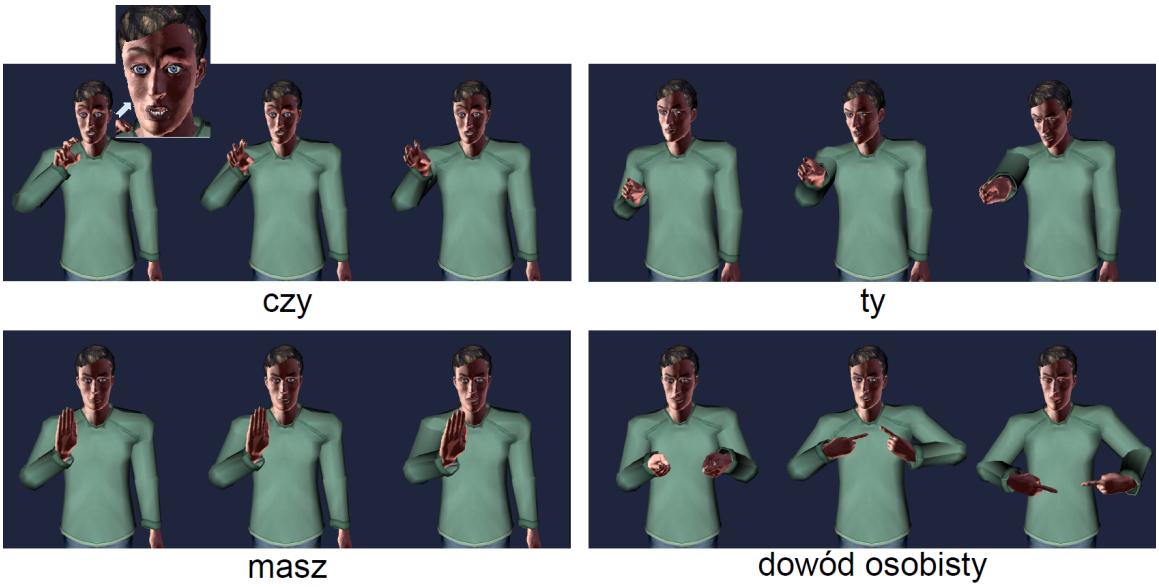
Rysunek 6: Aplikacja w trakcie wizualizacji przetłumaczonego zdania: *Lekarka mówiła, że musi zbadać moje dziecko*.

Rozpoznanie typu zdania i jego wewnętrznej struktury pozwala również określić, które wyrazy w zdaniu powinny dodatkowo przenosić informacje dotyczące mimiki twarzy modelu.

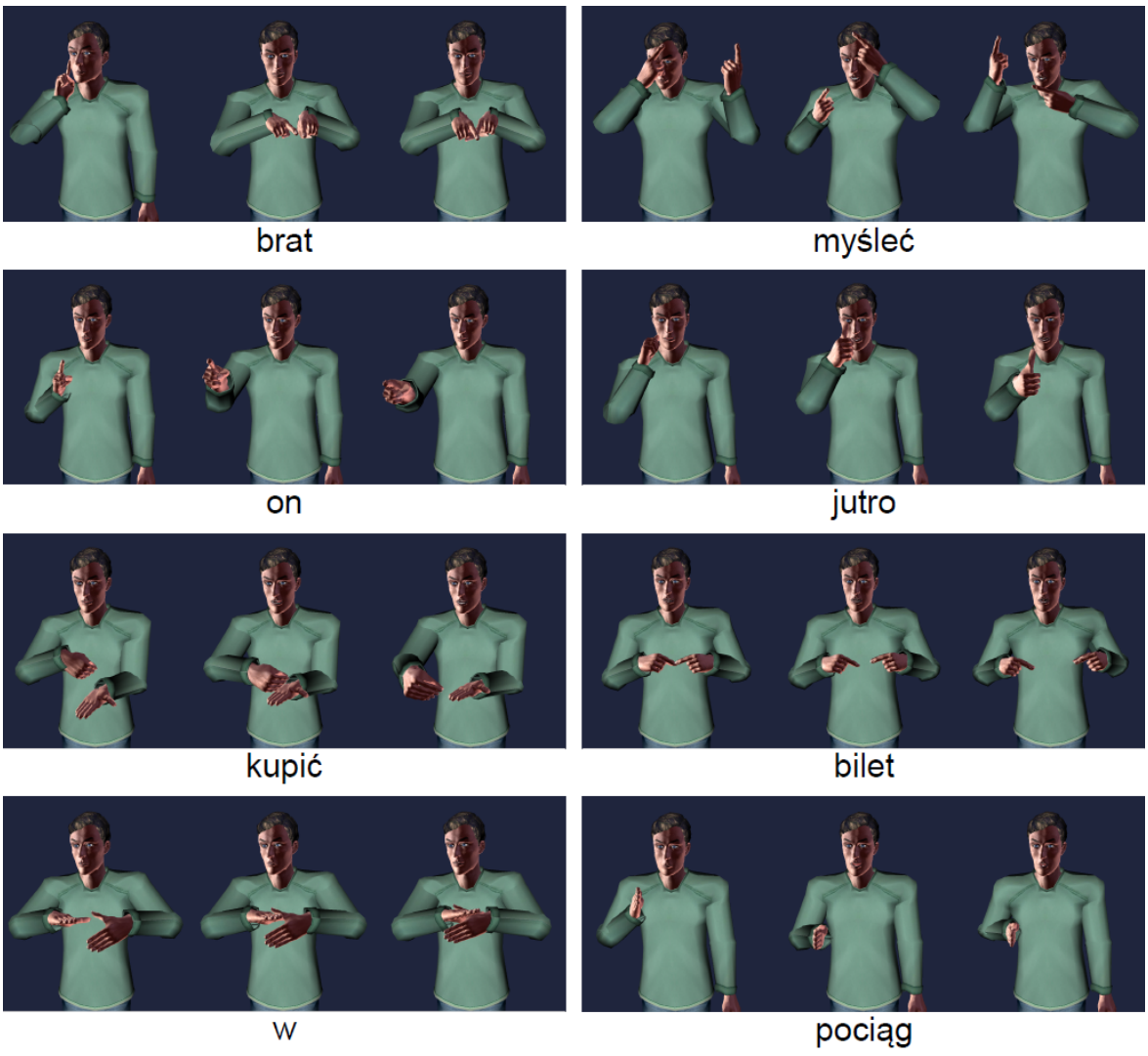
W efekcie procesu syntezy powstaje lista ułożonych sekwencyjnie członów, które zawierają informacje umożliwiające wygenerowanie animacji modelu. Każdy z członów zawiera identyfikator gestu języka migowego, identyfikator mimiki twarzy oraz tekst, który awatar będzie wypowiadał w trakcie migania. Na podstawie otrzymanych danych *aplikacja główna* pobiera z *głównej bazy danych* odpowiednie obiekty binarne opisujące ruch przestrzenny modelu dla gestów języka migowego oraz przekształcenia ust i mimiki twarzy. W dalszej kolejności obliczane są transjenty dla gestów i mimiki twarzy. Ostatecznie w aplikacji końcowej powstaje złożona struktura danych, sterująca zachowaniem się modelu, która tworzy animację w technologi 3D, przekazującą treść w języku migowym. Na rys. 6 pokazano zrzut ekranu przedstawiający działania aplikacji tłumaczącej.

