

RIA – Rich Internet Application

Komplex felhasználói felülettel rendelkező web-alkalmazások

Áttekintés

- ▶ Bevezető
- ▶ Flex – áttekintés
- ▶ ActionScript
- ▶ Felhasználói felület tervezése
- ▶ Események
- ▶ Szerver oldali szolgáltatásokkal való kommunikáció

web alkalmazás vs. desktop alkalmazás – előnyök:

- ▶ web alkalmazás – központilag karbantartható
- ▶ desktop alkalmazás – felhasználóbarát felület

a felhasználók mindkettő előnyét szeretnék kihasználni

desktop alkalmazás használatához hasonló élmény nyújtására való törekvés:

- ▶ alkalmazás-logika egy részének kliens oldalra való költöztetése (JavaScript keretrendszerek segítségével) ... böngészőfüggőség
- ▶ RIA – platformfüggetlenség

RIA – Rich Internet Application

RIA – általános jellemzők

- ▶ Web 1.0 (RIA-t megelőző alkalmazások):
 - ▶ hiperszövegen alapuló alkalmazás
 - ▶ HTML, JavaScript, CSS
- ▶ Web 2.0:
 - ▶ vizuális elemek dominálnak
 - ▶ általában böngésző plug-in szükséges
 - ▶ aszinkron kommunikáció a szerverrel
 - ▶ az alkalmazás jelentős része a kliens gépen fut (csökkentett hálózati forgalom)
 - ▶ komplex desktop alkalmazáshoz hasonló
 - ▶ biztonsági korlátozások (korlátozott hozzáférés a kliens géphez)
 - ▶ nem használhatók a böngészők vezérlőgombjai
 - ▶ nem indexelhető keresőmotrok által

RIA fejlesztő eszközök

- ▶ Flex (Adobe)
- ▶ Silverlight (Microsoft)
- ▶ JavaFX (Sun Microsystems → Oracle)
- ▶ ... AJAX

Flex keretrendszer - RIA development toolkit

Flex keretrendszer

- ▶ gazdag függvénykönyvtárat, komponenseket bocsát rendelkezésre
- ▶ az alábbi nyelvekre alapoz:
 - ▶ ActionScript – ECMA szabványnak megfelelő szkript-nyelv
 - ▶ MXML – XML alapú deklaratív nyelv (ActionScript-té fordul)

az alkalmazás telepíthető:

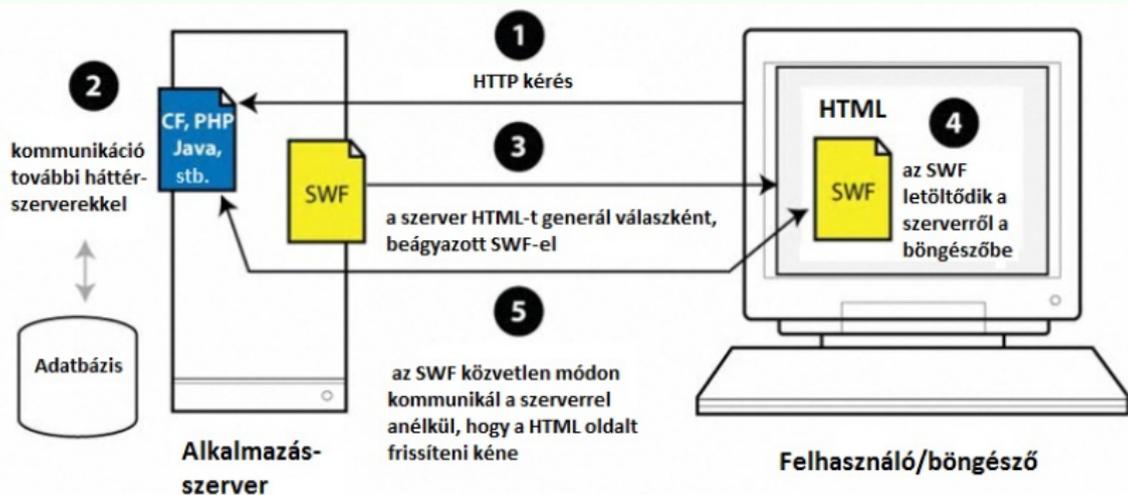
- ▶ web alkalmazásként (Flash Player - plug-in használata)
- ▶ desktop alkalmazásként (Adobe Integrated Runtime -AIR- használata)

