

RELONTOUML: RELÁCIÓS MODELLEN, ONTOLÓGIÁN ÉS UML-EN ALAPULÓ MODELL

RELONTOUML: DEVELOPMENT OF A MODEL BASED ON RELATIONAL MODEL, ONTOLOGY AND UML

Agárdi Anita

Miskolci Egyetem, Gépészmérnöki és Informatikai Kar, Informatikai Intézet, agardianita@it.uni-miskolc.hu

Abstract

This paper presents a model that combines ontology, UML modeling, and a relational model. The ontology model (and the Ontology Web Language - OWL), UML, and relational model are first introduced in the article. After a review of the literature, the comparison and conversion of the systems are presented. The created model is then presented and a real ontology is modeled using the presented model.

Keywords: *ontology, OWL, UML, relational model.*

Összefoglalás

Ez a cikk egy olyan modellt mutat be, amely egyesíti az ontológiát, az UML-modellezést és a relációs modellt. Az ontológiai modell (és az Ontology Web Language – OWL), az UML és a relációs modell kerül elsőként bemutatásra. Az irodalmi háttér bemutatása után a modellek összehasonlítását és átalakítását mutatja be a cikk. Ezután a létrehozott modell prezentálása és egy valós ontológia modellezése történik a bemutatott modell segítségével.

Kulcsszavak: *ontológia, OWL, UML, relációs modell.*

1. Bevezetés

A rendszer modellezése és megjelenítése fontos feladat. A szakirodalomban ilyen szabványos modellezési nyelvek az Unified Modeling Language (UML), a relációs adatbázismodell és az ontológiai modell. Az UML (Unified Modeling Language) [1] egy szabványos, általános célú modellező nyelv. A nagyszabású szoftverrendszerek modelljeinek vizuális dokumentálására szolgáló eszköz segítségével szöveges és grafikus modellek hozhatók létre különféle nézőpontokból, ideértve: rendszerek, szervezetek, folyamatok, szoftverek, programok, adatbázisok. Szerkezeti diagramokat és viselkedési/dinamikus diagramokat tartalmaznak. A szerkezeti diagramok a modellezett rendszer elemeire vonatkoznak. Ezen diagramok altípusai az alábbiak: osztálydiagramok, komponens diagramok, összetett szerkezeti diagramok, telepítési

diagramok, objektumdiagramok, csomagdiagramok. Viselkedési diagramok írják le, hogy minek kell történnie a modellezett rendszerben. Ez az alábbi diagramokból áll: tevékenységdiagramok, állapotgép-diagramok, használatieset-diagramok, interakciós diagramok.

Az ontológia [2] egy adott terület tudásának leírása. Szemantikai objektumokat tartalmaz egy adott területen. Az ontológia egyik legismertebb nyelve az Ontology Web Language (OWL). Az OWL osztályokból-álosztályokból építi fel a rendszert. Az egyed egy osztály meghatározott előfordulása. Az osztályok különböző tulajdonságokat tartalmazhatnak. Az egyedhez tartozik domain és range. Az adattípus-tulajdonságnál a domain egy osztály, a range adattípus, míg az objektumtulajdonságnál a domain és a range is egy osztály, tehát az adattípus-tulajdonság egy egye-

det társít egy adattípushoz, míg az objektumtulajdonság egy egyedet egy másik egyeddel társít. Az annotációs tulajdonság egy egyedet, osztályt, tulajdonságot köt össze egy annotációval.

Az OWL-ontológiát az UML-hez és a relációs modellhez hasonlóan sok területen használják, például egészségügy: egészségügyi ontológia-rendszer [3], COVID-19 [4, 5], oktatás: egyetemi ontológia [6], szoftvertechnológia [7].

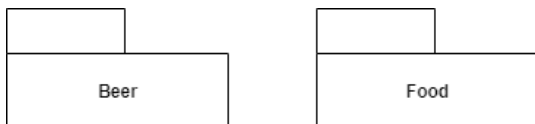
Az adatbázismodellek közül manapság a relációs modell [8] a legelterjedtebb. A reláció egy táblázat. Magát a relációt egyedi névvel látjuk el, sorai az adatokat jelölik. Az oszlopnevek egyedi névvel rendelkeznek a reláción belül, de egy másik reláció is tartalmazhat azonos nevű oszlopot. Egy oszlop és egy sor metszéspontját mezőnek nevezzük.

