POLSKIE TOWARZYSTWO INFORMACJI PRZESTRZENNEJ
ROCZNIKI GEOMATYKI 2011 ❍ Tom IX ❍ Zeszyt 3(47)

WYKORZYSTANIE OPROGRAMOWANIA OPEN SOURCE W POMIARACH BEZPOŚREDNICH NA PRZYKŁADZIE SYSTEMU QUANTUM GIS

THE USE OF OPEN SOURCE SOFTWARE IN SURVEYING ON THE EXAMPLE OF QUANTUM GIS

Michał Bednarczyk[1], Jacek Rapiński[2]

[1] Katedra Geodezji Szczegółowej, [2] Instytut Geodezji
Wydział Geodezji i Gospodarki Przestrzennej
Uniwersytet Warmińsko Mazurski w Olsztynie

Słowa kluczowe: Open source, GIS, Python, Quantum GIS, Linux, SynCE, pomiary bezpośrednie, programowanie
Keywords: Open source, GIS, Python, Quantum GIS, Linux, SynCE, surveying, programming

Wstęp

Szeroko rozumiane oprogramowanie open source, tak jak cała scena informatyczna, podlega ciągłemu rozwojowi. Jego historia rozpoczęła się od zastosowań ściśle informatycznych, takich jak systemy operacyjne serwerów i zarządzanie siecią komputerową. Obecnie zyskuje ono coraz większą popularność, powstaje coraz większa liczba programów, zwłaszcza w zastosowaniach takich jak aplikacje biurowe, programy graficzne czy multimedialne, to samo dotyczy systemów informacji przestrzennej. Taka sytuacja skłania do poszukiwań nowych rozwiązań i stawiania coraz trudniejszych zadań oprogramowaniu open source. Jego niezaprzeczalną zaletą jest dostępność bez opłat, co niejednokrotnie znacznie obniża koszty pozyskania oprogramowania. Należy oczywiście podchodzić do tej sprawy ostrożnie, gdyż brak opłat oznacza jednocześnie brak gwarancji działania i brak wsparcia technicznego.

Autorzy artykułu postanowili bliżej przyjrzeć się możliwościom open source pod kątem zastosowania w opracowaniu pomiarów bezpośrednich oraz przybliżyć nieco samą ideę wolnego oprogramowania, zasad jego licencjonowania i dystrybucji. Czytelnik znajdzie tu zatem informacje na temat pozyskania danych z odbiornika GPS oraz opracowania mapy z wykorzystaniem programów dostępnych bez opłat, które pracują pod kontrolą systemu Linux, będącego również systemem bezpłatnym. Autorzy wskazują również na duże możliwości systemu Quantum GIS, dającego możliwość rozszerzenia swojej funkcjonalności przez programowanie, co zostało zaprezentowane na przykładzie skryptu w języku Python, opracowanego specjalnie do realizacji celu niniejszego artykułu. Na początek jednak kilka zdań na temat idei open source.

Idea open source

W szeroko publikowanych materiałach źródłowych możemy natknąć się na dwa terminy: „wolne oprogramowanie" (ang. *free software*) oraz „otwarte oprogramowanie" (ang. *open source*). Niektórzy autorzy stosują je w swoich opracowaniach zamiennie dla określenia tej samej grupy oprogramowania. Inni wprowadzają rozgraniczenie polegające przede wszystkim na podejściu teoretycznym czy wręcz ideologicznym do omawianego zagadnienia.

W dokumentach urzędowych czy oficjalnych analizach można spotkać skrót FLOSS (ang. *Free Libre/Open Source Software*) obejmujący jednym mianem zarówno *free software* jak i *open source*.

Free software oraz związany z nim ruch *Free Software Movement* to pojęcie wprowadzone przez Richarda Stallmana w połowie lat 80. XX wieku (FSF, 2007). Jego intencją było stworzenie oprogramowania dostępnego dla każdego bez opłat, z możliwością dowolnej, darmowej dystrybucji i modyfikacji w taki sposób, aby rozprowadzanie kolejnych wersji zmienionych przez innych programistów odbywało się również za darmo. Trzymając się tej zasady Stallman rozpoczął tworzenie unixopodobnego systemu o nazwie GNU *(GNU's not UNIX)*, którego składniki są obecnie rozpowszechniane wraz z jądrem systemu Linux jako GNU/Linux w postaci tzw. dystrybucji (np. Slackware, Debian, RedHat i in.).

Stallman kładzie tu duży nacisk na znaczenie słowa *free* – jako „wolny" a nie darmowy. Według niego bezpłatne uzyskanie programu nie daje prawdziwej wolności. Dopiero ścisłe zastosowanie się do określonych przez niego zasad może ją zagwarantować. Zasady te dotyczą wolności w zakresie użytkowania programu, analizy jego działania (dostęp do kodu źródłowego), redystrybucji oprogramowania oraz dokonywania zmian w jego konstrukcji.

