TORUŃSKIE STUDIA BIBLIOLOGICZNE
2012, nr 2 (9)

Mariusz Jarocki
Instytut Informacji Naukowej i Bibliologii
Uniwersytet Mikołaja Kopernika w Toruniu
e-mail: maryan@umk.pl

Zastosowania skryptowych języków programowania w działalności informacyjnej

Streszczenie: Niezaprzeczalnym faktem ostatnich kilku lat jest gwałtowny przyrost publikowanych dokumentów w formie elektronicznej oraz rozwój Internetu. Ze względu na łatwy wgląd do utrwalonej w ten sposób informacji zwłaszcza ten ostatni zyskuje coraz liczniejsze grono zwolenników. Udostępniane w sieci dokumenty z technologicznego punktu widzenia są jednak niezwykle różnorodne: począwszy od prostej strony WWW, przez popularne pliki PDF, a na nagraniach audio i wideo skończywszy. Opracowanie precyzyjnej informacji charakteryzującej tak różne typy dokumentów jest zajęciem czasochłonnym i problematycznym. Obecny przyrost ilościowy publikowanych treści wyklucza także coraz częściej ręczny opis i przetwarzanie tego typu danych. Wśród dynamicznie zmieniających się struktur są to tylko niektóre problemy zajmującej się tym dziedziny wiedzy – informacji naukowej. Rozwiązania pomocnicze w tych kwestiach zazwyczaj są dostarczane przez inne dziedziny nauki, w tym przypadku informatykę. W większości do analizy i przetwarzania dokumentów wystarcza ogólnodostępne oprogramowanie, niekiedy jednak program temu służący trzeba stworzyć samodzielnie. Najprzystępniejsze w tym ostatnim przypadku wydaje się skorzystanie ze skryptowego języka programowania, o czym traktuje niniejszy artykuł.

Słowa kluczowe: biblioteki programistyczne Pythona, informacja naukowa, języki programowania, przetwarzanie dokumentów, Python, skrypty

Wprowadzenie

Początki automatyzacji procesów informacyjnych są ze swej definicji, tak jak każda inna dziedzina, związane z pojawieniem się komputerów. Upowszechnienie technologii informatycznych oraz wynikający z tego gwałtowny przyrost przechowywanych w formie cyfrowej informacji przypada jednak na dalszy okres. W latach 80. XX w. pojawiły się pierwsze komputery osobiste, a wraz z nimi powszechnie dostępne nośniki i aplikacje pozwalające na masowe utrwalanie danych. Kilka następnych lat przyniosło rewolucję w spojrzeniu na udostępnianie informacji. Coraz popularniejszy Internet zyskiwał rzeszę nowych użytkowników dzięki usłudze WWW, która z biegiem lat obok poczty elektronicznej stała się głównym, publicznym systemem wymiany poglądów. Usługa WWW od samego początku ulegała szybkim zmianom i była wzbogacana o coraz to wymyślniejsze i bardziej zaawansowane formy prezentacji informacji. Obecnie są one dołączane jako grafiki, dźwięk, wideo czy dokumenty tekstowe. Coraz częściej zapisywanie i przetwarzanie zasobów odbywa się także w oparciu o sieciowe bazy danych. Ilość dostępnych informacji powoduje, że problematyczne staje się opisanie i uporządkowanie gromadzonych źródeł wiedzy. Istnieje szereg aplikacji wspomagających powyższe zadania. W swojej złożoności cechują się one jednak ograniczeniami narzuconymi wcześniej przez przygotowującą je osobę – programistę. Do zastosowań typowych takie rozwiązania wydają się całkowicie wystarczające. Jednakże wyspecjalizowane potrzeby wymagają niesztampowego podejścia, które sprowadza się najczęściej do napisania nowego programu.

W celu stworzenia takiego programu trzeba poznać jeden z języków programowania (dalej: j. p.). W definicji są one określane jako sztuczne języki opierające się na określonym z góry zbiorze wyrazów oraz wiążących je reguł. Zbudowane na ich podstawie polecenia są przekładane za pomocą odpowiednich aplikacji – kompilatorów (bądź interpreterów), na formę zrozumiałą dla komputera. Istnieje wiele podziałów j. p.: według budowy, przeznaczenia lub sposobu ich wykonywania[1]. Języki skryptowe są wyodrębniane ze względu na sposób wykonywania programów. Powstałe

