

**Vas Réka Franciska**

**Tudásfelmérést támogató  
oktatási ontológia szerepe  
és alkalmazási lehetőségei**

Információrendszerek Tanszék

Témavezető: Dr. Gábor András

© Vas Réka Franciska

Budapesti Corvinus Egyetem  
Gazdálkodástani Doktori Iskola

# **Tudásfelmérést támogató oktatási ontológia szerepe és alkalmazási lehetőségei**

Ph.D. értekezés

Készítette:

Vas Réka Franciska

BCE, Információrendszerek Tanszék

Budapest, 2007

---

---

## Tartalomjegyzék

I.	A kutatás áttekintése.....	1
I.1.	Bevezetés.....	1
I.2.	A kutatás tárgyának megjelölése, a kutatási probléma megfogalmazása.....	1
I.3.	A témaválasztás indoklása.....	3
I.4.	A kutatás jelentősége és haszna a társadalmi gyakorlatot érintő szempontból.....	3
I.5.	Hipotézisek.....	4
I.6.	Kutatási megközelítés, módszertan.....	5
I.6.1.	Kvantitatív és kvalitatív megközelítés.....	5
I.6.2.	Deduktív és induktív módszer.....	6
I.6.3.	A kutatás célja szerinti megközelítés.....	6
I.7.	A dolgozat további sajátosságai, felépítése.....	7
II.	Oktatási ontológia – Elméleti háttér és modellezés.....	9
II.1.	Ontológiák – Bevezetés.....	9
II.1.1.	Ontológia fogalma.....	9
II.1.2.	Felhasználási területek.....	12
II.2.	Ontológia felépítése, módszertanok.....	15
II.2.1.	Grüninger és Fox féle módszertan.....	15
II.2.2.	Uschold és King féle módszertan.....	17
II.2.3.	CommonKADS módszertan.....	18
II.2.4.	A KACTUS projekt keretében kidolgozott módszertan ..	19
II.2.5.	Sure-Studer féle On-To-Knowledge módszertan.....	20
II.2.6.	Methontology.....	22
II.2.7.	Sensus.....	23
II.2.8.	Az ontológiaépítési módszertanok összehasonlító elemzése.....	24
II.3.	Ontológia nyelvek.....	28
II.3.1.	Elsőrendű logikán alapuló nyelvek.....	30
II.3.2.	Keret tudásábrázolási formán és elsőrendű logikán alapuló nyelvek.....	31
II.3.3.	Leíró logikai nyelvek.....	34
II.3.4.	Web alapú nyelvek.....	35
II.3.5.	Ontológia nyelvek összehasonlítása – Összefoglaló.....	42
II.4.	Kapcsolódó következtető rendszerek.....	42
II.4.1.	A kifejezőerő vizsgálata.....	44
II.4.2.	A következtető gépek tesztelése.....	45
II.4.3.	Eredmények, értékelések.....	45
II.5.	Ontológiák értékelése.....	47
II.5.1.	OntoClean-módszertan.....	51
II.6.	Ontológia-szerkesztő eszközök.....	51
II.6.1.	Általános jellemzők.....	52
II.6.2.	Architektúra jellemzői.....	53
II.6.3.	Módszertan, tudásábrázolás, ontológiaépítés jellemzői.....	54
II.6.4.	Együttműködési képesség.....	56
II.6.5.	Következtetés támogatása.....	57
II.6.6.	Az eszköz alkalmazhatósága.....	59

---

II.7.	A kutatás elvi modellje .....	61
II.7.1.	Kiindulási modell.....	61
II.7.2.	Oktatási ontológia modellje.....	62
II.7.3.	Tesztkérdések halmaza.....	66
II.7.4.	Oktatási ontológia modell alkalmazása.....	67
III.	Adaptív tesztelés.....	78
III.1.	Az adaptív tesztelés fogalma és története .....	78
III.2.	Számítógéppel támogatott adaptív tesztelés .....	79
III.2.1.	Kétértékű elemekből álló vizsgasor .....	80
III.2.2.	Többértékű elemekből álló vizsgasor .....	82
III.2.3.	Mérési modell kiválasztása.....	83
III.2.4.	A számítógépes adaptív tesztelés további módszerei ...	85
III.3.	Az Oktatási ontológiával támogatott adaptív tudásszint-felmérés módszere .....	88
III.3.1.	Az adaptív vizsgáztatás módszertana .....	88
III.4.	Oktatási ontológiával támogatott adaptív tesztrendszer működtetése .....	93
III.5.	Hiányterületek tételes feltárásának módszere .....	96
III.5.1.	A hiányterületek feltárására irányuló tesztelés menete..	97
III.5.2.	A megvalósítás és kivitelezés .....	98
III.5.3.	A hiányterületek feltárását támogató vizsgarendszer tesztelésének eredményei .....	100
IV.	Összefoglalás .....	104
IV.1.	A kutatás célkitűzései .....	104
IV.2.	A kutatás eredmények .....	104
IV.2.1	Az Oktatási ontológiához kapcsolódó megvalósíthatósági elemzés.....	105
IV.2.2	Az Oktatási ontológia fejlesztésének kiindulási fázisa .	105
IV.2.3	Az Oktatási ontológia finomítása .....	106
IV.2.4	Az Oktatási ontológia értékelése .....	107
IV.2.5	Az Oktatási ontológia karbantartása és fejlesztése .....	110
IV.2.6	A kutatási eredmények értékelése.....	110
IV.3.	További kutatási irányok .....	110
V.	Függelék .....	112
V.1.	Fogalomtár .....	112
V.1.2.	Bizonyítások tulajdonságai .....	117
V.2.	A szakképzettségben elvárt kompetenciák köre .....	119
V.3.	Az alapképzési és mester szak tanterve.....	121
V.4.	Az alapképzési és mester szak tanterve.....	124
VI.	Hivatkozások.....	126

---

## Ábrák jegyzéke

1. ábra: A TOVE ontológia szerkesztése és értékelése (OntoWeb, 2002a, p. 15.).....	17
2. ábra: A CommonKADS modelljei közötti kapcsolat (Molnár, 2003) ....	19
3. ábra: A KACTUS megközelítés (Schreiber et al., 1995) .....	20
4. ábra: Sure-Studer féle ontológiafejlesztés folyamata (Staab et al., 2001).....	21
5. ábra: A Methontology ontológia-életciklus (Corcho et l., 2003) .....	23
6. ábra: A Sensus megközelítés (OntoWeb, 2002a, p. 21.) .....	24
7. ábra: Az Oktatási ontológia kiindulási modellje .....	61
8. ábra: Tudásterületek és munkakörök közötti ontológiai kapcsolat .....	64
9. ábra: A tudásterületek belső szerkezetének ontológia modellje.....	66
10. ábra: Az Oktatási ontológia modellje.....	67
11. ábra: Tudásmenedzsment-vezető munkakör jellemzői .....	74
12. ábra: A Tudásmenedzsment-vezető munkakörhöz kapcsolódó feladatcsoportok .....	75
13. ábra: „Feladatcsoport” osztály példányának jellemzői és relációi .....	75
14. ábra: „Kompetencia” osztály példányának jellemzői és relációi .....	76
15. ábra: „Tudásterület” osztály példányának jellemzői és relációi .....	76
16. ábra: Altudásterület jellemzői és relációi .....	77
17. ábra: Az adaptív tesztelés folyamatábrája (Thissen és Mislevy, 1990).....	85
18. ábra: Problémakörök kapcsolata a Tudástér elméletben .....	87
19. ábra: Az Oktatási ontológiával támogatott adaptív vizsgáztatás folyamata .....	92
20. ábra: Tesztfelület a Protégé-ben.....	93
21. ábra: A „Kérdés” osztály jellemzői és relációi.....	93
22. ábra: Az adaptív vizsgarendszer kérdező felülete.....	94
23. ábra: A számonkérés eredményeinek értékelése – I. ....	94
24. ábra: A számonkérés eredményeinek értékelése – II. ....	95
25. ábra: A számonkérés eredményeinek értékelése – III. ....	95
26. ábra: Az adaptív tudástesztelési modul .....	97
27. ábra: A hiányterületek feltárását támogató gazdaságinformatika ontológia .....	99
28. ábra: Az eredményesség szórása tantárgyak szerint .....	102
29. ábra: Eredményességek eloszlása .....	108
30. ábra: Az Oktatási ontológia modellje.....	106

---

## Táblázatok jegyzéke

1. táblázat: Ontológiaépítési módszertanok összeilleszthetősége, kiegészíthetősége .....	27
2. táblázat: Ontológiaépítési módszertanok fogalmi rendszerezése, alkalmazott leíró és specifikációs nyelvek.....	28
3. táblázat: A tudásábrázolás és a következtetési mechanizmusok kapcsolata (Schreiber et al., 1998) .....	29
4. táblázat: Ontológiaszerkesztő eszközök általános jellemzői.....	53
5. táblázat: Ontológiaszerkesztő eszközök architektúra jellemzői.....	54
6. táblázat: Ontológiaszerkesztő eszközök ontológiaépítését támogató jellemzői.....	55
7. táblázat: Ontológiaszerkesztő eszközök együttműködési képességei .....	57
8. táblázat: Következtetés támogatása az ontológiaszerkesztő eszközöknél .....	58
9. táblázat: Az ontológiaszerkesztő eszközök alkalmazhatóságát befolyásoló tényezők .....	60
10. táblázat: A feldolgozott tananyagok, tudásterületeik és kapcsolódó kérdések száma.....	100
11. táblázat: A vizsgarendszer teszteléséhez kapcsolódó eredménymutatók .....	101

---

## **Köszönetnyilvánítás**

Ezúton is szeretnék köszönetet mondani témavezetőmnek, dr. Gábor Andrásnak útmutatásáért és nélkülözhetetlen tanácsaiért. Sok segítséget és értékes szakmai támogatást kaptam édesapámtól, dr. Vas Zoltántól a felkészülés, a dolgozat elkészítése és a nagyobb akadályok leküzdése során.

Köszönettel tartozom mindazon tanszéki kollégáimnak, akik javaslataikkal és észrevételeikkel segítettek munkám tökéletesítésében, a dolgozat elkészítésében: dr. Molnár Bálintnak, dr. Kő Andreának, Kovács Barnának és Szabó Ildikónak.

Családom rendületlen támogatása nélkül e dolgozat nem készülhetett volna el. Külön köszönöm férjemnek megértő türelmét és azokat az órákat, melyeket nekem adott, hogy e több éves munka eredményét letehessem az asztalra.



---

## I. A kutatás áttekintése

### I.1. Bevezetés

Napjainkban a szervezetek rendelkezésére álló információ és tudás rendezése, strukturálása vagy rendszerbe foglalása hatalmas kihívást jelent a szervezetek számára. Szinte napról napra jelennek meg újabb és újabb információs technológiák, informatikai eszközök, melyek hatékony segítséget próbálnak nyújtani a tudáskezeléshez kapcsolódó feladatokhoz. Az ontológia, mint a mesterséges intelligencia, tudásmenedzsment, adatbázis modellezés elméleti eszköze, a világ fogalmi szinten történő megragadására, leírására tesz kísérletet. Ma már jól látszik, hogy egyre több szakterületen merül fel igény az ontológiák használatára. Számos példát találhatunk ipari területen előállított ontológiákra (Enterprise Design Ontology<sup>1</sup>, Skills Management Ontology<sup>2</sup>, OntoShare<sup>3</sup>) és adott tudományterület<sup>4</sup> tudását keretbe foglaló ontológiákra is. Az alkalmazási területek széles köre is igazolja, hogy mind az ipar, mind a tudományos világ felismerte, hogy a tudás-feltérképezési és -leírási folyamat középpontjában – a fogalmak pontos definiálása mellett – a fogalmak között fennálló szemantikai viszonyok minél teljesebb és részletesebb feltárása kell, hogy álljon.

A bolognai folyamatok kihívásainak való megfelelés, s a felzárkózás érdekében nemcsak a hazai felsőoktatás szervezeti, az oktatás tartalmi és módszertani megújulását kell elérni, hanem az oktatásban alkalmazott eszközök és technológiák korszerűsítése is elengedhetetlen. Dolgozatom rámutat arra, hogy az oktatási területek ontológiai modellezése hogyan segíthet az oktatást modernizáló tudásalapú eszközök és rendszerek tervezésében, fejlesztésében és alkalmazásában.

### I.2. A kutatás tárgyának megjelölése, a kutatási probléma megfogalmazása

A dolgozat célja, hogy a Budapesti Corvinus Egyetem Gazdálkodástudományi Karának gazdaságinformatikus szakán megszerezhető kompetenciákon keresztül ontológiai kapcsolatot teremtsen a munkaerőpiac által támasztott követelmények, valamint az oktatott tárgyak tartalma között.

Célként tűzöm ki továbbá, hogy a modell támogassa a hallgatók által elsajátított ismeretek adaptív módon történő felmérését, különös tekintettel az oktatás eredményességére és a hiányterületek tételes feltárására.

---

<sup>1</sup> <http://www.eil.utoronto.ca/tove/enterprise.html>

<sup>2</sup> Fensel et al., 2003

<sup>3</sup> Davies et al., 2002

<sup>4</sup> Leginkább a természettudományok (kémia, biológia, matematika) területén található példákat erre.

---

A kutatási célok megvalósítása érdekében dolgozatomban a következő kutatási problémák elemzésére és a hozzájuk kapcsolódó kutatási feladatok elvégzésére törekszem:

- 1) Hogyan modellezzük a gazdaságinformatikus képzés keretében oktatott tárgyak tudásszerkezetét a fent meghatározott céloknak megfelelően? Hogyan építsük fel az oktatási ontológia modelljét?
  - a. Az ontológia, mint tudásreprezentációs eszköz, illetve az ontológia-komponensek fogalmainak tisztázása.
  - b. Az ontológiák felhasználási területeinek, alkalmazási lehetőségeinek elemzése.
  - c. A gazdaságinformatikus szak keretében oktatott tárgyak és a képzés során megszerezhető kompetenciák körének meghatározása.
  - d. Az eltérő területek közös vonásainak megragadására alkalmas modell kialakítása, az oktatási ontológia alapelemeinek és a köztük lévő kapcsolatok meghatározása.
- 2) Hogyan támogatja az alapmodellnek megfelelően kialakított ontológia az adaptív vizsgarendszer felépítését, annak működését?
  - a. Az adaptív vizsgáztatás elvei, számítógéppel támogatott adaptív tesztelés<sup>5</sup> alapjainak megismerése.
  - b. Az oktatási ontológia és az adaptív vizsgarendszer összekapcsolása.
- 3) Hogyan támogatja egy megfelelően kidolgozott módszertan az ontológiaépítést? Milyen módszertant alkalmazzunk az oktatási ontológia felépítéséhez?
  - a. Az ontológiafejlesztési módszertanok bemutatása, összehasonlító elemzése a szakirodalom alapján.
  - b. A létrehozott oktatási ontológia modelljéhez illeszkedő módszertan kidolgozása.
- 4) Milyen formalizmusok támogatják az ontológiában tárolt tudás reprezentálását? Hogyan válasszuk meg az ontológia leíró nyelvét?
  - a. Ontológianyelvek bemutatása, összehasonlító elemzése a szakirodalom alapján.
  - b. Az alkalmas ontológianyelv kiválasztása a kitűzött célok és a kialakítandó ontológiamodell alapján.
  - c. Az alkalmazható következtető gépek vizsgálata, elemzése.
  - d. Az ontológiaépítést támogató és egyszerűsítő ontológiaszerkesztő eszközök vizsgálata
- 5) Az ontológia – meghatározott részterületeinek – kidolgozása.
- 6) Milyen további alkalmazási lehetőségei vannak a kialakított ontológiának?

---

<sup>5</sup> Computer adaptive testing (CAT)

---

### **I.3. A témaválasztás indoklása**

Az utóbbi évtizedek gazdasági, technológiai, informatikai fejlődése a változások gyors és hatékony kezelését követeli meg a szervezetektől. Az alkalmazkodáshoz, a szervezetek hatékony működéséhez a tudás megfelelő kezelése, intenzív felhasználása elengedhetetlen feltétel. Ma már általánosan elfogadott az a megközelítés, hogy a tudás, az innovatív megoldások kulcsszerepet játszanak a versenyelőny megteremtésében.

Ez a gyorsan változó világ a felsőoktatást is komoly kihívások elé állította. A folyamatosan megújuló és bővülő tudás megszerzésének és átadásának versenyében csak akkor tudnak a helyt állni, ha képesek folyamatosan és gyorsan alkalmazkodni a kor követelményeihez. A magyar felsőoktatási intézményeknek hatékonyabban kell bekapcsolódnuk Európa felsőoktatási rendszerébe, annak érdekében, hogy a magyar egyetemek átjárhatóvá váljanak európai társaikkal.

A Bolognai Nyilatkozat – mely az európai felsőoktatás előtt álló kihívásokra, problémákra keresi a választ – legfontosabb célkitűzései közé sorolja a könnyen áttekinthető és összehasonlítható képzési rendszer bevezetését, hogy az EU bármely tagállamában szerzett végzettségeket, tudást és ismereteket az állampolgárok képesek legyenek elismertetni az EU egész területén (Kovács, 2004).

Az oktatott ismeretek, tudásterületek ontológiai modellezése, kompetenciáknak való megfeleltetése, az oktatási ontológiára épülő számítógépes adaptív vizsgáztatás eszközeinek és módszereinek kidolgozása, mind olyan tényezők, melyek a felsőoktatás versenyképességének növeléséhez és a fenti elvárások teljesüléséhez járul hozzá.

### **I.4. A kutatás jelentősége és haszna a társadalmi gyakorlatot érintő szempontból**

Egyes tudományterületeken, szakterületeken belül is találkozhatunk eltérő felfogásokkal, megközelítésekkel, attól függően, hogy milyen háttérrel rendelkeznek az érintett szakértők, oktatók. Az oktatási ontológia hozzájárulhat e problémák kezeléséhez az oktatás szerkezetének világossá tételével, a képzést felépítő tantárgyak és tudásterületek kapcsolatának pontos meghatározásával. A szakterületek egységes fogalomrendszerének kialakításával az ontológiáknak konszenzusteremtő ereje van, és hatékonyan támogatják az együttműködést.

Az oktatási ontológia tudásmegosztásban játszott szerepe mellett, a szakértelem újrafelhasználhatóságának biztosításában, a rendelkezésre álló tudás aktualizálásában is fontos szerepet játszik. Az ontológia segítségével ábrázolt tudáshalmaz – strukturáltsága, áttekinthetősége miatt. – gyorsabban és hatékonyabban bővíthető és frissíthető, az új ismeretek könnyebben beilleszthetők, a rögzített információk gyorsabban visszakereshetők. Mindemellett az ontológiák által ábrázolható, kezelhető tudás mélységére vonatkozóan sincsenek korlátozások.

---

Az ontológiák, formalizálási lehetőségei következtében, nemcsak az emberek, hanem a számítógépek közötti szemantikus együttműködést is lehetővé teszik, elősegítve ezzel a tudásalapú rendszerek fejlesztésének hatékonyabbá tételét, karbantartásuk és továbbfejlesztésük megkönnyítését. Az ontológiák további előnye, hogy számos területen, számos – akár eltérő funkciójú – rendszer felépítéséhez is alkalmazhatóak. A dolgozat egyik célja, annak bemutatása, hogy a kialakított ontológia milyen módon támogathatja a számítógépes adaptív tesztelést, melynek alkalmazása szintén az oktatás minőségének, hatékonyságának javulásához vezet. A vizsgáztatás e módja gyorsabb és rövidebb, ráadásul nagyobb biztonsággal határozza meg a vizsgázó tényleges tudásszintjét. A tesztelek aktualizálása, a tesztek kiértékelése, eredmények megállapítása szintén kevesebb időt vesz igénybe, ráadásul a legtöbb esetben a vizsgázó számára is kellemesebb vizsgaélményt nyújt (Linacre, 2000.).

A megfelelően kialakított oktatási ontológia modell és ontológia nemcsak a gazdaságinformatikus képzés minőségének javításához, hanem bármely más oktatási terület fejlesztéséhez hozzájárulhat. A kutatás várható eredményei így hozzájárulnak a magyar felsőoktatás versenyképességének növeléséhez, a tudásalapú gazdaság, illetve társadalom kiépüléséhez.

## **I.5. Hipotézisek**

*H1: Ontológia segítségével leírható egy meghatározott egyetemi (alapképzési / mesterképzési) szak képzési anyaga, figyelembe véve az elsajátítandó ismeretanyag és tudás jellemzőit, a szakképzettségben elvárt kompetenciák körét, valamint az ontológiaépítés és -fejlesztés módszertani, formalizálási követelményeit.*

A kutatás egyik célja, hogy a gazdaságinformatikus szakon megszerzhető kompetenciákon keresztül ontológiai kapcsolatot teremtsen a munkaerőpiac által támasztott követelmények, valamint az oktatott tárgyak tartalma között. Ennek érdekében meg kell vizsgálni, hogy a képzés által közvetített tudás reprezentálására kialakítható-e olyan ontológiai modell, mely az oktatott tárgyak széles köre és eltérő jellege ellenére is alkalmas az egyes területek fogalmainak és a köztük fennálló szemantikai viszonyok ábrázolására. A hipotézis alátámasztása érdekében tehát meg kell vizsgálni, hogy kialakítható-e olyan ontológia modell, mely valóban alkalmas az érintett képzési anyag modellezésére. Továbbá igazolni kell, hogy a modell megfelel a módszertani elvárásoknak és, hogy az ontológia nyelvek közül kiválasztható olyan, amely támogatja a modellben leírt tudás formalizálását.

*H2: Alkalmasan megválasztott adaptív tesztelési rendszerrel az egyén tudása megragadható, jól mérhető.*

A kutatás további célja a hallgatók által elsajátított ismeretek adaptív módon történő felmérésének támogatása, különös tekintettel az oktatás eredményességére és a hiányterületek tételes feltárására. A hipotézis igazolása érdekében elemezni kell az adaptív vizsgáztatás elveit és módszereit, meg kell ismerni a számítógéppel támogatott tesztelés megoldásait, és ki kell választani az alkalmazni kívánt eljárást. Ugyan-

---

akkor meg kell fogalmazni az egyénnel szembeni elvárásokat. Ennek során egyrészt meg kell határozni a felmérni kívánt tudásterületet, másrészt az elvárt tudásszintet. A kiválasztott vizsgáztatási módszer alkalmassága csak az egyénnel szembeni elvárások pontos meghatározásával tudjuk csak megállapítani. A tudásfelmérés és értékelés pedagógiai szempontjainak vizsgálata azonban túlmutat a dolgozat keretein, éppen ezért ennek elemzése nem célom.

*H3: Az egyén által ismerni vélt tudás hiányterületei tételesen feltárhatók az oktatási ontológia által reprezentált ismeretanyag és az adaptív tesztelés eredményei alapján.*

A kialakítani kívánt adaptív vizsgáztatási rendszer célja, nem csupán a vizsgáztatás hatékonyságának és gyorsaságának növelése, az eredmények gyors közlése, hanem az is, hogy rámutasson azokra a tudásterületekre, tananyag részekre, melyeken az egyénnek hiányossági vannak. A hipotézis bizonyításához meg kell vizsgálni, hogy mi lehet a szerepe az oktatási ontológiának a hiányterületek meghatározásában, hogy miért fontos a képzést felépítő tudásanyag rendszerezése, strukturálása. Továbbá be kell mutatni, hogyan tudja támogatni egy adaptív vizsgarendszer felépítését és működését a tudásterületet reprezentáló ontológia és azt, hogy ennek érdekében milyen megoldással kapcsolható össze a két rendszer.

## **I.6. Kutatási megközelítés, módszertan**

A választott kutatási téma az informatika és a társadalomtudományok háttérterületéhez tartozik, mely befolyással van a kutatás módszertanának kiválasztására is.

A kutatási módszertanokat elsősorban két nagyobb csoportra szokták bontani – a leginkább szervezetelemzési módszereknél elterjedt – kvantitatív és kvalitatív megközelítésekre (Balaton, Dobák 1991).

### **I.6.1. Kvantitatív és kvalitatív megközelítés**

Kvantitatív módszereket elsősorban olyan kutatások esetén alkalmazunk, melyek nagy mennyiségű adatokra, illetve az adatok sokváltozós statisztikai módszerekkel történő elemzésére épülnek (Füstös et al., 1986).

A kvalitatív módszerek ezzel szemben nem számszerű adatok elemzésére építenek, hanem az összefüggések minél mélyebb feltárására törekszenek. A kvalitatív megközelítése alkalmazása nem kiforrott, egységes fogalomrendszerrel még nem rendelkező területek esetén javasolt. Ez a megközelítés lehetőséget ad egyéni gondolkodásmódok megismerésére, a kutatás mélyebb kontextusának feltárására.

Mind a kvalitatív, mind a kvantitatív módszerekkel kapcsolatban számos kétely merült fel, melyek ellensúlyozására javasolják a módszertani trianguláció alkalmazását, melynek a következő típusait különböztetjük meg:

- kvantitatív módszereken belül többféle eljárás használata,

- 
- kvalitatív módszereken belül többféle eljárás használata, illetve
  - kvantitatív és kvalitatív módszerek kombinációja (Klimkó 2001).

A felvázolt kutatási problémák és feladatok alapján jól látható, hogy a jelenlegi kutatás elvégzésére csak kvalitatív módszerek alkalmazhatóak.

### **I.6.2. Deduktív és induktív módszer**

A kutatási területhez kapcsolódó elméletekhez való viszonya alapján további két megközelítést különböztethetünk meg: a deduktív és induktív módszereket.

A deduktív módszerek az elmélettesztelésre épülnek, az adott tudományterület ismert és elfogadott elméleteinek igazolására, vagy cáfolására törekcszenek. Amennyiben deduktív módszert kívánjuk alkalmazni a kiválasztott terület szakirodalmának elemzését követően meg kell határoznunk a kutatás kulcsfogalmait, azonosítanunk kell a változókat és felvázolni a változók közötti ismert összefüggéseket, tételeket. Végül mindezek mentén logikus okfejtéssel következtetéseket vonunk le a témára vonatkozóan. (Babbie, 1996)

A deduktív módszer lépései tehát a következők:

- hipotézisalkotás,
- megfigyelés,
- a hipotézis elfogadása vagy elvetése.

Az induktív módszer, az elmélet tesztelésével ellentétben, kutatási problémák feltárására és új elméletek alkotására törekszik. Az induktív, feltáró jellegű módszer esetén, nem fogalmazunk meg hipotéziseket. Adott kutatási probléma megfigyelése alapján összefüggéseket próbálunk keresni, melyekből feltételes következtetések vonhatók le (Babbie, 1996). Az induktív módszer lépései tehát a következők:

- A tudós megfigyeléseket végez.
- A megfigyelések alapján szabályszerűségeket próbál megállapítani.

A jelenlegi kutatási téma deduktív jellegű, az ontológia módszertanok, ontológianyelvek és kapcsolódó következtető gépek újabb területen való alkalmazhatóságát, tehát az elméletek igazolását helyezi a középpontba.

### **I.6.3. A kutatás célja szerinti megközelítés**

A kutatás céljára vonatkozóan gyakran hivatkozott megközelítés szerint a következő kategóriákat különböztetjük meg: felfedező, leíró és magyarázó (Yin 1994). A kutatási célok megvalósításához kapcsolódóan, pedig a következő kutatási stratégiákat különbözteti meg:

- kísérleten alapuló
- kérdőíves felmérésen alapuló,
- másodlagos elemzésen alapuló,

- 
- történeti elemzésen alapuló és
  - esettanulmány feldolgozáson alapuló stratégia (Yin 1994).

A felsorolt stratégiák közül az ontológiai modell alkalmazhatóságának vizsgálatára csak esettanulmányokon keresztüli igazolás vagy cáfolás tűnik alkalmazható megoldásnak.

Eisenhardt (1989) azonban fontosnak tartja kiemelni, hogy az esettanulmányok nemcsak felfedező módon alkalmazhatók, hanem következő célok bármelyikének elérésére is: illusztrációs céllal (az elmélet megvilágítására), elmélet kifejlesztésére, és a már meglévő elmélet tesztelésére, kiegészítésére is.

A kutatás elvégzéséhez alkalmazható egy vagy több esettanulmány is. Egyetlen eset alkalmazása esetén azonban több forrásra való támaszkodásra kell törekedni, hogy minél pontosabb és részletesebb képet kapjunk a kutatni kívánt területről.

Több eset alkalmazása pedig a különbségek megragadásában segíthet, bár az eltérések számszerűsítése nem célravezető, nem meghatározó. Ugyanakkor a hasonlóságok vagy az eltérések további támpontokat, a kutatás további irányait is meghatározhatják.

Babbie (1996) következőkben látja az esettanulmányok hátrányait:

- nehezen ismételtető,
- következtetések nem olyan egyértelműek,
- esetek megbízhatósága is vita tárgya.

A téma jellegéből adódóan csak részletes, mélyszintű, esettanulmányokon, a szakirodalom eredményein alapuló megközelítés vezethet eredményre.

## **I.7. A dolgozat további sajátosságai, felépítése**

A tézisjavaslat alapját korábbi kutatások eredményei és megállapításai adják. Az ontológiaépítés és – fejlesztés területén végzett kutatások bemutatása, összehasonlító elemzése, eredményeik kiértékelése biztosíthatja a kutatás kérdéseinek és modelljének megalapozottságát.

A kutatott terület mindössze néhány évtizedes múltra tekint vissza, mely több problémát is felvet. A terület kutatói közössége által elfogadott, széles körben használt, kiforrott módszertanokról, egységesen alkalmazott formalizmusokról még nem beszélhetünk. Ma már jól láthatók a szabványosítás irányába tett lépések, azonban a felhasznált fogalomkészlet még mindig sokszínű. Éppen ezért a fontosnak tartom a dolgozat kapcsán felmerült fogalmak értelmezését és rendszerezését, melyet a Függelék V.1. fejezetben (Fogalomtár) külön foglalkozás össze a fogalmak definíciójának megadásával. Külön figyelmet kell fordítani a fogalmak fordítására is, hogy az angol kifejezések legmegfelelőbb magyar párját megtaláljuk.

A dolgozat felépítése a következő: a II. fejezet foglalja össze a kutatás elméleti hátterét. Az ontológia fogalmának, felhasználási területeinek ismer-

---

tetése mellett, az ontológiaépítő módszertanok, ontológianyelvek bemutatására és részletes összehasonlító elemzésére kerül sor. Továbbá a fejezet tartalmazza az ontológianyelvekhez kapcsolódó következtető gépek elemzési szempontjainak meghatározását, az ontológiák értékelési módszereinek rövid összefoglalását, valamint az építés folyamatát támogató ontológiaszerkesztő-eszközök összehasonlítását is. A kutatás elméleti háttérének felhasználásával elkészítettem az Oktatási ontológia modelljét, melynek részleteit, szintén a II. fejezet tartalmazza. Az Oktatási ontológia építésének módszertanát és gyakorlati alkalmazását a Tudásmenedzsment tárgy ontológia modelljének példáján keresztül mutatom be. A III. fejezet tárgyalja az adaptív tudásfelmérés kérdéseit. A kapcsolódó szakirodalom elemzését és a már létező módszertanok összehasonlítását követően kidolgoztam egy saját, az oktatási ontológia alapján működő adaptív tudásfelmérési módszert. Szintén a Tudásmenedzsment tárgy példáján keresztül mutatom be a létrehozott tesztrendszer működését. Végül az Oktatási ontológia széles körű alkalmazásának bemutatására a fejezet végén újabb, eltérő megközelítésű vizsgáztatási módszert mutatok be. A IV. fejezetben összefoglalom a kutatás eddigi eredményeit és a további tervezett lépéseket, végül az V. fejezet tartalmazza a Függelékét .



---

## **II. Oktatási ontológia – Elméleti háttér és modellezés**

A kutatás egyik célja, hogy a gazdaságinformatikus szakon megszerzhető kompetenciákon keresztül ontológiai kapcsolatot teremtsen a munkaerőpiac által támasztott követelmények, valamint az oktatott tárgyak tartalma között. Ennek érdekében meg kell vizsgálni, hogy a képzés által közvetített tudás reprezentálására kialakítható-e olyan ontológiai modell, mely az oktatott tárgyak széles köre és eltérő jellege ellenére is alkalmas az egyes területek fogalmainak és a köztük fennálló szemantikai viszonyok ábrázolására. Az első hipotézis alátámasztása érdekében ki kell alakítani a képzés tudásanyagának reprezentálására alkalmas ontológiát, mely egyben megköveteli a módszertani elvárások, ontológia nyelvek és kapcsolódó elméleti háttér elemzését.

### **II.1. Ontológiák – Bevezetés**

#### **II.1.1. Ontológia fogalma**

##### ***II.1.1.1. Előzmények – az ontológia fogalma a filozófiában***

Az ontológia ma már kétségtelenül kulcs fogalom a modern számítástudomány területén és közkedvelt kifejezéssé vált a szemantikus Web és a számítógépes rendszerekhez használt formális ontológiák fejlesztése kapcsán, ugyanakkor nem feledkezhetünk meg róla, hogy maga a fogalom a filozófiából származik.

Az ontológia, mint a tudományelméletnek a létezőt, a létet és alapjait, tulajdonságait vizsgáló ága, a hagyományos értelemben vett metafizika egyik része, egészen az ókori görög filozófiáig nyúl vissza, bár maga a kifejezés csak a XVII. század elején jelent meg.

Arisztotelész megfogalmazása szerint a metafizika tárgya "a létező, mint létező". Másképp megfogalmazva a metafizika tárgya mindaz, ami különbözik a semmitől. Az ontológia metafizika, mert a létezők érzékfeletti (metafizikai) összetevőit és végső alapját (első okát) keresi. Az ontológia tehát a létezővel (on) foglalkozó alaptudomány (logosz).

Az ontológia kifejezést 1613-ban, egymástól függetlenül, két filozófus is megalkotta: Rudolf Göckel (Goclenius) *Lexicon philosophicum* című könyvében, illetve Jacob Lorhard (Lorhardus) *Theatrum philosophicum* című művében (Smith, 2002).

Az ókortól napjainkig számos filozófus (Arisztotelész, Leibniz, Kant, Quine) foglalkozott a metafizika, az ontológiák problémakörével. Az informatikában alkalmazott ontológiák leginkább a filozófia következő területeit elevénítik fel és töltik meg tartalommal: Egyrészt az ontológiákból kiindulva egy olyan arisztotelészi világ feltételezhető, melyben referenciákkal jelenítjük meg a szubsztanciákat (ahol a szubsztancia a dolgokról alkotott fogalmakat jelenti, a referencia pedig egy olyan jel, jelrendszer, szóalak mellyel

---

fizikai valóság egyes egyedeit, fogalmait jelöljük). Másrészt viszont az ontológia úgy is értelmezhető, mint terminológiai jellegű meghatározások halmaza, más szóval egy üres formális rendszer szimbólumai. Ez utóbbi értelmezés alapját quine-i elkötelezettségi tézis adja, amely szerint minden elmélet maga után von egy ontológiát. Quine megközelítése szerint csak azok a dolgok léteznek, amelyek kötött változók értékei lehetnek, tehát az ontológiai elkötelezettség határozza meg, milyen megközelítésben, milyen részletességgel reprezentáljuk a világ egyes részeit. Ez a két filozófiai megközelítés azonban ellentmondásban áll egymással, az egyik kizárja a másikat, s olyan problémákat vetnek fel, melyek a filozófia soha el nem dönthető kérdései közé tartoznak (Ungváry, 2004).

Guarino, aki az ontológiák építésének egyik legnagyobb kortárs tekintélye a filozófiában a nagy kezdőbetűs 'Ontológia', az informatikában a kisbetűs 'ontológia' terminusok használatát javasolja (Guarino et al., 1995).

### ***II.1.1.2. Általános megközelítés***

A tudásábrázolás és a mesterséges intelligencia területén számos definíció született, az ontológia fogalmának meghatározására. A definíciók folyamatosan változtak és fejlődtek, ennek ellenére mai napig nem alakult ki olyan meghatározás, melyet a szakértők széles körben elismernek és alkalmaznak. A következőkben bemutatom, hogy a tudásreprezentáció területén, ahol a fogalomi modellezésre helyezték a hangsúlyt, illetve a mesterséges intelligenciában milyen megközelítések és meghatározások születtek. Az áttekintés célja az ontológiák tulajdonságainak, szerkezetének, felépítésének minél pontosabb megismerése.

#### ***(1) Fogalmi modell, fogalomalkotás oldaláról***

Neches és kollégái elsők között adtak definíciót az ontológia fogalmára a tudásreprezentálás területén, mely így szól:

„az ontológia egy adott témakör szókészletét felépítő alapkifejezések, relációk és olyan szabályok összessége, melyek a kifejezések és relációk összekapcsolásával biztosítják a szókészlet bővítési lehetőségeinek meghatározását” (Neches et al., 1991, p. 40.)

A szerzők célja olyan számítógépes környezet felvázolása, melyben a tudásalapú rendszerek, mesterséges intelligencia alkalmazások és hagyományos szoftverek a tudás szintjén képesek együttműködni a közös ontológiák segítségével. Definíciójuk meghatározza az ontológia felépítésének módját és rávilágít arra, hogy nemcsak az explicit módon meghatározható kifejezések, hanem a belőlük következtethető tudás is része az ontológiának.

Grubertől származik a legtöbbet idézett ontológia-definíció:

"A fogalmi modell (fogalomalkotás) világos és részletes leírása." (Gruber, 1993. p. 199.)

Gruber meghatározását egészíti ki és pontosítja Borst a következőképpen:

„Az ontológia közös fogalmi modell formális leírása.” (Borst, 1997. p. 12.)

Mindkét definíció esetében a fogalmi modell a világ valamely jelenségének absztrakt modelljét jelenti, mely azonosítja a jelenség releváns kompo-

---

nenseit. A világos és részletes leírás a fogalmak és a fogalmakra vonatkozó megszorítások leírását foglalja magába. A formális pedig annyit tesz, hogy ennek a leírásnak számítógép által is olvashatónak kell lennie. Végül a közös kifejezés arra utal, hogy a fogalmi modellnek konszenzuson alapuló tudást kell tartalmaznia. (Studer et al., 1998)

Guarino és kollégái túllépve az előzőekben bemutatott definíciókon meghatározták az ontológiaépítés logikai elmélet kialakításán alapuló módját (Guarino et al., 1995).

Sowa megközelítése szerint az ontológia ott kezdődik, ahol a logika véget ér, melyet a következőkkel egészít ki:

„A logikában az  $\exists$  egzisztenciális kvantor annak jelölésére szolgál, hogy valami létezik. De a logikának önmagában nincs szótára a létező dolgok leírására. Az ontológia az, amely betölti ezt a hiányt. Az ontológia a létező tanulmányozásával foglalkozik, mindenféle olyan – absztrakt és konkrét – egyed tanulmányozásával, amelyből a világ felépül.” (Sowa, 2000)

Ezzel a meghatározással Sowa azt írja le, hogy ontológiának mindössze annyi a feladata, hogy a különböző számítógépes alkalmazásokhoz az adott tárgyterület leírásához szükséges szókinccset biztosítsa. (Egy megfelelő logikai nyelv és a quine-i értelemben vett ontológiai elköteleződések sorozatán keresztül egyre több valós jelenséget tudunk leírni.)

Corcho és munkatársai ugyan nem változtatnak a Gruber féle definíció lényegén, de több szempont egyidejű figyelembevételével pontosabb, jobban alkalmazható megoldást nyújtanak:

„az ontológiák a megegyezésen alapuló tudás általános és formális megragadását, újrahasznosítását, valamint alkalmazások és emberek különböző csoportjai közötti megosztásának biztosítását célozzák meg” (Corcho et al., 2003. p. 44.).

## (2) *Mesterséges intelligencia területén*

A mesterséges intelligencia irodalmába John McCarthy vezette be az ontológia kifejezést, azt elemezve milyen információkat kell figyelembe venni a világ megértéséhez. Megközelítése a filozófiában alkalmazott ontológia-fogalom kibővítésének tekinthető (McCarthy, 1980).

Az elmúlt évtizedekben a mesterséges intelligencia területén alapvetően két, egymáshoz közel álló megközelítés terjedt el: Az ontológia, mint értelmezési szótár feladata, hogy megadja az adott szakterületet leíró kifejezések értelmezését. Ebben az esetben nem a szótár, hanem a fogalomalkotás minősül ontológiának. Ez egyben azt is jelenti, hogy az ontológia szakkifejezéseinek egyik nyelvről a másikra fordításával fogalmi szinten nem változik meg az ontológia, illetve a kifejezések értelme. Ilyen szótár és fogalmi modell azonosítása a szakterület objektumainak és relációinak alapos és körültekintő elemzését kívánja (Chandrasedaran et al., 1999).

Másik jelentése szerint az ontológia feladata az adott szakterület leírása egy általánosan elfogadott és alkalmazott tudáshalmaz és értelmező szótár segítségével. Ebben az esetben az ontológia az értelmező szótár kifejezéseit felhasználva írja körül az adott szakterületre vonatkozó és a tu-

---

dáshalmazban tárolt tényeket. Az ontológia tehát, olyan általános tényeket tartalmazó gyűjtemény, amelyet meg szeretnénk osztani, hogy több szakterületen is alkalmazni tudjuk.

Sántáné Tóth Edit az ágensek közötti kommunikáció támogatásának szempontjából vizsgálja az ontológiákat. Megközelítése szerint egy közös ontológia olyan fogalomgyűjtemény, melynek felhasználásával ágensek kijelentéseket és kérdéseket „cserélhetnek ki” egymás között. Az ontológiai kötelezettség egy olyan szerződés, amelyben a felek vállalják, hogy következetes (koherens) módon, a szerződésben foglaltakkal megegyező (konzisztens) módon használják a közös, megosztott fogalomgyűjteményt, (mely nem igényli tudásbázisaik megosztását). (Sántáné Tóth Edit, 2001)

### **(3) Az ontológiaépítés folyamata, módszertana szerint**

Gyakorlati szempontból hasznos megvizsgálni a különböző kutatás-fejlesztési projektek által alkalmazott megközelítéseket, melyek az építkezési folyamat oldaláról közelítik meg az ontológia fogalmát.

A SENSUS ontológiai megközelítés szerint az ontológiaépítés egyik lehetséges módja nagy méretű ontológiák újrahaználása. Az ontológia pedig nem más, mint hierarchiába rendezett fogalmak halmaza, mely egy szakterület fogalmait gyűjti össze és megadja tudásbázisának vázát. (Swartout et al., 1997) Swartout és kollégái által kidolgozott módszertan szerint létrehozott ontológiák több tudásbázis felépítéséhez is alkalmazhatók, melyek így ugyanazzal a vázzal rendelkeznek. A megközelítés legfőbb előnye abban rejlik, hogy ha különböző rendszereket ugyanazzal az ontológiával építenek fel, akkor azok közös alapstruktúrával fognak rendelkezni, így tudásbázisaik megosztása és a következtetési mechanizmusok is könnyebben elvégezhetők.

A KACTUS projekt keretében az ontológiát valamely tudásbázisból kiszűrt, fogalmak segítségével leírt tudás feltérképezésére szolgáló eszköznek tekintik (Bernaras et al., 1996). Ez a megközelítés is gyakorlati szemléletű, hiszen az ontológiákat mindig valamilyen hasznosítási céllal hozzák létre, mely lehet egy újabb ontológia felépítése vagy szakterületek tudásbázisainak fejlesztése.

Az On-To-Knowledge projekt definícióját érdemes még kiemelni, mely szerint az ontológia feladata a különböző szakterületek információinak azonos struktúrában való ábrázolása (Staab et al., 2001). Ez a megközelítés is gyakorlati szempontokat emel ki, így az ontológiában foglalt tudás többszöri felhasználásának, alkalmazások és csoportok közötti megosztásának lehetőségeit kívánják biztosítani módszerükkel.

## **II.1.2. Felhasználási területek**

### **II.1.2.1. Tudásmenedzsment**

A folyamatos verseny, a felhasználói igények állandó változása, a termékek bonyolultságának fokozódása, mind a tudás kezelésének szisztematikus közelítését igényli. A tudásmenedzsment feladata a szervezet tudásfolyamatainak kezelése és irányítása. A tudásfolyamatok értelmezése

---

szervezetenként eltérő lehet. Ez jelentheti azt, hogy szervezetenként más-más feladatokból épülnek fel a folyamatok, de az is előfordulhat, hogy az eltérések csak látszólagosak és az alap fogalmi-modell tulajdonképpen megegyezik. Az információtechnológia által támogatott tudásmenedzsment megoldások akkor képesek megfelelően kezelni az adott szervezeti folyamatokat, ha az újrafelhasználható tudás és információ kezelése biztosított. Ez pedig nem képzelhető el csak úgy, ha a szóban forgó tudás és információ felhasználását, kezelését (minden szereplő által egyformán értett) közös ontológia támogatja.

Az ontológia-alkalmazás legfőbb céljai közé a tudásmegosztás, a tudás megszerzése és a tudásábrázolás támogatása tartozik, mely nyilvánvalóvá teszi tudásmenedzsmenttel való szoros kapcsolatát is. (O'Leary, 1998., Maedche, 2001) Ezek a problémák leginkább a tudás mérnökség (knowledge engineering), a tudásalapú rendszerek fejlesztéséhez kapcsolódnak. Az ontológiák alkalmazása segíthet a tudásalapú rendszerek fejlesztése során adódó olyan kihívások megoldásában, mint a tudás megszerzéshez kapcsolódó magas költségek csökkentése vagy a konszenzuson alapuló tudás hiányából adódó problémák kezelése.

Intelligens rendszerekkel összefüggésben először természetes nyelvek kezelésével kapcsolatos problémák (kétértelműség, tartalom azonosítása) támogatására alkalmazták széles körben az ontológiákat. A nyelvi ontológiák leginkább a gépi fordítás és elemzés támogatását célozták meg. (Corbett, 2001) Ilyen absztrakt ontológiák két legismertebb példája a CyC és a WordNet.

Az ontológiák leginkább azzal járulnak hozzá a tudásmenedzsment tevékenységek fejlesztéséhez, hogy kibővítve a korábbi tudásmenedzsment technológiák által alkalmazott módszereket, lehetőség nyílik újabb tudásterületek feltárására. A tudásmegosztásnak és a tudás megszerzésének ontológiai támogatásával, a tudás fellelése és összegyűjtése nem fog szűk keresztmetszetet jelenteni. A tudás strukturálásával megkönnyítik a szakterületek feltárását, a szakértelem újrafelhasználását.

Az ontológiák tudásmegosztásban játszott szerepének egyik legjobb példája a közösségi Web portálok támogatása. Ezek a portálok olyan felhasználókat fognak össze, akiknek azonos az érdeklődési területük és az alkalmazott fogalomrendszerük egyaránt. Magától értetődik, hogy a közösség konszenzuson alapuló tudásának explicit leírására ontológiát alkalmazunk, és segítségével reprezentáljuk az adott tudásterületet a közösség portálján. Az ontológia használatának közvetlen előnye pedig az, hogy lekérdezéseken keresztül lehetőségessé válik az intelligens visszakeresés, illetve következtető gépek alkalmazása is.

Az ontológiák azonban segítséget nyújthatnak az eltérő nézetekkel bíró egyének közötti tudásmegosztásban is. A felhasználók széles köre akkor lesz képes keresni és lekérdezni tudásbázisokat, portálokat, ha az információk több nézetben is reprezentálva vannak. Ekkor az ontológiák feladata a Web alapú alkalmazások szemantikai infrastruktúrájának meghatározása lesz.

---

### **II.1.2.2. Rendszerfejlesztés**

Guarino megközelítése szerint az információrendszerek és ontológiák kapcsolata, az ontológia-alkalmazás hatásai két dimenzió elemzésén keresztül ragadhatók meg. Egyrészt meg kell vizsgálni, hogy az ontológia milyen módon hat az információrendszerek egyes részeire (alkalmazási program, információforrás (adatbázisok, tudásbázisok), felhasználói felület), másrészt elemezni kell az ontológia-alkalmazásnak a rendszer fejlesztése, illetve működtetése során játszott szerepét. Guarino, tehát nemcsak az ontológiák rendszerfejlesztésben betöltött szerepét, hanem a rendszerek karbantartásához, fejlesztéséhez nyújtott támogatását is tanulmányozza.

A rendszerfejlesztés folyamatában betöltött szerepük vizsgálatakor abból a feltételezésből indul ki, hogy már kész ontológiák állnak rendelkezésünkre. Ezek az ontológiák lehetnek magas szintű vagy szakterületi és feladat ontológiák. A szakterületi / feladat ontológiák két módon használhatók fel információrendszerek fejlesztésében. Hagyományos információrendszerek esetében az ontológia által tartalmazott szemantikus tartalom kerül beépítésre a rendszer egyes elemeibe. Az úgynevezett ontológia-alapú rendszerek esetén, pedig a fejlesztés eredményeként egy újabb rendszerkomponens jön létre, az úgynevezett alkalmazási ontológia. Az ontológia-alkalmazás előnye abban rejlik, hogy az újrahazsnosítás magasabb szintje lesz elérhető, ráadásul azt is lehetővé teszi, hogy a fejlesztők a szakterület, az adott feladat szerkezetére, problémáira és ne az implementáció kérdéseire koncentráljanak. A szakterületi ontológiákkal szemben a magas szintű ontológiák nem a rendszer részét fogják képezni, hanem az elemzési folyamatok minőségének javításában játszanak fontos szerepet, elősegítve az alkalmazási ontológia felépítését. Végül azt is látni kell, hogy az ontológiák nem csak új rendszerek fejlesztése esetén, hanem már meglévő rendszerek karbantartása és továbbfejlesztése esetén is hatékonyan alkalmazhatók (Guarino, 1998).

A rendszer működtetése során az ontológia elsődleges feladat a szoftver ágensek közötti kommunikáció támogatása.

Guarino szintén megvizsgálta, hogy az egyes rendszerkomponensek (alkalmazási programok, adatbázisok, felhasználó felületek) hogyan tudják használni az ontológiákat. Adatbázisok fejlesztése során a követelményspecifikáció és a fogalmi modell kialakításában, valamint a különböző forrásokból származó információk egységesítésében játszik fontos szerepet az ontológia. Az adatbázisok működtetésében, pedig a lekérdezések támogatása, hatékonyabbá tétele az ontológia fő feladata.

A felhasználói felületek kialakítását, formázását, a felületen való keresést és lekérdezést támogathatja az ontológia.

Az alkalmazási programok statikus részeinek (az adatbázisokban explicit módon nem kifejezett tudást tartalmazó részek) fejlesztéséhez, valamint az implicit tudás reprezentálásában kapnak szerepet az ontológiák. Segítségükkel az alkalmazási programok karbantartása, bővítése könnyebbé, rugalmasabbá válik. (Guarino, 1998)

---

### **II.1.2.3. Szemantikus Web**

Az ontológiák fontos szerepet játszhatnak abban is, hogy a jelenleginél intelligensebb módon kereshessünk a Weben megtalálható honlapok és információk rengetegében. Ezeknek az információknak jelentős része természetes nyelvű szöveg, vagy olyan egyéb kép, videó vagy hanganyag, melyek jelentése a számítógépek számára nem értelmezhető. A jelenleg széles körben alkalmazott HTML nyelvet és a hozzá kapcsolódó technológiákat arra tervezték, hogy emberek számára közvetítsen információt. Az intelligens Internet megvalósítására irányuló legismertebb elképzelés, a szemantikus Web (Berners-Lee et al., 2001) megközelítés, mely két alapötletre épül. Az egyik ötlet szerint az Internetes erőforrásokhoz metaadatokat<sup>6</sup> kell kapcsolni. Másképp fogalmazva ez azt jelenti, hogy az erőforrásokhoz (a tárolt adatokhoz) pontosan definiált jelentést (szemantikát) társítunk. Az Internetes erőforrások fogalmát, ugyan tágan értelmezi ez a megközelítés, de éppen ezért alapvető követelmény, hogy metaadatokat társítsunk mindenhez, ami egyedileg azonosítható. A Szemantikus Web másik fontos tulajdonsága, hogy a metaadatok segítségével következtetni tudjunk. Az ontológiák pedig a szemantika meghatározásában, a következtetések támogatásában játszanak fontos szerepet.

