Ontológia kezelő modul
Specifikáció vázlat

Förhécz András

Az ontológiát beágyazó modul feladatai:

- moduláris ontológia importálása,
- az ontológia modulok függőségének kezelése,
- a moduláris ontológia betöltése a következtetőgépbe,
- a következtető leírólogikai szolgáltatásainak közvetítése,
- konkurrens kérések kiszolgálása.

Ontológia nyelv. Az ontológia nyelve az *OWL*, mely a *KAON*-nal szemben lehetőséget ad az entitások logikai formulákba öntött, pontos szemantikus ábrázolására.

Az *OWL* természetesen támogatja moduláris ontológiák építését, ahol az egyes modulok

- hivatkozhatnak más modulokon belüli entitásokat,
- egész modulokat illeszthetnek be a saját készletükbe.

Ontológia importálása. Az ontológia modulokat *file*-ból (a *base uri* megadásával), vagy *http* protokollon keresztül tölthetjük be az ontológiakezelőbe. Az ontológia-szerkesztő nem része a komponensnek, a későbbiekben azonban lehetőség van annak megoldására, hogy az ontológiát közvetlenül a szerkesztőből tölthessük a komponensbe. Ez a jelenleg használt szerkesztő (*Protégé*) esetén *plug-in* írásával lenne lehetséges, mely egyelőre aránytalanul sok többletmunkát igényel. A komponens tervezésekor ezt a lehetőséget azonban nem zárjuk ki.

Az importált modulokat a komponens lokálisan tárolja és feljegyzi az utolsó frissítés dátumát. A komponens újraindításával visszaállítja az eredeti konfigurációt, így nem szükséges a modulokat újra betölteni. Lehetőség van továbbá a modulok frissítésére, mentésére és kidobására is.

Függőségek kezelése. A szolgáltatott ontológia-lekérdező API-n keresztül a betöltött modulok összességén futtathatunk lekérdezéseket. A betöltött modulokat azonban a lekérdezhetőségük előtt *aktiválni* kell, ez jelenti a következtetőgépbe való betöltésüket.

A modulok aktiválásához biztosítani kell, hogy az általuk importált modulok szintén be legyenek töltve, illetve aktiválva legyenek. Ennek biztosítását jelenti a *modulok függőségének kezelése*.

Egy ontológia függ egy másiktól, ha azt az *owl:imports* direktívával importálja. Például:

```
<owl:imports
  rdf:resource="http://localhost/ontologies/base.owl"/>
```

Egy külső entitásra hivatkozást tehát nem tekintünk feltétlenül függőségnek. Például két, külön-külön is használható ontológiát *owl:equivalentClass* definíciókkal össze lehet kapcsolni anélkül, hogy a komponens megkövetelné az együttes használatukat.

Az ontológia függőségei kielégítettek, ha már minden hivatkozott ontológiája aktiválva van. Csak kielégített függőségek esetén lehet egy betöltött ontológiát aktiválni, azaz az API-n keresztül lekérdezhetővé tenni. Az *OWL* nem zárja ki körkörös importálások használatát, így lehetővé kell tenni egymást hivatkozó modulok együttes betöltését.

Az ontológia aktiválásával történik meg annak klasszifikálása, a következtetőbe való betöltése is.

Az ontológiák függőségének kezelésére egy belső, konfigurációs ontológiát használok, mely tárolja az ontológiák tulajdonságait és állapotát. Ezzel később akár az is automatikusan ellenőrizhetővé válik, hogy az aktiválandó ontológia minden külső entitás-hivatkozása megtalálható-e a már aktivált ontológiákban.

Moduláris ontológia betöltése a következtetőgépbe. Az egyes modulokat a függőségeknél részletezett módon lehet a következtetőgépbe betölteni. A komponens a *RACER* következtetőt használja. A következtető hívására a *JRacer* interfészt hívja, mely a *RACER* TCP protokollon keresztül kommunikáló, Java nyelvű API-ja.

Az *OWL* ontológia beolvasására, kezelésére használt *OWL API* egyelőre csak konzisztencia-ellenőrzést, illetve az ontológia a nyelv szintjei közé való besorolását (*OWL Lite*, *OWL DL*, *OWL Full*) támogatja. Meg kell tehát oldani az *OWL* definíciók *RACER* parancsokba való fordítását, mely az ontológia bejárásával és a definíciók egyesével történő fordításával oldható meg.

Leírólogikai szolgáltatások közvetítése. A *RACER* segítségével klasszifikálható az ontológia, amit a szokásos ontológia-lekérdező primitívek nyújtásával (pl. *subClasses*) érhetünk el. A szolgáltatásokat a *Java RMI* (Remote Method Invocation) távoli eljáráshívásával nyújhatjuk a többi komponens felé.

A klasszifikált ontológián túl milyen leírólogikai lekérdezéseket akarunk nyújtani, illetve milyen felületen oldható ez meg?