[1] Termin *język programowania* por. J. Kienzler, *Słownik terminów komputerowych: angielsko-polski i polsko-angielski*, Gdynia 2003, s. 170; Z. Płoski, *Słownik informatyczny*, Wrocław 2003, s. 110; B. Pffafenberger, *Słownik terminów komputerowych*, Warszawa 1999, s. 118–119.

za pomocą jednego z j. p. skrypty są interpretowane na podstawie innej, specjalnie do tego przygotowanej aplikacji. Jako największą zaletę tych języków traktuje się uproszczoną budowę i tym samym większą łatwość w ich opanowaniu. Są one także obarczone kilkoma wadami. Dotychczas za największą z nich była uznawana mała szybkość działania końcowego rozwiązania – takie programy ze względu na sposób uruchamiania są wolniejsze, gdyż każdorazowo wykonywany jest proces interpretacji całego wpisanego przez programistę kodu źródłowego. Z pozornej wady związanej ze skryptami i sposobem ich działania wynika także zaleta, jaką jest elastyczność tego typu rozwiązań. Są one bowiem w znacznym stopniu niezależne od systemu operacyjnego oraz platformy sprzętowej[2].

Skrypty są zazwyczaj wykorzystywane do tworzenia dynamicznych, często zmieniających się struktur tekstowych, a także do analizy, rozpoznawania i przetwarzania dokumentów według podanego wzorca. Z tego punktu widzenia języki skryptowe wydają się idealnym obiektem pomocniczym dla takiej dziedziny wiedzy jak informacja naukowa. Zajmuje się ona bowiem wszelkimi zagadnieniami związanymi z systemami informacyjno-wyszukiwawczymi, głównie ich projektowaniem i badaniem sposobu ich funkcjonowania. Nie mniej ważne miejsce w badaniach informacji naukowej zajmuje opracowywanie dokumentów i informacji faktograficznej[3]. W tej kwestii jest to dziedzina nauki podatna na wpływy interdyscyplinarne, w której wykorzystanie m.in. metod informatycznych staje się nieodzowne.

Przykładowe obszary wykorzystania języków skryptowych

Języki skryptowe w informacji naukowej wydają się rozwiązaniem idealnym. Pozwalają one m.in. na odczytanie i modyfikację niemal dowolnych typów danych. Umożliwiają automatyczne przetwarzanie najprost-

[2] Termin *język skryptowy* por. J. Kienzler, dz. cyt., s. 180–181; M. Trojański, *Słownik informatyki stosowanej*, Warszawa 2007, s. 385; B. Pffafenberger, dz. cyt., s. 119–120.

[3] Por. *informatologia*, *informacja naukowa*, [w:] *Podręczny słownik bibliotekarza*, oprac. G. Czapnik, Z. Gruszka, przy współpr. H. Tadeusiewicz, Warszawa 2011, s. 122; *informacja naukowa*, [w:] *Słownik encyklopedyczny informacji, języków i systemów informacyjno-wyszukiwawczych*, oprac. B. Bojar, Warszawa 2002, s. 90; J. M. Reitz, *information science*. W: *ODLIS – Online Dictionary for Library and Information Science* [on-line] [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://www.abc-clio.com/ODLIS/odlis_i.aspx.

szych informacji zawartych w zwykłych plikach tekstowych (rozszerzenie TXT), a także bardziej złożonych (np. w dokumentach PDF). Odpowiednie metody mogą posłużyć do przetworzenia zasobów pochodzących bezpośrednio ze stron umieszczonych w sieci. Daje to podstawę do stworzenia pełnoprawnego systemu informacyjno-wyszukiwawczego, wraz z robotem automatycznie gromadzącym dane oraz przetwarzającym je indekserem. Odpowiednio napisane rozwiązanie jest w stanie na bieżąco monitorować zmiany zachodzące w źródłowych dokumentach i odnotować to w swoich indeksach wyszukiwawczych przez umieszczenie stosownych uaktualnień w systemie.

Szereg dostępnych w ramach języków skryptowych funkcji matematycznych i statystycznych ułatwia natomiast testowanie wydajności i optymalizację serwisów informacyjno-wyszukiwawczych. Zgromadzone w ten sposób dane można zapisać w najpopularniejszych formatach bazodanowych, a nieco bardziej zaawansowani użytkownicy mogą podjąć próbę zaprojektowania własnej struktury bazy.

