Vitéz Gáborné - Drótos László - Moldován István - Simon Jennifer
(Vitéz Bt. - Országos Széchényi Könyvtár E-könyvtári Szolgáltatások Osztály)

DKA Képalbum

Egy webkettes alkalmazás fejlesztése QxTransformer/qooxdoo eszközökkel

1. Előzmény

Az Országos Széchényi Könyvtár Magyar Elektronikus Könyvtár műhelyében több mint 18 éve folyik a digitális könyvek és az időszaki kiadványok, újabban pedig a képek gyűjtése és feldolgozása. A könyvek száma ma már több mint 10 ezer, a periodikáké mintegy 2170 (ebből 360 kiadvány helyben archivált), a képeké pedig meghaladja a 30 ezret. A dokumentumok megnézhetők, letölthetők, kereshetők. Ám eddig nem készült ezekhez a rendszerekhez olyan felhasználóbarát felület, ahol a látogatók a kedvenc anyagaikat összeválogathatnák, rendezgethetnék, megjegyzéssel láthatnák el...

2. MEK műhely ötlet

2010-ben megfogalmazódott egy korszerű, személyre szabható, a népszerű web 2.0-ás könyves és tartalommegosztó oldalakhoz hasonló funkciókat (pl. címkézés, értékelés, saját gyűjtemények összeállítása) nyújtó felület igénye. Elsőként, kísérleti jelleggel a Digitális Képarchívumhoz készült el egy ilyen, *Képalbum* nevű alkalmazás, mellyel a felhasználók - Facebook vagy Google azonosítójukkal belépve - képválogatásokat alakíthatnak ki és oszthatnak meg másokkal. A rendszervázlat a következő elvárásokat tartalmazta:

„A Képalbum a Digitális Képarchívum egy kibővítése lesz, amely alternatív, személyre szabható felülettel, web 2.0-s funkciókkal, és szórakoztató jellegű elemekkel egészíti ki a szolgáltatást, a meglevő fájlrendszerre és a MySQL adatbázisra ráépülve. A dka.oszk.hu kezdőlapról egy külön menüpontból megnyíló ablakban a felhasználók a népszerű fotómegosztó oldalakhoz hasonló felületen böngészhetik a gyűjteményt és válogathatják albumokba a számukra érdekes képeket. (Lásd pl. a British Library képarchívumának Lightbox funkcióját). A fejlesztés másodlagos célja olyan ingyenes fejlesztő eszközök kipróbálása és megtanulása, amelyekkel azután majd a MEK és az EPA számára is lehet hasonló felületeket (pl. *Olvasósarok*) készíteni.

A Képalbum lehetséges funkciói:

- Felhasználói azonosítás: a felhasználók a meglevő Google, Facebook stb. azonosítójukkal tudnak saját területet használni a szerveren, ahol a beállításaikat a rendszer eltárolja. Vagyis nincs szükség külön regisztrációra, sőt akár azonosítás nélkül is lehet használni ezt a funkciót, csak ekkor nem őrződnek meg a beállítások.
- A felület külalakja a modern designt tükrözi: majdnem fehér háttér, pasztell színek és színátmenetek, apró grafikus elemek, kevés szöveg. Többféle skin is választható, és egyes elemek mozgathatók, vagy ki-be kapcsolhatók. Megvalósítás az objektum orientált JavaScript-, ill. XML-alapú, open source qooxdoo/QxTransformer fejlesztő eszközökkel.

- Ezen a felületen a felhasználó részgyűjtemény, témakör/altémakör, valamint népszerűség szerint böngészheti a gyűjteményt, továbbá a gyorskeresőt és a teljes szövegű keresőt használhatja. A találati listák mátrix-szerűen, a jelenleginél nagyobb thumbnail-eket mutatva jelennek meg és lapozhatók. A listákból kijelölhetők az érdekes képek (egyszerre több is) és ezekből albumok hozhatók létre. Az albumnak neve, létrehozási dátuma, címlapképe és rövid leírása lehet. Az albumok URL címe megosztható a népszerű közösségi oldalakon (az addthis.com megosztás aggregátorral), elküldhető e-mailben, továbbá a teljes album lementhető egy zip csomagban offline böngészéshez, készíthető belőle online diavetítés, és persze oda-vissza végig is lapozható. Az albumba tett képek rendezhetők (cím, méret, felkerülési dátum, népszerűség, vagy a felhasználó saját sorrendje szerint), törölhetők vagy átmásolhatók másik albumba is. És természetesen lehetőség van az albumok törlésére is (egy biztonsági rákérdezés után).
- A találati listából vagy az albumokból kiválasztott egyes képek nagyíthatók, forgathatók, sötétíthetők/világosíthatók, ajánlhatók/megoszthatók, könyvjelzőzhetők, címkézhetők és kommentelhetők (csak saját célra), pontozhatók (ezt mások is láthatják majd a Sikerlistában), menthetők, nyomtathatók, elküldhetők személyre szabott képeslapként, interaktív játékokba emelhetők át...
- A képek mellett alapesetben csak a legfontosabb metaadatok láthatók, de természetesen kattintással vagy az egér megfelelő helyre vitelével kétféle további részletességű leírás is megtekinthető. Egyes metaadatokra (pl. tárgyszó) kattintva új keresések indíthatók, egyes kitüntetett adatoknál pedig külső alkalmazások is meghívhatók (pl. földrajzi tárgyszónál a GoogleMaps, személyneveknél a Wikipédia). A felhasználó kiválaszthat egyes tárgyszavakat vagy altémaköröket, amelyekről a továbbiakban RSS csatornán kap értesítést, illetve a legközelebbi belépésekor automatikusan megjelenik számára egy találati lista az adott témában azóta felkerült képekről.”

3. Eszközök

Áttekintés

A rendszerterv készítése közben dönteni kellett a szoftvereszközökről is. A meglevő szolgáltatásaink PHP és MySQL környezetben működnek. A MySQL-t továbbra is használni akartuk az adatkezeléshez, de a felhasználói felületet már nem a megszokott PHP+HTML oldalakként képzeltük el, hanem valami szebbre, rugalmasabbra szerettük volna cserélni. Rendszergazdánk, Vitéz Gábor a JavaScript-alapú qooxdoo, open source web application framework-re épülő QxTransformer nevű eszközt javasolta.

