WYKORZYSTANIE SKRYPTÓW W PROGRAMIE VISUM, NA PRZYKŁADZIE MODELU RUCHU WOJEWÓDZTWA ŚLĄSKIEGO

Adam Konarski
mgr inż., Ove Arup & Partners, ul. Królewska 16, 00-103 Warszawa, tel.: +48 71 719 6872, e-mail: adam.konarski@arup.com

Streszczenie: *Artykuł ma na celu pokazanie praktycznego wykorzystania skryptów pisanych w języku programowania Python w celu usprawnienia korzystania z oprogramowania Visum. W artykule przedstawione są dwa przykłady. Pierwszym z nich jest tworzenie sieci transportu zbiorowego na podstawie bazy danych uzyskanej od operatora portalu służącego do planowania podróży. Drugi przykład to automatyczne wykonywanie obliczeń oraz eksport wyników z plików programu Visum do innych formatów.*

Słowa kluczowe: *Visum, skrypty, COM*

1. Wprowadzenie

W niniejszym artykule przedstawiono kilka sposobów wykorzystania programów oraz skryptów stworzonych w języku Python, korzystających z obiektu COM programu Visum. Narzędzia te stworzono w celu usprawnienia prac przy zagadnieniach, pojawiających się podczas tworzenia oraz modyfikacji modeli podróży. Przedstawione przykłady dotyczą:

- Automatycznego generowania modelu sieci transportu zbiorowego na podstawie rozkładów jazdy oraz danych o lokalizacji przystanków, pozyskanych z różnych źródeł oraz w różnych formatach;
- Automatyzacji obliczeń dla wariantów rozwoju sieci oraz wielu horyzontów czasowych oraz zapisu wyników w formacie ułatwiającym dalsze obliczenia takie jak m. in. analizy ekonomiczne.

Program Visum poza standardowymi narzędziami do tworzenia sieci transportu zbiorowego wykorzystującymi graficzny interfejs daje użytkownikowi także możliwość używania poleceń z poziomu, tak zwanej konsoli poprzez dołączony do oprogramowania obiekt COM. Obiekt COM pozwala użytkownikowi na pełną kontrolę nad programem Visum za pomocą jakiegokolwiek języka programowania posiadającego bibliotekę komunikacji COM [1]. Umiejętność korzystania z funkcji programu poprzez polecenia z poziomu konsoli, z pominięciem interfejsu graficznego, wraz z możliwościami języka programowania Python dają użytkowniko-

wi ogromne możliwości tworzenia własnych narzędzi. Narzędzia te mogą służyć, między innymi do automatyzacji żmudnych bądź skomplikowanych czynności.

2. Tworzenie sieci transportu zbiorowego

Zespół projektowy podczas prac nad Planem Transportowym Województwa Śląskiego stanął przed zadaniem zbudowania modelu ruchu pasażerskiego na terenie całego województwa. Jednym z wyzwań podczas procesu modelowania była budowa sieci połączeń transportu zbiorowego uwzględniającej przystanki oraz przypisane im rozkłady jazdy. Bardzo szeroki zakres koniecznych do wprowadzenia danych uzasadniał przeanalizowanie możliwości automatyzacji procesu wprowadzania danych do programu Visum. Przykładowo podczas prac nad modelem ruchu Miasta Szczecin wprowadzenie 230 kursów transportu zbiorowego zajęło jednej osobie około tygodnia, na omawianym projekcie liczba kursów do wprowadzenia wynosiła ponad 18 000.

Manualne wprowadzanie danych, za pomocą graficznego interfejsu, jest rozwiązaniem zbyt mało efektywnym, aby jego zastosowanie było uzasadnione w tym przypadku. Zdecydowano, więc o próbie wykorzystania skryptów napisanych w języku programowania Python wykorzystujących obiekt COM Visum.

2.1. Pozyskiwanie danych

Podstawowym wymogiem przy wybranym trybie pracy są dane o odpowiedniej jakości. Jest to kwestia kluczowa dla powodzenia całego projektu. Wybrana, automatyczna metoda wprowadzania danych wymusza, aby były one dostępne w formacie, który będzie łatwy do interpretacji za pomocą algorytmów. Struktura danych musi być, zatem jak najprostsza oraz jak najbardziej konsekwentna.