Osiągalne jest także pełnotekstowe przeszukiwanie treści dokumentów i stworzenie na podstawie powyższych działań metainformacji o zasobie lub streszczenia pozyskanych w ten sposób informacji. Proces ekstrakcji danych (ang. *data mining*) można zakończyć przez publikację wyników w postaci popularnych typów dokumentów. Jeśli umożliwia to docelowy format pliku, w którym są zapisywane treści, to mogą być one wzbogacone o wygenerowane automatycznie zaawansowane metody wizualizacji w postaci wykresów czy też grafiki.

Wykorzystując wieloplatformowość tworzonych skryptów przez ich uruchamianie na serwerze, można wspomagać, bądź wręcz wykonywać, cykliczne operacje. Wśród takich zadań mogłaby znaleźć się chociażby długoterminowa ekstrakcja danych publikowanych na łamach wybranego wcześniej serwisu informacyjnego. Automatyczne gromadzenie materiałów odbywałoby się w określonym wcześniej interwale czasowym, a pozyskane w ten sposób materiały stanowiłyby doskonałą podstawę do dalszych, bardziej szczegółowych analiz.

Powszechnie stosowane języki skryptowe

Istnieje wiele języków skryptowych, ale zaledwie kilka z nich zdobyło uznanie szerszego grona użytkowników. Według indeksu TIOBE ba-

dającego powyższe zagadnienie do czołowych pięciu języków skryptowych można zaliczyć: PHP, Perl, Python, JavaScript oraz Ruby[4]. Autor w dalszej części tekstu postanowił bliżej scharakteryzować właśnie te języki.

PHP – to język skryptowy, którego główne zastosowania są związane z tworzeniem dynamicznych stron WWW. Początki PHP sięgają drugiej połowy lat 90. ubiegłego wieku, kiedy to stworzył go Rasmus Lerdorf[5]. Szybko zauważono jednak potencjał tego rozwiązania, co przełożyło się na zainteresowanie środowiska i skupienie wokół niego aktywnej grupy osób wspomagających jego rozwój. Tworząc funkcje specyficzne dla tego języka, powiązano go z obsługą baz danych, np. MySQL. Do dzisiaj PHP jest uważany za najlepsze rozwiązanie w tej dziedzinie. Jego możliwości są jednak daleko bardziej zaawansowane, gdyż za pomocą jego odpowiednich funkcji można zbudować samodzielną aplikację pracującą w trybie graficznym, lecz przypadki właśnie takiego wykorzystania są niezwykle rzadkie. Najczęstszym przykładem zastosowania PHP do tworzenia serwisów informacyjnych WWW są systemy zarządzania treścią (ang. *Content Management System*, dalej: CMS). Najpopularniejsze z nich są wykorzystywane zarówno przez osoby prywatne, jak i instytucje, organizacje rządowe i firmy. Pod kontrolą PHP pracują takie CMS'y, jak: Typo3, WordPress, Joomla! i Drupal. Należy również nadmienić, że ze względu na swoją popularność i wykorzystanie w sieci jest to język, który obecnie instaluje się na każdym serwerze stron WWW.

JavaScript – to język stworzony w 1995 r. przez firmę Netscape (obecnie Fundacja Mozilla). Głównym – choć nie jedynym – autorem tego rozwiązania jest Brandan Eich[6]. Jego zastosowania, podobnie jak omawianego poprzednika, są związane z obsługą stron WWW. Za pomocą skryptów napisanych w tym języku można budować i obsługiwać elementy nawigacyjne, sprawdzać poprawność danych wpisywanych w formularze oraz

[4] TIOBE jest firmą, która specjalizuje się w badaniu jakości programów komputerowych. Prowadzony przez nią indeks jest budowany w oparciu o wieloletnie obserwacje rynku aplikacji i języków programowania. Por. *TIOBE Software: General Info* [on-line]. TIOBE Software [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html.

[5] *PHP: History of PHP – Manual* [on-line]. The PHP Group [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://www.php.net/manual/en/history.php.php.

[6] S. Chapman, *A Brief History of Javascript* [on-line]. About.com [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://javascript.about.com/od/reference/a/history.htm.

zapewniać interaktywność stron pod kątem zaistnienia wyznaczonych wcześniej warunków.

