

A jól szerkesztett mérnöki ontológiákról

Szóts Miklós, Simonyi András

Alkalmazott Logikai Laboratórium
1022 Budapest, Hankóczy J. u. 7.
e-mail:{szots,simonyi}@all.hu

Kivonat Az ImportNET projekt keretein belül folytatott munkánk során azzal a problémával szembesültünk, hogy nem léteznek a mérnöki tervezést hatékonyan segítő ontológiák. Cikkünkben olyan általános ontológiatervezési elveket és mintákat mutatunk be, melyek segítségével jól strukturált, a mérnöki szemlélethez közelálló csúcsontológiák hozhatók létre.

Kulcsszavak: mérnöki ontológia, az ontológiatervezés módszertana, ontológiamodularizáció, ontológaszegmentálás

1. Bevezetés

Az ImportNET projekt¹ [8] egy kollaboratív mechatronikai tervezést segítő, ontológiaalapú szoftver létrehozását tűzte ki célul. A megvalósult rendszer a kollaboratív tervezési folyamat megkezdésekor egy átfogó mechatronikai doménontológiából választja ki azt az ontológiaszegmenst (az ún. kollaborációs ontológiát), amely az adott kollaboráció szempontjából releváns mechatronikai tudást tartalmazza. Az ontológia szegmentálása félautomatikusan történik: a domént jól ismerő, de a formális ontológiák területén járatlan szakértő egy grafikus felhasználói felületen kiválaszt néhány, a tervezés során várhatóan releváns, illetve bizonyosan irreleváns fogalmat és relációt, és a rendszer ennek alapján automatikusan generál egy kollaborációs ontológiát, amelyet a felhasználó tovább finomíthat.

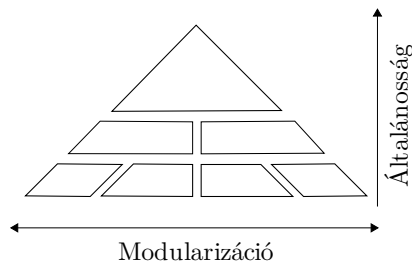
A projekt keretében végzett munkánk során azzal a problémával szembesültünk, hogy a doménontológia naív felhasználók általi szerkesztésének támogatása, illetve a szegmentálás csak megfelelően strukturált, jól szerkesztett ontológián végezhető el hatékonyan. A jól szerkesztettség általunk talált kritériumainak jelentős része ontológiafüggetlennek bizonyult – cikkünkben ezeknek az ontológiafüggetlen strukturális követelményeknek, illetve elveknek az összefoglalására teszünk kísérletet, az ImportNET projekthez kapcsolódó példákon mutat be gyakorlati alkalmazásukat.

¹ Az ImportNET projekt az Európai Bizottság támogatásával, a 6. Keretprogramom belül valósult meg, az IST-2006-033610 számú szerződés alapján.

2. Rétegzés és modularizáció

Az általunk talált egyik legfontosabb ontológiaszerkesztési elv a *rétegzés* elve: a reprezentálandó tudást célszerű az általánosság foka szerint rétegekre osztani. Az ontológia minden osztálya és relációja eleme egy és csak egy rétegnek, és a kevésbé általános rétegekhez tartozó osztályok részosztályai az általánosabb rétegek osztályainak. Mivel a specifikusabb rétegek többnyire komplexebbek, és több információt tartalmaznak az általánosabbaknál, ezért egy bizonyos általánossági szint alatt a rétegeket koordinált *modulokra* célszerű osztani. A programmodulokhoz hasonlóan az ontológiamodulok olyan ontológiai részek, melyek elemei között sok kapcsolat található, és melyeknek viszonylag kevés kapcsolata van a modulon kívüli elemekkel.

Az ontológiák szokásos kétdimenziós ábrázolására (az egyes osztályok és relációk részosztályaik, illetve részrelációik fölött helyezkednek el) támaszkodva azt mondhatjuk, hogy a rétegzés az ontológia vertikális, a modularizáció pedig horizontális felosztásának felel meg (lásd az 1. ábrát).



