

Kivonat

Az elmúlt évtizedek technológiai fejlődésének eredményeképp napjainkban rendkívül széles körben alkalmaznak programozható vezérlőket akár olyan kritikus területeken is, mint például az autóiipari beágyazott rendszerek. A „intelligens” eszközök korában ráadásul több szintre váltak szét az egy feladatot ellátó programok – például egy jármű berendezéseit közvetlenül irányító beágyazott számítógépeken futó alkalmazások szoros kapcsolatban állnak a központi számítógéppel, az pedig a felhőben futó szolgáltatásokkal. Az ilyen heterogén rendszerek komplexitása nagyon nagy lehet. A komplexitás kezelésének gyakori módszere a modellvezérelt fejlesztés, amely lehetővé teszi, hogy a fejlesztő a technikai részletek helyett a probléma logikai aspektusaira koncentráljon.

Jelen dolgozatban egy olyan, modellezést támogató eszközt mutatunk be, amely a fent körvonalazott rendszerek alábbi sajátosságaira ad választ.

1. **Komponens alapú felépítés:** A megcélzott rendszerek jellegzetessége, hogy komponensekből épülnek fel – ezért egy hierarchiát támogató modellezési nyelvre van szükség.
2. **Kommunikáció:** A komponensek jellegzetesen logikai jelekkel vagy üzenetekkel kommunikálnak – ennek támogatására jól definiált interfészekre és kommunikációs pontokra van szükség.
3. **Elosztott működés:** A komponensek nem csak egyetlen szoftvert reprezentálhatnak, hanem több, különböző fizikai eszközön futó programot alkothatnak – ez heterogén kommunikációt jelent, amihez többféle kompozicionális szemantikára van szükség.
4. **Minőségi és helyességi követelmények:** A rendszerek (bizonyos részei) gyakran kritikus feladatokat látnak el, ahol a helyes működés alapvető elvárás – ehhez helyes modellek építésére van szükség, ami validációval, verifikációval, a minőségi implementáció pedig automatikus kódgenerálással és teszteléssel támogatható.

A jelen dolgozatban bemutatott **gamma** keretrendszer egy hierarchikus, komponens-alapú reaktív rendszerek modellezését támogató eszköz. Az elemi komponensek viselkedésének leírásához az eszköz saját formális nyelvén túl a keretrendszerbe illeszthető külső eszközök (pl. Yakindu Statechart Tools) is használhatók. A komponensek háromféle szemantika szerint komponálhatók: az aszinkron-reaktív szemantika az elosztott kommunikáció modellezéséhez, a szinkron-reaktív szemantika az egy programon belüli vagy nagymértékben szinkron kommunikációhoz, a kaszkád kompozíció pedig egyetlen funkció logikai dekomponálásához nyújt megfelelő absztrakciót. A modellezési folyamatot a keretrendszer validációs szabályokkal segíti mind komponens, mind rendszer szinten. Az elkészített modellek minőségének biztosításához a keretrendszerhez illeszthető, a felhasználók elől elrejtett modellellenőrző eszközök (pl. UPPAAL) használhatók. A modellek alapján az implementáció automatikusan származtatható, amelynek helyessége a szintén automatikusan generálható validációs tesztkészlettel ellenőrizhető.