PERL – jest jednym z najstarszych skryptowych języków programowania, a jego geneza wiąże się z systemem UNIX. Pierwszy raz został on wykorzystany przez Larry'ego Walla w 1987 r. Był wzorowany na popularnym dotąd języku C oraz kilku istniejących wcześniej językach skryptowych (awk i sh)[7]. Obecnie jest udostępniany na licencjach zgodnych z Open Source[8], dzięki czemu wspiera go liczna rzesza jego entuzjastów. Początkowe zastosowania PERL'a wiązały się tylko z przetwarzaniem i analizą plików tekstowych, publikowaniem odnoszących się do tego raportów oraz automatyzacją czynności administracyjnych systemu operacyjnego. Jednak przez 24 lata istnienia języka zyskał on na wszechstronności. Dobrym tego przykładem jest olbrzymia baza rozszerzeń CPAN (ang. Comprehensive Perl Archive Network) zawierająca ponad 25 tys. modułów[9]. Dzięki nim możliwe jest m.in. obsługiwanie baz danych, tworzenie aplikacji sieciowych. PERL wykorzystuje się także sporadycznie w obsłudze systemów zarządzania treścią.

Ruby – to język, który stworzył Yukihiro Matsumoto w 1995 r. Wywodzi się on z połączenia kilku ulubionych języków tego twórcy, co jest powszechne dla wszystkich nowo powstających języków programowania. Utworzona przez autora fuzja miała w tym przypadku doprowadzić do powstania praktycznego i prostego języka przy zachowaniu jego ogromnych możliwości. Jest to język oparty na własnej licencji (Ruby) oraz Open Source (GPL)[10], który dynamicznie rozwija się na świecie dopiero od 2004 r. Wcześniej znany był w kraju jego autora. Jedną z przyczyn ogromnego rozkwitu wydaje się doceniana na całym świecie biblioteka programistyczna – Ruby on Rails – służąca do tworzenia aplikacji internetowych. W tej właśnie dziedzinie język Ruby wydaje się najsilniejszym rozwiązaniem. Nie jest to jego jedyne rozszerzenie, gdyż samych projek-

[7] J. Hietaniemi, *perlhist – perldoc.perl.org* [on-line]. Perl Programming Documentation [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://perldoc.perl.org/perlhist.html.

[8] W znacznym uproszczeniu jest to typ licencji oprogramowania zakładający dostępność do kodu źródłowego (zapisu poleceń programistycznych). Dzięki temu każda osoba posiadająca odpowiednią wiedzę oraz dostęp do kodu może brać czynny udział w rozwoju aplikacji.

[9] J. Hietaniemi, *The Comprehensive Perl Archive Network* [on-line] [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://www.cpan.org/.

[10] *O języku Ruby* [on-line]. Ruby. A Programmer's Best Friend [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://www.ruby-lang.org/pl/about/.

tów zwiększających jego możliwości jest około 10 tys., dostępnych tylko w ramach portalu RubyForge. Pod względem możliwości i budowy składni jest to język bardzo często porównywany z Pythonem.

Python – początki tego języka przypadają na lata 90. XX w. Jego głównym, ale nie jedynym, autorem jest Guido van Rossum. Od przeszło 10 lat nad rozwojem języka czuwa Python Software Foundation (PSF), która wszystkie jego wersje udostępnia na licencji zgodnej z Open Source[11]. Python może pochwalić się ogromną liczbą dodatkowych bibliotek programistycznych (około 22 tys.) dostępnych na stronie PyPI (ang. The Python Package Index). Zastosowania tego języka są rozległe, a do najważniejszych z nich można zaliczyć: przetwarzanie tekstów, wsparcie dla obliczeń naukowych i statystycznych, tworzenie dokumentacji, obsługę baz danych oraz tworzenie aplikacji internetowych. Mniej popularne, ale również możliwe, jest wykorzystanie tego języka w tworzeniu aplikacji multimedialnych, edukacyjnych i gier. Python jest określany jako przyjazny i uchodzi obok opisanego wcześniej języka Ruby za jeden z najbardziej przyszłościowych skryptowych języków programowania. Widoczne jest to chociażby podczas analizy wdrożeń opartych na Pythonie w instytucjach i firmach rozpoznawanych globalnie: YouTube, Google, NASA, IBM oraz Nokia[12].

