

## Üzleti szabályok modellalapú fejlesztése

Az üzleti szabályok egyre elterjedtebb megoldássá válnak a vállalati rendszerek területén, aminek oka, hogy ez a megközelítés lehetővé teszi a gyorsan változó üzleti igények deklaratív és magas szintű megfogalmazását. Ezáltal a vállalati folyamatok és architektúrák szakterület-specifikus döntési pontjainak tervezése és bizonyos szintű fejlesztése nem igényel komolyabb programozó tudást, így a munkafolyamatok elvégzése a szakterület-specialista munkatársak számára is lehetségessé válik.

A módszer további előnye, hogy elviekben platformfüggetlen, tehát egy adott dialektusban megírt üzleti szabály (pl. JBoss Drools, vagy IBM ILOG JRules) felhasználható Java, de ugyanakkor akár .NET platformon is. Jellemző azonban, hogy ahány gyártó, annyiféle metamodell létezik, aminek egységesítését több szabvány is biztosítani próbálja, több-kevesebb sikerrel (RuleML, Rule Interchange Format – RIF). Szintén gyártótól függ, hogy az adott üzleti szabály motor milyen szintaktikai és szemantikai elemeket definiál.

Problémákkal szembesülünk, amikor üzleti szabályaink száma elkezd nőni, hiszen nem fogjuk tudni követni, hogy egy-egy újonnan felmerült igényre létezik-e már megoldás, nem okoz-e ellentmondást valamelyik korábban már implementált szabállyal, stb. Emellett a validálás és verifikáció matematikai eszközeinek alkalmazása is gondot okozhat egy ilyen gyengén specifikált nyelven.

A fenti problémákra a megoldást a modellalapú megközelítésben láttam, hiszen egy UML, vagy egyéb DSM modell már egyfajta absztrakciót ad a szabálynak, ami egy nagy lépés a formális ellenőrző módszerek alkalmazhatósága felé.

A motiváció, ami miatt a területtel foglalkoztam, a nagyvállalati szolgáltatásbuszok (ESB) funkcionalitásának támogatása, illetve akár annak egy részének teljes implementációval történő lecserélése. Ilyen funkcionalitás lehet például a routing, amikor szabályok alapján döntünk a csomagtovábbításról. A hálózati forgalmat feldolgozó üzleti szabályok nagy adathalmazzal dolgoznak, amelynek elméleti hátterével a Complex Event Processing területe foglalkozik. E módszert a JBoss Drools platform is támogatja a Drools Fusion nevű eszközével. Emiatt a szabályok modellezésénél lehetőség szerint érdemes a CEP funkcionalitás támogatására is figyelni.

Az Önálló laboratórium 1. tárgyban célul tűztem ki a Drools platform és általában véve az üzleti szabályok megismerését, a megfelelő modellezési módszerek kidolgozását, valamint egy prototípus és egy alkalmas tesztkörnyezet előállítását a prototípus tesztelés céljából.

A modellezési eljárásokat egy TMR (Triple-modular redundancy) mintarendszeren próbáltam ki, amelyet szintén a tárgy keretein belül implementáltam.

A félév végére elkészült egy UML profil, egy, a mintarendszert leíró domain-specifikus metamodell, két, a mintarendszerrel kapcsolatos szabályokat leíró objektummodell, valamint egy modell-kód transzformáció, amely a fenti modellekből azonnal használható Drools-dialektusú üzleti szabályokat állít elő.

Az Önálló laboratórium 2. tárgyban – megfelelő előrehaladás esetén – a módszereket egy valódi ESB-n szeretném kipróbálni (WebSphere ESB, vagy SOPERA ESB), adott esetben kihasználva a Drools egyéb eszközeinek (Fusion, Flow) lehetőségeit is. Sikeres tesztelés esetén a modellek általánosabb szintre emelését és a megfelelő modellezőnyelv kialakítását tűzném ki célul.