Free software było popularne w środowiskach akademickich oraz w pewnym gronie sympatyków. Pojawienie się powszechnego dostępu do Internetu spowodowało wzrost zainteresowania oraz przyspieszyło jego rozwój, głównie ze względu na nowe możliwości współpracy i kontaktu między programistami. Masowa wymiana informacji przyczyniła się również do tego, że zaczęto dostrzegać nieco odmienne możliwości tkwiące w tym sposobie tworzenia aplikacji. W czasach gwałtownego rozwoju Internetu pod koniec lat 90., po spektakularnym sukcesie Linuxa w 1998 roku, pojawiło się nowe określenie: open source. Stworzyli je informatycy, m.in. John Hall, Larry Augustin, Eric Steven Raymond, Bruce Perens i inni (OSI, 2006) aby nieco odróżnić się od free software. W tym samym roku E.S.Raymond i B.Perens zapoczątkowali *Open Source Initiative (OSI)* (FSF, 2007), która jest organizacją non-profit, mającą na celu wspieranie i propagowanie idei open source w świecie. Podstawowe założenia open source pozostały w zasadzie takie same, niewiele odbiegając od propozycji Stallmana. Kierowały nimi głównie pobudki praktyczne. Ich pomysł był bardziej skierowany na zyskanie przychylności większej części użytkowników, w tym dużych producentów oprogramowania. Postanowiono zwrócić uwagę na techniczną stronę tego typu produktów, ich potencjalnie wyższą jakość, możliwości finansowania rozwoju, sposoby wdrażania i eksploatacji oraz związane z tym aspekty ekonomiczne. Przytaczane przez nich argumenty oraz wizja zastosowania open source w przemyśle czy administracji zdecydowanie bardziej trafiły do dużych firm i instytucji, zwiększając ich zainteresowanie zagadnieniem.

Podstawową przyczyną powodzenia była pewna – niewielka acz istotna – zmiana w podejściu do wytwarzania produktu informatycznego. E.S. Raymond (1998) analizuje tę sytuację w swoim artykule „The Cathedral and the Bazaar" na przykładzie pierwszego czło-

wieka, który odniósł duży sukces na tym polu – Linusa Torvalds'a – twórcy Linuxa. Analizowany przykład porównuje do swoich doświadczeń związanych z tworzonym przez niego programem pocztowym, udostępnionym na tych samych zasadach co Linux. Człowiekiem, którego działania były inspiracją dla L.Torvalds'a był R.Stallman, który jako pierwszy podjął próbę współpracy przez Internet na szerszą skalę, m.in. przy tworzeniu kolejnej wersji edytora Emacs. Jednak przykład kompletnego i złożonego systemu operacyjnego jakim jest Linux pokazuje prawdziwą skalę możliwości podejścia open source do tworzenia oprogramowania. Według Raymonda do sukcesu Linuxa przyczyniły się, przede wszystkim, następujące czynniki:

- otwarcie źródeł i dostęp do kodu na dużą skalę dzięki Internetowi;
- traktowanie użytkowników w sposób szczególny – każdy, kto modyfikuje kod lub zgłasza poprawki staje się współautorem oprogramowania i stara się jak najlepiej wykonać swoją pracę;
- wczesna i częsta publikacja kodu źródłowego – takie działanie pozwala na przerzucenie bardzo ważnego etapu tworzenia oprogramowania, jakim w inżynierii oprogramowania jest testowanie, na wielu użytkowników. Testowanie i poprawki w przypadku złożonego projektu zajmują najwięcej czasu. W firmach komercyjnych etap ten jest często skracany do niezbędnego minimum. W efekcie może powstać produkt niedopracowany czy wręcz wadliwy. Dokonywanie poprawek po wdrożeniu jest bardzo uciążliwe – zwłaszcza dla użytkowników. Osiągnięcie wymaganego poziomu funkcjonalności, jakości i bezpieczeństwa jest bardzo trudne. Open source daje możliwość przejścia ścieżki publikacja-testy-poprawki w czasie bardzo krótkim. W przypadku Linuxa, w fazie najintensywniejszego rozwoju, współpracowało wspólnie tak wielu ludzi, że możliwa była publikacja kolejnych wersji jądra systemu raz dziennie. Ta sama ścieżka w komercyjnej firmie, w której pracuje zespół kilku – kilkudziesięciu testerów, zajęłaby miesiące. Ma to ogromny wpływ na jakość powstającego produktu.

Raymond wskazuje właśnie na szeroko zakrojone testowanie jako kluczowy czynnik leżący u podstaw powodzenia Linuxa. Torvalds rozwijając jądro systemu Linux nie wymyślił w zasadzie niczego nowego. Potrafił natomiast umiejętnie wykorzystać siłę jaką daje Internet oraz odpowiednio kierować projektem i współpracować z ludźmi. Wkrótce tą drogą poszło wielu innych, co zaowocowało popularnymi programami, jak np. przeglądarka internetowa Netscape (potem również Mozilla czy Firefox), pakiet biurowy Open Office, różne serwery, miedzy innymi Apache, MySQL i inne.