1. ábra. Egy komplex ontológia rétegzése és modularizációja

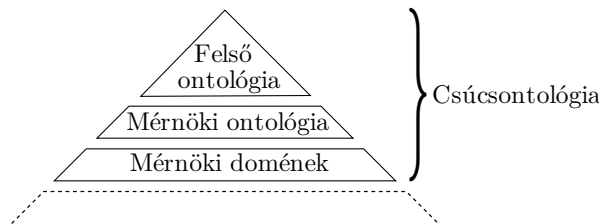
Egy átfogó ontológiában a legfontosabb vertikális tagolás a csúcsontológia (top ontology) elkülönítése. Bár a „csúcsontológia” terminust gyakran használják a legáltalánosabb, doménfüggetlen fogalmakat tartalmazó *felső ontológia* (upper ontology) értelemben, a mi szóhasználatunkban egy átfogó ontológia csúcsontológiai rétege az a szegmens, amely meghatározza a teljes ontológia alapszerkezetét azáltal, hogy rögzíti a *relációk* modellezésének módját. Ebből adódóan a csúcsréteg az ontológia összes relációját tartalmazza: az ontológia megmaradó része új elemként kizárólag osztályokat és individuumokat vezethet be.

Mivel a modellezési kérdések már a csúcsontológiai rétegben eldőlnék, ezért ontológiaszak-értői munkát csak ennek a rétegnek a kidolgozása igényel. Az ontológia további része többé-kevésbé mechanikus „T-box benépesítéssel” tölthető fel, pl. létező taxonómiák importálásával, vagy a doménszakértők által könnyen kezelhető, a feltöltést segítő felhasználói felületen keresztül.

3. Komplex ontológiák tagolása

A rétegzést egy több tudásterületet (domént) átfogó, komplex ontológiára alkalmazva olyan vertikálisan tagolt ontológiához jutunk, melynek csúcsontológiai része a következő rétegekből áll:

- a legátfogóbb, doménfüggetlen osztályokat és relációkat tartalmazó felső ontológia,
- a leírt doménokra együttesen alkalmazható, de nem doménfüggetlen osztályokat és relációkat tartalmazó réteg, végül pedig
- egy réteg, mely doménspecifikus tudást tartalmaz (az ImportNET ontológiában ez a réteg többek között mechanikai és elektronikai tudást fed le).



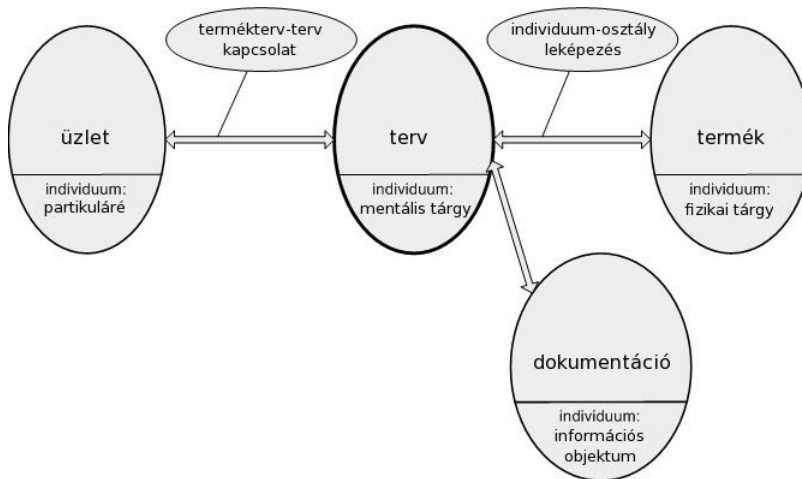
2. ábra. Egy vertikálisan tagolt, komplex ontológia felső rétegei

3.1. A felső ontológia

Mivel a közismert felső ontológiák, pl. a DOLCE [10], a SUMO [11] és a BFO [5] erősen különböző módon reprezentálják a legalapvetőbb relációkat (pl. a téridő viszonyokat), a felső ontológia megválasztása meghatározza a csúcsontológia további rétegeinek szerkezetét is. Ennek ellenére, az ImportNET projekt mérnöki ontológiájának fejlesztése során néhány olyan modellezési problémával is szembesültünk, melyek függetlenek ezektől a különbségektől.