## **II.2. Ontológia felépítése, módszertanok**

Az ontológia életciklus egyik legfontosabb szakasza az ontológia felépítése, melynek eredményességét alapvetően meghatározza az alkalmazott módszertan. A megvalósítandó oktatási ontológia esetében is meg kell fogalmazni, le kell írni az építkezési folyamat lépéseit, az alkalmazni kívánt módszereket. Mindehhez hasznos segítséget nyújt az elterjedt, széles körben elfogadott módszertanok áttekintése, összehasonlító elemzése. A módszertanok többségét gyakorlati ismeretek birtokában, ontológiák készítésének tapasztalatai alapján dolgozták ki, melyek közül a következőket elemzem és hasonlítom össze: Uschold és King, Grüninger és Fox, CommonKADS KACTUS projekt, SENSUS, On-To-Knowledge, Methontology.

Az ontológiaépítés kérdéseinek részletes vizsgálatát továbbá az is indokolja, hogy a tartalmi feltöltés eredményességét, a tudásreprezentálás minőségét, a továbbfejlesztési, összekapcsolási lehetőségek körét is meghatározza az alkalmazott módszertan.

### **II.2.1. Grüninger és Fox féle módszertan**

A Michael Grüninger és Mark S. Fox által kidolgozott ontológiaszerkesztési és -értékelési módszertant a TOVE<sup>7</sup> (Toronto Virtual Enterprise) projekt keretében dolgozták ki. A projekt a vállalati modellezést elősegítő, elsősorban ipari területen előállított ontológiák felépítésére összpontosított. A fő cél a szervezeti viselkedés (tevékenységek, állapotok, okozati összefüggések, idő) és az általa befolyásolt objektumok megragadása, ábrázo-

---

<sup>6</sup> Metaadatnak nevezzük az olyan adatot, mely adat egy adatról.

<sup>7</sup> <http://www.eil.utoronto.ca/enterprise-modelling/tove/>

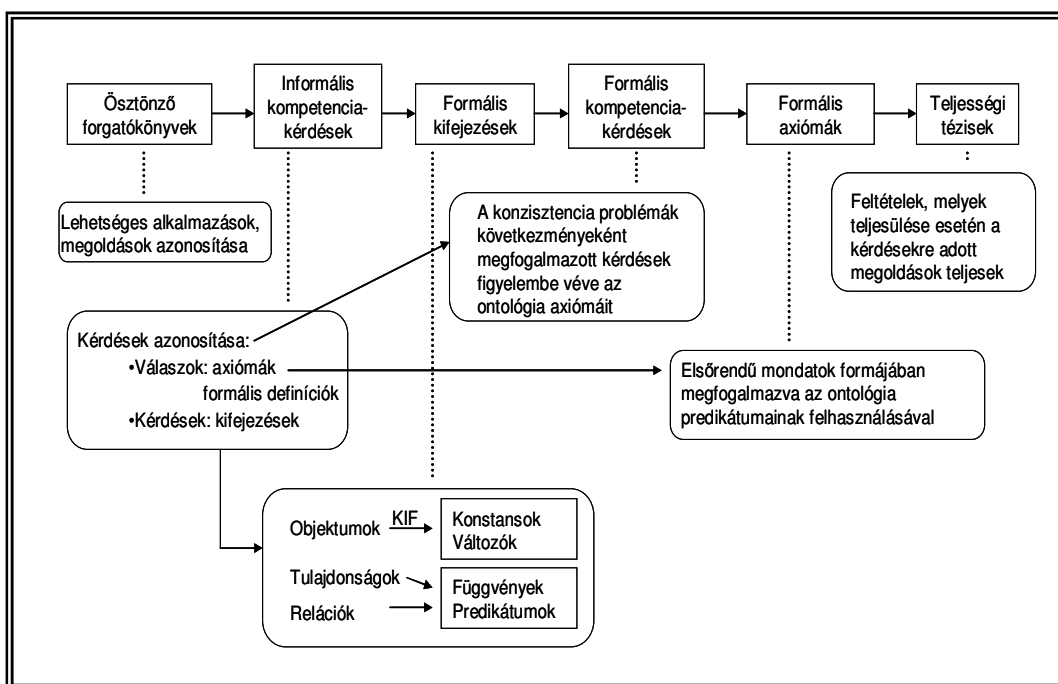
---

lása. Megközelítésük szerint az ontológiaszerkesztés a következő hat fő lépésből áll:

- 1) Ösztönző forgatókönyvek összegyűjtése: olyan forgatókönyvek, esetleírások összegyűjtése, melyek még nem kezelt szervezeti problémákat tárnak fel, esetlegesen irányt mutatnak a megoldásokra.
- 2) Informális kompetenciakérdések megfogalmazása: az ontológia kompetenciájával szemben állított követelményeket megtestesítő kérdéshalmaz forgatókönyvszerű leírását jelenti. A létrehozandó ontológiának meg kell tudnia válaszolni ezeket a (természetes nyelven leírt) kérdéseket.
- 3) Formális kifejezések kialakítása: a terminológia kialakítása két lépésben történik:
  - a. Az informális kifejezések (az objektumok, tulajdonságaik és relációik összegyűjtése) az informális kompetenciakérdésekből.
  - b. A terminológia formális nyelven (pl.: KIF-ben) történő megadása.
- 4) Formális kompetenciakérdések megfogalmazása: a formálisan leírt terminológia alapján az ontológiával szemben támasztott követelményeket is formalizáljuk.
- 5) Formális axiómák megfogalmazása: az axiómák megszorítások meghatározásával definiálják az ontológia kifejezéseit, megadják a kifejezések szemantikáját (elsőrendű logikai apparátus segítségével).
- 6) Teljességi tézisek meghatározása: az ontológia kiértékeléséhez meg kell határozni azokat a feltételeket, melyek teljesülése esetén az ontológia hatáskörébe tartozó kérdésekre adott válaszok, megoldások teljesnek tekinthetők. (Grüninger és Fox, 1994; Grüninger és Fox, 1995)



A módszertan fő lépéseit és kapcsolódó részfeladatokat az 1. ábra foglalja össze.



1. ábra: A TOVE ontológia szerkesztése és értékelése  
(OntoWeb, 2002a, p. 15.)

## II.2.2. Uschold és King féle módszertan

A Vállalati ontológia fejlesztése során szerzett tapasztalatok alapján vázolták fel az első módszertanra vonatkozó megközelítésüket (Uschold és King, 1995, Uschold és King, 1996a). Később a TOVE projekt tapasztalatait felhasználva dolgozták ki teljes, átfogó módszertanukat, mely ennek megfelelően a következő módon épül fel:

- 1) Célok azonosítása: az ontológia fejlesztési és felhasználási céljainak megfogalmazása, a formalizálás fokának meghatározás.
- 2) Az ontológia „megragadása”: specifikáció összeállítása, mely behatárolja a lefedni kívánt terület. Ez pedig a következő lépések végrehajtását igényli:
  - a. a kulcs fogalmak és relációk azonosítása,
  - b. a fogalmak és relációk egyértelmű szöveges definícióinak megadása,
  - c. a fogalmakat és relációkat jelölő kifejezések azonosítása.
- 3) Formalizálás: a specifikációban meghatározott kifejezések formális nyelven történő leírása.
- 4) Integrálás létező ontológiákkal: mind a specifikáció, mind a formalizálás során meg kell vizsgálni más ontológiák beillesztésének lehetőségét.

- 
- 5) Kiértékelés: a kitűzött célok teljesülésének, az ontológia teljességének ellenőrzése előre meghatározott (általános és ontológia specifikus) kritérium rendszer alapján.
  - 6) Dokumentáció: az ontológia dokumentálási irányvonalainak megadása, mely eltérő lehet az ontológia típusa és célja szerint. (Uschold, 1996b)

### **II.2.3. CommonKADS módszertan**

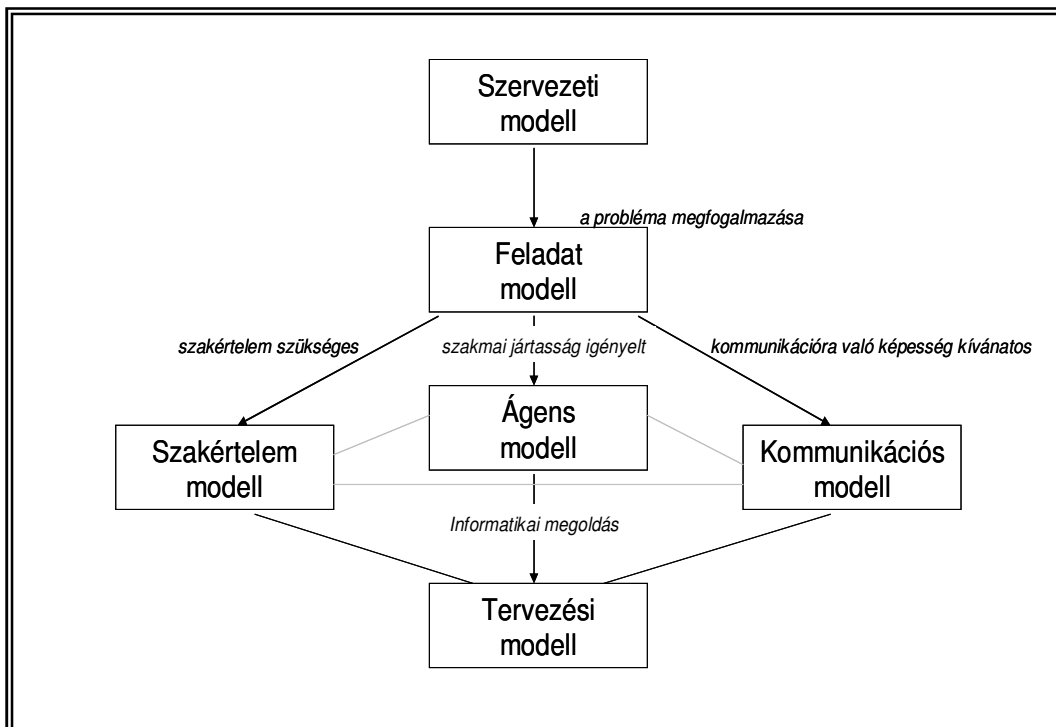
A CommonKADS módszertan (Schreiber et al., 1998) nem tartozik a szűk értelemben vett ontológiafejlesztő módszertanok közé. Alapvetően tudásintenzív rendszerek fejlesztési szabványának tekinthető, mely Fensel megközelítésében egyben ontológia szabvány is (Fensel, 1998). A KACTUS projekt keretében már kifejezetten a CommonKADS módszertan ontológiafejlesztési lehetőségeire összpontosítottak. A CommonKADS többek között azért válhatott az ontológiafejlesztés módszertanává is, mert alapvető tervezési elvei (a modularitás, áttervezhetőség és újrahasznosíthatóság (Schreiber et al., 1995, Wielinga et al., 1994)) általánosan elfogadottak az ontológiák tervezésével foglalkozó körökben is. Az újrafelhasználhatósági elveknek megfelelően egy ontológia felépíthető már létező ontológiák felhasználásával is. Két ontológia közötti leképezés pedig a következő két módon valósítható meg:

- 1) A leképezett ontológia kifejezéseinek szemantikájában nincs változás.
- 2) A leképezett ontológia szemantikájában változás csak akkor következik be, amikor annak értelmezésére sor kerül, interpretálása a másik ontológia értelmében szükségessé válik (Molnár, 1999).

A módszertan alapját jelentő modellkészlet a következő hat modell típust különbözteti meg:

- 1) Szervezeti modell: célja a működési környezet leírása.
- 2) Feladat modell: azon feladatok halmazát írja le, amely szerint működnek vagy működni fognak. A modell elemzi a feladatokat befolyásoló valamennyi tényezőt (a bemeneteket és kimeneteket, a feltételeket és a teljesítés követelményeit, az erőforrásokat is).
- 3) Ágens modell: a feladatmodellben azonosított tevékenységeket végrehajtó ágenseket, rendszerszereplőket, illetve azok jellemzőit írja le az ágens modell.
- 4) Kommunikációs modell: a rendszer ágensei, szereplői közötti kommunikáció, kölcsönhatás, információcsere leírása.
- 5) Szakértelem modellje: a feladatok teljesítése során alkalmazott tudás típusának és struktúrájának részletes leírása.
- 6) Tervezési modell: a technikai rendszer-specifikációt adja meg az eddigi modellekben leírtak figyelembevételével. (Schreiber et al., 1998)

A 2. ábra a megközelítés modelljei közötti kapcsolatokat ábrázolja.



2. ábra: A CommonKADS modelljei közötti kapcsolat (Molnár, 2003)

A CommonKADS szakértelem modelljének specifikálására egy félig formális nyelvet is kidolgoztak, (amely egyben az ontológia meghatározását is magában foglalja). Ez a CML, a Fogalmi Modellezés Nyelve (Conceptual Modeling Language).

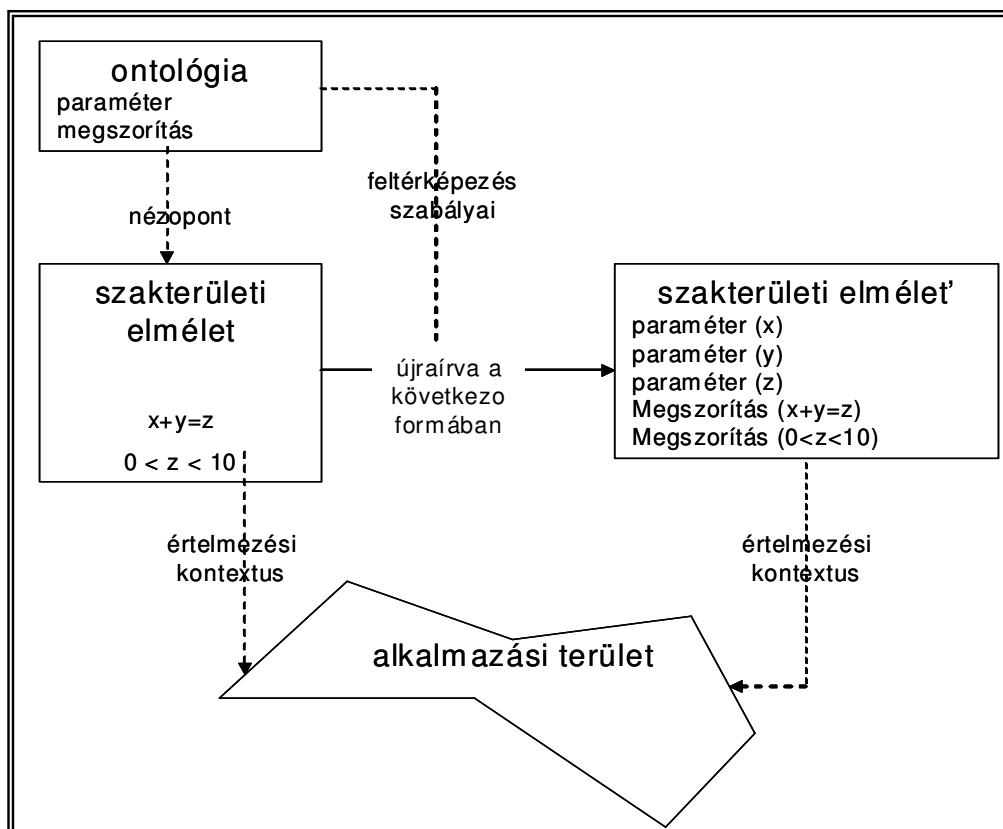
A CommonKADS módszertan számára azonban egy teljesen formális leíró nyelvet is kifejlesztettek, melyet FML-nek vagy  $(ML)^2$ -nek hívnak. Az  $(ML)^2$  elsőrendű predikátum kalkulusra épülő, meglehetősen bonyolult és összetett nyelv (Molnár, 2003).

#### II.2.4. A KACTUS projekt keretében kidolgozott módszertan

A CommonKADS a tudásalapú rendszerek (melyekben az ontológiák fontos szerepet játszanak) fejlesztésének módszertana. A KACTUS projekt a CommonKADS módszertanból az ontológiafejlesztésre koncentrált. Megközelítésük szerint az ontológiaszerkesztést mindig a fejlesztendő alkalmazások határozzák meg. Minden alkalmazásfejlesztés szükséges előfeltétele az adott terület tudását ábrázoló ontológia felépítése (Schreiber et al, 1995; Bernaras et al, 1996). Ezt az ontológiát pedig a következőképpen kell felépíteni:

- 1) Specifikáció megfogalmazása a konkrét alkalmazásnak megfelelően.
- 2) Tartalmi feltöltés.
- 3) Rendszerezés és szerkezeti finomítások.

Amennyiben már több ily módon létrejött ontológia is a rendelkezésünkre áll, a következő lépés az ontológiák egyesítése illetve egységesítése lehet (3. ábra).



3. ábra: A KACTUS megközelítés (Schreiber et al., 1995)

### II.2.5. Sure-Studer féle On-To-Knowledge módszertan

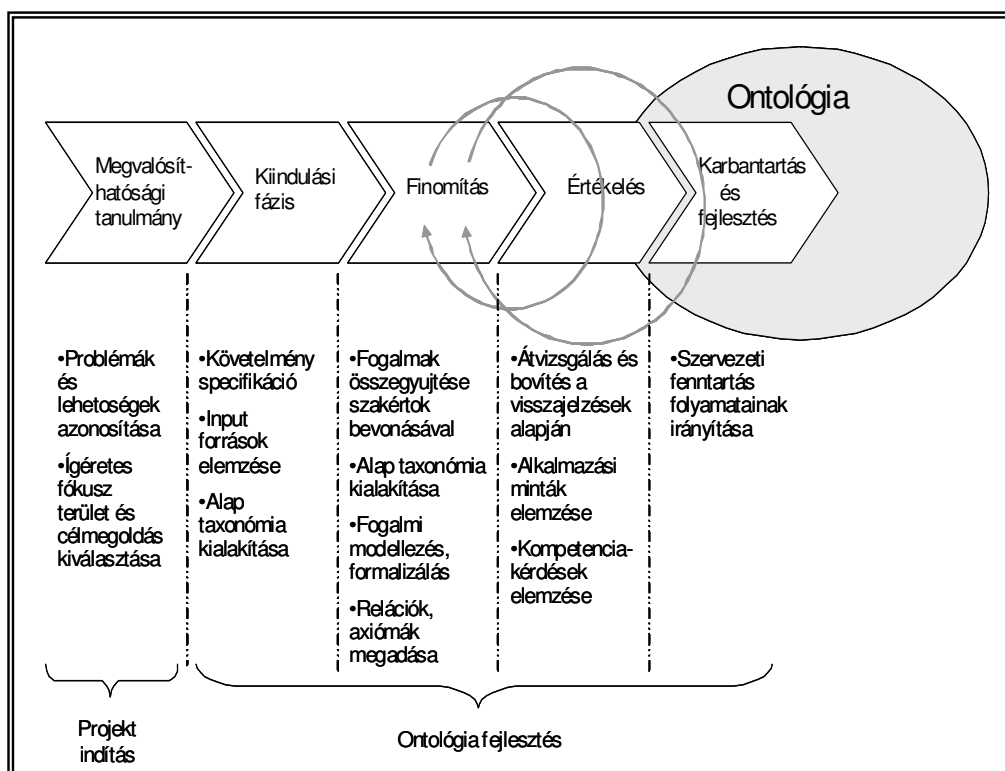
Az On-To-Knowledge<sup>8</sup> módszertan szintén a CommonKADS módszertanra épít, annak egy módosított változatát veszi alapul. Más megközelítésekkel ellentétben nem magára az ontológiára koncentrálnak, hanem az alkalmazásvezérelt ontológiafejlesztésre helyezi a hangsúlyt. Elsősorban vállalkozások számára kíván iránymutatást adni a tudáskezelés eszközeinek használatára. A ontológiafejlesztés javasolt lépései a következők:

- 1) Megvalósíthatósági tanulmány: nemcsak az alkalmazott technológia, hanem más szervezeti, környezeti tényezők is meghatározzák, hogy a fejlesztendő tudásmenedzsment rendszer mennyire lesz sikeres. Éppen ezért a tényleges fejlesztést megelőzően olyan megvalósíthatósági tanulmányt kell készíteni, mely szélesebb szervezeti kontextusban azonosítja a befolyásoló tényezőket, feltárja az esetleges problémákat és lehetőségeket.
- 2) Kiindulási fázis: a kiindulási fázisban kerül sor az ontológia követelményspecifikációját tartalmazó dokumentum elkészítésére. A dokumentumnak tisztázni kell mi az ontológia célja, tárgyterülete, milyen alkalmazásokat fog támogatni, milyen tudásforrásokat fog fel-

<sup>8</sup> <http://www.ontoknowledge.org/>

használni, meghatározza a potenciális felhasználók körét, felhasználói forgatókönyveket ír le és kompetenciakérdéseket fogalmaz meg. Szintén ebben a szakaszban kerül sor az esetlegesen újrahasználatos ontológiák részletes vizsgálatára.

- 3) **Finomítás:** A finomítási szakasz célja az alkalmazásorientált ontológia kialakítása. Több alszakaszra bontható, melynek során először a kiindulási szakaszban meghatározott fogalmak informális taxonómiáját kell felállítani, majd az elsődleges váz finomítására, bővítésére kerül sor szakértők bevonásával, végül az ontológia formális leírását kell megadni.
- 4) **Értékelés:** Célja a létrehozott ontológia hasznosságának elemzése. Az értékelés során meg kell vizsgálni, hogy az ontológia megfelelő követelményspecifikációban megfogalmazott elvárásoknak, képes-e megválaszolni a kompetenciakérdéseket és sor kerül az alkalmazási környezetben való tesztelésére is. Az így kapott visszajelzések alapján az ontológia tovább finomítható, pontosítható. Az értékelés szoros kapcsolatban van a finomítás szakaszával, ugyanis ez a két lépés mindaddig ciklikusan ismétlődik, míg a célonológia nem éri el a kívánt színvonalat.
- 5) **Karbantartás és fejlesztés:** Az ontológiával szembeni elvárások folyamatosan változhatnak a valós világban bekövetkező változásokat visszatükrözve. Éppen ezért elkerülhetetlen az ontológia folyamatos karbantartása, amely alapvetően egy szervezeti folyamat. Szigorú szabályokat kell megfogalmazni arra vonatkozóan, hogy milyen időközönként, hogyan és kinek van lehetősége a változtatásokra, hogy milyen módon történjen a verzióváltás. (Staab et al, 2001).



4. ábra: Sure-Studer féle ontológiafejlesztés folyamata (Staab et al., 2001)

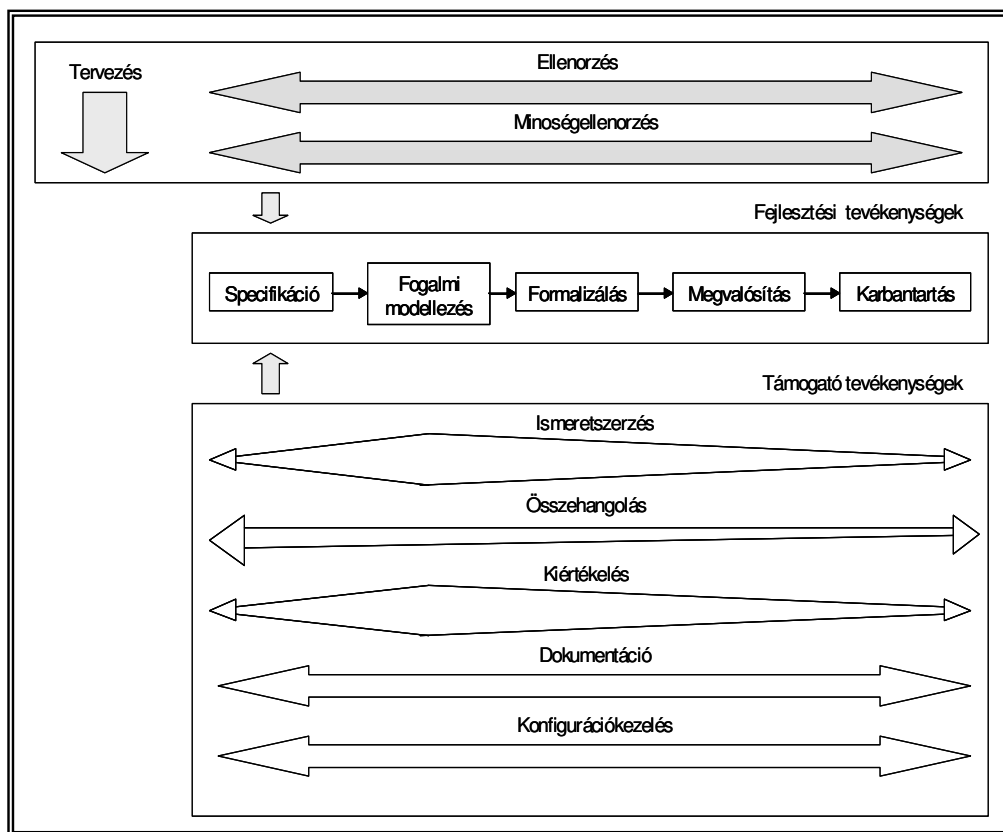
---

## II.2.6. Methontology

A Methontology nem csupán az ontológiaépítés folyamatának lépéseit részletesen leíró módszertan, hanem olyan keretrendszer is, mely meghatározza az ontológia életciklus szakaszait és iránymutatást ad az egyes szakaszokban elvégzendő feladatokra és alkalmazandó technikákra vonatkozóan is. Az ontológia életciklushoz kapcsolódóan a tevékenységek három nagy csoportját különböztetik meg:

- 1) Menedzsment tevékenységek: A tervezés ellenőrzés és folyamatos minőségbiztosítás feladatait foglalja magába.
- 2) Fejlesztési tevékenységek: Az ontológiaépítés szempontjából legfontosabb feladatokat részletezi, melyek a következők:
  - a. Specifikáció: az ontológia céljának megfogalmazása mellett meghatározásra kerül az ontológia formalizáltságának foka, a lehetséges felhasználók köre, az ontológia kiterjedése.
  - b. Fogalmi modell létrehozása: a szakterület kifejezéseit félig formális ábrázolási móddal adja meg. A szakterület szempontjából legfontosabb fogalmak, példányok összegyűjtése után a fa struktúrába, taxonómiába rendezi a fogalmakat és ad hoc relációkat határoz meg közöttük.
  - c. Formalizálás / Megvalósítás: az ontológiát formálisan leírják egy adott nyelven (általában a módszertanhoz kidolgozott szerkesztő eszköz (ODE) ezt automatikusan megvalósítja).
  - d. Karbantartás: a leírt szakterület változásait folyamatosan figyelemmel kell kísérni, és be kell építeni az ontológiába. A módszertan lehetőséget ad mind a folyamatos, mind az időszaki akadálytalan aktualizálásra.
- 3) Támogató tevékenységek: a fejlesztési tevékenységekkel párhuzamosan, azokat kiegészítve kerülnek végrehajtásra.
  - a. Ismeretszerzés, ismeretelemzés: a specifikációt és a fogalmi modell létrehozását megelőzően, vagy azokkal párhuzamosan folyhat. Az ismeretszerzés módjait tekintve a szakértőkkel folytatott interjúk és szövegek elemzésének fontosságát emeli ki a módszertan, de bármilyen más eszköz is alkalmazható.
  - b. Összehangolás: más ontológiák alkalmazásának, más ontológiákból származó fogalmak beillesztésének lehetőségét is meg kell vizsgálni az építkezés folyamán, bár ennek lépései nincsenek részletesen kidolgozva a módszertanban.
  - c. Kiértékelés: A kiértékelés hangsúlyos szerepet kap a módszertanban. Az ismeretbázisú rendszerek helyességének ellenőrzésére alkalmazott technikákat (verifikáció, validáció) veszi át és azok az ontológia egész életciklusát végigkísérik.
  - d. Dokumentáció: az egyes tevékenységekhez kapcsolódó, azokat részletesen ismertető dokumentáció összeállítását jelenti.

- e. Konfigurációkezelés: a verziókövetéshez kapcsolódó feladatokat foglalja magába. (Gómez-Pérez et al, 1996, Fernández et al, 1997) (5. ábra)

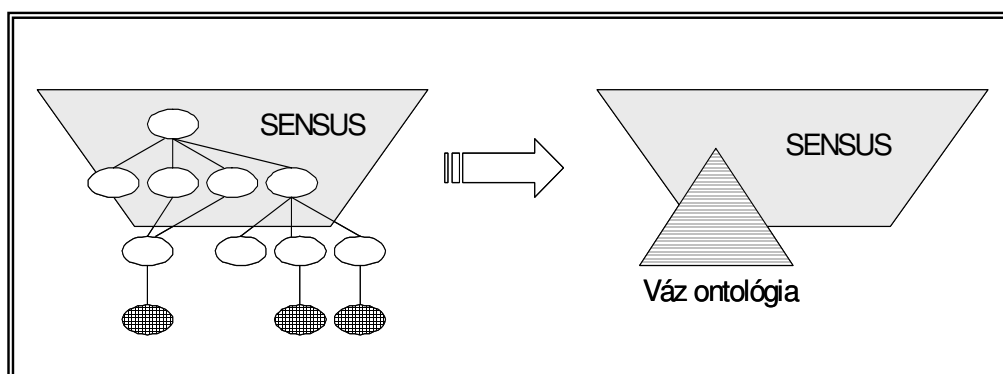


5. ábra: A Methontology ontológia-életciklus (Corcho et al., 2003)

### II.2.7. Sensus

A SENSUS módszertan az ontológiaépítést ciklikus folyamatnak tekinti és abból a feltételezésből indul ki, hogy ha két tudásbázist közös ontológiára építenek, akkor a két tudásbázis közötti tudásmegosztás sokkal hatékonyabb és kézenfekvőbb, mivel azonos alapstruktúrával rendelkeznek. A SENSUS megközelítése szerint első lépésben váz ontológiát kell felépíteni, mely általános (magas, közép szintű) és semmiképpen sem szakterület specifikus kifejezéseket tartalmaz. Ebben az általános ontológiában a fogalmakat hierarchikus relációkkal rendszerezik. Emellett szemantikus háló mentén is rendeződnek, amelyben különböző relációkkal kapcsolódnak egymáshoz. A hierarchia csúcsán helyezkednek el az absztrakt fogalmak, melyek nyelvi általánosításokat tükröznek vissza. Ugyanakkor az alsóbb szinten elhelyezkedő kifejezések sokkal inkább hétköznapiak. Mivel szakterülete nincs meghatározva, ezért a váz ontológiának nagyon tágnak kell lennie. A SENSUS fejlesztői a saját általános ontológiájukat más felső szintű ontológiák egyesítésével hozták létre: Penman Upper Model, WordNet és (angol, spanyol és japán) elektronikus szótárak szemantikus kategóriáinak összevonásával. Ma már több mint 70000 fogalom rendszere az általános ontológia.

A következő lépés a megfelelő szakterületi ontológia kialakítása. A SENSUS fejlesztői úgy találták, nem lenne praktikus a SENSUS ontológiából kiindulni és egyszerűen hozzákapcsolni az adott terület fogalmait, mivel a több, mint 70000 fogalom nagy része lényegtelen az adott szakterület szempontjából. Ehelyett a szakontológia az általános ontológia szűkítésével jön létre a következő módon: Az építkezés első lépésében meghatározzák a szakterület kulcsfogalmait, azaz a magfogalmakat (seed terms), melyet manuálisan hozzákapcsolnak az általános ontológiához. A következő lépésben a magfogalmak és SENSUS gyökere közötti úton található valamennyi fogalmat is a szakontológia részévé tették. Végül azoknál a csomópontoknál, amelyeken számos útvonal ment keresztül eldöntik, hogy a csomópont alatti teljes fa részét képezze-e az új ontológiának vagy sem (Swartout et al, 1997) (6. ábra).



6. ábra: A Sensus megközelítés (OntoWeb, 2002a, p. 21.)

### II.2.8. Az ontológiaépítési módszertanok összehasonlító elemzése

Számos tanulmány foglalkozik a fent bemutatott módszertanok összehasonlításával, a megfelelő szempontrendszer felállításával. Az egyes tanulmányokban kialakított kritériumrendszerek között vannak átfedések, különbségek egyaránt, melyek együttesen igen átfogó képet adnak a módszertanok erősségeiről, gyengeségeiről. Éppen ezért nem célom új szempontrendszer felállítása, hanem egy valamennyi szempontot átfogó összehasonlítás elvégzése.

Az elsők között jelent meg Fernández López írása, melyben a kiforrottnak még nem tekinthető ontológiafejlesztési módszertanok problémáinak, hiányosságainak feltárására törekedett. Megközelítésében az IEEE Szoftveréletről Folyamatfejlesztési Szabványa (Standard for Developing Software Life Cycle Processes) alapján alakította ki kritériumrendszerét, egyrészt abból a megfontolásból kiindulva, hogy a szoftverfejlesztés olyan terület, mely érett, jól kidolgozott módszertanokkal rendelkezik, tehát az ott szerzett tapasztalatokat érdemes figyelembe venni, másrészt az IEEE által megadott szoftver definíció alapján az ontológia is bizonyos szoftvertermékek részének tekinthető. A szabványnak való megfelelés vizsgálatával arra kapunk választ, hogy az egyes módszertanok mennyire részletesen foglalkoznak az ontológia életciklus egyes szakaszaival. E mellett a szerző



---

általános szempontrendszert is felállít, melyben többek között a módszertan részletezettségét, a formalizmusra tett ajánlást, a fogalomrendszer felépítésének módját, a felépítést segítő eszközök körét, az ontológia és az esetlegesen ráépülő alkalmazások kapcsolatát elemzi (Fernández López, 1999).

Fernández López kritériumrendszerét tovább finomították, pontosították, és elemzését kiterjesztették az újonnan megjelent módszertanokra is az OntoWeb<sup>9</sup> projekt keretében kidolgozott tanulmányban (OntoWeb, 2002a).

A legtöbb szempontot magába foglaló elemzést, mely a módszertanok felépítését, az általuk nyújtott tényleges lehetőségeket, a gyakorlati alkalmazás kérdéseit a legrészletesebben tárgyalja, a MEO<sup>10</sup> projekt keretében született meg (MEO, 2005a). A következőkben az összehasonlító elemzések eredményeinek értékelését ez utóbbi tanulmányban meghatározott sorrendet követve végzem el, kiemelve az oktatási ontológia szempontjából fontos aspektusokat.

### ***II.2.8.1. Módszertan kidolgozottsága, részletezettsége***

Az ontológiaépítési folyamat lépéseinek pontos, részletes leírása elősegíti a módszertan használatát és jó minőségű munka elvégzését. A folyamatleírás megértésének pedig alapfeltétele az ontológia elemeinek, alapfogalmainak precíz definiálása is. Kutatásomnak ugyan nem célja a teljes oktatási ontológia felépítése, azonban ki kell választani, testre kell szabni azt a módszertant, melynek segítségével átfogó, koherens ontológiát tudunk létrehozni, még abban az esetben is, ha a modellezni kívánt tudásterületeket más-más szakértők dolgozzák fel.

A folyamatleírás részletezettségét, kidolgozottságát tekintetve igen eltérők az egyes módszertanok. A bemutatott megközelítések közül csak a Methontology-ról mondható el, hogy részletesen leírja egy viszonylag teljes ontológia kidolgozásának a lépéseit, és hogy világosan meghatározza a módszertan alapfogalmait és a fogalmak közötti relációkat. Ugyan a SENSUS és On-To-Knowledge módszertan is megfelelő részletességgel írja le az ontológiaépítés egyes szakaszait, azonban egyik sem definiálja az ontológia alapelemeit (fogalom, reláció, példány, tulajdonságok, formális axióma stb.), míg a TOVE módszertan esetében éppen fordított a helyzet.

### ***II.2.8.2. Ontológia-életciklus***

Az értékelés fontos kérdése, hogy az adott módszertan az ontológiaépítést egy átfogó folyamat részének tekinti-e, azaz kitér-e az ontológia-életciklus egyes szakaszainak részletezésére, a végrehajtandó feladatok, tevékenységek meghatározására. Az összehasonlító tanulmányok mindegyike érinti ezt a területet, a legrészletesebb elemzéseket az OntoWeb (OntoWeb, 2002a) és Corcho (Corcho et al, 2003) tanulmányában találjuk. A korábban már említett szoftverfejlesztési, ismeretábrázolási módszertanok által definiált szakaszok meglétét vizsgálták. A bemutatott módszer-

---

<sup>9</sup> OntoWeb: Ontology-based Information Exchange for Knowledge Management and Electronic Commerce - <http://www.ontoweb.org/>

<sup>10</sup> MEO Magyar Egységes Ontológia - [http://ontologia.hu/project/project\\_MEO/](http://ontologia.hu/project/project_MEO/)

---

tanoknak azonban egyike sem tekinthető teljesen kiforrott, érett modellnek. Az általános projektmenedzsment tevékenységek elvégzését gyakorlatilag csak az On-To-Knowledge módszertan javasolja, de a végrehajtás módját, az elvégzendő feladatok körét már ez sem részletezi. Az ontológiaépítés, -fejlesztés tevékenységeit a Methontology módszertan írja le a legkörültekintőbben. A fejlesztést követő tevékenységek (bevezetés, üzemeltetés, támogatás, karbantartás, üzemén kívül helyezés) tekintetében valamennyi módszertannál jelentős hiányosságokat fedezhetünk fel. Csak a már korábban is kiemelt két módszertan (Methontology, On-To-Knowledge) foglalkozik a fenntartás, karbantartás kérdésével, bár ennek lépéseit nem határozzák meg. A fejlesztési folyamatot támogató, kiegészítő tevékenységek (így például a tudás megszerzése, ontológia kiértékelése, dokumentálás stb.) körét alaposan és részletesen szintén csak a Methontology módszertan írja le.

Összességében elmondható, hogy a módszertanok az ontológiaépítés folyamatára koncentrálnak, és bár felismerik az ontológia-életciklus többi szakaszának fontosságát, hozzájuk kapcsolódó tevékenységek és technikák kidolgozásával már nem foglalkoznak.

### ***II.2.8.3. Összeilleszthetőség, kiegészíthetőség***

Az 1. táblázatban<sup>11</sup> összefoglalt jellemzők gyakorlati szempontokat, olyan jellemzőket vizsgálnak, melyek az ontológia felépítését hatékonyabbá és eredményesebbé tehetik. Az összeilleszthetőség azt mutatja meg, hogy az azonos módszertan szerint, de (földrajzi, szervezeti, szakterületi értelemben) függetlenül dolgozó munkacsoportok munkája milyen mértékben illeszthető össze. Az oktatási ontológia fejlesztéséhez kiválasztott (vagy kidolgozandó) módszertan esetében ez megfontolandó kérdés, hiszen nemcsak egy tananyag által lefedett tudásterület modellezéséről, hanem egy tananyagcsoport, a teljes gazdaságinformatikus képzést átfogó tantárgyak összességének modellezéséről van szó. Minél részletesebben van kidolgozva az adott módszertan, annál könnyebben megoldhatók az összeillesztésből adódó nehézségek is.

A kiegészíthetőség, az ontológia aktualizálási, bővítési lehetőségeinek biztosítása alapvető fontosságú, hogy ontológiánk követni tudja a valós világ változásait, hogy mindig a tényleges tudást reprezentálja. Az oktatási ontológiában modellezni kívánt tudásterületek azonban nem feltétlenül tartoznak a legdinamikusabban változó területek közé. A Lineáris algebra, vagy a matematika más, a képzés során oktatott részterületei megalapozott, kiforrott diszciplínák, melyek keveset változnak. Ezzel szemben például a Szoftvertechnológia keretében oktatott ismeretek köre, akár szemeszterről, szemeszterre is változhat.

Az integrálási képesség a kiegészíthetőség részének tekinthető. Az integrálási képességgel azt jellemezzük, hogy az adott módszertan lehetővé teszi-e más ontológiák beillesztését, kapcsolódási pontok kialakítását.

A bemutatott módszertanok közül a SENSUS esetében mind a kiegészíthetőség, mind az integrálás a módszertan alapelveinek részét képezik,

---

<sup>11</sup> Lásd a 35. oldalon.

még ha ez utóbbi lépései nincsenek is részletesen kidolgozva. Az On-To-Knowledge, illetve Methontology esetében beszélhetünk még kidolgozott megoldásokról. Mindkét megközelítés lehetővé teszi a fejlesztés bármely szakaszában valamely korábbi szakaszokhoz való visszacsatolást – továbbra is betartva a lépések logikai sorrendjét.

	TOVE	USCHOLD & KING	KACTUS	ON-TO-KNOWLEDGE	METHONTOLOGY	SENSUS
Összeilleszthetőség	nem kidolgozott	nem kidolgozott	nem részletezi, részben kidolgozott	nem kidolgozott	probléma-mentes	probléma-mentes
Kiegészíthetőség	statikus	statikus	nincs erre vonatkozó leírás	dinamikus	dinamikus	dinamikus
Integrálási képesség	nincs kiemelt integrálási fázis	bővítés más ontológiákkal	bővítés és építkezés más ontológiákból is	nincs kiemelt integrálási fázis	bővítés és építkezés más ontológiákból is	bővítés más ontológiákkal

1. táblázat: Ontológiaépítési módszertanok összeilleszthetősége, kiegészíthetősége

#### **II.2.8.4. Fogalmi rendszer felépítése, fogalmak közötti kapcsolatfajták, alkalmazott ontológia nyelv**

Az ontológia célja, hogy a modellezni kívánt terület ismereteit a valóságnak a lehető legjobban megfelelő módon ábrázolja. A tudásanyag megszerzésének, a fogalmak osztályozásának módja éppen ezért jelentősen befolyásolja a létrehozandó ontológiai minőségét. A fogalmi rendszer felépítésének, fokozatos bővítésének módja meghatározza az ontológia életciklusának alakulását, egyes szakaszok tartalmi állapotát. A tudásanyag rendszerezése, a fogalmak osztályozása elsődleges szerkezeti áttekintést ad, sőt a későbbi formalizálás alapjait is meghatározza. A formalizálásra kiválasztott ontológia nyelv pedig megszabja az ontológia további alkalmazási lehetőségeit is. Így ezeket a szempontokat érdemes részletesen vizsgálni az egyes módszertanok esetében.

Az irodalom a fogalmi rendszer felépítésének három módját különbözteti meg:

- a legáltalánosabb fogalmaktól a legkonkrétabb felé (top - down),
- a legkonkrétabbtól a legáltalánosabb felé (bottom - up),
- a legfontosabb fogalmaktól a legáltalánosabb és a legkonkrétabb fogalmak felé egyaránt építkező módszer (middle-out).

A módszertanok többsége ez utóbbi módszer alkalmazását javasolja, abból a megközelítésből kiindulva, hogy a legfontosabb fogalmak általában szükségesek a többi fogalom definiálásához, illetve a rendszerben való elhelyezésükhöz. Az így felépített struktúra pedig könnyen érthető és áttekinthető lesz. Egyedül az On-To-Knowledge módszertan biztosít rugalmasságot a módszer kiválasztásában. Lehetővé teszi, hogy az alkalmazás, a modellezni kívánt terület sajátosságainak figyelembevételével válasszuk ki a megfelelő építési módot.

A fogalmak minden módszertan esetében hierarchikus módon kerülnek rendszerezésre. A rendezésnek ez a módja tükrözi leginkább az emberi gondolkodást, segítségével könnyen és gyorsan kialakíthatóak az elsődleges taxonómiák. Ugyanakkor a modellezni kívánt világnak nem feltétlenül lehet minden részét hierarchikusan felépíteni, amelyre azonban nem szolgálnak megoldást a módszertanok.

A fogalmak között kialakított relációk és az alkalmazott ontológia nyelv között szoros összefüggés van. Az ontológia nyelv meghatározza, hogy milyen módon mondhatók ki összefüggések a fogalmakról és a közöttük fennálló kapcsolatokról, hogy milyen logikai következtetéseket végezhetünk és mindezeket, hogyan írjuk le. A 2. táblázat alapján is jól látható, hogy ha a módszertan törekszik az ontológia egy adott nyelven történő formalizálására, akkor már fogalmak rendszerezésénél is pontosan meghatározott relációkat vesznek fel.

	TOVE	Ushold & King	KACTUS	On-To-Knowledge	Methontology	SENSUS
A fogalmi rendszer felépítésének módja	„middle-out”	„middle-out”	„top-down”	mindhárom módszer alkalmazható	„middle-out”	nincs meghatározva
A fogalmak rendszerezésének módja	hierarchikus rendezés és ad hoc relációk	hierarchikus rendezés és meghatározott relációtípusok	hierarchikus rendezés, asszociáción alapuló relációk (modulokba rendezés)	hierarchikus rendezés és meghatározott relációk	hierarchikus rendezés és ad hoc relációk	hierarchikus rendezés és meghatározott relációk
Alkalmazott leíró és specifikációs nyelvek	elsőrendű logika	Ontolingua	CML (Express Ontolingua)	Flogic, OIL, DAML-OIL	export / import: XML, RDF, OIL, DAML+OIL, OWL, CARIN, Flogic, Jess, Prolog	természetes nyelvű ontológia

2. táblázat: Ontológiaépítési módszertanok fogalmi rendszerezése, alkalmazott leíró és specifikációs nyelvek

### II.3. Ontológia nyelvek

Az ontológia alapvető jellemzőit, felhasználásának lehetőségeit, hasznosságát alapvetően befolyásolja a formális leírás nyelve, az alkalmazott logikai apparátus. A tudás reprezentálására kiválasztott nyelv azonban nem csak az ontológia használhatóságát, hanem az ontológiára épülő alkalmazások, rendszerek kifejezőerejét, következtetési képességeit, számítási igényeit is meghatározza. Az ontológiafejlesztés folyamatában éppen ezért meghatározó lépés, hogy mely nyelvet választjuk a formalizáláshoz. A megalapozott döntéshez a különböző nyelvek részletes elemzését és összehasonlítását el kell végezni. A nyelvek önmagukban nem hordoznak információt, csupán a tudás kifejezésének eszközeit biztosítják. A nyelvek összehasonlításának alapját adó kritériumrendszert éppen ezért úgy kell kialakítani, hogy segítségével a tényleges tudás megragadásának, kifejezésének képességét tudjuk vizsgálni.

Az irodalomban számos összehasonlító tanulmányt találhatunk (Corcho és Gómez-Pérez, 2000; Gómez-Pérez és Corcho, 2002; Corcho et al., 2003; Wisdom, 2005), melyek megfelelő kiindulást jelentenek a részletesebb vizsgálódásokhoz. A következőkben bemutatásra kerülő nyelvek esetén is a kifejezőerőre (Mit tudunk állítani?), a következtetési képességekre (Milyen kérdésekre adható válasz?) és követhetőség (Mennyi időn belül várható válasz?) szempontjaira kell koncentrálni. Éppen ezért érdemes kiemelni a CommonKADS keretrendszer ajánlását, melyben olyan értékelési dimenziókat és komponenseket határoznak meg és kapcsolnak össze, melynek segítségével könnyen átlátható, hogy a tényleges tudás elemeinek megfelelő leírása a későbbi következtetés mechanizmusok mely részeit befolyásolják (Schreiber et al, 1998) (3. táblázat).

Tudásábrázolás	Következtetési mechanizmus
Osztályok <i>Metaosztályok</i> <i>Tulajdonságok / Slot-ok</i> <i>Facet</i>	Kivételek Automatikus osztályozás
Taxonómiák	Öröklődés <i>Monoton, nem monoton</i> <i>Egyszerű, többszörös</i>
Eljárások	Eljárások végrehajtása
Relációk / Függvények Példányok / Egyedek / Tények Axiómák	Megszorítások ellenőrzése
Szabályok (production rules)	Szabályokkal történő következtetés <i>Előre haladó következtetés</i> <i>Visszafelé haladó következtetés</i>

3. táblázat: A tudásábrázolás és a következtetési mechanizmusok kapcsolata<sup>12</sup> (Schreiber et al., 1998)

Az elemzésre kerülő nyelveket több szempont szerint is csoportosíthatjuk, legkézenfekvőbb azonban az alkalmazott tudásábrázolási paradigma alapján kategorizálni őket, hiszen ennek segítségével ragadhatók meg legkönnyebben a közös jellemzők.

Bár kevés a tisztán elsőrendű (vagy másodrendű) logikán alapuló nyelv (KIF, CycL), azonban mégis érdemes bemutatni közös jellemzőiket, mivel az ontológianyelvek jelentős része alkalmazza eszközeit más tudásábrázolási formákkal kombinálva. Egy másik ok, amiért érdemes kiemelni az elsőrendű logikai nyelveket és jellemzőiket az az, hogy sok következtető rendszer van, mely saját – nem szabványos – ontológia nyelvet definiál. Ezek között viszont a legtöbb tisztán elsőrendű logikán alapul, pontosabban annak valamely szűkebb részén.

<sup>12</sup> A táblázatban szereplő fontosabb fogalmak meghatározása megtalálható a Fogalomtár (3), (4), (5), (11), (12), (15), (20), (21), (28), (31), (32), (34). pontjaiban.

---

A ma már klasszikusnak nevezhető ontológianyelvek többsége, így az Ontolingua, az OCML, az Flogic-hoz definiált ontológia nyelv a keretalapú tudásábrázolási formát alkalmazza, de ide sorolandó a KM-hez, az EPILOG-hoz és a SNePS-hez definiált ontológia nyelv is.

A tisztán leíró logikán alapuló nyelvek közül a LOOM és a PowerLOOM rendszerekhez kialakított nyelvek kerülnek bemutatásra.

Végül külön fejezetben foglalkozom azokkal a nyelvekkel, melyeket Internet környezetben fejlesztettek ki. Ezek a nyelvek leginkább a keretalapú tudásábrázolást, illetve leíró logikát veszik alapul és további, kifejezetten a Weben történő tudásábrázolást elősegítő szabványokat építenek be rendszerükbe (XOL, SHOE, RFML, DAML+OIL, SWRL, SWRL-FOL, RDFS, OWL). A bemutatandó nyelvek körét lehetne még bővíteni, akár az itt bemutatott nyelvek dialektusaival (az OWL-nek például számtalan dialektusa létezik, mint az OWL1.1, OWL Horst, OWL DLP) vagy akár további önálló formalizmusokkal. Ugyanakkor az elemzés kitér valamennyi releváns, az ontológiafejlesztés területén gyakran alkalmazott és hivatkozott nyelvekre, illetve az újdonságokra is.

### **II.3.1. Elsőrendű logikán alapuló nyelvek**

Az elsőrendű logika (vagy elsőrendű predikátum kalkulus) a propozíciós (nulladrendű) logikára épül, a világot saját azonosítóval rendelkező objektumok összességének tekinti. Predikátumokat fogalmaz meg, melyek az objektumok sajátosságait, az objektumok közötti viszonyokat, relációkat, kapcsolatokat írja le szimbolikus állítások formájában. Egy predikátum minden egyes interpretációjában egyértelmű igaz vagy hamis minősítéssel bír. (Itt azt is meg kell jegyezni, hogy az, hogy pontosan az egyikkel bír, nem feltétlenül jelenti azt, hogy tudjuk is, hogy a kettő közül melyikkel – annyit biztosan tudunk mindenről, hogy vagy igaz, vagy hamis, és nem mindkettő, de esetenként előfordulhat, hogy nem tudjuk, melyik a kettő közül. Körülbelül erre mondják, hogy az elsőrendű logika eldönthetetlen.) Legfőbb eltérése a propozíciós logikához képest, hogy nem csak a kijelentések, hanem azok tartalmának leírására is képes. Az elsőrendű logika ugyanis lehetővé teszi konstansok, változók, függvények alkalmazását is, melyek segítségével az állítások egy adott tartomány minden elemére vonatkoztathatóak. Továbbá új logikai operátorokat, a kvantorokat is alkalmazza, melyek segítségével a „minden” és „létezik” fordulatok is kifejezhetőek.

A predikátum logikában egy formulának végtelen sok interpretációja lehetséges az alaphalmaz, és az azon értelmezett függvények és predikátumok tetszőleges választhatósága miatt. Az elsőrendű logika megengedi az objektumok egymással valamely relációba hozását is és bonyolultabb tételek bizonyíthatók, mint a propozíciós logikában.