W praktyce dane wystarczająco szczegółowe i w odpowiednim formacie nie są dostępne podczas pracy przy większości projektów. Zamawiający często przekazują dane o przewoźnikach oraz rozkładach jazdy w formie papierowej lub jako pliki PDF stworzone poprzez skanowanie papierowych oryginałów. Dane te, zanim będą odpowiednie do użycia muszą zostać zdigitalizowane, co jest żmudnym i czasochłonnym procesem. Łatwiejszym i bardziej niezawodnym sposobem pozyskiwania informacji jest korzystanie z danych znajdujących się w różnego rodzaju serwisach internetowych oraz bazach danych. Jeżeli dany serwis oferuje narzędzie potocznie zwane planerem podróży, wyszukujące połączenia transportu zbiorowego pomiędzy zadanym źródłem i celem podróży, serwis ten posiada bazę danych ze wszystkimi informacjami koniecznymi do zbudowania modelu sieci transportu zbiorowego. Metody pozyskania opisanych powyżej danych mogą być różne w zależności od tego, kto dysponuje prawami autorskimi.

W przypadku omawianego projektu dane zakupiono od serwisu „e-Podróżnik". Jest to ogólnopolski serwis, oferujący ogromną bazę kolejowych oraz autobusowych rozkładów jazdy. Zawiera on informacje o połączeniach międzymiastowych,

a także o komunikacji miejskiej w wybranych miastach Polski. Jako format danych przyjęto prostą formę tabelaryczną zapisaną w arkuszu kalkulacyjnym gdzie każdy element konkretnego kursu (Time Profile Item) został opisany w jednym wierszu.

Innym rozwiązaniem, jest korzystanie z danych udostępnianych w formacie Google Transit Feed [2]. Struktura tego formatu to seria plików tekstowych, w których zawarte są kompletne informacje na temat systemu transportu zbiorowego na wybranym obszarze. Z uwagi na globalny zasięg firmy Google może on stać się dominującym formatem w dziedzinie zapisu informacji o komunikacji zbiorowej. Powszechną praktyką wśród wielu miast na świecie jest publikowanie oraz nieodpłatne udostępnianie rozkładów jazdy w tej formie. Aktualnie w portalu Google Transit znaleźć można 6 polskich miast [3]. Są to: Białystok, Łódź, Olsztyn, Szczecin, Warszawa oraz Zielona Góra. Dzięki standaryzacji oraz zgodności danych z aktualnym stanem oferty przewozowej można nie tylko budować, ale także regularnie aktualizować modele ruchu miast w zakresie transportu zbiorowego.

Dodatkowym ułatwieniem w korzystaniu z formatu Google Transit są dostępne nieodpłatnie narzędzia do przetwarzania danych, takie jak biblioteka „GoogleTransitDataFeed" dla języka Python [4]. Narzędzie to umożliwia wczytanie kompletu plików tekstowych w odpowiednim formacie a następnie zbudowanie obiektu, którego metod i atrybutów można wygodnie używać podczas programowania.

Innym formatem popularnym zwłaszcza na terenie Wielkiej Brytanii i Irlandii jest format TransXChange [5] oraz powiązany z nim NaPTAN. Format ten został opracowany przez brytyjski Departament Transportu i jest oparty na języku XML. Stanowi on obowiązujący standard służący do zapisywania danych o transporcie zbiorowym w wymienionych krajach, co powoduje, że wiele organizacji udostępnia dane w tym formacie. Dane w formacie TransXChange są także obsługiwane przez popularny w Wielkiej Brytanii pakiet Accession służący do analiz dostępności do transportu zbiorowego oraz indywidualnego. Biblioteki GoogleTransitDataFeed zawierają narzędzia do konwersji pomiędzy formatem TransXchange a formatem wprowadzonym przez Google, zatem nie ma potrzeby tworzenia osobnych skryptów dla różnych formatów danych używanych w projektach.

