

# **Szemantikai alapú jogi tudás- menedzsment technológiák**

PhD Doktori értekezés tézisei  
Pécsi Tudományegyetem  
Állam- és Jogtudományi Kar, Doktori Iskola

Készítette: Kilián Imre

Témavezető: Dr. Balogh Zsolt György PhD.

# Kilián Imre: Szemantikai alapú jogi tudásmenedzsment technológiák

PhD értekezés tézisei

## 1 Problémafelvetés

A jog rég elszakadt a népességtől. Olyan mértékben és olyan sebességgel fejlődik, bonyolódik és többszöröződik, minőségileg, de sajnos mennyiségileg is, hogy nem elvárható többé az átlagos állampolgártól még az sem, hogy legalább nagy vonalakban, és legalább a maga területén tisztában legyen azzal, hogy milyen megkötések vonatkoznak rá, és milyenek nem. A robbanásszerű gyarapodás – a törvénygyár – egyik velejárója (ez nemcsak jogi területen szokott így lenni) – a zűrzavar: már egy-egy törvényen belül is, de egyes törvények között még inkább szaporodnak a hibák, a pontatlanságok, az ütközések, a lefedetlen rések. A jogi átláthatatlansággal és áthatolhatatlansággal szemben kétféle módon lehet fellépni: egyrészt tudatos és tervezett bozótirtással, gyomlálással és egyszerűsítéssel, ami leginkább jogász feladat. Másrészt pedig számítástechnikai eszközökkel, ami számítástechnikai feladat. Bár csak a jogtisztítás lenne az igazán fontos és hatékony, a szerzőnek – az előképzettségénél fogva a második témában, a jogi szakterület számítógépes kezelésében van mit mondania. Ez az értekezés tárgya is.

## 2 Számítógéppel segített jogkezelés (CAJ/Computer Aided Juristics)

A számítógépes jogkezelés tekintetében felismerhető egy folyamat, amely az alkalmazott technológiák fejlődésével jellemezhető. Ezek az ún. mesterséges intelligencia technológiák és a természetes nyelvtechnológiák egyfajta kölcsönhatásával írhatók le. A kezdő technológiák, a szabad szavas keresés (freetext search), és a szóalakelemzés (morfológia) voltak (ez utóbbi értelemszerűen csak a magyarhoz hasonló, ún. todalékoló, vagy agglutinatív nyelvek esetén). Ezt továbbfejlesztették a tezauruszos kérdés kibővítés megoldásával, amely megoldás még szintén a piacra került. Az ezután következő nemzedék a korlátozott (természetes) nyelvtani (sekély-) -elemzéssel (Part Of Speech/POS Tagging), a dokumentumok XML alapú szöveges adatbázisban történő tárolásával működött. Később a pongyola szemantikájú teaurusz helyett esetleg kicsit fejesebb ontológiák alkalmazással, ahol még nem volt igény az ontológiákban ábrázolt szemantikai viszonyok gazdagsága. Ezekben a témákban már lezárt és eredményes kutatások folytak, a konkrét piaci bevezetésről (egyelőre) nincs híradás. A technológiák harmadik nemzedéke a nyelvi mélyelemzéssel és az ontológiák még részletesebb és pontosabb alkalmazásával, ontológiai következtetésekkel függ össze. Ez még erősen kutatási fázisban van, a következő években a technológiák területén esetleg várható komoly áttörés egyben áttörést jelenthet az alkalmazói világban, a piacra kerülő szoftverek világában is. Ez az értekezés tárgya: a szerző, elsősorban a technológiák harmadik nemzedékében végzett kutatásai.