Az első- és másodrendű logika közti különbség annyi, hogy míg az elsőrendű logikában a változók csak egy-egy objektumot helyettesíthetnek (így a kvantorok is csak ilyen változókra vonatkozhatnak), a másodrendű logikában már akár reláció- vagy függvényértékű változóink is lehetnek.

---

Elsőrendű logikára épül a KIF<sup>13</sup>, melyet különböző tudásábrázolási rendszerek közötti adatcsere elősegítésére fejlesztették ki. Az Ontolingua és más LISP<sup>14</sup> alapú nyelvek kedvelt adatcsere formátuma.

A vizsgált ontológianyelvek között több is található, mely tisztán logikai alapokra épül, de ezek közül itt most csak a CycL szerepel. A többi logikai alapú nyelv (SWRL, SHOE) egyéb jellemzőik miatt más kategóriában kerülnek bemutatásra.

### **II.3.1.1. CycL**

A CycL nyelvet nagyméretű tudásalapú rendszerekhez kapcsolódó ontológiák specifikálására hozták létre. Ebben a nyelvben elsőrendű logikai formulákkal írhatjuk le a kijelentéseinket, másodrendű fogalmakkal és számossági megszorításokkal (melyek csak a bevitelt könnyítik meg, s az elsőrendű logika nyelvén is megfogalmazhatók lennének) kiegészítve.

A CycL megkülönböztető sajátossága, hogy mikroelméleteket (kontextusokat) lehet létrehozni az ontológiákon belül. A mikroelmélet kijelentések halmaza, mely kapcsolódó körülmények, lényeges tények, HA-AKKOR szabályok és háttér feltételezések meghatározott halmazát reprezentálja. Ezen összekapcsolódó kijelentések csoportját pedig egy objektumnak tekintik.

Szintén kiemelendő tulajdonsága, hogy 5 igazságértéket különböztet meg (alapértelmezett igaz, monoton igaz, alapértelmezett hamis, monoton hamis és ismeretlen / default true, monotonically true, default false, monotonically false, unknown). Megengedi továbbá kivételek megadását, függvények és relációk értékészletének, értelmezési tartományának meghatározását. (Cyc, 2002)

A nyelv nem teljes, következtető rendszerét is bizonyos nyelvi elemekkel támogatja, hogy a következtetések gyorsulhassanak. További hátránya, hogy meglehetősen bonyolult, nehéz a tudásbázisát bővíteni.

Nyílt forrású változatát (OpenCyc) jelenleg is fejlesztik, illetve XML alapú cycML-t készítenek hozzá a tudásbázisok elmentéséhez és betöltéséhez.

Szintaxisa LISP alapú.

### **II.3.2. Keret tudásábrázolási formán és elsőrendű logikán alapuló nyelvek**

A keret tudásábrázolási forma újdonsága abban rejlik, hogy a tudáselemeket nagyobb tömbökben tárolja, sztereotip egységekbe, keretekbe szervezi. Ezek a tömbök magukba foglalhatnak objektumokat, tevékenységeket, vagy eseményeket. Az ábrázolt kapcsolatok inkább funkcionálisak és nem strukturálisak.

Marvin Minsky nevéhez fűződik a keret (frame) fogalmának definiálása. Az emberi gondolkodás azon jellemzőjéből indult ki, hogy egy átélt szituációt a hozzá tartozó viselkedéssel együtt az agy egy keretben tárolja, éppen

---

<sup>13</sup> KIF a Knowledge Interchange Format rövidítése. Lásd a Fogalomtár (40). pontjában.

<sup>14</sup> Programozási nyelv, melyet eredetileg mesterséges intelligencia kutatásokhoz fejlesztettek ki. Lásd a Fogalomtár (41). pontjában.

---

ezért tároljuk egy adott objektumról meglévő összes ismeretünket egy helyen (Minsky,1975).

A keret tudásábrázolási forma bevezeti a prototípus fogalmát, mely azt a felismerést tükrözi, hogy az emberi ismerettárolás sok sztereotip leképezést tartalmaz.

A különböző objektumokat leíró keretek hierarchikus keretrendszerre kapcsolódnak össze a tudásmodellezés során.

A szemantikus háló<sup>15</sup> továbbfejlesztésének tekinthető, hiszen amellet, hogy mind a két tudásábrázolási mód hierarchikus rendszert hoz létre, kiterjeszti a tulajdonságörökítést a procedurális tulajdonságokra is, s a keretalapú tudásábrázolás lehetővé teszi az attribútum-értékek megadását más keretekre való utalással, attribútumok értékészletének, értéktartományának, alap (default) értékének meghatározását, értékeket előállító függvények megadását, értékváltozásra működésbe lépő mechanizmusok alkalmazását is.

Patrick Hayes adott először elsőrendű logikai szemantikát a keret alapú rendszereknek. Bár azt is hozzá kell tenni, hogy vannak olyan részei is a keret alapú rendszereknek (kiszámító függvények, értékváltozásra működésbe lépő mechanizmusok), melyek megfoghatatlanok elsőrendű logikával (Hayes,1979).

### **II.3.2.1. SNePS**

A SNePS a Semantic Network Processing System rövidítése. Ugyan nem keretalapú nyelv, de megközelítését tekintve ehhez áll a legközelebb. Azzal a céllal fejlesztették, hogy képes legyen az egyedek által tett kijelentések értelmezésére, illetve szituációk elemzésére (Shapiro, 2004).

Szintaxisa LISP alapú és következtető rendszer is létezik hozzá (angol nyelvű interface-szel). Zárt világszemléletű<sup>16</sup> és egyedi név feltételezéssel él.<sup>17</sup>

### **II.3.2.2. Ontolingua**

Az Ontolingua<sup>18</sup> elsőrendű logikára alapozott nyelv, mely a Knowledge Interchange Format (KIF) eszközeit modularizálási lehetőségekkel bővíti ki (azaz olyan keretet biztosít, mely tárgyterület-független, újrafelhasználható primitívek jól dokumentált, axiomatikus leírását teszi lehetővé) (Farquhar, 1996). Szerkesztőeszköze nem publikus, de a honlapján működtetett kiszolgáló állomáson keresztül új projekteket lehet indítani, amit szerverük el is tárol. Az Ontolingua előnye, hogy fordítóprogrammal is rendelkezik, amely biztosítja több reprezentációs nyelvvel való kompatibilitását.

A bemutatott nyelvek közül az Ontolingua bír a legnagyobb kifejezőerővel. Több eszköz van hozzá, melyekkel az ontológiákon műveleteket végezhetünk, de – éppen kifejezőerejének nagysága miatt – nincs hozzá következtető rendszer.

---

<sup>15</sup> Lásd a Foglomtár (42). pontjában.

<sup>16</sup> Lásd a Foglomtár (36). pontjában..

<sup>17</sup> Lásd a Foglomtár (37). pontjában.

<sup>18</sup> <http://www.ksl.stanford.edu/software/ontolingua/>



---

### **II.3.2.3. OCML**

Eredetileg a VITAL projekt (Shadbolt et al., 1993) keretében fejlesztették ki kiszámítható / operatív modellezés (operational modelling) biztosítása érdekében, de tekinthetjük az Ontolingua kezelhető részének is. Jelenleg a 6.3-as verziónál tartanak.

Az OCML az objektum-orientált és a reláció-orientált szemléletet ötvözi és három fajta konstruktor specifikálását teszi lehetővé, melyek a következők: funkcionális kifejezés, kontroll kifejezés és logikai kifejezés. Míg a funkcionális kifejezés mindig az éppen vizsgált terület egy objektumát írja le, addig a kontroll kifejezések cselekvéseket és azok végrehajtási sorrendjét határozzák meg. A definíciókban pedig megengedett számítási eljárások alkalmazása is (Domingue et al., 1999).

Nagy kifejező erővel rendelkezik (hasonlóan az Ontolingua-hoz) és ugyan létezik következtetési rendszere, az nem teljes<sup>19</sup> (sőt: nem képes valamennyi igaz és elvileg levezethető állítást sem levezetni).

Zárt világszemléletet alkalmaz és egyedi név feltételezéssel él. Szintaxisa LISP-szerű és nem létezik XML alakú leírása.

### **II.3.2.4. Flogic**

Az Flogic nyelvet<sup>20</sup>, melynek neve a Frame Logic kifejezésből származik, objektumorientált adatbázisok, frame alapú rendszerek és logikai programok specifikálására fejlesztették ki (Kifer et al., 1995). Az Flogic újdonsága abban rejlik, hogy a fogalmi modellezés konstruktorait (osztályok, tulajdonságok, értelmezési tartományra és értékészletre vonatkozó megszorítások, öröklődés, axiómák) egységes logikai keretbe foglalja.

Az FLogic következtető motorja zárt világszemléletet alkalmaz, így csak a leírt dolgokat tekinti létezőnek. Továbbá egyedi név feltételezéssel is él.

Az Flogic segítségével objektumokról tehetünk logikai és relációs kijelentéseket (ahol az osztályok, metódusok, paraméterek is objektumok). Ugyanakkor a nyelv nem tesz szintaktikai különbséget az objektumok és osztályok között, és így az osztályok is lehetnek más osztályok objektumai. A metódusokhoz/relációkhoz kiszámító programokat, eljárásokat nem lehet rendelni (ellentétben például az OCML-lel).

Elsőrendű logikai kifejezőereje van. Az ehhez kapcsolódó bizonyító rendszere teljes és helyes<sup>21</sup> (de csak félig eldönthető<sup>22</sup>).

### **II.3.2.5. Knowledge Machine**

A Knowledge Machine egy ontológianyelvből, következtető gépből és lekérdőző nyelvből álló rendszer. A Knowledge Machine nyelve olyan keret (frame) alapú nyelv, melyben elsősorban elsőrendű logikai formában írhatjuk le az állításokat (Clark és Porter, 2004).

---

<sup>19</sup> Lásd a Fogalomtár V.1.2 Bizonyítások tulajdonságai alfejezet (2). pontjában.

<sup>20</sup> [http://www.ontoprise.de/content/e5/e190/e191/tutorial\\_flogic\\_ger.pdf](http://www.ontoprise.de/content/e5/e190/e191/tutorial_flogic_ger.pdf)

<sup>21</sup> Lásd a Fogalomtár V.1.2 Bizonyítások tulajdonságai alfejezet (1). pontjában.

<sup>22</sup> Lásd a Fogalomtár V.1.2 Bizonyítások tulajdonságai alfejezet (3). pontjában.

---

Széles eszköztárral rendelkezik, így többek között lehetővé teszi kvantorok alkalmazását a tulajdonságok definiálásánál, illetve típusokra és értékekre is tehetünk megszorításokat. Továbbá cselekvések, szituációk, események leírására is képes. Mindezeknek köszönhetően kifejező ereje nagy, ugyanakkor emiatt nem lehet teljes egy hozzá írt következtető rendszer (és így a Knowledge Machine-ben jelen lévő következtető rendszer sem az).

A Knowledge Machine sajátossága, hogy angolul értelmesnek hangzó mondatok formájában tehetünk vele kijelentéseket. Emellett bizonyos mértékben támogatja természetes nyelvű szöveg készítését, illetve felismerését, mivel lehetővé teszi, hogy adott objektumhoz, relációhoz szöveget is rendeljünk, melybe a hozzájuk kapcsolódó objektumok, relációk szövegei behelyettesíthetők.

Szintaxisa LISP alapú.

### **II.3.2.6. EPILOG**

Az EPILOG az Episodic Logic rövidítése. Ez a keret alapú nyelv inkább tudásbázis-nyelvnek tekinthető, mivel közvetlenül még a taxonómiákat sem támogatja. Tudásábrázolásra alapvetően elsőrendű intenzionális logikai formulákat alkalmaz, de a szokásos elsőrendű készleten kívül vannak eszközei az időbeliség kifejezésére is. Így többek között meghatározható, hogy valami mikor igaz vagy ok-okozati összefüggések is leírhatók segítségével (Schaeffer et al., 1993).

További sajátossága, hogy egyedek által tett kijelentések megadását is támogatja. Szintén lehetővé teszi, hogy a szabályokhoz egyben valószínűségeket is megadjunk. Következtető rendszere azonban nem teljes és úgy tűnik nem is kívánják továbbfejleszteni.

Nagy hangsúlyt fektettek a fejlesztés során arra, hogy olyan nyelvet hozzanak létre, mely az angol nyelvet logikai formába tudja fordítani, illetve vissza. Más természetes nyelvre azonban nem nagyon lehet alkalmazni.

LISP alapú nyelvek közé tartozik.

### **II.3.3. Leíró logikai nyelvek**

A leíró logika az ábrázolandó tudásterületet magába foglaló tudásbázist két nagyobb egységre bontja: terminológia dobozra (T-doboz) és adatdobozra (A-doboz). A T-doboz (TBox) fogalmakra és szerepekre vonatkozó állítások, terminológiai axiómák halmaza. Az A-doboz (ABox) pedig adat-axiómák halmaza, tehát az egyedekre vonatkozó tudásunkat tartalmazza. Az A-doboz elkészítésének megkezdése előtt a T-dobozt (legalábbis nagy részben) meg kell adni.

#### **II.3.3.1. Loom**

A LOOM nyelve a leíró logikai nyelvek közé tartozik, elsősorban szakértő rendszerek és más intelligens alkalmazások fejlesztésének támogatására dolgozták ki (ISX Corporation, 1991).

---

A LOOM sajátossága, hogy a relációk definíciójában is lehetővé teszi változók alkalmazását, ugyanakkor diszjunkció (vagy) általában nem szerepelhet bennük.

A nyelv osztályozó és következtető rendszerrel is rendelkezik. Összehasonlítva más leíró nyelvekkel kifejező ereje igen jónak tekinthető.

További jellemzői közül ki kell emelni, hogy lehetővé teszi a nyílt<sup>23</sup> és zárt világszemlélet felváltva történő alkalmazását, rendelkezik eszközökkel az idő, illetve szituációk kezeléséhez.

Szintaxisa LISP jellegű, de nincs hozzá Java vagy C++ interfész. Rendelkezik azonban webes szerkesztő eszközzel (Ontosaurus).

### **II.3.3.2. PowerLoom**

A LOOM tovább fejlesztett változatáról van szó, mely szintén tartalmaz osztályozó és következtető rendszert. A LOOM nyújtotta lehetőségeken túllépve, azonban a hibakeresést is támogatja, mivel képes felfedni következtetési útját. A PowerLoom-ból azonban hiányzik néhány, a Loom nyelvében még szereplő konstrukció, mint például a fentebb említett változó-alkalmazás lehetősége a relációk definíciójában (Chalupsky et al., 2003).

A PowerLoom következtető rendszere alapvetően nyílt világszemléletet alkalmaz, de tetszőleges részei átállíthatók zárt világszemléletre, mely gyakorlati szempontból igen hasznos tulajdonság.

A tudást hierarchikus modulokba tárolja és jelenleg csak Horn-formulákat<sup>24</sup> képes kezelni. A jövőbeni cél azonban az, hogy az Ontolingua nyelven írt kifejezéseket is alkalmazni tudja.

Szintaxisa szintén LISP alapú, de már képes kódot generálni különböző program nyelvekhez (Java, C++).

### **II.3.4. Web alapú nyelvek**

A Szemantikus Webbel szembeni elvárások teljesítésének érdekében egyre több Web alapú tudásábrázolási formát fejlesztettek ki.

A legelfogadottabb megközelítés Tim Berners-Lee nevéhez fűződik, aki szerint a szemantikus Web a hagyományos Web olyan kiterjesztése, melynek célja a webtartalom strukturálása, a fellelhető információk jól definiált jelentéssel való megtöltése a felhasználók és számítógépek közötti hatékony együttműködés támogatása érdekében (Berners-Lee, et al. 2001).

A hagyományos tudásábrázolási rendszerek központosított módon működtek, azaz minden felhasználótól megkövetelték, hogy a közös fogalmakhoz pontosan ugyanazt a definíciót kapcsolják. Ezzel szemben a szemantikus Web kutatói a sokoldalúság érdekében hajlandóak elfogadni paradoxonok és megválaszolatlan kérdések előfordulásának lehetőségét is. Éppen ezért a Weben történő tudásábrázolási formákat úgy kell kialakí-

---

<sup>23</sup> Lásd a Fogalomtár (35). pontjában.

<sup>24</sup> Egy Horn-formula olyan predikátumok összekapcsolása a 'vagy' művelettel, melyek közül legfeljebb egy nincs negálva, a többi negált. Lásd a Fogalomtár (38). pontjában.

---

tani, hogy a logika az objektumok lehető legszélesebb körének kifejezésére legyen alkalmas, ugyanakkor paradoxonok, ellentmondások előfordulása se akassza meg a működést.

A Szemantikus Web fejlesztését támogató technikák közül kettőt kell kiemelni, melyeket már ma is széles körben alkalmaznak. Az egyik az XML<sup>25</sup> (eXtensible Markup Language), a másik pedig az RDF<sup>26</sup> (Resource Description Framework).

Az XML-t adatok leírására fejlesztették ki, egy általános, szabványos adatcsere formátum, melynek segítségével tetszőleges struktúrát adhatunk a dokumentumoknak, ugyanakkor arról nem tudunk meg semmit, mit jelent maga a struktúra.

Ezt a hiányosságot kívánja betölteni az RDF (a Resource Description Framework / Erőforrás Leíró Keretrendszer). Az RDF egy W3C ajánlás, olyan általános adatleíró nyelv, melynek célja a Weben történő információábrázolás támogatása. Az RDF-ben az úgynevezett triplet-ek halmaza szolgál a jelentés kifejezésére. A triplet egy állítás, melynek három eleme van: az alany, az állítmány és a tárgy. A triplet-eket nevezik „RDF Statement”-nek is. Az RDF-ben olyan állítások fogalmazhatók meg, melyek leírják, hogy az adott objektum milyen értékekkel bíró tulajdonságokkal rendelkezik. Ily módon olyan struktúra jön létre, mely a gépek által feldolgozott adatok nagy részének természetes módon történő leírását adja. Mind a tárgy, mind az alany meghatározására URI-t (Universal Resource Identifier) alkalmaznak.

A következőkben a Web alapú tudásábrázolásra kidolgozott ontológia nyelvek közül bemutatom a legfontosabb XML, illetve RDF alapú nyelveket.

#### **II.3.4.1. Szabványok, ajánlások**

##### **(1) RDF Schema**

Az RDFS az RDF Schema (RDF Séma) rövidítése. Ez a nyelv az RDF szemantikai bővítése és az RDF szókészlet-leíró nyelv specifikációját foglalja magába. RDFS segítségével specifikálhatók az egymással kapcsolatban álló erőforrások csoportjai és ezen erőforrások közötti viszonyokat leíró mechanizmusok (Brickley és Guha, 2004).

Az RDFS is támogatja az alap ontológiai struktúrák alkalmazását (azaz az osztályok, egyedek, részosztályok, példányok használatát).

Az RDFS abban különbözik a legtöbb objektum orientált programozási nyelvtől, hogy nem az osztály egyedeit jellemző tulajdonságokkal definiál egy osztályt, hanem azokkal az erőforrás-osztályokkal, amelyekre ezek a tulajdonságok vonatkoznak. Mindezt, pedig a relációkra vonatkozó érvényességi kör és értéktartomány mechanizmusok alkalmazása teszi lehetővé (illetve a relációk között is lehetőség van “része” kapcsolat értelmezésére).

---

<sup>25</sup> <http://www.w3.org/XML/>

<sup>26</sup> <http://w3c.hu/forditasok/RDF/REC-rdf-primer-20040210.html>

---

További sajátossága a nyelvnek, hogy lehetőség van konténerek, kollekciónak és „statement”-ek megadására is. Mind a konténerek, mind a kollekciónak tartalmazhatnak egyedeket, osztályokat, sőt önmagukat is. Csupán annyi közöttük a különbség, hogy a konténerek esetében nem jelenthetjük ki, hogy a felsorolt elemeken kívül nincs más elemük. Ez pontosan azt jelenti, hogy a konténerekre nyílt világszemlélet szerint kell tekinteni. A kollekciónak esetében pedig kimondhatjuk, hogy a felsorolt elemeken kívül is lehetnek elemeik. Ez egyben azt jelenti, hogy a kollekciónak lehet kezelni nyílt és zárt világszemlélet szerint is.

A „statement”-ek (más néven RDF Triplet-ek) segítségével két tetszőleges elem (az „alany” és a „tárgy”) közötti reláció fennállását mondhatjuk ki. Ez egyben azt is jelenti, hogy nem csak egyedek, hanem osztályok között is megadhatók relációk, sőt relációk közt is. (Például a „Szomszédja domain Ember” annyit tesz, hogy a szomszédja reláció alaphalmaz az Ember halmaz.) A nyelv nem követeli meg az „egyed” és az „osztály” fogalmának éles megkülönböztetését.

## (2) *OWL*

Az OWL nyelvet elsősorban a Web tartalom gépi értelmezésének magasabb fokú támogatására fejlesztették ki. Túl kíván lépni az XML, az RDF és az RDF Séma nyújtotta lehetőségeken bővített szókészlet és formális szemantika alkalmazásával (McGuinness és van Harmelen, 2004; Smith et al., 2004).

Az OWL egyik legnagyobb előnye, hogy már számtalan projektben alkalmazták, így jelentős tapasztalatok gyűltek össze az ontológiaépítés kapcsán. Alkalmazhatóságát tovább növeli, hogy jól használható szerkesztő eszközök (Pl.: Protégé 3.2 OWL) is léteznek hozzá.

A nyelv általános tulajdonságai közül ki kell emelni a nyílt világszemlélet alkalmazását, azaz ha valami nem szerepel az ontológiában arról nem teszi fel, hogy nem is létezik. Szintén lényeges jellemzője, hogy nincs egyedi név feltételezés, tehát nem feltételezi, hogy a különböző kifejezések valóban különböző fogalmakat takarnak.

Szerkesztése ugyan nehézkes, de az eredmény mindenképpen jól használható (többek között a konzisztencia ellenőrzésnek és a definíciók alapján történő automatikus osztályozásnak köszönhetően, melyeket a nyelvhez (pontosabban a Lite-hoz és a DL-hez) kifejlesztett programok végeznek.).

A nyelv legnagyobb hátránya, hogy nem engedi meg változók alkalmazását, így elsőrendű logikai formulák sem alkalmazhatóak. Ennek következtében a nyelv kifejezőereje korlátozott.

Az OWL három alnyelvből áll, melyek a következők: OWL Lite, OWL DL és OWL Full.

### **OWL Lite**

Az OWL Lite elsősorban osztályozási hierarchiák és egyszerű korlátozások alkalmazására alkalmas. A tulajdonságoknak (relációknak) már itt is

---

számos jellemzője megadható, így meghatározható, hogy tranzitív-e, szimmetrikus-e, funkcionális-e az adott tulajdonság, hogy melyik reláció az inverze, és hogy az egyik tulajdonság része-e a másiknak (McGuinness és van Harmelen, 2004; Smith et al., 2004).

A lehetséges kifejezőerőre vonatkozóan erős megkötéseket tartalmaz. Támogatja ugyan a relációkra vonatkozó számossági korlátozás<sup>27</sup> alkalmazását, de a kardinalitás értéke csak 0 vagy 1 lehet. Mindezek következtében komplexebb dolgok leírására nem alkalmas, vagy az nagyon körülményes lesz.

### **OWL DL**

Az OWL DL kifejező ereje már nagyobb, mint az előbbi Lite alnyelvé, legfőbb jellemzője, hogy eldönthető (azaz elvileg minden konklúzió kiszámítható, és minden számítás véges időn belül be is fejeződik, bár nem feltétlenül hatékonyan). A kifejezhető definíciók a SHION<sup>28</sup> leíró logikának megfelelő osztályba esnek (McGuinness és van Harmelen, 2004; Smith et al., 2004).

Az OWL DL tartalmazza az összes OWL nyelvi konstrukciót, azonban ezek csak bizonyos korlátozásokkal használhatók. Például egy osztály nem lehet egyede egy másik osztálynak, nem lehet a tranzitív tulajdonságokra számossági korlátozást alkalmazni. Ezen korlátozások épp azt a célt szolgálják, hogy az OWL DL eldönthető maradjon: az OWL DL így lesz az OWL Full-nak egy maximális eldönthető része.

### **OWL Full**

Az OWL Full nyelvben a maximális kifejező erő biztosítása a cél. Ennek megfelelően valamennyi OWL és RDF(S) konstrukció használata megengedett, minden korlátozás nélkül. Azonban ebben a változatban már elvileg sem lehetséges minden kijelentés logikai feldolgozása, nem tudunk minden esetben nyilatkozni helyességükről (McGuinness és van Harmelen, 2004; Smith et al., 2004).

#### **II.3.4.2. XML alapú nyelvek**

##### **(1) XOL**

Az XOL az XML-based Ontology Exchange Language rövidítése (Karp et al., 1999).

Az XOL fejlesztésének célja olyan közvetítő nyelv létrehozása, melynek segítségével az ontológiák átadhatók különböző adatbázis rendszereknek, ontológiafejlesztő eszközöknek vagy alkalmazási programoknak.

---

<sup>27</sup> Lásd a Fogalomtár (23). pontjában.

<sup>28</sup> A „SHION leíró logika” egy betűszó a kifejezőerőre. A betűk a következőt jelentik: s: a tranzitivitás kifejezhető; h: relációkra is kimondható, hogy része egyik a másiknak; l: relációkra kimondható, hogy egymás inverzei; o: osztály megadható elemeinek felsorolásával; n: tehetőek számossági megszorítások. A SHION szó arra utal, hogy ezek mind elemei a nyelvnek.

---

A nyelv szintaktikája az XML nyelven alapszik, míg szemantikájának alapját az OKBC Lite adja (mely az OKBC<sup>29</sup> egy egyszerűsített változata). Jelentős hatással voltak a nyelv fejlesztésére a hasonló céllal létrehozott Ontolingua és OML nyelvek is (míg az előbbtől szintaktikájában, addig az utóbbitól szemantikájában tér el).

Támogatja az alap ontológiai struktúrákat (osztály, individuum, részosztály, példány), ráadásul az ontológianyelvek többségétől eltérően XOL esetén egy osztály is lehet egy másik osztály példánya.

Az XOL-ben a relációkat slot-oknak nevezik és szigorúan csak kétváltozósak lehetnek (mint általában a frame alapú rendszerekben). Ugyanakkor megadható a slot értelmezési tartománya, értékészlete és a számossága is. A XOL további előnye a hasonló nyelvekhez képest, hogy kijelenthető, hogy egy slot inverze egy másiknak, illetve többértékű slot esetén meghatározható a slot típusa (halmaz, multihalmaz vagy lista).

## (2) *SHOE*

A SHOE nyelvet, a legkülönbözőbb Internet ágensek igényeinek figyelembevételével, elsősorban Web dokumentumok annotálásának támogatására fejlesztették ki. A SHOE-t a következő jellemzők különböztetik meg a többi nyelvtől és teszik alkalmassá a fentebb megfogalmazott célok teljesítésére:

Ugyan nem kapcsolódik a legismertebb szabványokhoz (így például sem az RDF-hez, sem az OKBC-hez), s eredetileg nem tudásreprezentálás támogatása volt a célja, szintaxisa a HTML-re épül, annak teljesen kompatibilis kiegészítése. Ráadásul ma már létezik egy ezzel majdnem egyenértékű XML szintaxisa is.

Nagy hangsúlyt fektet a lehetséges ellentmondások megelőzésére. Ennek megfelelően csak kijelentések (assertion) tehetők a nyelv segítségével, visszavonás pedig nem lehetséges. Továbbá nem teszi lehetővé a negálást, és a kijelentés megfogalmazóját (tehát azt a Web oldalt vagy más erőforrást, ahonnan az információ származik) mindig hozzákapcsolja az adott kijelentéshez.

Fontos jellemzője a nyelvnek, hogy tetszőleges változós számú relációkat definiálhatunk benne, a következtetési szabályokat, pedig Horn-klóz formában adhatjuk meg – vagyis kifejezőereje határozottan kisebb, mint az elsőrendű logikáé (Luke és Herlin, 2000).

Végül érdemes kiemelni, hogy minden ontológia elemnek csak pontosan egy definíciója létezhet, és az ontológia nem tartalmazhat olyan elemeket, melyeknek ugyanaz a nevük.

## (3) *RFML*

Az RFML a Relational Functional Markup Language rövidítése.

---

<sup>29</sup> Egy protokoll, ami a tudásreprezentációs rendszerek (Knowledge Representation Systems, KRS-ek) közti kommunikáció módját definiálja. Bővebben lásd a Fogalomtárban. Lásd a Fogalomtár (39). pontjában.

---

A nyelv alapjaiban a SHOE nyelvhez és a később RFML alapokon készült RuleML<sup>30</sup>-hez áll a legközelebb. A RuleML szintaxisánál, annyiban több, hogy függvény jellegű részei is vannak. A SHOE nyelven pedig azzal lép túl, hogy a Horn formulák kiterjesztett változatát alkalmazza (Boley, 2000).

Képes a kettőnél több változós relációkat is kezelni, de a függvényekről csak korlátozottan tudunk kijelentéseket tenni, lényegében csak definiálásukat támogatja. Vitathatatlan előnye azonban, hogy mivel megengedi függvényértékű függvények alkalmazását is, ezeket egymásba ágyazva egészen bonyolult rendszereket lehet létrehozni. A nyelvet azonban 2001 óta nem fejlesztették és dokumentációja is minimális, így nem sorolható a jól alkalmazható nyelvek közé.

#### (4) DAML+OIL

A DAML+OIL<sup>31</sup> a W3C ajánlasként kidolgozott RDF-re és RDF Sémára épít, kibővítve az általuk alkalmazott modellezési primitiveket. A DAML+OIL a frame alapú nyelvek modellezési primitiveit alkalmazza. Ezt a nyelvet az eredeti DAML (Darpa Agent Markup Language) ontológia nyelv alapján hozták létre, az OIL (Ontology Interface Layer) számos nyelvi elemének beépítésével (van Harmelen et al., 2001).

Támogatja a teljes RDFS szintaxist, és ebben a nyelvben is csak kétváltozós relációk alkalmazása lehetséges. Az alkalmazható megszorítások, az osztályokkal, relációkkal kapcsolatos műveletek köre azonban szélesebb.

Az ontológia két osztályáról kijelenthetjük, hogy egybeesnek, illetve két vagy több osztályról, hogy diszjunktak. A nyelv támogatja az osztályok particionálását, osztályok Boole-kombinációját és osztály megadását elemeinek felsorolásával is. Lehetővé teszi osztályokra vonatkozó lokális megszorítások (allValuesFrom, someValuesFrom, hasValue<sup>32</sup>) megadását is. A tulajdonságokra vonatkozóan is lehet megszorításokat megfogalmazni, így lehetséges globális és lokális számossági korlátozás<sup>33</sup> megadása is. Relációk esetében pedig kijelenthető, hogy tranzitívak, függvények, injektívek és hogy melyik reláció az inverzük.

Egyedek esetében nem él egyedi név feltételezéssel: két egyedről kijelenthető, hogy azonosak vagy különbözőek.

Összehasonlítva a többi XML alapú nyelvvel, megállapítható, hogy csak azok rendelkeznek legalábbakkora kifejezőerővel, mint a DAML+OIL, melyek rá épülnek (így például az OWL és az SWRL).

---

<sup>30</sup> <http://www.ruleml.org/>

<sup>31</sup> A DAML+OIL alapjait adó DAML és OIL nyelvek egyike sem volt önmagában elterjedt. A két nyelv külön bemutatása azért nem szükséges, mert amiket elmondhatunk róluk, azok a DAML+OIL kapcsán is bemutatásra kerülnek.

<sup>32</sup> Lásd a Fogalomtár (24). pontjában.

<sup>33</sup> Lásd a Fogalomtár (22). pontjában.



---

## (5) SWRL

Az SWRL a Semantic Web Rules Language rövidítése. Néhány W3C tag által kidolgozott nyelvről van szó, amelyet ugyan a W3C oldalán már publikáltak, de még nem fogadták el szabványként.

A nyelv az OWL és a RuleML alnyelveire épül, azokat fejleszti tovább azáltal a céllal, hogy a különböző relációk egymással való kapcsolatát is ki lehessen fejezni, le lehessen írni. Ugyan az OWL-ben sok mindent el tudunk mondani az osztályok közötti kapcsolatokról, a relációk kapcsolatáról szinte semmit nem állíthatunk, s emiatt viszonylag kicsi is a nyelv kifejezőereje.

Az SWRL fájl azon túl, hogy tartalmazhat minden OWL DL-beli elemet, elsőrendű változókat tartalmazó, Horn-formula alakú szabályokat is magába foglal. A formulák atomjai, melyekből a szabályok felépülnek, pedig a következők lehetnek:

- „x eleme C” (ahol x egy változó vagy egy konstans egyed, C egy OWL osztály);
- „x P relációban van y-nal” (ahol x,y változó vagy konstans, P pedig egy OWL tulajdonság);
- „x ugyanaz, mint y” (x,y két változó vagy konstans);
- „x nem ugyanaz, mint y” (x,y két változó vagy konstans);
- összehasonlító operátorral képzett atom: =, <=, >=, ... (numerikus értékekre értelmes)
- aritmetikai operátorral képzett atom, pl. Add(x,y,z) akkor igaz, ha  $x+y=z$ ;
- dátum/idő művelettel képzett atom;
- listaművelettel képzett atom (Horrocks et al., 2004).

Az SWRL tehát olyan szabályokkal egészíti ki az OWL DL nyelvet, mely lehetővé teszi a relációk közötti kapcsolatok, így akár két reláció kompozíciójának kifejezését is. Ebből a tulajdonságból azonban az is következik, hogy a nyelv eldönthetlenné válik.

Létezik SWRL plugin a Protégé szerkesztőhöz, mellyel szabályokat is lehet szerkeszteni.

## (6) SWRL-FOL

Az SWRL FOL elnevezésben a FOL a First Order Logic rövidítése. Az SWRL egyik fejlesztője bővítette tovább a nyelvben szereplő szabályok körét. A nyelvnek ebben a változatában a szabályok axiómák (assertion) formájában definiálhatók, mely azért hasznos, mert eltűnik az eddig kötelező implikációs forma. A nyelv így alkalmas a teljes elsőrendű logikai készlet („és”, „vagy”, „nem” konnektívák, elsőrendű kvantorok) kezelésére (Patel-Schneider, 2004).

A bemutatott XML alapú nyelvek közül így ez bír a legnagyobb kifejezőerővel, bár kérdéses, hogy jól hasznosítható-e ez a többlettudás.

---

### II.3.5. Ontológia nyelvek összehasonlítása – Összefoglaló

A vizsgált nyelvek közül nem lehet egy vagy két nyelvet megnevezni, mely egyértelműen kiemelkedik a többi közül hasznosságát, alkalmazhatóságát tekintve. Minden esetben meghatározó, hogy milyen céllal hozzuk létre az ontológiánkat, mire szeretnénk azt alkalmazni és csak ezek függvényében választható ki a formalizálás legmegfelelőbb nyelve. Az elemzések alapján jól látható, hogy minél kifejezőbb egy ontológia nyelv, minél pontosabban tudja ábrázolni a tényleges tudást, annál kevésbé lesz alkalmas következtetések elvégzésre.

Az alapvető ontológia struktúra elemeit (osztály, részosztály, egyed) valamennyi vizsgált nyelv támogatja, bár az elemek taxonómiákban történő ábrázolásra alkalmazott primitívek már eltérőek. Azok a nyelvek lesznek kifejezőbbek (így például Ontolingua, Loom, DAML+OIL, OWL, CycL stb.) melyek támogatják új fogalmak diszjunkt fogalmakkal való meghatározását, amelyekben kimondhatjuk részosztályok diszjunktságát is.

Szintén befolyásolja a kifejező erőt, hogy mennyire könnyű relációk, függvények meghatározása, hogy alkalmazhatunk-e axiómákat, melyek az ontológiákban történő tudásreprezentálás erőteljes eszközei.

Kifejezőerő tekintetében mindenképpen az Ontolingua áll az első helyen, de nagy a kifejezőereje a ráépülő OCML-nek is. A leíró logikai nyelvek közül a Loom-ot, az XML alapú nyelvek közül pedig az DAML+OIL-t, OWL-t és az SWRL-t kell kiemelni kifejező erejük tekintetében.

A következtető rendszer tekintetében már rosszabb a helyzet. A nyelvek egyik feléhez egyáltalán nem is fejlesztettek ilyen rendszert, a többi nyelv esetében ugyan létezik következtető rendszer, de egy kivételével (OWL DL), ezek egyike sem teljes. A teljesség hiánya a kifejezőerő következménye: ha elég nagy a kifejezőerő, akkor nem lehet teljes a következtető rendszer. Az OWL DL-hez ugyan létezik következtető rendszer (például a FaCT vagy a Racer), de ezen rendszerek (mivel az OWL csak annyira alkalmas, hogy osztályok közti kapcsolatokat definiáljon) annyira képesek csak, hogy sorba rendezzék az osztályokat aszerint, hogy melyik tartalmazza melyiket. Ezt „consumption”-nak nevezik, ha csak két osztályról van szó, és „classification”-nek, ha az összes osztályról. Továbbá kifejezhető a nyelvvel, hogy van-e olyan osztály, amelyik mindenképp üres kell, legyen (például ami két, diszjunkt deklarált osztály metszeteként van megadva).

### II.4. Kapcsolódó következtető rendszerek

Kutatásom egyik célja olyan ontológiai modell kialakítása, mely támogatja a hallgatók által elsajátított ismeretek adaptív módon történő felmérését, azaz megteremti számítógéppel támogatott adaptív vizsgáztatási rendszer alapjait.

Jelen fejezetben egyik célom a vizsgarendszerben alkalmazható következtető gép kiválasztását elősegítő szempontrendszer bemutatása, a másik célkitűzés pedig, hogy javaslatot tegyek a vizsgálatra érdemes következtető rendszerekről.

---

Az előző fejezetben bemutatásra került ontológia nyelvek a kialakítani kívánt ontológia formalizmusát adják meg. A formalizmus magába foglalja a szintaxist, mely meghatározza, hogy leírásunk milyen alapelemekből épülhet fel és ezeket hogyan kapcsolhatjuk össze; valamint a szemantikát, mely azt határozza meg, hogy az egyes szintaktikai elemeket és konnektívákat hogyan értelmezzük.

Következtető rendszernek nevezzük azokat a programokat, melyek képesek beolvasni valamely ontológia nyelven leírt ontológiát, majd reagálni a felhasználó által feltett kérdésekre. Alapesetben egy kérdés szintén az adott ontológia nyelven megfogalmazott állítás, melynek igazságtartalmáról érdeklődünk. A „reakció”, melyet a gép adhat, ebben az esetben az „igaz”, „hamis”, és „nem tudom” válaszok valamelyike lehet; ez a válasz kiegészülhet azt alátámasztó indoklással.

A következtető rendszerek vizsgálata azért is indokolt, mert használhatóságukat meghatározza az alkalmazott ontológia nyelv kifejezőereje. Minél nagyobb egy adott nyelv kifejezőereje, azaz minél összetettebb összefüggéseket képes leírni, annál kisebb lesz azon kérdések száma, melyre egyáltalán lehetséges választ adni. A cél tehát megtalálni azt az egyensúlyi állapotot, melyben már megfelelően bonyolult összefüggéseket tudunk feltenni és a kérdések megválaszolása sem lehetetlen.

Az alkalmazott következtető rendszer hasznosságát azonban nemcsak az ontológia nyelv, hanem a rendelkezésre álló következtetési mechanizmusok és kiválasztott algoritmusok köre, illetve az optimalizálás implementációja is nagymértékben befolyásolja.

A kialakítandó vizsgálati szempontrendszernek figyelembe kell vennie ezt a kettősséget, mely az alkalmazható elemzési módszerek körét is befolyásolja.

Az ontológiafejlesztési módszertanok elemzése is rámutatott arra, hogy az ontológiákkal foglalkozó kutatói kör napjainkig az ontológiaépítés folyamataira koncentrált, kevés figyelmet szentelve az ontológia-életciklus más szakaszainak, így többek között ontológiák alkalmazásának kérdésére is. A kapcsolódó irodalom legfeljebb azt vizsgálja, hogy az ontológiaszerkesztő eszközök rendelkeznek-e valamilyen következtető géppel, azok minőségével, használhatóságukkal már nem foglalkoznak. Kivételt jelent a már korábban is említett MEO projekt, melynek keretében több mint ötven következtető rendszer részletes elemzését és összehasonlítását végezték el.

Tanulmányukban javasolt szempontrendszer nemcsak az adaptív vizsgarendszer esetén, hanem bármely ontológia alapú alkalmazáshoz kapcsolódó következtető rendszer kiválasztásában nyújt megalapozott és hatékony megoldást (MEO, 2005c).

Javasolt módszer szerint két lépésben kell elvégezni a vizsgálatot. Az első lépésben a rendszerhez kapcsolódó ontológia nyelv kifejezőerejére vonatkozó kérdéseket kell megválaszolni. Majd második lépésben, a kifejezőerőre vonatkozó elemzések segítségével kiválasztott következtető rendszerek gyakorlati elemzését, tesztelését végezzük el, melynek segítségével megállapítható, hogy a kiválasztott algoritmusok az implementált optimá-

---

zálásokat kihasználva mennyire hatékonyan képesek együttműködni. Az így kialakított megközelítés előnye abban rejlik, hogy az ontológia nyelv kifejező erejére vonatkozó kérdések szigorúan elméleti megfontolások alapján megválaszolhatók, az így kapott válaszok objektívak és megbízhatóak lesznek és a gyakorlati tesztelésre is alkalmas következtető rendszerek köre gyorsan leszűkíthető, felgyorsítva a kiválasztás folyamatát.

#### **II.4.1. A kifejezőerő vizsgálata**

A kifejezőerőre vonatkozó elemzés szempontrendszerének felállításakor abból megállapításból indultak ki, hogy az összes vizsgálandó rendszer nyelve beágyazható az (egyenlőséges) elsőrendű logikába (esetleg egy minimális aritmetikai támogatással). Ennek köszönhetően a következtető motorok nyelvének kifejezőereje megragadható azzal, hogy mekkora részét támogatják az elsőrendű logikának. A tapasztalatok alapján a következő csoportokat különítették el:

- azon rendszerek, melyek csak Horn-szabályokat támogatnak;
- azon rendszerek, melyeknél az implikációk törzsében engedélyezett a negálás;
- azon rendszerek, melyek tetszőleges elsőrendű formulát képesek feldolgozni.

Ez a három csoport kifejezőerő szempontjából ebben a sorrendben válik egyre erősebbé.

Továbbá, ettől függetlenül egy rendszer támogathatja a következő tulajdonságokat (melyeket szintén megvizsgálták a MEO projekt keretében folytatott elemzések során): egyenlőség, függvényszimbólumok, egzisztenciális kvantálás, aritmetika, fuzzy értékek, sorted univerzum és valószínűségek.

Vizsgálatuk következő szempontja a lekérdezésre, a lekérdező nyelv nyújtotta lehetőségekre irányult, a következő kérdések vizsgálatával:

- Fel lehet-e tenni több független kérdést is egymás után (anélkül, hogy újra betöltenénk a tudásbázist)?
- Csak eldöntendő kérdéseket támogat vagy olyat is, amire egy objektum, esetleg azok valamely halmaza a válasz?
- Feltehetőek-e metakérdések (pl. „Mely tulajdonságok különböztetik meg az embert az asztaltól”)?

A következtető motor feladata, hogy a bevitt szabályok, és tények alapján megpróbáljon válaszolni a feltett kérdésekre. A keresés irányának három típusát különböztethetjük meg: előrekövetkeztető, a hátrakövetkeztető és a vegyes rendszerű következtetést. A választott stratégia befolyásolhatja, hogy mennyi következtetést kell elvégeznünk, éppen ezért érdemes ezt is megvizsgálni a következtető rendszerek elemzésekor.

Végül olyan szempontokat is figyelembe vettek, melyek sem a motor sebességét, sem kifejezőerejét nem érintik, ám használhatóságát befolyásolják. Így a következő kérdéseket vizsgálták még:

- 
- Lehet-e fejleszteni a következtető rendszeren („finomra hangolni” a kívánt alkalmazási területhez), és ha igen, mennyire egyszerű ennek folyamata?
  - Automatizálható-e tudásbázisának (időszakos vagy eseti) frissítése?
  - Támogatja-e valamely szabvány adatcsere-formátumot?
  - Létezik-e hozzá szerkesztőfelület, és ha igen, az mennyire felhasználóbarát?

A kapott eredménnyel kapcsolatosan a következő kérdések merülnek még fel:

- Csak a választ adja vissza a kérdésre vagy az ahhoz vezető utat (azaz indoklást) is?
- Képes-e felismerni (és tudatni a felhasználóval), ha a válaszhoz szükséges összefüggéseknek nem mindegyike áll rendelkezésére?
- Ha igen, képes-e olyan kérdést feltenni, mely közelebb viszi a válaszhoz?

#### **II.4.2. Következtető gépek tesztelése**

Az előző alfejezetben bemutatott szempontrendszer alapján több mint 50 következtető rendszert vizsgáltak meg, s egyben meghatároztak néhány szűrőfeltételt is, mely alapján kiválasztották a tesztelésre is érdemes rendszereket. Az alábbi feltételek nem teljesülése esetén az adott motorral nem foglalkoztak tovább részletesebben:

- Horn-formulákénál nagyobb legyen a kifejezőereje – ezt a feltételt az általuk feldolgozandó problématerület sajátosságaiból következően fogalmazták meg. A projekt keretében választott problémacsoportok legtöbbjét ugyanis nem lehet megfogalmazni ezen megszorítás mellett.
- Be lehessen vinni a motornak az (automatikusan generált) tesztek alkalmas input file-ból – az alkalmazott tudásbázist is ilyen formában fogja kapni.
- Megismerését segítse elegendő dokumentáció.

Az előírt feltételeknek megfelelő rendszereken végezték el a konkrét tesztek, amely a gyakorlati oldalról vizsgálható tulajdonságok összeméréstette lehetővé. Az így kialakított kétfázisú módszertan további előnye, hogy mivel a gépek nagy részét az első fázisban tudták zárni, így a második lépésben jelentősen rövidebb idő alatt alaposabb vizsgálatokat tudtak végrehajtani. Természetesen más vizsgálandó problémakör esetén nem feltétlenül kell, hogy szűrőfeltétel legyen például a Horn formulákénál nagyobb kifejezőerő, ezt minden esetben külön vizsgálatot igényel.

#### **II.4.3. Eredmények, értékelések**

A felállított szempontrendszer alapján az ötven vizsgált rendszer közül a következőket találták részletesebb elemzésre érdemesnek: Carine, E,

---

EPILOG, Gandalf, JTP, Otter, SNePS, SPASS (melyeken ténylegesen végre is hajtották a teszteléseket,) illetve DCTP, E-SETHEO, HOL, Isabelle, PTTP, Paradox, Vampire, Waldmeister (melyeket nagy bonyolultságuk vagy hiányos dokumentációjuk miatt nem teszteltek).

Teszteléseik egyik érdekes eredménye az volt, hogy az általános célú ontológianyelvekhez kapcsolódó következtető rendszerek (az EPILOG és a SNePS) nem voltak olyan sikeresek a válasz keresésében, mint a többi vizsgált rendszer.

A végső sorrend megállapításánál elsősorban a hatékonyságot, a következtetések gyorsaságát és helyességét vették figyelembe, mely ennek megfelelően a következő: E, SPASS, Gandalf, Otter, JTP, Carine, SNePS. A szerzők fontosnak tekintették kiemelni, hogy a sorrend nem teljesen valós, hiszen például az EPILOG nem igazán azonos célú, és támogat olyan elemeket is (például: modális kijelentéseket, valószínűségeket, időt, valamint a tudás egy jelentős részét hatékonyabban kezeli, ha a specialistái kapják meg, nem pedig elsőrendű formában kapja meg), melyekre a többi rendszer nem alkalmas. A sor utolsó három helyére, pedig azok a rendszerek kerültek, melyek az egyenlőséget nem támogatják. Azonban olyan területen, ahol nincs szükség egyenlőségre, ott elegendő lehet például a Carine alkalmazása is, (bár ekkor is érdemes lehet egyenlőséget támogató rendszert használni, mivel a Carine dokumentációja igen szegényes, a másik két motor pedig nem bizonyult túl megbízhatónak).

Az elemző tanulmány másik érdekessége, hogy az ontológiafejlesztéssel, ontológianyelvekkel, kapcsolódó alkalmazásokkal, Szemantikus Web fejlesztéssel foglalkozó projektek kapcsán kifejlesztett eszközök, következtető rendszerek közül egyik sem bizonyult tesztelésre érdemesnek. Az ontológia nyelvek összemérésével foglalkozó tanulmányok nem érintik a nyelvek és következtető gépek kapcsolatát. Az irodalomban csak az ontológia szerkesztő eszközök<sup>34</sup> kapcsán merül fel következtetési mechanizmusok vizsgálatának kérdése (OntoWeb 2002b, Corcho et al, 2003). Az elemzések azonban csak arra irányulnak, hogy rendelkeznek-e az adott eszközök beépített következtető géppel, hogy léteznek-e más illeszthető rendszerek és megvalósítható-e ezek segítségével a konzisztencia ellenőrzés, automatikus osztályozás, a kivételkezelés.

Érdekes közelebbről is megvizsgálni néhány (a MEO tanulmányban is elemzett) népszerű következtető rendszert: FaCT, JESS, Jena2, Racer.

Az említett rendszerek legfőbb hiányossága, hogy egyik sem képes tetszőleges elsőrendű formulákat feldolgozni. Bináris relációkat valamennyi motor képes kezelni, többváltozós relációkat már csak a Jess. Az alkalmazható szabályok tekintetében is Jess nyújtotta lehetőségek a legjobbak. A következtetés módszerét tekintve, azonban a Jena2 emelkedik ki az említett rendszerek közül, hiszen annak szinte bármely módjára képes.

Az adaptív vizsgarendszerrel szembeni követelmények, elvárások pontos és részletes elemzése, illetve az oktatási ontológia formalizálásához

---

<sup>34</sup> Az ontológia szerkesztő eszközök ismertetésére, szerepük bemutatására a II.6 fejezetben kerül sor.

---

használt nyelv alapján tudjuk majd azt eldönteni, hogy mely következtető motor lesz a legalkalmasabb céljainkra.

## II.5. Ontológiák értékelése

Az ontológia kidolgozásával, az adott tudásterület ontológiai struktúrájának elkészítésével nem ér véget a fejlesztők munkája. Az ontológia életciklusának szintén része a karbantartás és továbbfejlesztés, melynek elengedhetetlen előfeltétele a létrehozott modell értékelése. Az alfejezet célja az ontológia-értékelés fogalmának tisztázása és a kapcsolódó megközelítések bemutatása.