A legfontosabb ilyen kérdések egyike az volt, hogy miként célszerű reprezentálni a mérnöki terveket, a tervek alapján készülő konkrét termékeket, valamint a köztük fennálló viszonyt. Mivel egy felső ontológia elképzelhetetlen egy, a konkrét fizikai tárgyakat tartalmazó osztály nélkül, ezért a konkrét termékek kategorizációja viszonylag könnyű feladat: pl. a DOLCE felső ontológiában ezek a PHYSICAL-OBJECT osztály példányainak tekinthetők. A *tervek* kategóriájának meghatározása már jóval nehezebb feladat. Habár a mérnökök rajzok és írott (papíron vagy elektronikus formában tárolt) dokumentumok segítségével reprezentálják terveiket, azok nem azonosak konkrét fizikai reprezentációikkal — pontosan azért, mert az utóbbiak csupán reprezentálják őket. A tervek kategorizációs problémájának két legfontosabb megközelítését „realista” és „konstruktivista” megközelítésnek nevezhetjük.

Az első megközelítés a terveket téren és időn kívüli absztrakt objektumoknak tekinti, melyek ontológiai státusza hasonló ahhoz, melyet a matematikai platonisták tulajdonítanak a matematika tárgyainak: az ember nem létrehozza, csupán felfedez(het)i őket. A DOLCE és a SUMO esetében ez a megoldás a terveket az ABSTRACT osztály példányainak tekintené. A realista megközelítéssel ellentétben a konstruktivista felfogás a terveket *mentális objektumok*ként kezeli, melyek ez emberi elme tevékenységének eredményei. Ennek megfelelően a konstruktivista felfogás szerint minden terv csak egy adott időponttól kezdve létezik. A DOLCE tartalmaz egy MENTAL-OBJECT osztályt, más felső ontológiák azonban csak közvetett eszközökkel rendelkeznek a mentális objektumok leírásához. A SUMO-ban pl. található egy INTENTIONAL-PROCESS osztály, melyhez egy műszaki cikk megtervezésének *folyamata* tartozik, és a tervek maguk olyan dolgokként jellemezhetőek, melyek résztvevői egy tervezési folyamatnak (vö. [7]).



3. ábra. Egy modularizált mérnöki ontológia

3.2. Modularizáció

A 3. ábra egy mérnöki ontológia egy természetesnek tűnő modularizációját mutatja, mely a következő egységekre bontja a műszaki réteget:

- **Üzleti ontológia.** Ez a modul a kollaborációkkal kapcsolatos üzleti-gazdasági tudást fedi le, így pl. tartalmazza azokat az osztályokat és relációkat, melyek az együttműködésben résztvevő vállalatokra, illetve dolgozóikra vonatkozó információ reprezentációjához szükségesek, különös tekintettel az együttműködésben betöltött szerepükre (pl. ki a kollaborációs projekt vezetője stb.). A termékmenedzsmenttel kapcsolatos tudást szintén ez a modul reprezentálja.

- **Tervezési ontológia.** A tervezési ontológia indiviuumtartománya kizárólag mérnöki tervekben áll, vagyis olyan objektumokból, amelyek a jövőben gyártásra kerülő konkrét termékek tulajdonságait reprezentálják. A modul osztályhierarchiájának jelentős része izomorf a termékek osztályainak hierarchiájával (lásd a következő részt).
- **Dokumentációs ontológia.** A dokumentációs ontológia azokat az információs objektumokat reprezentálja, melyek a kollaboráció során jönnek létre, pl. írott terveket, tervrajzokat, műszaki dokumentációt stb. Ezek az információs objektumok nem keverendők össze konkrét fizikai megvalósulásaikkal: egy tervrajznak (ami egy információs objektum) sok különböző fizikai példánya, másolata létezhet.
- **Gyártási ontológia.** Ha a kollaboráció sikeres volt, akkor az elkészült tervek alapján legyárthatóak a konkrét termékek. A gyártási ontológia indiviuumai fizikai tárgyak: az elkészülő műszaki cikkek és részek.

A modularizációval szemben támasztott követelményünkkel összhangban a különböző modulok elemei között viszonylag kevés a kapcsolódás.

Az üzleti és a tervezési modul között egyetlen fontos kapcsolat áll fent: bizonyos tervek *terméktervvé* válnak, vagyis az általuk leírt tárgyakat gyártják és forgalomba hozzák. A *termékterv* fogalom tipikus szerepfogalom (abban az értelemben, ahogyan ezt a metatulajdonságot az OntoClean [6] metodológia használja), mivel akkor alkalmazható egy individuumra, ha az részt vesz egy *kontingens* üzleti folyamatban (v.ö. [6, 16]). Ebből adódóan a terméktervek osztályát nem célszerű egyszerűen a TERV osztály részének tekinteni — előnyösebb megoldás a tervek üzleti szerepeinek reifikációja, mely esetben a szerepeket kizárólag az üzleti ontológia indiviuumtartományában szükséges szerepeltetni.