Aplikacje narzędziowe dla języka Python

Rozpoczęcie pracy z większością skryptowych języków programowania wymaga przygotowania sprzyjającego temu środowiska pracy. Do najczęściej wykorzystywanych aplikacji wspomagających pracę programisty można zaliczyć zintegrowane środowisko programistyczne (ang. *Integrated Development Enviroment*, dalej: IDE). W jego skład wchodzą takie elementy, jak: specjalistyczny edytor, kompilator oraz debugger. Omówienie praktycznych rozwiązań, które odnosiłyby się do każdego z wcześniej opisanych języków, na łamach jakiegokolwiek pojedynczego artykułu jest niemożliwe. Poniżej zostaną przedstawione wybrane przez autora roz-

[11] *History and License – Python v2.7.3 documentation* [on-line]. Python Software Foundation [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://docs.python.org/license.html.

[12] *OrganizationsUsingPython* [on-line]. PythonInfo Wiki [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://wiki.python.org/moin/OrganizationsUsingPython.

wiązania ułatwiające pracę z językiem Python, który wybrano ze względu na jego wszechstronność i dużą popularność.

Eclipse – jest potężnym środowiskiem wspomagającym programowanie w wielu językach. Początkowego sukcesu tej aplikacji można upatrywać we wsparciu wielkich koncernów branży informatycznej, np. IBM, Hewlett Packard oraz Borland. W 2004 r. oprogramowanie to zostało udostępnione na licencji Open Source[13] i dzięki temu prace nad dalszym rozwojem mogą być kontynuowane przy współdziałaniu ze społecznością programistów. Wykorzystanie tego środowiska do pracy z omawianym szerzej j. p. wymaga zainstalowania i skonfigurowania trzech elementów: głównego pakietu instalacyjnego języka Python, środowiska Eclipse w wersji Classic oraz wtyczki PyDev (ang. Python IDE for Eclipse)[14].

The Eric Python IDE – stanowi rozwiązanie predefiniowane do pracy z dwoma konkurującymi ze sobą językami skryptowymi, tj. Ruby i Python. Ta aplikacja jest wyposażona w wygodny system umożliwiający szybkie rozszerzenie funkcjonalności za pomocą łatwego w obsłudze systemu wtyczek. Niestety, samo przygotowanie tego środowiska wymaga dużo pracy, gdyż wiąże się z pobraniem i zainstalowaniem pięciu niezbędnych komponentów: głównego pakietu instalacyjnego Pythona, biblioteki Qt, biblioteki PyQt, edytora QScintilla oraz samej aplikacji Eric[15].

Boa Constructor – jest środowiskiem pracy, które ma ułatwiony dostęp do projektowania i tworzenia tzw. aplikacji okienkowych. Dużą zaletę tego środowiska stanowi także zaawansowany debugger. Również w przypadku tego IDE niezbędne jest wcześniejsze wyposażenie komputera w dodatkowe składniki, takie jak: biblioteka interfejsu graficznego wxPython, główny pakiet instalacyjny Python oraz aplikacja Boa Constructor[16].

ActivePython – zdecydowanie najprostszym rozwiązaniem dla początkujących programistów wydaje się środowisko udostępniane przez firmę ActiveState. Istnieją trzy jego wersje, za darmo można skorzystać

---

[13] *Eclipse* [on-line]. The Eclipse Foundation open source community website [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://www.eclipse.org/.

[14] *PyDev* [on-line]. Appcelerator [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://pydev.org/.

[15] *The Eric Python IDE – Download* [on-line]. python-projects.org [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://eric-ide.python-projects.org/eric-download.html.

[16] R. Booysen, *Boa Contructor Home* [on-line]. SourceForge.net [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://boa-constructor.sourceforge.net/.

tylko z tej najbardziej okrojonej – Community Edition. Mimo to ma ona wiele ciekawych rozwiązań godnych uwagi. Jednym z nich jest PyPM (ang. Python Package Manager)[17] umożliwiający szybki dostęp do przygotowanych specjalnie dla tej aplikacji bibliotek programistycznych. Na uwagę zasługuje także fakt, że spośród prezentowanych rozwiązań jest to jedyne środowisku, do którego użycia wystarczy instalacja pojedynczego, zintegrowanego pakietu.

Biblioteki Pythona w wyborze

O popularności języka programowania decyduje wiele czynników. Najistotniejsza dla przyszłych użytkowników jest zazwyczaj kwestia dostępu do dużej liczby zróżnicowanych tematycznie modułów. Nie sposób wymienić i scharakteryzować każdej z bibliotek programistycznych. Zapotrzebowanie programisty jest zdefiniowane przez konkretny problem, przed którym go postawiono, tak więc dobór rozwiązań to w większości przypadków sprawa indywidualna. W związku z tym poniżej zostanie przedstawione zaledwie niewielkie spektrum dostępnych w obrębie wybranego języka modułów o różnorodnych zastosowaniach, jednak ze szczególnym uwzględnieniem zagadnień dotyczących wyszukiwania, gromadzenia, przetwarzania i udostępniania informacji.