Az ontológiaértékelési módszerek alapjait a tudásbázisok és tudásalapú rendszerek értékelési módszereiből kiindulva alakították ki. Ez a megközelítés abból a feltételezésből indult ki, hogy egy ontológia olyan, mint egy tudásbázis. A fő hasonlóság közöttük, hogy mindkettő információkat gyűjt össze, melyek idővel formálódnak, fejlődnek. Azonban számos eltérést is létezik, így különbség van:

- az információk általánosságában (az ontológiák meghatározásainak általánosabbaknak kell lenniük, mint a tudásbázisok definícióinak),
- a következtető géptől való elkülönítésben,
- a kiválasztott, számítógép által olvasható nyelv szemantikájában és kifejezőerejében,
- a teljesítményértékelésben és
- abban, hogy meghatároznak-e előre jól definiált és strukturált követelményeket a fejlesztés első szakaszában. (Gómez-Pérez, 1994)

Az eltérések ellenére a tudásbázisok és tudásalapú rendszerek értékelési módszertanából számos ötlet hasznosítható az ontológiaértékelési keretrendszer kialakítása során. Így többek között a következők:

- az értékelésnek két fő típusát érdemes megkülönböztetni (technikai és felhasználói értékelés),
- a szükséges kifejezések és standard meghatározásaik,
- a technikai és felhasználói értékelési folyamatok főbb kritériumai,
- az értékelési tevékenységek beépítése az ontológiafejlesztési módszertanokba,
- már létező ontológiák értékelésére alkalmas eszközök kifejlesztése és
- értékelési modulok beépítése az ontológia-szerkesztő eszközökbe. (Gómez-Pérez, 1999)

Az ontológiaértékelés területén tehát a létező és fejlesztés alatt álló ontológiák értékelését biztosító speciális módszertanok és az ontológiaszerkesztő eszközök értékelési teljesítményére vonatkozó mélyreható információk hiánya mellett, az ontológia hasznosságát és használhatóságát

---

megítelő alkalmazásspecifikus és végfelhasználói módszertanok hiánya jelentettek problémát. (Gómez-Pérez, 1999)

Az ontológiák értékelését tehát több szempont szerint is el kell végeznünk a technikai, logikai szempontoktól kezdve egészen a felhasználói elvárásokig (Gómez-Pérez et al., 2004). Az értékelési módszereket alapvetően két nagyobb csoportba sorolhatjuk, melyeket az ontológiák életciklusának más-más szakaszaiban alkalmazunk:

- Az ontológia-értékelés (ontology evaluation) során az ontológiák technikai, logikai vagy más informatikai szabályoknak való megfelelését vizsgáljuk. A viszonyítás alapja pedig egy előre rögzített referencia keret (kompetencia-kérdések, követelmény specifikáció, a tényleges világ). Az ontológia-értékelést további két csoportra bonthatjuk:
  - Az ontológiaverifikálás (ontology verification) során az ontológiaépítés helyességét ellenőrizzük azaz, hogy a fogalmakat és relációikat valóban az ontológiaépítkezés követelményeinek megfelelően határozták-e meg.
  - Az ontológiaválidálás (ontology validation) során pedig azt vizsgáljuk, hogy az ontológia fogalmai és állításai megfelelnek-e a világ azon szeletének, amit modellezni kívántunk.
- Az ontológia kiértékelés (ontology assessment) az ontológia hasznosságának, használhatóságának és minőségének a felhasználó szempontjából történő megítélését jelenti.

Az ontológiaépítés szakaszában elsősorban a technikai, logikai kritériumoknak kell megfelelnünk. A felhasználói igényeket sem lehet figyelmen kívül hagyni, de ekkor ezek még nem elsődleges fontosságúak. Az ontológiák ellenőrzésének öt fő szempontja került meghatározásra, melyekből ez első három a technikai, módszertani szempontok ellenőrzésére vonatkozik.

- Konzisztencia (consistency): A konzisztencia vizsgálata az ontológiában szereplő fogalmak meghatározásainak logikai helyességének, érvényességének elemzését jelenti.
- Teljesség (completeness): A teljesség vizsgálata során azt ellenőrizzük, hogy szerepel-e vagy levezethető-e más fogalmakból minden olyan fogalom az ontológiában, mely szükséges a felhasználási cél tekintetében, illetve magába foglalja a meghatározások teljességének, érvényességének elemzését is.
- Tömörség (conciseness): A tömörség kritériumának akkor felel meg az adott ontológia, ha csak a szükséges fogalmak szerepelnek benne (azaz explicit redundancia nem áll fenn), illetve ha más fogalmakból sem vezethető le felesleges fogalom (implicit redundancia).
- Bővíthetőség (expandability): A bővíthetőség kapcsán azt vizsgáljuk milyen erőfeszítésekre van szükség új meghatározás ontológiába történő felvétele során, vagy ha bővíteni szeretnénk egy megha-



---

tározás információ tartalmát, anélkül, hogy meg kellene változtatnunk már jól meghatározott tulajdonságok egy halmazát.

- Érzékenység (sensitivity): Az érzékenység arra vonatkozik, hogy egy definíció tartalmának kis mértékű megváltoztatása milyen mértékű változást okoz a korábban meghatározott és szavatolt tulajdonsághalmazokon. (Gómez-Pérez et al., 2004).

A bemutatott szempontrendszer többféle ellenőrzésre is lehetőséget ad. Az esetek többségében az ellenőrzés az ontológiák vázát adó taxonómiák technikai szempontú értékelésére, azaz a technikai hibák feltárására irányul. Az értékeléshez tisztázni kell milyen elemekből épül fel egy ontológia, azaz mely elemeit és azok mely tulajdonságait kell vizsgálni. Az ontológiákban az alábbi fogalomtípusokat különböztetjük meg:

- Osztályfogalmak
- Relációfogalmak
- Tulajdonságfogalmak
- Individuumfogalmak

Továbbá az értékeléshez szem előtt kell tartani, hogy az ontológiában a klasszifikáció során hozzuk létre az osztályokat, melyek az értelmezési tartományukra nézve lehetnek:

- Egymástól elkülönült, diszjunkt osztályok – ezek az osztályok nem rendelkeznek közös példánnyal.
- Teljes, kimerítő osztályok – Ebben az esetben az értelmezési tartomány minden eleme beletartozik valamelyik osztályba.
- Diszjunkt és kimerítő osztályok

Az előzőekben meghatározott értékelési szempontok alapján a technikai értékelés során a következő három fő hibatípus előfordulását vizsgáljuk:

- inkonzisztencia
- teljesség hiánya
- redundancia.

A fenti fogalmakra támaszkodva pontosíthatjuk, mit foglalnak magukba ezek a hibatípusok:

- Inkonzisztencia hibák
  - Körkörösségi hibák – Példa: ÜGYFÉL alá van rendelve az EMBER, az EMBER alá a FÉRFI, a FÉRFI alá az ÜGYFÉL
  - Particionálási hibák
    - közös osztály diszjunkt taxonómiában – Példa: Létrehozunk a KUTYA és MACSKA osztályokat, mint az EMLŐS halmaz diszjunkt alosztályait. Hibásan járunk el, ha ezek után a VIZSLA osztályt mind a két osztály alá rendeljük.

- 
- közös individuum diszjunkt taxonómiában – Példa: Példa: Létrehozzuk a KUTYA és MACSKA osztályokat, mint az EMLŐS halmaz diszjunkt alosztályait. Hibásan járunk el, ha ezek után Plútót mind a két osztály példányává tesszük.
  - besorolatlan individuum kimerítő taxonómiában – Példa: Kimerítő felosztás esetén (páros és páratlan számok osztálya, az egész számok halmazában) egy individuum több alosztályba is be van sorolva (tehát a 3 szám a páros és páratlan osztályhoz is be lett sorolva).
  - Szemantikus inkonzisztencia hibák – Példa: A KUTYA osztály a JÁRMŰ osztály alá van rendelve.
  - Teljesség hiánya
    - Nem teljesen lefedő fogalmi klasszifikáció – Nem szerepel az ontológiában a modellezni kívánt terület minden fogalma. Példa: A hangszerek fogalmi rendszerezésekor csak a FÚVOS és HÚROS osztályokat definiálják, az ÜTŐS osztályt pedig kihagyják.
  - Particionálási hibák
    - Hiányzó elkülönülési korlát – Nem határozzák meg, hogy a részosztályok diszjunktak.
    - Hiányzó lefedési korlát – Nem határozzák meg, hogy a részosztályok teljesen lefedik az alaposztályt. Példa: Meghatározzák a PÁROS és PÁRATLAN osztályokat, mint az EGÉSZSZÁM osztály részosztályait, de nem rögzítik, hogy teljesen lefedik az alaposztályt.
  - Redundancia
    - Részosztálya reláció többszörös megadása – Két osztályfogalom között többször felveszik a részosztálya relációt. Példa: a MACSKA 'fajtája' EMLŐS, az EMLŐS 'fajtája' ÁLLAT, a KUTYA 'fajtája' ÁLLAT (indirekt redundancia)
    - Példánya reláció többszörös megadása - Egy individuum és több osztály esetében többször felveszik a példánya relációt. Példa: Plútó 'példánya' KUTYA, a KUTYA 'fajtája' ÁLLAT, Plútó 'példánya' ÁLLAT (indirekt redundancia)
    - Azonos formális osztálydefiníció – eltérő nevek alatt teljesen azonos formális definícióval rendelkezik két vagy több osztályfogalom.
    - Azonos formális individuumdefiníció - eltérő nevek alatt teljesen azonos formális definícióval rendelkezik két vagy több individuumfogalom.
-

---

### II.5.1. OntoClean módszertan

Külön ki kell emelni az OntoClean módszertant, mely megközelítésében teljesen eltér a fent bemutatott megközelítésektől. Az OntoClean szerint a fogalmakat – metatulajdonságaik mentén jellemezve – különböző típusokba sorolhatjuk. A metatulajdonságok alapján a fogalmak közti ontológiai kapcsolatokra szabályokat, kényszereket lehet felállítani, melyeket minden változtatás során/után ellenőrizni kell. Ezzel a kérdéssel külön tanulmányban foglalkoznak (Guarino és Welty, 2004).

## II.6. Ontológia-szerkesztő eszközök

Az ontológia építését, fejlesztését alkalmazását támogató eszközök köre igen széles. Bizonyos eszközök csak egy-egy funkciót látnak el, míg más eszközök az ontológia életciklus több szakaszához kapcsolódó tevékenységeket is támogatnak. A különböző ontológia eszközöket az OntoWeb kapcsolódó tanulmányában hat kategóriába sorolták be:

- Ontológiaszerkesztő eszközök
- Ontológiák illesztését, integrálását támogató eszközök (merge and integration tools)
- Ontológia kiértékelő eszközök
- Ontológia alapú annotációs eszközök
- Ontológiatárolás és – lekérdezés eszközei
- Ontológiai tanulás eszközei (ontology learning tools) (OntoWeb, 2002b)

Jelen fejezet célja az ontológiaszerkesztő eszközök összehasonlító vizsgálata. Mindazokat az eszközöket, rendszereket soroljuk ebbe a kategóriába, melyek az ontológiaépítés folyamatára koncentrálnak, függetlenül attól, hogy teljesen új ontológia kialakításáról van szó, vagy már létező ontológiák újrafelhasználásáról. Az ontológiaszerkesztő eszközök célja, hogy a formális ontológia nyelvek részletes ismerete, közvetlen alkalmazása nélkül is fel lehessen építeni az ontológiát. Az ontológiák és ontológia alapú alkalmazások rohamos terjedésével megnőtt az igény az ontológiafejlesztést hatékonyan támogató eszközök iránt. A különböző ontológiafejlesztési projektek keretében igyekeztek is minél inkább eleget tenni ezeknek az igényeknek, így rengeteg ilyen eszköz született az elmúlt években. Denny 2002-es tanulmányában már több mint 50 ontológiaszerkesztő eszközt hasonlít össze, támaszkodva az OntoWeb projekt felméréseire (OntoWeb, 2002b). Denny összehasonlító táblázatát folyamatosan bővítette és így már több, mint 90 eszköz jellemzőit foglalja magába<sup>35</sup>.

Az ontológiaszerkesztő eszközök vizsgálata elméletileg nem szükségszerű, azonban nagy méretű ontológiák esetén célszerű alkalmazásuk. Ezen eszközök segítségével gyorsan és könnyen el lehet érni a megfelelő számosságot, amely ugyanakkor már informatikai támogatást is igényel az

---

<sup>35</sup> [http://ontologia.hu/document/proj\\_doc/ontologiai\\_szerkesztok/Ontology\\_Editor\\_Survey\\_2004\\_Table\\_-\\_Michael\\_Denny.pdf](http://ontologia.hu/document/proj_doc/ontologiai_szerkesztok/Ontology_Editor_Survey_2004_Table_-_Michael_Denny.pdf)

---

ontológia elemei közötti kapcsolatok, összefüggések, a kapcsolódó axiómák és megszorítások számának gyors növekedése miatt.

Az ontológiaszerkesztő eszközök széles köre miatt nem törekszem azok elemző leírására, értékelésére, hanem csupán meghatározott szempontok szerinti összemérésükre és az oktatási ontológia szempontjából fontosnak ítélt tulajdonságaik kiemelésére, ismertetésére. A szempontok összeállításában Corcho (Corcho et al., 2003), Denny (Denny, 2002) és Dunieveld (Dunieveld et al., 1999) munkáit vettem alapul.

Az ontológiaszerkesztő eszközök összehasonlításának szempontjai ennek megfelelően a következők:

- 1) Általános jellemzők
- 2) Architektúra jellemzői
- 3) Módszertan, tudásábrázolás, ontológiaépítés jellemzői
- 4) Együttműködési képesség
- 5) Következtetés támogatása
- 6) Az eszköz alkalmazhatósága

Az OntoWeb (OntoWeb, 2002b) és Corcho (Corcho et al., 2003) tanulmányának összevetése alapján a következő hét eszköz részletes elemzésére érdemes koncentrálni: **OILED**<sup>36</sup> (Bechhofer et al., 2001), **OntoEdit**<sup>37</sup> (Sure et al., 2002, OntoEdit, 2003), **Ontolingua Server**<sup>38</sup> (Farquhar et al., 1996), **Ontosaurus**<sup>39</sup>, **Protégé 2000**<sup>40</sup>, **WebODE** és a **WebOnto**. Továbbá szükségesnek és célszerűnek találtam kiegészíteni az összehasonlítást a fentebb említett tanulmányok megjelenése óta kifejlesztett **KAON**, elemzésével illetve **PC Pack 4** szerkesztőeszköz vizsgálatával, melynek használatával kapcsolatban személyes tapasztalatokkal is rendelkezem a **GUIDE**<sup>41</sup> projekt kapcsán. A következőkben táblázatos formában összegzem az egyes eszközökre vonatkozó megállapításokat az előbbieken meghatározott szempontrendszer alapján.

### II.6.1. Általános jellemzők

Az eszközök fejlesztőiről, az aktuális verziószámáról, hozzáférhetőségükről tartalmaznak információkat. Ezek a jellemzők ugyan kevésbé meghatározóak az adott ontológiafejlesztés szempontjából legmegfelelőbb eszköz kiválasztásában, azonban annak fejlődéséről, karbantartásáról hasznos információkat nyújtanak.

---

<sup>36</sup> <http://oiled.man.ac.uk/download.shtml>

<sup>37</sup> Az elemzés elsősorban az OntoEdit ingyenes és nem professzionális változatára vonatkozik.

<sup>38</sup> <http://ontolingua.stanford.edu/>

<sup>39</sup> <http://www.isi.edu/isd/ontosaurus.html>

<sup>40</sup> <http://protege.stanford.edu/>

<sup>41</sup> **GUIDE** - Creating an European Identity Management Architecture for eGovernment (FP6 IST project – Networked businesses and governments (IST-2002-2.3.1.9)). A projekt egyik tudományos célkitűzése az európai környezet sajátosságait előtérbe helyező elektronikus személyazonosság-menedzsmentre vonatkozó fogalmi keretrendszer definiálása az elektronikus kormányzás területén.

- a) Fejlesztők
- b) Verzió
- c) Hozzáférhetőség (availability): Annak vizsgálatát jelenti, hogy nyílt forráskódú, ingyenes vagy szerzői jogvédelem alatt álló szoftverről van szó.

<b>Általános jellemzők</b>	<b>Fejlesztők</b>	<b>Verzió</b>	<b>Hozzáférhetőség</b>
<i>OilEd</i>	University of Manchester	3.5.5 (2003. október)	Nyílt forráskódú
<i>OntoEdit</i>	Ontoprise	2.7 (2004. április)	Ingyenes szoftver (szerzői jog-védelem alatt álló szoftver)
<i>Ontolingua Server</i>	KSL, Stanford University	1.0.1.045 (2003 augusztus)	Szabad Webes hozzáférés
<i>Ontosaurus</i>	ISI, University of Southern California	1.9 (2002 március)	Nyílt forráskódú
<i>Protégé-2000</i>	SMI, Stanford University	3.2.1 (2006 december)	Nyílt forráskódú
<i>WebODE</i>	Ontology Group, UPM	2.0.9 (2003 november)	Szerzői jog-védelem alatt álló szoftver / szabad Webes hozzáférés
<i>WebOnto</i>	KMI, Open University	2.3 (2002 május)	Szabad Webes hozzáférés
<i>KAON</i>	FZI Research Center & AIFB Institute, University of Karlsruhe	1.2.7 (2004 április)	Szabad Webes hozzáférés
<i>PC Pack 4</i>	Epistemics Ltd	B2.9h, (2003 április)	Szerzői jog-védelem alatt álló szoftver

4. táblázat: Ontológiaszerkesztő eszközök általános jellemzői

## II.6.2. Architektúra jellemzői

Az itt felsorolt szempontok az eszköz alkalmazásának alapfeltételeit határozzák meg, elősegítve az igényeikhez leginkább alkalmazkodó architektúra kiválasztását.

- a) Szoftver architektúra: Annak vizsgálatát jelenti, hogy önálló rendszerről, kliens / szerver megoldásról vagy 3 szintű architektúráról van-e szó.
- b) Bővíthetőség: A létrehozandó ontológia, az adott tudásterület modellezési sajátosságai miatt bizonyos esetekben fontos lehet a rendszer finomhangolási, plugin-alkalmazási lehetőségeinek vizsgálata.
- c) Ontológia tárolása: Annak vizsgálata, hogy az eszköz milyen formában tárolja el a létrehozott ontológiát (például fájlokban vagy adatbázisokban).

Architektúra jellemzői	Szoftver architektúra	Bővíthetőség	Ontológia tárolása
<i>OilEd</i>	önálló rendszer	nem lehetséges	fájlokban
<i>OntoEdit</i>	önálló rendszer	plugin-ek segítségével	fájlokban
<i>Ontolingua Server</i>	kliens / szerver	nem lehetséges	fájlokban
<i>Ontosaurus</i>	kliens / szerver	nem lehetséges	fájlokban
<i>Protégé-2000</i>	önálló rendszer	plugin-ek segítségével	fájlokban, DBMS-ben
<i>WebODE</i>	3-szintű architektúra	plugin-ek segítségével	DBMS-ben
<i>WebOnto</i>	kliens / szerver	nem lehetséges	fájlokban
<i>KAON</i>	önálló rendszer	plugin-ek segítségével	relációs adatbázisokban
<i>PC Pack 4</i>	önálló rendszer	nem lehetséges	fájlokban

5. táblázat: Ontológiaszerkesztő eszközök architektúra jellemzői

Az architektúra tekintetében igen változatos képet mutatnak a szerkesztő-eszközök. Nem lehet egyértelmű megállapítást tenni, hogy melyek a jobban preferált megoldások. Az ontológia tárolása tekintetében azonban egységesebb a kép. A Protégé, a WebODE és KAON kivételével valamennyi eszköz csak fájl formátumban képes tárolni az adatokat. A jövőbeli elképzelések tekintetében is egységesebb a kép, ugyanis többségük Java platformok és a bővíthetőség irányába kíván fejlődni.

### II.6.3. Módszertan, tudásábrázolás, ontológiaépítés jellemzői

A modellezési lehetőségek vizsgálata mellett a létrehozandó ontológia minőségét, kifejezőerejét és alkalmazhatóságát befolyásoló tényezők elemzését foglalja magába ez a szempontcsoport.

- a) Támogató módszertan: Annak elemzését jelenti, hogy a szerkesztőeszköz támaszkodik-e valamely jól kidolgozott módszertanra, mely pontosan leírja és kidolgozza tartalmi feltöltés, a tudásreprezentálás, a továbbfejlesztés elvi alapjait ezzel is növelve az eszköz elvi megalapozottságát és alkalmazásának egyszerűségét.
- b) Alkalmazott tudásábrázolási paradigma: Az alkalmazott tudásábrázolási paradigma az ontológiaépítési lehetőségeket is befolyásolja. Meghatározza, hogy az adott eszköz segítségével a fogalmakról, fogalmak közötti kapcsolatokról hogyan mondhatók ki összefüggések, milyen logikai következtetések alkalmazhatók.
- c) Alap ontológianyelv: Az alkalmazott ontológia nyelv nem független az előző szemponttól. Tulajdonképpen annak további pontosítását, finomítását, konkretizálását jelenti.
- d) Információgyűjtés (information extraction): Annak vizsgálata, hogy a kívánt ismeretek, információk (az ontológia által meghatározott módon történő) kigyűjtését és erre alapozva az ontológia kidolgozását milyen módon támogatja a tanulmányozott eszköz.
- e) Szöveges támogatás: a létrehozott ontológia kezelését, alkalmazását jelentősen megkönnyíti a felhasználók számára, ha az ontológia

egyres elemeihez szöveges hivatkozások kapcsolhatók (például definíciók, szinonimák) és a szöveges tartalom kezelésére is képes az eszköz (például szövegben való keresés, ontológia elemek szerinti szűrés támogatásával).

- f) Ontológia könyvtárak: A fejlesztés folyamatát felgyorsíthatja és megkönnyítheti az is, ha az adott eszköz példa ontológiák segítségével teszi érthetőbbé a folyamatokat, illetve ha rendelkezésre bocsát az adott eszközzel korábban létrehozott ontológiákat is, melyek újrahasznosíthatóak.

Ontológia-építés jellemzői	Módszertani támogatás	Alkalmazott tudás-ábrázolási paradigma	Alap ontológia nyelv	Információgyűjtés	Szöveges támogatás	Ontológia könyvtár(ak)
<b>OiEd</b>	nincs	leíró logika	DAML+OIL	nem támogatott	korlátozott (szinonimák, metaadatok)	nincs
<b>OntoEdit</b>	On-To-Knowledge	frame + első rendű logika	FLogic	nem (bár a professzionális verzióban lehetséges a relációs adatbázisok feltérképezése)	szinonimák meghatározása lehetséges	nincs / létezik
<b>Ontolingua Server</b>	nincs	frame + első rendű logika	Ontolingua	nem támogatott	minden betöltött ontológiában rá lehet keresni a kifejezésekre	létezik
<b>Ontosaurus</b>	nincs	leíró logika	Loom	nem támogatott	nem	nincs
<b>Protégé-2000</b>	nincs	frame + első rendű logika + leíró logika	OKBC	nem támogatott	keresés lehetséges, szinonimák megadását nem támogatja	létezik
<b>WebODE</b>	Methontology	frame + első rendű logika	belső nyelv	lehetséges WebPicker alkalmazásával	szinonimák és rövidítések megadása lehetséges	nincs
<b>WebOnto</b>	nincs	frame + első rendű logika	OCML	(hozzáférhető az OCML alapú MnM eszköz segítségével)	nincs	létezik
<b>KAON</b>	nincs	-	RDFS KAON kiterjesztése	nem támogatott	szöveges információ bármely ontológia elemhez kapcsolható, mint metaadat	nincs
<b>PC Pack 4</b>	nincs	-	XML	nem támogatott	korlátozott	nincs

6. táblázat: Ontológiaszerkesztő eszközök ontológiaépítést támogató jellemzői

A vizsgált eszközök közül csupán kettő támaszkodik az ontológiafejlesztést átfogóan leíró módszertanra (az OntoEdit és WebODE), azonban még ennél a két eszköznél sem találhatunk a projekt menedzsmentet, az ontológia karbantartását, ontológiák kiértékelését támogató funkciókat és

---

megoldásokat. A szerkesztők tehát valóban csak az ontológiaépítés szakaszára koncentrálnak és nem törekszenek a teljes ontológia életciklust lefedő funkciók kialakítására.

A tudásábrázolásnak alapvetően két módja elterjedt: keret alapú tudásábrázolás első rendű logikával kombinálva, illetve a leíró logika, az alkalmazott ontológia nyelv tekintetében, azonban már sokkal színesebb a kép. Minél szorosabban kapcsolódik egy eszköz az ontológiát formalizáló nyelvhez, kezelése annál nehezekebb lesz, annál több háttértudás megletét követeli meg a felhasználótól.

Az ontológia felépítéséhez szükséges információk kigyűjtését a WebODE kivételével egyetlen eszköz sem tudja érdemben támogatni, holott mind a kezdeti struktúra kiépítésében, mint az ontológia bővítésében fontos lenne a szerepe.

Az ontológiaépítést és az értelmezési problémák kezelését megkönnyíti és ezáltal elősegíti a valóság precízebb ábrázolását is, ha az eszköz a szöveges támogatás különböző lehetőségeit biztosítja. A táblázat alapján az is jól látható, hogy a fejlesztők felismerték ennek fontosságát és egyre több megoldás születik.

Az ontológia könyvtárak tekintetében, azonban nem ilyen kedvező a kép. Nem csak a bemutatott eszközök kapcsán, de általában is elmondható, hogy kevés olyan ontológia létezik, mely átfogó, felhasználók széles körében elfogadott és rendszeresen alkalmazott. Jól felépített ontológiák és mintapéldák jelentősen hozzájárulhatnak új ontológiák hatékony elkészítéséhez.

#### **II.6.4. Együtműködési képesség**

Az ontológiák, ontológia alapú alkalmazások számának rohamos emelkedésével egyre inkább fontossá válik az eszközök közötti átjárhatóság, az együtműködés biztosítása.

- a) Import formátumok: Annak rögzítése, hogy az adott eszköz milyen ontológianyelveken írt ontológiákat képes kezelni.
- b) Export formátumok: Annak meghatározása, hogy az adott eszköz milyen más nyelvekre képes lefordítani az alkalmazásával létrehozott ontológiákat.
- c) Ontológiák egymásba ágyazása, összefűzése (merging): Annak elemzése, hogy az adott eszköz mennyiben támogatja különböző ontológiák összehasonlítását



Együttműködési képesség	Import formátumok	Export formátumok	Ontológiák egymásba ágyazása
<b>OilEd</b>	RDF(S), OIL, DAML+OIL	OIL, RDF(S), DAML+OIL, SHIQ, HTML,	nem lehetséges
<b>OntoEdit</b>	XML, RDF(S), FLogic, DAML+OIL	XML, RDF(S), FLogic, DAML+OIL	egyszerű összefűzés lehetséges
<b>Ontolingua Server</b>	Ontolingua, IDL, KIF	KIF, CLIPS, CML ATP, EpiKit, IDL, KSL, Loom, OKBC, Prolog szintakszis	fél-automatikus módon a Chimaera segítségével lehetséges
<b>Ontosaurus</b>	Loom, IDL, KIF, C++	Loom, IDL, KIF, C++	nem lehetséges
<b>Protégé-2000</b>	XML, RDF(S), XML Séma	XML, RDF(S), XML Séma, FLogic, CLIPS, Java, HTML	fél-automatikus módon PROMPT segítségével
<b>WebODE</b>	XML, RDF(S), CARIN	XML, RDF(S), OIL, DAML+OIL, CARIN, FLogic, Prolog, Jess, Java, HTML	nem ellenőrzött módon szinonima és hiperonima táblázatok segítségével
<b>WebOnto</b>	OCML	OCML, Ontolingua, GXL, RDF(S), OIL	nem lehetséges
<b>KAON</b>	XML, RDF(S)	RDFS, Protégé RDFS	nem lehetséges
<b>PC Pack 4</b>	XML, RDF	XML	nem lehetséges

7. táblázat: *Ontológiaszerkesztő eszközök együttműködési képességei*

Jó látható, hogy az eszközök többsége számos exportálási, importálási lehetőséggel bír, mely igen hasznos átjárhatóság tekintetében. Problémát jelent azonban, hogy nincsenek empirikus adatok arra vonatkozóan, hogy milyen hatással van az ontológia minőségére, ha az egyik nyelvről lefordítjuk a másikra. A legfontosabb kérdés az lenne, hogy ilyen fordítás alkalmával a modellezett tudás mekkora hányada veszik el, mennyiben játszanak ebben szerepet a nyelvek felépítésének, struktúrájának különbségei, illetve egyéb szubjektív tényezők.

A már létező ontológiák újrafelhasználásának kérdésével szinte valameny-i ontológiaépítési módszertan foglalkozik, s kiemeli ennek fontosságát. Ennek ellenére a szerkesztőeszközök többsége semmilyen megoldást nem nyújt a problémára, illetve az eddig kialakított eszközök sem kiforrottak még.

### II.6.5. Következtetés támogatása

Adott terület tudásának modellezése, pontos leírása mellett, az ontológia alkalmazhatóságának fontos eleme, hogy lekérdezéseket, következtetéseket tudjunk végezni. Az összehasonlítás e része csupán azt vizsgálja, hogy rendelkeznek-e az adott eszközök a következtető géppel és nem foglalkozik azok következtetési képességeivel. (A következtető gépek részletes elemzése a II.4. fejezetben található.)

- a) Saját következtető gép.
- b) Illeszthető következtető gépek.

- c) Konzisztencia ellenőrzés: A levonható következtetések helyességét és teljességét, a következtetés gyorsaságát az is meghatározza, hogy az eszköz milyen mértékben támogatja az automatikus konzisztencia ellenőrzést. A konzisztencia ellenőrzés pedig az ontológia szintaktikai, szemantikai, logikai helyességének vizsgálatát jelenti.
- d) Automatikus osztályozás: Az automatikus osztályozás a konzisztencia ellenőrzés részét képezi, tehát nem tekinthető független vizsgálati szempontnak, azonban az összehasonlító elemzések többségében külön ki szokták emelni. Az automatikus osztályozás új fogalmak egyed tulajdonságok és osztály relációk segítségével történő meghatározását jelenti.

Az OntoWeb tanulmány szerzői fontosnak ítélték a kivételek kezelésére vonatkozó szempont vizsgálatát. Azonban az eddig kifejlesztett eszközök egyike sem nyújt megoldást erre a problémára, ezért ennek további vizsgálatától eltekintek.

Következtetés támogatása	Beépített következtető gép	Illeszthető következtető gép	Konzisztencia ellenőrzés	Automatikus osztályozás
<i>OilEd</i>	van (FaCT)	nincs	van, FaCT segítségével	van
<i>OntoEdit</i>	nincs (OntoBroker a professzionális változatban)	nincs	van, (megszorítások és konzisztencia ellenőrzése külön plugin segítségével)	nincs
<i>Ontolingua Server</i>	nincs	ATP	nincs, (Chimaera segítségével azonban alapos ellenőrzés végezhető)	nincs
<i>Ontosaurus</i>	van	van	van	van
<i>Protégé-2000</i>	van (PAL)	Jess, FaCT, FLogic	van, (minden facet ellenőrzésre kerül, a felhasználó által megadott megszorítások csak külön kérés esetén)	nincs
<i>WebODE</i>	Van (Prolog)	Jess	van	nincs
<i>WebOnto</i>	van	nincs	van	nincs
<i>KAON</i>	nincs	van	van	nincs
<i>PC Pack 4</i>	nincs	nincs	csak logikailag konzisztens modellek létrehozását engedi	nincs

8. táblázat: Következtetés támogatása az ontológiaszerkesztő eszközöknél

A bemutatott eszközök mindegyike támogatja a következtetést vagy saját, vagy illesztett, vagy akár többféle következtető gép szerkesztőeszközhöz való kapcsolásával. Mindezekből azonban nem lehet messzemenő követ-

---

keztetéseket levonni, hiszen az alkalmazott ontológianyelv, az adott motor tulajdonságai is erősen befolyásolják, hogyan és mire tudjuk alkalmazni ezeket a lehetőségeket. Pozitívumként emelhető ki, hogy az ontológia konzisztensségének ellenőrzésére minden eszköz nyújt megoldásokat, azonban – csak úgy mint az exportálási / importálási lehetőségek esetén – itt is alaposabb vizsgálatot igényel, hogy ezek a fél-automatikus / automatikus eszközök mennyire pontos és alapos ellenőrzést végeznek, mennyire megbízhatóak. Az eddig kidolgozott megoldások igazán a szintaktikai hibák keresésére alkalmasak, a szemantikai hibák megtalálására annál kevésbé.

#### **II.6.6. Az eszköz alkalmazhatósága**

Olyan szempontok elemzése, melyek nincsenek közvetlen hatással a létrehozandó ontológia minőségére, de fejlesztés, építés folyamatát megkönnyítik, támogatják.

- a) Grafikus nézet: Annak vizsgálata, hogy az ontológia felépítését, nyomon követését, szerkesztését grafikus megjelenítéssel támogatja-e az adott eszköz.
- b) Webes támogatás: Annak elemzése, hogy van-e lehetőség az eszköz Weben keresztüli alkalmazására, illetve Web kompatibilis ontológiák létrehozására.
- c) Többfelhasználós üzemmód: Annak vizsgálatát jelenti, hogy az eszköz nyújt-e megoldásokat az ontológia több felhasználó által egy időben történő fejlesztésére.

<b>Eszközők alkalmazhatósága</b>	<b>Grafikus nézet</b>	<b>Webes támogatás</b>	<b>Több-felhasználós üzemmód</b>
<b><i>OilEd</i></b>	alárendelt osztályokat tartalmazó fájlok böngészése lehetséges	RDF URI-k, korlátozott névterek és XML Séma, HTML export	nincs
<b><i>OntoEdit</i></b>	van, navigációra és szerkesztésre	RDF URI-k	tervezik többfelhasználós üzemmód funkcióinak bevezetését
<b><i>Ontolingua Server</i></b>	nincs	szolgáltatásokhoz való webes hozzáférés	felhasználók hozzáférési szintjének meghatározásával
<b><i>Ontosaurus</i></b>	osztály hierarchia böngészése	nincs	globális zárolás
<b><i>Protégé-2000</i></b>	osztályok és tulajdonságok böngészése plugin-ek segítségével, hálószerű gráfok szerkesztése	referencia ontológiák megadása URI-vel, korlátozott névtér alkalmazás	a több-felhasználós szerver és kliens fejlesztés alatt van
<b><i>WebODE</i></b>	gráf nézet, melyben az osztályok, relációk, partíciók, metatulajdonságok stb. szerkeszthetők	névterek, XML séma alkalmazása	van, felhasználó csoportok szinkronizálási, autentikációs és hozzáférési jogainak megadásával
<b><i>WebOnto</i></b>	osztály-relációk gráf nézete	"Web service deployment site"	globális írási jogosultság zárolása, a változtatásokról való értesítéssel
<b><i>KAON</i></b>	igen, nagy méretű ontológiák felépítésére hozták létre	van (Java Webstart)	egyidejű hozzáférési jogosultság ellenőrzés, tranzakció orientált zárolással
<b><i>PC Pack 4</i></b>	ER diagrammok, osztály-hierarchia, OO nézetek	HTML kimenet	van

9. táblázat: Az ontológiaszerkesztő eszközök alkalmazhatóságát befolyásoló tényezők

A grafikus nézet, a taxonómiák grafikus ábrázolása ma már szinte magától értetődő része a szerkesztőeszközöknek, mely nem kérdéses, hogy jelentősen megkönnyít alkalmazásukat.

Minél szélesebb körű az adott eszköz által nyújtott webes támogatás köre, annál könnyebb alkalmazása is. Az eszközök egy része nemcsak webes támogatást nyújt, hanem azon keresztül is működtetik. Ezek az eszközök (például Ontolingua Server) általában a többfelhasználós üzemmód tekintetében előbbre járnak, mint lokálisan telepítendő társaik. Ez utóbbiak előnye azonban, hogy sokkal gyorsabban működnek és dolgozzák fel a bevitt adatokat. Ugyan a több felhasználó által egy időben történő fejlesztést is egyre több eszköz és egyre hatékonyabban kezeli, többségüknél még mindig problémát jelent, hogy a változások nem vagy csak nehézkesen követhetők.

## II.7. A kutatás elvi modellje

A dolgozat egyik fő célkitűzése, hogy a gazdaságinformatikus szakon megszerezhető kompetenciákon keresztül ontológiai kapcsolatot teremtsen a munkaerőpiac által támasztott követelmények, valamint az oktatott tárgyak tartalma között. Szintén kiemelt cél, hogy a modell támogassa a hallgatók által elsajátított ismeretek adaptív módon történő felmérését, különös tekintettel az oktatás eredményességére és a hiányterületek tételes feltárására.

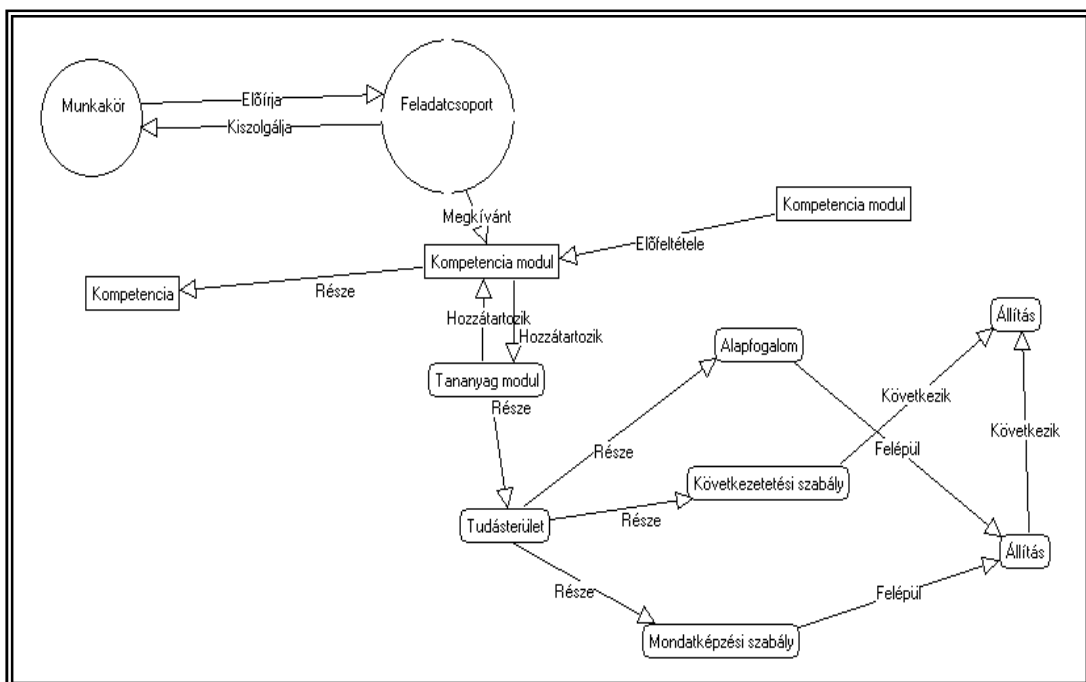
Az elméleti háttér feldolgozásával és a kapott eredmények segítségével elvégezhető a gazdaságinformatikus képzés keretében oktatott tárgyak tudásszerkezetének ontológiai modellezése a fent meghatározott célok figyelembevételével. Az fejezet célja az eltérő területek (tananyagok) közös vonásainak megragadására alkalmas modell kialakítása, az oktatási ontológia alapelemeinek és a köztük lévő kapcsolatok meghatározása.

### II.7.1. Kiindulási modell

Az általam tervezett ontológia kiindulási alapja az Információrendszerek Tanszék által kidolgozott modell, melyben a képzési programokban célként megfogalmazott kompetenciák állnak a középpontban (Szabó, 2006). A kiindulási modell szerint a kompetencia modulokat hozzákapcsoljuk az egyes tananyagmodulokhoz, annak érdekében, hogy végig lehessen követni milyen ismerethalmazzal rendelkezik egy adott hallgató.

A kiindulási modellben a kompetencia-modulokat a tananyagmodulokkal történő összekapcsolásán túl meg lehet feleltetni a munkaerőpiac elvárásainak is az adott munkakör ellátásához szükséges feladatcsoportok és az általuk megkívánt kompetencia-modulok megfeleltetésével.

A kiindulási modellt a következőképpen ábrázoltuk:



7. ábra: Az Oktatási ontológia kiindulási modellje

---

### **II.7.1.1. Javasolt módosítások**

A kitűzött célok elérése érdekében az előzőekben leírt modell átalakítására van szükség. Emellett ismertetem, hogyan képzelem a kidolgozandó vizsgáztatási rendszer és az oktatási ontológia kapcsolódásának és együttműködésének lehetőségeit.

A modell módosítása mellett tisztázni kell a formalizálásra vonatkozó kérdéseket is. Az ontológia alapvető jellemzőit jelentősen befolyásolhatja a formális leírás nyelve. Az egyes ontológianyelvek alapvetően abban különböznek egymástól, hogy milyen lehetőségeket biztosítanak a tényleges tudás kifejezésére és rögzítésére. Az Oktatási ontológia hasznosítási lehetőségeit is meghatározza az alkalmazott ontológianyelv. A megfelelő ontológianyelv kiválasztása során figyelembe kell venni az alkalmazni kívánt ontológiaépítési módszertan sajátosságait, illetve a hozzáférhető ontológiaszerkesztő eszközök körét is.

Az ontológianyelveket összehasonlító elemzés eredményeként az OWL DL tűnik a legalkalmasabb választásnak. Leíró logikára épül, mely a leginkább használható ontológianyelvek közös alapját adja. Egy OWL DL szintű leírás alkalmas lehet az osztályok közötti kapcsolatok, illetve a modell konzisztenciájának ellenőrzésére is. Az OWL DL-en alapuló formális leírás előnye, hogy XML-es szintaxisa miatt az ontológia másik nyelvre történő átalakítása viszonylag egyszerű. Továbbá az is fontos szempont, hogy több szerkesztőrendszer is készült hozzá (így például a Protégé).

Az alábbiakban egyrészt kifejtem az általam tervezett modell sajátosságait, másrészt leírom tesztkérdések tárolásának modelljét, a modell szerepét és kapcsolódását az oktatási ontológiához.

Dolgozatomban az oktatás modelljét és annak meghatározott részeit fogom kidolgozni, továbbá a modellben keretet teremtek arra, hogy a párhuzamosan folyó projekteknél elkészüljenek az oktatási ontológiához kapcsolódó tesztkérdések, és ki fogom fejteni, hogy az adaptív tesztelés milyen módon használhatja ezt az ontológiát.

### **II.7.2. Oktatási ontológia modellje**

Az ontológiaspecifikáció, az ontológia vázának felépítése az előzőekben leírt modell finomítását igényli. A következőkben ismertetem módosításokat, illetve az ontológiát felépítő osztályokat a köztük fennálló kapcsolatokat.

#### **II.7.2.1. Munkakör**

A „Munkakörök” osztályába kerülhet minden olyan szakma, foglalkozás, feladatkör, melyhez szükséges kompetenciákat biztosítja a gazdaságinformatikus képzés. Az ontológiában a munkakörök közötti hierarchia is kialakítható, mellyel jelezhető, ha egy munkakör egy másik munkakör speciális esete (ami egyben azt is jelenti, hogy a speciálisabb munkakör igényli az általánosabb összes feladatát).

---

### **II.7.2.2. Feladatcsoport és Kompetencia modul**

A kiindulási modellben a „Feladatcsoport” és „Kompetencia modul” osztályok teremtették meg a munkakörök és a képzés során elsajátítható tudásterületek között a kapcsolatot. A tervezett megvalósítás szempontjából túl általánosak ezek az osztályok, ezért e helyett az alábbi elemek alkalmazását javaslom:

- Legyen „Feladat” és „Kompetencia” osztály;
- Egy munkakör „előírja”-„kiszolgálja” viszonyban közvetlenül feladatokkal legyen (így egy munkakör több feladatot ír elő, nem pedig egy feladatcsoportot)
- Egy-egy feladat pedig „megkívánja” viszonyban legyen a kompetenciákkal.

Ez azt jelenti, hogy nem a halmazok (kompetenciamodulok, feladatcsoportok) kapcsolódnak egymáshoz, hanem az elemeik. Ezzel az eredeti modellt finomítottuk. Segítségével két kompetencia-modul összehasonlítása, illetve a kettő közti különbségek kiszámolása is könnyebben kivitelezhető.

Emellett továbbra is megtarthatjuk mind a „Feladatcsoport” osztályt, mind a „Kompetenciamodul” osztályt, hogy a feladatok, illetve kompetenciák halmazait is meg tudjuk adni. Adott feladatcsoporthoz „*eleme*” relációval kapcsolódnak a feladatok, míg a „Feladatcsoport” osztály – a „Feladat” osztályhoz hasonlóan – „*előírja*”, „*kiszolgálja*” relációban van a „Munkakör” osztállyal. A kompetencia modulnak szintén elemei a kompetenciák és az osztály a „*megkívánja*” relációval kapcsolódik a „Feladat” osztályhoz.

### **II.7.2.3. Tananyag modul**

A tervezett módosítás lényege az, hogy míg az eredeti modellben a tudásterület a tananyag modulon keresztül kapcsolódott a kompetencia modulhoz, addig a tervezett modellben a tudásterület és a kompetencia közvetlen kapcsolatba kerül, „*igényli*” és „*biztosítja*” relációkon keresztül. (Adott kompetencia igényli a tudásterület ismeretét. A tudásterület ismerete biztosítja a kompetencia meglétét.)

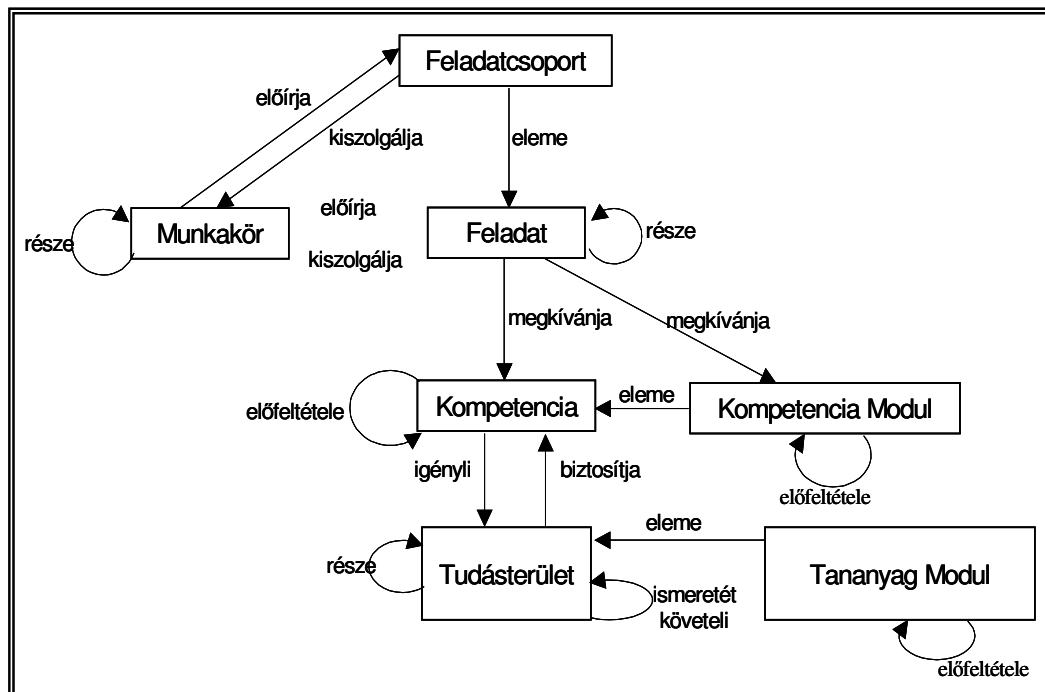
Amennyiben a rendszer alkalmazása ezt igényli, úgy a modellben helyet kap a Tantárgy/Tananyag modul, a „*eleme*” illetve „*hozzátartozik*” reláción keresztül kapcsolódva tudásterülethez. (Tananyagnak eleme a tudásterület, illetve az adott tudásterület mindig hozzátartozik valamely tananyaghoz.)

A tudásterületek osztálya az ontológia azon csomópontja, ahol a modell két részre válik a következők szerint:

- a tudásterületek és munkaerő-piaci elvárások között kapcsolatot teremtő modellrészre, amely az eddig ismertetett elemekből épül fel és
- a tudásterületek belső szerkezetére.

Mindezek alapján az ontológia tudásterületek „feletti” szintjét a következőképpen ábrázolhatjuk:

- a téglalapok osztályokat jelölnek
- az összes nyíl 0-N kapcsolat jelöl (azaz egy kompetenciának több előfeltétele is lehet, egy munkakör több feladatot is előírhat, illetve lehet, hogy egy kompetencia nem rendelkezik előfeltételekkel (ezért 0)).



8. ábra: Tudásterületek és munkakörök közötti ontológiai kapcsolat

#### II.7.2.4. Tudásterületek

Az eddigi modellben és annak továbbfejlesztéseiben a tudásterületek rész-tudásterületekre, illetve alapfogalmakra voltak felbontva. Kitzűzött céljainak érdekében (jelen esetben elsősorban az adaptív vizsgarendszer kidolgozása érdekében) célszerű a tudásterületek belső szerkezetét is tovább finomítani.

Megtartjuk a tudásterület „része” relációt (azaz a tudásterületek rész-tudásterületekre történő bontását) és felvesszük az „ismeretét követeli” relációt. Ez utóbbira az adaptív vizsgáztatás miatt van szükségünk. Ha a vizsgáztatás során kiderül, hogy adott tudásterületen jelentős hiányosságai vannak a vizsgázónak, akkor rá tudunk kérdezni azokra a tudásterületekre, melyekre épül. Szintén a vizsgáztatási szempontok miatt, a tudásterületek elemeként felvesszünk (felsorolunk) minden olyan objektumot, amelyekre vonatkozóan tesztkérdések tehetők fel. Nevezzük a tudásterületek (rész-tudásterületek) ezen elemeit TUDÁSELEMEK-nek.

Az Oktatási ontológia modelljében a következő tudáselemek szerepelnek:

- Alapfogalom (Például: rendszer, rendszerfejlesztési módszertan)
- Összefüggés / Állítás / Tétel (Például: MERISE rendszerszemléletű módszertan)



- 
- Példák (Például: SSADM, MERISE)

Indokolt a tudáselemek között relációk bevezetése is.

Az egyes tudáselemek belső szerkezetét azonban nem vizsgáljuk mélyebben. (Például nem feltétlenül szükséges megvizsgálni, hogy egy adott fogalom hogyan épül más fogalmakra.) A tudáselemeket olyan szempontból vizsgáljuk, hogy miként tudnak az adaptív teszteléshez támogatást nyújtani.

Azért is célszerű ezt az utat követni, mert a tudáselemek belső szerkezetének teljes mélységű elemzése hatalmas, majdhogynem kivitelezhetetlen munka lenne. Ehhez képest a tudáselemek felsorolása relatíve egyszerűbb feladat, és ha hozzákapcsoljuk a tesztkérdéseket, akkor az már működőképpé teszi a rendszert.

Ez azonban nem zárja ki, hogy a tudáselemek tartalmára vonatkozó szemantikai összefüggéseket vizsgáljuk. Így érdemes megvizsgálni az összefüggések mindkét oldalát és meghatározni a kapcsolatokat azokkal az alapfogalmakkal, melyeket felhasznál az összefüggés, illetve azokkal az alapfogalmakkal, melyekre következtetünk az összefüggés alapján. Vegyünk egy példát a Projektmenedzsment tananyagából és vizsgáljuk a következő összefüggések és alapfogalmak kapcsolatát:

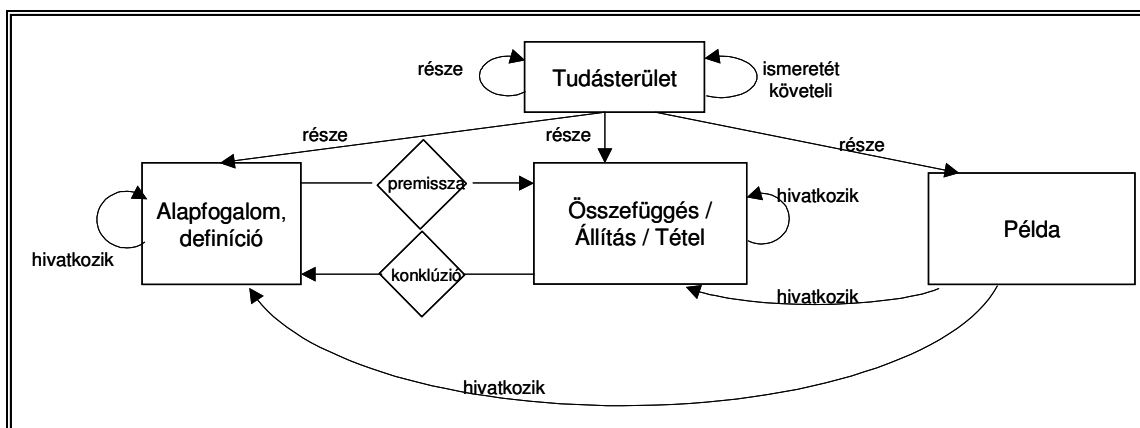
- Összefüggés 'A': Prince projekt szervezet felépítése
- Összefüggés 'B': Minőségi szemle lefolytatása
- Fogalom 1: projekt board
- Fogalom 2: projekt menedzser
- Fogalom 3: minőségbiztosítás

Az Oktatási ontológiában fel kell vennünk az 'A' összefüggés kapcsolatát mind a három felsorolt fogalommal. Ennek azért is van kiemelt fontossága, mert az egyes fogalmak nem csak egy összefüggéshez kapcsolódhatnak. Így a Fogalom 3 a 'B' összefüggéssel is kapcsolatban áll, ehhez is hozzá kell kötni.