Fejlesztéshez szükséges eszközök

- ▶ Flex keretrendszer – nyílt forráskódú
- ▶ Flex SDK (fordító, debugger) – ingyenes
→ .swf
- ▶ tetszőleges (ingyenes) szerkesztő
vagy
Flash Builder 4 fejlesztői környezet (Eclipse plug-in) – kereskedelmi
(korábbi neve: Flex Builder)
- ▶ Flash Player (böngésző plug-in)

Flash vs. Flex

- ▶ Flash – inkább animáció létrehozására (idővonal, rétegek, keretek)
- ▶ Flex – adatfeldolgozás



A szerverrel való kommunikáció egy SWF-ből

ActionScript + MXML

- ▶ MXML – tipikusan a felhasználói felület megadására
- ▶ **ActionScript** – alkalmazás-logika implementálására

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:s="library://ns.adobe.com/flex/spark"
  <fx:Script source="myFunctions.as"/>
  <fx:Script>
    <![CDATA[ // ActionScript kód:
      import mx.controls.Alert;
    ]]>
  </fx:Script>
  <s:Button label="Klikk" click="Alert.show('Proba')"/>
</s:Application>
```

ActionScript (3.0)

- ▶ ECMAScript szabványnak megfelelő (JavaScript-hez hasonlóan), objektumorientált script nyelv
- ▶ Pontos adattípus meghatározást használ (strict data typing)
`var myVar:String="Bla";`
- ▶ statikus (fordításidejű) illetve dinamikus (futásidejű) típusellenőrzés
- ▶ a változók mind objektumok (Flex 4)

változó típusok (top-level package):

- ▶ egyszerű: int, uint, Number, String, Boolean, Null, void
- ▶ összetett: Object, Array, Date, Error, Function, RegExp, XML, XMLList

stb.

Szintaxis

- ▶ nyelvi elemek (JavaScript-hez hasonló): lásd ActionScript 3.0 referencia → *Statements, Keywords & Directives*
- ▶ tömb, objektum esetén az értékadás történhet JSON-hoz hasonló szintaxissal

függvénydeklaráció:

```
function funcName([param0, param1, ...paramN]) : retType{  
    // kód  
}
```

Adatkapcsolás (binding)

implementálása (a háttérben) az eseménykezelő rendszer segítségével történik

Binding pl.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:s="library://ns.adobe.com/flex/spark">
  <fx:Script>
    <![CDATA[
      [Bindable]
      public var s:String="";
    ]]>
  </fx:Script>
  <s:TextInput id="text1"/>
  <s:Button label="Update" click="s=text1.text" x="130" />
  <s:Label x="0" y="30" text="{s}" />
</s:Application>
```

A kód eredménye:



Osztályok/objektumok

lehetőségek:

- ▶ osztály deklarációja (class kulcsszó)
- ▶ mezők, metódusok hatóköre: public, protected, private, internal, static
- ▶ osztály hozzáférés módosító: public, internal, (dynamic, final)
- ▶ (egyszeres) öröklődés
- ▶ absztrakt osztály
- ▶ interfész

Felhasználói felület tervezése

– tipikusan MXML segítségével történik

komponens készletek:

- ▶ Halo (MX) – régebbi (Flex 3)
- ▶ Spark – újabb (Flex 4)

komponensek elrendezése a konténeren belül (Spark) – Layouts

- ▶ BasicLayout (absolute layout), HorizontalLayout, VerticalLayout, TileLayout

Konténerek:

- ▶ Spark: Application, Group, SkinnableContainer, Panel, DataGroup, SkinnableDataContainer
- ▶ MX (Spark megfelelő nélkül): form, grid

kontrol-ok:

- ▶ egyszerű form-kontrollok
- ▶ rich text editor, csúszka, számláló, dátum-kiválasztó, színekiválasztó
- ▶ adatvezérelt kontrollok (dataProvider objektum segítségével megadott tartalom) – listák, rácsok, faszerkezet
- ▶ navigációs komponensek (menü, menüsor, gombsor, fülek (TabNavigator), nézetverem (ViewStack), harmonika (Accordion))
- ▶ diagramkészítés

előugró ablakok (pop-up windows):