Biblioteka standardowa – każdy z j. p. ma wbudowany zasób funkcji i procedur wykonujących zadania najbardziej elementarne. W większości przypadków możliwości tych modułów w każdym z języków są bardzo zbliżone. Biblioteka standardowa Pythona jest niezwykle rozbudowana i pozwala na pracę z wieloma typami danych i struktur. Dzięki procedurom dostępnym w modułach OS i SYS można przeprowadzić podstawowe czynności na systemie operacyjnym i plikach. W ramach osiągalnych bibliotek znajdują się także rozwiązania pozwalające na przetwarzanie informacji zapisywanych za pomocą znaczników takich języków, jak: HTML, XML czy SGML (np. moduł HTMLParse). Wśród najczęściej używanych składowych biblioteki standardowej warto wymienić również zbiór rozwiązań odpowiadający za obsługę zarchiwizowanych plików (*zipfile*), zarządzanie czasem i kalendarzem (*datetime*, *calendar*) czy moduły za-

[17] *ActivePython is Python for Windows, Mac, Linux, AIX, HP-UX & Solaris* [on-line]. ActiveState.com [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://www.activestate.com/activepython.

wierające podstawowe funkcje matematyczne (*math*, *cmath*)[18]. Biblioteka standardowa dzięki wszechstronnemu spektrum oferowanych procedur może służyć do tworzenia zróżnicowanych skryptów. Napisany za jej pomocą program może przetwarzać proste dane matematyczne, ale dzięki niej istnieje także podstawa do zbudowania zintegrowanego systemu zarządzania treścią (CMS).

Moduły wspomagające przetwarzanie tekstów – do często wykorzystywanych w tej kategorii z całą pewnością należy zaliczyć: Pypdf[19] oraz reportlab[20]. Wspomagają one odczytywanie i analizę informacji zawartych w dokumentach najpopularniejszego w ostatnich latach standardu – PDF. W przypadku chęci wygenerowania pliku z przetworzonymi już informacjami można to zrobić za pomocą tych samych rozwiązań, wzbogacając powstający dokument o tabele, wykresy albo grafikę. Wiele systemów opartych na architekturze baz danych wykorzystuje wymienione wcześniej moduły jako elementy uzupełniające ich faktyczne przeznaczenie. Wspomniane struktury operują na ogromnej ilości informacji, które w wielu przypadkach muszą zostać przefiltrowane i wyodrębnione. Wyniki otrzymane w ten sposób z systemu mają najczęściej charakter niezwykle dynamiczny, gdyż składowe samej struktury ulegają nieustannym zmianom. Zastosowanie tych modułów jest więc związane z wydrukiem zaawansowanych graficznie i składniowo dokumentów, których część może mieć charakter ulotny i ciężki do odtworzenia w późniejszym okresie. Nie wyklucza to oczywiście w żadnym razie możliwości publikowania raportów PDF zawierających dane niezmienne, statyczne.

Analiza arkuszy kalkulacyjnych – odbywa się w oparciu o dwa najbardziej znane moduły: XLRD[21] oraz XLWT[22]. Pozwalają one m.in. na przetwarzanie informacji zgromadzonych w skoroszytach programu Microsoft Excel. Możliwy jest także zapis w istniejących wcześniej dokumentach oraz tworzenie całkiem nowych plików tego rodzaju. Wymienione powy-

[18] *Python v2.7.3 documentation* [on-line]. The Python Standard Library [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://docs.python.org/library/.

[19] M. Fenniak, *PyPDF* [on-line]. Fenniak.net [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://pybrary.net/pyPdf/.

[20] *Download the ReportLab Toolkit* [on-line]. ReportLab.com [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://www.reportlab.com/software/opensource/rl-toolkit/download/.

[21] *xlrd 0.7.9* [on-line]. PyPI – the Python Package Index [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://pypi.python.org/pypi/xlrd/.

[22] *xlwt 0.7.4* [on-line]. PyPI – the Python Package Index [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://pypi.python.org/pypi/xlwt/.