Poniżej przedstawiono przykłady użycia:

a) Przykładowe zdanie z mimiką pytajną: *Czy masz dowód osobisty?*



b) Przykładowe zdanie złożone: *Brat myślał, że kupi jutro bilet w pociągu*



7. Podsumowanie

Niniejsza praca przynosi informatyczny opis systemu tłumaczącego z języka polskiego na język migowy z wykorzystaniem szerokiego instrumentarium pojęciowego współczesnej lingwistyki. Najważniejszym z punktu widzenia osiągniętych wyników jest w przekonaniu autora skonstruowanie bazy wiedzy językowej i opracowanie algorytmów analizy tekstów pisanych i struktury ekwiwalencji dla komunikatów języka migowego. Równie istotnym rezultatem jest skompletowanie możliwie pełnej bazy gestów języka migowego w technologii 3D, z dopracowaniem toru subkodu mimicznego spełniającego bardzo ważną rolę w porozumiewaniu się osób niesłyszących.

W procesie akwizycji gestów wykorzystano wizyjny system rejestracji ruchu – motion capture. Wprowadziło to znaczny stopień innowacyjności w stosunku do istniejących tego typu projektów w Polsce. Jego efektem jest uzyskanie wysokiego realizmu i naturalności ruchu awatara 3D.

Konstrukcja odzwierciedla wielotorowość przekazu typową dla komunikacji osób niesłyszących i łączy w sobie ścieżkę ideograficzną – znaki migowe, daktylograficzną – literowanie układami palców i dłoni, mimiczną i tekstową. Ścieżki te uzupełniają się wzajemnie w procesie komunikacji, tworząc układ komplementarny.

Perspektywy rozwojowe systemu wiążą się przede wszystkim z potrzebą dalszego zbliżania automatycznego przekładu do naturalnego sposobu komunikacji osób niesłyszących. Roboczo ten postulat można wiązać z koniecznością opracowania i zastosowania modelu przeciążeń pragmatycznych, zdolnych do ujęcia warunków sytuacyjno-kontekstowych, związanych z interakcyjnością przekazu treści. Można się o tym przekonać analizując bardzo częste przypadki elizji elementów np. przyimków możliwych w danej sytuacji do wywnioskowania przez aktywnego odbiorcę.

O wiele istotniejsze wydaje się jednak stworzenie podstawy do skonstruowania kompatybilnego systemu tłumaczącego w układzie dwukierunkowej komunikacji. Wiąże się to z najtrudniejszym z naukowego punku widzenia zadaniem skutecznego analizatora komunikatów języka migowego, rozpoznawanych w czasie rzeczywistym.

W związku z osiągniętymi celami badawczymi wyłaniają się dalsze kwestie. Chodzi tu przede wszystkim o możliwość dalszych zastosowań służących aktywizacji zawodowej osób niesłyszących, przeciwdziałaniu ich wykluczeniu społecznemu i poprawie ogólnego komfortu życia i poziomu aktywności społecznej. System ma na celu eliminację barier komunikacyjnych, jakie występują pomiędzy społeczeństwem słyszącym i niesłyszącym, głównie przez wspomaganie procesu komunikacji w konkretnych sytuacjach życiowych, np. przy załatwianiu spraw w urzędach, bankach, przychodniach, dworcach, miejscach pracy, itp. Ten wart podkreślenia walor społeczny zrealizowanego projektu wiąże się bezpośrednio z zapisami Ustawy Sejmu RP z dnia 19 sierpnia 2011 r. o języku migowym i innych środkach komunikowania i z jej praktyczną realizacją.

Można w związku z tym przewidywać oprócz upowszechnienia samodzielnej aplikacji tłumaczącej w wersji desktopowej również skonstruowanie kiosków, wspomagających komunikację z osobami niesłyszącymi w miejscach użyteczności publicznej, powstanie odpowiedniej wersji aplikacji na telefony komórkowe oraz świadczenie usług translatorskich przez Internet.