- ▶ PopUpManager osztály segítségével
 - ▶ különböző konténerek szolgálhatnak alapul
-
- ▶ drag-and-drop lehetőség (DragManager osztály)

Adathelyesség ellenőrzése, adatformázás

- ▶ Adathelyesség ellenőrzők: StringValidator, NumberValidator, DateValidator, EmailValidator, RegExpValidator, stb.
- ▶ Adatformázók: NumberFormatter, DateFormatter, stb.

Állapotok (States)

- ▶ egy RIA felhasználói felülete több nézetből állhat
- ▶ egy nézetnek különböző *állapotai* lehetnek

Események

A Flex **eseményorientált** keretrendszer
(a különböző komponensek/osztályok eseményeken keresztül kommunikálhatnak egymással)

- ▶ az eseménykezelő paraméterként megkap egy ún. *eseményobjektumot*
(információ az esemény eredetéről, újrafelhasználható kód)
- ▶ esemény kézbesítés folyamata – ki(k)/ milyen sorrendben értesülhet(nek) az eseményről
- ▶ lehetőség saját eseményobjektum definiálására

eseménykezelés MXML, illetve ActionScript eseménykezelő segítségével:

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:s="library://ns.adobe.com/flex/spark">
  <fx:Script>
    <![CDATA[
      import mx.controls.Alert;
      protected function onClick(event:Event):void{
        Alert.show(event.target.label + " clicked");
      }
    ]]>
  </fx:Script>
  <s:Button label="Button 1" click="onClick(event)" />
</s:Application>
```

Pl. click esemény kezelésére ActionScript segítségével:

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:s="library://ns.adobe.com/flex/spark"
  applicationComplete="init()">
  <fx:Script>
  <![CDATA[
    import mx.controls.Alert;
    protected function init():void{
      button1.addEventListener(MouseEvent.CLICK, onClick);
    }
    protected function onClick(event:Event):void{
      Alert.show(event.target.label + " clicked");
    }
  ]]>
  </fx:Script>
  <s:Button id="button1" label="Button 1" />
</s:Application>
```

Szerver oldali szolgáltatásokkal való kommunikáció

- ▶ a Flex kommunikálhat különböző szerver oldali technológiákkal: PHP, Java, ASP.NET, ColdFusion, stb.

kommunikációs protokollok:

- ▶ HTTP
- ▶ **AMF** (Action Message Format) – bináris adatcsere
 - ▶ AMF+PHP: Zend_AMF modul
 - ▶ AMF+Java: BlazeDS, LiveCycle Data Services
- ▶ SOAP/WSDL – webszolgáltatással (WebService) való kommunikáció

Flex alkalmazás felépítése

A Flex keretrendszer MVC-elvekre alapoz

Mikroarchitektúra:

keretrendszer, mely bizonyos tervezési mintákat alkalmaz, illetve az alkalmazás szerkezetére vonatkozó szabályok betartására ösztönöz – használata hatékonyabbá teheti komplex Flex alkalmazások fejlesztését

Mikroarchitektúrák

Első generációs mikroarchitektúrák:

- ▶ PureMVC
- ▶ Cairngorm

Második generációs mikroarchitektúrák:

- ▶ Swiz
- ▶ Mate
- ▶ Robotlegs

Hasznos hivatkozások

– Adobe Flex hivatalos oldala:

<http://www.adobe.com/products/flex/>

– ActionScript 3.0 Referencia:

http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/

Dokumentáció, példák, video-tutorialok:

- ▶ Flex and Flash Builder Documentation

<http://www.adobe.com/devnet/flex/documentation.html>

- ▶ Adobe Flex 4

http://help.adobe.com/en_US/flex/using/index.html

- ▶ **Flex Test Drive** - Build an application in an hour

<http://www.adobe.com/devnet/flex/testdrive.html>

- ▶ Flex in a Week video training

<http://www.adobe.com/devnet/flex/videotraining.html>