2. RELONTOUML-modell

A következőkben a kidolgozott modellt mutatom be. A modell a relációs modellen, az ontológián és az UML-modellen alapul, és RelOntoUML-modellnek neveztem el. A cél egy olyan modell kialakítása volt, amely a szoftverfejlesztők szemléletmódjához is közel áll, amelyet a szoftverfejlesztők is könnyen megértenek. A módszer lényege, hogy mind az UML, ontológia- és relációs modellekből merít tulajdonságokat, ezzel is könnyebbé teszi a szoftverrendszerek modellezését.

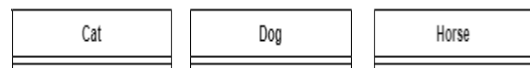
Az átalakítás lépéseit az alábbiakban ismertetem, mely során az ontológiai modellezést konvertálom át a RelOntoUML-modellre.

Az ontológia névteretet használ; egyedi elnevezéseknek kell lennie egy névtéren belül. Az UML-ben a csomagon belüli osztályoknak és az adatbázison belüli tábláknak egyedi nevüknek kell lenniük [9]. Ezt RelOntoUML-modellre konvertálva, csomagként jelenik meg.



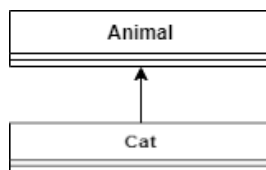
1. ábra. Névtér (RelOntoUML)

Az osztályok a rendszer leírói. Az ontológiai osztályokat egyszerűen UML-osztályokká alakítjuk a RelOntoUML-leírásnál [9].



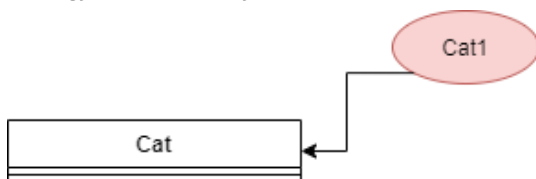
2. ábra. Osztályok (RelOntoUML)

Az osztály-alosztály hierarchia megmarad a RelOntoUML-modellben. Az OWL osztály-alosztály hierarchiáját az alábbi ábra szemlélteti. [9]



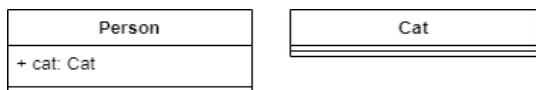
3. ábra. Alosztály (RelOntoUML)

Az egyedek az osztályok előfordulásai [9].



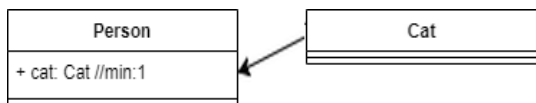
4. ábra. Egyed (RelOntoUML)

Az objektumtulajdonság két objektumot köt össze, így a domain és a range is objektumok. [9]



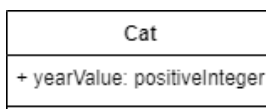
5. ábra. Objektumtulajdonság (RelOntoUML)

A minCardinality, maxCardinality határozza meg, hogy egy osztály hányszor kapcsolódik egy másik osztályhoz vagy adattípushoz [9].



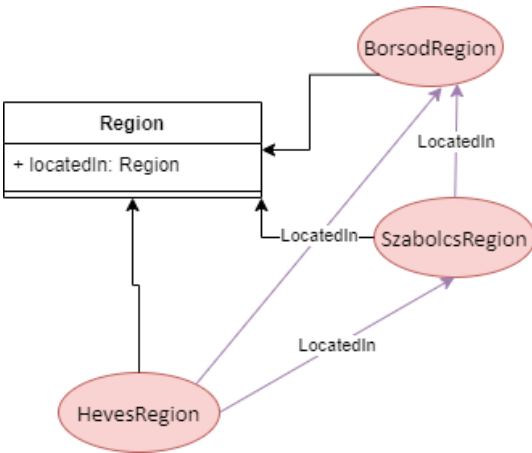
6. ábra. minCardinality, maxCardinality (RelOntoUML)

Az adattípus-tulajdonság egy egyedet társít egy adattípushoz [9].