1. A fenti trend emeli a szoftver technológiai színvonalát, de közben növeli a bonyolultságát és ezzel a kutatás és kidolgozás költségeit is.
2. A jogi szakterület választása az említett mesterséges intelligencia- és nyelvtechnológiák szemszögéből is kedvező választás, mert mindegyik érintett szakterületre kölcsönösen megtermékenyítő hatású lehet. Komolyan leegyszerűsíti a megoldandó feladatokat, ha legalább a cél-szakterület és -nyelv szemantikája

(legalább szándékában) pontos, feszes, márpedig a jogi területen legalábbis a szándék és a cél szerint mindenképpen fontos a pontos fogalmazás. Ezen túl, bizonyos korlátok elismerése mellett a technológia alkalmas lehet a jogi nyelv pontatlanságainak, a jogi részeknek, felesleges átfedéseknek és a jogi ütközéseknek (kollízió) a felderítésére is.

### 3 Hipotézis

A jogi feladatok számítógépes kezelésében még hatalmas potenciálok rejlenek: a kiaknázatlan lehetőségek elsősorban a természetes nyelvek feldolgozását, másrészt a mesterséges intelligencia technológiák alkalmazását jelentik. A kiaknázatlan lehetőségek mellett a jog tartalmi, értelemközpontú kezelése terén még igen sok feltáratlan felhasználói igény is létezhet. Ezek a jelen pillanatban ugyan még rejtettek, de ennek legfőbb oka, hogy a járatos technológia efféle műveletekre képtelen. Mindezen technológiai irányok előrehaladásával eddig számítógépes eszközökkel még nem valósított, sőt meg sem közelített feladatosztályok válhatnak megvalósíthatóvá.

### 4 Munkamódszer

A vázolt problémák megoldása több-tudományzakos (interdiszciplináris, de inkább multidiszciplináris) megközelítést igényel: a megoldás valahol a számítógépes nyelvészet, a matematikai logika, az informatika és a jogtudomány közös határterületén mozog. A fent körvonalazott hipotézist a következő munkamódszerrel lehet alátámasztani:

1. Megvizsgáljuk, hogy a mi volt a természetes nyelvek feldolgozási technológiáinak az eddigi fejlődése, mi az, ami „state-of-commerce”, és mi az, ami még csak „state-of-the-art” állapotú. Rámutatunk azokra a pontokra, ahol újszerű, vagy eddig még nem alkalmazott megoldások áttörést vagy legalábbis komoly előrehaladást jelenthetnek.
2. Megvizsgáljuk ugyanezeket a mesterséges intelligencia-technológiákkal kapcsolatban is.
3. Megvizsgáljuk, hogy hasonló kutatási projektek, esetleg termékek esetében milyen eredményeket célszerű átvenni. Itt elsősorban a korábbi kutatások létrehozta szabványok figyelembe vétele és átvétele, ill. egyes, a szabványok kezelésével, ellenőrzésével és minőségbiztosításával összefüggő szoftvertermékek átvétele lehet érdekes.
4. Saját kísérleti tanulmány-szoftverek kifejlesztése olyan esetekben, ahol ez az elméleti indoklásnál egyszerűbbnek tűnik, ill. ahol a szoftver-elem kifejlesztése további más szoftverek fejlesztéséhez nyit kaput (kulcs-szoftverek). Semmiképpen sem cél, és nem is várható el, hogy az értekezés mellékeredményeképpen egy akár csak prototípus állapotú, teljes jogi tudáskezelő szoftver létrejöjjön.
5. Az eddigi pontokban leírt vizsgálatok és kifejlesztett szoftverek meghatároznak egy szoftver felépítményt, és egy műveletkört. Az eredmények kiértékelésével meg kell adni a megvalósítható műveletek körét, és rá kell mutatni azokra a feladatokra is, amely kihívásaira a vázolt felépítmény nem tud választ adni.

