

# Állapotterkép alapú alkalmazásfejlesztő környezet

## Önálló laboratórium feladat összefoglalója (1. félév)

Sütő László (BFIAV1)

Konzulens: Pintér Gergely

**BME Méréstechnika és Információs Rendszerek Tanszék**  
**Informatikai infrastruktúra tervezése szakirány, 2005/2006. I. félév**

Az állapotterképek leírják egy eseményvezérelt komponens viselkedését. Bizonyos események állapotváltozásokat és akciókat válthatnak ki, amik akár újabb eseményeket is indukálhatnak. Ehhez természetesen szükséges, hogy a komponens viselkedése jól leírható állapotokból álljon. Az állapotok hierarchikusak, így lehetővé teszik, hogy alacsonyabb szinten specifikáljuk azok működését, nagyban megkönnyítve ezzel a modell átláthatóságát és későbbi módosíthatóságát. A rengeteg fajta állapotterkép közül én az UML2 állapotterképpel foglalkozom.

Ezekből a modellekből automatikusan generálhatóak konkrét implementációk egy adott nyelvre. Többféle ilyen általános generálási minta létezik, de ezek általában csak egyszintű állapotterképekre alkalmazhatóak. Az általam megvalósított kódgenerátor Miro Samek Practical Statecharts in C/C++ könyve alapján készül, amely támogatja a hierarchikus állapotterképeket és ezenkívül egy nagyon hatékony megvalósítása így akár beágyazott rendszerek tervezésében is felhasználható lenne.

A félév során először az irodalom egy részét tanulmányoztam át, kezdve az UML 2-es szabvány állapotterkép specifikációjával majd pedig Miro Samek könyvével. Eközben kutattam a felhasználható technológiák után is, melynek eredményeként az Eclipse-et választottam alapvető fejlesztő környezetnek. Ebben rendelkezésre áll sok-sok minden más között egy elég jó CDT elnevezésű C/C++ fejlesztő környezet, egy UML2 plugin, mely a legnaprakészebben valósítja meg az UML2-es modellek objektumstruktúrába való leképezését és XMI2 -es formátumban való tárolását. Ezen felül hasznomra vált egy az Eclipse-hez kifejlesztett kódgenerálást segítő leíró nyelv is a Java Emitting Templates (JET).

Tehát az Eclipse-hez készítettem el a kódgenerátor plugin-t, amelyet akár már meglévő Eclipse alapú modellező keretrendszerbe is könnyedén lehet integrálni pl.: Software Architect-be. A plugin képes arra, hogy egy UML2-es modellből kiválasztott állapotterképek C++ implementációját csomaghelyes könyvtárstruktúrába legenerálja. Az implementáció a már említett könyv alapján készült, mely az úgynevezett Behavior Inheritance tervezési mintát tartalmazza. Ezen minta nem fedi le ugyan a teljes állapotterkép funkcionalitást, de logikus közelítést ad a legtöbb esetben. Az általam elkészített implementáció nem támogatja az összes lehetőséget, ami az irodalomban szerepel, de már működő kódot generál. A működés kipróbálását egyszerűsítendő a plugin egy tesztprogramot is generál, melyben az összes esemény kipróbálható, és minden állapotba átvihető az állapotterkép. Természetesen az implementáció működéséhez szükséges egy alapvető keretrendszer ami az alapvető állapotterkép működést általánosan megvalósítja, amit a plugin által generált kód testre szab az aktuálisan megvalósítandó állapotterképnek megfelelően.

Felmerült továbbhaladási irányok a témával kapcsolatban újabb programozási nyelvek támogatása és természetesen az állapotterkép támogatott funkcióinak bővítése.