STUDIA INFORMATICA 2000
Volume 21 Number 3 (41)

Zdzisław SROCZYŃSKI
Politechnika Śląska, Instytut Matematyki

ROZSZERZANIE MOŻLIWOŚCI PRZEGLĄDAREK INTERNETOWYCH NA PRZYKŁADZIE MODUŁÓW PLUG-IN DLA NETSCAPE NAVIGATOR

Streszczenie. W pracy przedstawiono podstawowe założenia projektowania i tworzenia modułów plug-in dla przeglądarki Netscape Navigator. Istotnym elementem pracy jest porównanie wydajności różnych metod rozszerzania możliwości przeglądarek internetowych. Wskazano również na celowość zastosowania narzędzi programistycznych typu RAD w procesie projektowania modułów plug-in.

PLUG-IN MODULES FOR NETSCAPE NAVIGATOR DEVELOPMENT

Summary. The paper describes main problems connected with developing plug-ins for Netscape Navigator web browser. Important part of the article is a comparison of different ways of extending browsers capabilities, which points native plug-ins as the fastest method. As a conclusion from testing Netscape plug-in SDK there was presented a proposal to use RAD programming tools (as Borland C++Builder and Delphi) in plug-in development.

1. Wprowadzenie – przegląd metod rozszerzania funkcjonalności przeglądarek internetowych

Przeglądarki internetowe z założenia zdolne są przedstawiać i wizualizować typy danych, związane z dokumentami internetowymi, w szczególności pliki HTML oraz graficzne. Coraz częściej jednak zdarza się konieczność umieszczenia na stronie WWW specyficznych danych spoza tego podstawowego zestawu. Ma to miejsce nie tylko w przypadku danych multimedialnych, takich jak dźwięki, animacje i prezentacje, ale również wraz z rozwojem zapotrze-

bowania na przedstawianie pewnych specyficznych typów danych, na przykład wzorów matematycznych w formacie MathML czy LaTeX.

Rozwiązaniem tych problemów mogą być specjalnie przygotowane rozszerzenia przeglądarki (najczęściej projektowane dla konkretnego programu, a nawet systemu operacyjnego). Dla Netscape Navigatora (Komunikatora) są to tzw. „wtyczki" — moduły plug-in, dla Internet Explorera rolę taką mogą pełnić kontrolki ActiveX. Konkurencyjną ideą jest język Java, przenośny i uniwersalny, ale przez to potencjalnie znacznie mniej wydajny.

Pomimo to, że w pracy niniejszej uwaga skupia się na modułach plug-in dla Navigatora, przed przystąpieniem do dalszych rozważań warto przeanalizować wyliczone powyżej metody rozszerzania funkcjonalności przeglądarek ze względu na ich wydajność.

2. Porównanie wydajności różnych metod rozszerzania możliwości przeglądarek internetowych

W celu określenia najbardziej efektywnego sposobu rozszerzenia możliwości przeglądarek internetowych przeprowadzono zestaw testów. Do testów użyto najpopularniejszych wśród polskich internautów przeglądarek Sylaba Navigator w wersji 4.7 oraz MS Internet Explorer w wersji 5. W ramach testu przygotowano moduł wyświetlający na ekranie wiele ciągów znaków w różnych, każdorazowo obliczanych pozycjach. Jako punkt odniesienia posłużył program napisany w C++ (za pomocą pakietu Borland C++ Builder). Natomiast rozszerzenia przeglądarek opracowano jako moduł plug-in (tylko dla Navigatora), applet Javy oraz kontrolkę ActiveX (tylko dla MSIE). Applet Javy został przygotowany za pomocą JDK1.2.2 firmy SUN, kontrolka ActiveX powstała przy użyciu kompilatora Visual Basic 5 CCE (Control Creation Edition).

ActiveX jest jak dotąd technologią nieprzenośną w przeciwieństwie do Javy i (przynajmniej częściowo) plug-in'ów Netscape. Jest to z pewnością istotna wada. Wieloplatformowa Java daje też dużo większy poziom bezpieczeństwa, i to zarówno w porównaniu z ActiveX, jak i Netscape plug-in (są to bowiem *de facto* programy wykonywalne skompilowane dla konkretnego systemu operacyjnego). Jednak w wyścigu o wydajność zalety takie stają się przeszkodami. Jak dotąd, żadna wirtualna maszyna Javy nie zbliżyła się osiągami do programów kompilowanych dla konkretnego typu procesora.

