

Ajax webalkalmazások automatikus kódgenerálása

Önálló laboratórium feladat összefoglalója (2. félév)

Kasza Bálint (GXUV56)

Konzulens: Tóth Dániel

**BME Méréstechnika és Információs Rendszerek Tanszék
Informatikai infrastruktúra tervezése szakirány, 2006/2007. II. félév**

A választott téma feladata a tanszéken fejlesztett Viatra GUI szerkesztőjének segítségével történő Ajaxos webalkalmazások kódgenerálása. Egy webalkalmazás esetén mind a kliens, mind a szerver oldallal foglalkozni kell. A téma nagyobb, összetettebb részét a kliens oldal teszi ki. Az előző féléves terv szerint a kiszolgáló oldallal Szabó Marcell foglalkozott volna, azonban ő másik témát választott, ezért ezt a részt is én fogom megoldani. Ebben a félévben a kliens oldallal foglalkoztam, a szerver oldallal kapcsolatos teendők még kidolgozásra várnak egy későbbi munka során.

A feladathoz adott volt a keretrendszer, amiben dolgozni kellett: ez a Viatra nevű Eclipse plugin. Ennek része a konzulensem által fejlesztett UiPIM, azaz egy platform-független GUI-szerkesztő. Ezzel a szerkesztővel felhasználói felületeket hozhatunk létre. Az így létrehozott felületek nincsenek platformhoz kötve, az én feladatomból az, hogy ezeket a felületeket leképezem a Webes környezetes megfelelőjükre. A megvalósítás során a modell vezérelt fejlesztési elvet követtem: az UiPIM-el létrehozott felületek gyakorlatilag (platformfüggetlen) modelleket jelentenek. A Viatra keretrendszerrel transzformáltam át a Webes platformot leíró modellre, s végül ebből a platformspecifikus modellből (PSM) generáltam a kódot.

A feladat első lépése a fentebb leírt platformfüggetlen modellből platformspecifikus modellre történő leképezés. Ehhez létre kellett hozni ennek a PSM-nek az ún. metamodelljét: az egyes PSM modellek leíróját. Ennek kialakítását előző félévben elkezdtem, azonban ezen alakítani kellett miután jobban megismertem a keretrendszert. A következő lépés a platformfüggetlen modell egyes elemeit leképezni a PSM-re. Ezt a feladatot szétválasztottam több részre, lévén, hogy a Webes környezet több különböző részből áll. A struktúrát a HTML szerkezet jelenti. Az UiPIM egyes elemeit leképeztem HTML elemekre. Az összetettebb UiPIM elemek leképezése nem volt triviális, mert nem létezik nekik megfelelő HTML elem, ezért ezeket több kisebb elemből állítottam össze JavaScript függvények segítségével. A platformfüggetlen modell megjelenését szabványos módon alakíthatjuk. Ez a szabvány a CSS, s a Webes környezethez is erre van szükség, tehát a leképezés triviális lenne, így itt az UiPIM modelljét használtam a PSM modellben is. Az UiPIM elemek egyes attribútumait változókhoz köthetjük, ezt nevezzük adatkötésnek. Ezeket az adatkötéseket a Webes platform JavaScript (JS) részével oldottam meg. A platformfüggetlen modell változóit leképeztem JS változókra, míg az adatkötéseket különböző JS függvényekkel oldottam meg. Az UiPIM eseménykezelése az ún. signal-slot elven alapszik. Ezt úgy képeztem le a Webes környezetre, hogy az egyes signalokhoz eseménykezelő attribútumokat rendeltem a HTML elemekhez, míg magát az eseményt (slot) általam írt JavaScript függvények jelentik.

A feladat másik nagy része a feljebb leírt módon létrehozott platformspecifikus modellből történő kódgenerálás. A kódgenerátor külön fájlt készít a megjelenést szabályozó CSS-hez, s magát a lapot jelentő HTML-hez is. A működést szabályozó JavaScript rész a HTML kódba ágyazva generálódik.

A félév végére a kliens oldal kódgenerálását megoldottam, következő lépésben a szerver oldallal fogok foglalkozni.