Zastosowanie opisywanego oprogramowania open source w praktyce zależy od indywidualnych potrzeb i możliwości potencjalnych użytkowników. Powinno zostać poprzedzone analizą wymagań konkretnego odbiorcy. Panuje przekonanie, że darmowe rozwiązania są funkcjonalnie uboższe od komercyjnych, co w wielu przypadkach jest uzasadnione. Jednak opisywane w niniejszej pracy systemy mają na tyle szeroki wachlarz funkcji, że mogą sprostać różnorakim wymaganiom. Posiadają przede wszystkim najbardziej przydatne opcje. Komercyjne systemy często mają dużo większe możliwości, z których w praktyce nie wszystkie zostają wykorzystane.

Każdy, kto zdecyduje się na open source musi liczyć się z trudniejszą obsługą oraz koniecznością (choć coraz rzadziej) samodzielnego zdobywania wiedzy na temat instalacji, konfiguracji i obsługi tego oprogramowania. Większe projekty posiadają stałe wsparcie techniczne, dostępne są szkolenia i konsultacje tak jak przy produktach komercyjnych – często jednak nie są to usługi darmowe.

Stabilność pracy i bezpieczeństwo, w przypadku wielu produktów open source oceniane są przez użytkowników dość wysoko. Niestabilność pracy i błędy częściej zdarzają się w przypadku projektów nowych, dopiero rozwijających się oraz takich, które nie zyskały większej popularności. Stosowanie idei open source w cyklu produkcji oprogramowania okazało się dobrym pomysłem. Jego skuteczność tkwi w liczbie programistów/testerów oraz otwartości kodu źródłowego. Warunkiem powodzenia jest umiejętne zarządzanie projektami, promowanie go wśród potencjalnych twórców i użytkowników oraz ewentualne zapewnienie wsparcia technicznego. W efekcie powstają produkty, które mogą rywalizować z komercyjnymi na rynku oprogramowania.

Licencjonowanie oprogramowania open source

Angielskie określenie open source można by w sposób bezpośredni przetłumaczyć jako „otwarte źródło". Odnosi się ono do programu komputerowego, którego kod źródłowy jest dostępny (otwarty) dla każdego użytkownika. Według definicji *(OSD – Open Source Definition)* opublikowanej na stronie www.opensource.org przez *Open Source Initiative* (OSI), termin ten określa nie tylko dostępność do źródła aplikacji. Określa także warunki używania i dystrybucji tego typu oprogramowania, jakie powinny być zdefiniowane w licencji. Opracowano je na podstawie wytycznych Debiana (2004) dotyczących Wolnego Oprogramowania. W oparciu o nie autorzy oprogramowania tworzą własne wersje licencji, wśród których można wymienić:

- GNU General Public License (GPL),
- Lesser General Public License – GPL dla bibliotek,
- IBM Public Linense,
- Intel Open Source License,
- Mozilla Public Licence i inne.

Pełna lista licencji zaakceptowanych przez OSI dostępna jest na stronie internetowej tej organizacji (http://www.opensource.org).

Najbardziej popularną spośród wymienionych jest licencja GPL opracowana przez Free Software Foundation. Zawiera ona bardzo rygorystyczną klauzulę „copyleft", która zobowiązuje autorów do rozpowszechniania ich programów na licencji GNU GPL, jeżeli są one w określonym związku z programem już objętym tą licencją, np. gdy zawierają część lub całość tego programu (Siewicz, 2004). Zapis ten niekiedy określany jest mianem „wirusa copyleft", jednak nie ma na celu zastawienia pułapki na licencjobiorcę. Klauzula „copyleft" ma chronić interesy osób zaangażowanych w tworzenie wolnego oprogramowania oraz uniemożliwić monopolizowanie efektów tej pracy przez innych. Jednakże wielu producentów decydujących się na model open source, redagując licencje, nie korzysta z klauzul „copyleft" lub proponuje je w znacznie złagodzonej formie. Warto w tym miejscu wspomnieć, że korzystanie we własnym zakresie z programów połączonych z wolnym oprogramowaniem jest zgodne z GPL. Obowiązek udostępniania kodu powstaje dopiero w momencie rozpowszechniania programu przez licencjobiorcę.

Program Quantum GIS

Domena zastosowań aplikacji open source jest ogromna. Są wśród nich zarówno aplikacje biurowe, systemy operacyjne, serwery czy systemy CAD i GIS. W przypadku zastosowań specjalistycznych, takich jak geodezja, nie istnieje jeden system obejmujący większość zadań. Poszukując narzędzi geodezyjnych znajdziemy zazwyczaj aplikacje ukierunkowane głównie na GIS i zastosowania pokrewne, w tym serwery danych oraz proste bądź bardziej rozbudowane programy narzędziowe.