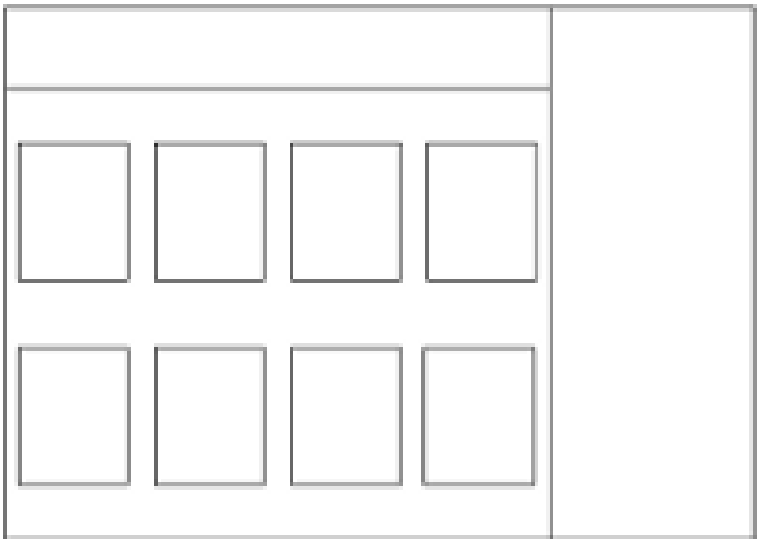
A *QxTransformer* egy platformfüggetlen, Python-alapokon működő, gyors alkalmazásfejlesztő eszköz, qooxdoo keretrendszerrel. Eszközkészlete XML szintaxist használ. Előnyei közé sorolják az átlátható kódot, az önellenőrzést, az adatkapcsolhatóságot (bind), a hordozhatóságot, a kiterjeszthetőséget...

A *qooxdoo* pedig egy erős, flexibilis framework, mellyel szép, interaktív webalapú GUI-t tudunk építeni. A qooxdoo nemcsak szebb felületet biztosít, mint a HTML, hanem lehetőséget nyújt a desktop operációs rendszerekből ismer widget-ek használatára, és megszabadít a hagyományos és fáradságos „input, submit & page refresh” modelltől.

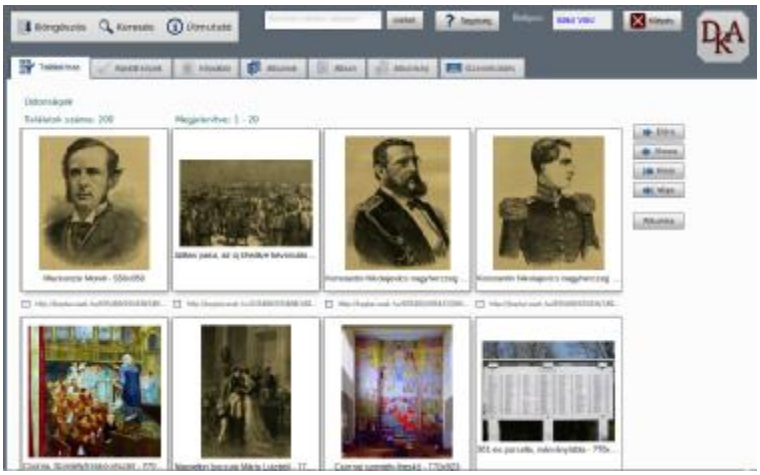
Számos előnye ellenére nem lehetett csak QxTransformerrel megvalósítani a rendszert, szükségessé vált a JavaScript-alapú qooxdoo használata is. Mivel egymásra épülő eszközökről van szó, ezért lehetett vegyíteni a két megoldást.

Elkezdődött tehát a Képalbum fejlesztése a QxTransformerrel, majd fordulatot vett a qooxdoo irányába.

Honnan, hová?



A képernyőterv ...



... és a megvalósult Képalbum

(Az képernyőterven látszik, hogy milyen méretű képecskékkel képzeltük el a találati oldalt. 180x180 pixel körüli érték volt a terv. Mivel ilyen méretű képek nem álltak rendelkezésre, készíteni kellett több ezer albumképet, lehetőleg parancssoros programmal. A *convert* Linux paranccsal ezt meg lehetett oldani. De ez csak egy kis kitérő volt.)

A továbbiakban a fejlesztés útját követjük végig az első lépésektől a megvalósult rendszerig.

3.1 QxTransformer

Mint oly sok programnyelvnél, itt sem hagyhatjuk ki a jól ismert „Hello World!” programot. Első lépésként máris egy Button widgettel találkozunk.

A ***Button*** widget rendelkezik néhány tulajdonsággal: *label*, *icon*, pozícionális paraméterek: *qxt:left*, *qxt:top*, és *listener* tartozik hozzá.

A listener az *execute* eseményre indul. Az akció a „Hello World!” feliratú *alert* üzenet feldobása.

Ebből máris sejthető, a widget leírása XML-ben elég egyszerű: *qx:button* a widget neve, tulajdonságai és azok értékei a név után sorolhatók fel.

```
<qx:button label="First Label" icon="helloworld/test.png" qxt:left="100"
qxt:top="50"> ....
```

A kiváltó esemény pedig a listener *type* tulajdonságával írható le.

```
<qxt:listener type="execute">
       <![CDATA[
             alert("Hello World!")
      ]]>
</qxt:listener>
```

