

Szoftver alapú hibatűrés támogatása aspektus-orientált programozással

Kákonyi István V. Inf.

Konzulens: dr. Majzik István,
Méréstechnika és Információs Rendszerek Tanszék

Napjainkban egyre inkább követelmény egy informatikai rendszerrel szemben, hogy rendelkezésre állása magas legyen, tehát szolgáltatásait megbízhatóan, igazolhatóan nyújtsa üzemelési idejének szinte teljes egészében. Különösen jogos ez az elvárás elosztott rendszerek esetén, ahol az egyes hardver illetve szoftver komponensek meghibásodása nem okozhatja a rendszer által nyújtott szolgáltatások kiesését. Egy ilyen elosztott rendszer szoftverének tervezése során ezért integrálni kell a rendszerbe olyan szoftverkomponenseket, melyek a megfelelő hibakezeléssel és a redundáns komponensek menedzselésével garantálják az elvárt szolgáltatás minőségét, más szóval a szolgáltatásbiztonságot. Ezeket a komponenseket tipikusan az alkalmazáslogikába ágyazva valósítják meg, megnehezítve így a szolgáltatásbiztonság elemzését, az erre vonatkozó tervezői döntések áttekintését és a későbbi módosítások kivitelezését. Dolgozatomban egy olyan módszert ismertetek, amelyet a tervezés és implementáció során alkalmazva ezek a hibatűrést megvalósító komponensek az alkalmazáslogikától függetlenül kezelhetőek.

Ennek a módszernek az alapötlete a következő: a szoftver tervezésekor és implementálásakor használjuk az aspektus-orientált paradigmát, ami az objektum-orientált paradigma kiterjesztésének tekinthető [1]. Nevezetesen a hibakezelést megvalósító szoftver algoritmusokat úgynevezett aspektusokban, a rendszer többi részétől elkülönítve tervezzük és valósítjuk meg. Majd az aspektusokban meghatározzuk, hogy ezek az algoritmusok az alkalmazáslogika mely kódrészleteinek futásakor hajtódjanak végre. Ennek a módszernek előnye, hogy az ily módon, a modellezés során elkülönített, hibatűrést megvalósító komponensek tervezési mintaként újra felhasználhatóak más projektekben. Továbbá az alkalmazáslogika tervezése és implementálása teljesen elkülönül a hibakezelést megvalósító algoritmusok tervezésétől és implementálásától. Így a két feladatot akár két, a maga területén szakértő csoport is végezheti.

A dolgozatban először ismertetem az objektum-orientált rendszerekben elterjedt redundancia kezelési, hibatűrés megoldásokat (mellékelve a megoldások tervezési mintáit), majd az olvasó képet kap az aspektus-orientált programozásról, modellezésről. Ezután következik a javasolt módszer gyakorlati példákon történő demonstrálása. Előbb egy hibatűrés kliens – szerver rendszer aspektus-orientált tervezését és Java nyelvű implementációját mutatom be. Második példaként pedig egy olyan elosztott rendszert ismertetek, melyben koordinált állapotmentés és helyreállítás segítségével biztosítható a rendelkezésre állás.

Az aspektus-orientált tervezést támogató különböző UML Profile-okat használva a javasolt megoldás illeszkedik a modell alapú tervezési folyamatba, s segíti a modell alapú analízist.

Irodalom:

1. G. Kiczales et. al.: Aspect-Oriented Programming. In Proc. European Conference on Object-Oriented Programming, Springer-Verlag LNCS 1241, June 1997.