Systemem, który zasługuje na szczególną uwagę jest Quantum GIS (QGIS). Zyskał on dość dużą popularność, głównie ze względu na coraz większą liczbę nowych, ciekawych funkcji oraz wiele możliwości wykorzystania. Udostępniany jest bezpłatnie na licencji GNU/GPL (Nowotarska, 2009). Dostępny jest w wersjach dla wielu systemów operacyjnych między innymi: Linux, Mac OSX, Windows, Unix. Obsługuje wiele znanych formatów wektorowych i rastrowych, w tym: SHP, DXF, GML, MIF, TAB, DGN, TIFF, JPG. Potrafi łączyć się z bazami danych takimi jak: MySQL, Postgre, SqlLite, Esri Personal Geodatabase. Obsługuje również źródła danych WMS *(Web Map Service)* i WFS *(Web Feature Service)*. QGIS został wyposażony w wiele przydatnych funkcji do edycji grafiki wektorowej i rastrowej oraz wykonywania analiz. Z pewnością nie jest to system wszechstronny i niezawodny, ale został tak napisany, aby realizować większość najczęściej spotykanych zadań stawianych tego rodzaju oprogramowaniu. Podstawowa wersja QGIS nie posiada zbyt wielu funkcji. Dopiero instalacja dodatkowych rozszerzeń w postaci wtyczek ujawnia pełne możliwości systemu. Zestaw gotowych do podłączenia wtyczek uzyskuje się już w momencie pobrania pakietu QGIS z witryny projektu. Dodatkowe wtyczki można pobrać i zainstalować samodzielnie, bądź stworzyć własne.

QGIS został napisany w języku C++ (Hugentobler, Duster, Sutton, 2008) Udostępniono w nim możliwość tworzenia własnych rozszerzeń w dwóch językach programowania: C++ oraz Python. Interfejs programowania rozszerzeń do QGIS w języku Python nosi nazwę PyQGIS. Interfejsu PyQGIS można używać na trzy sposoby:

- poprzez konsolę Python'a dostępną w QGIS z poziomu interfejsu graficznego – sposób ten jest szczególnie przydatny do testowania kodu;
- do tworzenia wtyczek (ang. *plugins*) napisanych w języku Python – wtyczki tak napisane mogą być podłączone do interfejsu graficznego i używane dokładnie tak samo jak wtyczki napisane w C++;
- do tworzenia samodzielnych aplikacji napisanych w języku Python z własnym interfejsem, które mogą wykorzystywać funkcjonalność bibliotek systemu QGIS.

Python jest językiem skryptowym. Podstawowym założeniem jego twórców była łatwość programowania. Składnia kodu została uproszczona do minimum, natomiast styl programowania narzucany przez interpreter Pythona zwiększa czytelność algorytmów (m.in. przez obowiązek stosowania wcięć dla podrzędnych fragmentów programu). Poza tym, Python posiada mechanizm automatycznego uwalniania pamięci, która nie jest już używana, co znacznie upraszcza zarządzanie pamięcią komputera w trakcie działania programu.

Wtyczki oparte na interfejsie PyQGIS wykorzystują funkcjonalność bibliotek *libqgis_core.so* i *libqgis_gui.so*, które są licencjonowane na zasadach GPL. Oznacza to, że każda wtyczka QGIS napisana w języku Python również podlega tej licencji. Jeżeli autor wtyczki będzie wykorzystywał ją do własnych celów, nie ma obowiązku udostępniania jej innym użytkownikom. W przypadku gdy zdecyduje się na udostępnianie, musi tego dokonać na zasadach GPL, czyli rozpowszechnić swoją wtyczkę jako oprogramowanie open source.

Stworzenie mapy na podstawie danych z pomiarów

Współczesna aplikacja geodezyjna, służąca do opracowywania pomiarów bezpośrednich charakteryzuje się najczęściej następującymi cechami:

- wczytywanie danych z plików zewnętrznych lub bezpośrednio z rejestratora,
- wykonanie wszelkich niezbędnych obliczeń,
- tworzenie zbiorów kartograficznych w powiązaniu z bazą danych,
- kartowanie i uzupełnienie atrybutów pomierzonych szczegółów terenowych,
- stworzenie dokumentów niezbędnych do operatu (raporty obliczeń, dzienniki pomiarowe)
- wydruk mapy,
- inne możliwości, charakterystyczne dla konkretnego oprogramowania.

Niektóre programy (np. Cgeo) integrują wszystkie te cechy w jednej aplikacji, w innym przypadku mogą zostać rozdzielone na program obliczeniowy i edytor graficzny (np. Winkalk i Mikromap).

Niektóre z wymienionych cech pokrywają się z funkcjonalnością QGIS. Biorąc pod uwagę możliwość zwiększenia jego funkcjonalności przez tworzenie własnych rozszerzeń, autorzy postanowili podjąć próbę opracowania zbioru danych z pomiarów bezpośrednich z wykorzystaniem QGIS oraz innych dostępnych narzędzi open source. Celem takiego działania jest sprawdzenie, czy i w jakim stopniu aplikacje open source, w tym zwłaszcza QGIS, pozwolą na opracowanie danych pochodzących z pomiaru.

