

Asynchronous JavaScript and XML Custom Controls, User Controls *ASP.NET*

Wiktor Zychla

Uniwersytet Wrocławski

Instytut Informatyki

29 listopad 2005

AJAX

AJAX (Asynchronous Javascript and XML) - nowa nazwa dla składanki 5 technologii:

- prezentacja w HTML/XHTML...
- ...danych XML...
- ...przenoszonych przez XMLHttpRequest...
- ...przetwarzanych w modelu DOM...
- ...za pomocą Javascript.

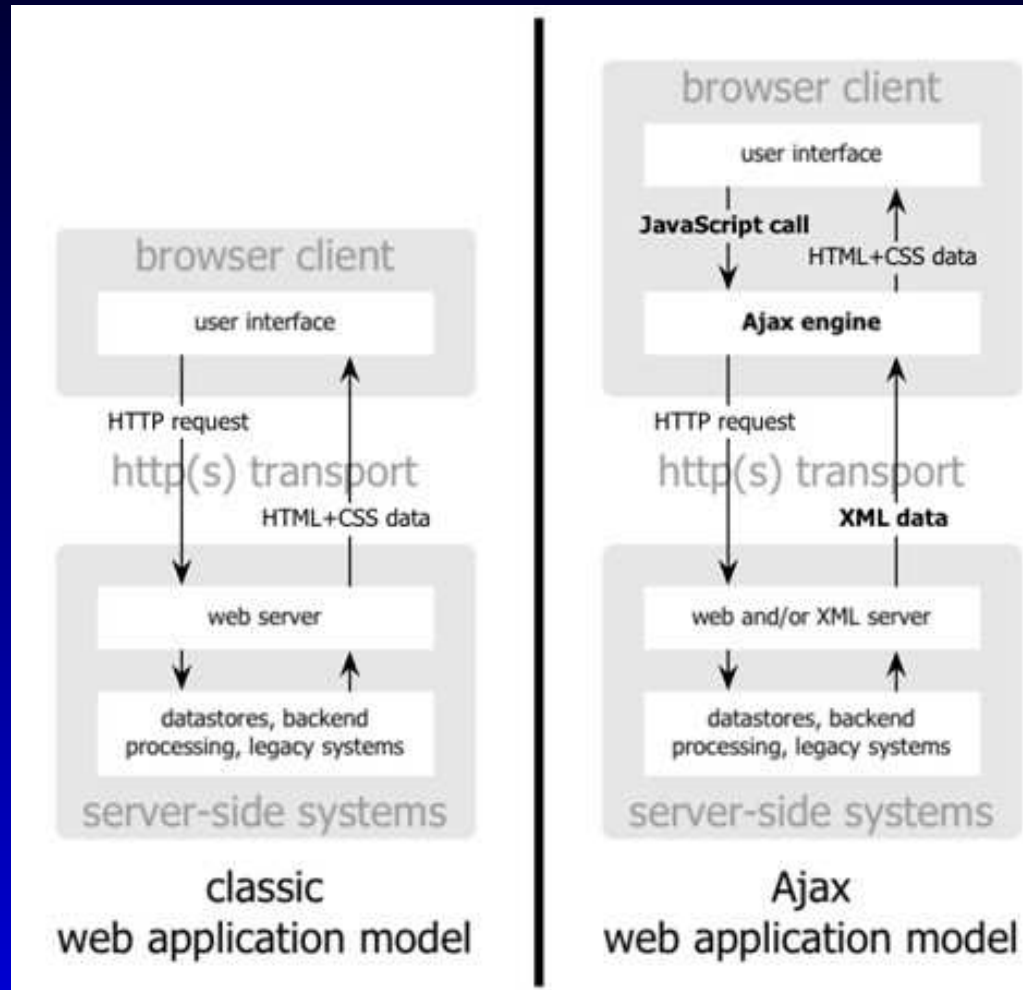
Na przykład Google Suggest czy Google Maps.

AJAX

Callback zamiast postback:

- do serwera nie jest odsyłany `</FORM>` jak przy postback
- zamiast tego używa się JavaScript do zbudowania żądania skierowanego do określonego miejsca na serwerze
- serwer odpowiada zwykłym tekstem lub XML
- klient przetwarza odpowiedź za pomocą JavaScript

AJAX



AJAX

Różne implementacje w ramach .NET, m.in.:

- Scripting Callbacks (ASP.NET 2.0) - nie wybiega znacznie poza wariant klasyczny
- AJAX.NET (atrybuty + logika serwera mapowana na logikę klienta) - piękna architektura ale dyskusyjna wydajność
- inne...

W ASP.NET 2.0 niektóre funkcje komponentów wbudowanych oparto o Scripting Callbacks, na przykład dynamicznie rozwijający węzły TreeView, czy nie wymagający Postback GridView.

AJAX

Zalety AJAX:

- rewelacyjne zachowanie się interfejsu użytkownika aplikacji Web
- znaczne zmniejszenie ruchu klient-serwer
- możliwość

Wady AJAX:

- nie wspierany przez wszystkie przeglądarki
- w ujęciu klasycznym - niewygodny do integracji z istniejącymi technologiami
- wiele różnych implementacji w tym oficjalne i nieoficjalne
- wymaga programowania JavaScript

AJAX

DEMO: minimalny przykład

AJAX

Literatura:

- Jessie James Garret - Ajax: A New Approach to Web Applications
- Ajax.NET - A free library for the Microsoft .NET Framework
- Dino Esposito - Script Callbacks in ASP.NET

Kontrolki ASP.NET

- Podobnie jak w przypadku aplikacji desktop, programista ma możliwość tworzenia własnych komponentów ASP.NET. Okoliczności w których warto pisać własne kontrolki są dokładnie takie same.
- Dwa modele tworzenia kontrolek:
 - deklaracyjny - User Controls
 - programowany - Custom Controls

User Controls

- pliki *.ASCX
- budowane wizualnie
- nie wymagają kompilacji
- przy próbie użycia kompilator tworzy kod klasy dla kontrolki dokładnie tak samo jak w przypadku stron (*.ASPX)
- z tego powodu w przypadku tworzenia dynamicznego nie można używać new tylko LoadControl

Zastosowanie: fragmenty interfejsu użytkownika o wielokrotnym zastosowaniu.

User Controls

DEMO: SiteHeader+Site
Footer

Custom controls

- klasy dziedziczące z `Control` lub `WebControl`
- zawartość budowana w kodzie
- wymagają kompilacji
- osadzane w zestawach jak wszystkie inne klasy
- w przypadku tworzenia dynamicznego wystarczy `new`

Zastosowanie: jakakolwiek logika po stronie serwera, która może (ale nie musi) mieć jakąś postać wizualną

Custom controls

Modele dziedziczenia:

- `Control` - dla elementów niewizualnych
- `WebControl` - dla elementów wizualnych, automatycznie udostępnia m.in. `Font`, `Width`, `Height`, `ForeColor`, `BackColor`
- jakakolwiek istniejąca kontrolka, np. `TextBox` czy `Button`

Custom Controls

DEMO: kontrolki Control i
WebControl

Custom Controls

DEMO: kontrolki Control i
WebControl

Custom Controls

DEMO: kontrolka niewizualna

User vs Custom Controls

	User	Custom
deployment	single app	możliwy GAC
layout	łatwiejszy statyczny	dowolny
design-time	słaby	duże wsparcie
wydajność	taka sama	taka sama

Kontrolki ASP.NET

Literatura:

- Nikhil Kothari, Vandana Datye - Developing ASP.NET Server Controls and Components