

Eclipse Equinox alapú elosztott szolgáltatások integrációja

Önálló laboratórium feladat összefoglalója (2. félév)

Horváth Ádám Gábor (XYUIVF)

Konzulens: Ráth István

**BME Méréstechnika és Információs Rendszerek Tanszék
Informatikai infrastruktúra tervezése szakirány, 2007/2008. II. félév**

Idei önálló labor témám a tavalyi témám folytatása. A félév célja a szolgáltatások integrációját már lokálisan (egy gépen) megvalósító Sensoria CASE Tool továbbfejlesztése úgy, hogy az képes legyen távoli tool-ok (szolgáltatások) elérésére, metódusaik meghívására. Követelmény, hogy a megalkotott rendszerben a szerverként működő példány (Eclipse Equinox) OSGi rendszerben, grafikus felület nélkül is működtethető legyen, hiszen a téma lényeges motivációja, hogy számításigényes feladatokat egy erős szerveren tudjunk futtatni.

A félév első részében a feladat megoldására alkalmas technológiákkal foglalkoztam és egy-egy, egyszerű példa összerakása után a tapasztalatokat levonva konzulensemmel közösen választottuk ki a kényelmes megvalósításra leginkább alkalmasat. Elsőként az R-OSGi (<http://r-osgi.sf.net>) keretrendszerrel ismerkedtem meg. Ez egy OSGi szolgáltatást definiál, mely lehetővé teszi más OSGi szolgáltatások megosztását OSGi rendszerek között (ahol az R-OSGi szolgáltatás szintén fut). A szolgáltatás lekérésekor kliens oldalra futásidőben generálódik egy „proxy” csomag (bundle), amely az eredeti szolgáltatás interfészét valósítja meg, de a hívásokat egy egyszerű TCP-alapú protokollon a szerver-oldali példányhoz továbbítja. A második általam megismert technológia tulajdonképpen egy komplett kommunikációs keretrendszer, az ECF (<http://www.eclipse.org/ecf>) (Eclipse Communication Framework). A keretrendszer egységes API segítségével lehetőséget teremt többféle kommunikációs csatorna felhasználására. Felhasználása azonban (a csekély dokumentáltság miatt) eléggé nehézkes, ezért a munkát az R-OSGi rendszerrel folytattam tovább.

A továbbhaladáshoz először az SCT modelljén kellett néhány átalakítást végezni. Az SCT alapvetően két fontos részből, a toolstore-ból (szolgáltatásokat leíró modell), és a blackboardból (objektumtároló a hívások eredményei számára) áll. Ahhoz hogy ezeket távolról el lehessen érni, az adatokat sorosítani kell. Így első feladatomban a meglévő SCT modellből a toolstore és blackboardhoz kapcsolódó adattípusok interfészének kiemelése volt, majd ez alapján egy második, szerializálható modell megírása. Az interfészek kiemelése mechanikus, egyszerű feladat, a sorosítható modell megírása azonban már hordozott magában kihívásokat. Lényeges pont volt például a távoli tool-on történő metódushívás végrehajtása: itt a Sensoria modell, és az R-OSGi koncepciója szerencsére remekül egybeesnek, és ezért ezzel sok gond nem adódott. A különbség annyi, hogy távoli tool esetében a szolgáltatást nyújtó objektum lekérésekor proxy objektumot kapunk az R-OSGi rendszertől, amely egy távoli tool-nak továbbítja a hívásokat.

Miután a modell elkészült, megírtam a szükséges logikát, ami a lokális modell alapján felépíti a szerializálható modellt, illetve a lokális szolgáltatásokat távoli szolgáltatásként elérhetővé teszi az R-OSGi segítségével, végül pedig egy kis kiterjesztést SCT-hez, amivel egy távoli Sensoria Core-hoz (egy SCT példány központi eleme) kapcsolódni lehet.

Ezen kész prototípus, továbbfejlesztésében Sisak Áron volt nagy segítségemre – ő egy speciális SAL modellellenőrzéshez használatos tool-t fejlesztett, a prototípust felhasználva, és közben lényeges hiányosságokra derített fényt. Az egyik következmény a távoli blackboard eseménykezelésének megváltoztatása volt, mely azóta lehetővé teszi eseménykezelők hozzáadását, és ezáltal a GUI dinamikus frissítését (akár a lokális blackboard esetében). Egy

másik lényeges hibára is fény derült: a tool-ok távoli elérésének engedélyezésekor nem az őt tartalmazó bundle került átadásra az R-OSGi keretrendszernek, és ez a függőségek rossz megállapítása miatt a kliens oldalon okozott furcsa hibajelenségeket.

Összességében a tool-ok távoli elérésének architektúrája véglegesnek tekinthető, és bonyolultabb, saját szerializálható típusokat használó esetekben, a gyakorlatban is működik. Szükséges még a későbbiekben speciális, konfigurálható tool-okkal történő tesztelés (Property és Option állítási lehetőségek). Egyelőre korlátozás, hogy a kliens és a szerver ugyanazon a gépen kell, hogy legyen (a szerver fixen a localhost címre bindol), illetve, hogy egy bundle aktiválását kézzel kell elvégeznünk. Ezeket még a későbbiekben orvosolni fogom. Későbbi lehetséges érdekes téma különböző hívási szemantikák támogatásának lehetősége, illetve autentikáció támogatása – ezek nagyobb munkának ígérkeznek, mivel az R-OSGi egy egyszerű, mobil eszközökbe szánt rendszer, mely ezekhez nem biztosít támogatást.