żej dwa przykładowe zbiory procedur programistycznych mogą być wykorzystywane jako pomost między najpopularniejszym obecnie arkuszem kalkulacyjnym a darmowymi i otwartymi rozwiązaniami. Dzięki temu raz przetworzone informacje w programie firmy Microsoft w łatwy sposób mogą zostać zaadaptowane w autorskich rozwiązaniach skryptowych i odwrotnie.

Przetwarzanie metadanych – jest to zagadnienie mocno związane zarówno z informacją naukową, jak i z bibliologią. Wśród możliwych rozwiązań wykorzystywanych w opisie różnych typów obiektów można wymienić takie standardy, jak MARC i Dublin Core. Moduły pozwalające na ich zapis i odczyt to np. Pymarc[23], zope.dublincore[24] oraz dublincore[25]. Zastosowania funkcji wchodzących w skład wymienionych modułów wiążą się najczęściej z obsługą systemów bibliotecznych, repozytoriów, bibliotek cyfrowych czy też baz bibliograficznych. Dzięki możliwości odczytu i zapisu wymienionych wcześniej formatów łatwo wykonać m.in. ich swobodną konwersję. Pozwala to przykładowo na przetworzenie w krótkim czasie[26] kilkudziesięciu tysięcy rekordów w formacie Dublin Core i zapisanie ich za pomocą standardu MARC21.

Obsługa baz danych – jest możliwa dzięki bibliotece standardowej Pythona, za pomocą której tworzy się w większości przypadków proste, autorskie struktury bazodanowe. Wśród rozwiązań odchodzących w zapomnienie, ale ciągle możliwych do zastosowania wymienia się standard ISIS i obsługujący go moduł PyIsis[27]. Najczęściej wykorzystywane są jednak powszechnie uznane za standard rozwiązania oparte na technologiach MySQL (SQLAlchemy[28]). Ostatnia z wymienionych technologii jest obecnie stosowana w większości systemów Open Source, czyli także tych

[23] *pymarc 2.8.4* [on-line]. PyPI – the Python Package Index [dostep 30 listopada 2012]. Dostępny w World Wide Web: http://pypi.python.org/pypi/pymarc/2.8.4.

[24] *zope.dublincore 3.8.2* [on-line]. PyPI – the Python Package Index [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://pypi.python.org/pypi/zope.dublincore/3.8.2.

[25] *dublincore 1.0* [on-line]. PyPI – the Python Package Index [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://pypi.python.org/pypi/dublincore/1.0.

[26] Czas poświęcony na konwersję jest zależny od szybkości komputera, na którym jest wykonywana powyższa operacja. Przy obecnej prędkości komputerów nie powinien on przekroczyć kilkunastu minut.

[27] *pyIsis 0.1* [on-line]. PyPI – the Python Package Index [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://pypi.python.org/pypi/pyIsis/0.1.

[28] *The Database Toolkit for Python* [on-line]. SQLAlchemy [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://www.sqlalchemy.org/.

związanych z biblioteką i jej działalnością informacyjną. Przykładowo MySQL znajduje zastosowanie w systemach CMS, bibliotekach cyfrowych oraz systemach bibliotecznych. Moduł SQLAlchemy umożliwia więc pobieranie i przetwarzanie danych zapisanych w tym formacie, czyli bezpośrednią ingerencję w struktury wspomnianych wcześniej typów aplikacji.

Interfejsy graficzne – w większości przypadków są zbędne, gdyż duża część tworzonych skryptów nie wymaga takiego poziomu złożoności, by wdrożyć formę atrakcyjną wizualnie. Jeśli jednak z jakichś powodów istnieje taka potrzeba, do wyboru jest kilka odrębnych bibliotek programistycznych: PyQt, PyGTK lub wxPython[29]. Zastosowanie wymienionych bibliotek programistycznych jest zasadne w przypadku chęci zwiększenia wygody korzystania z programu oraz w ramach pomocy dla mniej wprawnych użytkowników. Interfejs takiego skryptu może zostać dostosowany tak, by wykonywać kilka bądź kilkadziesiąt operacji przy minimalnym zaangażowaniu obsługującej go osoby.

Aplikacje internetowe – są związane z wieloma aspektami, ale wykorzystanie specjalnych modułów, takich jak Django[30] czy Pylons[31], jest wymagane tylko i wyłącznie w przypadku tworzenia rozwiązań o dużej złożoności. W ramach wymienionych modułów istnieje szereg udogodnień pozwalających np. na projektowanie szablonów, dzięki czemu znacznie szybciej i łatwiej można stworzyć system zarządzania treścią. Dostępne metody pozwalają także na tworzenie wielojęzycznych interfejsów oraz nadzorowanie użytkowników i ich uprawnień w ramach tworzonego systemu.