Terjedelmi okok miatt nem térhetünk ki az üzleti és a dokumentációs modul között fennálló, igen komplex kapcsolatrendszerre, de a tervek és termékek közötti viszony olyan kiemelkedő fontosságú, hogy mindenképpen szólnunk kell róla röviden.

3.3. Tervek és termékek

A tervek és termékek viszonyával kapcsolatos reprezentációs nehézségek a következő feszültségből adódnak: Egyfelől, a tervek lényegileg különböznek a szerintük legyártott termékektől, mivel tulajdonságaik túlnyomó része különbözik (pl. egy számítógép terve maga nem számítógép). Természetesen van kapcsolat egy számítógépterv és a „számítógépnek lenni” tulajdonság között: a tervek valamiképpen reprezentálják, illetve *kódolják* a tulajdonságot, és minden, a tervet megvalósító tárgy exemplifikálja azt. Másfelől, a tervezőmérnökök gyakran kezelik úgy a terveiket, mintha azok rendelkeznének az általuk kódolt tulajdonságokkal — ez a gyakorlat különösen hasznos akkor, amikor tervekkel kapcsolatos következtetéseket kell végezni. Pl. természetesnek tűnik az a következtetés, hogy ha minden számítógép tartalmaz egy processzort, akkor hiányosak azok a számítógéptervek, melyekből „hiányzik a processzor.”

A tervek és az őket megvalósító termékek közti viszony most vázolt két oldala két egymást kiegészítő követelményhez vezet a viszony formális reprezentációjára nézve:

- A reprezentáció nem feltételezheti, hogy a tervek és az őket megvalósító fizikai tárgyak általában ugyanazon osztályok példányai.
- Ennek ellenére, tükröznie kell azt a tényt, hogy szoros kapcsolat áll fent a tervek és a termékek tulajdonságai között, amely a következőképpen jellemezhető:
 - Minden t tervre van olyan φ osztály, melynek példányai azok a fizikai tárgyak, melyek *megvalósítják* a tervet:

$$\forall t \exists \varphi \forall a (\varphi(a) \equiv \mathcal{M}(a, t)) \quad (1)$$

- Létezik egy \mathcal{K} kódolás reláció a tervek és a termékosztályok között, amely a következő tulajdonságokkal bír:
 - * Ha egy terv kódol egy tulajdonságot, akkor minden, a szóbanforgó terv szerint gyártott termék rendelkezik az adott tulajdonsággal:

$$\forall t \forall \varphi (\mathcal{K}(t, \varphi) \rightarrow \forall a (\mathcal{M}(a, t) \rightarrow \varphi(a))). \quad (2)$$

- * Ha a φ -t kódoló tervek osztálya részosztálya a ψ -t kódoló tervek osztályának, akkor φ részosztálya ψ -nek:

$$\forall \varphi \forall \psi (\forall t (\mathcal{K}(t, \varphi) \rightarrow \mathcal{K}(t, \psi)) \rightarrow \forall a (\varphi(a) \rightarrow \psi(a))). \quad (3)$$

Sajnos a fenti formális jellemzés nem fejezhető ki közvetlenül deskriptív logikai (DL) nyelveken a standard DL-szemantika segítségével, mivel osztályok fölött kvantifikál, és szerepel benne a másodrendű \mathcal{K} reláció. Ebből adódóan, ha a mérnöki ontológiát egy DL-formalizmusra támaszkodva kívánjuk reprezentálni, akkor a tervek és termékek közti viszonyt vagy egy nemstandard DL-szemantika használatával, vagy a használt DL nyelven kívüleső eszközökkel fejezhetjük ki. A következőkben a probléma három lehetséges megközelítését tárgyaljuk röviden: külön termék- és tervontológia használatát egy köztük megadott leképezéssel (ontology mapping), egyetlen ontológia használatát metaszabályokkal, és végül a DOLCE Descriptions and Situations kiterjesztésének [3,4] alkalmazását.