### 5 Megoldási javaslatok és eredmények

A megcélzott eredményeket és technológiákat a következő megoldások alkalmazásával lehet elérni és megvalósítani:

1. Az adatok modellezését, leképezéseit, a tudás ábrázolását és a tudásanyagon végezhető következtetéseket csak logikai alapon lehet és érdemes elvégezni. Jogi esetek és normatívák elemzéséhez azonban a klasszikus logika eszköztára már nem

elegendő, legalábbis a modális logikai<sup>1</sup> formalizmusok használatára van szükség. Az alkalmazandó modalitásnak egyszerre kell multimodálisnak és polimodálisnak lennie, vagyis többféle modális operátort és akár többargumentumú operátorokat is használnia kell.

- a. Modális időlogikákra (temporary modality) van szükség az időbeli viszonyok leírására
  - b. Episztemikus és doxasztikus, több cselekvőjű (multi-agent) modális logikákra<sup>2</sup> van szükség a jogi esetek és eljárások leírására.
  - c. Deontikus modális logikákra<sup>3 4</sup> van szükség a normatívák leírására.
2. A kiválasztott logikán következtetési feladatokat kell végrehajtani. Az erre vonatkozó módszerek (pl. az alap/bináris rezolúciós módszer) konkrét értelmezési tartományok (pl. egész számok) esetén nem eléggé hatékonyak, ezért esetleg célmegoldók alkalmazása szükséges. Ennek körében a szerző meghatározta a beágyazott rezolúciós tételbizonyítás fogalmát, amely egy általános rezolúciós tételbizonyítás keretében egyes célmegoldók bekapcsolását teszi lehetővé.
3. Az alkalmazott szoftver eszközök tekintetében úgy véljük, hogy a megcélzott bonyolultság egyértelműen csakis logikai programozási paradigma eszközeivel érhető el. Ez szűkebb értelmezésben jelenti a Prolog programnyelvet,<sup>5</sup> és a köré épült kiterjesztések családját, másrészt, tágabb értelmezésben pedig jelenti a matematikai logikára épülő és építhető szoftver -tervezési és -megvalósítási eszközöket. A Prolog már maga is tételbizonyító, de a kitűzött célokra a saját tételbizonyítási képességei feltehetőleg nem lesznek elegendőek. Viszont a Prolog programozási nyelvcsalád egy sor hasznos kiegészítést és kiterjesztést tartalmaz, és újabb ilyen kiterjesztések létrehozására is könnyen ad lehetőséget:
- a. A Prolog programnyelv típusatlan, de könnyű kiterjeszteni típusfogalommal. A kutatás egyik részfeladatuként a szerző kifejlesztette a ProType Prolog alapú típusdeklarációs résznyelvet, és a hozzá tartozó típusellenőrző programot.
  - b. A szerző kifejlesztette a Prologba beépített visszafelé haladó (backward-chaining) következtetési mechanizmust előre haladó módon (forward chaining) kiterjesztő Contralog csomagot, és a kettőt összekapcsoló PC-LOG (Pro-Contra-LOG) csomagot.<sup>6</sup>
  - c. Változatos feladatokra más célokra is kifejlesztettek már korlát-logikai (CLP) megoldó csomagokat.<sup>7</sup> Szükséges lehet efféle csomagok alkalmazása intelligens jogi alkalmazások területén is.
4. A tágan értelmezett logikai programozási paradigmára építve a kitűzött célokat csak valamiféle egységes ábrázolású és matematikai-logikailag jól megalapozott tudástár

---

<sup>1</sup> Ruzsa Imre: Klasszikus, modális és intenzionális logika [Ru84]

<sup>2</sup> Hendricks, Vincent and Symons, John: Epistemic Logic [HeSy09]

<sup>3</sup> Lokhorst, Gert-Jan: Mally's Deontic Logic [Lo08]

<sup>4</sup> McNamara, Paul: Deontic Logic [Na10]

<sup>5</sup> W.F.Clockshin-C.S.Mellish: Programming in Prolog [CloMe]

<sup>6</sup> Kilián Imre: Contralog: egy előre haladó Prolog motor és alkalmazása ReALIS nyelvi elemzésre [Kil11.1]