Podsumowanie

Potencjał skryptowych języków programowania i wynikających z niego korzyści dla zagadnień znajdujących się w sferze zainteresowań informa-

[29] *24.7. Other Graphical User Interface Packages* [on-line]. The Python Standard Library [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://docs.python.org/library/othergui.html.

[30] *Framework webowy dla perfekcjonistów z terminami* [on-line]. Django.pl [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://www.django.pl/.

[31] *About The Pylons Project Framework* [on-line]. The Pylons Project [dostęp 30 listopada 2012]. Dostępny w World Wide Web: http://www.pylonsproject.org/projects/pylons-framework/about.

cji naukowej jest znaczny. Wnioski z analizy dostępnych rozwiązań nie są jednak optymistyczne i wskazują na wykorzystanie skryptów głównie w działalności komercyjnej, ze szczególnym uwzględnieniem sektora teleinformatycznego oraz biznesu elektronicznego. Firmy o wymienionych profilach stać bowiem na wynajęcie programistów rozwijających skrypty predefiniowane tylko na ich potrzeby. Dla pozostałych sektorów działalności proces tworzenia skryptów jest często zbędny, gdyż większość firm i instytucji opiera się na ogólnodostępnych aplikacjach. Brak potrzebnych funkcji i elastyczności istniejących rozwiązań w wielu przypadkach stanowi sprawę na tyle marginalną, że nie powoduje potrzeby szukania innych, niestandardowych opcji rozwikłania problematycznych kwestii.

Zakładając jednak chęć wykorzystania skryptowych języków programowania do wsparcia zagadnień z zakresu informacji naukowej, można napotkać na szereg utrudnień. Podstawowym problemem jest ciągle sam proces tworzenia programu. Podjęto co prawda kilka prób mających na celu upowszechnienie składni języków programistycznych, np. przez upodobnienie jej do wyrażeń języka naturalnego, lecz nie odniosło to przewidywanego, spektakularnego sukcesu. Dla każdego z języków wymagane jest bowiem opanowanie dużej liczby dozwolonych w użyciu struktur. Sytuacji nie ułatwiają także zaawansowane aplikacje narzędziowe. Ich głównym zadaniem jest uwolnienie osoby pracującej nad skryptem od wielu niedogodności technicznych. Nie ułatwiają one jednak w znaczący sposób samego procesu kreowania rozwiązań. W tym zakresie potrzebna jest chociaż podstawowa wiedza z zakresu stosowanych w informatyce algorytmów. Zaskakującym problemem okazuje się również szerokie spektrum oferowanych bibliotek programistycznych. Gdy zrealizowanie rozważanego procesu jest możliwe za pomocą kilku dostępnych modułów, należy wybrać sposób optymalny dla konkretnego zastosowania pod takimi względami, jak np. szybkość, stabilność, bezpieczeństwo czy zadowalająca szczegółowość wyniku przetwarzanej informacji. O ile powyższe problemy są powszechnie znane każdemu informatykowi, o tyle osobom postronnym mogą one sprawić niemały kłopot. Dlatego obecnie poprawne wydaje się stwierdzenie, że przetwarzanie informacji za pomocą skryptowych języków programowania znacznie ułatwiło ten proces… programistom, a tworzenie jakichkolwiek skryptów dla szerszego grona użytkowników nadal okazuje się prawie niewykonalne.

Applications of scripting languages in information services

Abstract: The undeniable fact of the past few years is the rapid increase in published documents in electronic form and the development of the Internet. Because of its easy access to the established information in this way he gains a larger group of users. Available in this mode documents from the technological point of view are extremely diverse. Starting from a simple web page, through the popular PDF files, and audio and video ending. A precise characterization information as different types of documents is time consuming and problematic activity. The current increase in quantity of published content also excludes issues more often manual description and processing of such data. Among such a dynamically changing structures are just some of the problems encountered in the charge of this branch of knowledge – information science. Solutions to assist with these issues often come in the other fields of science, in this case science. In most of the analysis and processing of documents is enough software written by the public, but sometimes used this program you need to create yourself. Possibly the most in the latter case then it seems to use the scripting programming language, which is treated in this article.

Keywords: document processing, information science, programming languages, programming libraries in Python, Python, scripting