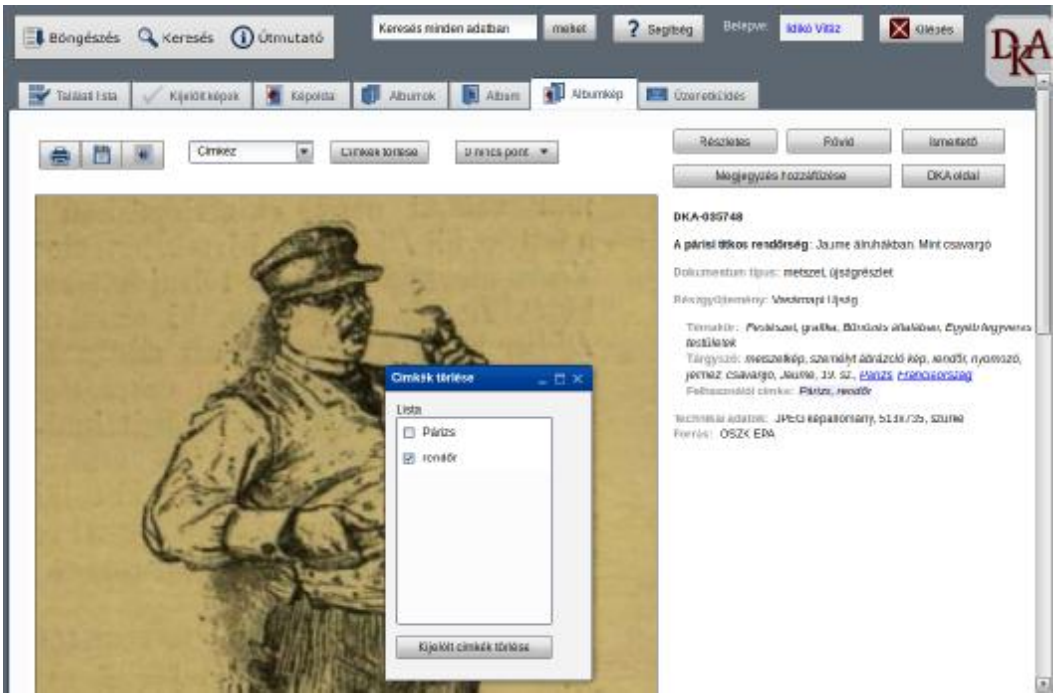
Ezzel már rá is tértünk a QxTransformer/qooxdoo alapvető építőblokkjára, a ***widget***-re. Számos widget áll rendelkezésre a qooxdoo-ban, ezek nagy része a QxTransformerből is elérhető. A widget-eket *Core*, *Content*, *Container*, *Building blocks*, *Popups*, *Embed*, *Form*, *Virtual* és *Indicators* tulajdonságaik alapján csoportosíthatjuk.

Alapvető Core widget a *Widget class*, melynek a többi widget az alosztálya (subclass). A Content widget-ek közül alap widget-eknek tekinthetők a *Label*, az *Image* és az *Atom*. A Container widget-ek közül a *Composite* kihagyhatatlan.

A Building block-ok már komolyabb építőelemek, a *Tabview* és a *MenuBar* ebben a kis mintaalkalmazásban látható: http://mek.niif.hu/~qxd/helloworld/build

A Popups-ok közül a *PopUp* és a *ToolTip* nagyon jól használható, ezek a Képalbum alkalmazásnak kiemelt elemei. Az Embed widgetek közül a *HTMLEmbed* alkalmas HTML kód megjelenítésére. A Form widgetek a HTML-ből is jól ismert funkciókat látják el, csak jóval árnyaltabban és változatosabban.

A Virtual widgetek összetett építőelemek. Erre is láthatunk példát a Képalbumban egy checkbox-szal kombinált lista megjelenítésénél.



Felhasználói címkék törlésére szolgáló checkbox lista

A widgetek főbb tulajdonságai:

1. integráció az eseményrendszerrel
2. fókusz kezelés
3. „drag and drop”
4. automatikus méretezés
5. kinézet (theming)
6. tooltip
7. context menu
8. láthatóság kezelés
9. sub widget kezelés

A widgetek legalább három HTML elemből állnak. Egy tartalmazó elemből, amellyel a szülő widgethez kötődnek, és két gyermek elemből: ezek a dekoráció és a tartalom.

Nézzünk meg részletesebben néhány widgetet!

Az egyik alap-widget a ***Label***. Fontosabb tulajdonságai:

1. *value*: értéke a megjelenő szöveg
2. *selectable*: kijelölhető és másolható (copy, paste)
3. *nativ context menu*: megjelenhet-e a jobb egérgomb menü
4. *rich*: egyszerű text a megjelenítendő szöveg, vagy HTML

Egy másik alap-widget az ***Image***. Fontosabb tulajdonságai:

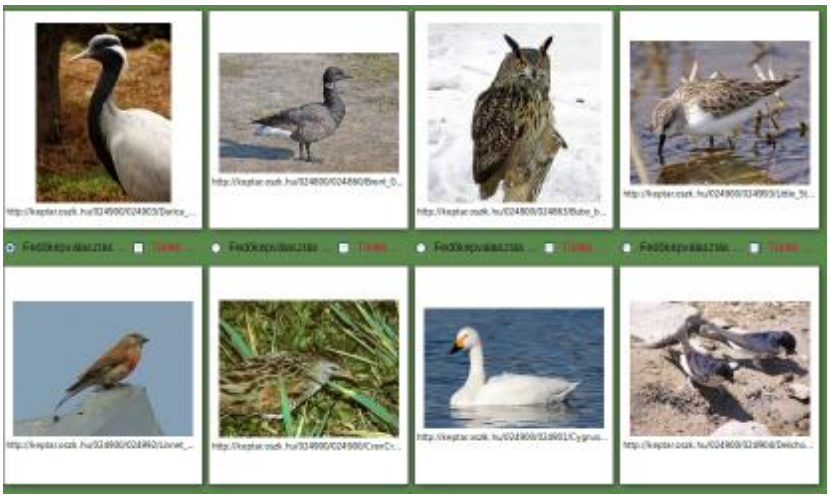
1. *allowGrowX*: növelhető-e az elem horizontálisan
2. *AllowGrowY*: növelhető-e az elem vertikálisan
3. *AllowShrinkX*: csökkenhet-e az elem horizontálisan
4. *AllowShrinkY*: csökkenhet-e az elem vertikálisan
5. *scale*: skálázható-e az elem
6. *source*: a kép URL-je

Az ***Atom*** widget egy Image-t kombinál egy Label-lel. Tulajdonságai közül párat emeljünk ki:

