

# **AJAX alapú DSM szerkesztő fejlesztése**

## **Önálló laboratórium feladat kiírása**

**Czotter Márk (U4F0H0)**

**Konzulens: Ráth István**

**BME Méréstechnika és Információs Rendszerek Tanszék  
Rendszertervezés ágazat, 2008/2009. II. félév**

A téma célja egy olyan webes alkalmazás készítése, melynek segítségével egy fejlesztő, vagy kliens könnyedén tud létrehozni, szerkeszteni domain specifikus modelleket. Mivel webes alkalmazás készítése a cél, mind kliens, mind szerver oldallal foglalkozni kell. A félév során mindkét oldallal foglalkoztam, de még mindig van teendő a tökéletesítésre.

A webes alkalmazás kliens oldala alapvetően egy böngészőben futó program, melynek segítségével a kijelölt modellt a felhasználó szabadon változtathatja. A megvalósításhoz szükség van egy a webes környezetben használható objektum-orientált nyelvre, hogy objektumokat tudjunk létrehozni. Ehhez használtam a Prototype JavaScript könyvtárat, melynek segítségével ez egyszerűen véghez vihető volt. Miután a kliens oldalon a megfelelő elemeket létrehoztam, a megjelenítés következett. Mivel a böngészők alapvetően szöveg alapú megjelenítők, ezért egyszerű HTML-ben nem lehet megvalósítani modellező szoftvert. Ehhez szükség volt egy új technológiára, melynek segítségével könnyedén lehet böngészőben rajzolófelületet biztosítani, ez a HTML 5.0 Canvas, mely kevés böngészőben elérhető, és nem teljes a megvalósításuk sem. Ennek megtanulása, és a megvalósítás után, az eseménykezelés és a kommunikáció következett. Az eseménykezelés megvalósítása a már ismert Listener modell alapján történt, tehát minden egyes létrehozott elemnek saját eseménykezelője van, így elsősorban a mozgatást, illetve létrehozást valósítottam meg. A kommunikációhoz természetesen AJAX kommunikációt használtam, melyet szintén a Prototype nyújtott. A kommunikáció JSON sztringek formájában valósul meg, az egyes létrehozott modelleket, JSON formátumba alakítva küldjük a szervernek, melyben a feldolgozás, létrehozás volt a következő nagy feladat.

A szerver a beérkező kérések feldolgozásáért, a modellek tárolásáért felelős. A megvalósításához szükség van egy szerverre. Mivel a fejlesztés e része Eclipse alatt történik, a választás az Eclipse alól is elérhető Jetty Server-re esett. E szerverben egy JAVA Servlet dolgozza fel a kéréseket, melyek a kliens felől érkező modellben való szerkesztési eseményeket tartalmazzák (Tulajdonképpen a modell egyes létrehozott elemei utaznak JSON formában). A kommunikáció szerver oldali része volt még kulcskérdés, hiszen többféle megvalósítás közül lehet választani. Az első megoldás egy ma sokszor használt, de sok hálózati forgalmat lebonyolító megoldás volt (Polling), de ez helyett a szerkesztőnek jobban megfelelő Long Polling technológiát használok. A tárolás első körben belső Java osztályokban valósult meg, így csak ezen a nyelven lehetne modellezni, ezért a félév végén az EMF integrációját kezdtem el, de ennek befejezése valószínűleg a következő időszakra marad.

Jelen pillanatban a szoftver kliens oldali része egy nagyon egyszerű csak modellek létrehozására alkalmas megvalósítás, erre az adott kliens oldali nyelvek illetve a szerver oldali technológiák tesztelése miatt volt szükség. Ez a rész egy jó kiinduló alap a következő fokozatra váltáshoz, melyben a szerkesztő egy komoly ráncfelvarráson esik át, hogy a komolyabb technológiákhoz használni lehessen. E technológiák lesznek elsősorban az EMF, illetve a GMF, melyeket a szerver oldalba ágyazva, könnyedén lehet generáltatni, adott domain specifikus nyelvekhez webes szerkesztőt.