<sup>7</sup> Roman Barták: Constraint Programming: in Pursuit of the Holy Gral [Bar99]

formalizmus és eszköz alkalmazásával lehet elérni. Erre a következő megoldási lehetőségek kínálkoznak:

- a. Az UML (Unified Modelling Language) szoftver-tervezési eszköz,<sup>8</sup> és különösen annak az osztálydiagramjai különösen hatékony és jól felfogható grafikus ábrázolásmódot alkalmaznak. Szokásos ellenérv, hogy az UML szemantikája túlságosan pongyola, pontatlan. A szerző a dolgozatban kidolgozta és bemutatta az UML osztálydiagramok szemantikus megalapozását, amivel létrehozta a kapcsolatot az UML és a logikai programozási paradigma között. Mindezek ellenére az UML-t inkább csak, mint grafikai nyelvet használtuk.
- b. A tudástárakat kezelő szoftver eszközök alapmodelljeként a szerző kidolgozta a modell- és ontológiavezérelt szoftver felépítmények fogalmát.<sup>9</sup> Ezeket nevezhetjük kétszintű szoftvereknek is, mert legfőbb jellegzetességük, hogy a konkrét adatok szintjén túl a tárgyadatok modellszintjén is adatként kezelik. Modellszinten az információk modelljét/ontológiáját találjuk, és az azokon végzett különféle műveleteket (mentés, betöltés, szerkesztés, ellenőrzés, stb.). A modellműveletek felhasználói mindenképpen csak képzett szakemberek lehetnek. Az adatszinten található műveletek viszont általánosak: a műveletek végrehajtását a modellszint vezérlni. Az adatszint összekapcsolódhat egy alkalmazói szoftver réteggel, amin keresztül a műveletekhez a végfelhasználó is hozzáférhet.
- c. A modell- és ontológiavezérelt alkalmazások körében a korábban már említett beágyazott rezolúciós elv alkalmazásával a szerző egy UML modellellenőrzőt állított össze és valósított meg, ami a SILK kutatási projektben alkalmazásra is került. Az ellenőrző csomag kifejlesztése az OWL nyelvre már nem szükséges, az efféle algoritmusok (a tabló algoritmus és különféle változatai) már készen állnak.
- d. A megcélzott, hasonló technológiákat a Szemantikus Web nemzetközi kutatási kezdeményezés alkalmaz.<sup>10</sup> A kezdeményezés a DL (Description Logic/leíró logika) alapjain ontológialeíró nyelv-szabványt (OWL) terjesztett elő. A DL logikai formalizmus azért érdekes, mert benne még készíthető tételbizonyító és konzisztencia-ellenőrző algoritmus (az ún. tabló algoritmus) anélkül, hogy a magasabb rendű logikák eldönthetelenségi és megoldhatatlansági elméleti problémáival szembesülnénk. Mindazonáltal rengeteg célontológia, valamint alapozó- vagy csúcsontológia, többek között jogi részontológiák, valamint az ontológiákat szerkesztő, kezelő, ellenőrző szoftverek is készültek már OWL nyelven. A megcélzott feladatokhoz mindenképpen az OWL nyelvre alapuló technológiák alkalmazását,<sup>11</sup> és a rendelkezésre álló ontológiakészítési tapasztalatok, esetleg meglevő csúcsontológiák átvételét látjuk szükségesnek.