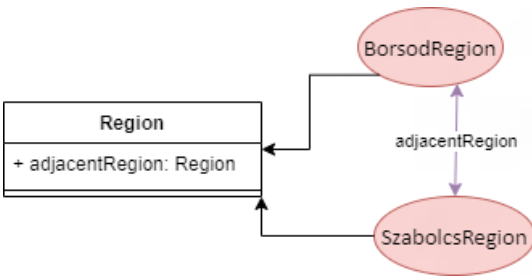
7. ábra. Adattípus-tulajdonság (RelOntoUML)

A tranzitív tulajdonság azt jelenti, hogy ha egy tulajdonság (A,B) és (B,C), akkor valójában (A,C) is [9].



8. ábra. *Tranzitív tulajdonság (RelOntoUML)*

A szimmetrikus tulajdonság az ontológiában azt jelenti, hogy ha egy tulajdonság (A,B), akkor az (B,A) is [9].



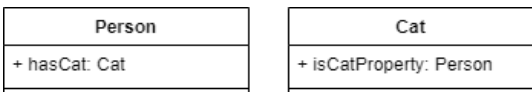
9. ábra. *Szimmetrikus tulajdonság (RelOntoUML)*

A funkcionális tulajdonság azt jelenti, hogy ha egy tulajdonság (A,B) és (A,C), akkor B = C [9].



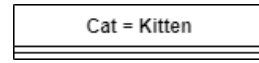
10. ábra. *Funkcionális tulajdonság (RelOntoUML)*

Az inverzobjektum-tulajdonság azt jelenti, hogy az egyik objektumtulajdonság a másik inverze [9].

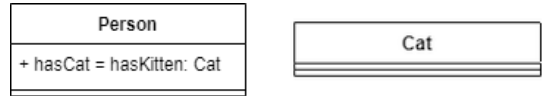


11. ábra. *InverseOf tulajdonság (RelOntoUML)*

Egy ontológiában létrehozhatunk két különböző nevű osztályt, amelyeket ha equivalentClass jelzővel jelölünk, akkor a két osztály valójában azonos. [9].

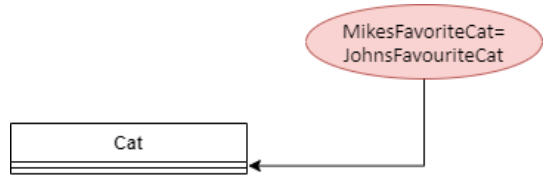


12. ábra. *Equivalent classes (RelOntoUML)*



13. ábra. *Equivalent property (RelOntoUML)*

A sameAs azt jelenti, hogy az ontológiában két különböző nevű egyed megegyezik [9].



14. ábra. *sameAs (RelOntoUML)*

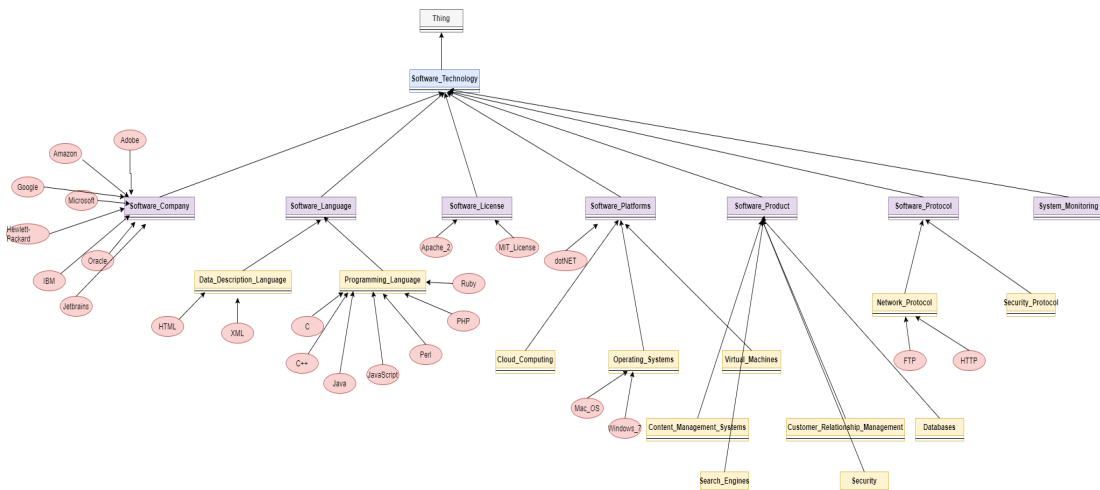
3. RELENTOUML-modell alkalmazása mintarendszere