A CommonKADS módszertan megközelítése szerint az ilyen típusú relációkat kapcsoló objektumként kell felvenni az ontológiába. A kapcsolatok is osztályok, de speciális tulajdonságokat várunk el tőlük, így például a kapcsolat mindkét oldalán megjelenő jellemezőket megfelelően le kell írni. A projektmenedzsment példára is alkalmazható megoldás lenne a 'kapcsolat' osztály bevezetése (amely azt fejezné ki, hogy az ontológia két eleme hierarchikus kapcsolatban áll). Bizonyos esetekben a kapcsolatok sorrendisége is számít (például a minőségi szemle adott cselekvések meghatározott sorrendjéből épül fel). Ezek pontos leírása megköveteli, elkerülhetlenné teszi a valóság elemzését, melyhez szükség van a 'kapcsolat' osztály felvételére. Mindezeknek megfelelően ezt a reláció típust a „*premissza*”, „*konklúzió*” kapcsolóelemek „Alapfogalom” és „Összefüggés / Állítás / Tétel” osztályok közötti felvételével írrom le.

A „*hivatkozik*” relációval pedig megadhatjuk, hogy az adott tudáselem, mely másik tudáselemre épül. Hivatkozhat egy alapfogalom, vagy összefüggés másik alapfogalomra, vagy összefüggésre, illetve egy példa a másik két tudáselem bármelyikére.

Az előző jelölés rendszert alkalmazva a tudásterületek belső szerkezetét a következőképpen ábrázolhatjuk:



9. ábra: A tudásterületek belső szerkezetének ontológia modellje

### II.7.3. Tesztkérdések halmaza

Ahhoz, hogy az oktatási ontológia megfelelően tudja támogatni a létrehozandó vizsgáztatási rendszert, bizonyos elvi alapokat, megfontolásokat már ebben a modellben is tisztázni kell.

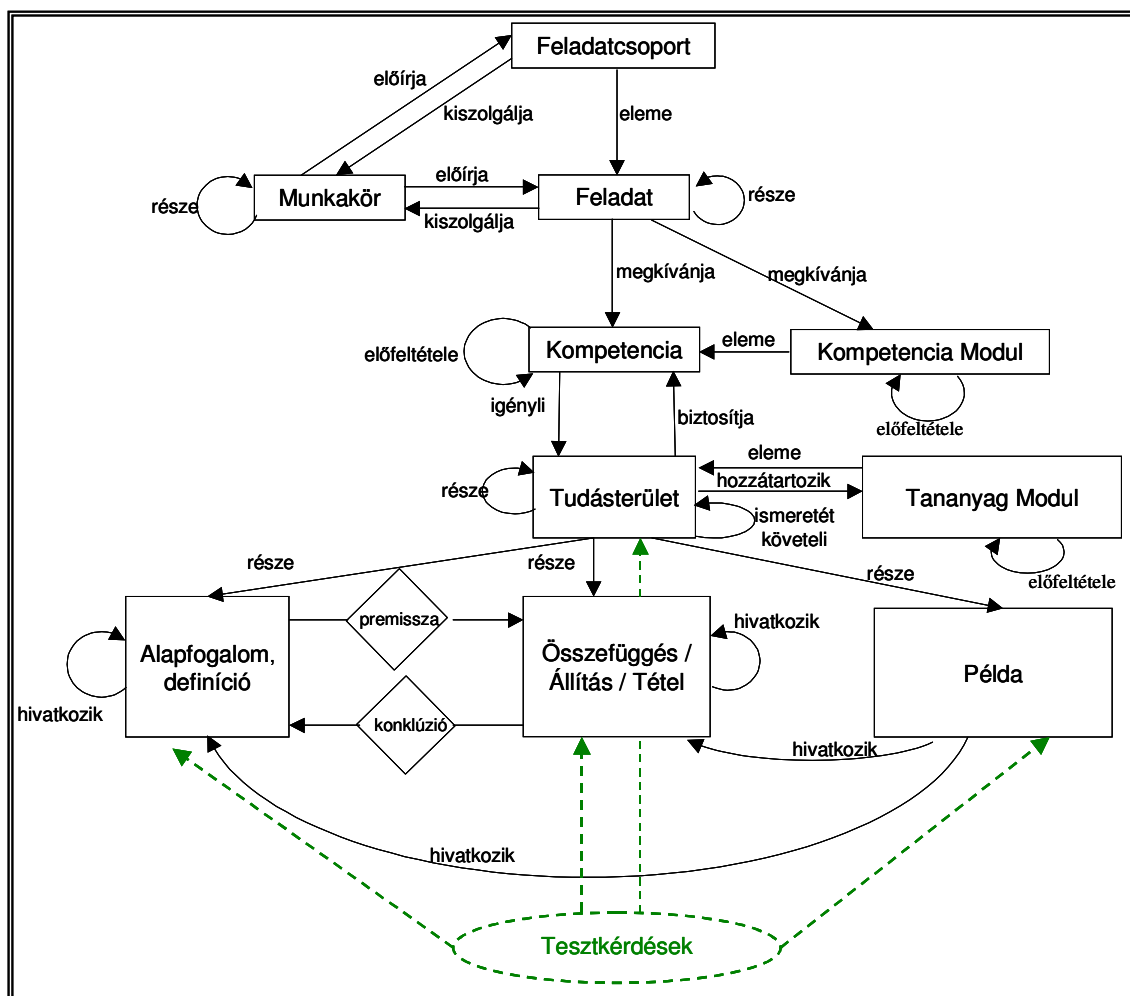
A vizsgáztatási rendszer egyik pillére a tesztkérdések halmaza lesz, melyekről a következőket mondhatjuk:

Tesztkérdések jellemzői:

- Mindig egy vagy több „Tudáselemhez” kapcsolódik és bármely tudáselemhez akár több tesztkérdés is tartozhat (ezáltal a tesztkérdések halmaza az oktatási ontológia által strukturálódik)
- Minden kérdéshez súlyok is rendelhetők
- Tesztkérdések elemei (abból kiindulva, hogy MCQ<sup>42</sup> típusú tesztek-ről van szó):
  - Kérdés
  - Valós válasz
  - Álválaszok
- További meghatározható tulajdonságaik:
  - Elmélet/gyakorlati
  - Minősége stb.

<sup>42</sup> Multiple Choice Question

Az oktatási ontológia végleges és teljes modellje a következő ábrán látható. Az ábra egyben azt is megmutatja, hogyan kapcsolódnak a tesztkérdések az ontológia egyes elemeihez (szaggatott vonallal jelölve, hogy jól látható legyen, hogy a tesztkérdések nem képezik az ontológia részét).



10. ábra: Az Oktatási ontológia modellje

#### II.7.4. Az Oktatási ontológia modell alkalmazása

A modell gyakorlati alkalmazásának bemutatása mellett az Oktatási ontológia fejlesztési módszertanának kidolgozása is célja e fejezetnek. A kiinduláshoz szükséges elemzések és a fent bemutatott ontológia modell kialakítása is a fejlesztés részét képezi, így a módszertannak is ki kell térnie. Az érthetőség és átláthatóság érdekében a fejlesztés ezen első lépését is összefoglalom az alkalmazni kívánt módszertani megközelítés ismeretése során.

A modell alkalmazhatóságát a tantárgyak tudásanyagának tényleges reprezentálásával lehet szemléltetni. A gazdaságinformatikus képzés keretében BSc szinten több, mint negyven, MSc szinten pedig több, mint húsz

---

tárgyat<sup>43</sup> oktatunk egyetemünkön. Dolgozatomnak nem célja és nem is fér kereteibe valamennyi tárgy struktúrájának és kapcsolódó vizsgakérdéseinek kidolgozása. A cél olyan tananyag feldolgozása, mely alátámasztja a megközelítés jogosultságát és bemutatja az ontológia modell egyes elemeinek szerepét. A kérdés az, melyik tárgy a legmegfelelőbb erre a célra.

Számos olyan tantárgy létezik, melyek természetüknél fogva jól strukturáltak és így a modellbe is könnyen beilleszthetők. Nem igényelne sok erőfeszítést eldönteni, hogy a matematikai fogalmakat miként illesszük modellünkbe. Például a 'háromszög', 'háromszög súlyvonala' és 'háromszög súlyvonalai egy pontban metszik egymást' fogalmakról könnyen megállapítható, hogy míg az első kettő az „Alapfogalom”, addig a harmadik az „Összefüggések” osztályba tartozik. Emellett a matematikai fogalmak, tételek és tudásterületek egymásra épülése is világos és egyértelmű. Érdekes kérdés azonban, hogy kevésbé strukturált tananyagok esetében, mint például a különböző menedzsment tárgyak, hogyan soroljuk be a fogalmakat az egyes osztályokba. Választásom éppen ezért a Tudásmenedzsment kurzus tananyagának feldolgozására esett. Tanszékünk sok éves tapasztalattal rendelkezik e tárgy oktatásában, melyet az új bolognai rendszer keretében is oktatunk.

#### ***II.7.4.1. Az Oktatási Ontológia építésének módszertana***

Teljesen új módszertan kialakítása helyett érdemes a már létező módszertanokat számba venni, elemezni és végül a legalkalmasabb jelöltet testreszabni az Oktatási ontológia sajátosságainak figyelembevételével. Az ontológiaépítés folyamatát leíró módszertanok összehasonlító elemzése<sup>44</sup> alapján az On-To-Knowledge és a Methontology megközelítését lehet kiemelni, melyek az értékelési kritériumok többségének jól megfeleltek. Néhány szempontból ugyan jobban teljesített a Methontology, azonban továbbra is szem előtt kell tartani, hogy a cél az Oktatási ontológia széles körű alkalmazhatóságának biztosítása, ezért a módszertani megközelítések közül is érdemes a közérthetőbb megoldást választani. Más megközelítésekkel ellentétben az On-To-Knowledge módszertan nem az ontológiára, hanem az alkalmazásvezérelt ontológiafejlesztésre helyezi a hangsúlyt, amely jelen kutatás esetében is kiemelt fontossággal bír. Az On-To-Knowledge további előnye, hogy lehetővé teszi az ontológia dinamikus bővítését, az ontológiafejlesztés bármely szakaszában lehetőség van a korábbi szakaszokhoz való visszacsatolásra – a lépések logikai sorrendjének betartásával –, és a fogalmi rendszer felépítésének módjára vonatkozóan sincs semmilyen korlátozás<sup>45</sup>.

Mindezeknek megfelelően az On-To-Knowledge lépéseit követve írom le az Oktatási ontológia fejlesztési módszertanát, melynek során a Tudásmenedzsment tárgy ontológiai modelljének elkészítésével mutatom be a gyakorlati szempontokat.

---

<sup>43</sup> A gazdaságinformatikus képzés BSc és tervezett MSc tantervét a Függelék V.3 alfejezetében található.

<sup>44</sup> Lásd a II.2 fejezetben.

<sup>45</sup> Ezt a rugalmasságot csak az On-To-Knowledge biztosítja.

---

### (1) *Megvalósíthatósági vizsgálata*

Nemcsak az alkalmazott technológia, hanem más szervezeti, környezeti tényezők is meghatározzák, hogy a fejlesztendő rendszer mennyire lesz sikeres. A megvalósíthatósági vizsgálat keretében meg kell határozni az ontológiaépítés elsődleges célját, a modellezni kívánt szakterület jellemzőit, az érintettek / potenciális felhasználók körét. Az elemzés alapján határozhatók meg az ontológiával szembeni elvárások, a követelményspecifikáció.

Az Oktatási ontológia elsődleges célja a gazdaságinformatikus képzés keretében oktatott tárgyak tudásanyagának modellezése a terület fogalmainak és a fogalmak közötti szemantikai viszonyok meghatározásával. Szem előtt kell tartani, hogy oktatási anyagok modellezése a cél, melyeknél az alapfogalmaknak, az összefüggéseknek és ezek megértését támogató magyarázatoknak kiemelt szerepe van. Az egyes területek általános jellemzőin túl, az adott intézményben követett elvek, elfogadott módszerek illetve az oktatók gondolkodásmódja, tanítási technikái is befolyásolhatják az ontológia felépítését és tartalmát.

Az oktatási intézmények elsődleges feladata a hallgatói igényeknek való megfelelés. Ugyanakkor nem lehet figyelmen kívül hagyni a munkaadók elvárásait és az Európai Felsőoktatási Térséghez is igazodnunk kell. Ráadásul az adott képzésben résztvevő oktatók köre is igen széles lehet. Alkalmazott módszereik és oktatási gyakorlatuk igen eltérő lehet, de ennek ellenére is biztosítani kell a tárgyak egymásra épülését. A kidolgozásra kerülő ontológia közvetlen felhasználói körét az oktatók, illetve – az adaptív vizsgáztatás révén – a hallgatók adják, de közvetett módon a munkaerőpiac szereplő is a felhasználó kör részét képezik. Ráadásul az oktatóknak a fejlesztésben, illetve a karbantartásban, a frissítések elvégzésében is szerepet kell vállalniuk. A rendszer sikerességét és megvalósíthatóságát alapvetően ezek a környezeti és szervezeti tényezők határozzák meg.

### (2) *Kiindulási fázis*

A kiindulási fázisban kerül sor az ontológia követelményspecifikációjának elkészítésére. Tisztázni kell mi az ontológia célja, tárgyterülete, milyen alkalmazásokat fog támogatni, milyen tudásforrásokat fog felhasználni és ebből milyen módon kell/lehet az információkat kinyerni, illetve meghatározza a tervezési, szerkesztési elveket is. Szintén ebben a szakaszban kerül sor az esetlegesen újrhasználható ontológiák részletes vizsgálatára.

Az **ontológia célja** – ahogy ezt korábban már kifejtettem – egyrészt a gazdaságinformatikus szakon megszerzhető kompetenciákon keresztül ontológiai kapcsolatot teremteni a munkaerőpiac által támasztott követelmények, valamint az oktatott tárgyak tartalma között. Másrészt cél, hogy a modell támogassa a hallgatók által elsajátított ismeretek adaptív módon történő felmérését, különös tekintettel az oktatás eredményességére és a hiányterületek tételes feltárására. Ez egyben meghatározza az ontológia **tárgyterületét** is, ami nem más, mint a gazdaságinformatikus képzés tantárgyainak tudásanyaga. A **kapcsolódó alkalmazás** pedig az adaptív vizsgarendszer lesz, melynek működését elsősorban a modellezett tudás-

---

anyag és a tesztkérdések összekapcsolásának biztosításával lehet elérni. Jelen esetben kiemelten a Tudásmenedzsment tárgy elemzésére fókuszálunk. A **tudásforrást** az oktatáshoz kidolgozott előadás anyagok<sup>46</sup> és a kötelező irodalom<sup>47</sup> adja (valamennyi tárgy esetében). A Tudásmenedzsment témakörét, illetve tananyagát feldolgozó ontológia még nem született, de más tárgyak, például a Matematika területén már léteznek kidolgozott ontológiák. Természetesen alapos vizsgálatot igényel, hogy a már létező megoldások mennyiben illeszthetők a jelenlegi modellhez, ami nem csak a struktúrák eltérésének vizsgálatát jelentik, hanem technológiai megoldások (például az alkalmazott ontológia nyelv) összevetését is igényli.

Módszertani szempontból ebben a fázisban kell sort keríteni az ontológia osztályainak meghatározására, melyre fontossága miatt és az érthetőség érdekében már az előző fejezetben sor került (II.7.2 fejezet). A **tervezési elvek** – melyeket részben az ontológia célja is meghatároz – a következők:

- Tananyagok szerkezete, felépítése több szempontból is eltérhet egymástól, eltérő gondolkodásmódot igényelhet megértésük és feldolgozásuk.
- Adott tananyag tartalmát pontosan meg kell feleltetni a képzés során elsajátítandó képességeknek és kompetenciáknak.
- A munkaerő-piaci elvárásokat össze kell egyeztetni a tananyag tartalmával.
- Átlátható, világos modell kialakítása a cél, melyet oktatók széles köre el tud fogadni és alkalmazni is képes.
- Az ontológiának támogatnia kell a számítógépes adaptív tesztrendszer működését.
- Az alkalmazni kívánt formális leírás nyelve is kereteket szab, melyet szem előtt kell tartani.

Az elsődleges modell megszületése után kerülhet sor a megfelelő fejlesztőeszköz kiválasztására, majd az egyes tantárgyak tudásanyagának tényleges feldolgozására.

### (3) *Finomítás*

A finomítási szakasz célja az alkalmazásorientált ontológia kialakítása. Több alszakaszra bontható, melynek során először a informális taxonómiát kell felállítani, majd az elsődleges váz finomítására, bővítésére kerül sor szakértők bevonásával, végül az ontológia formális leírását kell megadni.

A végső cél az oktatás eredményességének, illetve a hallgatók tudásszintjének felmérése, melyet szem előtt kell tartani az ontológiaépítés teljes szakaszában. A fogalmi modell felépítésének mind a három módszere alkalmazható az On-To-Knowledge módszertan szerint. Jelen esetben – az

---

<sup>46</sup> A Tudásmenedzsment előadások anyagait az Információrendszerek Tanszék két oktatója, Kő Andrea és Fehér Péter dolgozta ki.

<sup>47</sup> Davenport és Prusak (2001): Tudásmenedzsment, Kossuth Kiadó

---

Oktatási ontológia modelljének megfelelően – felülről-lefelé történő építkezést kell alkalmazni. Az általános fogalmak meghatározását követően kerül sor az egyre speciálisabb fogalmak definiálására. Ez a megközelítés az általános oktatási módszereknek is megfelel. Egy tárgy kidolgozása során először a tematikát állítjuk össze meghatározva a fontosabb témaköröket (melyek az ontológiában a tudásterületeknek feleltethetők meg), majd az órák keretében ezeket lebontjuk részterületekre (altudásterületekre) és feltárjuk, értelmezzük összefüggéseket az alapfogalmak meghatározásának segítségével.

A Tudásmenedzsment tárgyat a következő témakörökre osztottam fel:

- Tudásmenedzsment alapjai (bevezetés, alapfogalmak és kapcsolataik)
- Tudásmenedzsment stratégiák
- Tudásmenedzsment technológiai támogatása
- Tudásteremtés és fejlesztés
- Tudásátadás
- Intellektuális tőke mérése
- Tudásmenedzsment szervezeti kérdései
- Tudásmenedzsment emberi erőforrás kérdései<sup>48</sup>

Ezek a témakörök lesznek a Tudásmenedzsment ontológia „Tudásterület” osztályai. A következő lépés a tudásterületek felosztása nagyobb egységeire, azaz altudásterületekre. Jelen esetben a nyolc tudásterület további 28 altudásterületre bontottam fel. Bizonyos esetekben az alterületeket is további részterületekre kell bontani. Ezzel egy elsődleges taxonómia készült el. Mielőtt meghatároznánk a tudásterületek alapfogalmait, összefüggéseit és példáit tisztázni kell még a tudásterületek közötti összefüggéseket is. Meg kell nézni, hogy adott tudásterület megköveteli-e valamely más tudásterület ismertét, előzetes elsajátítását. Szem előtt kell tartani, hogy itt más tananyagok tudásterületeinek ismerete is szükséges egy-egy tudásterület elsajátításához. A Tudásmenedzsment esetében több ilyen kapcsolat is létezik. A tudásmenedzsment stratégiák elsajátításhoz ismerni kell a Stratégia menedzsment egyes részeit is (például a vállalat erőforrás alapú megközelítését, illetve a vállalat alapvető képességek megközelítését is). Ezen kívül a Vezetés-szervezés és Emberi erőforrás menedzsment tárgyak egyes tudásterületeinek ismerete is elkerülhetetlen a Tudásmenedzsment bizonyos témaköreinek megértéséhez.

Ezt követően kerülhet sor az alapfogalmak, összefüggések és példák meghatározására, és a tudáselemek közötti relációk feltárására. Ez aprólékos, precíz munkát igényel. Nem elég egy összefüggést kimondani és felvenni az ontológiába, alárendelve valamely tudásterületnek, hanem

---

<sup>48</sup> A képzési programtól, a szemeszter hosszától függően adott tematikában több témakört találhatunk, amelyek azonban csak a fent bemutatott tudásterületek részekre bontását jelentik.

---

pontosan meg kell határozni, hogy mely további fogalmak és összefüggések ismeretét követeli meg.

Másrészről ki kell dolgozni az ontológia tudásterület „feletti” részeit is. A kompetenciákat a szakalapítási dokumentumok<sup>49</sup> tartalmazzák. A bolognai elvárásoknak megfelelően az ontológiában két halmazra bontottam fel képességeket: a BSc és MSc kompetencia modulba. Ezeket részletezve 54 különböző kompetenciát vettem fel az ontológiába.

Továbbá három olyan munkakört is felvettem az ontológia „Munkakör” osztályába, melyek ellátásához feltétlenül szükséges a Tudásmenedzsment tárgy anyagának ismerete. Ideális esetben ezeket a munkaköröket és a hozzájuk kapcsolódó feladatokat a potenciális munkaadók határozzák meg. A példa esetben a következő munkaköröket vettem fel az ontológiába:

- Tudásmenedzsment szakértők
- Tudásmenedzsment-projekt vezető
- Tudásmenedzsment vezető

Végül meghatároztam az adott munkakör keretében ellátandó feladatok körét (Ezek a „Feladat” osztály példányai).

#### **(i) Oktatási ontológia építésének folyamata tetszőleges tárgy esetében**

A következő lépéseket követve készíthetjük el adott tárgy ontológia modelljét (abból a feltételezésből kiindulva, hogy az ontológia tudásterület „feletti” szintje (a kompetenciák, feladatok munkakörök) már elkészült):

1. Tudásterületek meghatározása – a tárgy tematikájából kiindulva meg kell határozni azt a 8-12 témakört, melyek a tárgy egészét lefedik. Fontos, hogy olyan tudásterületeket („Tudásterület” osztályokat) vegyünk fel, melyek tényleges tartalommal rendelkeznek, pontosan definiálhatók, tehát az érintett szakterület egy-egy átfogó problémakörét foglalják magukba. Mivel tananyagokról van szó, a kapcsolódó irodalom egy-egy fejezetének megnevezése sok esetben egy tudásterület megnevezése is lesz (például Tudásmenedzsment stratégiák). Azonban ezzel óvatosan kell bánni, mert nem feltétlenül lehet tudásterületként definiálni minden egyes fejezetet (például „A tudásmenedzsment történeti áttekintése” fejezet-címnek tökéletes, de ilyen tudásterület nem vehető fel az ontológiába, hiszen nem a Tudásmenedzsment egy részét írja le, hanem hogy mi vezetett a terület kialakulásához.)
  - a. A tudásterület *belső szerkezetének* kialakítása a következő lépés, mely a következő szakaszokból áll:
    - i. A tudásterület altudásterületeinek meghatározása– Átfogó, nagy terjedelmű tudásterületek esetén a tudásterületet részeire bontjuk (meghatározzuk az adott osztály részosztályait).

---

<sup>49</sup> Lásd a Függelék V.2 alfejezetét.



- 
- ii. A tudásterület tudáselemeinek meghatározása – Ebben a lépésben meghatározzuk mindazokat az alapfogalmakat, összefüggéseket, példákat melyek az adott tudásterülethez kapcsolódnak. Célszerű az alapfogalmakkal kezdeni, hiszen bizonyos esetekben egy tétel, összefüggés kimondásának, meghatározásának elengedhetetlen feltétele az alapfogalom ismerete.
  - iii. Tudáselemek relációinak kialakítása – Az előző lépéssel párhuzamosan meg kell határozni, hogy fennáll-e valamilyen kapcsolat a felvett alapfogalmak és összefüggések között, illetve hol szükséges hivatkozni valamely alapfogalomra vagy összefüggésre.
- b. Tudásterület *kapcsolatainak* meghatározása – A kapcsolatokat egy része meghatározásra került (a altudásterületek definiálásával), de következő kapcsolatokat is meg kell vizsgálni:
- i. vannak-e olyan tudásterületek (a tananyagban belül vagy más tananyagokban) melyeket előzetesen el kell sajátítani a tudásterület megértéséhez.
  - ii. mely kompetenciák, képességek elsajátítását biztosítja az adott tudásterület ismerete.
2. Altudásterületek meghatározása – Az altudásterületek ontológiai modelljének kialakítása teljesen megegyezik a tudásterületeknél bemutatott lépésekkel. Az egyetlen kivétel, hogy altudásterületeket nem kapcsoljuk közvetlenül össze a kompetenciákkal.

Ha minden tudásterületet és altudásterületet meghatároztunk és kialakítottuk a belső szerkezetét és kapcsolatrendszerét, akkor elkészült az adott tananyag ontológiai modellje.

#### (4) *Értékelés*

Célja a létrehozott ontológia hasznosságának elemzése. Az értékelés során meg kell vizsgálni, hogy az ontológia megfelel-e követelményspecifikációban megfogalmazott elvárásoknak, az alkalmazási környezetben való tesztelésére is itt kerül sor. Az így kapott visszajelzések alapján az ontológia tovább finomítható, pontosítható. Az értékelés szoros kapcsolatban van a finomítás szakaszával, ugyanis ez a két lépés mindaddig ciklikusan ismétlődik, míg a célontológia nem éri el a kívánt színvonalat. Ehhez szükség van adaptív vizsgarendszer kifejlesztésére is, melyre a következő fejezetben kerül sor.

#### (5) *Karbantartás és fejlesztés*

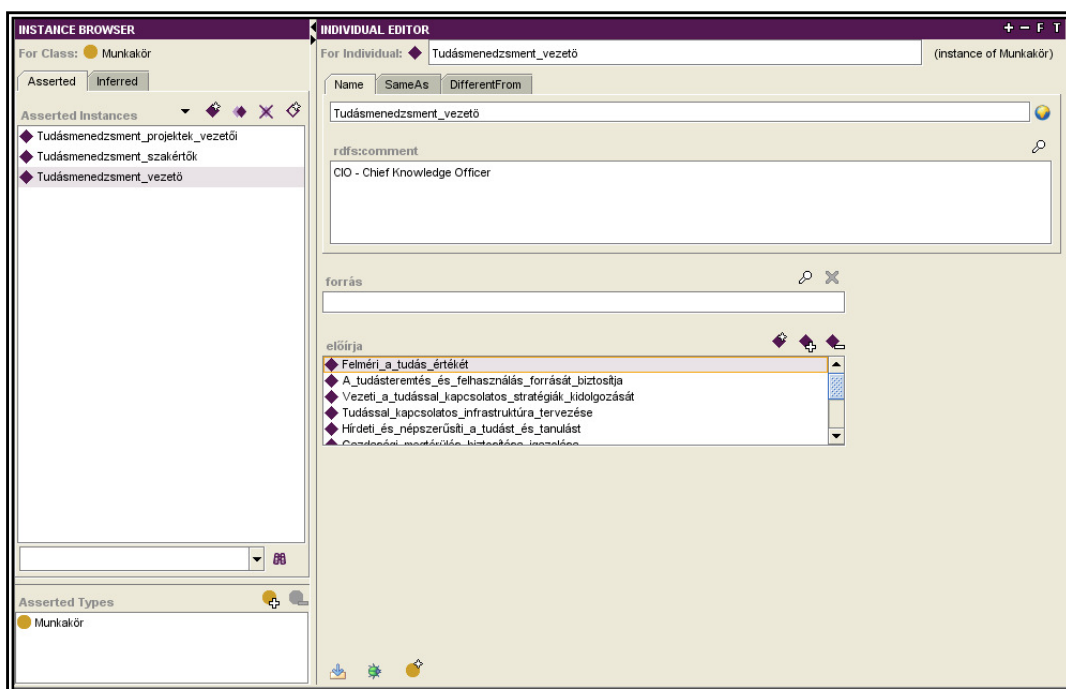
Az ontológiával szembeni elvárások folyamatosan változhatnak, a tárgyakat fejleszteni, aktualizálni kell a bekövetkező változásokat visszatükrözve. Éppen ezért elkerülhetetlen az ontológia folyamatos karbantartása, amely alapvetően egy szervezeti folyamat. Szigorú szabályokat kell megfogalmazni arra vonatkozóan, hogy milyen időközönként, hogyan és kinek van lehetősége a változtatásokra, hogy milyen módon történjen a verzióváltás. Célszerű ebben az oktatás féléves felosztását követni és ehhez igazítani a

karbantartási feladatokat. Az új félév tananyagának aktualizálásával párhuzamosan az Oktatási ontológiában is el kell végezni a kapcsolódó változtatásokat.

#### II.7.4.2. A Tudásmenedzsment ontológia

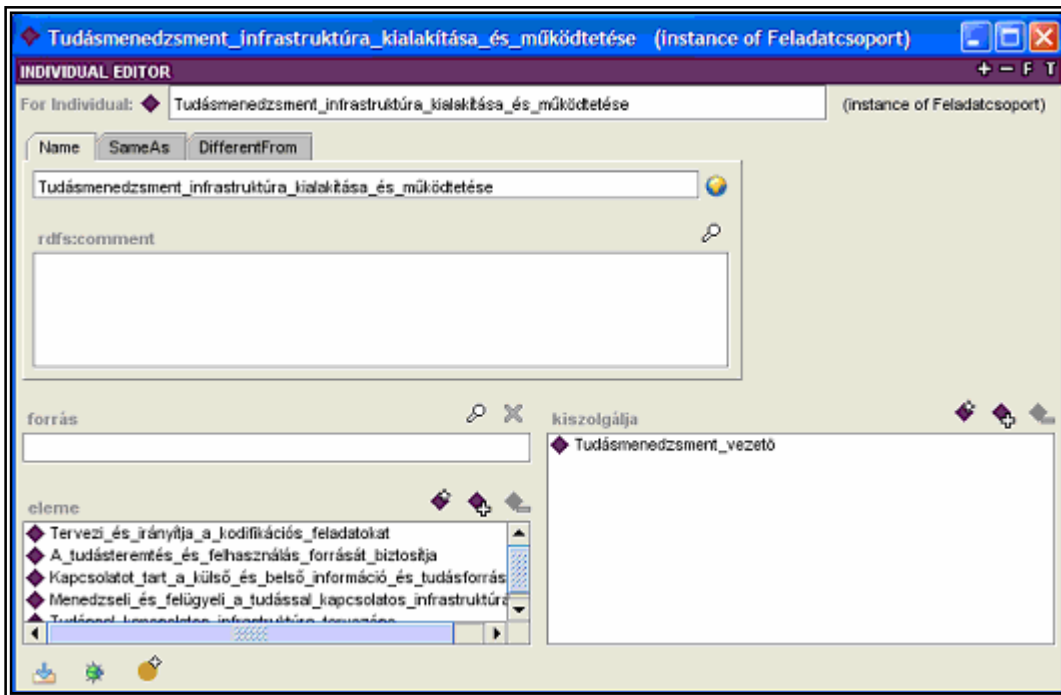
A következőkben a Protégé-ben kidolgozott Tudásmenedzsment ontológia struktúráját mutatom be. Egész pontosan egy munkakörből kiindulva mutatom be, hogy a kompetenciákon keresztül hogyan jutunk el az egyes tudásterületekig, illetve a tudáselemekig. A Tudásmenedzsment tárgy teljes ontológia modellje elkészült ugyan, de annak teljes, minden fogalmat és relációt bemutató leírása nagyon nehézkes, melyre ebben a fejezetben nincs lehetőség.

A 11. ábra a tudásmenedzsment vezető munkakörhöz kapcsolódó feladatcsoportokat és feladatokat mutatja be (az „előírja” reláció alatt). A listában a „Feladatcsoport” osztály elemei és a „Feladat” osztály elemei egyaránt szerepelnek, hiszen mindkettő az „előírja” relációval kapcsolódik a munkakörhöz.



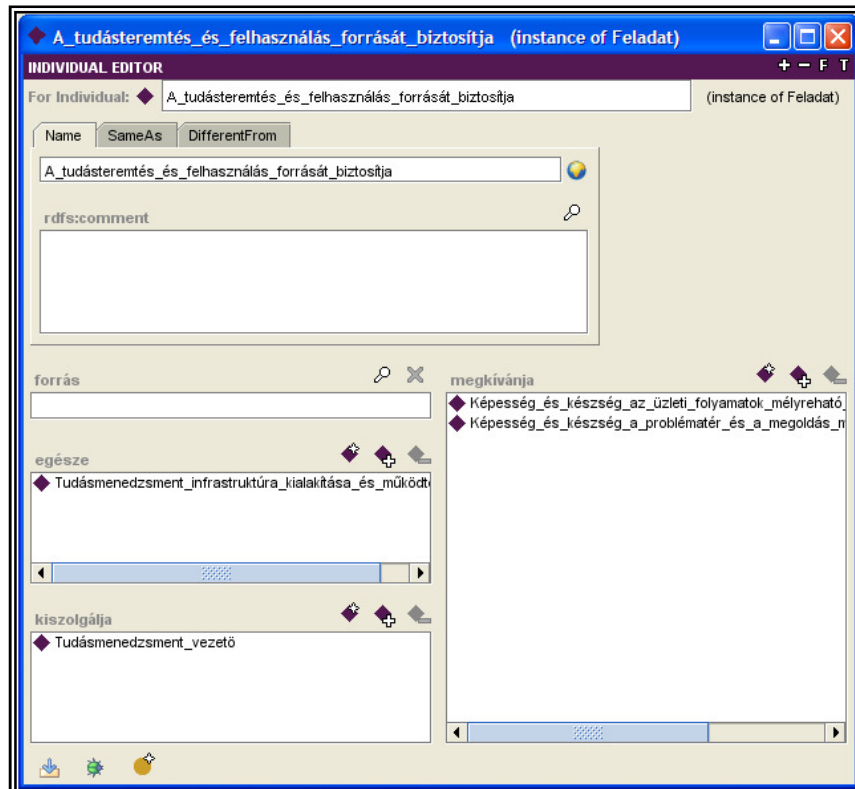
11. ábra: Tudásmenedzsment-vezető munkakör jellemzői

A 12. ábra a Tudásmenedzsment vezető munkakörhöz kapcsolódó három feladatcsoport egyikét és a hozzá tartozó feladatokat mutatja be. (A másik két feladatcsoport a tudásmenedzsment gazdasági megtérülésének biztosításához és igazolásához kapcsolódó feladatokat, illetve a tudásmenedzsmentet támogató kultúra kialakításához kapcsolódó feladatokat foglalja magába.)



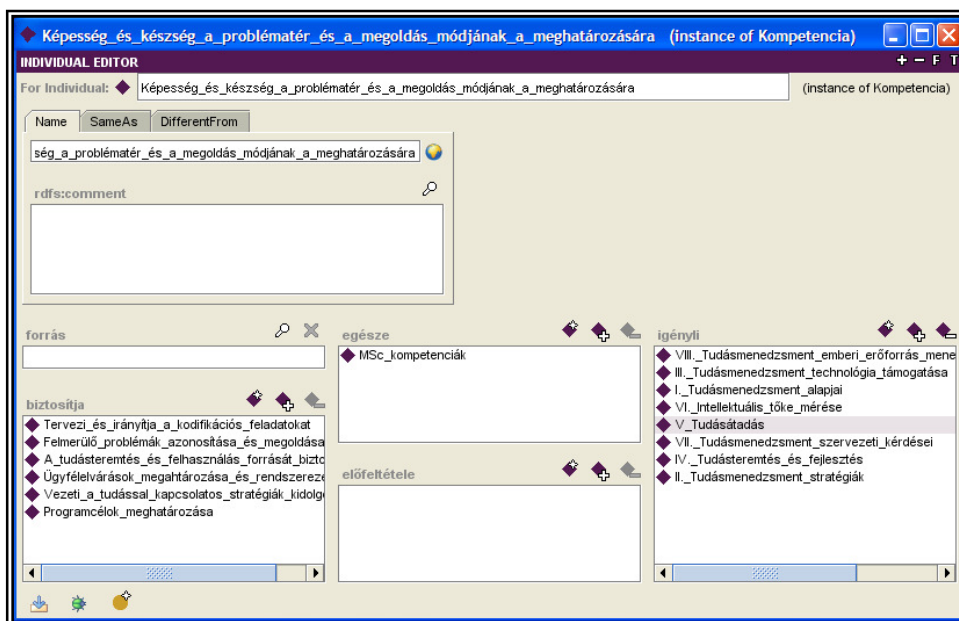
12. ábra: A Tudásmenedzsmet-vezető munkakörhöz kapcsolódó feladatcsoportok

A 13. ábra egy kiválasztott feladatot (A tudásteremtés és felhasználás forrásának biztosítása) és a feladat ellátásához szükséges kompetenciákat jelöli meg.

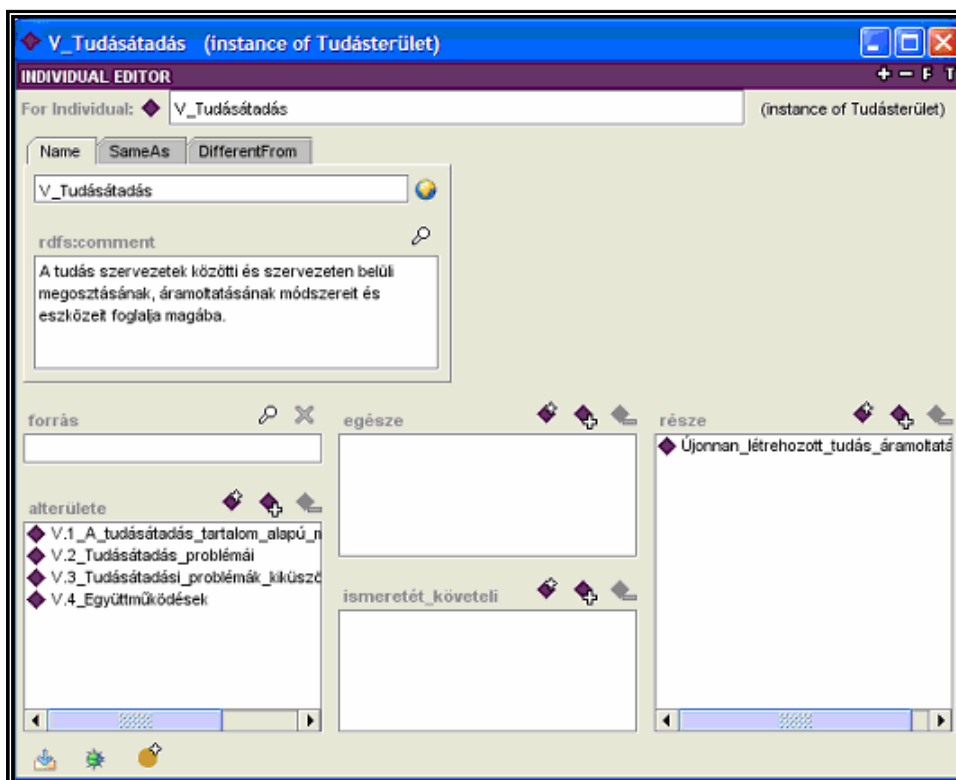


13. ábra: „Feladat” osztály példányának jellemzői és relációi

Az 14. ábrán láthatjuk, hogy mely tudásterületeket kell elsajátítani a kijelölt képesség (kompetencia) megszerzéséhez, illetve, hogy milyen további feladatok ellátását tudja biztosítani ez a kompetencia. Egyik esetben sem feltétlenül teljes a lista, hiszen ahhoz az összes, a gazdaságinformatika körében oktatott tárgyat fel kellene vennünk az ontológiába.

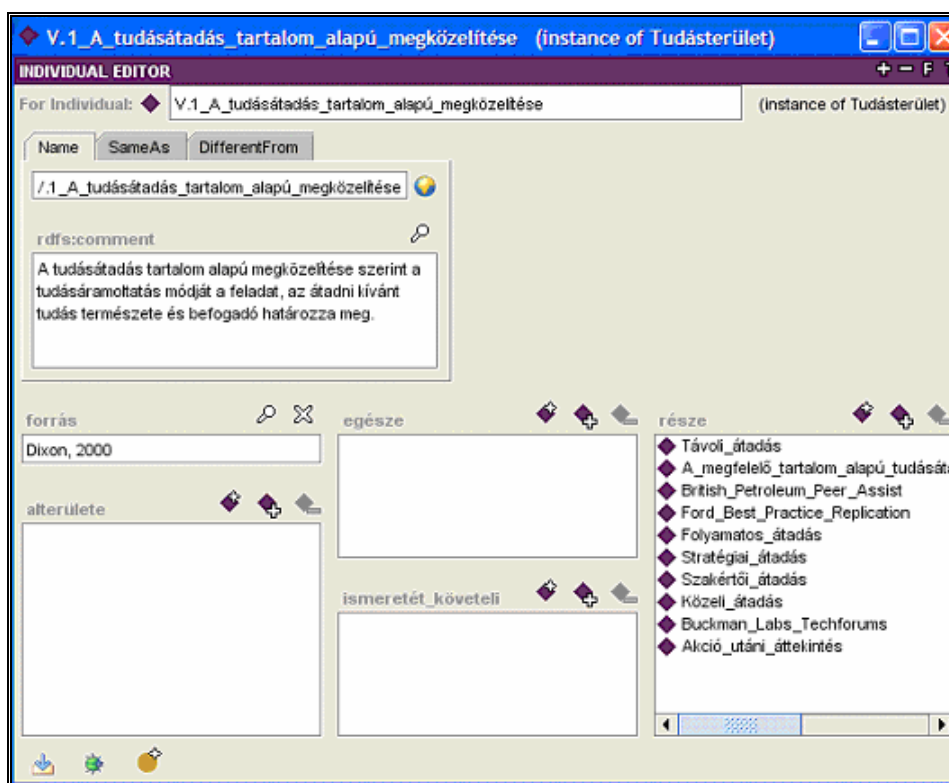


14. ábra: „Kompetencia” osztály példányának jellemzői és relációi



15. ábra: „Tudásterület” osztály példányának jellemzői és relációi

Végül a kompetencia által igényelt tudásterületek közül választottam ki egyet (15. ábra), illetve ennek egy alterületét (16. ábra). Mindkét ábrán a része reláció alatt található a területhez kapcsolódó alapfogalmak, összefüggések és példák.



16. ábra: Altudásterület jellemzői és relációi

A tudásmenedzsment tárgy esetében kifejezetten hasznosnak bizonyult a Példa, mint tudáselem alkalmazása az ontológiában. A tárgy során sok esettanulmányt, konkrét esetet mutatunk be a hallgatóknak az összefüggések, megközelítések szemléltetésére. A „Példa” osztály segítségével ezeket is beépíthettük az ontológiába, így a vizsgáztatás során is vissza lehet kérdezni ezekre.

A tudáshiány keresésének eszköze a teszt. A kitűzött cél és az ontológia meghatározza az adott felméréshez tartozó kérdések összességét. Ezen teszt-kérdésekből kialakított tesztet végeztetjük el a jelölttel. A kérdésekre adott helyes és helytelen válaszoknak az ontológia struktúráján való eloszlása alapján képesek vagyunk képet alkotni a jelölt tudásáról, illetve tudáshiányáról.

Tehát a cél által meghatározott tesztkérdések összességéből egy-egy teszt összeállításában az oktatási ontológia struktúrája és a tudáselemek közötti szemantikus viszonyok biztosítanak támogatást.

Másképp megfogalmazva, az adaptív tesztek kialakítását és kezelését biztosító rendszer kettős alappillérként használja az oktatási ontológiát és a hozzá kapcsolt tesztalmanax. Ez természetesen egy önálló rendszer, amelyben különböző stratégiák alkalmazhatók a tesztek megalkotásához és értékeléséhez. Ennek megfelelően a következő fejezetben az adaptív vizsgáztatás módszereivel foglalkozom.

---

### III. Adaptív tesztelés

Az információs és kommunikációs technológiák rohamos fejlődése az oktatásban alkalmazott eszközök körét is jelentősen bővítette. A különböző elektronikus oktatási rendszerek (e-learning alkalmazások, oktatásmenedzsment rendszerek<sup>50</sup>, oktatási tartalomkezelő rendszerek<sup>51</sup>) megjelenése és elterjedése a hatékony elektronikus számonkérési módszerek iránti igényt is növelte. Az adaptív tudásfelmérés módszere sok évtizedes múltra tekint vissza, de a számítógéppel támogatott formái csak az elmúlt néhány évben jelentek meg.

A második hipotézis szerint „alkalmasan megválasztott adaptív tesztelési rendszerrel az egyén tudása megragadható, jól mérhető.” A hipotézis igazolása érdekében tisztázni kell az adaptív vizsgáztatás fogalmát, elemezni kell elveit és módszereit, meg kell ismerni a számítógéppel támogatott tesztelés megoldásait, és ki kell választani az alkalmazni kívánt eljárást, de a dolgozatnak nem feladata és célja a módszerek pedagógiai szempontjainak vizsgálata. Ez olyan kérdéseket és problémaköröket vet fel, mely túlmutat jelen dolgozat keretein. A vizsgálódás során szem előtt kell tartani, hogy a tudásfelmérés célja az oktatás eredményességének megállapítása és a hiányterületek tételes feltárása.

#### III.1. Az adaptív tesztelés fogalma és története

Az adaptív vizsgáztatás során a tesztelemelek (kérdések) számát és a kérdések feltevésének sorrendjét a felmérés folyamán határozzák meg, azzal a céllal, hogy a lehető legkevesebb kérdés feltevésével, a lehető legpontosabban meg tudják becsülni a hallgató tudásszintjét. Az adaptív tesztelés korántsem új keletű megközelítés és bár számos előnnyel rendelkezik a hagyományos vizsgáztatási módszerekkel szemben, alkalmazása még nem terjedt el széles körben.

Nem kell mélyreható elemzéseket végeznünk, hogy belássuk az oktatásban alkalmazott tesztek és vizsgasorok összeállításakor elsősorban a tesztkészítők igényeit tartják szem előtt. A megfelelő vizsgáztatási módszer kiválasztása, jó teszt összeállítása korántsem könnyű feladat. Az sem jó, ha túl könnyű a teszt, de az sem, ha túl nehéz. Ha túl könnyű a teszt, a jelöltek nem veszik elég komolyan a vizsgát és figyelmetlenségből hibákat követnek el, vagy trükkösnek feltételezik a kérdéseket és így szándékosan rossz válaszokat jelölnek meg. Ugyanakkor túl nehéz kérdésekből álló tesztsorok sem nyújtanak megbízható eredményeket, mert a vizsgázó túl korán feladja, és meg sem próbálja megválaszolni a kérdéseket, csupán találgat és sablonos válaszokat.

A hagyományos, legelterjedtebb számonkérési formák szerint valamennyi jelölt ugyanazokat a kérdéseket kapja és így az eredmények és a vizsgá-

---

<sup>50</sup> Learning Management System

<sup>51</sup> Learning Content Management System

---

zók teljesítményei egyszerűen és világosan összehasonlíthatók. Az adaptív tesztelés szakít ezzel a szemlélettel. Minden vizsgázó más-más kérdéseket kap és a kérdések sorrendje sincs előre meghatározva. Mindig az előző, már megválaszolt kérdésekre adott megoldások határozzák meg, hogy a következő lépésben milyen kérdést teszünk fel a vizsgázónak. Azonban úgy igazítani egy tesztet, hogy az illeszkedjen minden egyes vizsgázó teljesítmény-szintjéhez, vitatható és nem tisztességes. A kérdés az, hogyan tudjuk összehasonlítani az egyes vizsgázókat, ha mindenki más tesztet tölt ki. Az adaptív tesztelés módszerére, a kérdésfeltevés módjára vonatkozóan számos megközelítés és elmélet látott napvilágot. Elsőként a pszichológiában alkalmazták ezt a módszert.

Az adaptív tesztelés első formáját Alfred Binet dolgozta ki már 1905-ben. Olyan **intelligencia tesztet** alakított ki, melynek célja az egyének és nem a csoport elemzése volt, tehát az igazságtalanság kérdése, az az elvárás, hogy mindenki ugyanazt a tesztet töltse ki, nem merült fel. Binet rájött, ha a kérdéseket<sup>52</sup> nehézségi szint szerinti rangsorolásával a tesztek testreszabhatók. Első lépésben megbecsülte a jelölt képességszintjét és ennek a szintnek megfelelő nehézségű kérdést tett fel elsőként. Binet megközelítésében, ha a jelölt folyamatosan helyesen válaszolt, akkor egyre nehezebb kérdéseket kell feltennünk mindaddig, amíg a jelölt nem követ el sorozatosan hibákat. Fordított esetben, ha a jelölt már az első kérdésre vagy kérdéshalmazra rosszul válaszolt, akkor a következő lépésben egyre könnyebb kérdéseket kell feltenni, mindaddig, amíg nem ad folyamatosan jó válaszokat. Ezzel a módszerrel Binet meg tudta határozni a jelölt képesség szintjét (Binet, 1905).

Binet módszerét számos szerző fejlesztette, pontosította, így többek között Lord (1980), Henning (1987), Lewis és Sheehan (1990). Az említett szerzők módszereiben nehézségük szerinti rétegekbe rendezték az egyes kérdéseket úgy, hogy valamennyi nehézségi szinthez több kérdést is hozzárendeltek. A teszt menete, pedig a következő: A kérdések egyik halmazát kiválasztották, majd kikérdezték és a válaszok helyessége alapján léptek tovább felfelé vagy lefelé a nehézségi szintben. Néhány halmaz adminisztrálása után a jelölt képességi szintjét meg tudták határozni.

Az adaptív tesztelés fejlődésnek következő lépcsőjét a folyamat számítógéppel történő támogatása jelentette. Az első számítógéppel támogatott adaptív tesztelési módszert Reckase fejlesztette ki (Reckase, 1974.)

### **III.2. Számítógéppel támogatott adaptív tesztelés**

A számítógépes adaptív tesztelés során a megfelelő algoritmus kialakítása és alkalmazása révén már nem a vizsgázatónak kell kiválasztani az első kérdést, meghatározni a további kérdések sorrendjét és értékelni az eredményeket, hiszen mindezt a gép elvégzi helyettünk. A számítógépes adaptív tesztelés esetében is a jelölt a becsült képességének megfelelő nehézségű feladatot kap, csökkentve ezzel a vizsgadrukot, egyben bizto-

---

<sup>52</sup> Az angol az 'item' kifejezést használja ('item' = kérdés egység, teszt elem), utalva arra, hogy kérdés mellett a jó, illetve a rossz válaszlehetőségek is befolyásolják a megoldandó feladat nehézségét, ezért ezeket együtt kell kezelni.

---

sítva a megfelelő motiváltsági szintet is. A legfontosabb még is az, hogy így a teszt teljes ideje is jelentősen csökken (Welch és Frick, 1993).

Linacre (2000) munkájában részletes leírást találhatunk a számítógépes adaptív tesztelés (CAT<sup>53</sup>) elméleti háttéréről, fő típusairól és ezek működési logikájáról. A CAT több módszertani megközelítést is kidolgoztak az évek folyamán, melyek a pszichometrikus felmérés<sup>54</sup> elveiből indulnak ki. A számos variáció közül különösen a 'Válaszadási Elmélet' (Item Response Theory (Wainer & Mislevy, 1990) és Tudástér Elmélet (Knowledge Space Theory (Falmagne et al., 1990)) terjedt el széles körben. Mindkét megközelítés a felmériendő tudásterületet problémakörként ábrázolja, mely egy adott tananyag vagy tananyag rész problémáit / probléma-osztályait tartalmazza. A következő alfejezetek a felsorolt megközelítések bemutatásával foglalkoznak.