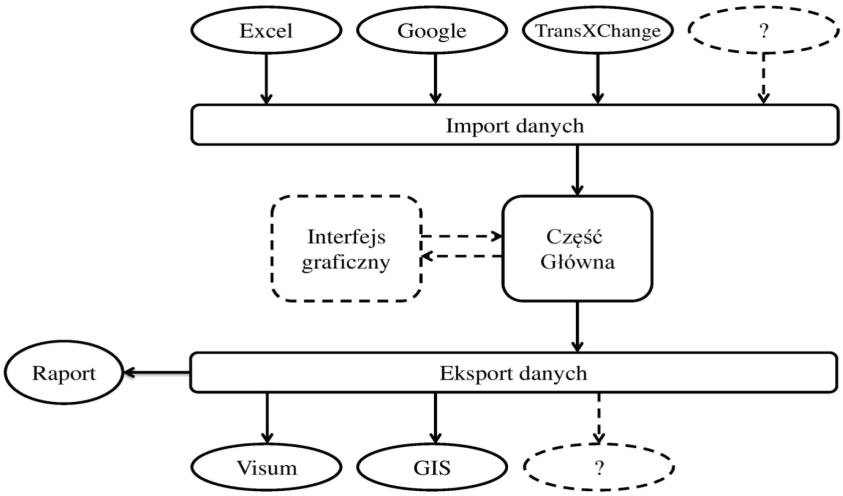
2.2. Założenia oraz sposób działania skryptu

Głównym założeniem, jakie przyjęto była maksymalna elastyczność i uniwersalność tworzonego narzędzia. W świetle mnogości opisanych wcześniej standardów zapisywania danych, istotna okazała się możliwość wykorzystywania go nie tylko na jednym, ale również na przyszłych projektach. Dodatkową opcją było również umożliwienie zapisu danych także dla innych celów niż tylko, jako element modelu w programie Visum. W związku z powyższymi wymaganiami skryptowi nadano budowę modułową. Moduły składające się na opisywane narzędzie to:

- Główna część, czyli obiekt reprezentujący sieć, bardzo zbliżony budową, w zakresie części dotyczącej transportu zbiorowego, do obiektu Visum.Net programu Visum;

- Część odpowiedzialna za import danych z różnych źródeł do obiektu głównego;
- Część odpowiedzialna za wprowadzanie danych zgromadzonych w głównym obiekcie do modelu (ewentualnie eksport danych do innego formatu na przykład, jako pliki w standardzie GIS) a także generowanie raportów ze szczegółowymi informacjami na temat ewentualnych błędów i ostrzeżeń.

Schemat budowy narzędzia przedstawiono na poniższym rys. 1.



Rys. 1. Schemat budowy opisywanego skryptu
Źródło: opracowanie własne

Powyższy podział daje możliwość łatwego dodawania kolejnych modułów w przyszłości. Przykładowo, zamiast wczytywania danych wejściowych z pliku Excel w krótkim czasie możliwe było opracowanie modułu pozyskującego dane z plików w standardzie Google Transit Feed a także moduł pozwalający na zapisywanie informacji o trasach w postaci plików SHP. Wszystko to przy minimalnym nakładzie pracy gdyż podstawowe funkcje znajdują się „wewnątrz" obiektu głównego, którego nie trzeba zmieniać. Jeżeli będzie to uzasadnione, możliwe jest również zaprojektowanie prostego, graficznego interfejsu użytkownika umożliwiającego wprowadzenie okien dialogowych do wyboru źródeł danych czy sposobu zapisu wyników.

2.3. Wprowadzanie danych do modelu

Odcinki kolejowe oraz drogowe wprowadzono do modelu za pomocą standardowych narzędzi. Źródłem danych były otrzymane od zamawiającego pliki SHP dla sieci drogowej oraz instrukcja Id-12 dla sieci kolejowej. Obszar opracowania został podzielony na rejony komunikacyjne odpowiadające podziałowi terytorial-

nemu na gminy. Z uwagi na to przystanki autobusowe znajdujące się na terenie jednej gminy zostały zagregowane. Przystanki kolejowe również wprowadzono na podstawie instrukcji Id-12 oraz podłączono je do rejonów komunikacyjnych w odpowiadających gminach.

Na podstawie informacji pozyskanych z serwisu e-podróżnik, do prac przy modelu ruchu województwa śląskiego wykorzystano dane zapisane jako arkusz kalkulacyjny programu Excel. Dane w arkuszu przedstawione były w postaci jednej tabeli. Każdy punkt na trasie (Line Route Item) konkretnego kursu transportu zbiorowego został opisany: nazwa linii, nazwa kursu, nazwą przewoźnika, nazwa przystanku początkowego oraz końcowego danego kursu, identyfikator przystanku a także czas przyjazdu oraz odjazdu z danego przystanku.