---

<sup>8</sup> Rumbaugh-Jacobson-Booch: Unified Modelling Language Reference Manual [UML99]

<sup>9</sup> Kilián Imre: Modellvezérelt szoftverek készítése [Kil08]

<sup>10</sup> T. Berners-Lee - J. Hendler - O. Lassila The Semantic Web *Scientific American Magazine*. 2001. [BLHL01]

<sup>11</sup> Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter Patel-Schneider, and Ulrike Sattler: OWL 2: The next step for OWL [CHMP08]

- e. Azok az ontológiakészítési eredmények, amelyek átvételét komolyan meg kell fontolni: SUO,<sup>12</sup> az amerikai szabványos közcélú ontológia, DOLCE és testvérei, olasz ontológiák,<sup>13</sup> amelyek elsősorban természetes nyelvek feldolgozására, ill. az LKIF, az Estrella<sup>14</sup> nemzetközi projekt jogi csúcsontológiája.
  - f. Az OWL nyelvre alapuló technológiák alapvetően klasszikus logikai rendszerekre épülnek. Intelligens jogi rendszerek céljaira ezek már nem elegendők, hanem az OWL alapú rendszerek modális kiegészítésére van szükség.
5. A nyelvtechnológiai elvek nyelvészeti mélyelemzési megoldást követelnek. A mélyelemzés a mondatok értelmének, logikai szerkezetének megragadására törekszik, és az elemzett mondat határánál lényegesen szélesebb összefüggésben gondolkodik, amely technikát szokás diskurzusreprezentációnak is nevezni. A szerző – a PTE-BTK-n folyó ReALIS nyelvi mélyelemzési kutatóprojekt<sup>15</sup> munkájába bekapcsolódva a következő részeredményeket érte el:
- a. A korábban már említett ProType Prolog típusleíró kiegészítésre építve a szerző meghatározta a ReALLan nyelvet, amely a nyelvi információk megadására szolgál. A nyelv a Prolog kifejezésnyelvének egy alkalmazása (részhalmaza), amelynek jólformáltságát a ProType segítségével lehet ellenőrizni.
  - b. Szintén a ProType segítségével a szerző meghatározta a ReALM modális logikai nyelvet, amely a ReALIS elemzés végeredményeképpen állna elő. A nyelv tükrözi a projekt által felépített elmereprezentáció alapjait, az OWL modális logikai kiterjesztését, és az episztemikus és doxasztikus modalitáson túl magában foglalja a jogi alkalmazásokhoz szükséges deontikus modalitásfogalmat is.
  - c. A szerző rögzítette a nyelvi elemzés tárgymodelljét:<sup>16</sup> vagyis azt a Prolog kódmintát, amelyet a ReALLan-ból fordítás útján kaphatunk meg, és amely maga végzi az elemzést. Ennek során a szerző meghatározta a relációs tárgymodell és a következtetési tárgymodell fogalmait, elkészítette a ReALIS következtetési tárgymodelljét, és kísérleti elemző programot állított össze a következtetési tárgymodell alapján. A kísérleti elemző programot a Prolog visszafelé következtető stratégiája, és a Contralog előre következtető stratégiája alapján is lefuttatta. Az eredmények igazolták az elvárásokat.
  - d. A szerző elkészítette az Alberti Gábor által meghatározott polaritások hatáslánccsalád-modell kísérleti programját.<sup>17</sup> Az eredmények itt is igazolták

<sup>12</sup> Niles, I-Pease, A: Origins of the Standard Upper Merged Ontology: A Proposal for the IEEE Standard Upper Ontology [SUO01]

<sup>13</sup> Claudio Masolo-Stefano Borgo-Aldo Gangemi-Nicola Guarino-Alessandro Oltramari-Luc Schneider: The WonderWeb Library of Foundational Ontologies Preliminary Report [WW03]

<sup>14</sup> [www.estrellaproject.org](http://www.estrellaproject.org)

<sup>15</sup> Alberti Gábor: ReALIS. Interpretálók a világban, világok az interpretálóban [AG11]

<sup>16</sup> Kilián Imre: Tárgymodell változatok a ReALIS nyelvi elemzéshez [Kil11.2]

<sup>17</sup> Alberti, Gábor-Kilián Imre (2010): Vonzatkeretlisták helyett polaritások hatáslánc-családok – avagy a ReALIS  $\sigma$  függvénye [AGKI10]

az elvárásokat: a helyenként fellépő felesleges nondeterminisztikus eredmények között egy később beépítendő ontológiai ellenőrzési művelet segítségével lehet választani.