Pozyskanie danych z instrumentów pomiarowych

Większość współczesnych narzędzi pomiarowych pracuje pod kontrolą systemu WindowsCE w różnych wersjach. W przypadku odbiorników GNSS system ten jest instalowany na kontrolerze, w przypadku tachimetrów bezpośrednio w instrumencie. Z większości instrumentów pomiarowych pozyskiwane są docelowe współrzędne, a nie surowe obserwacje. W celu przeniesienia danych na komputer i dalszego ich opracowania, w większości przypadków można wykorzystać aplikację ActiveSync dla WindowsXP/Vista. Niestety producent nie udostępnia oprogramowania ActiveSync dla innych systemów operacyjnych.

W przypadku Systemów z rodziny Linuxa, do komunikacji pomiędzy komputerem a kontrolerem możena skorzystać z oprogramowania rozwijanego w ramach projektu SynCE. Oprogramowanie jest dostępne na stronach SourceForge.net oraz w wielu repozytoriach. Wiele szczegółowych informacji na temat instalacji można znaleźć w książce Edwarda L. Haletky "Deploying Linux on the Desktop" (Edwarda, Haletky, 2005). Dane do niniejszego opracowania zostały pozyskane z wykorzystaniem odbiornika GPS/RTK Leica Viva. W tego rodzaju pomiarze uzyskuje się współrzędne poszczególnych punktów, które następnie muszą zostać przetransmitowane do komputera w celu dalszej obróbki. Podstawowym zadaniem do zrealizowania było zatem wykorzystanie oprogramowania open source do odczytu danych z odbiornika oraz opracowania fragmentu mapy zasadniczej.

Instalacja pakietu SynCE nie jest kłopotliwa – w wielu dystrybucjach pakiet dostępny jest w postaci binarnej (np. w repozytoriach Ubuntu lub jako RPM dla Fedory). Przy pracy z najnowszą wersją programu należy pobrać źródła i samemu je skompilować. Ponieważ w większości przypadków połączenie będzie realizowane przez port USB należy dołączyć do jądra moduł iPaq:

$ /sbin/modprobe ipaq

Po instalacji należy poinformować program do jakiego portu podłączone jest urządzenie z systemem WindowsCE (w tym przypadku port USB 1):

$ synce-serial-config ttyUSB1

Ostatnim krokiem jest uruchomienie demona, który będzie czekał na połączenie z kontrolerem.

 # synce-serial-start

W ramach pakietu dostępne są między innymi następujące narzędzia uruchamiane z linii poleceń:

- synce-pstatus – pokazuje status połączenia,
- synce-pls – wyświetla listę plików w urządzeniu,
- synce-pmkdir – tworzy katalog,
- synce-pcp – pozwala na kopiowanie plików pomiędzy urządzeniem a komputerem,
- synce-install-cab – instaluje oprogramowanie z pliku .cab.

W przypadku pobierania pliku z danymi pomiarowymi z kontrolera do komputera należy użyć komendy w postaci:

$ synce-pcp „:/My Documents/Geodezja/obiekt1/punkty.txt" /home/user/Dokumenty/

SynCE oferuje również dodatkowe narzędzia np. do synchronizacji książki adresowej czy poczty. Dostępne są również wczesne wersje rozwojowe interfejsów graficznych zarówno dla KDE jak i Gnome.

Wykorzystanie skryptów Pythona w QGIS umożliwia stworzenie rozszerzenia importującego dane za pomocą powyższych narzędzi z poziomu programu QGIS w sposób zautomatyzowany.