Każdemu przystankowi transportu zbiorowego w modelu przyporządkowano niepowtarzalny identyfikator. Po zakończeniu prac wstępnych przygotowano skrypt, który wprowadza do modelu w programie Visum wszystkie elementy sieci transportu zbiorowego:

- Przewoźników (Operators),
- Linie (Lines),
- Trasy (Line Routes),
- Elementy tras (Line Route Items),
- Kursy (Time Profiles),
- Elementy kursów (Time Profile Items).

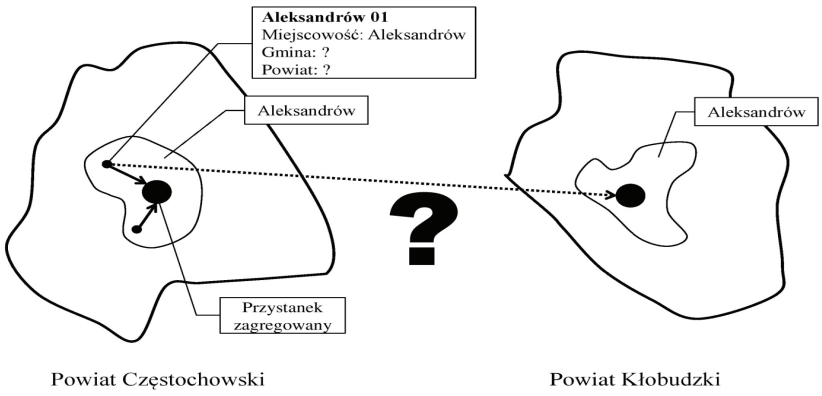
Proces wprowadzania wszystkich powyższych danych nie zajmował w tym przypadku więcej niż godzinę czasu maszyny.

2.4. Napotkane trudności

W trakcie pracy nad modelem napotkano na kilka technicznych przeszkód mających wpływ na efektywność wykorzystanych rozwiązań.

Najbardziej problematyczne okazało się wprowadzanie linii transportu zbiorowego, których przebiegi przecinały granicę opracowania. Jest to związane ze sposobem modelowania ruchu zewnętrznego. Wszystkie linie, których przebiegi wykraczały poza obszar opracowania powinny zostać przecięte a końce przebiegów podłączone do odpowiednich rejonów zewnętrznych. W tym celu należało zidentyfikować wszystkie miejsca przecięć pomiędzy trasami a granicą opracowania, a następnie wprowadzić poprawki do bazy danych o elementach kursów. W tym przypadku zadecydowano o wykorzystaniu metody manualnej. Poprawki te okazały się być jednak bardzo czasochłonne. W przyszłości rozwiązaniem zaistniałego problemu mogłoby być uwzględnienie w algorytmie skryptu „fałszywych" przystanków na granicy opracowania, podłączonych do odpowiadających im rejonów zewnętrznych a następnie wykrywanie, kiedy wprowadzana automatycznie trasa przechodzi przez taki przystanek. To powodowałoby zatrzymywanie procesu dodawania kolejnych elementów trasy. Czas przejazdu byłby interpolowany pomiędzy przystankiem poprzedzającym a kolejnym względem miejsca przecięcia.

Kolejnym ważnym czynnikiem była spójność wykorzystywanych danych. Ma ona znaczny wpływ na wyniki stosowania opisywanej metody. W opisywanym przypadku dysponowano danymi o położeniu przystanków transportu zbiorowego pozyskanymi z jednego źródła oraz danymi o rozkładach jazdy z innego źródła. Powstała zatem konieczność przyporządkowania każdemu przystankowi, w obu kompletach danych, niepowtarzalnego identyfikatora w celu przeprowadzenia automatycznego generowania kursów. Przy pracy nad omawianym modelem ruchu przystanki identyfikowane były na podstawie nazwy miejscowości, w której się znajdują a następnie agregowane do poziomu gmin. Nie przewidziano jednak, że brak danych o przyporządkowaniu nazwy miejscowości, w której znajduje się dany przystanek do odpowiadającego jej powiatu i województwa będzie stanowił problem. Zbyt mały stopień szczegółowości danych nie wystarczył do automatycznego, jednoznacznego określenia położenia przystanku, gdyż w Polsce nazwy miejscowości bardzo często się powtarzają. Ilustrację problemy przedstawiono na rys. 2.