A megközelítéstől függetlenül valamennyi számítógéppel támogatott adaptív vizsgáztatás rendelkezik a következő tulajdonságokkal:

- a jelölt bármikor elvégezheti a tesztelést, nincs szükség csoportos vizsgák szervezésére,
- minden egyes vizsgasor testreszabott, nincs két egyforma kérdés-sor, ami a puskázás lehetőségét is csökkenti,
- a kérdéseket számítógépen tölti ki a tesztet,
- amennyiben a jelölt válaszolt a kérdésre és ezt rögzítette, nincs módjában megváltoztatni,
- a jelölt nem hagyhat ki egy kérdést sem, és nem térhet vissza már megválaszolt kérdésekre,
- a jelöltnek válaszolnia kell az éppen aktuális kérdésre, hogy továbbléphessen a következő kérdésre és végül
- a következő teszt-elem kiválasztását és a vizsgáztatás befejezését a vizsgázó által adott válaszok dinamikusan irányítják (Thissen és Mislevy, 1990).

### III.2.1. Kétértékű elemekből álló vizsgasor<sup>55</sup>

A pszichometrikus megközelítés elveit veszi alapul ez a módszer, mely az egyik legegyszerűbb változata az adaptív tesztelésnek. A vizsgasor dichotóm elemekből (pl. feleletválasztós kérdésekből) áll, melyeket egy lineáris skála mentén súlyozunk. A megközelítés lényege, hogy a kérdésre adott válasz csak két értéket vehet fel, azaz helyes, vagy helytelen lehet. A jelölt által megadott válaszokhoz csak ezt a két értéket lehet hozzárendelni. Linacre példája szemléletesen mutatja be, hogyan állapítjuk meg a jelölt képességszintjét e megközelítés szerint.

---

<sup>53</sup> CAT – Computerized Adaptive Testing

<sup>54</sup> A pszichometria a lelki folyamatok (intenzitásuk, gyakoriságuk, sebességük) mérésére/értékelésére szolgáló módszerek összessége

<sup>55</sup> Dichotomous items



---

*„Tegyük fel, hogy a bankban szereplő valamennyi item 0 és 100 közötti nehézségi fokkal bír és hogy a  $2+2=4$  feladat nehézségi foka 5 egység. A gyerekek közül azok rendelkeznek 5 egységnél magasabb képességszinttel, akik könnyedén megoldják ezt a példát, illetve fordítva azok gyerekek, akiknek a helyes megoldás megadása túl bonyolult feladatnak bizonyul, képességi szintje 5 egység alatt található. Azoknak a gyerekeknek lesz pontosan 5 egységnyi a képességi szintje, akik 50%-os eséllyel számítják ki helyesen ez a példát.” (Linacre, 2000 p. 5.)*

A teszt szempontjából nem lényeges, hogy melyik kérdést tesszük fel elsőként, pszichológiai szempontból azonban már korántsem mindegy. Ha rögtön az első kérdés túl nehéz, a jelölt könnyen feladja, és meg sem próbál jól teljesíteni. Ellenkező esetben, ha túl könnyűnek bizonyul az első kérdés, akkor előfordulhat, hogy a jelölt nem veszi komolyan a tesztet és figyelmetlenségéből követ el hibákat. Gershon (1992) javaslata szerint elsőként mindig a becsült képességszintnél valamivel könnyebb kérdést vagy kérdéseket kell feltenni.

A legtöbb vizsga, vagy felmérés esetén előre meghatároznak egy szintet, melyet el kell érni ahhoz, hogy a jelölt átmenjen a vizsgán. Ilyen esetben az első kérdés nehézségi szintje valamivel alacsonyabb kell, hogy legyen, mint a teljesítés feltételeként megszabott minimum szint.

A vizsga menete ebben az esetben a következő: *„Tegyük fel, hogy az első feltett kérdés nehézségi szintje 30 és a hallgató becsült képességi szintje 50. Az első kérdést valószínűleg könnyen megválaszolja a jelölt, és így a következő lépésben a gép már nehezebb kérdést tesz fel, melynek nehézségi szintje 40 egység. A hallgató újra helyes választ ad, ekkor a számítógép egy 50 egység nehézségű kérdést tesz fel. A hallgató képességi szintje és a kérdés nehézségi szintje ekkor összhangba kerül, azaz 50%-os eséllyel válaszolhat helyesen a jelölt. Tegyük fel, hogy rosszul válaszol. Ebben az esetben a számítógép egy kicsivel könnyebb kérdést tesz fel a következő lépésben, ami 40-nél magasabb nehézségi szintű, például 45 egységnyi. Ha erre is jól válaszol, akkor a következő kérdést a 45 és 50 közötti nehézségi szintű kérdések közül választja ki a rendszer, például 48 nehézségi szintűt. Tegyük fel, hogy erre is jól válaszol a jelölt. A hallgató a 40 és 48 nehézségi szintek közé eső kérdésekre jól válaszolt. Ez azt mutatja, hogy korábban az 50-es szintnél rosszul megválaszolt kérdés szerencsétlen tévedés volt. A következő lépésben így feltehetünk egy 50-nél nehezebb nehézségi fokú kérdést. Legyen 52-es nehézségű kérdés. Ekkor még mindig közel 50% az esélye, hogy jól válaszoljon. A hallgató helyesen válaszolt, így a következő kérdés nehézsége már 54 egység lesz. Erre azonban már rossz választ ad, így az újabb kérdés 51 egység nehézségű lesz. A hallgató ismét rosszul válaszol, így a következő lépésben 49 nehézségi fokú kérdést tesz fel a gép, amire ismét jó válasz érkezik. Ez a folyamat folytatódik mindaddig, amíg a gép nem lesz egyre biztosabb, hogy a vizsgázó képesség szintje 50 egység körül van. Minél több kérdést teszünk fel, annál biztosabban határozható meg a tudásszint. A teszt azonban nem lehet végtelen hosszú, így a programnak tartalmaznia kell úgynevezett „leállási szabályokat”, melyek meghatározzák milyen feltételek teljesülése esetén ér véget a felmérés.” (Linacre, 2000 pp. 5-6)*

---

A bemutatott módszer közelebb visz az adaptív tesztelés lényegének megértéséhez és világosan rámutat, már ez az egyszerű megközelítés is, hogy a kérdések rangsorolásával pontosabb képet kaphatunk a jelölt tudásszintjéről. Ugyanakkor az oktatás eredményességére vonatkozóan nem jutunk semmi plusz információhoz, nem világos mi a kapcsolat a fel-tett kérdések és a tananyag között. Ráadásul a tesztkérdések és a vizsgázó közötti kapcsolatról sem tudunk semmit mondani.

### III.2.2. Többértékű elemekből álló vizsgasor<sup>56</sup>

Többértékű elemekből álló teszt esetén a megadott válaszok nem csak két értéket vehetnek fel. Azokat a teszt-elemeket, melyek több értéket is felvehetnek lényegében ugyanúgy kezeljük, mint azokat az elemeket, melyek csak két értéket vehetnek fel. A vizsgázó a teszt során nem is vesz észre semmi különbséget. A két megközelítés között a pontozásban van az eltérés. Nemcsak a helyes válaszáért lehet pontot adni, hanem a helytelen válaszok között is van olyan, mely közelebb áll a jó megoldáshoz, ezért több pontot is ér. A legtöbb pont természetesen a helyes válaszáért jár. A többértékű elemek kiválasztása azonban nagyobb körültekintést igényel, mint a kétértékű elemeké. Nem lehet világosan meghatározni minden kérdés nehézségét, hiszen köztes kategóriák is vannak, melyek más-más képességi szintet jelölhetnek. Linacre erre is kidolgozott egy szemléltető példát:

*„A tesztbank tehát olyan elemeket tartalmaz, mely különböző értékeket vehet fel a teljesítmény függvényében. A különböző teljesítményeket 0-tól 3-ig pontozzuk. Tegyük fel, hogy a jelölt képesség szintje 50 egységnyi, ezért ennél a szintnél könnyebb, azaz 30-as nehézségi fokú kérdést tesz fel először a gép. Ha jelölt képességi szintje 30 egységnyi lenne, akkor azt várnánk tőle, hogy 2 pontot kapjon a kérdésre a megadott válasz alapján. A mi esetünkben a vizsgázó helyesen válaszol, és így 3 pontot kap, ami egyben azt is jelenti, hogy a képességi szintje magasabb 30 egységnél. Amennyiben a 2 és 3 pontos értékelés közötti váltás, azaz a küszöbérték 35 egységnél van, akkor a jelöltünk képességi szintje az első kérdés megválaszolása tükrében már 38 egységnyinek tekinthető. Mivel az első kérdés könnyűnek bizonyult, a következő lépésben olyan kérdést tesz fel a gép, melyre valószínűleg 1 pontot kap a vizsgázó. A jelölt azonban 2 pontot ér el. Harmadikként még nehezebb kérdést teszünk fel. Újra 2 pontot kap a hallgató. A 4. lépésben megpróbáljuk azt a legnehezebb kérdést feltenni, amire még valószínűleg maximális pontot kap a jelölt. A jelölt azonban erre a kérdésre csak 1 pontot kap. Az 5. lépésben szintén kísérletet teszünk megtalálni azt a legnehezebb kérdést, amelynél el tudja érni a 3 pontot a vizsgázó. Így egy könnyebb kérdést teszünk fel és a jelöltünk 3 pontot kap. A következő lépésekben megkíséreljük meghatározni, milyen nehéznek kell lennie a kérdésnek, ahhoz, hogy teljesen „megbukjon” a vizsgázó. Ezt azt előzőekhez hasonlóan végezzük. A teszt ebben az esetben is akkor ér véget, ha valamelyik „leállási szabály” érvényesül.” (Linacre, 2000 pp. 8-9)*

---

<sup>56</sup> Polytomous items

---

Többértékű elemekből álló kérdéssor összeállítása és pontszámok megállapítása azonban sokkal nehezebb, mint kétértékű elemekből álló teszt összeállítása. Ugyanakkor megbízhatóbb eredmények születnek, lényegesen kevesebb kérdés feltevésével. A hallgatóknak azonban több időt vesz igénybe egy-egy kérdés megválaszolása.

### III.2.3. Mérési modell kiválasztása

A kutatás egyik célkitűzése számítógéppel támogatott adaptív tesztelés megvalósítása, melynek elengedhetetlen feltétele olyan objektív mérési modell kialakítása, melynek segítségével a gép értékelni tudja az eredményeket. Az számítógépes adaptív tesztelési módszerek döntő többsége az itt bemutatásra kerülő Rasch és Wright modellekre épülnek. (Rasch, 1960/1992; Wright, 1988)

A képességeket felmérő tesztek abból a feltételezésből indulnak ki, hogy képességeket egy dimenzió mentén lehet értékelni és rangsorolni. Ha azt mondjuk, hogy az egyik jelölt magasabb pontszámot ért el, mint egy másik, akkor ez a feltételezés áll az értékelés háttérében. A valóságban azonban a tesztek nem egy dimenzió mentén rangsorolnak. Ha a vizsgázókat egymáshoz képest akarjuk rangsorolni, vagy valamilyen teljesítmény szinthez viszonyítjuk őket, akkor legalább megközelítőleg biztosítanunk kell, hogy egy dimenzió mentén történjen az értékelés. Minden alkalommal, ha új kérdést akarunk felvenni a tesztbankunkba, felmerül a kérdés, hogyan állapítsuk meg a kérdés nehézségi fokát és biztosítsuk továbbra is, hogy a teszt egydimenziójú maradjon. A legtöbb esetben a probléma megoldása az új és a régi elemek közötti konzisztens és sztochasztikus rangsor fenntartására korlátozódik. A tényleges nehézségi szint megállapítása helyett, a többi elemhez viszonyítva határozzák meg az új kérdés nehézségi szintjét és helyét a rangsorban. Ha újabb kérdéseket akarunk felvenni a tesztbankunkba, akkor a konzisztencia fenntartása érdekében, az új kérdést egyszerűen a már meglévő kérdésekhez viszonyítjuk, és ennek megfelelően állapítjuk meg a helyét a rangsorban. Az elsődleges cél a sztochasztikus sorrend fenntartása, melynek legelterjedtebb eszköze a Rasch modell (Roskam és Jansen, 1984).

A dichotóm Rasch modell jelentősége abban áll, hogy egyszerű kapcsolatot teremt a vizsgázó és a teszt elemek között. Minden vizsgázót képességi szintjének meghatározásával jellemzünk, melyet a releváns képességek végtelen lineáris skáláján egy számmal jelölünk. A kényelem kedvéért megválasztjuk a skála lokális kezdőpontját. Az  $n$ -edik vizsgázó képességi szintje a lokális kezdőponttól  $B_n$  egységnyire helyezkedik el. Ehhez hasonlóan minden teszt-elemet a nehézségi szintjével jellemzünk, melyet a releváns képességek végtelen skáláján elhelyezkedő számmal fejezünk ki. Az  $i$ -dik elem nehézségi szintje  $D_i$  egységnyire helyezkedik el a képességi skála lokális kezdőpontjától. A kapcsolatot a vizsgázó és az a teszt-elem között a következő képlet írja le Rasch modelljében (Rasch, 1960/1992):

$$\log(P_{ni1} / P_{ni0}) = B_n - D_i ,$$

ahol  $P_{ni1}$  annak a valószínűsége, hogy az  $n$ -edik vizsgázó helyesen válaszol az  $i$ -dik elemre.  $P_{ni0}$  pedig a kudarc valószínűsége.

---

A Rasch modell további előnye, hogy kifejlesztettek hozzá egy a tesztelési folyamat működését leíró egyszerű algoritmust is, mely a számítógéppel támogatott adaptív tesztelés első változata. Wright algoritmusát könnyű kivitelezni és sikeresen alkalmazható minden tanulási modul végén a hallgató fejlődésének nyomonkövetésére – így az Oktatási ontológiára épülő adaptív vizsgáztatás kialakításához is jó alapot biztosít.

Wright dolgozta ki a Rasch modellen alapuló adaptív tesztelés lépéseit:

- 1) Bekérjük a következő vizsgázót. Beállítjuk a következőket  $D=0$ ,  $L=0$  és  $R=0$ , ahol  $D$  a tudásszint becsült értékét,  $L$  a feltett kérdések számát,  $R$  pedig a helyes válaszok számát jelöli.
- 2) Keresünk egy teszt elemet  $D$  nehézségi szint közelében.
- 3) Átállítjuk  $D$ -t az adott teszt-elem aktuális értékére.
- 4) Feltesszük a kérdést.
- 5) Bekérjük a választ.
- 6) Értékeljük a választ.
- 7) Megszámoljuk az eddig feltett kérdéseket (teszt-elemeket):  $L=L+1$
- 8) Összeadjuk az eddigi nehézségi szinteket:  $H=H+D$
- 9) Ha a válasz helytelen, akkor a kérdés nehézségét a következőképpen módosítjuk:  $D=D - 2/L$
- 10) Ha a válasz helyes, akkor a kérdés nehézségét a következőképpen módosítjuk:  $D=D + 2/L$
- 11) Ha a válasz jó, akkor összeadjuk az eddigi helyes válaszokat:  $R=R+1$
- 12) Ha még nem dönthető el, hogy átment, vagy megbukott a vizsgázó, akkor visszamegyünk a 2. lépéshez.
- 13) Ha eldönthető, hogy átment, vagy megbukott a vizsgázó, akkor megszámláljuk a helytelen válaszokat is:  $W=L-R$
- 14) Megbecsüljük a képesség mértékét:  $B=H/L+\log(R/W)$
- 15) Megbecsüljük a standard hibát:  $S=[L/(R*W)]$
- 16) Összehasonlítjuk  $B$ -t a teljesítés kritériumaként elérendő  $T$  szinttel.
- 17) Ha  $(T - S) < B < (T + S)$ , akkor visszatérünk a második lépéshez.
- 18) Ha  $(B - S) > T$ , akkor átment a jelölt.
- 19) Ha  $(B - S) < T$ , akkor megbukott.
- 20) Térjünk vissza az első lépéshez. (Wright, 1988)

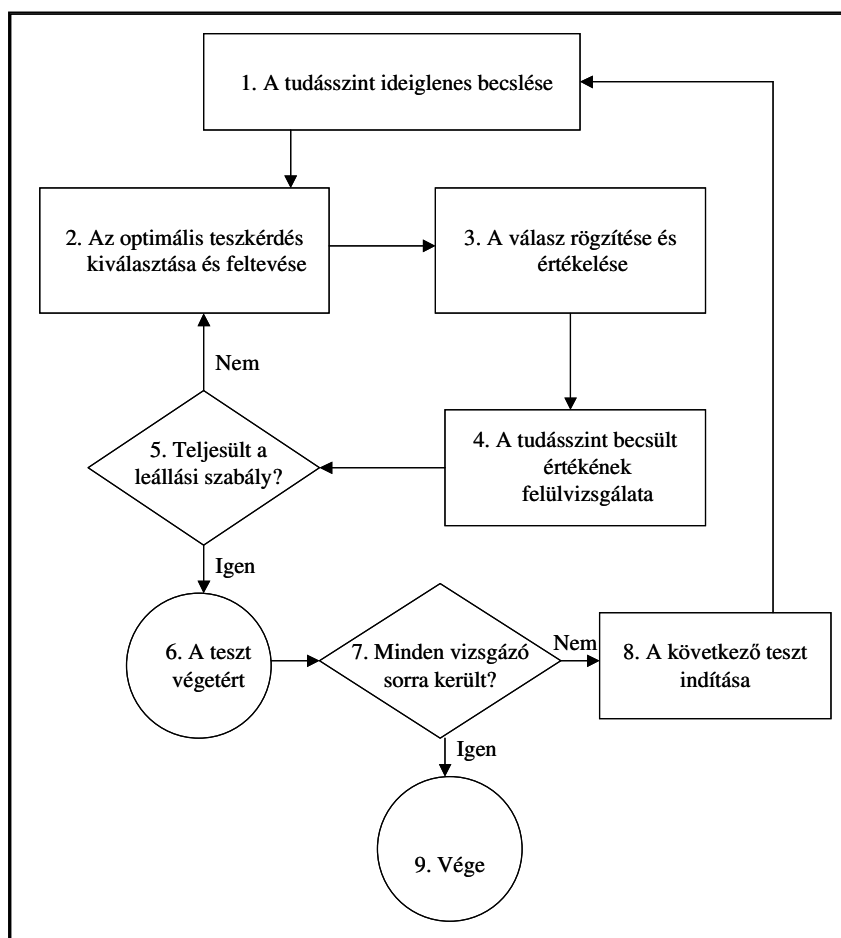
Ezt az algoritmust gyorsan és könnyen lehet alkalmazni bármely számítógépes adaptív tesztelés alapjaként. Nem csak a teszt folyamatát, de „leátlási szabályokat” és az eredmény megállapításának módját is magába foglalja, világos és megalapozott elveket követve.

### III.2.4. A számítógépes adaptív tesztelés további módszerei

#### III.2.4.1. Válaszadási elmélet

A Válaszadási Elmélet (Wainer & Mislevy, 1990) olyan statisztikai keretrendszer, amelyben a vizsgázót a szellemi képességeit leíró pontszámok segítségével jellemezzük, melyeket az egyes tesztkérdésekhez kapcsolódó teljesítmények, kérdés-statisztikák és a vizsgázó képességeinek összekapcsolásával kapunk.

A tesztkérdésekhez különböző paramétereket társíthatunk, így nehézségi szint, megkülönböztető erő, találgatási tényező paraméterét. A nehézségi szint mutatja meg, hogy milyen nehéz az adott tesztkérdés. A megkülönböztető erő határozza meg mennyire képes az adott kérdés különbséget tenni az eltérő tudásszinttel rendelkező hallgatók között. A találgatási tényező, pedig meghatározza, hogy mekkora a valószínűsége annak, hogy a hallgató egyszerű találgatással a helyes választ jelöli meg. Minden vizsgáztatást megelőzően szükség van a tesztbank tartalmának kiegyensúlyozására is. Ellenőrizni kell, hogy nincs sem alul-, sem túltesztelt tudásterület. Végül a tesztelés megkezdése előtt az egyes tesztkérdésekhez kapcsolódó paraméterek értékeit is meg kell határozni. A tesztelés folyamatát a következő 17. ábra mutatja be:



17. ábra: Az adaptív tesztelés folyamatábrája  
(Thissen és Mislevy, 1990)

---

A Válaszadási Elmélet és Linacre megközelítése között nincs a tesztelés lényegét érintő különbség. A vizsgáztatás folyamata mind a két esetben a 17.ábrán bemutatott lépéseknek megfelelően zajlik. Az eltérés az egyes tesztelemekek értékelésében mutatkozik. Linacre mindkét megközelítésében a válasz helyessége határozza meg, hogy milyen teljesítmény értéket rendelünk az egyes teszt elemekhez. A Válaszadási Elméletben a kérdések nehézségi szintje és a válaszok helyessége mellett, további, a kérdésekhez rendelt paraméterekkel finomítják, pontosítják a jelöltek teljesítmény-értékelését. Nem szükséges az összes paramétert felhasználni a kérdések jellemzéséhez. Az alkalmazott paraméterek számától függően más-más képletet kell alkalmazni a szignifikancia szint (azaz a tudás becsült szintjének) meghatározására. Ha csak egy paramétert (a nehézségi szint meghatározására szolgáló paramétert) alkalmazunk, akkor a Rasch modellt kapjuk.

### **III.2.4.2. Tudástér Elmélet**

A Tudástér Elmélet (Knowledge Space Theory – Doignon és Falmagne, 1985, Falmagne et al., 1990) szerint az ellenőrizendő területet csomópontok irányított gráfjaként ábrázoljuk, ahol minden csomópont egy problémát, vagy a problémák egy osztályát jelöli, az élek pedig a köztük lévő kapcsolatokat. A hallgató tudását reprezentáló modell ennek a gráfnak lesz egy része. A Válaszadási Elmélettel ellentétben, ahol a vizsgázónak egyszerre csak egy képességét lehet vizsgálni, a Tudástér Elméleten alapuló adaptív tesztelés során a hallgató által elsajátított ismeretek, képességek egy halmazát is fel tudjuk mérni. Ez a halmazt tudásállapotnak nevezzük.

A tesztelt tudásterület egy tudástér formáját veszi fel. Ha a kérdéses tudásterület négy problémakörrel (tesztelendő elemmel) jellemezhető, akkor a vizsgázó 24 féle tudásállapottal rendelkezhet. Tegyük fel, hogy a tudástér a következő problémakörökkel írható le: {a, b, c, d}. Ebben az esetben a jelöltek a következő tudás állapotok egyikével rendelkezhetnek:

{}, {a}, {b}, {c}, {d}  
{a, b} {a, c} {a, d} {b, c} {b, d} {c, d}  
{a, b, c} {a, c, d} {a, b, d} {b, c, d} {a, b, c, d},

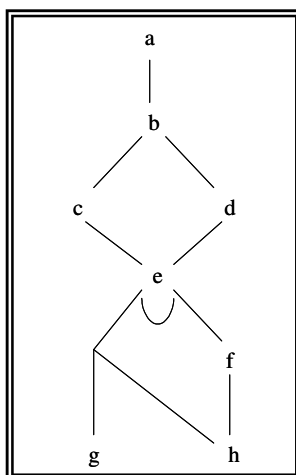
ahol {} a null állapotot jelöli, mely ahhoz a hallgatóhoz kapcsolódik, aki egyetlen problémakört sem tud megoldani. Továbbá azt is figyelembe kell venni, hogy nem minden esetben lehetséges valamennyi felsorolt tudásállapot elérése. Nemcsak azon tudásállapotok elérése lehetséges, melynek minden problémakörét helyesen megoldotta a vizsgázó. A tesztelemekek között meghatározott kapcsolatoknak köszönhetően következtetni is tudunk egyes képességek meglétére. Tegyük fel, hogy a vizsgázó meg tudja oldani a 'b' problémakört és így az előre meghatározott kapcsolatok alapján arra is tudunk következtetni, hogy az 'a' problémakörhöz kapcsolódó tudást is elsajátította. Ez egyben azt is jelenti, hogy minden olyan tudásállapot, mely tartalmazza 'b', annak tartalmaznia kell 'a'-t is. Ebben az eset-

---

ben  $\{b\}$ ,  $\{b, c\}$ ,  $\{b, d\}$ ,  $\{b, c, d\}$  tudásállapotok elérése nem lehetséges. A lehetséges tudásállapotok összessége a tudásstruktúra.

A felméréndő terület tudástérként való ábrázolását követően az adaptív tesztelés célja, hogy olyan hatékonyan és pontosan határozzuk meg a hallgató tudásállapotát, amennyire csak lehetséges. Ha az első feltett kérdésre helyesen válaszol a hallgató, akkor ebből már következtethetünk arra, hogy meg tudja oldani a kérdés előfeltételeként meghatározott (szülő) problématerületet is, éppen ezért nem kell külön megvizsgálnunk ez utóbbi területet. A következtetés lehetővé teszi a tesztelési folyamat jelentős rövidítését.

A 18. ábra egy tudásteret ábrázol ÉS/VAGY gráf formájában. A csomópontok jelölik a problématerületeket, az élek pedig kapcsolatokat (hogymelyik problémakörnek mi az előfeltétele). A 'b' csomópont ÉS kapcsolatot jelöl, azaz ha erre helyesen válaszol a jelölt, akkor arra következtethetünk, hogy mind a két előfeltételét, azaz 'c'-t és 'd'-t is elsajátította. Az 'e' csomópont VAGY kapcsolatot jelöl, így ha erre helyesen válaszol a hallgató, akkor ez egyben azt jelenti, hogy a következő két algráf egyikét elsajátította: 'g' és 'h' csomópontokat vagy az 'f' és 'h' csomópontokat.



18. ábra: Problémakörök kapcsolata a Tudástér elméletben

Ez a megközelítés túllép a már bemutatott módszereken, hiszen az adott tananyag egyes részeihez, problématerületeihez kapcsolja a kérdéseket. A problématerületek közötti kapcsolatok meghatározásával pedig a felmérés felgyorsítható. A módszer tehát már az oktatás eredményességére vonatkozóan is nyújt információkat. Ugyanakkor a tananyag strukturálása elnagyolt, számos kapcsolat ábrázolására nincs lehetőség, így a jelölt tudásáról sem kaphatunk pontos képet.

Az adaptív vizsgáztatásra számos további módszert dolgoztak ki, azonban ezek lényeges tulajdonságaikban nem térnek el a fent bemutatott megközelítésekől.

---

### **III.3. Az Oktatási ontológiával támogatott adaptív tudásszint-felmérés módszere**

A harmadik hipotézis szerint „az egyén által ismerni vélt tudás hiányterületei tételesen feltárhatók az oktatási ontológia által reprezentált ismeretanyag és az adaptív tesztelés eredményei alapján”.

A kialakítani kívánt adaptív vizsgáztatási rendszer célja, nem csupán a vizsgáztatás hatékonyságának és gyorsaságának növelése, az eredmények gyors közzétevése, hanem az is, hogy rámutasson azokra a tudásterületekre, tananyag részekre, melyeken a egyénnek hiányossági vannak. A tudásanyag szemantikai viszonyainak feltárását biztosítja az oktatási ontológia, így a hipotézis igazolásához még szükség van a vizsgáztatás módszerének kidolgozására is. Így be kell mutatni az adaptív vizsgarendszer felépítését és működését és azt, hogy milyen megoldással kapcsolható össze a két rendszer. Végül a vizsgarendszer és az Oktatási ontológia összekapcsolását követően konkrét tananyag tesztelésre is sor kerülhet.

#### **III.3.1. Az adaptív vizsgáztatás módszertana**

Cél olyan módszertan kialakítása, melynek segítségével a hallgató tudásszintje a lehető legkevesebb kérdés feltevésével pontosan meghatározható. A hiányterületek tételesen feltárásához az ontológiának megfelelően modellezett tananyag és kérdésbank nyújt segítséget.

##### ***III.3.1.1. A vizsgáztatás célja, a probléma bemutatása***

A felmérés során a kérdések előre megadott halmazából válogatunk. Minden kérdés a következő jellemzőkkel bír:

- Egy – pontosan egy – 'T' tudásterület kapcsolódik a kérdéshez. Az adott kérdés ezt a területet teszteli.
- A kérdés szövege.
- A válaszok halmaza, melyek közül egy meg van jelölve, mint a helyes válasz.
- A kérdés „nehézségi” értéke (például 0-100 közötti skálán mérve). Ez az érték megmutatja, hogy a kérdéshez csatolt tudásterületnek milyen mély ismerete szükséges a kérdés helyes megválaszolásához.

A megfelelő módszer kialakítását megelőzően rögzíteni kell a tudásfelmérés célját és a követendő elvek körét. Ezen tényezők és elvárások alapján határozható meg a módszertan alapvető jellemzői:

- Ha a vizsgázó magasan az elvárt szint fölött teljesít, tehát megfelelő szinten ismeri a területet, akkor átment a vizsgán.
- Ha a vizsgázó messze az elvárt szint alatt teljesít, tehát nem ismeri a területet, akkor megbukik.



- 
- Végül, ha a vizsgázó (alulról vagy felülről) közelíti az elvárt szintet, akkor a végeredményt a tudásterület részterületeinek tesztelése után mondjuk ki.

Emellett szükség van arra is, hogy ha a fő tudásterületet megfelelő szinten ismeri a jelölt, akkor is fel lehessen tárni, hogy az altudásterületek közül melyeken küzd komoly hiányosságokkal. A módszertannak biztosítania kell, hogy a fő tudásterületek alterületeit is ki lehessen kérdezni.

### **III.3.1.2. Vizsga**

#### **(1) A vizsga indítása**

A vizsga indításához a következő paramétereket kell megadni:

- a tesztelendő 'T' tudásterületet,
- a  $0 \leq alacsony \leq elégséges \leq magas \leq 100$  értékeket, illetve
- egy igaz/hamis értéket, mely előírja, hogy a 'T' tudásterület alterületeinek kikérdezése feltétlenül szükséges-e vagy sem.

Az alacsony, elégséges és magas értékek szerepe a következő: a vizsga során folyamatosan, minden lépésben meg kell becsülni a vizsgázó aktuális tudásszintjét. Az *elégséges érték* az a szint, melyet a vizsgázónak teljesítenie kell a „megfelelt” (vagy elégséges) minősítéshez. Amennyiben a jelölt – tartósan – az *alacsony érték* alatt marad, akkor kijelenthetjük, hogy az anyagról „semmilyen” tudása nincs. Ebben az esetben felesleges kikérdezni az altudásterületeket is, hiszen az egész 'T' tudásterület ismerete egyetlen lyuk. Amennyiben a vizsgázó – tartósan – a *magas érték* felett teljesít, akkor ismerete a 'T' tudásterületről jóval több, mint elégséges; így úgy tekintjük, hogy a jelölt jól ismeri az egész területet és nincsenek hiányosságai.

#### **(2) A vizsga menete**

A kiindulási feltételek meghatározása után megkezdhetjük a vizsgáztatást. A tesztelés menetének kidolgozásához a Rasch modell (Rasch, 1992) megállapításait használtam és következő lépésekből áll:

- Felteszünk néhány kérdést, melyek nehézségi szintjei az elégséges érték körül mozognak.
- A válaszok helyességéből kiszámítjuk a  $D$  értéket, mely a vizsgázó aktuális, becsült tudásszintjét jelöli.
- Újabb kérdéseket teszünk fel, melyek nehézségi szintje mindig az éppen megbecsült tudásszintnek felel meg. A válaszok alapján korrigáljuk  $D$ -t. Mindezt a leállási feltételek teljesüléséig ismétljük<sup>57</sup>.
- Amennyiben  $D < alacsony$ , a tudásterület ismerete teljesen hiányos, így ennek a tudásterületet felmérését lezárhatjuk.
- Amennyiben  $D > magas$  a tudásterület ismerete teljes, felmérést lezárhatjuk.

---

<sup>57</sup> A leállási feltételek meghatározása 97. oldalon található.

- Amennyiben *alacsony*  $< D <$  *magas*, le kell mérnünk az alterületek ismeretét is. Külön-külön meghatározzuk minden alterület felmérésének eredményét. Így meghatározhatjuk, hogy a vizsgázó tudása elegendő-e vagy sem az adott altudásterületen, illetve a részeredmények aggregálásával kapunk egy értéket arra is, hogy a főterület mennyire ismeri a vizsgázó. Ha ez az érték  $<$  elégséges, a vizsgázó megbukott, egyébként átment a vizsgán.

A kérdésfeltevés módjára vonatkozóan pontosítást kell megfogalmazni. A kérdéseket ugyanis nem csak a közvetlenül 'T' tudásterülethez csatolt kérdések köréből választhatjuk ki, hanem a 'T' alterületeihez kapcsolt, illetve azok alterületeihez kapcsolt (és így tovább) kérdések valamelyikét is kiválaszthatjuk. A bemutatott vizsgamenet estén a következő intuíció alkalmazásával biztosíthatjuk a kérdésfeltevésnek ezt a módját: ha egy nagyobb témakörből vizsgáztatunk, akkor az alterületeket kevésbé mélyen kell ismernie a jelöltnek (legalábbis beszámolni róluk), mint ha csak az adott alterületből vizsgázna. Ez annyit jelent a vizsgáztató szemszögéből, hogy ha egy főterület tesztelésekor egy  $N$  nehézségű kérdést szeretnénk feltenni, és ezt a kérdést valamely alterületből választjuk, akkor onnan egy  $N$ -nél könnyebb nehézségű kérdést kell feltenni. Így elérjük azt, hogy minél nagyobb témát ölel fel a vizsga, annál „átfogóbb”<sup>58</sup> kérdések születnek az alterületekből. Szükség van tehát egy korrekciós tényezőre, ami egy kérdés 'T<sub>1</sub>'-re vonatkozó nehézségét átszámítja 'T<sub>2</sub>'-re vonatkozó nehézségre. Például ha 'T<sub>1</sub>'- alterülete 'T<sub>2</sub>'-nek, és 'T<sub>1</sub>'-ben egy kérdés  $N$  nehézségű, akkor mennyinek számítsuk 'T<sub>2</sub>' tesztelésekor?). A következő függvény alkalmas ennek átszámítására:

$$jav(N) := N \cdot c^{\#T_1 / \#T_2 - 1},$$

ahol  $c$  egy konstans 0 és 1 között<sup>59</sup>,  $\#T$  pedig az adott tudásterület „nagyságát” jelenti. A tudásterület nagyságát meghatározhatjuk a kapcsolódó kérdések száma, a tudásterületen belül található fogalmak, összefüggések száma, vagy akár a tudásterület példáinak száma alapján is. Adott tudásterület összetettségét és így a nagyságát is leginkább a kapcsolódó fogalmak és összefüggések száma írja le, így a következőkben ezt értem a tudásterület nagysága alatt. Így a vizsgázó becsült tudásszintjét a jól megválaszolt kérdések korrigált nehézséggel súlyozott összege és az összes feltett kérdés korrigált nehézség értékének hányadosa adja. Vagyis

$$D = \frac{\sum_i x_i \cdot jav(n_i)}{\sum_i jav(n_i)} \cdot 100.$$

Ha tehát a vizsga végén ez az érték az alacsony szint alatt vagy a magas szint fölött van, akkor végeztünk a tudásterület tesztelésével és kiírjuk a megfelelő vizsgaeredményt. Ha pedig a két szint között van ez az érték, akkor ugyanezzel az algoritmussal rekurzív módon teszteljük az

<sup>58</sup> Az átfogó kérdés annyit jelent, hogy az adott alterület alapjaira, vagy más alterületekkel való kapcsolatára kérdezzünk rá és nem az alterület mélységeire.

<sup>59</sup> A  $c$  értékének a 0,9 érték megfelelő lehet.

---

alterületeket is. Az aktuális becsült tudásszint vagy az elégséges szint értéke, vagy pedig a jelenlegi  $D$  érték lesz. Az alterületek felmérésének eredményeiből képezzük a végeredményt, közben kiírva, hogy az aktuális tudásterületet jól vagy rosszul tudja a vizsgázó az elvárt szinthez képest.

### (3) *A következő kérdés kiválasztása*

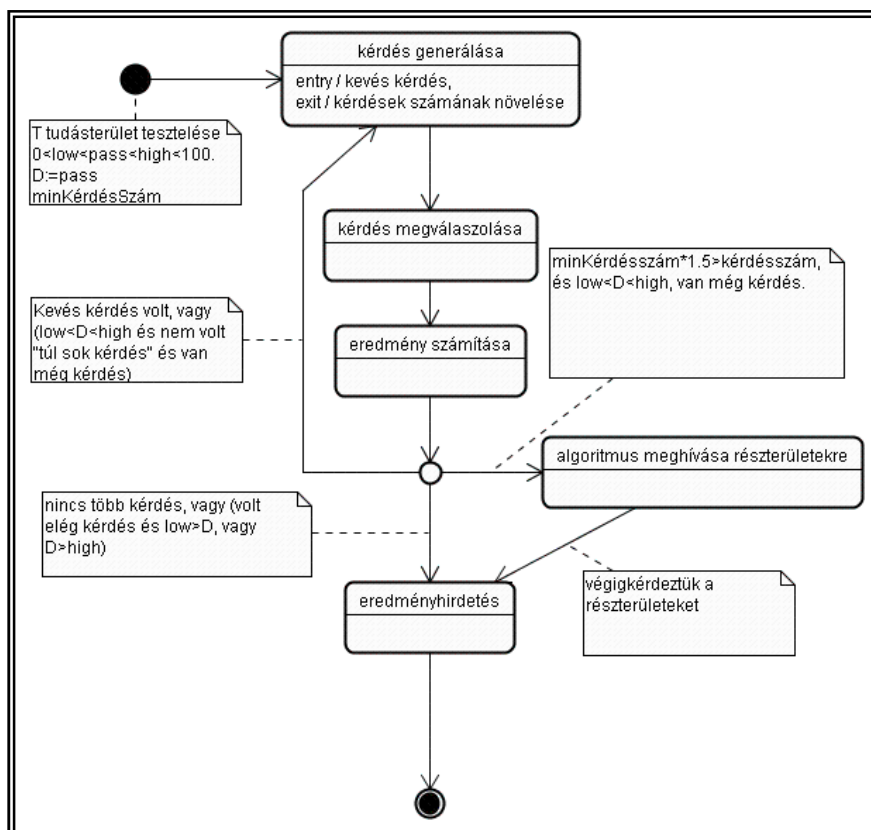
Ha a fő tudásterületeket az altudásterületekhez kapcsolt kérdések segítségével is kikérdezhetjük, akkor következő felméréendő kérdés kiválasztásának eljárási módját is meg kell határozni. Legyen a vizsgázó aktuális becsült tudásszintje  $D$ . Mivel súlyozott átlagot képzünk, ezért olyan kérdést szeretnénk feltenni legközelebb, ami indokolt ehhez a  $D$  szinthez képest. Tegyük fel lehetőség szerint legalább egy  $1,1 D$  korrigált nehézségű kérdést. (Annak érdekében, hogy a vizsga ne legyen előre kiszámítható, véletlen nehézségű kérdést kell választani, vagyis az  $1,1 D$  nehézségű kérdések köréből véletlenül kell választani, mondjuk normál eloszlás szerint). A megfelelő eredmények érdekében biztosítani kell, hogy az alterületek esetében nagyjából homogén legyen a kérdések eloszlása. Ez úgy érhető el, hogy az első lépésben kiválasztjuk a következő vizsgálandó területet (ami lehet a főterület vagy annak valamely alterülete is), mondjuk  $\#T_i$  szerint súlyozott esélyekkel. (Ha a kiválasztott területhez nem kapcsolódik egyetlen olyan kérdés sem, amit még nem tettünk fel, akkor újrásorsolunk addig, amíg nem találunk egy olyat, amely tartalmaz. Alterületeken belül már nem sorsoljuk ki az alterületet). Végül a kiválasztott terület csatolt kérdései közül választunk ki egy  $1,1 D$  körüli nehézségűt.

### (4) *Leállási feltételek*

Végül meg kell határozni azokat a szabályokat, melyek teljesülése esetén a tesztelés véget ér, biztosítva ezzel, hogy a lehető legkevesebb kérdés feltevésével végezzük el az értékelést.

Amennyiben már nincs több az adott tudásterülethez kapcsolódó kérdés, akkor nyilvánvalóan nem teszünk fel onnan további kérdést. A tesztelés szintén leáll, ha a minimális kérdésszám (vagy minden részterületet érintő kérdéssorozat) után a  $D < alacsony$ , vagy  $magas < D$ .

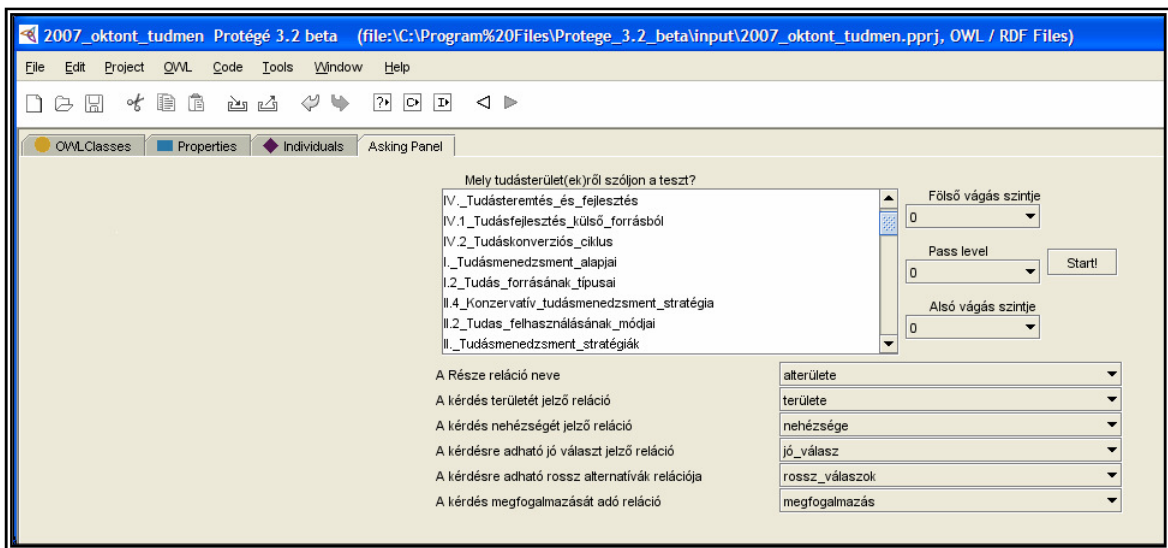
Ugyanakkor, ha a minimális kérdésszámnál másfélszer többet tettünk fel és  $alacsony < D < magas$ , akkor a 'T' tudásterület helyett a részterületeit vizsgáljuk alaposabban. Elindítjuk minden részterületre ugyanezt az algoritmust (átadva az aktuális kérdéslistát is). Miután mindegyikre elvégeztük a tesztelést, kiszámítjuk az aktuális  $D$  értéket a közben végrehajtott kérdések és válaszok alapján, majd ha  $D < elégséges$ , akkor a vizsgázó nem felelt meg, különben megfelelt. (Visszaadjuk  $D$ -t és a kérdések listáját, a részeredmények ezekből alakulnak ki.)



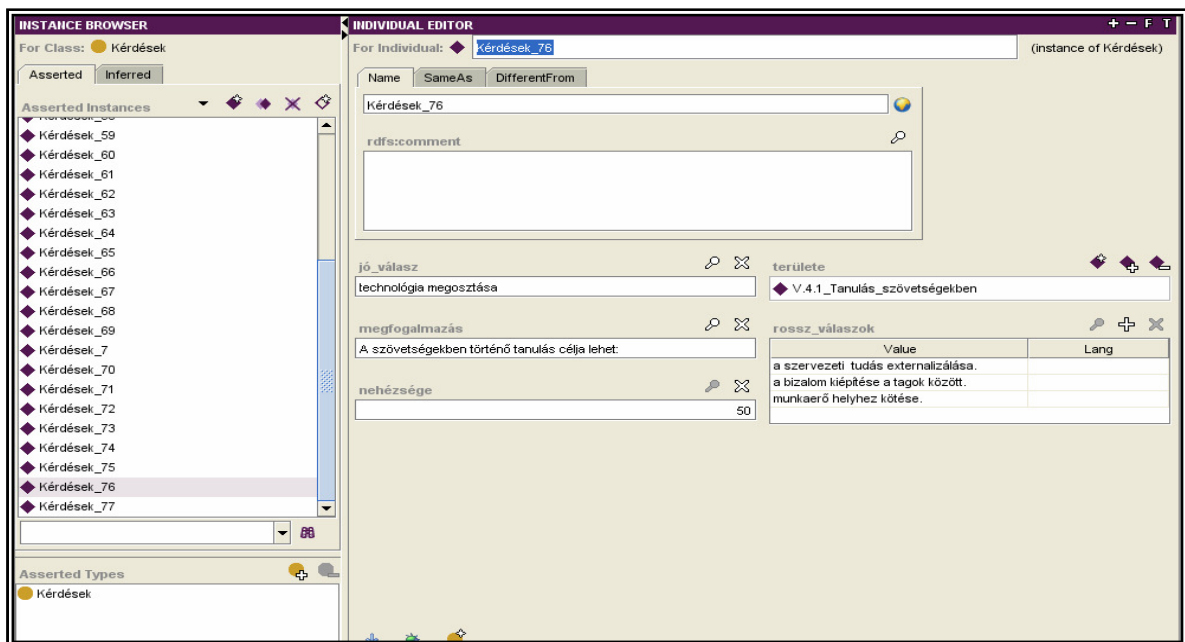
19. ábra: Az Oktatási ontológiával támogatott adaptív vizsgáztatás folyamata

### III.4. Oktatási ontológiával támogatott adaptív teszrendszer működtetése

A fejezet célja az ontológia modellhez kapcsoló adaptív vizsgarendszer gyakorlati alkalmazásának bemutatása, a teszt menetének szemléltetése. A kidolgozott algoritmusra épülő alkalmazás úgy lett kifejlesztve, hogy a Protégé-2000 programba be lehessen illeszteni (20. ábra). Ennek következtében az eredeti elképzeléssel ellentétben a tesztbankot is beillesztetem az ontológiába. Az ontológiastruktúrában külön osztályként jelenik meg a „Kérdés” osztály. Ez nem jelenti az ontológia lényegi változását, azonban ezzel biztosított, hogy a tesztbank struktúráját egyértelműen az ontológia modell határozza meg. Új elem felvétele esetén a következő jellemzőit kell megadni a kérdésnek: az ontológia azon elme, amelyre rákérdez, a kérdés megfogalmazása, a jó válasz, a rossz válaszok és a kérdés nehézsége (21. ábra).

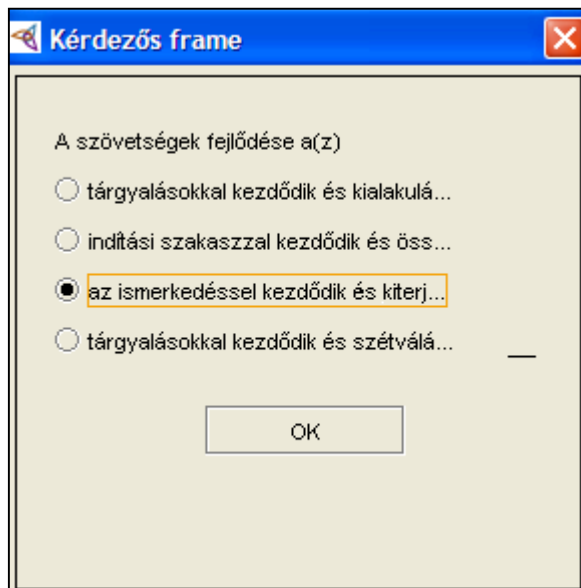


21. ábra: A tesztfelület a Protégé-ben



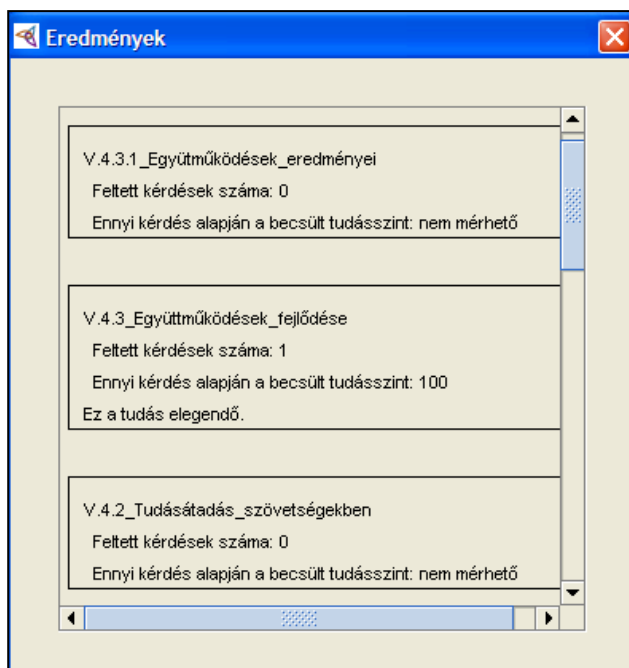
20. ábra: A „Kérdés” osztály jellemzői és relációi

A vizsga megkezdése előtt a teszt felületen be kell állítani, a küszöbértékeket (alacsony, megfelelt és magas szintet), illetve, hogy mely tudásterületeket szeretnénk kikérdezni. Ez a felület szolgál a teszt fő paramétereinek beállítására (20. ábra). A bemutatni kívánt példa esetében a Tudásátadás tudásterületet és annak valamennyi altudásterületét kérdeztem ki. A 22. ábra már a kérdező felületet mutatja be.

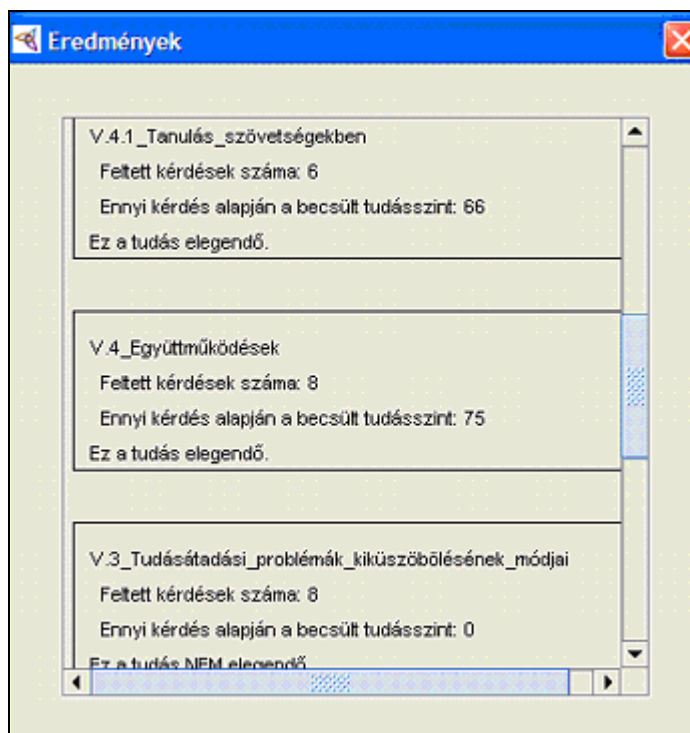


22. ábra: Az adaptív vizsgarendszer kérdező felülete

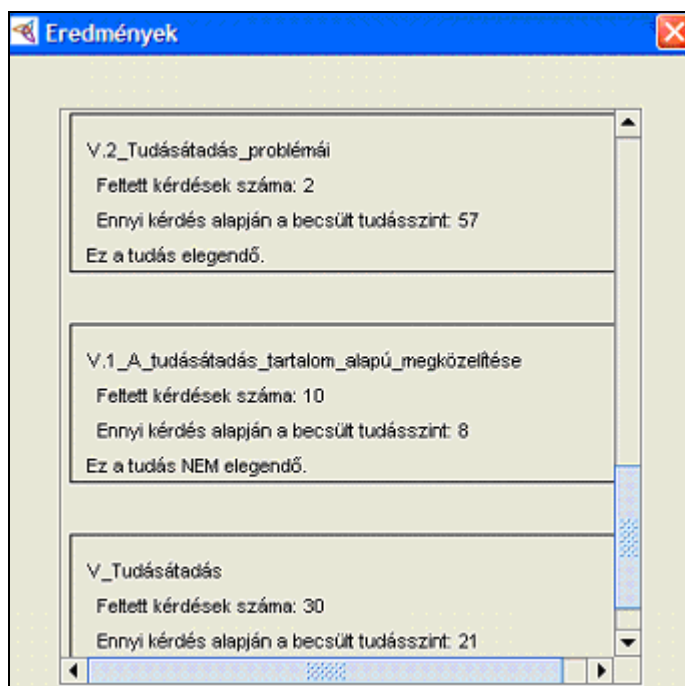
A vizsgarendszer a megadott válaszok helyességétől függően teszi fel a következő kérdést és ez határozza meg, hogy hány kérdést tesz fel összesen a vizsga folyamán. A példa esetében a Tudásátadás tudásterület mellett további nyolc altudásterületet kérdeztem ki, ennek felmérésre 30 kérdést tett fel a rendszer. Ez a 25. ábrán, a Tudásátadás tudásterületnél látható. (Az alterületekre vonatkozó kérdéseket is beleszámolja). A 23., 24. és 25. ábra az eredményeket mutatja be tudásterületenként.