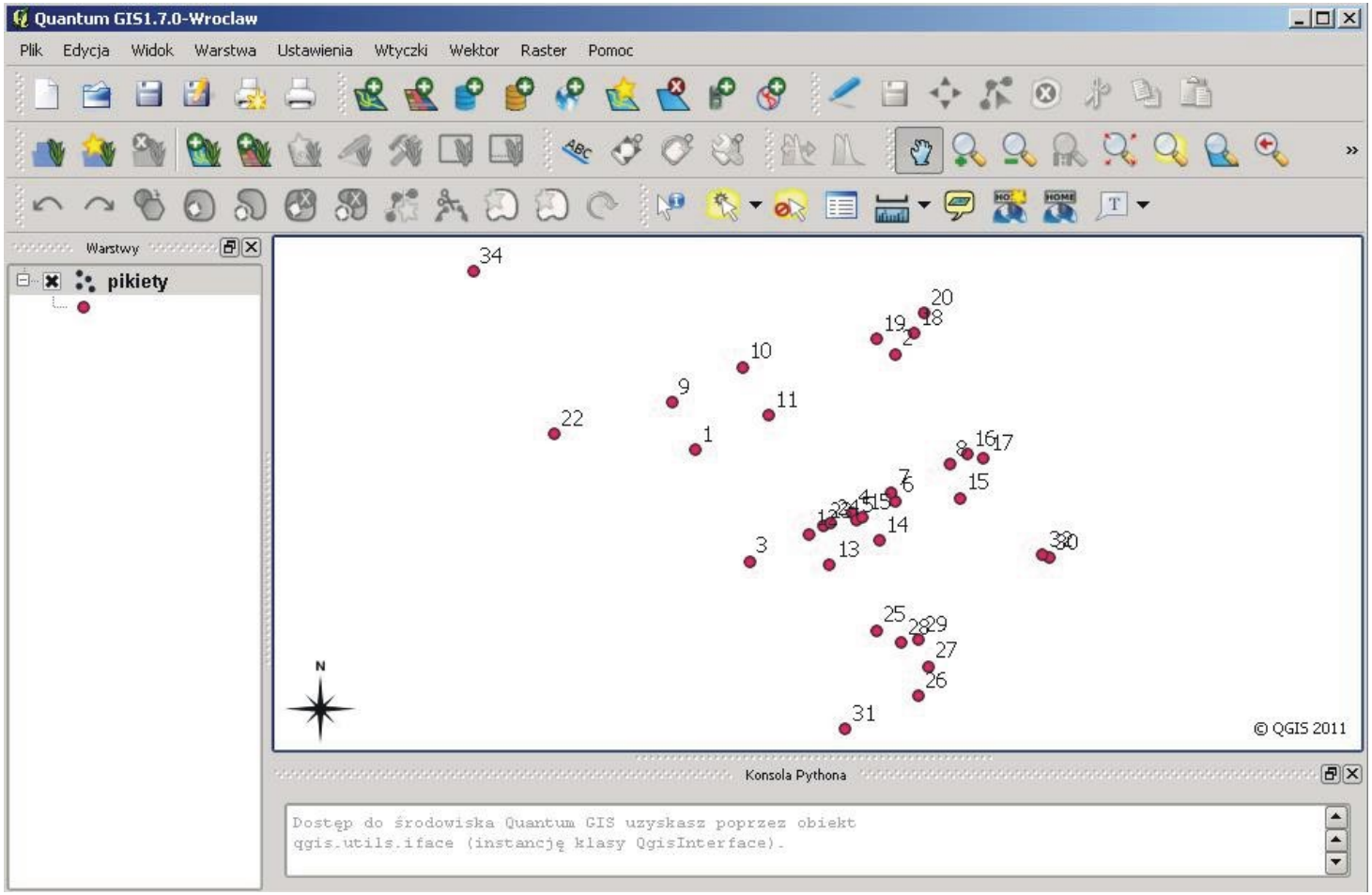
Kartowanie obiektów w Quantum GIS

Z uwagi na swoją obecną funkcjonalność – patrząc pod kątem zastosowania w pomiarach bezpośrednich – QGIS najbardziej nadaje się do kartowania obiektów i stworzenia bazy atrybutów opisowych. Niemniej jest on bardzo ukierunkowany na zadania z zakresu GIS, głównie analizy przestrzenne i atrybutowe czy edycję bazy danych. Edycja grafiki wektorowej jest tu ograniczona do niezbędnego minimum. W pomiarach bezpośrednich metodą biegunową bądź RTK powstają zbiory ponumerowanych punktów, powiązanych z sytuacją przedstawioną na szkicach polowych. Najprostszym i najbardziej skutecznym sposobem kartowania szczegółów jest wprowadzanie obiektów punkt po punkcie, zgodnie z planem połączeń, na podstawie ich numerów lub kodów. W QGIS nie znajdziemy takiej opcji, dlatego autorzy niniejszego artykułu postanowili stworzyć własną funkcję, która umożliwi realizację tego zadania.

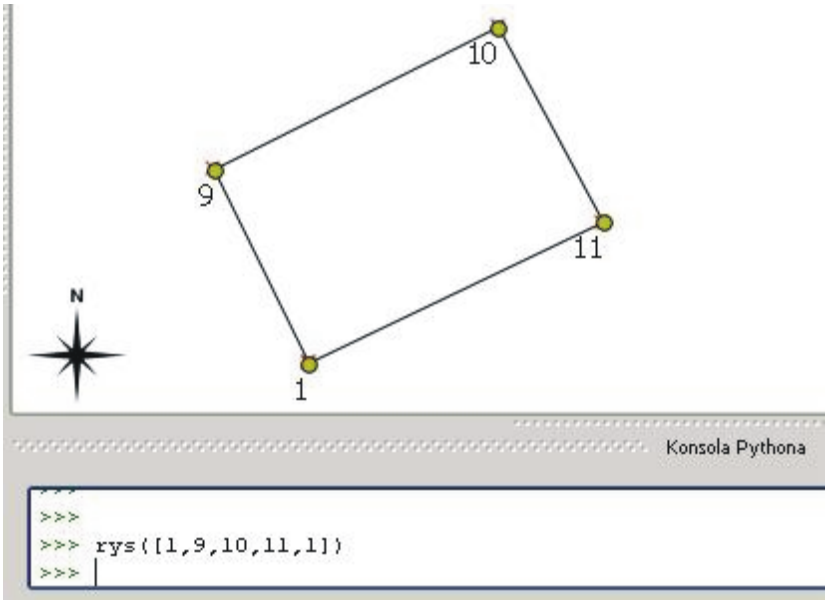
Obliczone współrzędne pikiet zostały wczytane do QGIS z pliku tekstowego na warstwę „pikiety" za pomocą wtyczki: „Dodaj warstwę tekstową rozdzieloną separatorami" (rys. 1).