Ontológialeképezés. A leképezésalapú megközelítés külön tervontológia és gyártási ontológia létrehozását igényli, melyek terméktulajdonságokra utaló közös osztályneveket tartalmaznak, pl. ‘CPU’, ‘32BIT_CPU’ stb. A közös nevek szemantikája különböző, de közel álló: Ha egy F osztálynév az $\{x : \varphi(x)\}$ termékosztályra referál a gyártási ontológiában, akkor az $\{x : \mathcal{K}(x, \varphi)\}$ tervosztályra referál a tervontológiában, vagyis azon tervek osztályára, amelyek *kódolják* az F által kifejezett tulajdonságot. Például míg a ‘CPU(a_{17})’ formula interpretációja a gyártási ontológiában az lehet, hogy ‘ a_{17} ’ referenciája egy konkrét központi processzor, addig a tervontológiában ugyanez a formula azt jelenheti, hogy ‘ a_{17} ’ referenciája egy központi processzor *terve*. A két ontológia közti \mathcal{L} leképezésnek a következő tulajdonságokkal kell rendelkeznie:

- Összhangban az (1) megkötéssel a tervontológia individuumaikat (vagyis a terveket) a gyártási ontológia azon osztályaira képezi le, melyek az adott terv alapján legyártott termékeket tartalmazzák. Például a tervontológia ‘ a_{17} ’ nevű terve a gyártási ontológia ‘INTEL80486DX’ nevű osztályára lehet leképezve. Habár az \mathcal{M} megvalósítás reláció nem fejezhető ki a két ontológiában, a leképezés segítségével könnyen definiálható: egy a termék pontosan akkor a megvalósítása egy t tervnek, ha a példánya a $\mathcal{L}(t)$ osztálynak.
- A (2) megkötést követve a tervontológia A-box formuláit a gyártási ontológia bizonyos T-box formuláira képezi le: az ‘ $F(t)$ ’ alakú állításokhoz, ahol t egy tervre utal, F pedig a $\{x : \mathcal{K}(x, \varphi)\}$ osztályra, az ‘ $F \sqsupseteq \mathcal{L}(t)$ ’ állításra képezi le. Például a tervontológia ‘32BIT_CPU(a_{17})’ állításának képe a gyártási ontológia ‘32BIT_CPU \sqsupseteq INTEL80486DX’ állítása lesz.
- Végezetül, a (3) megkötésnek megfelelően ha F az $\{x : \mathcal{K}(x, \varphi)\}$ osztályt jelöli a tervontológiában, és az $\{x : \varphi(x)\}$ osztályt a gyártási ontológiában, és egyúttal G az $\{x : \mathcal{K}(x, \psi)\}$ osztályt jelöli a tervontológiában és az $\{x : \psi(x)\}$ osztályt gyártási ontológiában, akkor a tervontológia ‘ $F \sqsubseteq G$ ’ formulájának a leképezés szerinti képe sajátmaga.

A leképezésen alapuló megoldás legfontosabb előnye az, hogy (a lehetőségekhez mérten) megfelel a tervezőmérnöki szemléletnek, és követi a tervezők nyelvi gyakorlatát. E szerint a megközelítés szerint a CPU-hoz hasonló predikátumok tervekre és termékekre egyaránt alkalmazhatóak, de „szisztematikusan többértelműek”: míg termékekre alkalmazva azt állítják, hogy a kérdéses termék rendelkezik egy tulajdonsággal, addig egy tervről azt mondják, hogy *kódolja* a szóban forgó tulajdonságot.

Egyetlen ontológia metaszabályokkal. A második megközelítés egyetlen ontológiában reprezentálja mind a terveket, mind a termékeket, és az (1), (2) és (3) megkötéseket részben az ontológia metanyelvén fejezi ki. Fontos előnye ennek a megoldásnak, hogy az \mathcal{M} megvalósítási reláció az ontológia nyelvén reprezentálható, és ebből adódóan egy t tervet megvalósító termékek osztálya egyszerűen definiálható a [MEGVALÓSÍTTJA : t] osztályként. A leképezésen alapuló megoldással szemben egy tulajdonság példányainak osztálya és az ugyanezen tulajdonságot kódoló tervek osztálya nem kaphat azonos nevet, de az összetartozó nevek összekapcsolhatók egy megfelelően választott elnevezési séma segítségével, pl. kiköthető, hogy ha F az $\{x : \varphi(x)\}$ osztályt jelöli, akkor az ‘ $F_KÓDOLÓJA$ ’ osztálynév az $\{x : \mathcal{K}(x, \varphi)\}$ osztályt jelölje. Ezt az elnevezési sémát használva a szükséges metaszabályok a következőképpen fogalmazhatók meg:

- Ha az ontológia tartalmaz egy ‘ $F_KÓDOLÓJA$ ’ alakú osztálynevet, akkor tartalmaz egy F nevű osztályt is.
- Ha az ontológia tartalmaz egy ‘ $F_KÓDOLÓJA(t)$ ’ alakú állítást, akkor tartalmaz egy ‘[MEGVALÓSÍTTJA : t] $\sqsubseteq F$ ’ alakú állítást is.
- Ha az ontológia tartalmaz egy ‘ $F_KÓDOLÓJA \sqsubseteq G_KÓDOLÓJA$ ’ alakú állítást, akkor tartalmazza az ‘ $F \sqsubseteq G$ ’ állítást is.

Descriptions and Situations. Az utolsó megközelítés, amelyet röviden meg szeretnénk említeni, a DOLCE felső ontológia Desriptions and Situations (röviden DnS) kiterjesztésének segítségével reprezentálja a tervek és termékek kapcsolatát. A DnS-t sikeresen használtuk az ImportNET projektben a cselekvési tervek formális reprezentációjára [1], és olyan osztályhierarchiával rendelkezik, amelybe a tervek egyszerűen elhelyezhetők, mivel a SYSTEM-DESIGN osztály példányainak tekinthetők. Ennek ellenére ez a megoldás meglehetősen problematikus.

Az egyik nehézség az, hogy a megvalósítás reláció egyetlen szóba jöhető reprezentánsa a DnS rendszerben a SATISFIES, amelynek az értelmezési tartománya a SITUATION osztály, amely viszont részosztálya a NON-PHYSICAL-OBJECT osztálynak. Ebből adódóan a DnS keretei között a termékek csak nemfizikai individuumoknak tekinthetők. Bár fontos filozófiai érvek szólnak amellett, hogy a termékeket szociális konstrukciónak, és ne fizikai tárgynak tekintsük, (lásd pl. [12]), ez a megkülönböztetés idegen a mérnöki szemlélettől, és nyilvánvaló előnyök nélkül növeli a reprezentáció bonyolultságát. A módszer egy másik hiányossága, hogy nem teszi lehetővé a tervek komponensek egymáshoz való viszonyának a mérnöki szemléletnek megfelelő reprezentációját. Végezetül megjegyzendő, hogy az előző szakaszban tárgyalt megoldáshoz hasonlóan a DnS-alapú megközelítés egyetlen ontológiában reprezentálja a terveket és a termékeket, és ezért alkalmazása esetén a megvalósítás reláció fontos jellemzői csak meta-szabályokkal vagy egyéb, az ontológia nyelvén kívül eső eszközökkel fejezhető ki.

3.4. Reprezentációs mélység

A ‘mérnöki ontológia’ kifejezés többértelmű. Még ha rögzítjük is a domént (pl. az elektronika területét), a reprezentáció *mélysége* nyitott kérdés marad: nem lesz tisztázott, hogy a tervezési folyamat mely fázisait támogatja az ontológia. A következő reprezentációs szinteket különböztethetjük meg:

1. PDM (termékadat-kezelés) szint: a tervezett tárgyak komponenseinek tulajdonságait és mereológiai viszonyait ábrázolja az ontológia.
2. Topológiai szint: a komponensek topológiai kapcsolatait szintén reprezentálja az ontológia, de a pontos geometriai részletek nélkül (pl. csak áramkör diagramokat ad meg).
3. Geometriai szint: a komponensek elhelyezkedése és mérete is ábrázolásra kerül.
4. Működési szint: az ontológia a tervezett tárgy működését is reprezentálja, esetleg a működés helyessége is ellenőrizhető a segítségével (helyesek-e az elvégzett számítások, a méretezés stb.).

A szintek fenti sorrendje egyúttal bonyolultsági, összetettségi sorrend is: a lejjebb elhelyezkedő szintek a magasabban lévő fogalmi eszközeit is felhasználják. A négy szint közül a tervek PDM-szintű reprezentációja jól ismert, és viszonylag egyszerű eszközökkel elvégezhető, mivel csak

- a tervek atomi és nematomai komponenseinek megkülönböztetését, valamint
- a RÉSZE reláció és a
- a tervek komponensek tulajdonságainak reprezentációját igényli (ez utóbbi megtehető pl. a DOLCE minőségeket reprezentáló mechanizmusának OWL-DL implementációjára támaszkodva [10, 16]).

