Diplomaterv-bírálat

Diplomaterv címe:	Komponensalapú reaktív rendszerek lépésenként vezérelhető szimulációja precíz formális szemantika szerint
Szerző:	Szkupien Péter
Szak:	Mérnök informatikus szak
Dátum:	Budapest, 2023. január 9

A diplomatervben kitűzött feladatok

A diplomaterv célja, hogy a jelölt fejlesszen egy eszközt állapottérképek precíz szemantikán alapuló szimulációjára. A munka fókuszában az összetett átmenetek és változók részletes megfigyelhetősége áll, ennek megfelelően az állapotkonfigurációk kiszámítását egy már létező (modellellenőrzőt használó) megoldás végzi, az elsődleges hozzájárulás az összetett átmenetek megfigyelhetővé tétele a köztes leírásmódként használt XSTS átmenetek előfeldolgozásával.

A diplomaterv szerkezete, felépítése

Az első fejezet hangsúlyozza a formális módszerek terjedését a beágyazott és biztonságkritikus rendszerekben és felhívja a figyelmet a szimuláció fontosságára a mindennapi gyakorlatban. A második fejezet a munka elméleti hátterét (modellellenőrzés fogalma, tesztelés és modellellenőrzés különbsége, véges állapotgépek formalizmusa, állapottérképek, nemdeterminizmus lehetőségei, Extended Symbolic Transition System (XSTS)) és a felhasznált eszközöket (Gamma Statechart Composition Framework és Theta Model Checking Framework) mutatja be. A harmadik fejezet a fejlesztendő szimulátor eszközzel szemben megfogalmazott követelményeket sorolja fel. A negyedik fejezet hordozza a munka elméleti hozzájárulását, azzal, hogy formálisan leírja az XSTS formalizmus átmeneteinek szükséges előfeldolgozását a szimuláció lehetővé tétele érdekében. Az ötödik fejezet a megvalósítás szerkezetét vázolja, majd a hatodik fejezet egy példát mutat a szimulátor használatára; noha az itt tárgyalt példa elméletileg értékes következtetések levonását teszi lehetővé, egy szimulációról szóló dolgozatban a túlzott egyszerűsége miatt nem tűnik szerencsésnek. A záró fejezet összefoglalja a dolgozatban tárgyaltakat és vázolja a további fejlesztési lehetőségeket.

A diplomaterv értékelése

A dolgozat témaválasztása időszerű, az azonosított megoldandó probléma valós. A dolgozat szerkezete logikus, formája példamutató, a formális részek tárgyalása precíz. A példák és ábrák száma megfelelő, jó minőségűek és segítik a megértést. A jelölt egyértelmű tanúbizonyságát tette annak, hogy képes megérteni nemtriviális formalizmusokat és érdemben hozzájárulni az elmélet bővítéséhez és az így formalizált újdonságokat meg is tudja valósítani modern technológiákkal. A dolgozat erényeinek hangsúlyozása után célszerű megemlíteni, hogy, noha a hatodik fejezetben bemutatott (egyetlen) példa elméleti szempontból érdekes és rávilágít a modellezési nyelvben (UML állapotgépek és aktivitási diagramok) máig fennálló specifikációs problémákra, az olvasó egy *esettanulmány* című fejezetben inkább egy *nemtriviális állapotgép* szimulációját várta volna, különösen, hogy a bevezető helyesen felhívta a figyelmet a modellek növekvő komplexitására és a szimuláció szerepére a biztonságkritikus rendszerek fejlesztése során. A dolgozat példája egy két állapotból álló lineáris vezérlésfolyam az állapotgép szempontjából, ami nehezen tekinthető bármilyen valós probléma illusztrációjának, a jövőben célszerűbben látszik életszerű problémák bemutatásával érvelni egy új formalizmus hasznossága mellett.

Kérdések a jelölthöz

1. A 2.2.3.1 szakasz *"Multiple fireable transitions: In real-life engineering work, especially with the growing complexity of state hierarchies and guard expressions, statecharts are usually not deterministic and fully defined."* állítása nehezen tűnik védhetőnek. Noha egy külső eseményre adott

válasz egy sok komponensből álló rendszerben valóban tekinthető nemdeterminisztikusnak, ez csak abból ered, hogy a *sok komponens állapota* nincs szigorú szinkronban – *egy* konkrét állapotgéppel modellezett komponens modelljében a nemdeterminizmus minden bizonnyal hiba. Milyen valós biztonságkritikus rendszerben tartja elképzelhetőnek, hogy a fejlesztők szándékosan nemdeterminisztikus viselkedést modelleznek egy állapotgépen belül?

2. A 2.2.4.1 szakaszban bevezetett szemantika az UML összetett átmeneteit "operation"-ök alkotta szekvenciák, elágazások, párhuzamos működés, stb. struktúrájaként írja le, ahol az értékadások kb. az átmenetekhez rendelt aktivitásoknak, az "assumption"-nek nevezett mellékhatásmentes kifejezések kiértékelése pedig az őrfeltételeknek felelnek meg. Hogyan kezeli a szemantika azt a helyzetet, ha egy (akár elágazásmentes) átmenet láncon az első elemi átmenet (operation) őrfeltétele (assumption) teljesül, de a hozzá tartozó aktivitás (assignment) úgy módosítja az állapotteret (változók értékét), hogy a láncban következő átmenet őrfeltétele (assumption) már nem teljesül (akár úgy, hogy az átmenetek közötti vertex egy junction pszeudoállapot) – hogyan kerüli el a szemantika azt, hogy ilyen helyzetben ne álljon elő egy félig végrehajtott átmenet? Ha ezt a kérdést még egy modellellenőrzési probléma keretében lehet is kezelni, mit tudunk kezdeni azzal, ha egy valós szoftverben a modellezett elemi átmenethez rendelt aktivitások módosítják a környezetet, vagyis nem vonhatók vissza? A kérdésről a 10. oldal alján ez szerepel: *"Note that assumptions may cause any composite operation to yield an empty set as the set of successor states. This allows us to use the choice operation as a guarded branching operator, ruling out branches where an assumption fails by yielding an empty set as the result of that branch. In this work, we make the following assumptions, which can be easily guaranteed by simple pre-processing."* – mi az az "egyszerű előfeldolgozás", ami garantálja, hogy egy tetszőleges bonyolult elemi átmenet struktúra elkezdése valamilyen kiindulási állapotból biztosan nem okoz olyan változást a változók kiértékelésében, hogy a lánc egy későbbi "assumption"-ja ne teljesüljön?
3. A 11. példa kifejezésében az első sor a PC-nek 2-t ad értékül, majd az összes alatta levő sor "assumption"-jében pc=1 szerepel ÉS kapcsolatban valami mással: ez helyes így?
4. 6.3-ban felderített deadlock lehetőség az olvasónak azt sugallhatja, hogy az UML több évtizedes története során javasolt számtalan formalizálási próbálkozás után máig nem sikerült olyan szemantikát leírni a szabványban, amely legalább egy ilyen triviális modellre egyértelmű lenne. Mikorra várható az UML-nek egy olyan verziója, amely mentes lesz ezektől a gyerekbetegségektől?

Budapest, 2023. január 9.

_____

Dr. Pintér Gergely

thyssenkrupp Components Technology Hungary Kft.