23. ábra: A számonkérés eredményeinek értékelése - I.



24. ábra: A számonkérés eredményeinek értékelése - II.



25. ábra: A számonkérés eredményeinek értékelése - III.

A rendszer valamennyi tudásterületnél kiírja, hogy hány kérdést tettünk fel és meghatározza a becsült tudásszintet is a feltett kérdések nehézsége és a válaszok alapján. Például a 'Együtműködések' tudásterületre vonatkozóan 8 kérdést tett fel, amelyből hét az altudásterületekre vonatkozott. Összesítve az eredményeket ezt altudásterületet 75%-os szinten ismerte

---

a jelölt (24. ábra), azonban a többi altudásterületen olyan gyöngén szerepelt, hogy összességében csak 21%-ra teljesített a Tudásátadás területén.

Az ábrákból, az is jól látszik, hogy két tudásterületre nem tett fel kérdést. Ennek az az oka, hogy ezekhez a területekhez nincs kérdés hozzárendelve. A rendszer segítségével tehát az is jól látható, hol hiányos a tesztbankunk.

### **III.5. Hiányterületek tételes feltárásának módszere**

Az ontológiai megközelítés előnyeit és széleskörű felhasználási lehetőségeit igazolja a következőkben bemutatásra kerülő új, az előzőtől eltérő tesztelési módszertan. Az újabb megközelítés a HEFOP 3.3.1 „Tudás-szint-kiegyenlítő, rövid ciklusú e-learning kurzusok kifejlesztése” elnevezésű projekt keretében került kidolgozásra. Alapját szintén az általam kifejlesztett Oktatási ontológia modell adja, azonban a vizsgáztatás célja, menete és módszere eltérő. Sem a tudásszint pontos meghatározása, sem becslése nem tartozik az elsődleges célok közé ebben az esetben. A hangsúly a hiányterületek részletes feltárásán van, hogy valóban megtaláljunk minden egyes területet, amelyen a jelöltnek hiányosságai vannak. A projekt keretében kialakított tudásértékelési módszertan és az erre épülő vizsgarendszer prototípusának ismertetése után, a rendszer tesztelése során született eredményeket elemzem és összegzem.

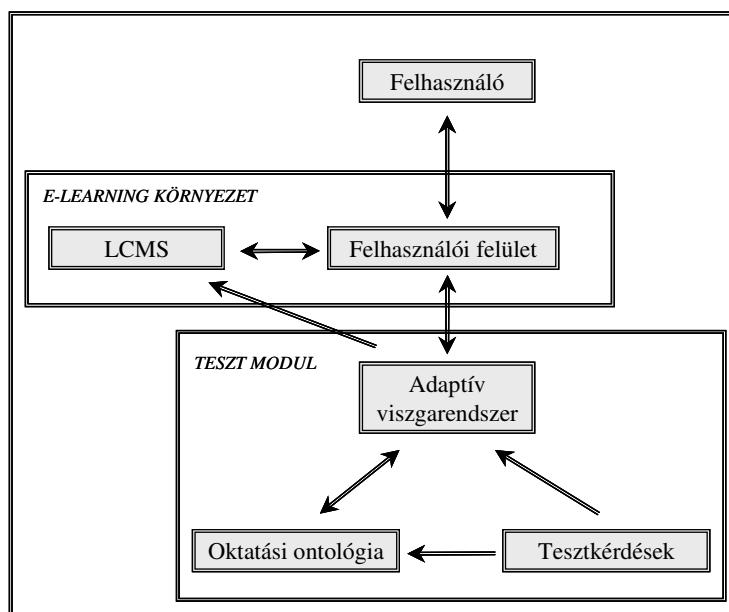
A megközelítés jogosultságát és szükségességét alátámasztják a bolognai folyamat keretében támasztott követelmények. Az Európai Felsőoktatási Térséghez való csatlakozás megköveteli a képzési szintek közti átmenet és a hallgatói mobilitás biztosítását. Ugyanakkor az egyes képzési szintekre belépni kívánók sokféle és különböző színvonalú előképzettségi, nyelvi, képzési formabeli kombinációkkal rendelkezhetnek. Adott képzési program által nyújtott ismereteken, képességeken túl az egyéntől is függ a tudás hiányának összetétele és mértéke. Így gyakori és a specialítások miatt nehezen kezelhető probléma lesz az adott alap vagy mesterszak belépési követelményeire való felkészítés.

Az ilyen jellegű igények földrajzilag elosztott módon, helyi szinten kis létszámban, de országosan akár nagy tömegben is jelentkezhetnek, ezek intézményi szintű kielégítése a speciális igények és egyéni problémák, jellegzetességek miatt nagyon nehézkes és fajlagosan költséges lenne hagyományos formában. E probléma megoldására nyújtottak választ a projekt keretében kifejlesztett távoktatási kurzusok. Az e-learning testreszabhatósága, általános elérhetősége, alacsony fajlagos költsége (mértéghasznosságos szempontok) ideális eszköz arra, hogy a felsőoktatás szintjei és a munka világa közti mozgások feltételrendszerét biztosítsa.

Az átjárhatósági probléma kezelésének lényege egy olyan interfész kialakítása, amely egy adott képzési szintre belépni szándékozó egyénnél az illető korábbi teljesített szintek, vállalati képzések és gyakorlati tapasztalatok figyelembevételével személyre szabott képzési/felkészítési programot alakít ki, megoldva ezzel az előzetesen szerzett ismeretek beszámításának problémáját. A megoldás bizonyos elemei akár a képzési szintek teljesítése során felismert hiányosságok menetközbeni korrekcióját is segít-



hetik, az oktatást kísérő ad hoc támogatás formájában. A megközelítés fő összetevőit ábrázolja a 26. ábra. Az adaptív tudástesztelési modell két fő modulból áll. Az egyik a Teszt modul, mely az Oktatási ontológiát, a tesztbankot és a vizsgarendszert foglalja magába. A másik modul pedig az e-learning környezet, mely egy oktatási tartalommenedzsment rendszert (Learning Content Management System) és hallgatókkal való kommunikációt biztosító oktatásmenedzsment rendszert (Learning Management System) tartalmazza (Kismihók és Vas, 2006). Az ábra nyilai az információáramlás irányát mutatják.



26. ábra: Az adaptív tudástesztelési modul

A projekt keretében a képzési struktúrán belül elsősorban az alap (BSc) és mester (MSc) szint közötti átmenet biztosítását tűztük ki célul. A megoldás szemléltetésére a gazdaságinformatikai képzést választottuk.

A mobilitás legfőbb akadálya, hogy a különböző alapképzések kimenetei nem adnak homogén bemenetet adott mesterképzés számára. Éppen ezért olyan módszert kell kidolgozni, mellyel fel tudjuk térképezni, hogy mit nem tud a hallgató. Másképp fogalmazva meg kell határoznunk azokat a tudásterületeket, melyeket feltétlenül el kell sajátítani a hallgatónak a mesterképzés megkezdése előtt. A felzárkózás elősegítése érdekében célirányosan és egyénre szabottan ki kell jelölni az elsajátítandó tananyagot a hallgató számára. A tanulást követően pedig, újra felmérést kell végezni, azaz meg kell bizonyosodni róla, hogy a hallgató valóban elsajátította-e a számára kijelölt tananyagot.

### III.5.1. A hiányterületek feltárására irányuló tesztelés menete

A tananyagok ontológiai modelljének kidolgozása és a tesztkérdések megfogalmazása után kerülhet sor a tesztelésre. A tudásértékelésre és a hi-

---

ányterületek feltárására irányuló vizsga menete<sup>60</sup> a következő: A tesztelés a tudásterületek hierarchiájának csúcsából indul ki. A hallgató olyan vizsgasort kap, mely az éppen vizsgálandó tudásterületet teljesen lefedi. Ha a jelölt helyesen válaszol a kérdésekre – azaz ha a kérdésekre adott válaszok alapján elér egy bizonyos pontszámot (például 60%-ot) – akkor az adott tudásterülethez kapcsolódó valamennyi altudásterületre és alapfogalomra vonatkozó kérdéseket is feltesszük. Ha a jelölt az adott tudásterület bármelyik alapfogalmára (összefüggésekre, stb.) vonatkozó kérdésre rosszul válaszol, akkor az adott tudásterületet nem fogadjuk el, hiszen a terület alapjaival nincs tisztában. Ha azonban az adott tudásterület valamely altudásterületével kapcsolatos kérdésre válaszol helytelenül a jelölt, akkor csak az adott altudásterületen tekintjük hiányosnak a vizsgázó ismereteit, de az egész tudásterületet nem vetjük el. Ha valamely altudásterületre vonatkozó kérdésekre jól válaszol a hallgató, akkor ugyanúgy járunk el, mint a fő tudásterületknél. Azaz kikérdezzük az altudásterülethez kapcsolódó valamennyi altudásterületet és alapfogalmat, összefüggést. Ha a jelölt nem tud helyesen válaszolni a vonatkozó kérdésekre, akkor a tesztelési algoritmus lezárja a tananyag struktúrájának ezt az ágát és az altudásterület újbóli elsajátítását javasolja a hallgatónak. A tesztelés akkor ér véget, ha az adott tudásterület minden alterülete (és a struktúra általuk lefedett ága) esetében meggyőződünk a jelölt tudásáról vagy hiányosságairól. A hallgató ekkor kap egy összefoglaló értékelést, amelyben szerepelnek mind a helyesen, mind a helytelenül megválaszolt kérdések. Ez utóbbiak esetében az is pontosan meg van jelölve, hogy a tananyag mely fejezetét, alfejezetét kell a hallgatónak átvennie, hogy hiányos tudását pótolni tudja. A Függelék V.4 pontjában található az értékelő lap egy mintája.

### **III.5.2. A megvalósítás és kivitelezés**

A projektben tizenkét olyan hazai felsőoktatás intézmény vett részt, ahol gazdaságinformatikai képzés folyik. Intézményenként más-más e-learning tananyagok kerültek kifejlesztésre, melyek a következők:

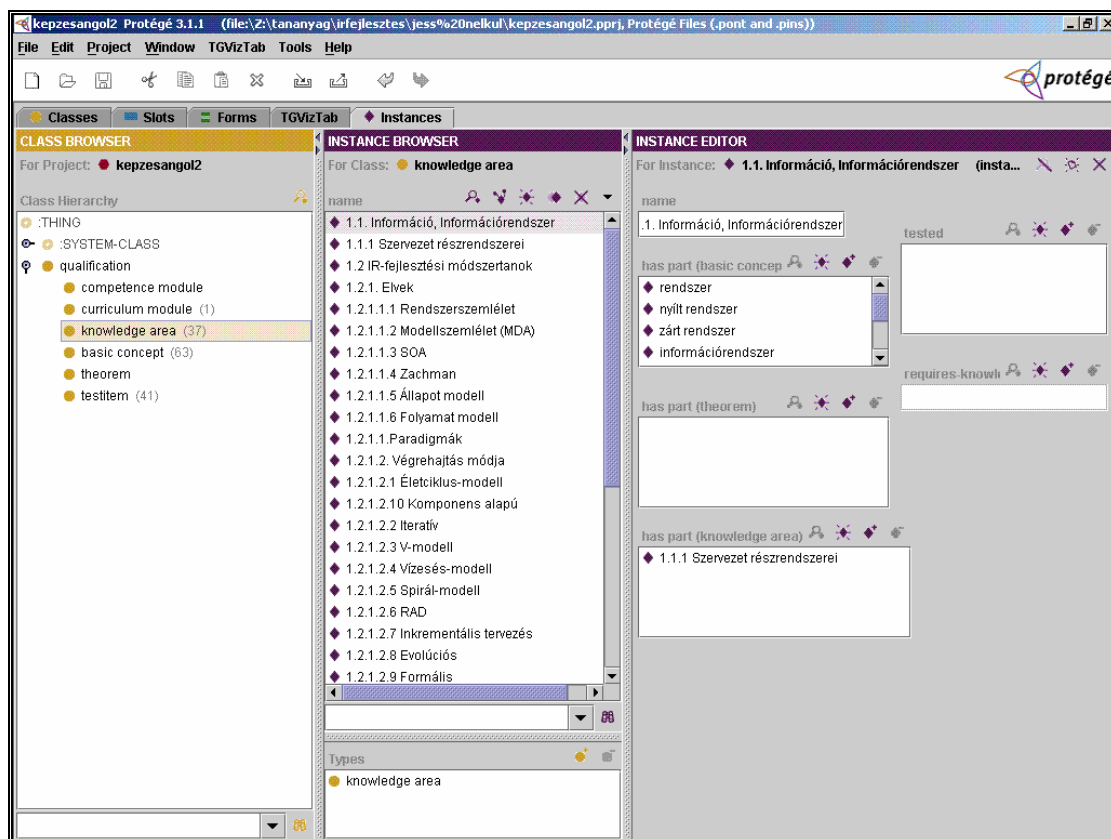
- Adatbázisok
- Alkalmazások fejlesztése
- Informatika
- Lineáris algebra
- Matematika vagy Számelmélet
- OO programozás és Java
- Operációkutatás
- Rendszerfejlesztés I.
- Rendszerfejlesztés II.
- Számítógép architektúrák

---

<sup>60</sup> A vizsga módszertanát az Információrendszerek tanszék munkatársai dolgozták ki: Gábor András, Kő Andrea, Szabó Ildikó, Kismihók Gábor és Vas Réka.

- Vezetői számvitel
- Vezetés-szervezés

A tananyagok elkészítése mellett a szerzők feladata volt az adott terület ontológiai modelljének kidolgozása is. A tesztelési algoritmus és a tesztrendszer prototípusának<sup>61</sup> kidolgozása a Corvinus Egyetem feladata volt. Az ontológia szerkesztésére szintén a Protégé-2000 ontológiaszerkesztő eszközt választottuk, mely számos projektben bizonyította már megfelelőségét és hatékonyságát (Szabó, 2006). A 27. ábra az ontológia egy részét mutatja a Protégé-2000 szerkesztő eszközben.



27. ábra: A hiányterületek feltárását támogató gazdaságinformatika ontológia

Az eredeti ontológia modellel összehasonlítva néhány változtatás mellett döntöttünk a projekt során, melyek a következők:

A „része” reláció pontosításra szorult. Kiemeltük (a relációk elnevezésében is), hogy a tudásterületnek alapfogalom, összefüggés és példa is lehet a része. Az ontológia tudáselemekkel történő feltöltése során ugyanis nehéz volt megkülönböztetni, hogy melyik „része” reláció jelöli például a tudásterület és összefüggés közötti „része” relációt.

A tesztelés megkönnyítése érdekében a felelet-választós kérdéseket is beépítettük a tudásmodellbe osztály formájában. Ez az osztály a tudásterület, alapfogalom és összefüggés osztályokkal van kapcsolatban. (Vas és Szabó, 2006)

<sup>61</sup> A teszt rendszer prototípusát Szabó Ildikó fejlesztette.

A tesztelési algoritmus Java programnyelven íródott, mely lehetővé tette, a szintén Java-alapú Protégé-be történő beillesztését. (Szabó, 2006a).

### III.5.3. A hiányterületek feltárását támogató vizsgarendszer tesztelésének eredményei

A projekt keretében 12 intézmény 291 hallgatója tesztelte a tananyagokat és a tesztelési rendszert, melynek során 3755 tesztet töltöttek ki, ami 74504 kérdés megválaszolását jelenti. A 10. táblázat intézményi bontásban tartalmazza az adatokat. Az adott oktatási intézmény által feldolgozott tananyag megnevezése mellett a tananyaghoz tartozó tudásterületek és altudásterületek és a kapcsolódó kérdések száma került feltüntetésre a táblázatban.

KIDOLGOZÓ INTÉZMÉNY	TANTÁRGY MEGNEVEZÉSE	VIZSGÁLHATÓ TUDÁSTERÜLETEK SZÁMA	KÉRDÉSEK SZÁMA
Pécsi Tudományegyetem	Adatbázisok	134	285
Eötvös Loránd Tudományegyetem	Alkalmazások fejlesztése	40	70
Dunaújvárosi Főiskola	Számítógép architektúrák	341	306
Berzsenyi Dániel Főiskola	Informatika	104	150
Budapesti Műszaki és Gazdaságtudományi Egyetem	Rendszerfejlesztés II.	97	133
Széchenyi István Egyetem	Rendszerfejlesztés I.	182	81
Nyugat-Magyarországi Egyetem	OO programozás és Java	260	267
Eötvös Loránd Tudományegyetem, Kaposvári Egyetem, Szegedi Tudományegyetem	Matematika, Lineáris algebra, Operációkutatás	365	497
Budapesti Gazdasági Főiskola	Vezetői számvitel	135	177
Miskolci Egyetem	Vezetés-szervezés	183	156
<b>ÖSSZESEN</b>	<b>–</b>	<b>1841</b>	<b>2122</b>

10. táblázat: A feldolgozott tananyagok, tudásterületeik és kapcsolódó kérdések száma

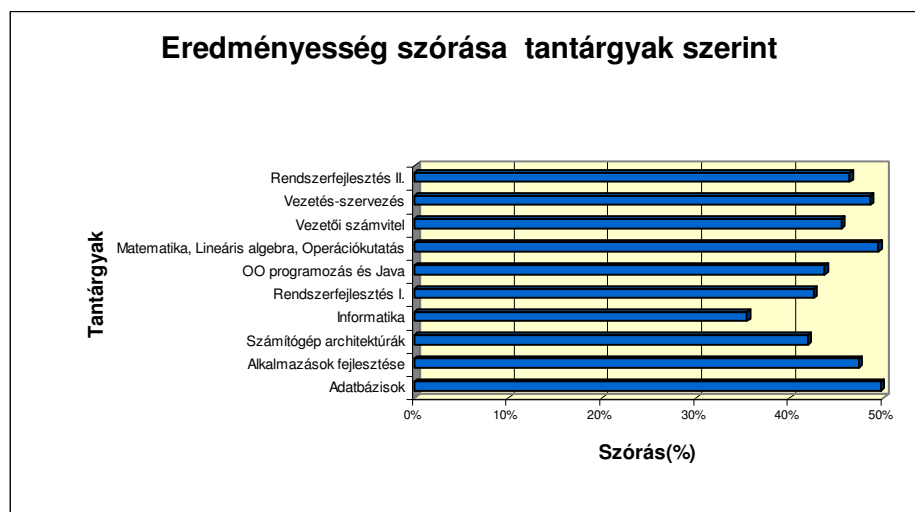
A 11. táblázat foglalja össze a tesztelés legfontosabb teljesítménymutatóit, így az átlagos válaszidőt tudásterületenként, az átlagos eredményességet és a válaszok és eredmények szórását. Az átlagos eredményességet a helyes válaszok és az összes feltett kérdés hányadosaként kapjuk. Érdeemes azonban megfigyelni a tesztek, pontosabban a válaszok két további jellemzőjét. Az egyik – az eredményességhez kapcsolódóan – az eredmények szórása. Látható, hogy az esetek többségében a válaszlehetősé-

gek terjedelmének csaknem 50%-án szóródtak a válaszok. Jelentősebb eltérést csak az Informatika teszt mutat, ahol viszonylag szűkebb sávban születtek az eredmények. A válaszadások eredményességét is figyelembe véve megállapítható, hogy ez a teszt az átlagosnál könnyebbre sikeredett (ezt alátámasztja a 29. ábra is). A másik jellemző a válaszidő, amelynél nagy eltéréseket ugyan nem tapasztalhatunk a különböző tárgyaknál, viszont elgondolkodtató, hogy a hallgatók mindössze 26 másodpercet töltöttek a kérdés tanulmányozásával és a válasz megadásával. Ez különösen a kérdések – tárgyból fakadó – komplexitása miatt érdekes, hiszen a Matematika tesztnél a meglehetősen bonyolult kérdések értelmezésénél ennél nagyobb időszükségletet feltételeztünk.

TESZTEK	ÁTLAGOS VÁLASZIDŐ (MÁSODPERC)	ÁTLAGOS EREDMÉNYESSÉG	EREDMÉNYEK SZÓRÁSA	VÁLASZOK SZÁMA
Adatbázisok	23,284	55,20%	49,73%	8175
Alkalmazások fejlesztése	26,085	66,07%	47,35%	5730
Számítógép architektúrák	27,000	77,30%	41,90%	2982
Informatika	24,545	85,28%	35,43%	23618
Rendszerfejlesztés I.	27,000	76,33%	42,52%	1259
OO programozás és Java	26,217	74,25%	43,73%	8854
Matematika, Lineáris algebra, Operációkutatás	26,176	57,36%	49,46%	14201
Vezetői számvitel	26,069	70,77%	45,49%	2145
Vezetés-szervezés	27,000	61,96%	48,57%	1112
Rendszerfejlesztés II.	26,875	65,93%	46,37%	6428
<b>ÖSSZESEN</b>	<b>26,025</b>	<b>69,05%</b>	<b>45,06%</b>	<b>74504</b>

11. táblázat: A vizsgarendszer teszteléséhez kapcsolódó eredménymutatók

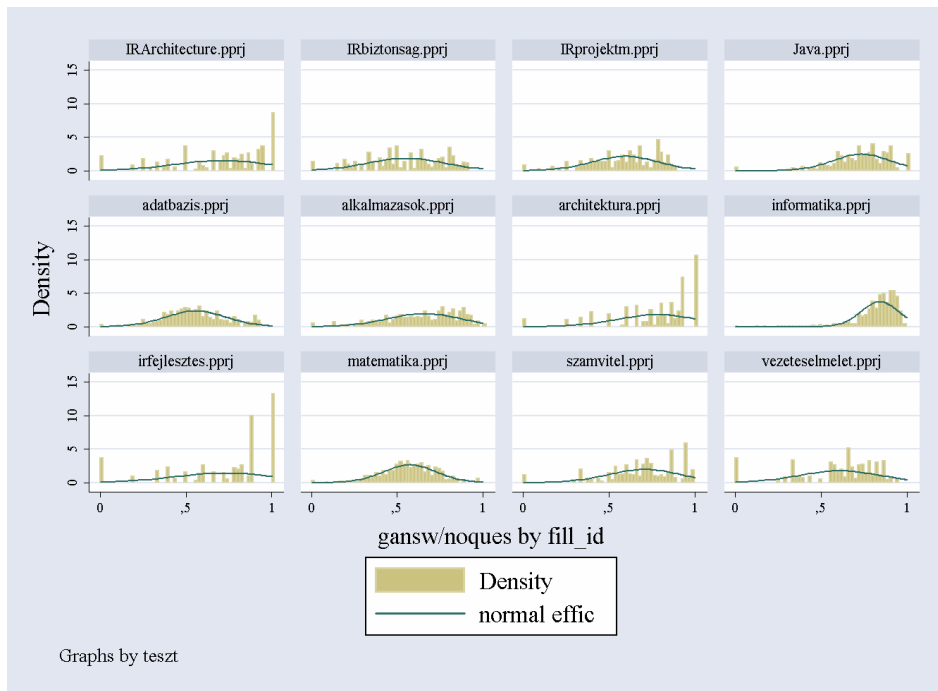
Ugyanakkor azt is figyelembe kell venni, hogy egyes tananyagok tesztelésére kevesebb idő állt rendelkezésre és területek nehézsége sem feltétlenül egyforma. A megválaszolt kérdések számát tekintve a Matematika (Számelmélet, Lineáris algebra és Operációkutatás) végzett a második helyen. Az átlagos eredményesség azonban ennél a területnél volt a legrosszabb és a szórás is a második legmagasabb értékkel bír. Ez az eredmény korántsem meglepő, hiszen évről – évre a matematika tárgyak bizonyulnak a legnehezebbeknek a gazdasági, gazdaságinformatikai szakok hallgatóinak. A 28. ábra a tudásterületenkénti eredményesség szórását mutatja.



28. ábra: Az eredményesség szórása tantárgyak szerint:

A 29. ábra X tengelye az eredményességet ábrázolja 0 és 1 közötti értékekkel. Az egyes grafikonok a tesztek eredményeit jelentik ebben a térben, továbbá minden teszt eredményére a normális eloszlás görbáját illesztettük a könnyebb elemzés kedvéért. A függőleges tengelyen a kitöltött tesztek száma szerepel sűrűségként. Az ábra tehát azt mutatja, hogy az egyes tesztekben különböző 0-1 között adott eredményeket elért teszt-kitöltések milyen mértékben fordultak elő. Ebből következtetni lehet az adott teszt nehézségére, erősségére is. Az eddigi következtetéseink alapján várható a Matematika teszt erős szereplése, amely közelíti a normális eloszlást, ahogy az az ábrán is látható. Ellenpélda lehet a két architektúrával foglalkozó teszt, ahol a normális eloszlást egyáltalán nem követő, hibátlan tesztkitöltéseket kiemelkedő mértékben tartalmazó eredmények születtek. Hasonló még az Információrendszerek fejlesztése teszt is, amely azonban kevésbé releváns a relatíve kevés rendelkezésre álló adat miatt. Érdekes megfigyelni az eloszlások csúcspontjainak elhelyezkedését is. A Matematika, Alkalmazások fejlesztése és Adatbázisok tesztek kevéssel az átlagosnak feltételezett 50%-os eredményesség fölött található, ráadásul szépen követik a normális eloszlást. A Java és Informatika tesztek szintén jól igazodnak a normális eloszláshoz, ám szórásuk kisebb, csúcspontjuk pedig a magasabb eredményességi tartományban található – ami a tesztek relatív könnyebbségére utal. A többi tesztnél nehezen figyelhető meg hasonló sűrűsödés – hacsak a néhányuknál erősen kiemelkedő 100%-os vagy annak közelében levő eredményesség.

Jelen kutatás szempontjából a legfontosabb eredmény azonban az, hogy a tizenkét felsőoktatási intézmény oktatói alkalmasnak találták az Oktatási ontológia modelljét az egyes tananyagok tudásanyagának strukturálására és feldolgozására, illetve a gyakorlatban is alkalmazták modellt.



29. ábra: Eredményességek eloszlása

---

## IV. Összefoglalás

### IV.1. A kutatás célkitűzései

Az Európai Felsőoktatási Térséghez való csatlakozás megköveteli a képzési szintek közti átmenet és a hallgatói mobilitás biztosítását. Ugyanakkor az egyes képzési szintekre belépni kívánók sokféle és különböző színvonalú előképzettségi, nyelvi, képzési formabeli kombinációkkal rendelkezhetnek. Adott képzési program által nyújtott ismereteken, képességeken túl az egyéntől is függ a tudás hiányának összetétele és mértéke. Így gyakori és a specialitások miatt nehezen kezelhető probléma lesz a hallgatók adott alap vagy mesterszak belépési követelményeire való felkészítése. Versenyképességük megőrzése érdekében számos követelménynek kell a felsőoktatási intézményeknek megfelelniük kezdve a hallgatói igényektől, a munkaerő-piaci elvárásokon át egészen a technikai, technológia kihívásokig. Dolgozatom rámutatott, hogy az oktatási területek ontológiai modellezése hogyan segíti az oktatást modernizáló tudásalapú eszközök és rendszerek tervezésében, fejlesztésében és alkalmazásában.

A dolgozat elsődleges célja, olyan ontológia modell kialakítása volt, mely a Budapesti Corvinus Egyetem Gazdálkodástudományi Karának gazdaságinformatikus szakán megszerezhető kompetenciákon keresztül kapcsolatot teremt a munkaerőpiac által támasztott követelmények, valamint az oktatott tárgyak tartalma között.

Célként tűztem ki továbbá, hogy az ontológia modell támogassa a hallgatók által elsajátított ismeretek adaptív módon történő felmérését, különös tekintettel az oktatás eredményességére és a hiányterületek tételes feltárására.

### IV.2. A kutatás eredményei

A kutatás kitűzött céljainak megfelelően a kiindulási pontot a gazdaságinformatikus szak keretében oktatott tárgyak és a képzés során megszerezhető kompetenciák körének meghatározása, illetve az eltérő területek közös vonásainak megragadására alkalmas ontológia modell kialakítása jelentette. Azonban a módszertani elvárásoknak megfelelően nemcsak az ontológia osztályainak, felépítésének lépésit, hanem az ontológia teljes életciklusához kapcsolódó tevékenységeket meg kell határozni és nyomon kell követni megvalósulásukat. Dolgozatomban a Sure-Studer féle megközelítés alapján kidolgozott módszertan segítségével mutatom be végrehajtandó feladatok körét és a kutatás eredményeit, melyek a következők:

- Az **Oktatási ontológia modellje**, mely alkalmas különböző megközelítésű és felépítésű tananyagok fogalmainak és a fogalmak közötti szemantikai viszonyok leírására és megragadására
  - Tudásmenedzsment tárgy ontológia modellje
- Az **adaptív vizsgáztatás módszertana**, mely az Oktatási ontológia által meghatározott struktúrára épít



- 
- Tudásszint felmérést támogató **számítógépes adaptív vizsgarendszer**, melynek működését az Oktatási ontológia és adaptív vizsgáztatási módszertan elvei határozzák meg

Továbbá a testreszabott ontológiaépítési módszertanon belül kidolgoztam a *tananyagok feldolgozásának lépéseit*, elősegítve ezzel az ontológia modell gyakorlati alkalmazását. A kutatás közvetett eredményének tekinthető a *hiányterületek tételes feltárását támogató vizsgarendszer*, mely szintén az Oktatási ontológiára épít (míg a tudásfelmérés módszertana eltér az általam kidolgozott megközelítéstől).

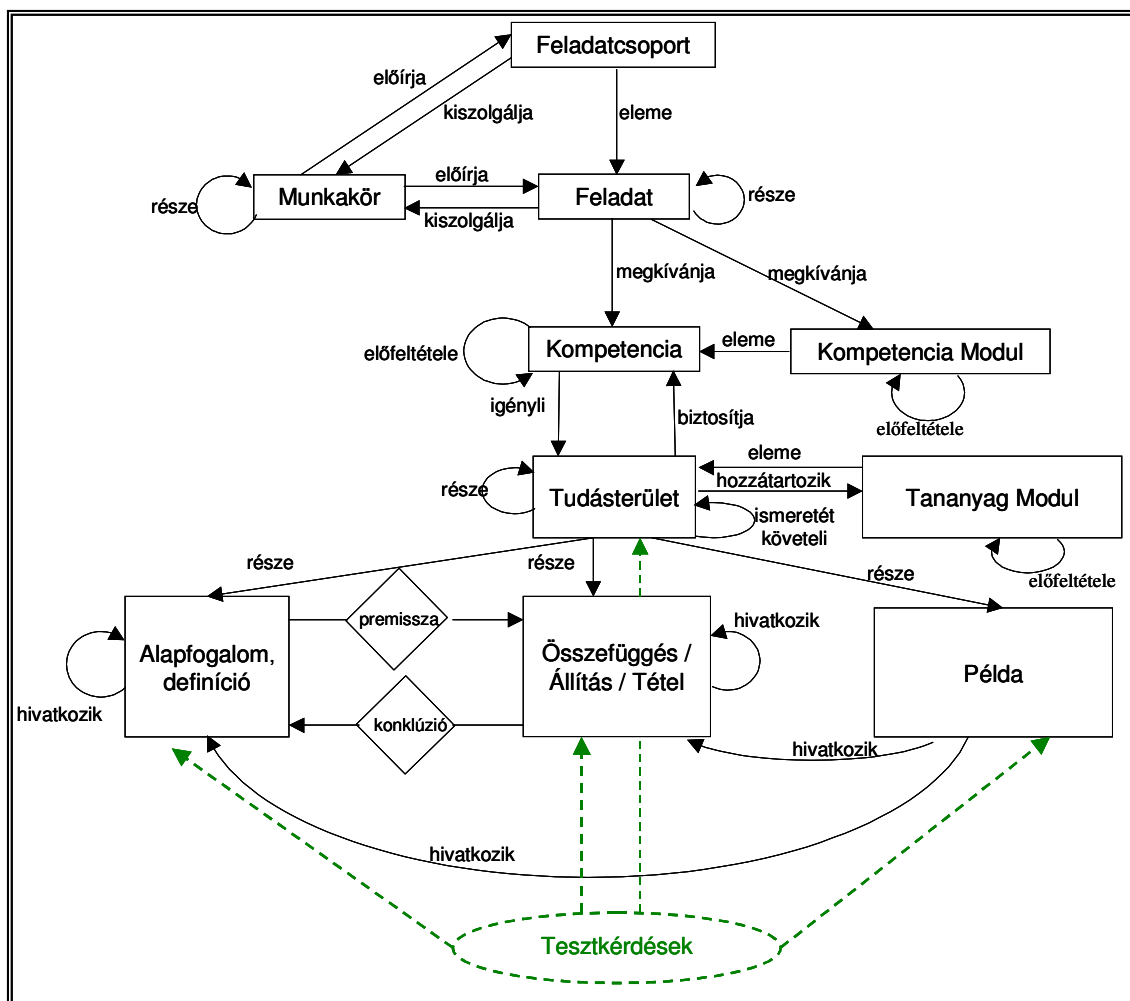
#### **IV.2.1 Az Oktatási ontológiához kapcsolódó megvalósíthatósági elemzés**

Az oktatási intézmények elsődleges feladata a hallgatói igényeknek való megfelelés. Ugyanakkor nem lehet figyelmen kívül hagyni a munkaadók elvárásait és az Európai Felsőoktatási Térséghez is igazodnunk kell. Ráadásul az adott képzésben résztvevő oktatók köre igen széles, alkalmazott módszereik és oktatási gyakorlatuk jelentősen eltérhet egymástól, de mindezek ellenére is biztosítani kell a tárgyak egymásra épülését. A rendszer megvalósíthatósága és sikeressége érdekében ezeknek a környezeti és szervezeti kihívásoknak kell megfelelni.

#### **IV.2.2 Az Oktatási ontológia fejlesztésének kiindulási fázisa**

A kiindulási fázisban kerül sor az ontológia követelményspecifikációjának elkészítésére. Tisztázni kell mi az ontológia célja, tárgyterülete, milyen alkalmazásokat fog támogatni, milyen tudásforrásokat fog felhasználni, illetve meg kell határozni a legfontosabb tervezési és szerkesztési elveket. Szintén ebben a szakaszban kerül sor az esetlegesen újrahasználható ontológiai részletes vizsgálatára, az ontológiát osztályainak kialakítására.

Ennek érdekében meg kell vizsgálni, hogy a képzés által közvetített tudás reprezentálására kialakítható-e olyan ontológiai modell, mely az oktatott tárgyak széles köre és eltérő jellege ellenére is alkalmas az egyes területek fogalmainak és a köztük fennálló szemantikai viszonyok ábrázolására. A fenti célt szem előtt tartva kiemelt szerepe van az ontológia „Tudásterület” osztályának, melynek elemeiből épül fel az adott tárgy tananyaga. A tudásterület adott tananyag egy-egy átfogó területének, nagyobb problémakörének ismeretanyagát foglalja magába. Ezek a tudásterületek már elég konkrétak ahhoz, hogy meg állapíthassuk, mely kompetenciák elsajátítását biztosítják, ugyanakkor könnyedén elemeire (alapfogalmak, összefüggések, példák) bonthatók a számonkérés támogatása érdekében. A mindezek figyelembevételével kialakított Oktatási ontológia modellje a 28. ábrán látható (ahol a szaggatott vonalak azt jelölik, a tesztkérdések az ontológia mely elemeihez kapcsolódnak).



30. ábra: Az Oktatási ontológia modellje

### IV.2.3 Az Oktatási ontológia finomítása

A finomítási szakasz célja az alkalmazásorientált ontológia kialakítása. Jelen kutatás esetében ez az egyes tantárgyak ontológia modelljének kialakítását jelenti. Több alszakaszra bontható, melynek során először az informális taxonómiáját kell felállítani, majd az elsődleges váz finomítására, bővítésére kerül sor szakértők bevonásával, végül az ontológia formális leírását kell megadni.

A végső cél az oktatás eredményességének, illetve a hallgatók tudásszintjének felmérése, melyet szem előtt kell tartani az ontológiaépítés teljes szakaszában. A fogalmi modell felépítése során a felülről-lefelé történő építkezés módszerét alkalmaztam. Az oktatás hasonló elven működik, hiszen, először meghatározzuk a fontosabb témaköröket és az órák keretében ezeket lebontjuk részterületekre és meghatározzuk összefüggéseit és alapfogalmait is. Az Oktatási ontológia alkalmazásának bemutatására és a vizsgarendszer tesztelésére szolgáló Tudásmenedzsment tárgy ontológia modelljében a következő elemek tartoznak a Tudásterület osztályba:

- Tudásmenedzsment alapjai (bevezetés, alapfogalmak és kapcsolataik)

- 
- Tudásmenedzsment stratégiák
  - Tudásmenedzsment technológiai támogatása
  - Tudásteremtés és fejlesztés
  - Tudásátadás
  - Intellektuális tőke mérése
  - Tudásmenedzsment szervezeti kérdései
  - Tudásmenedzsment emberi erőforrás kérdései

A következő lépés a tudásterületek felosztása nagyobb egységeire, azaz altudásterületekre. Jelen esetben a nyolc tudásterületet további 28 altudásterületre bontottam fel. Bizonyos esetekben az alterületeket is további részterületekre kell bontani. Ezzel egy elsődleges taxonómia készült el. Mielőtt meghatároznánk a tudásterületek alapfogalmait, összefüggéseit és példáit tisztázni kell még a tudásterületek közötti összefüggéseket is. Meg kell nézni, hogy adott tudásterület megköveteli-e valamely más tudásterület ismertét, előzetes elsajátítását. Szem előtt kell tartani, hogy itt más tananyagok tudásterületeinek ismerete is szükséges egy-egy tudásterület elsajátításához. A Tudásmenedzsment esetében több ilyen kapcsolat is létezik. A tudásmenedzsment stratégiák elsajátításhoz ismerni kell a Stratégia menedzsment egyes részeit is (például a vállalat erőforrás alapú megközelítését, illetve a vállalat alapvető képességek megközelítését is). Ezen kívül a Vezetés-szervezés és Emberi erőforrás menedzsment tárgyak egyes tudásterületeinek ismerete is elkerülhetetlen a tudásmenedzsment meghatározott részeinek megértéséhez. Ezt követően kerülhet sor az alapfogalmak, összefüggések és példák meghatározására, majd az utolsó lépés a tudáselemek közötti relációk feltárására.

Másrészről szükség van az ontológia tudásterület „feletti” részének kidolgozására is. A kompetenciákat a szakalapítási dokumentumok tartalmazzák. A bolognai elvárásoknak megfelelően két modulba soroltam őket: a BSc és MSc kompetencia modulba. Ezeket felbontva 54 különböző kompetenciát vettem fel az ontológiába. Továbbá három munkakört is meghatároztam, melyek ellátásához feltétlenül szükséges a Tudásmenedzsment tananyag ismerete. (Ezek a következők: Tudásmenedzsment szakértők, Tudásmenedzsment-projektvezető, Tudásmenedzsment vezető) Végül meghatároztam az adott munkakör keretében ellátandó feladatok körét és ezzel az ontológia valamennyi osztályát feltöltöttem.

#### **IV.2.4 Az Oktatási ontológia értékelése**

Célja a létrehozott ontológia hasznosságának elemzése. Az értékelés során meg kell vizsgálni, hogy az ontológia megfelel-e a követelményspecifikációban megfogalmazott elvárásoknak, az alkalmazási környezetben való tesztelésére is itt kerül sor. Az így kapott visszajelzések alapján az ontológia tovább finomítható, pontosítható. Az értékelés szoros kapcsolatban van a finomítás szakaszával, ugyanis ez a két lépés mindaddig ciklikusan ismétlődik, míg a célontológia nem éri el a kívánt színvonalat.

---

Az ontológia alkalmazási környezetben történő teszteléshez ki kell alakítani az adaptív vizsgáztatás módszertanát és a számítógépes adaptív tesztelés rendszerét egyaránt.

#### ***IV.2.4.1 Tudásszint felmérését támogató adaptív vizsgáztatás módszertana***

A rendszerben a vizsga indításához a következő paramétereket kell megadni:

- a tesztelendő 'T' tudásterületet, melynek struktúráját és elemeit az Oktatási ontológia pontosan leírja
- a  $0 \leq \text{alacsony} \leq \text{elégséges} \leq \text{magas} \leq 100$  értékeket, illetve
- egy igaz/hamis értéket, mely előírja, hogy a 'T' tudásterület alterületeinek kikérdezése feltétlenül szükséges-e vagy sem.

Az '*alacsony*', '*elégséges*' és '*magas*' értékek szerepe a következő: a vizsga során folyamatosan, minden lépésben meg kell becsülni a vizsgázó aktuális tudásszintjét. Az elégséges érték az a szint, melyet a vizsgázónak teljesítenie kell a „megfelelt” (vagy elégséges) minősítéshez. Amennyiben a jelölt – tartósan – az alacsony érték alatt marad, akkor kijelenthetjük, hogy az anyagról „semmilyen” tudása nincs. Így a vizsga kevesebb kérdés felvetésével lezárható. Ebben az esetben felesleges kikérdezni az altudásterületeket (a mélyebb részleteket) is, hiszen az egész 'T' tudásterület ismerete egyetlen lyuk. Amennyiben a vizsgázó – tartósan – a magas szint felett teljesít, akkor ismerete a 'T' tudásterületről jóval több, mint elégséges; így úgy tekintjük, hogy a jelölt jól ismeri az egész területet és nincsenek hiányosságai.

#### **Tudásszint felmérése – A vizsga menete**

A kiindulási feltételek meghatározása után megkezdhetjük a vizsgáztatást. A tesztelés menetének kidolgozásához a Rasch modell (Rasch, 1992) megállapításait használtam és következő lépésekből áll:

- Felteszünk néhány kérdést, melyek nehézségei az elégséges érték körül mozognak.
- A válaszok helyességéből kiszámítjuk a  $D$  értéket, mely a vizsgázó aktuális, becsült tudásszintjét jelöli.
- Újabb kérdéseket teszünk fel, melyek nehézségi szintje mindig az éppen megbecsült tudásszintnek felel meg. A válaszok alapján korrigáljuk  $D$ -t. Mindezt a leállási feltételek teljesüléséig ismételjük.
- Amennyiben  $D < \text{alacsony}$ , a tudásterület ismerete teljesen hiányos, így ennek a tudásterületet felmérését lezárhatjuk.
- Amennyiben  $D > \text{magas}$  a tudásterület ismerete teljes, felmérést lezárhatjuk.
- Amennyiben  $\text{alacsony} < D < \text{magas}$ , le kell mérnünk az alterületek ismeretét is. Külön-külön meghatározzuk minden alterület felmérésének eredményét. Így meghatározhatjuk, hogy a vizsgázó tudása elegendő-e vagy sem az adott altudásterületen, illetve a részered-

---

mények aggregálásával kapunk egy értéket arra is, hogy a főterület mennyire ismeri a vizsgázó. Ha ez az érték < *elégséges*, a vizsgázó megbukott, egyébként átment a vizsgán.

Az itt bemutatott módszert alkalmazó adaptív vizsgarendszert az Oktatási ontológia szerkesztésére alkalmazott Protégé-2000 ontológiaszerkesztő rendszerhez illesztettem, biztosítva ezzel a két megoldás gyors és hatékony együttműködését.

#### ***IV.2.4.2 Hiányterületek tételes feltárását támogató adaptív vizsgarendszer***

Az ontológiai megközelítés előnyeit és széleskörű felhasználási lehetőségeit igazolja a hiányterületek tételes feltárására irányuló, az általam kidolgozott módszertől eltérő megközelítés. Sem a tudásszint pontos meghatározása, sem becslése nem tartozik az elsődleges célok közé ebben az esetben. A hangsúly a hiányterületek részletes feltárásán van, hogy valóban megtaláljunk minden egyes területet, amelyen a jelöltnek hiányosságai vannak.

Az Európai Felsőoktatási Térséghez való csatlakozás megköveteli a képzési szintek közti átmenet és a hallgatói mobilitás biztosítását. Ugyanakkor az egyes képzési szintekre belépni kívánók sokféle és különböző színvonalú előképzettségi, nyelvi, képzési formabeli kombinációkkal rendelkezhetnek. Ez a módszer csak erre a problémára kíván megoldást nyújtani. A tudás felmérésnek célja, a tesztelés menete és elvei eltérőek, de a működés alapjait szintén az Oktatási ontológia határozza meg.

#### **A hiányterületek tételes feltárása – A vizsga menete**

A tesztelés a tudásterületek hierarchiájának csúcsából indul ki. A hallgató olyan vizsgasort kap, mely az éppen vizsgálandó tudásterületet teljesen lefedi. Ha a jelölt helyesen válaszol a kérdésekre – azaz ha a kérdésekre adott válaszok alapján elér egy bizonyos pontszámot (például 60%-ot) – akkor az adott tudásterülethez kapcsolódó valamennyi altudásterületre és alapfogalomra vonatkozó kérdéseket is feltesszük. Ha a jelölt az adott tudásterület bármelyik alapfogalmára (összefüggésekre, stb.) vonatkozó kérdésre rosszul válaszol, akkor az adott tudásterületet nem fogadjuk el, hiszen a terület alapjaival nincs tisztában. Ha azonban az adott tudásterület valamely altudásterületével kapcsolatos kérdésre válaszol helytelenül a jelölt, akkor csak az adott altudásterületen tekintjük hiányosnak a vizsgázó ismereteit, de az egész tudásterületet nem vetjük el. Ha valamely altudásterületre vonatkozó kérdésekre jól válaszol a hallgató, akkor ugyanúgy járunk el, mint a fő tudásterületeknél. Azaz kikérdezzük az altudásterülethez kapcsolódó valamennyi altudásterületet és alapfogalmat, összefüggést. Ha a jelölt nem tud helyesen válaszolni a vonatkozó kérdésekre, akkor a tesztelési algoritmus lezárja a tananyag struktúrájának ezt az ágát és az altudásterület újbóli elsajátítását javasolja a hallgatónak. A tesztelés akkor ér véget, ha az adott tudásterület minden alterülete (és a struktúra általuk lefedett ága) esetében meggyőződünk a jelölt tudásáról vagy hiányosságairól. A hallgató ekkor kap egy összefoglaló értékelést,

---

amelyben szerepelnek mind a helyesen, mind a helytelenül megválasztott kérdések. Ez utóbbiak esetében az is pontosan meg van jelölve, hogy a tananyag mely fejezetét, alfejezetét kell a hallgatónak átvennie, hogy hiányos tudását pótolni tudja.

Ez utóbbi megközelítés széles körű terjesztésére és tesztelésére volt lehetőség egy 12 magyar felsőoktatási intézményt magába foglaló kutatás keretében. Minden résztvevő kidolgozott egy-egy tananyagot, ennek elkészítette ontológia modelljét, majd intézményenként 20-30 hallgató tesztelte mind a tananyagokat, mind az adaptív vizsgarendszert. Jelen kutatás tekintetében kiemelten fontos siker, hogy a projektben résztvevő 12 intézmény oktatói alkalmasnak találták az Oktatási ontológiát saját tantárgyaik ismeretanyagának modellezésére. További fontos eredmény, hogy bizonyítottá vált, az ontológia modell alkalmas többféle megközelítésen alapuló vizsgáztatás támogatására is.

#### **IV.2.5 Az Oktatási ontológia karbantartása és fejlesztése**

Az ontológiával szembeni elvárások folyamatosan változhatnak, a tárgyakat fejleszteni, aktualizálni kell a bekövetkező változásokat visszatükrözve. Éppen ezért elkerülhetetlen az ontológia folyamatos karbantartása, amely alapvetően egy szervezeti folyamat. Szigorú szabályokat kell megfogalmazni arra vonatkozóan, hogy milyen időközönként, hogyan és kinek van lehetősége a változtatásokra, hogy milyen módon történjen a verzióváltás. Célszerű ebben az oktatás féléves felosztását követni, és ehhez igazítani a karbantartási feladatokat.

#### **IV.2.6 A kutatási eredmények értékelése**

A kutatás sikeresnek tekinthető, hiszen az alkalmazott módszertani megközelítés alkalmas volt a kutatási kérdések megválaszolására. Elkészült az Oktatási ontológia modellje, mely képes az oktatott tárgyak tudásterületeinek, fogalmainak és a köztük fennálló szemantikai viszonyoknak az ábrázolására. Ugyan léteznek már a hallgatók tudásának mérésére alkalmas adaptív tudásfelmérési módszerek, de emellett a dolgozat kereteiben kidolgoztam egy saját megoldást (illetve egy további kutatási projektben szintén újabb megközelítést alakítottunk ki), mely innovatív módon lép túl az eddigi megoldásokon. Végül elkészült és sikeresen működik az Oktatási ontológián alapuló adaptív vizsgarendszer. A bemutatott eredmények alapján mindhárom hipotézist igazoltnak tekinthetjük.

### **IV.3. További kutatási irányok**

A bemutatott eredmények és módszerek több területen is hasznosíthatók, több kutatási téma is kapcsolódhat a felvetett kérdésekhez. Az Oktatási ontológiának azért is van kiemelt szerep, mert az adaptív vizsgarendszer mellett számos más alkalmazás alapját adhatja. Az egységes fogalomkezelésnek és szemantikai viszonyok rendezésének kiemelt szerepe van, minden alkalmazásfejlesztési folyamatban.

---

A felépített Oktatási ontológia és Teszthalmaz a tudásszint meghatározása mellett más problémák megoldásában is segítségünkre van. Alkalmazhatjuk adminisztratív jellegű problémák megválaszolására, így például:

- Megkérdezhetjük a rendszertől mi a feltétele adott feladat ellátásának (milyen tudásterületet, kompetenciát igényel).
- Az alapján, hogy mit állít magáról adott munkakör betöltésére jelentkező jelölt, a rendszer adatokat szolgáltat arról, hogy elvileg milyen feladatokat képes ellátni.
- A kettő keveréke is megvalósítható. Ebben az esetben, ha megmondja a jelölt, hogy milyen feladatot akar ellátni (milyen munkakört akar betölteni) és azt is, hogy milyen tudással bír, akkor a modell alapján megállapíthatjuk, hogy milyen kompetenciákat kell még megszereznie.

A kutatás előrehaladása érdekében a legfontosabb a gazdaság-informatikus képzés tananyagainak feldolgozása. Ugyan az ontológia modell kialakítása hosszadalmas, sok odafigyelést igénylő feladat, a befektetés többszörösen megtérül. Minél több tárgyból áll rendelkezésünkre az ontológia modell és a kapcsolódó kérdéshalmaz, annál több diákot lehet bevonni a rendszer használatába, melynek segítségével nagy volumenű statisztikai felmérések is készíthetők a rendszerrel kapcsolatban. A tudáshiány tételes feltárásával minden egyes diák számára egyedileg meg lehet határozni a szükséges fejlődési irányokat, megkönnyítve ezzel a képzési szintek közötti váltást is.

A rendszer nem csak a tudáshiány megragadásában, hanem olyan további információk összegyűjtésében is segítséget nyújthat, mint a tipikus hibák köre, hogy átlagosan mennyi időt tölthettek egy-egy kérdés megválaszolásával (ez jelzi, hogy melyik kérdés bizonyult túl könnyűnek, melyik túl nehéznek), illetve az oktatás eredményességét meghatározó tényezők.

