

# **Aszinkron protokollok verifikációja és validációja**

## **Önálló laboratórium feladat összefoglalója (1. félév)**

**Pintér Norbert (DEGA6C)**

**Konzulens: Bokor Péter**

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

A félév első harmadában az önálló labor keretében megismerkedtem az aszinkron rendszerek legfontosabb jellemzőivel, a konszenzus illetve az atomi üzenetszórás (atomic broadcast) problémával és ezek lehetséges megvalósításaival különböző típusú hibadetektorok alkalmazása esetén, crash típusú meghibásodásokat feltételezve az egyes processzekre. Láttam, hogy az eltérő hibadetektorosztályok esetén hány processz hibásodhat meg, hogy még meg lehessen oldani a fent említett két problémát, illetve hogy a különböző hibadetektor osztályok között milyen kapcsolatok állnak fent (ekvivalensek, erősebbek, ...).

A félév második harmadában megismerkedtem a Neko projekt protokoll validációs környezettel, amely java nyelven van implementálva és benne már rendelkezésre áll rengeteg előre implementált algoritmus (hibadetektorok, konszenzus, illetve atomic broadcast algoritmusok). Ezen program segítségével a java nyelven írt protokollokat könnyen lehet szimulálni, illetve tesztelni, valamint a program maga is könnyen bővíthető különböző algoritmusokkal. A környezetben a hibadetektorok, illetve az egyes problémák algoritmusai különálló modulokként vannak implementálva, ezért azok könnyen módosíthatók, lecserélhetők és könnyű megvalósítani az egyes modulok egymásra épülését is (pl. az egyes hibadetektorokra ráépülhetnek a fentiekben ismertetett problémák). Ezáltal ez egy nagyon kényelmes és hasznos szimulációs rendszer.

A félév utolsó harmadában pedig hasznos ismeretekre tettem szert a formális verifikáció területén. Tapasztalatokat szereztem a TLC nevezetű model-checker-ről. Ez a model-checker a TLA formális nyelven leírt protokollokon képes különféle vizsgálatokat elvégezni, mint például élőségvizsgálat. Ezt a formális nyelvet, illetve a model-checker-t Leslie Lamport dolgozta ki és implementálta. Majd mivel a TLA egy alacsony szintű programnyelv és elég bonyolult implementálni benne a különböző protokollokat, mivel a TLA nyelv eléggé matematika közeli nyelv, ezért sokkal könnyebben használható a +CAL magas szintű nyelv, melyet szintén Lamport dolgozott ki és ezek a +CAL nyelven megírt protokollok egy fordító segítségével könnyen átkonvertálhatók TLA nyelvben leírt protokollá! Ezért ezzel a magasszintű nyelvvel is megismerkedtem.

Ez a félév egy kis „sétát” jelentett az aszinkron protokollok és elosztott aszinkron rendszerek világába és azok szimulációjába és verifikációjába.