Testy zostały przeprowadzone w środowisku MS Windows 98SE PL na trzech komputerach różniących się wydajnością procesora i ilością pamięci operacyjnej. Niezależnie od konfiguracji sprzętowej zaobserwowano wyraźną przewagę rozwiązania opartego na module plug-in dla przeglądarki Netscape Navigator. Należy w tym miejscu zaznaczyć, iż można się

spodziewać podobnej wydajności kontrolek ActiveX tworzonych przy użyciu języków kompilowanych, takich jak C++ czy Delphi, a nie Visual Basic'a. Test nie uwzględniał czasu ładowania plug-in, OCX czy wirtualnej maszyny Javy ze względu na trudność w precyzyjnym pomiarze. Jednak prosta obserwacja pozwala stwierdzić kilkakrotnie dłuższą inicjalizację otoczenia Javy w porównaniu z konkurencyjnymi rozwiązaniami. Wyniki przedstawione są w tabeli 1 oraz na rys. 1.

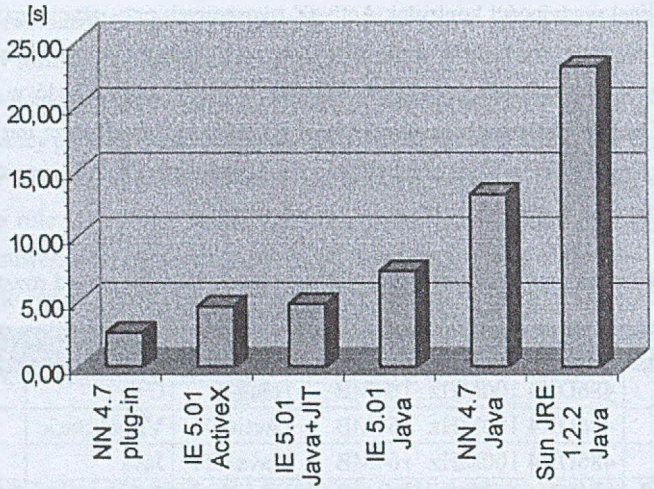
Tabela 1

Prędkość wykonania przykładowego zadania za pomocą różnych metod rozszerzania możliwości przeglądarek internetowych

Przeglądarka	Procesor	Pamięć	Metoda	Język	Wynik (s)
NN 4.7	486DX4 100MHz	64MB	plug-in	C++	17,30
IE 5.00	486DX4 100MHz	64MB	ActiveX	Visual Basic	33,60
IE 5.00	486DX4 100MHz	64MB	Java+JIT	Java	55,00
IE 5.00	486DX4 100MHz	64MB	Java	Java	94,00
NN 4.7	486DX4 100MHz	64MB	Java	Java	134,50
Sun JRE 1.2.2	486DX4 100MHz	64MB	Java	Java	—
NN 4.7	AMD K2 400MHz	32MB	plug-in	C++	2,60
IE 5.01	AMD K2 400MHz	32MB	ActiveX	Visual Basic	4,60
IE 5.01	AMD K2 400MHz	32MB	Java+JIT	Java	4,80
IE 5.01	AMD K2 400MHz	32MB	Java	Java	7,30
NN 4.7	AMD K2 400MHz	32MB	Java	Java	13,20
Sun JRE 1.2.2	AMD K2 400MHz	32MB	Java	Java	23,00
NN 4.7	AMD K2 400MHz	64MB	plug-in	C++	1,00
IE 5.00	AMD K2 400MHz	64MB	ActiveX	Visual Basic	2,30
IE 5.00	AMD K2 400MHz	64MB	Java+JIT	Java	2,40
IE 5.00	AMD K2 400MHz	64MB	Java	Java	4,00
NN 4.7	AMD K2 400MHz	64MB	Java	Java	10,00
Sun JRE 1.2.2	AMD K2 400MHz	64MB	Java	Java	—

Przedstawione powyżej wyniki wskazują na bardzo dużą wydajność kompilatora Javy Just In Time firmy Microsoft, która zbliża się do oferowanej przez języki kompilowane. W teście celowo nie porównywano wydajności innych wirtualnych maszyn Javy, np. firm IBM czy Novell, chodziło bowiem o przetestowanie tych środowisk, z którymi styka się przeciętny użytkownik Internetu.

W zastosowaniach, w których wydajność wizualizacji jest sprawą krytyczną, moduł plug-in dla Navigatora jest najlepszym wyborem metody rozszerzenia możliwości przeglądarki.