1. *icon*: egy URI, melyet az Image támogat
2. *iconPosition*: az ikon pozíciója a text-hez viszonyítva
3. *label*: a widget-en megjeleníthető szöveg
4. *rich*: kapcsoló a HTML és text tartalom között
5. *show*: a láthatóságot állítja be

Ahhoz, hogy strukturáltan helyezhessünk el widgeteket az alkalmazásunkban, ***Layout***-okat kell választani. Ezt úgy lehet megvalósítani QxTransformer/qooxdoo-ban, hogy választunk először egy Container elemet. Gyakorta a Composite-ot szokták erre használni. A Composite kezeli a gyerek widgeteket, és kezeli a struktúra létrehozását a *Layout manager*-en keresztül. QxTransformerrel ez is egyszerű: kiválasztjuk a Container widgetet, ezt követi a Layout tag, majd fel kell sorolni a gyerek widgeteket, melyek a Layout által megadott struktúrában fognak elhelyezkedni.

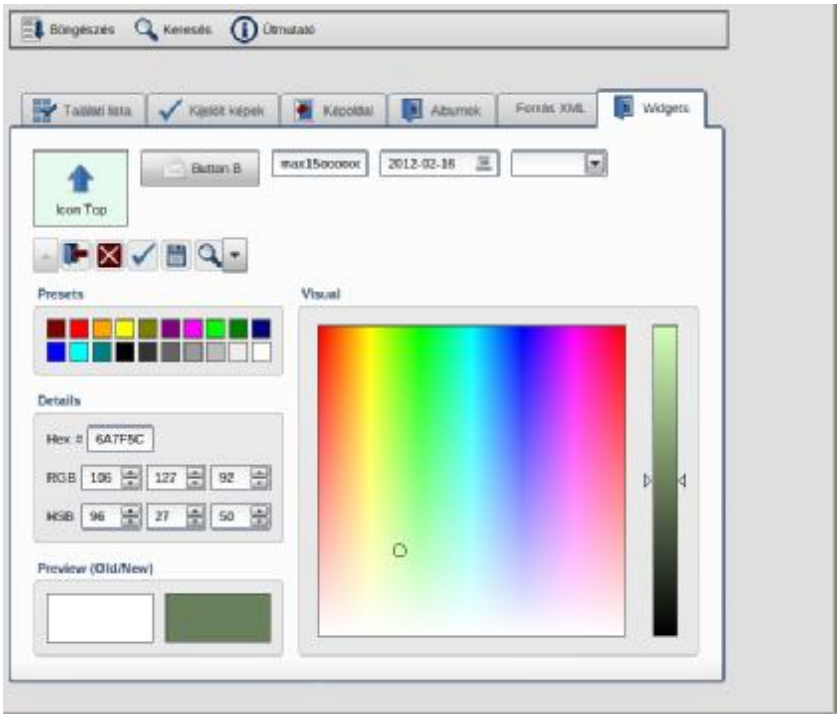
A Layout készlet elég bőséges, párat kiemelek a fontosabbak közül: *Basic*, *Canvas*, *Dock*, *Grid*, *Hbox*, *Vbox*. A Képalbum alkalmazásban a Hbox, Vbox és Grid a leggyakrabban használtak.



Grid layout a Képalbumban

A fent említett kis mintaalkalmazásban pedig Atom, Button, TextField, DateField, ComboBox widget-ek láthatók egy Hbox Layout-ra feltéve. Ezek többnyire a HTML oldalakról is ismertek, de mint láttuk, a tulajdonságaikat tekintve gazdagabbak. A DateField egy kis kalendárium, a ComboBox pedig a HTML select tagjához hasonlatos, ám input mezőt is kínál.

Jól használhatók az összetett widgetek. A SlideBar például hosszabb képsor megjelenítésére kiválóan alkalmas, de a ColorSelector is hasznos, ha a felhasználónak kinézet-módosítási lehetőséget szeretnénk nyújtani, azaz pl. átfestheti az alkalmazás hátterét a színkiválasztás segítségével. (Ezek a mintaalkalmazás középső részén láthatók.)



Mintaalkalmazás különféle widget-ekkel

A kis mintaalkalmazás struktúráját két összetett widget határozza meg. Fent egy többszintű menü valósul meg menuBar widgettel, alatta pedig tabView-t láthatunk.

Itt újra a QxTransformerrel megfogalmazva, egy jól strukturált, átlátható XML kóddal állíthatunk elő összetett, többszintű menüt - meglepően gyorsan:

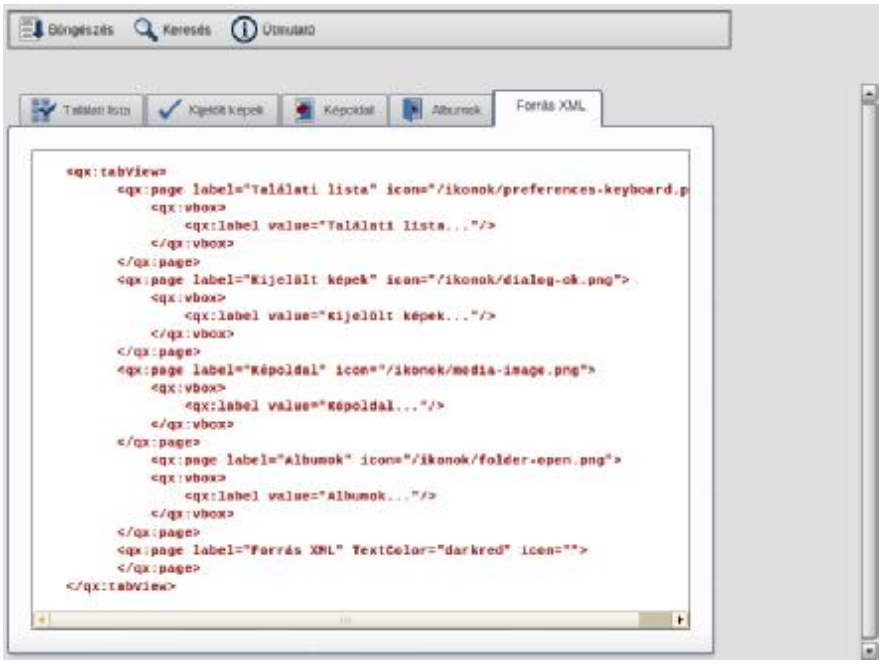
```
<qx:menuBar width="600">
 <qx:menuBarButton label="Böngészés" icon="/ikonok/view-sort-ascending.png">
  <qx:menu>
    <qx:menuButton label="Újdonságok" icon="">
    </qx:menuButton>
    <qx:menuButton label="Saját cimkék" icon="icon/16/actions/document-
new.png">
    </qx:menuButton>
    <qx:menuButton label="Részgyűjtmények" icon="icon/16/actions/document-
new.png">
     <qx:menu>
       <qx:menuButton label="A magyar hajózás" icon="icon/16/actions/document-
new.png"/>
       <qx:menuButton label="Ásványvilág" icon="icon/16/actions/document-
new.png"/>
     </qx:menu>
    </qx:menuButton>
  </qx:menu>
 </qx:menuBarButton>
....
</qx:menuBar>
```