A PDM-szinttel szemben a többi szint reprezentációja komoly kihívást jelentő feladat. A topológiai szint speciális relációinak reprezentációja minden bizonnyal a kapcsolatok reifikációjával oldható csak meg, a geometriai tulajdonságok leírását pedig nagyon megnehezíti a hely fogalmának relativitása [2]. Végezetül, egy a működési szintet is reprezentáló ontológia kifejlesztése valószínűleg egy kvalitatív fizikai elmélet formalizációját is szükségessé teszi, ami (az esetleges kvantitatív adatokat is figyelembe véve) igen komoly nehézségekbe ütközhet egy DL-alapú környezetben.

4. Összefoglalás

Cikkünkben olyan ontológiafejlesztési elveket javasoltunk, melyek segítségével jól strukturált mérnöki ontológiák hozhatók létre. Véleményünk szerint kifejleszthetők olyan mérnöki ontológiák, melyek eleget tesznek ezeknek a módszertani elveknek, és egyúttal jól modellezik a gyakorló mérnökök szemléletmódját, fogalomrendszerét.

Két, az átfogó ontológiák szerkesztése során alkalmazható szerkezeti alapelvet ismertettünk: a rétegzés, vagyis az általánosság foka szerinti vertikális szegmentáció elvét és a modularizáció, vagyis a viszonylag kevés külső kapcsolattal rendelkező horizontális szegmensekre, modulokra bontás elvét.

Egy mérnöki ontológia fontos további dimenziója, hogy milyen mélységig képes reprezentálni műszaki terveket, illetve tervezési folyamatokat. Amellett érveltünk, hogy a tervezett tárgyak mereológiai szerkezetét és komponenseik tulajdonságait viszonylag egyszerű reprezentálni, ugyanez azonban távról sem mondható el a komponensek topológiai, geometriai és működési viszonyairól, mivel az utóbbi három terület formális reprezentációja komoly kihívást jelentő feladat, különösen DL-alapú ontológiai nyelvek használata esetén.

Hivatkozások

1. Damjanovic, V., Behrendt, W., Plössnig, M., Holzapfel, M.: Developing Ontologies for Collaborative Engineering in Mechatronics. In: Proceedings of the 4th European Semantic Web Conference, Innsbruck (2007)
2. Donnelly, M.: Relative Places. *Applied Ontology* **1** (2005) 55–75
3. Gangemi, A., Mika, P.: Understanding the Semantic Web through Descriptions and Situations. In: Meersman, R. (ed.): Proceedings of ODBASE'03 Conference, Springer (2003)
4. Gangemi, A., Borgo, S., Catenacci, C., Lehmann, J.: Task Taxonomies for Knowledge Content. Deliverable D07 of the METOKIS Project (2005)

5. Grenon, P.: BFO in a Nutshell: A Bi-categorical Axiomatization of BFO and Comparison with DOLCE. Technical report. Ifomis (2003)
6. Guarino, N., Welty, C.: An Overview of OntoClean. In: Handbook on Ontologies. Springer (2004) 151-159
7. Hung, L.C., Beng, L.H., Wah, N.G., Yin, H.K.: Plan Ontology and its Applications. In: 7th Int. Conference on Information Fusion (2004)
8. Mahl, A., Semenenko, A., Ovtcharova, J.: Virtual Organisation In Cross Domain Engineering. In: Establishing The Foundation Of Collaborative Networks. Springer (2007) 601-608
9. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: WonderWeb Deliverable D18: Ontology Library. Technical report. Laboratory for Applied Ontology (2003)
10. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: WonderWeb Deliverable D18: Ontology Library. Technical report. Laboratory for Applied Ontology (2003)
11. Niles, I., Pease, A.: Towards a Standard Upper Ontology. In: FOIS '01: Proceedings of the International Conference on Formal Ontology in Information Systems. New York, ACM (2001) 2-9
12. Vieu, L., Borgo, S., Masolo, C.: Artefacts and Roles: Modelling Strategies in a Multiplicative Ontology. In: Proceedings of FOIS 2008 (2008)