Rys. 1. Porównanie wydajności różnych metod rozszerzania możliwości przeglądarek internetowych – czas wykonania przykładowego testu na komputerze AMD K2 400MHz z 32MB RAM z systemem MS Windows 98

Fig. 1. Extending browsers — performance comparison for AMD 400MHz system with 32MB RAM and MS Windows 98

3. Moduły plug-in dla Netscape Navigator — możliwości i zastosowania

Podstawowe zadania modułów plug-in to prezentacja danych o specyficznych, nie rozpoznawanych przez przeglądarkę formatach oraz uzupełnianie możliwości funkcjonalnych przeglądarki.

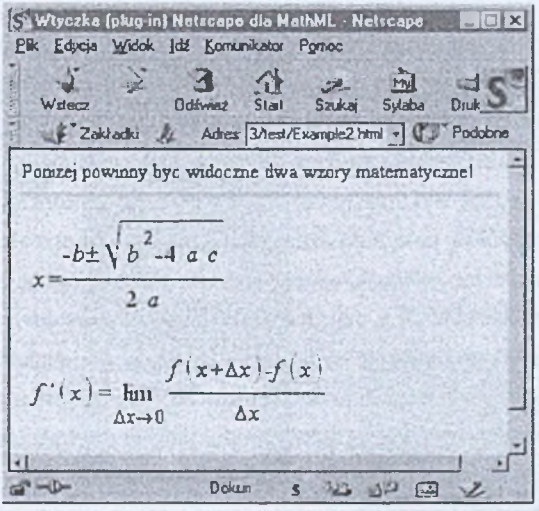
Funkcje udostępniane w Plug-in API umożliwiają dostęp do zasobów przeglądarki i pozwalają rozszerzyć jej funkcjonalność, między innymi poprzez:

- rejestrację modułu do obsługi określonych typów MIME,
- wyświetlanie informacji w pewnej określonej części okienka przeglądarki,
- przechwytywanie zdarzeń związanych z klawiaturą i myszką,
- wysyłanie i odbieranie danych poprzez Internet,
- wywoływanie nowych adresów URL w przeglądarce,
- wyświetlanie informacji we fragmentach dokumentu HTML.

W niektórych przypadkach w razie konieczności rozszerzenia możliwości Navigatora zastosowanie modułów plug-in jest najbardziej uzasadnione. Dzieje się tak wówczas, gdy w rozpatrywanym projekcie istotną rolę odgrywa:

- wykorzystanie istniejącego kodu C++,
- duża wydajność,
- wykorzystywanie możliwości konkretnego systemu operacyjnego,
- integracja istniejących aplikacji z Internetem/Intranetem,
- używanie rodzimych metod niskopoziomowych niedostępnych przy użyciu języka Java.

Efekty działania przykładowego modułu plug-in, wyświetlającego wzory matematyczne zapisane w standardzie MathML, przedstawia rys. 2.



Rys. 2. Przykład działania modułu plug-in dla Netscape (Sylaba) Navigator 4.x wyświetlającego wzory zapisane w języku MathML
Fig. 2. Example of MathML plug-in for Netscape Navigator 4.x

4. Cykl życia modułu plug-in dla Netscape Navigator

Plug-in API dla Komunikatora składa się z dwóch grup funkcji oraz struktur danych:

- funkcje, których implementacja należy do autora modułu, których nazwa zaczyna się na NPP, np. NPP_New. Funkcje te są wywoływane przez Komunikatora,
- funkcje implementowane przez przeglądarkę, których nazwa zaczyna się na NPN – wywoływane przez plug-in, np. NPN_Write,
- struktury danych z nazwą zaczynającą się od NP, np. NPWindow.

Cykl życia modułu plug-in jest uzależniony od zawartości przetwarzanej przez Navigatora strony WWW. Plug-in wywoływany jest dopiero wówczas, gdy przeglądarka napotka odpowiedni znacznik HTML (EMBED lub OBJECT). Po rozpoznaniu typu MIME dla danych wskazanych w napotkanym znaczniku następuje załadowanie kodu „wtyczki" do pamięci; jeśli po raz pierwszy, przeglądarka wywołuje NPP_Initialize, aby zainicjalizować moduł.

Następnie tworzony jest nowy egzemplarz modułu (za pomocą funkcji NPP_New). Mogą istnieć wielokrotne egzemplarze modułu, jeśli na jednej stronie znajduje się kilka osadzonych obiektów lub jednocześnie trwa wizualizacja kilku stron zawierających dane tego samego typu.