## 6 Egy jogi szemantikus rendszer terve

Az előző pontokban lefektetett elvek, valamint a már megtalálható vagy saját kidolgozásban létrejött megoldások és eszközök alkalmazásaképpen a szerző megalkotta egy jogi szemantikus tudásmenedzsment szoftver rendszer tervét.

1. A használati eset elemzésben a szerző körvonalazta egy általános célú jogi informatikai rendszer lehetséges használati eseteit. Az elemzés mélyreható nem lehetett, hiszen konkrét felhasználói igényekről nem volt szó. A vázolt esetek az elképzelhető műveletek legkisebb közös többszöröseként álltak elő, vagyis egy konkrét szoftver esetében egyes műveletek esetleg kihagyhatók belőlük.
2. A szerző felvázolta egy jogi informatikai rendszer rendszertervét. A rendszerterv elsősorban az alkalmazott technológiákat tükrözte, amelyek a használati eseteket megalapozhatják, és független volt a konkrétan megvalósítandó műveletektől. A tervezés során az egyik fontos szempont volt a terv elemeinek többcélű újrahasznosíthatósága, vagyis az, hogy az egyes szoftver összetevők szerepe eléggé általános legyen ahhoz, hogy ne csak az őket aktuálisan használó ügyfélprogram legyen képes használni. A másik igen fontos szempont volt a rendszer skálázhatósága, vagyis az a képesség, hogy esetlegesen az egész rendszer esetleg egyetlen gépen is működőképes legyen, de a felhasználói igények növekedésével a rendszer több kiszolgálóra is gond nélkül szétosztható legyen. A rendszerterv készítésekor három nagy egység különült el:
  - a. Az alkalmazói szoftver. Ez legcélszerűbben egy önálló kiszolgálón kaphat helyet. Ebben az összetevőben vannak a használati eset elemzésben rögzített műveletek megvalósítva, és ezt lehet szűkíteni vagy bővíteni a konkrét műveleti igények függvényében.
  - b. A természetes nyelvi feldolgozó csomag, amelynek tárolnia kell a működését vezérlő lexikon-adatbázist, és kapcsolódnia kell egy ontológiai alapú tudástárhoz. A lexikon a nyelvi információk forrásalakjából egy előfeldolgozó fázis után kerül a programba beszerkesztésre. A tudástár lehet közös is az alkalmazói rendszer által elért tudástárral, de lehetséges külön csak a természetes nyelvi feldolgozás céljaira szolgáló tudástárat is tervezni, mindaddig, amíg a különálló tudástárak tárolta információk egymáshoz illeszkednek, csereszabatosak. Az is egy lehetséges, hogy a kétféle célú tudástár csak forrás alakban tartalmaz közös szegmenseket (pl. közös csúcsontológiát), amelyeket a rendszer indításakor egy file-kiszolgáló-szerű szolgáltatástól kérünk el.
  - c. A jogi rendszer számára kifejlesztett tudástár. A tárolt jogi tudásnak ki kell terjednie: 1. a jogi normatívák tárolására, 2. a jogi esetek tárolására, 3. a jogi folyamatok és a folyamatokat kísérő dokumentumok (bizonyítékok, végzések, stb.) tárolására. A tárolásnak követnie kell az ontológiaszerkesztés már rögzített elveit.