A tabView-val pedig szinte megtöbbszörözhetjük a felületet, annyi *page*-t használva, amennyire szükségünk van. Mindkét elem a Képalbumnak is alapvető szerkezeti eleme.

Nézzük meg a tabView-t QxTransformer-rel megfogalmazva! Az egyik page-t felhasználtam arra, hogy egy Embed widgetet bemutassak.

A HtmlEmbed lehetővé teszi tetszőleges HTML kód megjelenítését, támogatja a CSS-t, beállítható, hogy a fókusz megjelenjen-e vizuálisan, és kontrollálható a szelektálhatóság is. Továbbá kezeli a szöveg túlcsordulását.

A tabView widget XML forrását HTML *Character Entities*-zel jelenítem meg egy htmlEmbed widget segítségével, így:



A tabView widget

Az HtmlEmbed pedig QxTransformerrel leírva:

```
<qx:embedHtml overflow="{js}'auto','auto'" decorator="main"
backgroundColor="white" width="900" height="400">
              <qxt:property name="html">
                        <![CDATA[
                                  <pre><b>
&lt;qx:tabView&gt;<br>
&lt;qx:page label="Találati lista" icon="/ikonok/preferences-
keyboard.png"&gt;<br>
&lt;qx:vbox&gt;<br>
&lt;qx:label value="Találati lista..."/&gt;<br>
&lt;/qx:vbox&gt;<br>
&lt;/qx:page&gt;<br>
....
&lt;/qx:tabView&gt;<br>
                                  </b></pre>
                        ]]>
                </qxt:property>
</qx:embedHtml>
```

Események, scriptek

Eddig a widgetek külső tulajdonságait tekintettük át madártávlatból. Vessünk pár pillantást az eseménykezelésre és a scriptekre.

A QxTransformer lehetőséget nyújt arra, hogy bárhol, egy-egy widget leírása után (közben) megnyissunk egy scriptet.

Dátumválasztó widget

A dateField widget létrehozása után szükségünk van néhány beállításra, pl. aznapi dátum, dátum formátum. Ezt le lehet írni `<qxt:script>` és `</qxt:script>` között. Ezek a sorok végrehajtódnak a applikáció betöltésekor.

```
<qx:dateField id="kezbesites_ido" width="100" maxWidth="100" height="22"
MaxHeight="22">
         <qxt:script>
              <![CDATA[
                qx.locale.Manager.getInstance().setLocale("en");
                kezbesites_ido.setValue(new Date());
                var format4 = new qx.util.format.DateFormat("yyyy-MM-dd");
                kezbesites_ido.setDateFormat(format4);
              ]]>
         </qxt:script>
</qx:dateField>
```

A Képalbumban a dateField mezőt használjuk az időzített képeslapküldés kézbesítési dátumának beállítására.

Hasonlóan kell megformázni az események hatására induló scripteket is.

Itt a *Button B* lenyomására indul a script, a `<qxt:listener type=...>` és `</qxt:listener>` tag-ek között kell leírni a kódot.

```
<qx:button label="Button B" id="gomb" icon="icon/22/apps/internet-mail.png"
enabled="true" width="100" maxWidth="100" height="22" MaxHeight="22">
        <qxt:listener type="execute">
                <![CDATA[
                    alert("ok");
                ]]>
        </qxt:listener>
</qx:button>
```

Általában a qooxdoo-ban leírt, az adott widget által kezelt esemény nevét kell feltüntetni a type= után.

Annyira sokszínű az eseménykezelés a qooxdoo-ban, hogy itt most szinte csak a felsorolásra szorítkozhatunk.

A *Core Widget* osztály eseményeit érdemes kiemelni, mert a többi widget ennek az alosztálya.

Core Widget event: *appear, blur, changeBackgroundColor, changeEnabled, changeFont, changeShadow, changeTextcolor, changeToolTipText, changeVisibility,* ***click****, contextmenu, dbclick, deactivate, disappear, drag, drop, focus, keydown,* ***keypress,*** *mousedown, move, resize, ...*

A Képalbum alkalmazásban leggyakrabban használt események:

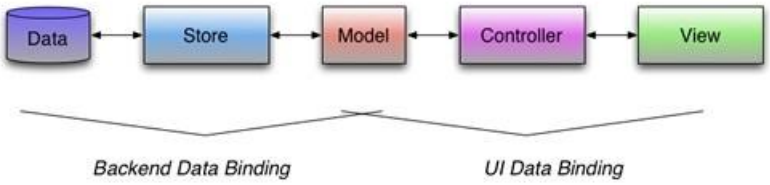
Button widget: ***execute***
List widget: ***changeSelection***

3.2 QxTransformer és qooxdoo együtt

Az eddigiekből is látható, hogy a QxTransformer jól használható, egyszerű eszköz, mégis az adott feladat esetén a background programokkal létrejövő kommunikációt már qooxdoo-val kellett megvalósítani. A dinamikus menü is qooxdoo kód beépítését igényelte.

a.) Nézzük, hogyan lehet használni a QxTransformert qooxdoo kóddal kiegészítve a kommunikáció esetén!

