

FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA

Kolozsvár, 2004. március 26-27.

IDŐVEZÉRELT RENDSZEREK UML ALAPÚ TERVEZÉSE

Kovács Péter Tamás

Abstract: This paper presents how the advantages of visual design provided by UML can be used during the desing of embedded systems. The integration is achieved by interfacing the existing modeling tools. This way the advantages of both embedded systems' development environment such as platform-specific task and communication scheduling and fault tolerance, and UML such as documentation, model checking and support can be exploited.

Összefoglalás: Jelen dolgozat bemutatja, hogy hogyan használhatóak ki az UML nyújtotta vizuális tervezés előnyei beágyazott rendszerek tervezése során. Az integrációt a meglévő tervezőeszközökhöz készített illesztések valósítják meg, amelyek révén mind a beágyazott rendszerek fejlesztőrendszerének előnyei, mint a platformspecifikus taszk és kommunikáció ütemezés, vagy a hibatűrés, mind az UML által nyújtott dokumentáltság, modellellenőrzés, és a széleskörű támogatottság kiaknázhatóak.

Bevezetés

Az idővezérelt architektúra [2] (Time Triggered Architecture, TTA) napjainkban terjedőben van a szigorúan valós idejű és hibatűrő szolgáltatásokat igénylő beágyazott számítógépes rendszerekben (pl. gépjárművekben, repülőgépekben). Ez az architektúra, amely tipikusan egy elosztott rendszer, képes garanciát adni a válaszidőkre, támogatja a redundancia kialakítását, így használatával nagyfokú hibatűrés érhető el.

Ilyen rendszerek tervezésére szolgál a TTP-Tools, amely nem él a vizuális tervezés lehetőségeivel, ezen kívül az alkalmazás kódjának vizuális tervezését és automatikus generálását sem támogatja.

Célunk ezért a tervezés vizuálissá tétele a klaszter szintű tervezéstől egészen az alkalmazáskód generálásáig. A tervek előállításánál az UML eszközkészletére fogunk támaszkodni, mivel elterjedt, és számos eszköz támogatja. A vizuális tervezés megkönnyíti a rendszer létrehozását, mivel a tervet áttekinthetőbbé és könnyen módosíthatóvá teszi.

Az idővezérelt architektúra

Az *idővezérelt* kifejezés esetünkben azt jelenti, hogy a hálózaton történő kommunikáció időzítési viszonyai tervezési időben meghatározottak. Ennek a megközelítésnek számos előnye van: az üzenetek késleltetése előre meghatározott, a hálózat sohasem terhelődhet túl, illetve a csúcsterhelési viselkedés előre ismert, így a rendszer működése determinisztikus.

Egy TTA *klasztert* csomópontok alkotnak, amelyek redundáns kommunikációs sínen osztoznak. A *csomópontok* egy kommunikációvezérlőből és egy hoszt számítógépből épülnek fel. Az ezek közötti interfész a CNI (Communication Network Interface), amelyen keresztül a csomópontok periodikusan kapják az új állapotinformációkat. A csomópontokon futó alkalmazásoknak ezek alapján kell végrehajtaniuk a megfelelő akciót az előre meghatározott határidőn belül. A hibatűrő kommunikációt az FT-COM (Fault Tolerant Communication) réteg biztosítja az alkalmazások számára.

A TTTech tervezési folyamat

A TTA fejlesztői környezet [6] kétszintű tervezési filozófiát követ, amelynek nagy előnye, hogy a teljes rendszer és az alrendszerek tervezése élesen elkülönül egymástól. Klaszter szinten a tervező meghatározza a kommunikációs felületeket, majd az alrendszerek tervezői ehhez igazodva tervezhetik meg a hosztokat. A hosztok így egymástól függetlenül fejleszthetők, és a rendszer integrációja a hosztokból mint komponensekből várhatóan problémamentes lesz. A két szintet a fejlesztői környezet két tervezőprogramja, a TTPplan (klaszter szintű) és a TTPbuild (hoszt szintű) valósítja meg.

Ezeknek a programoknak alapvető hiányossága, hogy nem élnek a vizuális tervezés nyújtotta lehetőségekkel, továbbá az alkalmazáskód vizuális tervezését és automatikus generálását sem támogatják.