Rys. 2. Ilustracja problemu przyporządkowania przystanku do odpowiedniej gminy
Źródło: opracowanie własne

Wiele problemów stworzyć może także sieć odcinków drogowych i kolejowych. Kłopotliwe są występujące na sieci nieciągłości, takie jak: brakujące odcinki, zamknięte relacje skrętne czy też nieodpowiednie przyporządkowanie dozwolonych systemów transportowych. Tworzenie tras transportu zbiorowego za pomocą skryptu korzysta z wbudowanego w program Visum algorytmu wyszukiwania ścieżki pomiędzy kolejnymi wprowadzanymi elementami trasy. Użytkownik posiada, co prawda możliwość wpływu na parametry tej procedury (maksymalny stosunek długości znalezionej ścieżki do odległości pomiędzy punktami, możliwość automatycznego otwierania relacji skrętnych), jednak wizualne sprawdzenie przebiegów tras jest konieczne, aby wyeliminować możliwe pomyłki. Podczas manualnego wprowadzania danych użytkownik na bieżąco weryfikuje poprawność wprowadzania tras. W przypadku skryptu należy bezwzględnie wykonać spraw-

dzenie po zakończonym procesie. Aby ułatwić weryfikację, zdarzenia takie jak niemożliwość odnalezienia ścieżki przez algorytm, brak na sieci jakiegoś elementu czy nieodpowiednie czasy przyjazdu/odjazdu notowane są w raporcie zapisywanym w postaci pliku tekstowego po zakończeniu działania skryptu.

2.5. Ocena wykorzystanego rozwiązania

Duży nakład pracy, jaki wiąże się z odpowiednim przygotowaniem danych oraz stworzeniem głównego algorytmu został zrekompensowany z nawiązką poprzez skrócenie czasu pracy przy wprowadzaniu linii a także niewielki czas potrzebny przy dokonywaniu poprawek i zmian na dalszych etapach projektu. Zastosowane rozwiązanie pozwoliło na zaoszczędzenie znacznej ilości czasu. Jakiekolwiek zmiany rozkładów jazdy (przyspieszanie/opóźnianie wybranych kursów) czy też zmiany przebiegów, związane z wariantowaniem łatwiej jest przeprowadzać na pliku źródłowym z danymi niż wewnątrz modelu w programie Visum.

Dodatkową korzyścią jest to, że raz stworzone narzędzie może być używane na kolejnych projektach a zdobyte doświadczenie w pozyskiwaniu i przygotowywaniu danych prawdopodobnie pozwoli na dalsze skrócenie czasu pracy.

Niewątpliwie nastąpi również postęp, jeśli chodzi o dostęp do danych zapisanych w formie elektronicznej, co pozwoli wyeliminować wiele nieudogodnień związanych ze wstępnym przygotowywaniem danych wejściowych.

3. Automatyzacja obliczeń

Po zakończonym procesie modelowania oraz stworzeniu serii scenariuszy, wykonano analizy efektywności ekonomicznej. Model ekonomiczny przygotowany został w postaci arkuszy programu Excel. Danymi niezbędnymi do wykonania analizy były wyniki uzyskane z prognoz otrzymanych przy wykorzystaniu modelu ruchu. Przy znacznej liczbie scenariuszy oraz możliwej dużej liczbie iteracji problemem stało się każdorazowe przenoszenie wyników obliczeń z modeli w programie Visum do modelu ekonomicznego. Danymi były listy obiektów programu Visum takie jak:

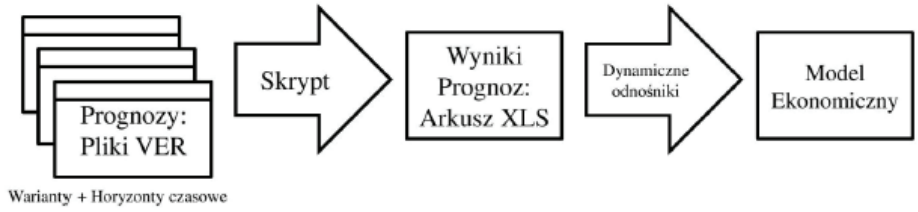
- Odcinki sieci drogowej oraz kolejowej wraz z atrybutami,
- Elementy tras transportu zbiorowego wraz z atrybutami z podziałem,
- Systemy transportowe wraz z atrybutami.