A kommunikációhoz először a qooxdoo adatkezelési sémáját kellett megérteni:



Data: a tárolt kiinduló adat

Store: adat kinyerés, elhelyezés a Model-be

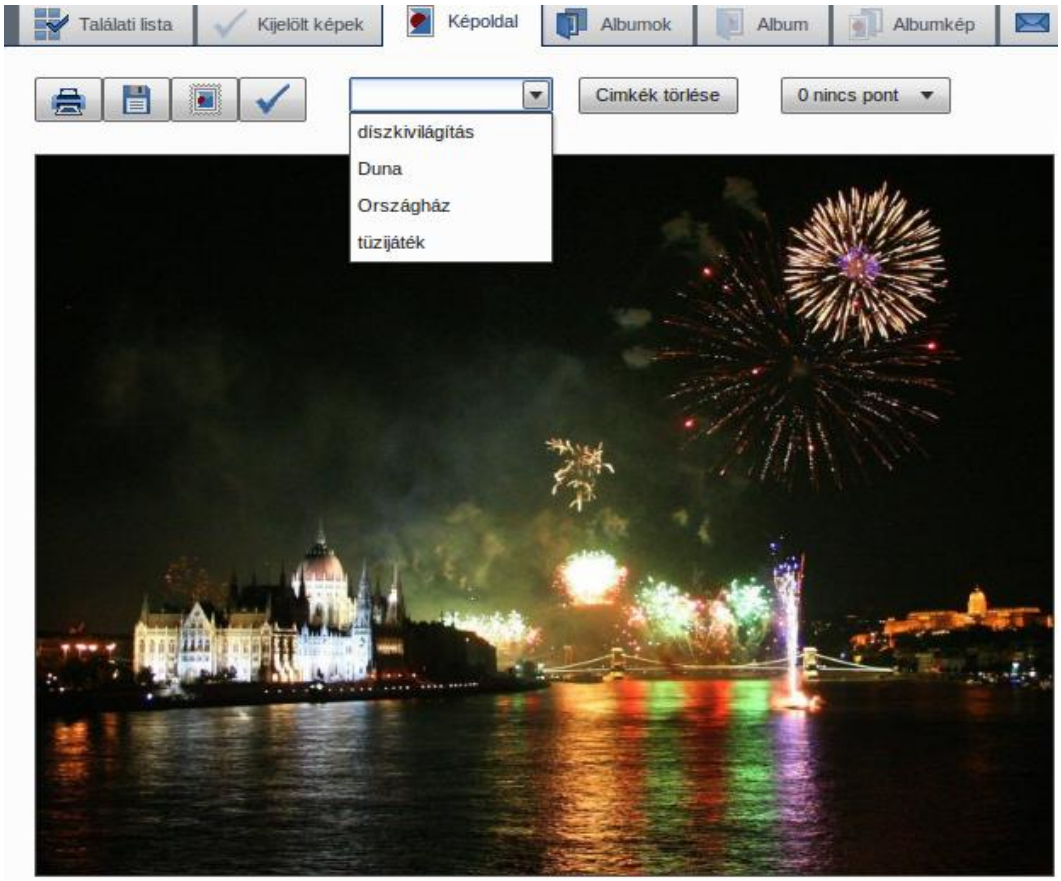
Model: a Store és a Controller integrációs pontja

Controller: összekapcsolja a Model-ben lévő adatot a View komponenssel

View: majdnem bármelyik widget lehet

A feladatban egy QxTransformerrel definiált ComboBox-ot adatbázisból kellett feltölteni a lista funkció ellátására. Ugyanakkor a ComboBox input elemét használva ez a lista dinamikusan növekedhetett. (A Képalbumban a ComboBox-ot a felhasználói címkézés megvalósítására használjuk.)

A meglevő címkéket adatbázisból vesszük (Data), majd egy listában tesszük választhatóvá őket (View), de közben az is követelmény volt, hogy ez lista újabb címkékkel bővülhessen. Ezért választottuk a ComboBox-ot, mert mindkét funkciót tudja nyújtani.



Címkézés ComboBox-szal

```
<qx:comboBox id="labelcombo" height="22" MaxHeight="22" marginLeft="25">
```

A Model megvalósítása ilyesmi lehet:

Data: a MySQL adatbázisból

Store: `mydatastore_l = new qx.data.store.Json("/"+progutvonal+"/cimkek.php");`

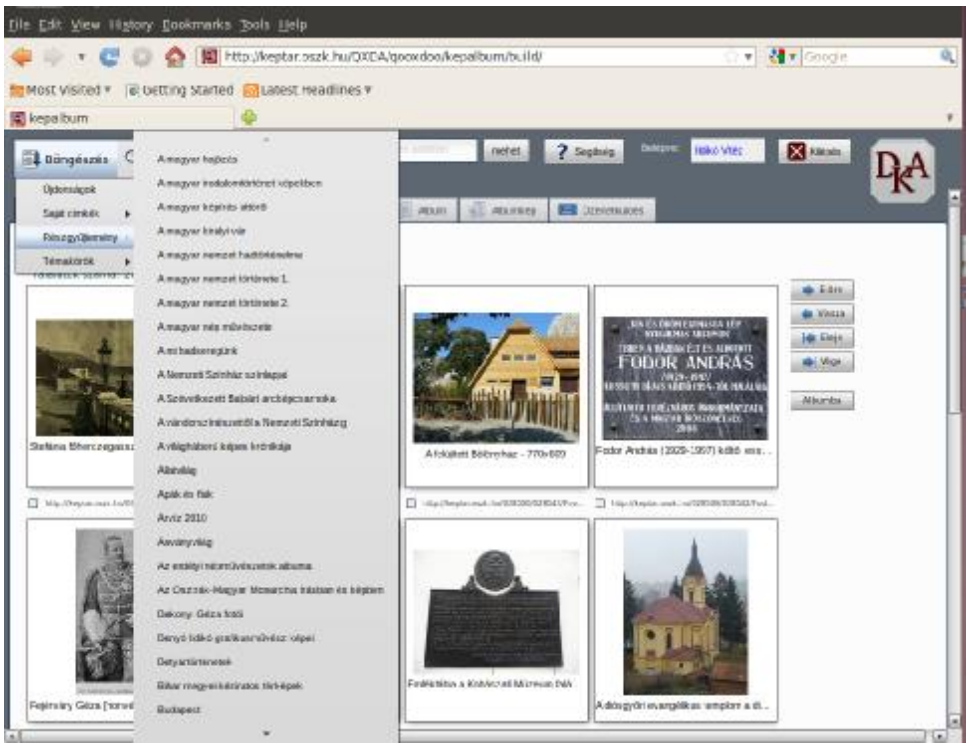
Controller:
```
var lcontroller = new qx.data.controller.List(null, labelcombo);
lcontroller.setLabelPath("nev");
```