Vizuális nyelvek a rendszertervezésben

A Unified Modeling Language [4] a napjainkban legelterjedtebb grafikus modellezőnyelv, az Object Management Group szabványa. Az UML számos diagramtípusa a specifikációtól egészen a telepítésig segítséget nyújt. A tervek előállításánál során az osztálydiagramot és az aktivitási diagramot használjuk, ezért ezek néhány jellemzőjét ismertetjük az alábbiakban.

Az osztálydiagram a rendszer statikus felépítését mutatja meg. Az osztálydiagramok fő eleme az osztály, amely objektumok egy olyan készletét jellemzi, amelyek hasonló felépítéssel, viselkedéssel és kapcsolatokkal rendelkeznek. Az attribútumok egy osztály tulajdonságait írják le. Az asszociációk az osztályok közötti statikus relációkat jelölik.

Az aktivitási diagram a rendszer dinamikus viselkedését írja le. Az aktivitási diagram az állapotterkép olyan változata, ahol az állapotok bizonyos akciók elvégzését jelzik. Az aktivitási diagramot aktivitások, elágazások, kezdő- és végállapotok alkotják. Egy osztály viselkedését leíró aktivitási diagram az osztályhoz csatolható.

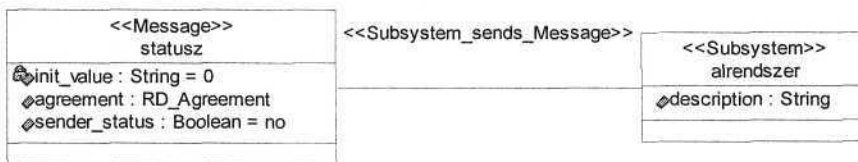
Az UML számos bővítési lehetőséget nyújt. Ilyenek a szabad formájú megjegyzések; a sztereotípiák, amelyek egy elem egy bizonyos kategóriához való tartozását jelzik; a címkézett értékek (tulajdonság-érték párok). Az UML specifikáció meghatározza továbbá az OCL nyelvet, melynek szerepe az, hogy a specifikációhoz tartozó olyan kötöttségeket, szabályokat is be lehessen vinni a modellbe, amelyeket egy diagramban másképpen nem tudnánk megadni.

Az UML jelölésrendszere testreszabható az UML profilok felhasználásával. Egy ilyen profil a bővítési lehetőségek kihasználásával kiterjesztheti a jelölésrendszert. Egy profilt általában sztereotípiák, címkézett értékek és korlátok felhasználásával adnak meg.

A TTA klaszter tervezésének támogatása

Az integráció lényeges kérdése az UML nyelven megadott terv betöltésének folyamata a TTTech programjaiba. A klaszter tervét elvben bármilyen UML modellező eszközzel előállíthatjuk, amely a modellt képes szabványos formátumba exportálni, de a Rational Rose-hoz a tervezést nagyban segítő bővítmények készültek. A modellek átvitelének szabványos formátuma az XMI [5], amely egy XML alapú leíró nyelv.

A vizuális tervben a TTP-Plan ill. TTP-Build objektumai - amelyek a klaszter és a csomópontok szoftver és hardver elemeit reprezentálják - sztereotípiával ellátott UML osztályokként jelennek meg. A közöttük levő kapcsolatokat pedig sztereotípiákkal ellátott asszociációk ill. asszociációs osztályok reprezentálják (ld. 1. ábra). Az elemek és kapcsolatok tulajdonságait azok attribútumai hordozzák (pl. az init_value a kezdőértéket jelenti). A használható elemek halmaza képezi az UML profilunkat.

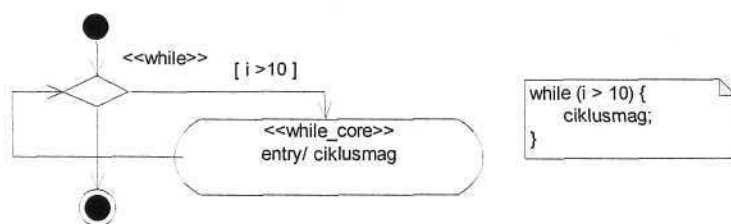


1. ábra Egy Message és egy Subsystem sztereotípiájú osztály, Subsystem_sends_Message sztereotípiájú asszociációval összekötve

Azt, hogy elemeinknek kötelezően rendelkezniük kell a hozzájuk tartozó attribútumokkal, és hogy két meghatározott típusú elem között csak meghatározott típusú asszociáció húzódhat, OCL kifejezésekkel írtuk le.