Przenoszenie powyższych danych pomiędzy programami Visum i Excel jest kłopotliwe i długotrwałe a czasem prawie niemożliwe na standardowej stacji roboczej przy bardzo dużej liczbie elementów (lista elementów tras transportu zbiorowego składała się na opisywanym projekcie z czterystu pięćdziesięciu tysięcy wierszy). W związku z powyższym zdecydowano się na wykorzystanie skryptów w języku Python do eksportowania wyników obliczeń.

3.1. Zasada działania skryptu

Pliki VER (pliki zawierające modele stworzone przy pomocy programu Visum) umieszczone w wybranym folderze po kolei są otwierane przez skrypt, listy odpowiednich elementów wraz z wybranymi atrybutami zostają utworzone a następnie zapisane, jako arkusze w wyjściowym pliku Excel. Następnie arkusz zawierający model ekonomiczny pobiera odpowiednio sformatowane dane z zapisanego pliku za pomocą dynamicznych odnośników. Cały proces zapisywania wyników ze wszystkich scenariuszy we wszystkich horyzontach czasowych oraz aktualizacja danych w modelu ekonomicznym trwa około 20-30 minut pracy maszyny.

Schemat procesu przedstawiono na rys. 3.



Rys. 3. Schemat procesu transferu danych z modeli podróży do modelu ekonomicznego
Źródło: opracowanie własne

3.2. Ocena wykorzystanego rozwiązania

Wybrane rozwiązanie pozwoliło przede wszystkim na zaoszczędzenie znacznej ilości czasu, co przy wielokrotnie zmieniających się założeniach dla poszczególnych scenariuszy uzasadniło włożenie dużego nakładu pracy na początku projektu w przygotowanie skryptów oraz modyfikację arkuszy Excel z modelami ekonomicznymi. Dodatkową korzyścią było również to, że po początkowym okresie testowania i usunięciu wszelkich niedociągnięć można było mieć pewność, że operacja jest wykonywana nie tylko błyskawicznie, ale również bezbłędnie, czego nie można zazwyczaj powiedzieć o żmudnych czynnościach wykonywanych za każdym razem ręcznie.

4. Podsumowanie

Oba opisane przypadki użycia skryptów do usprawniania pracy za pomocą programu Visum charakteryzują się tym, że konieczny jest duży początkowy nakład pracy a także bardzo staranne przygotowanie danych wejściowych (lub uzyskanie dobrych danych wprost ze źródła). Inwestycja ta jednak jest rekompensowana w późniejszych fazach projektu, kiedy to wiele czynności można wykonać automatycznie. Pozwala to zminimalizować poświęcany czas a także wyeliminować

pomyłki. Zysk z wprowadzania takich rozwiązań uwidacznia się zwłaszcza w przypadkach, kiedy wielokrotnie zmieniają się założenia bądź wprowadzane są do projektu zmiany i poprawki.

Ciągle powiększająca się ilość oraz szczegółowość danych wejściowych w połączeniu z coraz większymi możliwościami sprzętowymi powoduje, że modele ruchu stają się bardziej rozbudowane. Często ilość danych, jakie trzeba wprowadzić do modeli jest tak duża, że zupełnie nieefektywne jest wykonywanie takiej pracy manualnie. Według autora jest to tendencja nieodwracalna, dlatego też automatyzacja wielokrotnie powtarzanych żmudnych czynności stanie się nie tylko sposobem na zmniejszenie nakładu pracy przy projektach, ale w wielu przypadkach wręcz koniecznością.

Bibliografia

[1] Kucharski R., Otwieranie oprogramowania – skrypty i programy w programie Visum. Ogólnopolska Konferencja Naukowo-Techniczna „Modelowanie podróży i prognozowanie ruchu", Kraków, 2010.
[2] General Transit Feed Specification Reference [online], [dostęp 19 lutego 2014], dostępny w Internecie: https://developers.google.com/transit/gtfs/reference.
[3] Google Transit Cities Covered [online], [dostęp 20 stycznia 2014], dostępny w Internecie: http://www.google.com/landing/transit/cities/.
[4] GoogleTransitDataFeed project summary [online], [dostęp 19 lutego 2014], dostępny w Internecie: https://code.google.com/p/googletransitdatafeed/.
[5] TransXChange overview [online], [dostęp 19 lut 2014], dotępny w Internecie: https://www.gov.uk/government/publications/transxchange-overview.