View: `mydatastore_l.bind("model.cimkek", lcontroller, "model");`

b.) A másik példa egy dinamikus menü, amelyet ugyancsak qooxdoo kiegészítéssel kellett megoldani.

A qx:menu még QxTransformer XML kód, a scripten belül a Button-ok már qooxdoo objektumokként jönnek létre, majd a qx:menu-höz hozzáadjuk a Button-okat szintén qooxdoo metódussal.

```
<qx:menu id="subcollectionmenu" SpacingX="3" arrowColumnWidth="5">
  <qxt:script>
   <![CDATA[
    var gyujtemenyekszama=gyujtemenytomb.length;
    for(var j=0; j<gyujtemenyekszama/2; j++)
    {
        var cimke=gyujtemenytomb[j];
        var tooltipszoveg=gyujtemenytomb[j + gyujtemenyekszama/2];
        var gyuj = new qx.ui.menu.Button(cimke,"");
        gyuj.setBlockToolTip(false);
        var menutooltip = new qx.ui.tooltip.ToolTip(tooltipszoveg);
        subcollectionmenu.add(gyuj);
        gyuj.setToolTip(menutooltip);
        gyuj.addListener("execute",this.subcoll_);
        gyuj.addListener("execute",this.kepek_oldala);
    }
   ]]>
  </qxt:script>
</qx:menu>
```



Részgyűjtemény, dinamikus menü

4. Eredmény

Ideje rátérni a bemutatott eszközökkel megvalósított *Képalbum* alkalmazásra, melynek belépési pontja a dka.oszk.hu/html/kepalbum.php címen található.

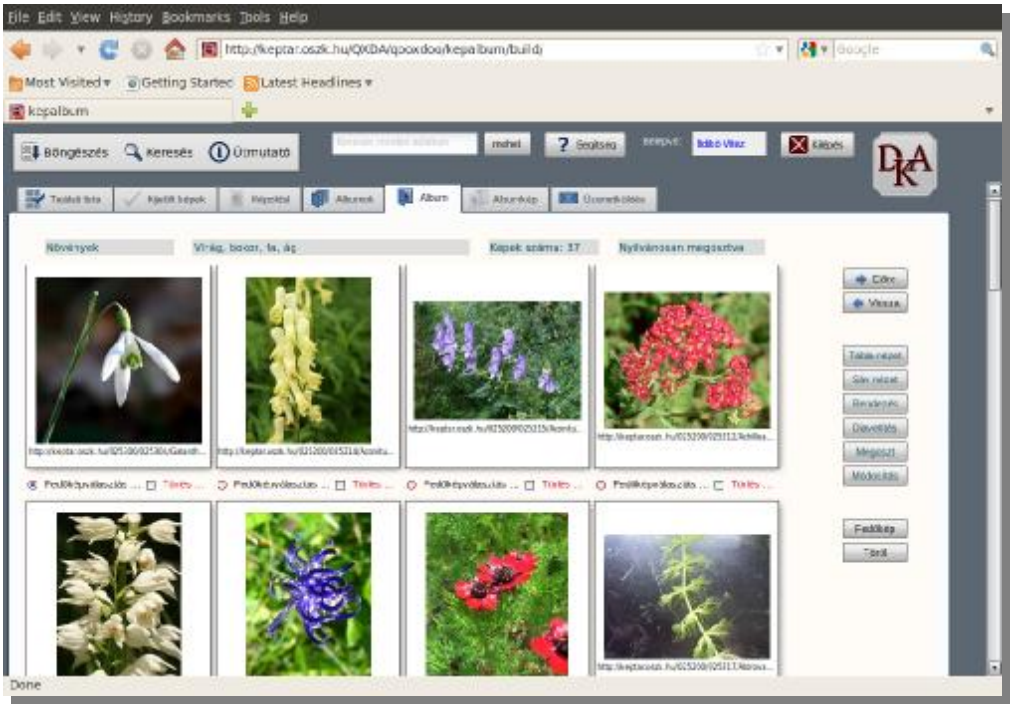
Az egyik cél az volt, hogy Google vagy Facebook azonosítással tudjon belépni a felhasználó a rendszerbe, vagyis ne kelljen külön regisztrálnia a Képalbum használatához. De ha nem szeretne élni a személyre szabható lehetőségekkel, akkor ehhez anonim hozzáférést is biztosítunk. Ez esetben is lehet használni a böngészési, keresési és egyéb funkciókat, ám a létrehozott albumok ilyenkor nem őrződnek meg.

A Képalbumba való belépéskor egy rövid ismertető fogadja a felhasználót, aki később helyzetérzékeny súgókat is előhívhat a ? *(Segítség)* gombra kattintva, illetve a teljes súgót is

átnézheti az *Útmutató* menüpont alatt. Mivel ennek a felületnek a használata lényegesen eltér a „közönséges” weblapok működésétől (pl. nem lehet visszalépni a böngésző nyílgombjával), ezért különösen fontosnak tartottuk a részletes használati utasítást.