## 7 Kísérleti megvalósítás

Az ontológiaépítési elvek kísérleti bizonyítékként a szerző irányításával végzett hallgatói laborgyakorlatként létrehoztunk egy miniatűr közlekedésjogi ontológiát. A kísérleti alkalmazás mindenesetre nem használ nyelvtechnológiai eszközöket: a megfelelő normatívák kézzel lettek ontológiai formátumba átültetve. Az alkalmazás beváltotta a reményeket: feltételezve egy közlekedésjogi háttér-térképet (amelyen a jelzőtáblák és egyéb közlekedésjogi létesítmények is szerepelnek), és feltételezve egy GPS-szerű eszközzel megadható esetleírást, a rendszer kikövetkezteti a szabályáthágásokat. Egy hasonló szerkezet, a repülőgépek fekete dobozához hasonló megoldással a gépjárművekbe beépítve vagy konkrét GPS eszközöket kiegészítve akár konkrét alkalmazásra is lelhet.

## 8 Következtetések

A legfontosabb hipotézis beigazolódt: a természetes nyelvtechnológiák és a mesterséges intelligencia technikák alkalmazásával komolyan meg lehet növelni az elektronikus jogi alkalmazások hatékonyságát. A megfelelő technológiai elemek azonban részben még kutatási fázisban vannak, de semmiképpen sincsenek közvetlenül alkalmazható állapotban. Egy mindezeknek megfelelő bevezető alkalmazás létrehozása ezért (is) várhatóan nem lesz olcsó sem és zökkenőmentes sem.

1. Jogi módszerek számítógépes alkalmazását a jog területén igen nagy számban tapasztalható nem teljesen egyértelmű megfogalmazás, ill. szándékosan képlékenyen megadott fogalmak komolyan megnehezítik. Ezek, pl. „a jó gazda gondosságával” gondozott bérlemény, a gyülekezési jog megsértése esetén alkalmazott „arányos beavatkozás”, a „látási és útviszonyoknak megfelelően megválasztott sebesség”. Az efféle megfogalmazások kezelését az értekezés nem tárgyalja, azokhoz az ún. elmosódó (fuzzy) logikai rendszerek lehetnek alkalmasak.
2. A jogászai munkaterületeken szintén sok esetben fellépő intuitív / megérzéses munkamódszer (ha a tapasztalt bíró szinte ránézésre meg tudja állapítani, hogy ki a bűnelkövető) szintén nem váltható ki számítógépes támogatással.
3. Számítógépes jogi eszközök alkalmazása számos esetben lehet kézenfekvő és magától értetődő. Létezik és létezhet viszont egy sor olyan alkalmazási körülmény, amely további jogi (pl. emberi jogi), vagy etikai kérdéseket vet fel. (Pl. Beépíthetünk-e „fekete dobozt” magángépjárművekbe, rögzíthetjük-e azok mozgását? Rábízhatjuk-e emberek sorsa felett a döntést egy gépre?). Ezek felvetése, vizsgálata és megválaszolása semmiképpen sem tárgya a jelen értekezésnek. Zárásként legfeljebb annyit tehetünk hozzá: a számítógéppel segített megoldások semmilyen alkalmazási területen helyettesítik az emberi műveletvégzést, és nem vállalják át az emberi felelősséget. Segítenek a műveletek gyorsabbá, hatékonyabbá vagy pontosabbá tételében, de a végső szót minden esetben az ember mondja ki, a kimondott vagy leírt szóért, de legalábbis az aláírásért a felelősség minden esetben az embert terheli.

## 9 Javaslatok a továbbiakra

A következtetésekből megfogalmazottak szinte ki is jelölik a továbblépés lehetséges irányát:

1. Célszerű lenne – valamilyen nagyobb szabású kutatási együttműködés keretén belül az értekezésben leírtakat megvalósítani. Úgy véljük, hogy egy általános célú, de természetesen jogi területen bevethető szemantikus tudásmenedzsment platform és/vagy eszközzrendszer igen sokrétű konkrét alkalmazásra találhat