Moduł implementuje pewien podzbiór funkcji z grupy NPP, wywołując potrzebne funkcje z grupy NPN.

Egzemplarz modułu jest usuwany za pomocą funkcji NPP_Destroy, gdy użytkownik opuszcza stronę lub zamyka okienko przeglądarki. Wraz z usunięciem ostatniego egzemplarza kod modułu jest usuwany z pamięci — Komunikator wywołuje funkcję NPP_Shutdown.

Moduły plug-in dla Navigatora to dynamicznie ładowane biblioteki (DLL w przypadku systemu Windows), mogące wykorzystywać w pełni możliwości konkretnego systemu operacyjnego. Realizacja modułów polega na rozszerzeniu możliwości przeglądarki, nie mogą więc one działać jako samodzielne aplikacje, co daje znaczną oszczędność kodu. Plug-in jest znacznie mniejszy od komponentów OLE czy OpenDOC znanych z systemów Windows i MacOS, charakteryzuje się przy tym podobną wydajnością wynikającą z implementacji jako rodzima biblioteka systemowa. Będąc jedynie rozszerzeniem Navigatora jest przy tym łatwiejszy do zaprojektowania i wdrożenia. Z drugiej strony moduł plug-in pozwala uzyskać podobną funkcjonalność i integrację z przeglądarką jak języki uniwersalne, np. Java czy Javascript, przewyższając je często łatwością implementacji (można wykorzystywać dostępny kod w języku C) oraz zawsze, jeśli chodzi o prędkość wykonania (gdyż w przypadku modułu plug-in nie ma oczywiście mowy o jakiejkolwiek interpretacji czy ładowaniu „wirtualnej maszyny"). Trzeba przyznać, iż bezpieczeństwo użycia Javy jest znacznie większe niż przy zastosowaniu modułów plug-in, lecz właśnie dostęp do wszystkich możliwości konkretnego systemu stanowi o przewadze tych drugich, szczególnie w zakresie multimediów, gdzie wydajność ma decydujące znaczenie.

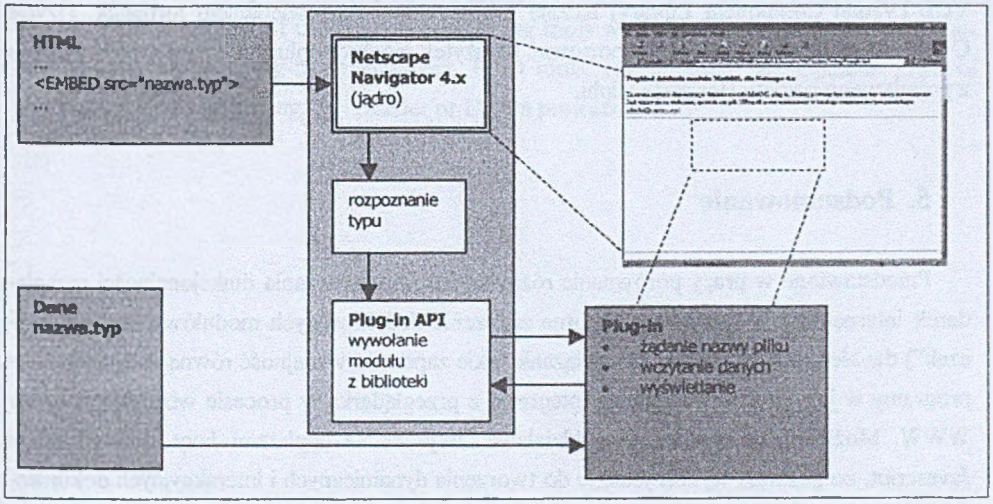
Moduły plug-in mogą występować w dwóch wersjach w zależności od sposobu wykorzystania ekranu:

- okienkowe, mogące umieszczać informacje we własnym oknie,
- pozbawione okienka, rysujące na fragmencie dokumentu HTML w okienku przeglądarki.

4.1. Implementacja modułu plug-in

Przygotowanie najprostszej „wtyczki" dla Navigatora czytającej dane z pliku tymczasowego na dysku lokalnym sprowadza się do rozszerzenia szablonu zawartego w Netscape Plug-in SDK o implementacje funkcji NPP_Initialize, NPP_NewStream, NPP_StreamAsFile oraz obsługi komunikatu WM_PAINT w funkcji obsługi okna (ustawianej za pomocą funkcji NPP_SetWindow). Po odczytaniu danych z pliku można wymusić odrysowanie fragmentu okienka przypisanego do modułu plug-in za pomocą funkcji NPN_InvalidateRect i NPN_ForceRedraw.