Następnie stworzono poszczególne warstwy, określono ich styl graficzny oraz geometrię i atrybuty opisowe. Kolejny krok to kartowanie szczegółów na poszczególnych warstwach. Tu posłużono się specjalnie stworzoną do tego celu funkcją" rys()" ułatwiającą kartowanie (rys. 2). Została stworzona jako skrypt w języku Python i w tej formie może być przekształcona na wtyczkę ładowaną do interfejsu QGIS. W obecnej formie funkcję tę wywołuje się z poziomu konsoli Pythona w sposób następujący:

rys([lista pikiet])



Rys. 1. Wczytane pikiety



Rys. 2. Kartowanie szczegółu za pomocą funkcji „rys()”

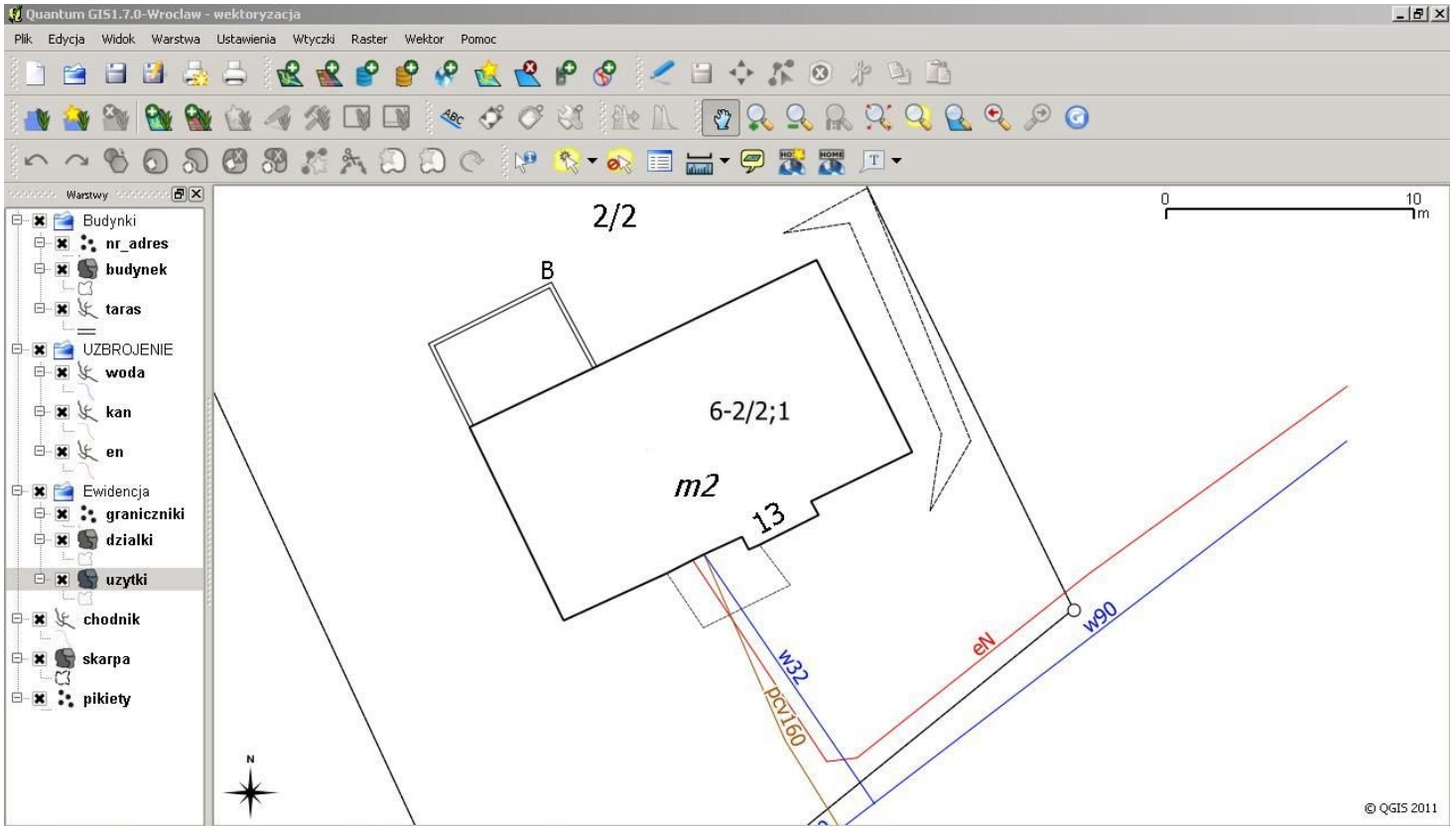
Gdzie [lista pikiet] to lista numerów pikiet oddzielonych przecinkami. Numery te są zapisane w bazie danych w polu o nazwie „nr” i zostały nadane w trakcie pomiaru i zaimportowane wraz z warstwą „pikiety”. Po wywołaniu funkcji, podaniu argumentów i wykonaniu, na mapie zostanie narysowany żądany obiekt. W zależności od wybranej warstwy w legendzie będzie to punkt, linia lub wielolinia narysowana stylem przypisanym do warstwy.

