Vasúti térközbiztosító rendszerek modellezése és szimulációja

Önálló laboratórium feladat összefoglalója

Ágoston István (FQ2J73)
Konzulens: Polgár Balázs
BME Méréstechnika és Információs Rendszerek Tanszék
Rendszertervezés ágazat, 2008/2009. II. félév

A feladat során vasúti rendszerek általános modelljét kellett elkészíteni, beleértve ebbe a pályaszakaszok felépítését, felügyeleti eszközeit és ezek jelzésrendszerét. Mivel ezt a témát 3-an végeztük, a feladatot 3 részre bontottuk:

1) A metamodell helyességének ellenőrzése
2) Grafikus szerkesztő készítése, mellyel a metamodellnek megfelelő vasúti modell készíthető
3) Szimulátor, mellyel egy konkrét modellt lehet „életre kelteni", működését vizsgálni

A félév első 8 hetében közösen létrehoztuk a vasúti rendszerek metamodelljét a *Papyrus* nevű UML diagram szerkesztővel (ez volt a kiindulási alap mindannyiunk munkájához), majd a fenti feladatot szétosztottuk egymás közt. Én a domain specifikus grafikus szerkesztő elkészítését kaptam. Ahhoz, hogy az editor *domainjeként* szereplő metamodell az editor számára is feldolgozható legyen az *EMF (Eclipse Modeling Framework)* Eclipse-es komponenst használtam fel. Ez lehetővé teszi, hogy UML-ben leírt modelleket könnyedén transzformáljunk *.ecore* fájllá (azaz EMF metamodellé), majd ebből legeneráljuk a megfelelő modell fájlokat (Java osztályokat). Ez nagyon sok dologban támogatást nyújt, a legfontosabbak:

- új modell objektumok létrehozása *Factory* metódusok segítségével
- a modell osztályok implementálják az *EMF Notification* interfészt, mely a modellen történő változtatások esetén értesíti a feliratkozott figyelőket (ez elengedhetetlen egy editornál)
- támogatja a modell a tárolását (*EMF* formátumú fájlba írását)

Az editor további részeihez a *GEF (Graphical Editing Framework)* Eclipse-es plugint használtam fel. Ez egy általános grafikus szerkesztő funkcióit megvalósító, *MVC (Model-View-Controller)* szemléletet követő Java osztálykönyvtár. *Model*-ként az előbbiekben ismertetett *EMF* osztályok szolgáltak, a *View* rétegben pedig létre hoztam az egyes modellelemek grafikus megfelelőit (A *GEF* a Java *Draw2D* osztályait támogatja). A szerkeszthető grafikus elemek, és az ezekhez tartozó modellobjektumok között egy-egy kapcsolatot kellet létrehozni. A *GEF*-ben erre a különböző *EditPart* osztályok használhatók, melyek implementációjában felhasználtam az *EMF* nyújtotta értesítéseket, ezen keresztül lehet követni a modell változásait és reagálni ezekre. Az editorhoz szükség volt még palettára, ahol kiválaszthatjuk, hogy milyen elemet szeretnénk létrehozni, ezt a *GEF* segítségével könnyen meg lehet oldani. Ezen felül már „csak" a *Command* osztályokat kellett elkészteni (ez az *MVC Controller* része). Minden felhasználói beavatkozásra külön *Command* osztályt kellet létrehozni: különböző elemek létrehozása, törlése, mozgatása. A munkát inkrementálisan végeztem, egyszerre mindig csak egy új elemet vettem fel, majd megvalósítottam az ehhez tartozó funkcionalitást. A fent leírtak egy Eclipse-pluginba lettek integrálva, tehát az egész szerkesztő egy önálló Eclipse plugin lett.