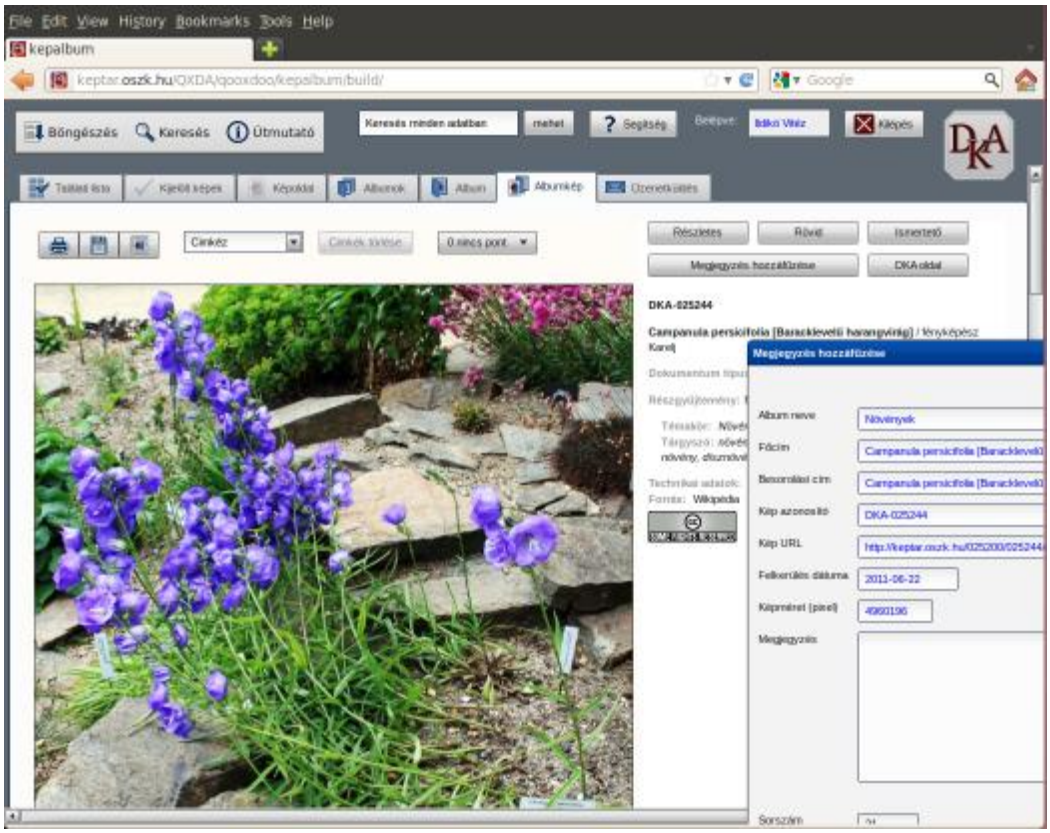
A legfrissebb képek listája automatikusan megjelenik, ez időben visszafelé lapozható (200 tételig). A felhasználó böngészni és keresni tud a gyűjteményben. A *böngészés* történhet részgyűjtemények, témakörök, valamint saját címkék szerint (amennyiben a felhasználó korábban már megcímkézte a képek egy részét). A *keresés*hez pedig mindig rendelkezésre áll egy egysoros gyorskereső a képernyő tetején, amely minden fontos adatban keres, de választhatunk egy háromsoros keresőűrlapot is, amivel a cím, az alkotó és a tárgyszó mezőkre tudunk összetett keresőkérdéseket megfogalmazni.

Akár böngészünk, akár keresünk, az eredményül kapott találati listában kipipálhatunk olyan tételeket, amelyek szeretnénk egy albumba gyűjteni. Ezek az *Albumba* gomb megnyomása után a *Kijelölt képek* nevű lapra kerülnek, ahol választhatunk a korábban létrehozott albumaink közül, vagy készíthetünk egy újat. Az albumoknak nevük, leírásuk, háttérszínük és fedőképük lehet (utóbbit a beletett képek közül választhatjuk ki a *Fedőkép* gombbal). Az albumok tartalma alapesetben táblázatos nézetben nyílik meg, vagyis mátrix formában elrendezett kisképekkel.



Egy album táblázatos nézetben

Ha valamelyik kisképre kattintunk, akkor az *Albumkép* nevű lapon megnézhetjük azt nagyobb méretben, sőt egy újabb kattintással még tovább nagyíthatjuk. Ugyanezen az oldalon jelennek meg a kép metaadatai is, kétféle részletességgel, a földrajzi tárgyszavak pedig a GoogleMaps segítségével térképen is megnézhetők. A felhasználó saját címkéket adhat a könyvtárosi leíráshoz (ezeket eltérő háttérszínnel jelzi a program), továbbá hosszabb szöveges megjegyzést is fűzhet a képhez, valamint értékelheti annak szépségét/fontosságát egy 1-től 10-ig terjedő skálán. A képet elmentheti, kinyomtathatja, vagy elküldheti valakinek képeslapként (utóbbi esetben különböző képességű levelezőkliensekhez optimalizált formázások közül választhat).



Egy kép metaadatokkal és megjegyzés-ablakkal

Az albumok tartalma egyéb nézetekben is megjeleníthető. A *Sáv nézet* esetében egy vízszintesen gördíthető listában jelennek meg a kisképek, az egérkurzor alatti tétel pedig automatikusan felnagyítódik. A *Rendezés* gombra kattintva egy hasonló listában „drag and drop” módszerrel (egérrel megfogva majd elengedve) átvariálhatjuk a képek sorrendjét. Egy további nézet a *Diavetítés*, itt egy új ablakban nézhetjük végig az album tartalmát, egy Javascript-es animációval. Az album „normál” weblap formájában ki is exportálható a qooxdoo alkalmazásból, vagyis nyilvánossá tehető és *megosztható* másokkal (nekik természetesen csak megtekintési joguk van). Reményeink szerint a Képalbum jól felhasználható lesz például az oktatásban: a tanárok vagy akár a diákok is egy-egy témához vagy eseményhez kötődő válogatásokat készíthetnek vele és publikálhatják őket különféle honlapokon, blogokban vagy közösségi oldalakon.

Az alkalmazás fejlesztése még nem zárult le, további funkciók beépítését is tervezzük, pl. keresőkérdések eltárolása és újrafuttatása, találati listák többszempontú rendezése, különféle képalapú játékok.