Postępując w wyżej opisany sposób opracowano kompletną mapę, której fragment przedstawiono na rysunku 3. Większość funkcji jakie są potrzebne do opracowania mapy istnieje już w QGIS. Można definiować własne style linii i symboli. Problemem jest tworzenie symboli złożonych, takich jak na przykład skarpa czy schody. Tu również należałoby posłużyć się odpowiednio skonstruowanym skryptem bądź wtyczką. Mimo wszystko, ten niewielki przykład pokazuje, że rozbudowanie funkcjonalności QGIS do poziomu niezbędnego do realizacji bardziej złożonych zadań z dziedziny geodezji jest możliwe.

Podsumowanie i wnioski

Oprogramowanie open source stale się rozwija, zyskując coraz nowsze i ciekawsze produkty. Istniejące od lat i znane projekty, takie jak QGIS, również stają się coraz lepsze i bardziej funkcjonalne. W artykule opisano próbę wykonania pewnego zadania z zakresu geodezji jakim jest pomiar szczegółów i skartowanie wyników. Udało się to w pełni z wykorzystaniem oprogramowania open source, były to w tym przypadku: Linux, SynCE i Quantum GIS, który zasługuje na szczególną uwagę.

W kontekście podjętego zadania QGIS najbardziej nadaje się do kartowania szczegółów oraz prowadzenia bazy danych atrybutów opisowych poszczególnych obiektów. W chwili



Rys. 3. Gotowa mapa opracowana w QGIS

obecnej jego funkcjonalność nie pozwala na wykonanie wszystkich zazwyczaj realizowanych zadań z zakresu opracowania pomiarów bezpośrednich (jak np. pobranie danych z instrumentu, stworzenie raportu z pomiaru czy obliczeń). Część z nich należy wykonać w innym oprogramowaniu, używając QGIS do opracowania kartograficznego. Nie przekreśla to jednak możliwości wykorzystywania QGIS w opracowywaniu zbiorów danych z pomiarów sytuacyjnych w pełniejszym wymiarze. Ze względu na możliwość tworzenia własnych rozszerzeń oraz istniejącą już funkcjonalność z zakresu GIS (zwłaszcza interfejs graficzny i współpraca z bazą danych), QGIS doskonale nadaje się jako podstawa stworzenia w pełni funkcjonalnego programu open source do opracowywania danych z pomiarów bezpośrednich. Zostało to zaprezentowane na przykładzie samodzielnie opracowanej funkcji „rys()" wspomagającej kartowanie. Powstanie takiego rozwiązania na licencji open source, o funkcjonalności niezbędnej do kompleksowej realizacji zadań z zakresu opracowań geodezyjnych (w tym m.in. pobranie danych, obliczenia, kartowanie i tworzenie mapy numerycznej) mogłoby przyczynić się do obniżenia kosztów oprogramowania w tej dziedzinie, choć i w chwili obecnej prezentowane w artykule oprogramowanie spełnia swoją rolę i może być przydatne.

Literatura

FSF, 2007: About the GNU Project. Free Software Foundation, http://www.gnu.org/gnu/

Debian, 2004: Debian Social Contract, http://www.debian.org/social_contract.html#guidelines

Edwarda L., Haletky, 2005: Deploying Linux on the Desktop. Digital Press.

OSI, 2006: History of the OSI. Open Source Initiative, http://www.opensource.org/history

Hugentobler M., Duster H., Sutton T., 2008: Extending the Functionality of QGIS with Python Plugins, http://qgis.osgeo.org/qgiswiki/index.php?title=File:Python_workshop_en.pdf

Nowotarska M., 2009: Wprowadzenie do Quantum GIS, http://quantum-gis.pl/_media/czytelnia/wprowadzenie_do_quantum_gis.pdf

Raymond E.S., 1998: The Cathedral and the Bazaar. First Monday vol.3 no.3 March 1998, http://www.firstmonday.org/issues/issue3_3/raymond/index.html

Siewicz K., 2004: Prawna ochrona oprogramowania Open Source. http://7thguard.net/files/Siewicz_PrawoOSS.pdf

Abstract

Users of computer software, regardless of the field of applications, often look for solutions that both meet their expectations and are inexpensive. Computer market offers many products with various functionalities. Among these there are open source applications and systems, usually available without charge. Among open source applications for geodesy there are mostly GIS systems and tools associated with GIS. The functionality of some of them, however, is broad enough to look for other uses for these programs. The authors of this paper decided to take a closer look at the popular Quantum GIS and see if it is possible to use it in surveying, especially for map editing. The main reason, which tends to take up this issue is the possibility of programming in Quantum GIS using Python or C languages and creating user applications that help at work. Thanks to that QGIS may be used for other tasks than processing GIS data.

dr inż. Michał Bednarczyk
w.m.bednarczyk@wp.pl

dr inż. Jacek Rapiński
jacek.rapinski@gmail.com