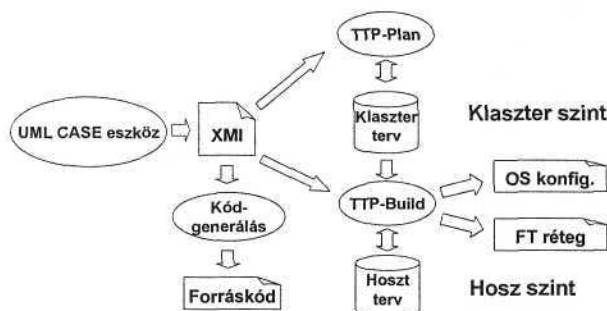
Tehát a tervezés során a profil által meghatározott osztályokat hozunk létre, módosítjuk attribútumaikat, meghatározzuk a közöttük levő kapcsolatokat és a kapcsolatok tulajdonságait.

A csomópontokon futó alkalmazás kódját aktivitási diagramok formájában adjuk meg. Ahhoz, hogy az ezekből előálló kód struktúráit legyen, a vezérlési szerkezetek ábrázolását meg kell határoznunk. Azokat a vezérlési szerkezeteket valósítottuk meg, amelyek a hagyományos struktúráit programozási nyelvekben szerepelnek. Ilyenek az elágazások (if, if-else, switch), a ciklusok (while, for, do-while) és a függvények. Ezek jelöléséhez az aktivitási diagramokra egy profilt adtunk meg, amely a specializált elemeket definiálja.



2. ábra Elöltesztelő ciklus aktivitási diagramként

A modelt XMI formátumba exportáljuk, ezután a TTP-Plan-be töltjük be a klaszter tervét, és készítettjük el vele a kommunikáció ütemezését leíró információkat, majd a TTP-Build-be töltjük be egyenként a hosztok terveit, és generáltatjuk a taszkok ütemezési tábláit, az FT-COM réteg kódját és az operációs rendszer konfigurációját. Ezekből egy kódváz áll elő, amelybe az alkalmazáskódot egy külön program generálja a tervben található aktivitási diagramok alapján.



3. ábra A vizuális tervezési folyamat

Összefoglalás

Eddigi munkánk eredménye az UML profil, az UML és a meglévő eszközök közötti adattranszformációt megvalósító eszközök, valamint a modell konstrukcióját és ellenőrzését segítő eszközök. A prototípus rendszer segítségével számos működő mintapéldát hoztunk létre, amely során világossá vált, hogy a vizuális tervezés valóban felgyorsítja a fejlesztési folyamatot, és eközben az idővezérelt architektúra előnyeit sem veszítjük el.

Irodalomjegyzék

- [1] Kopetz, H.: Real-Time Systems, Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers, 1997.
- [2] Kopetz, H., Bauer, G.: The Time Triggered Architecture. Proceedings of the IEEE, Special Issue on Modeling and Design of Embedded Software, October 2002.
- [3] Booch, G., Rumbaugh, J., Jacobson, L: UML User Guide. Addison-Wesley Publishing Company, 1998.
- [4] UML Specification version 1.4. Object Management Group, 2001. <http://www.omg.org/>
- [5] Object Management Group: *XML Metadata Interchange (XMI) 1.2*, 2002. <http://www.omg.org/technology/documents/formal/xmi.htm>
- [6] TTP-Tools. TTTech Computertechnik AG. <http://www.tttech.com/>

Kovács Péter Tamás / V. évf. műszaki informatika szakos hallgató

Budapesti Műszaki és Gazdaságtudományi Egyetem,

Méréstechnika és Információs Rendszerek Tanszék / 1117 Budapest, XI. Magyar tudósok körútja 2.

Telefon: (+36)30/515-07-55; E-mail: kovacsp@sch.bme.hu