Ze względu na specyfikę zarządzania pamięcią przez Navigatora, alokowanie i zwalnianie obszarów pamięci powinno odbywać się za pomocą funkcji NPN_MemAlloc oraz NPN_MemFree.



Rys. 3. Schemat działania modułu plug-in dla Netscape (Sylaba) Navigator 4.x
Fig. 3. Plug-in structure for Netscape Navigator 4.x

4.2. Współpraca modułów plug-in z językami Java i Javascript

Istnieje również możliwość połączenia modułu plug-in z Javą czy Javascript'em za pomocą technologii Netscape LiveConnect. Użycie w języku Javascript funkcji zaimplementowanej w C++ polega wówczas na wywołaniu właściwej metody obiektu odpowiadającego „wtyczce". Takie podejście daje niezwykłe możliwości tworzenia dynamicznych stron WWW wyświetlających specyficzne dane (np. wzory matematyczne) zmieniane za pomocą skryptów Javascript automatycznie czy też w interakcji z użytkownikiem. Jednym z potencjalnych zastosowań takiej technologii może być rozbudowanie standardu języka MathML, pozwalającego

opisywać wzory matematyczne zgodnie z założeniami XML, o elementy makrodefinicji i przekazywania parametrów zmieniających się w czasie wizualizacji strony WWW (w szczególności chodzi o dane wprowadzane przez użytkownika).

4.3. Wykorzystanie pakietów RAD C++Builder i Delphi do tworzenia „wtyczek" dla Navigatora

Bardzo istotna dla rozwijania modułów plug-in jest możliwość kompilowania ich po niewielkich przeróbkach za pomocą pakietu C++Builder firmy Borland. Projekt i testowanie „wtyczki" przeprowadza się wówczas w środowisku RAD (Rapid Application Development), korzystając m.in. z wygodnego debuggera. Dopiero ostateczna kompilacja do biblioteki DLL tworzy właściwy moduł plug-in. Stworzona w ten sposób „wtyczka" może używać biblioteki VCL (Visual Component Library) łącznie z tworzeniem i wyświetlaniem formatek. Użycie C++Buildera pozwala również importować na użytek modułów plug-in kod w Object Pascalu z popularnego pakietu Borland Delphi.

5. Podsumowanie

Przedstawione w pracy porównanie różnych metod rozszerzania funkcjonalności przeglądarek internetowych wskazuje na istotne znaczenie dedykowanych modułów plug-in („wtyczek") dla Netscape Navigator. Rozwiązanie takie zapewnia wydajność równą osiąganej przez programy w języku C++ oraz pełną integrację z przeglądarką w procesie wizualizacji strony WWW. Możliwe jest również współdziałanie „wtyczek" z appletami Javy oraz skryptami Javascript, co pozwala wykorzystać je do tworzenia dynamicznych i interakcyjnych dokumentów internetowych.

Istotnym uzupełnieniem oferowanego przez firmę Netscape pakietu Plug-in SDK może być użycie w procesie projektowania „wtyczek" narzędzi programistycznych typu RAD firmy Borland. Oprócz zwiększenia szybkości tworzenia rozwiązanie takie daje możliwość łatwego przeniesienia na platformę Internetu istniejących aplikacji stworzonych w popularnych pakietach Delphi i C++Builder.

LITERATURA

1. Netscape Communications Corporation, *Plug-in Guide*, 1998.
2. Netscape Communications Corporation, *LiveConnecting Plug-ins with Java*, 1998.

Recenzent: Dr inż. Ryszard Winiarczyk

Wpłynęło do Redakcji 24 maja 2000 r.

Abstract

Web documents are in general intended to contain formatted text and some graphic. But very often a necessity to show a special kind of data appears. Comparison of ways of extending web browsers possibilities is important part of this article. The result pointed Netscape native plug-ins as the best solution in performance-sensitive applications (Table 1, Fig. 1).

Next part of the paper is a description of plug-in structure and lifetime cycle of the plug-in (Fig. 3). The possibility of use RAD programming tools with Netscape Plug-in SDK was noticed during the test plug-in development. RAD tools, except faster development, improve also code re-use for existing C++Builder or Delphi projects.