2. Hasznos lenne – az értekezés tárgyán túlmutató – fuzzy megoldások kutatási szintű vizsgálata, és integrálásuk a kutatás tárgyát képező szoftver rendszerjavaslatokba
3. A közlekedési alkalmazás egy kész, megvalósításra érett dolog, amelyben kockázatot jelentő kutatási feladat minimális, vagy egyáltalán nincs is. Közvetlenül megbecsülhető fejlesztési erőforrásokat igényel, az eredmény alkalmazhatósága nem kétséges. Meg kell találni a fejlesztési erőforrásokat, a piaci modellt, szakjogászai munkával el kell oszlatni az esetleges kételyeket, ki kell fejleszteni és alkalmazásba kell venni az eszközt. Gyakorlati szempontból az eszköz esetlegesen piaci GPS alapú térképi rendszerekbe is könnyen beépíthető lehet.

## 10 Hivatkozások

- [CloMe] W. F. Clockshin-C. S. Mellish: Programming in Prolog  
Springer Verlag, New York 1987.
- [Bar99] Roman Barták: Constraint Programming: in Pursuit of the Holy Grail  
Charles University. Fac.of Mathematics and Physics, 1998.
- [UML99] Rumbaugh-Jacobson-Booch: Unified Modelling Language Reference  
Manual  
Addison-Wesley-Longman Inc. 1999.
- [OWL209] Motik, Bruno - Patel, Peter F. - Schneider-Bijan, Parsia (eds, 2009): OWL2  
Web Ontology Language Structural Specification and Functional Style  
Syntax, W3C Recommendation
- [CHMP08] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter  
Patel-Schneider, and Ulrike Sattler: [OWL 2: The next step for OWL](#).  
Journal of Web Semantics, 6(4):309-322, Nov 2008.
- [SUO01] Niles, I - Pease, A: Origins of the Standard Upper Merged Ontology: A  
Proposal for the IEEE Standard Upper Ontology  
Proceedings of Measuring Intelligence and Performance of Intelligent  
Systems Conference, 2001.
- [WW03] Claudio Masolo-Stefano Borgo-Aldo Gangemi-Nicola Guarino-Alessandro  
Oltramari-Luc Schneider: The WonderWeb Library of Foundational  
Ontologies Preliminary Report  
ISIB-CNR, Padova, Italy, 2003
- [AG11] Alberti Gábor:  $\Re$ ALIS. Interpretálók a világban, világok az  
interpretálóban.  
Akadémiai Kiadó, Budapest, 2011.
- [AGKI10] Alberti, Gábor - Kilián Imre (2010): Vonzatkeretlisták helyett polaritások  
hatáslánc-családok – avagy a  $\Re$ ALIS  $\sigma$  függvénye, In: Tanács Attila-  
Vincze Veronika szerk. VII. Magyar Számítógépes Nyelvészeti Konferencia,  
University of Szeged, Informatics Department Group, Szeged. 113-127.
- [Ru84] Ruzsa Imre: Klasszikus, modális és intenzionális logika  
Akadémiai Kiadó, 1984.
- [Lo08] Lokhorst, Gert - Jan: Mally's Deontic Logic, *The Stanford Encyclopedia of  
Philosophy (Winter 2008 Edition)*, Edward N. Zalta (ed.),  
(<http://plato.stanford.edu/archives/win2008/entries/mally-deontic>, letöltve:  
21-Már-2012.)

- [Kil11.1] Kilián Imre: Contralog: egy előre haladó Prolog motor és alkalmazása  
ReALIS nyelvi elemzésre  
Erdélyi Magyar Műszaki Tudományos Társaság, SzámOkt 2011.  
konferencia kiadványa,  
Kolozsvár, 2011, pp. 199-205.
- [Kil11.2] Kilián Imre: Tárgymodell változatok a ReALIS nyelvi elemzéshez  
VIII. Magyar Számítógépes Nyelvészeti Konferencia, Tanács Attila-Vincze  
Veronika szerk.  
Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged, 2011.

Kilián Imre doktorjelölt

Kelt Gyűrűfűn, 2013. április hó 22-én.