Az Oktatási ontológia világosan megmutatja hogyan épülnek egymásra a képzés tananyagai, hol vannak átfedések a tárgyak ismeretanyagai között. Mindezekon túl az ontológia modell kialakítása lehetőséget ad adott fogalmak eltérő értelmezésének tisztázására és a tananyagokban fellelhető ellentmondások kiszűrésére is.

Külön kutatásban lehet foglalkozni a kérdezés, kérdésfeltevés módszereivel. A kapcsolódó pedagógiai kérdésekkel.

A teljes ontológia kialakítása hosszadalmas folyamat és vizsgarendszer sem válthatja ki a tudásfelmérés minden típusát, azonban nagy előrelépést jelent a hatékonyság növelésében, az oktatás és az oktatási intézmény versenyképességének jelentős fejlesztésében is.

---

## V. Függelék

### V.1. Fogalomtár

#### (1) Fogalom (Concept)

A fogalmak, mint minden gondolkodás építőelemei alkotják az ontológiák alapját is. Még pontosabban, a fogalom olyan objektum osztályok mentális ábrázolása, melyek néhány közös tulajdonságuknak köszönhetően tartoznak össze.

#### (2) Ontológia összetevői (Ontology components)

Ontológiáknak, általában, öt fő alkotóelemét különböztetik meg: osztályok, relációk, tulajdonságok, axiómák és egyedek.

#### (3) Osztály (Class)

Az osztály olyan objektumok halmaza, melyek közös struktúrával és (/vagy) jellemzőkkel bírnak.

#### (4) Metaosztály (metaclass)

Olyan osztály, melynek egyedei maguk is osztályok vagy fogalmak.

#### (5) Taxonómia (Taxonomy)

A taxonómiák feladata az ontológiai vagy szakterületi tudás rendszerezése az általánosítás és specializáció relációinak alkalmazásával, melynek segítségével egyszerű és többszörös öröklődés is kifejezhető.

#### (6) Előre definiált primitívek (Predefined primitives)

Az előre definiált primitívek a következők: alosztálya, diszjunkt felsorolás, alosztályok teljes felsorolása, nem alosztálya.

#### (7) Alosztálya (Subclass of)

Ez a kapcsolat azt fejezi ki, hogy a fölérendelt (vagy általános) fogalomnak egyfajta speciális esete az alárendelt (vagy specifikus) fogalom.

#### (8) Diszjunkt felsorolás (Disjoint decomposition)

Olyan felosztás, melyben az érintett fogalmak mind alosztályai egy általános fogalomnak. A felsorolásnak nem kell teljesnek lennie, ami egyben azt is jelenti, hogy az adott általános fogalomnak olyan egyedei is lehetnek, melyek egyetlen alosztályban sem szerepelnek. Ugyanakkor az sem fordulhat elő, hogy létezik két olyan alosztály a felsoroltak között, melyeknek van közös részük.

#### (9) Alosztályok teljes felsorolása (Exhaustive subclass decomposition)

Ezt a relációt teljes diszjunkt felsorolásnak is nevezzük, ami azt jelenti, hogy az adott fogalom (ősosztály) bármely egyede a felsorolásban szereplő fogalmak valamelyikének (és pontosan egyikének) mindenképpen egyede kell, hogy legyen.



---

### **(10) Nem alosztálya (Not subclass of)**

A 'nem alosztálya' kapcsolatot annak kifejezésére alkalmazhatjuk, hogy egy fogalom nem speciális esete egy másik fogalomnak. Ezt a típusú kapcsolatot a legtöbb esetben az 'alosztálya' reláció tagadásaként fejezik ki.

### **(11) Relációk (Relations)**

A szakterület fogalmi és tulajdonságai közötti kapcsolatokat fejezik ki a relációk.

### **(12) Függvények (Functions)**

A függvények olyan n-változós relációk, melyeknél az első n-1 paraméter értéke egyértelműen meghatározza az n. paraméter (az eredmény) értékét.

### **(13) Integritási feltétel (Integrity Constraint)**

Az integritási feltétel olyan (általában egyszerűen ellenőrizhető) feltétel, amelynek nem teljesülése azt jelenti, hogy a vizsgált adat rossz, azaz valahol hiba van az ontológiában.

### **(14) Kiszámító eljárásokkal való meghatározás (Operational definition)**

A kiszámító eljárásokkal való meghatározás egy adott állítás értékeire eljárások, formulák vagy szabályok megadásával való következtetést vagy az állítás szemantikájának axiómák, szabályok segítségével történő meghatározását jelenti.

### **(15) Tulajdonságok (Attributes)**

A tulajdonságok (slot-ok) fogalmak, osztályok leírását, specializálását teszik lehetővé.

### **(16) Lokális tulajdonság (Local attribute)**

Lokális tulajdonságról akkor beszélünk, ha az csak egy speciális fogalomhoz kapcsolódik.

### **(17) Globális tulajdonság (Global attribute)**

Globális tulajdonság esetén a tulajdonság értelmezési tartománya nincs meghatározva (így az ontológia bármely eleméhez (egyedéhez, osztályához) kapcsolódhat).

### **(18) Osztály-tulajdonság (Class attribute)**

Az osztály-tulajdonság olyan tulajdonság, melynek értéke az adott fogalomhoz köthető, azaz annak minden egyes egyede esetén ugyanazt az értéket veszi fel.

### **(19) Egyed-tulajdonság (Instance attribute)**

Egyed-tulajdonságról akkor beszélünk, ha annak értéke a fogalom minden egyes egyede vagy az osztály minden fogalma esetén különböző lehet.

### **(20) Facet (Facet)**

Tulajdonságokra vonatkozó megszorítások (típus, alapértelmezett típus, számossági megszorítás).

---

## (21) Megszorítások (constraint)

A megszorítások segítségével osztályok és relációk közti összefüggéseket adhatunk meg. Hatáskör szempontjából kétféle megszorítás létezik: lokális és globális. Hatásmechanizmus szempontjából pedig meg szokás különböztetni a számossági megszorításokat és a felvett értékek szerinti megszorításokat.

## (22) Megszorítások hatáskör szerinti csoportosítása

### (a) Lokális megszorítás (Local constraint)

Egy megszorítás lokális, ha csak az ontológia egy konkrét osztályára mondjuk ki.

### (b) Globális megszorítás (Global constraint)

Egy megszorítás globális, ha az egész ontológiára érvényes.

## (23) Számossági megszorítások (Cardinality constraints)

A számossági megszorítások segítségével meghatározhatjuk az adott tulajdonság által egyedenként felvehető különböző értékek számának minimumát és/vagy maximumát is. Három számossági megszorítás létezik: a minimum, a maximum és a pontosan.

### (a) Számossági minimum (minCardinality)

A 'számossági minimum' megszorítás alkalmazásának eredménye azoknak az egyedeknek a halmaza, melyek egy adott relációban legalább a megadott minimumnak megfelelő számú, eltérő egyeddel állnak.

A megszorítás által előállított osztályt, ha a reláció  $R$  és a megadott minimum  $N$ , a leíró logikákban a  $\geq NR$  kifejezés jelöli.

### (b) Számossági maximum (maxCardinality)

A 'számossági maximum' alkalmazásának eredménye azoknak az egyedeknek a halmaza, melyek az adott relációban legfeljebb a megadott maximumnak megfelelő számú különböző egyeddel állnak.

A megszorítás által előállított osztályt, ha a reláció  $R$  és a megadott minimum  $N$ , a leíró logikákban a  $\leq NR$  kifejezés jelöli.

### (c) Előírt számosság (Cardinality)

A számossági előírás alkalmazásának eredménye azoknak az egyedeknek a halmaza, melyek az adott relációban pontosan a megadott értéknek megfelelő számú különböző egyeddel állnak.

A megszorítás által előállított osztályt, ha a reláció  $R$  és a megadott minimum  $N$ , a leíró logikákban az  $=NR$  kifejezés jelöli.

## (24) Felvett értékek szerinti megszorítások

Felvett értékek szerinti megszorításokat alkalmazunk olyankor, ha olyan osztályra van szükségünk, melynek elemeiben az a közös jellemző, hogy milyen más elemekkel állnak relációban. Három típusa létezik: az allValuesFrom, a someValuesFrom és a hasValue megszorítás.

---

**(a) Minden felvett érték korlátozása (allValuesFrom) megszorítás**

Az allValuesFrom megszorítás, mely egy relációt és egy osztályt kap paraméterül, azt az osztályt adja eredményül, melynek elemei pontosan azok az egyedek, melyek a megadott relációban csak a megadott osztálybeli elemekkel állnak. A relációnak egyed-értékű relációnak kell lennie.

A megszorítás által előállított osztályt, ha a reláció  $R$  és a paraméter osztály  $C$ , a leíró logikákban a  $\forall R.C$  kifejezés jelöli.

**(b) Felvett érték előírása (someValuesFrom) megszorítás**

Ez a megszorítás szintén egy relációt és egy osztályt kap paraméterül; azt az osztályt adja eredményül, melynek elemei pontosan azok az egyedek, melyek a megadott relációban vannak legalább egy olyan egyeddel, mely beleesik a megadott osztályba. A relációnak egyed-értékű relációnak kell lennie.

A megszorítás által előállított osztályt, ha a reláció  $R$  és a paraméter osztály  $C$ , a leíró logikákban a  $\exists R.C$  kifejezés jelöli.

**(c) Felvett érték előírása (hasValue) megszorítás**

Ez a megszorítás egy relációt és egy értéket kap paraméterül; azt az osztályt adja eredményül, melynek elemei pontosan azok az egyedek, melyek a megadott relációban állnak ezzel az értékkel (is). Ekkor a reláció lehet számértékű is.

A megszorítás által előállított osztályt, ha a reláció  $R$  és  $o$  a paraméter érték, a leíró logikákban a  $\exists R.\{o\}$  kifejezés jelöli.

**(25) Alapértelmezett slot érték (Default slot value)**

Feladata, hogy értéket rendeljen a fogalom adott tulajdonságához, abban az esetben, ha nincs explicit módon meghatározott értéke.

**(26) Típus (Type)**

A tulajdonság típusának – értékészletének – megszorítására alkalmazzák. Más terminológiákban ezt Range globális megszorításként definiálják.

**(27) Slot dokumentáció (Slot documentation)**

A tulajdonság természetes nyelvű meghatározását tartalmazza.

**(28) Axiómák (Axioms)**

Az axiómák olyan mondatok, melyek mindig igazak. Szerepük az ontológiákban, hogy megszorításokat tegyenek a tárolt információra, illetve, hogy igazolják azok helyességét és további, új információkra következtessenek.

**(29) Beágyazott axiómák (Embedded axioms)**

A beágyazott axiómák más olyan elemek meghatározásában szerepelnek, mint a fogalmak, relációk stb.

---

### (30) Megnevezett axiómák (Named axioms)

A megnevezett axiómák az ontológiák független elemeiként kerülnek bevezetésre.

### (31) Egyedek (Instances)

Az egyedek olyan kifejezések, melyek az adott szakterület elemeit írják le; általában egy adott fogalom elemeit reprezentálják.

### (32) Tények (Facts)

Elemek közötti reláció fennállását fejezik ki.

### (33) Kijelentések (Claims)

Adott egyed által állított tények kifejezése.

### (34) Szabályok (Rules)

A szabályok szerepe, hogy a tevékenységek, heurisztikák halmazát azok alkalmazásától függetlenül fejezzék ki.

### (35) Nyílt világszemlélet (Open world assumption)

A nyílt világszemléletben a fogalmak definiálásakor figyelni kell arra, hogy ha valami nem szerepel az ontológiában, arról nem teszi fel, hogy nem létezik, hanem úgy tekinti, hogy vagy létezik, vagy nem. A nyílt világszemléletben megfogalmazhatunk zárt világszemléletnek megfelelő kijelentéseket is, amennyiben a nyelv erre lehetőséget ad, (általában ad). Nyílt világszemléletű nyelvek: OWL, RDF(S), CycL

### (36) Zárt világszemlélet (Closed world assumption)

A nyílt világszemlélettel szemben csak a leírt dolgokat tekinti létezőnek, a nem leírtak biztosan nem léteznek. Ennek a tulajdonságnak azért van jelentősége, mert a következő rendszerek is építenek erre a nemlétezésre. A zárt világszemlélettel létrehozott ontológiát nehéz nyitottá tenni, önmagán belül nem is lehetséges. Zárt világszemléletű nyelvek: adatbázisok, Prolog, OCML, Ontolingua

### (37) Egyedi névfeltételezés (Unique name assumption)

A különböző nevű fogalmakról, relációkról felteszi, hogy különbözőek. (Azok a nyelvek, melyek nem élnek ezzel a feltevással, többnyire megengedik, hogy kifejezzük két különböző nevű objektumról, hogy azok valójában azonosak.)

### (38) Horn-klóz (Horn formula)

$(X_1 \wedge X_2 \wedge \dots \wedge X_n) \rightarrow (Y_1 \wedge Y_2 \wedge \dots \wedge Y_m)$  alakú logikai kifejezés, ahol az  $X_i$  és  $Y_i$  predikátumok *nem* lehetnek negáltak. Ekkor a  $(X_1 \wedge X_2 \wedge \dots \wedge X_n)$  részt *törzsnek*, a  $(Y_1 \wedge Y_2 \wedge \dots \wedge Y_m)$  részt *következménynek* vagy *fejnek* is nevezzük. Itt valójában egy „kiterjesztett Horn-klóz” látható, ugyanis egy Horn-klóz csak egyetlen  $Y$  tagot enged meg a jobb oldalon. A fenti kiterjesztett formát, azonban bármikor definiálhatjuk  $m$  darab Horn-klóz segítségével.

---

### **(39) OKBC**

Az Open Knowledge Base Connectivity rövidítése. Ez egy szabvány kommunikációs protokoll, melyet abból a célból dolgoztak ki, hogy különböző fejlesztések eredményeképp létrejövő tudásbázisok ezen szabvány alapján képesek legyenek egymással kommunikálni. Lényegében itt vannak leírva a keret alapú tudás-reprezentációk alapelemeinek a ma is használatos nevei (mint pl. a Class, Thing, Subclass-of, Slot, Cardinality stb.)

### **(40) KIF**

A Knowledge Interchange Format rövidítése. Ez egy szabvány adatcsere formátum, aminek segítségével különböző fejlesztések eredményeképp létrejövő tudásbázisok egymás – különböző formátumban tárolt – adatait le tudják kérni. Elsősorban automatikus folyamatok adatcseréjének formátumának szánták, azaz nem arra való, hogy egy élő személy ebben a formátumban kapja meg az általa kért adatokat; nem az emberek számára érthetőség, hanem a számítógéppel könnyen beolvashatóság és értelmezhetőség volt az elsődleges tervezési szempont. Ez a formátum képes leírni minden elsőrendű logikai összefüggést (azaz tulajdonképpen az elsőrendű logikai formulák egy alternatív szintaxisa, mely számítógéppel is könnyen elemezhető).

### **(41) LISP**

Egy programozási nyelv, melyet eredetileg a mesterséges intelligencia kutatásaihoz fejlesztettek ki. A List Processing szóösszetétel rövidítése; e két szó mind a nyelv szintaxisát, mind működését jól jellemzi. Ugyan első változatát 1958-ban specifikálták (és így a ma is használt nyelvek közül csak a Fortran régebbi nála), mégis ma is – viszonylag széles körben használt, olyan területeken is, mint a webfejlesztés. Tudásalapú rendszerekkel gyakran LISP szintaxisú parancsokkal lehet kommunikálni – ennek egy oka lehet, hogy ezek a parancsok könnyen elemezhetők. További érv, hogy a KIF szabvány is a LISP szintaxisának szellemében készült.

### **(42) Szemantikus háló**

A szemantikus háló egy gráfot jelent, melyben a fogalmak, objektumok a gráf csúcsai, a relációk pedig a köztük lévő élek. A szemantikus háló kísérlet arra, hogy modellezze az emberi gondolkodás asszociatív képességét, amelyet az ember oly sikeresen alkalmaz, amikor analógiákban gondolkodik, metaforákban beszél. A szemantikus hálóban létre lehet hozni ilyen asszociatív kapcsolatokat a fogalmak közvetlen összekapcsolásával, vagy legalábbis közeli elhelyezésével.

## **V.1.2. Bizonyítások tulajdonságai**

### **(1) Helyesség**

Egy levezetési rendszer helyes, ha csak olyan formulákat vezethetünk le, melyek az axiómák logikai következményei. Általában a helyesség a bizonyítási módszereknél alapkövetelmény.

---

## (2) Teljesség

Ha a bizonyítási módszerünk teljes, akkor teljesül a következő: Ha egy állítás igaz, akkor azt be is tudja bizonyítani. Az elsőrendű logikára még van teljességi tétel, de a másodrendűre már nincs.

## (3) Kiszámíthatóság

Az alábbi definíciók inkább a gyakorlati oldalról próbálják megmagyarázni a fogalmakat:

Egy  $L$  nyelv kiszámítható/eldönthető, ha az összes,  $L$ -ben megfogalmazott igaz állításról be lehet bizonyítani, hogy igaz, és az összes hamis állításról, hogy hamis.

Egy  $L$  nyelv félig eldönthető, ha minden, az  $L$  nyelven megfogalmazott igaz állításról be lehet bizonyítani, hogy igaz; viszont a hamis állításokról nem tudjuk feltétlenül bebizonyítani, hogy hamisak.

Egy  $L$  nyelv eldönthetetlen, ha van olyan, az  $L$  nyelven megfogalmazott igaz állítás, melyről nem tudjuk bebizonyítani, hogy igaz; továbbá van olyan hamis állítás is, melyről nem tudjuk bebizonyítani, hogy hamis.

Például:

- a leíró logikák közül SHIQ és SHION (és minden alattuk) eldönthető;
- az elsőrendű logika félig eldönthető;
- a másodrendű logika eldönthetetlen.

---

## V.2. A szakképzettségben elvárt kompetenciák köre

A szakképzettségben elvárt kompetenciák köre BSc szinten:

- Üzleti folyamatok megértése, elemzése, a végrehajtást segítő szoftveralkalmazások tervezési munkáinak elvégzése, egyszerűbb programozási feladatok végrehajtása
- Rendszerfejlesztési elvek és módszerek ismerete, alkalmazásának készsége, fejlesztőeszköz használata (üzleti modellezés és/vagy számítógéppel támogatott fejlesztés eszközei)
- Adatbázisok tervezésével, létrehozásával és menedzselésével kapcsolatos feladatok ellátása
- Gazdasági alkalmazások adaptációja, az IT-alkalmazások bevezetéséhez szükséges szervezeti változtatások kezdeményezése, együttműködés a végrehajtásban
- Megoldási változatok készítése gazdasági problémákra, az informatikai támogatás és/vagy fejlesztés kezdeményezése, igény szerinti végrehajtása az üzleti és informatikai szakemberekkel együttműködve, felhasználva a leghatékonyabb IT-megoldásokat.
- A szervezet informatikai egységének menedzselése, a működtetési kockázatok kezelése, kisebb fejlesztési és üzemeltetési projektek tervezése és irányítása, együttműködés informatikai feladatok outsourcing megoldásaiban és az auditálásban.
- Gazdasági alkalmazások működtetése, felhasználói szolgáltatások ellátása (operatív, menedzsment-szintű és felsővezetői információigények meghatározása és kielégítése, vállalatirányítási és döntéstámogató rendszerek használata, kliens-szerver architektúrák és egyéb hálózati környezetek adat- és rendszermegosztási feladatainak ellátása)
- Szakértői rendszerek, intelligens megoldások ismerete, alkalmazása

A szakképzettségben elvárt kompetenciák köre MSc szinten:

- képesség és készség a modern információrendszerek szerepének és jelentőségének a megértésére, az üzleti problémák felismerésére, a problémater és a megoldás módjának a meghatározására, az üzleti folyamatok mélyreható elemzésére,
- rendszerfejlesztési módszertanok és alkalmazás-adaptációs technikák, számítógéppel támogatott megoldások részletes ismerete, készség az új megoldások befogadására és alkalmazására
- absztrakción alapuló vállalatmodellezési feladatok végrehajtása (üzleti és folyamat modell, adat-/objektum-modell, adatbázis-tervezés)

- 
- az üzletmenet-folytonosságot garantáló informatikai biztonsági megoldások tervezéséhez és megvalósításához szükséges ismeretek és készségek
  - az informatikai stratégia kialakításához, az alkalmazások tervezéséhez és menedzseléséhez szükséges ismeretek és készségek
  - a különböző információtechnológiai megoldások ismerete, az IT üzleti/szervezeti hatásának a kezelése, az üzleti alkalmazások működtetésének, a felhasználói igényekhez való folyamatos illesztésének, a hatékony működtetésnek és a változások menedzselésének a végrehajtása
  - A képzés szakirányának megfelelő területen az alapvető gyakorlati módszerek és megoldások, informatikai projektirányítás, valamint az alapvető kutatási irányok mélyreható ismerete, a kutatás-fejlesztéshez szükséges készségek elsajátítása.



### V.3. Az alapképzési és mester szak tanterve

#### Az alapképzési szak (BSc) tanterve

I. félév	Típus	Kredit	Idősáv	II. félév	Típus	Kredit	Idősáv
Analízis	TT	5	1e+1gy	Lineáris algebra	TT	5	1e+1gy
Statisztika I.	TT	3	1e+1gy	Jog	GTH	5	1e+1gy
Közgazdaságtan	TT	7	1e+1gy	Vállalatgazdaságtan	GTH	5	1e+1gy
Gazdaságinformatika alapjai	RT	5	1e+1gy	Számítógép architektúra	RT	5	1e+1gy
Szabadon választható	SZ	10	4	Számítógép-hálózatok	RT	5	1e+1gy
				Szabadon választható	SZ	5	1e+1gy

III. félév	Típus	Kredit	Idősáv	IV. félév	Típus	Kredit	Idősáv
Információ-menedzsment	IR	5	1e+1gy	Operációkutatás	TT	5	1e+1gy
Valószínűség-számítás	TT	5	1e+1gy	Statisztika II.	TT	5	1e+1gy
Vezetés-szervezés	GTH	5	1e+1gy	Számítástudomány	TT	5	1e+1gy
Operációs rendszerek	RT	5	1e+1gy	Adatbázis rendszerek	SZT	5	1e+1gy
Szoftver-technológia I.	SZT	5	1e+1gy	Szoftver-technológia II.	SZT	5	1e+1gy
Szabadon választható	SZ	5	1e+2gy	Szabadon választható	SZ	5	1e+1gy

V. félév	Típus	Kredit	Idősáv	VI. félév	Típus	Kredit	Idősáv
Informatikai rendszerek fejlesztése	IR	5	1e+1gy	Integrált rendszerek fejlesztése	IR	5	1e+1gy
Vállalati pénzügyek	GTH	5	1e+1gy	Emberierőforrás- gazdálkodás	GTH	5	1e+1gy
Számvitel	GTH	5	1e+1gy	Menedzsment kontroll (Controlling)	GTH	5	1e+1gy
Modellezés	DIFF	5	1e+1gy	Médiagazdaságtan	DIFF	5	1e+1gy
Tudásmenedzsment	DIFF	5	1e+1gy	Infrastruktúra menedzsment	DIFF	5	1e+1gy
Szabadon választható	SZ	5	1e+1gy	Diplomamunka	DIP	5	0e+2gy

## Az mesterképzési szak (MSc) tanterve

I. félév	Típus	Kredit	Tanóra	II. félév	Típus	Kredit	Tanóra
Számítástudomány közgazdasági alkalmazásokkal	TT	4	2/2/0/v	Adatbányászat és adattárház	SZT	4	2/0/2/v
Operációkutatási modellek	TT	4	2/2/0/v	Menedzsment kontroll	GH	4	2/2/0/v
Stratégiai Menedzsment	GH	4	2/2/0/v	Stratégiai és szervezeti modellek	GH	4	2/2/0/v
Számvitel vezetőknek	GH	4	2/2/0/v	IT stratégia és menedzsment	SZT	4	2/2/0/v
Hálózati technológiák I.	SZT	4	2/0/2/v	Hálózati technológiák II.	SZT	4	2/0/2/v
Többváltozós Statisztika	TT	4	2/0/2/v	Hálózati biztonság	SZT	4	2/2/0/v
Szoftver engineering	SZT	3	2/0/1/v	Rendszerfejlesztés és modellezés	SZT	3	2/1/0/v
Szabadon választható		3	3	Szabadon választható		3	3
<b>Összesen</b>		30	30	<b>Összesen</b>		30	30
III. félév	Típus	Kredit	Tanóra	IV. félév	Típus	Kredit	Tanóra
Szakirány tárgy	DIFF	3	3	Szakirány tárgy	DIFF	3	3
Szakirány tárgy	DIFF	3	3	Szakirány tárgy	DIFF	3	3
Szakirány tárgy	DIFF	3	3	Szakirány tárgy	DIFF	3	3
Szakirány Kötelezően választható	DIFF	3	3	Szakirány Kötelezően választható	DIFF	3	3
Szakirány Szabadon választható	DIFF	3	3	Szakirány Szabadon választható	DIFF	3	3
Diplomakurzus		15	4	Diplomakurzus		15	4
<b>Összesen</b>		30	19	<b>Összesen</b>		30	19

<b>Üzleti Informatika szakirány</b>							
III. félév	Típus	Kredit	Tanóra	IV. félév	Típus	Kredit	Tanóra
Alkalmazási integráció és integrált rendszerek	DIFF	3	2/0/1/v	IT Audit	DIFF	3	2/1/0/v
Üzleti intelligencia	DIFF	3	2/0/1/v	IT szolgáltatások menedzsmentje	DIFF	3	2/1/0/v
Folyamat és workflow menedzsment	DIFF	3	2/1/0/v	Tudásmenedzsment	DIFF	3	2/1/0/v
<b>Köszolgálati szakirány</b>							
III. félév	Típus	Kredit	Tanóra	IV. félév	Típus	Kredit	Tanóra
Nagy közigazgatási rendszerek menedzsmentje	DIFF	3	2/1/0/v	Belső ellenőrzés	DIFF	3	2/1/0/v
Informatikai jog	DIFF	3	3/0/0/v	IT szolgáltatások menedzsment a közigazgatásban	DIFF	3	2/1/0/v
Folyamat és workflow menedzsment	DIFF	3	2/1/0/v	E-government	DIFF	3	3/0/0/v
<b>Modellezési szakirány</b>							
III. félév	Típus	Kredit	Tanóra	IV. félév	Típus	Kredit	Tanóra
Játékelmélet	DIFF	3	2/1/0/v	Mesterséges intelligencia módszerei	DIFF	3	2/0/1/v
Makrogazdasági modellezés	DIFF	3	2/1/0/v	E-business üzleti modelljei	DIFF	3	2/1/0/v
Sztochasztikus rendszerek modellezése	DIFF	3	2/1/0/v	Banki és biztosítási modellek	DIFF	3	3/0/0/v

## V.4. Teszt eredmények értékelő lapja

A(z) Adatbázisok teszt eredménye

Az alábbi kérdésekben ismereteid bővítése szükséges

Témakör	Tananyag hivatkozás	Válaszadás ideje
1.1.2 Az Access indítása, az adatbázis-ablak felépítése és tulajdonságai	Az Access indítása, az adatbázis-ablak felépítése és tulajdonságai	2007-01-24 14:08:43.0
1.1.3. Adatbázis-összetevők	Adatbázis-összetevők	2007-01-24 14:08:45.0
1.1.4 Adatbázis létrehozása	Adatbázis létrehozása	2007-01-24 14:08:46.0
1.1.5 Tábla létrehozása	Tábla létrehozása	2007-01-24 14:08:47.0
1.1.6 A táblaterv módosítása	A táblaterv módosítása	2007-01-24 14:08:49.0
varázsló	Az Access általános jellemzői	2007-01-24 14:09:02.0
1.2. Adat- és mezőtulajdonságok használata	Adat- és mezőtulajdonságok használata	2007-01-24 14:09:04.0
1.3.1 Táblakapcsolatok áttekintése és kapcsolatok létrehozása	Táblakapcsolatok áttekintése és kapcsolatok létrehozása	2007-01-24 14:09:07.0
1.3.2 Hivatkozás-integritás, kaszkádolt művelet-végrehajtás és az illesztés típus beállításai.	Hivatkozás-integritás, kaszkádolt művelet-végrehajtás és az illesztés típus beállításai	2007-01-24 14:09:09.0
1.3.4 Adatbázis-elemző eszközök	Adatbázis-elemző eszközök	2007-01-24 14:09:12.0
1.4.1 Szűrők és lekérdezések áttekintése	Szűrők és lekérdezések áttekintése	2007-01-24 14:09:20.0
1.4.2 Összehasonlító operátorok használata	Összehasonlító operátorok használata	2007-01-24 14:09:22.0
1.4.3 Választó lekérdezés létrehozása és módosítása	Választó lekérdezés létrehozása és módosítása	2007-01-24 14:09:24.0
1.4.5 Számítások végrehajtása és aggregálás lekérdezésekben	Számítások végrehajtása és aggregálás lekérdezésekben	2007-01-24 14:09:27.0

Az alábbi kérdésekre jól válaszoltál

<b>Témakör</b>	<b>Tananyag hivatkozás</b>	<b>Válaszadás ideje</b>
1.1. Adatbázisok és táblák létrehozása	Adatbázisok és táblák létrehozása	2007-01-24 14:08:35.0
1.1.1 Az Access általános jellemzői	Az Access általános jellemzői	2007-01-24 14:08:40.0
1.1.1 Az Access általános jellemzői	Az Access általános jellemzői	2007-01-24 14:08:57.0
adatbázis	Az Access általános jellemzői	2007-01-24 14:08:59.0
rekord	Az Access általános jellemzői	2007-01-24 14:09:01.0
1.3. Táblák összekapcsolása, összekapcsolt táblák használata	Táblák összekapcsolása, összekapcsolt táblák használata	2007-01-24 14:09:06.0
1.3.3 Keresőoszlop létrehozása és használata adatbeírásra	Keresőoszlop létrehozása és használata adatbeírásra	2007-01-24 14:09:11.0
1.3.3 Keresőoszlop létrehozása és használata adatbeírásra	Keresőoszlop létrehozása és használata adatbeírásra	2007-01-24 14:09:14.0
1.4. Szűrők és lekérdezések használata	Szűrők és lekérdezések használata	2007-01-24 14:09:18.0
1.4.4 Feltételek használata választó lekérdezésben	Feltételek használata választó lekérdezésben	2007-01-24 14:09:25.0
1.4.6 A lekérdezés eredményének finomítása	A lekérdezés eredményének finomítása	2007-01-24 14:09:28.0
1.4.7 Kimutatás és keresztáblás lekérdezés létrehozása, módosítása	Kimutatás és keresztáblás lekérdezés létrehozása, módosítása	2007-01-24 14:09:30.0
1.4.8 Akció lekérdezések létrehozása és módosítása	Akció lekérdezések létrehozása és módosítása	2007-01-24 14:09:31.0
frissítő lekérdezés	Akció lekérdezések létrehozása és módosítása	2007-01-24 14:09:34.0

---

## VI. Hivatkozások

- Babbie, E. (1996): *A társadalomtudományi kutatás gyakorlata*, Balassi Kiadó, Budapest
- Balaton, K. – Dobák, M. (1991): *Mennyiségi és minőségi módszerek az empirikus szervezatkutatásban*, megjelent: Antal-Mokos, Z. – Drótos, Gy. – Kovács, S. (szerk.): *Módszertani gyűjtemény a vezetés és szervezés tárgyhoz*, Aula Kiadó, Budapest
- Bechhofer, S. – Horrocks, I. – Goble, C. – Stevens, R. (2001): *OilEd: a Reason-able Ontology Editor for the Semantic Web*, in: Proceedings of KI2001, Joint German/Austrian Conference on Artificial Intelligence, Springer-Verlag, Vienna, Austria, September 19-21, pp. 396-408.
- Bernaras, A. – Laresgioti, I. – Corera, J. (1996): *Building and Reusing Ontologies for Electrical Network Applications*, in: Proceedings of the European Conference on Artificial Intelligence (ECAI'96), pp. 298-303.
- Berners-Lee, T. – Hendler, J. – Lassila, O. (2001): *The Semantic Web*, Scientific American, May
- Binet, A., Simon, and Th. (1905) *Méthodes nouvelles pour le diagnostic du niveau intellectuel des anormaux*. Année psychol., 1905, 11, pp.191-244.
- Boley, H. (2000): *The Relational-Functional Markup Language RFML Draft Specification*, <http://www.relfun.org/rfml/rfml.ps>, (letöltve: 2005. szeptember 30.)
- Borst, W. N. (1997): *Construction of Engineering Ontologies*, PhD Thesis, University of Twente, Enschede, The Netherlands – Centre for Telematica and Information Technology
- Brickley, D. – Guha, R. (eds.), (2004): *RDF Vocabulary Description Language 1.0: RDF Schema*, W3C Recommendation, <http://www.w3.org/TR/rdf-schema/>, (letöltve: 2005. március 18.)
- Chalupsky, H. – MacGregor, R. M. – Russ, T. (2003): *PowerLoom Manual*, <http://www.isi.edu/isd/LOOM/PowerLoom/documentation/manual/manual.pdf>, (letöltve: 2005. szeptember 14.)
- Chandrasekaran, B – Josephson, J.R. – Benjamins, V.R. (1999): *What are ontologies, and why do we need them?* IEEE Intelligent Systems, Vol.14. No. 1, pp. 20–26.
- Clark, P. – Porter, B. (2004): *KM - The Knowledge Machine 2.0: User Manual*, <http://www.cs.utexas.edu/users/mfkb/km/userman.pdf>, (letöltve: 2005. szeptember 30.)
- Corbett, D. (2001): *A Method for Reasoning with Ontologies Represented as Conceptual Graphs*. In proceedings of Australian Joint Conference on Artificial Intelligence, pp. 130-141.

- 
- Corcho, O. – Gómez-Pérez, A. (2000): *Evaluating knowledge representation and reasoning capabilities of ontology specification languages*, in: Proceedings of the ECAI 2000 Workshop on Applications of Ontologies and Problem-Solving Methods, Berlin
- Corcho, O. – Fernández-López, M. – Gómez-Pérez, A. (2003): *Methodologies, tools and languages for building ontologies. Where is their meeting point?*, in: Data & Knowledge Engineering, Vol. 46, pp.41-64.
- Cyc (2002): *The Syntax of CycL*, <http://www.cyc.com/cycdoc/ref/cycl-syntax.html>, (letöltve: 2005. május 30.)
- Davenport, T. H. – Prusak, L. (2001): *Tudásmenedzsment*, Kossuth Kiadó, Budapest. Fordította: Andó Éva,
- Davies, J. – Duke, A. – Stonkus, A. (2002): *OntoShare: Using Ontologies for Knowledge Sharing*, in Proceedings of the WWW2002 Semantic Web Workshop, 11th International WWW Conference, Hawaii, USA
- Denny, M. (2002): *Ontology Building: A Survey of Editing Tools*, <http://www.xml.com/pub/a/2002/11/06/ontologies.html>, (letöltve: 2004. november 25.)
- Doignon, J.P. – Falmagne, J.C. (1985). *Spaces for the Assessment of Knowledge*. International Journal of Man-Machine Studies Vol. 23 No. 2., pp.175-196.
- Domingue, J. – Motta, E. – Corcho, O. (1999): *Knowledge Modelling in WebOnto and OCML, A User Guide*, <http://kmi.open.ac.uk/projects/ocml/ocml-webonto-guide.zip>, (letöltve: 2005. szeptember 18.)
- Duineveld, A.J. – Stoter, R. – Weiden, M.R. – Kenepa, B. – Benjamins, V.R. (1999): *Wonder Tools? A comparative study of ontological engineering tools*, in: Proceedings of the 12th International Workshop on Knowledge Acquisition, Modeling and Management (KAW'99), Banff, Canada
- Eisenhardt, K. M. (1989): *Building Theories from Case Study Research*, Academy of Management Review, Vol.14 No 4 pp. 532-550.
- Falmagne, J.C. – Doignon, J.P. – Koppen, M. – Villano, M. – Johannesen L. (1990). *Introduction to Knowledge Spaces - How to Build, Test, and Search Them*. Psychological Review Vol. 97 No. 2., pp.201-224
- Farquhar, A. – Fikes, R. – Rice, J. (1996): *The Ontolingua Server: A Tool for Collaborative Ontology Construction*, in: Proceedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop, Alberta, Canada
- Fensel, D. (1998): *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Springer-Verlag, Berlin, 2001, ISBN 3-540-41602-1, p. 112.

- 
- Fensel, D. – van Harmelen, F. – Davies, J. (2003): *Towards the Semantic Web – Ontology driven knowledge management*, John Wiley & Sons Ltd, West Sussex, England
- Fernandez, M. – Gómez-Pérez, A. – Juristo, N. (1997): *METHONTOLOGY: From Ontological Art Towards Ontological Engineering*, AAAI-97 Spring Symposium on Ontological Engineering, Stanford University, March 24-26th.
- Fernández-López, M. (1999): *Overview of Methodologies for Building Ontologies*, in: Benjamins, V. R. (ed.): IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends, CEUR Publications, Amsterdam
- Fernández López, M. – Gómez-Pérez, A. – Pazos Sierra, J. (1999): *Building a Chemical Ontology Using Methontology and the Ontology Design Environment*, IEEE Intelligent Systems, Vol. 14, No. 1, pp. 37-46.
- Füstös L. – Meszéna Gy. – Simonné Mosolygó N. (1986): *A sokváltozós adatelemzés statisztikai módszerei*, Akadémiai Kiadó, Budapest
- Gershon, R.C. (1992) Test Anxiety and Item Order: New Concerns for Item Response Theory. In M Wilson (ed.) *Objective Measurement: Theory into Practice*. Vol. 1. Ablex, Norwood NJ
- Gómez-Pérez, A. (1994): *Some Ideas and Examples to Evaluate Ontologies*, Technical Report KSL 94-65., Knowledge Systems Laboratory, Stanford University.
- Gómez-Pérez, A. – Fernández-López, M. – De Vicente, A. J. (1996): *Towards a Method to Conceptualize Domain Ontologies*, ECAI-96 Workshop on Ontological Engineering, Budapestú
- Gómez-Pérez, A. (1999): *Evaluation of taxonomic knowledge in ontologies and knowledge bases*. In proceedings of the 12th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Alberta, Canada.
- Gómez-Pérez, A. – Corcho, O. (2002): *Ontology Languages for the Semantic Web*, IEEE Intelligent Systems, Vol. 17, No. 1, pp. 54-60.
- Gómez-Pérez, A. (2004): *Ontology Evaluation*. In Staab, S. and Studer, R. (eds) *Handbook on Ontologies*, Springer Verlag, Heidelberg, pp. 251-274
- Gruber, T. R. (1993): *A Translation Approach to Portable Ontology Specifications*, Knowledge Acquisition, Vol. 5, No.2, pp. 199-220.
- Grüninger, M. – Fox, M.S. (1994): *The Role of Competency Questions in Enterprise Engineering*, IFIP WG5.7 Workshop on Benchmarking - Theory and Practice, Trondheim, Norway.
- Grüninger, M. – Fox, M. S. (1995): *Methodology for the Design and Evaluation of Ontologies*, IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, August 19-20th.



- 
- Guarino, N. – Carrara, M. – Giaretta, P. (1995): *Ontologies and Knowledge Bases: Towards a Terminological Clarification*, in: Mars, N. (ed.): *Towards Very Large Knowledge Bases*, Knowledge Building and Knowledge Sharing, IOS Press, Amsterdam, pp. 25-32.
- Guarino, N. (1998): *Formal Ontology and Information Systems*, in: Guarino, N. (ed.): *Formal Ontology in Information Systems*, Proceedings of FOIS'98, IOS Press, Amsterdam, pp. 3-15.
- Guarino, N. – Welty, C. (2004). *An Overview of OntoClean*. In Staab, S. and Studer, R. (eds) *Handbook on Ontologies*, Springer Verlag, Heidelberg. pp. 151-172.
- van Harmelen, F. – Patel-Schneider, P. – Horrocks, I. (eds), (2001): *Reference description of the DAML+OIL ontology markup language*, <http://www.daml.org/2001>, (letöltve: 2005. június 24.)
- Hart, C. (1998): *Doing a Literature Review – Releasing the Social Science Research Imagination*, Sage Publications, London
- Hayes, P. (1979): *The logic of frames*, in: Metzger, D. (ed.): *Frame Conceptions and Text Understanding*, de Gruyter, Berlin
- Henning, G. (1987) *A guide to language testing*. Cambridge, Mass.: Newbury House
- Horrocks, I. – Patel-Schneider, P. F. – Boley, H. – Tabet, S. – Grosz, B. – Dean, M. (2004): *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, W3C Member Submission, <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>, (letöltve: 2005. október 17.)
- ISX Corporation (1991): *Loom Users' Guide*, <http://www.isi.edu/isd/LOOM/documentation/usersguide1.4.pdf>, (letöltve: 2005. szeptember 14.)
- Karp, P. D. – Chaudry, V. K. – Thomere, J. (1999): *XOL: An XML-Based Ontology Exchange Language*, <http://www.ai.sri.com/pkarp/xol/xol.html> (letöltve: 2005. április 10.)
- Kifer, M. – Lausen, G. – Wu, J. (1995): *Logical Foundations of Object-Oriented and Frame-Based Languages*, Journal of the ACM, Vol. 42., No. 4., pp. 741-843.
- Kismihók G. – Vas R.: *Ontology based Adaptive Examination System in E-Learning Environment*, In Proceedings of 28th International Conference on Information Technology Interfaces, Cavtat, Croatia pp. 77-82
- Klimkó G. (2001): *A szervezeti tudás feltérképezése*, Phd disszertáció, BKÁE, Információrendszerek tanszék, Budapest
- Kovács I. V. (2004): *A lisszaboni folyamat és az oktatás: Barcelonától az első időközi jelentés elfogadásáig (2002-2004 március)*, Kézirat, Társi, Budapest
- Linacre, J. M. (2000): *Computer-adaptive testing: A methodology whose time has come*, in Chae, S. - Kang, U. – Jeon, E. – Linacre, J. M. (eds.): *Development of Computerized Middle School Achievement*

- 
- Tests, MESA Research Memorandum No. 69., Komesa Press, Seoul, South Korea
- Luke, S. – Heflin, J. (2000): *Shoe 1.01 Specification*, <http://www.cs.umd.edu/projects/plus/SHOE/spec.html>, (letöltve: 2005. szeptember 14.)
- Maedche, A. (2001) Tutorial on Development and Application of Ontologies, in: Proceedings of ECML/PKDD
- McCarthy, J. (1980): *Circumscription – A Form of Nonmonotonic Reasoning*, in: Artificial Intelligence, Vol.13, No. 1, pp. 27-39.
- McGuinness D. – van Harmelen F. (eds.) (2004): *OWL Web Ontology Language Overview*, W3C Recommendation <http://www.w3.org/TR/owl-features/>, (letöltve: 2005. március 18.)
- MEO, (2005a): *Ontológia építési módszertanok összehasonlítása*, MEO Projekt, Scriptum Informatika Rt. [http://ontologia.hu/document/paper/Ontologiaepitesi\\_ModszerTanok\\_tanulmánya.pdf](http://ontologia.hu/document/paper/Ontologiaepitesi_ModszerTanok_tanulmánya.pdf), (letöltve: 2005. november 05.)
- MEO, (2005b): *Ontológia-építő nyelvek értékelése, elemző összehasonlítása*, MEO Projekt, Scriptum Informatika Rt. [http://ontologia.hu/document/proj\\_doc/ontologia\\_epito\\_nyelv/](http://ontologia.hu/document/proj_doc/ontologia_epito_nyelv/), (letöltve: 2005. november 05.)
- MEO (2005c): *Következtető rendszerek részletes vizsgálata*, MEO Projekt, Scriptum Informatika Rt. <http://ontologia.hu/document/paper/kovetkeztetorendszerek/>, (letöltve: 2005. november 05.)
- Minsky, M. (1975): *A framework for representing knowledge*, in: Winston, P. H. (ed.): *The Psychology of Computer Vision*, McGraw-Hill, New York
- Molnár B. (1999): „Ismeretszerzés”, megjelent: Gábor, A. (szerk.): *Információmenedzsment*, Aula Kiadó, Budapest, 206-212. oldalak
- Molnár B. (2003): *CommonKADS módszertan*, MTA Információtechnológiai Alapítvány, Budapest
- Neches, R. – Fikes, R. E. – Finin, T. – Gruber, T. R. – Senator, T. – Swartout, W.R. (1991): *Enabling technology for knowledge sharing*, AI Magazine Vol.12, No. 3, pp. 36–56.
- O’Leary, D.E. (1998): *Using AI in Knowledge Management: Knowledge Bases and Ontologies*, IEEE Intelligent Systems Vol. 13, pp. 334-39
- OntoEdit (2003): *How to Work with OntoEdit*, [http://www.ontoprise.de/documents/tutorial\\_ontoedit.pdf](http://www.ontoprise.de/documents/tutorial_ontoedit.pdf) (letöltve: 2005. március 21.)
- OntoWeb (2002a): *A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies*, OntoWeb Technical Report, 2002. [http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/OntoWeb\\_Del\\_1-4.pdf](http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/OntoWeb_Del_1-4.pdf), (letöltve: 2004. szeptember 15.)
- OntoWeb (2002b): *A survey on ontology tools*, OntoWeb Technical Report, 2002. <http://www.aifb.uni->
-

---

[karlsruhe.de/WBS/ysu/publications/OntoWeb\\_Del\\_1-3.pdf](http://karlsruhe.de/WBS/ysu/publications/OntoWeb_Del_1-3.pdf), (letöltve: 2004. szeptember 15.)

- Patel-Schneider, P. F. (2004): *A Proposal for a SWRL Extension to First-Order Logic*, <http://www.daml.org/2004/11/fof/proposal>, (letöltve: 2005. szeptember 30.)
- Rasch, G. (1992) *Probabilistic Models for Some Intelligence and Attainment Tests*. Copenhagen and Chicago: MESA Press.
- Reckase, M.D. (1974) *An interactive computer program for tailored testing based on the one-parameter logistic model*. Behavior Research Methods and Instrumentation Vol. 6 No. 2. pp. 208-212
- Roskam, E. E. – Jansen, P. G. W. (1984) *A new derivation of the Rasch model*. In Degreef E. – van Buggenhaut J. (eds) Trends in Mathematical Psychology. Amsterdam, pp. 293-307
- Sántáné-Tóth E. (2001): *Ontológia – oktatási segédlet rövidített változata*, ELTE
- Schaeffer, S. A. – Hwang, C. H. – de Haan, J. – Schubert, L. K. (1993): *EPILOG: The Computational System for Episodic Logic (User's Guide)*, Technical Report, Department of Computational Science, University of Alberta, Edmonton, Alberta
- Schreiber, G. – Wielinga, B. – Jansweijer, W. (1995): *The Kactus View on the 'O' Word*; Workshop on Basic Ontological Issues in Knowledge Sharing/IJCAI95; Montreal, Canada
- Schreiber, A. Th. – Akkermans, J. M. – Anjewierden, A. A. – Dehoog, R. – Rhabbold, N. R. – Velde, W. V. D. – Wielinga, B. J. (1998): *Knowledge Engineering and Management - The CommonKADS Methodology*. University of Amsterdam
- Shapiro, S. C. (2004): *SNePS 2.6.1 User's Manual*, <http://www.cse.buffalo.edu/sneps/Manuals/manual261.pdf>, (letöltve: 2005. szeptember 30.)
- Shadbolt, N. – Motta, E. – Rouge, A. (1993): *Constructing Knowledge-Based Systems*, IEEE Software, Vol. 10, No. 6, pp. 34-39.
- Smith, B. (2002): *Ontology and Information Systems*, [http://www.ontology.buffalo.edu/ontology\(PIC\).pdf](http://www.ontology.buffalo.edu/ontology(PIC).pdf), (letöltve: 2005. november 25.)
- Smith, M. – Welty, C. – McGuinness, D. (eds.) (2004): *OWL Web Ontology Language Guide*, <http://www.w3.org/TR/owl-guide/>, (letöltve: 2005. március 18.)
- Sowa, J. F. (2000): *Knowledge Representation: Logical, Philosophical and Computational Foundations*, Brooks Cole Publishing Co.
- Staab, S. – Studer, R. – Schnurr, H. P. – Sure, Y. (2001): *Knowledge Processes and Ontologies*, IEEE Intelligent Systems Vol. 16, No. 1, pp. 26-34.

- 
- Studer, R. – Benjamins, V. R. – Fensel, D. (1998): *Knowledge Engineering: Principles and Methods*, Data and Knowledge Engineering, Vol. 25, No. 1-2, pp. 161-197.
- Sure, Y. – Staab, S. – Angele, J. (2002): *Ontoedit: Guiding ontology development by methodology and inferencing*, in: International Conference on Ontologies, Databases and Applications of Semantics (ODBASE'02), Irvine, USA
- Swartout, W. R. – Patil, R. – Knight, K. – Russ, T. (1997): *Towards Distributed Use of Large-Scale Ontologies*, AAAI-97 Spring Symposium on Ontological Engineering, Stanford University.
- Szabó, I. (2006): *Educational ontology for transparency and student mobility between universities*, In Proceedings of 28th International Conference on Information Technology Interfaces, Cavtat, Croatia pp. 301-306
- Szabó, I. (2006a) *The Implementation of the Educational Ontology*, In Fehér, P. (ed.) Proceedings of the 7th European Conference on Knowledge Management, Corvinus University of Budapest, Hungary, 4-5 September 2006
- Thissen, D. - Mislevy, R.J. (1990). *Testing Algorithms*. In: Wainer, H. Computerized Adaptive Testing: A Primer. Lawrence Erlbaum Associates, Publishers, New Jersey, pp. 103-135.
- Ungváry R. (2004): *Tezaurusz és ontológia, avagy a fogalmi ismertetőjegek generikus öröklődésének formalizálása* In Tudományos és műszaki tájékoztatás: könyvtár-és információtudományi szakfolyóirat, ISSN 0041-3917, Vol.51. No.5. pp.175-191.
- Uschold, M. – King, M. (1995): *Towards A Methodology for Building Ontologies*, IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada
- Uschold, M. (1996a): *Converting an Informal Ontology into Ontolingua: Some Experiences*, ECAI-96 Workshop on Ontological Engineering, Budapest, August 13th
- Uschold, M. (1996b): *Building Ontologies: Towards A Unified Methodology*, in: Proceedings of Expert Systems '96, Cambridge, December 16-18th
- Vas, R. – Szabó, I. (2006): *Educational Ontology and Adaptive Knowledge Testing*, NITIM Ph.D. Consortium sponsored by CEMS, 1-4 November, 2006
- Wainer, H. – Mislevy, R.J. (1990). *Item Response Theory, Item Calibration and Proficiency Estimation*. In: Wainer, H., (ed.), Computerized Adaptive Testing: A Primer. Lawrence Erlbaum Associates, Publishers, New Jersey, pp. 65-102.
- Welch, R.E. - Frick, T.W. (1993). *Computerized Adaptive Testing in Instructional Settings*. Educational Technology Research & Development Vol. 41 No. 3, pp.47-62.

- 
- Wielinga, B.J. – Schreiber, A.Th. – Jansweijer, W.H. – Anjewierden, A. – Van Harmelen, F. (1994): *Framework and Formalism for Expressing Ontologies*, KACTUS Project Deliverable DO1b.1, University of Amsterdam.
- Wisdom (2005): *Critical analysis of emerging ontology languages and standards*, Technical Report, [http://dbgroup.unimo.it/wisdom/deliverables/fase\\_1/d1r1.pdf](http://dbgroup.unimo.it/wisdom/deliverables/fase_1/d1r1.pdf), (letöltve: 2005. november 18.)
- Wright, B.D. (1988) *Practical adaptive testing*. Rasch Measurement Transactions Vol. 2 No. 2. p. 21
- Yin, R.K. (1994): *Case Study Research* (2nd edition) Sage Publications, London