A SoftwareTechnology [10] egy nagyobb ontológia, amely szoftvertechnológiát modellez. Modellezi a szoftverfejlesztő cégeket, programozási és adatleíró nyelveket, szoftverliceneket, platformokat (felhő, operációs rendszer), protokollokat (pl. http, ftp), szoftverterméktípusokat (pl. CMS, CRM).

Az ontológia modellezi a szoftvertechnológiát 4 szinten osztályokkal, és egy szinten egyedekkel. Nincsenek tulajdonságai sem egyedeknek, sem osztályoknak.

Az első szinten a „Thing” osztály található, amely az összes osztály őse minden ontológiában. A második szinten a 'Software_Technology' található. A harmadik szinten a következő osztályok találhatóak: Software_Company, Software_Language, Software_License, Software_Platforms, Software_Product, Software_Protocol, System_Monitoring. A negyedik szinten a következő osztályok találhatóak: Data_Description_Language, Programming_Language, Cloud_Computing, Operating_Systems, Virtual_Machines, Content_Management_Systems, Search_Engines, Customer_Relationship_Management, Security, Network_Protocol, Security_Protocol.

Az ontológia számos egyed is tartalmaz. A Software_Company a következő egyedeket tartalmazza: Adobe, Amazon, Google, Microsoft, Hewlett-Packard, IBM, Oracle, JetBrains. A Data_Description_Language a következő egyedeket tartalmazza: HTML, XML. És a következő programozási nyelveket tartalmazza: C, C ++, Java, JavaScript, Perl, PHP, Ruby.



15. ábra. A SoftwareTechnology vizualizálása a RelOntoUML segítségével

4. Összefoglalás

A cikk ontológiai, relációs és UML-modelleket mutat be. A modellkonverzió irodalma után bemutatásra került a RelOntoUML-moddell, valamint az ontológia OWL-átalakítása RelOntoUML-moddellre. A SoftwareTechnology OWL-moddell RelOntoUML-moddellre átalakítása is megtörtént a cikkben. A modell bemutatja a szoftvertechnológiát és annak elemeit, mint szoftvercég, programozási nyelv, licenc, platform, szoftvertermék, protokoll. A kifejlesztett modell előnye, hogy a szoftverfejlesztők számára egy jól kezelhető modellezést nyújt. Az így készített rendszer jól strukturált, könnyen érthető lesz. Jövőbeli kutatási területem az új ontológiák átalakítása az itt bemutatott RelOntoUML-moddellre, és a modell szükség szerinti finomítása.

Köszönetnyilvánítás

Az innovációs és technológiai minisztérium ÚNKP-21-3 kódszámú új nemzeti kiválóság programjának a nemzeti kutatási, fejlesztési és innovációs alapról finanszírozott szakmai támogatásával készült.

Szakirodalmi hivatkozások

[1] Dobing B., Parsons J.: *Dimensions of UML Diagram Use: a Survey of Practitioners*. Journal of Database Management (JDM), 19/1. (2008) 1–18.

[2] Antoniou G., Van Harmelen F.: *Web Ontology Language: Owl*. In: Handbook on Ontologies (2004) 67–92.

[3] Kiong Y. C., Palaniappan S., Yahaya N. A.: *Health Ontology System*. In: 2011 7th International Conference on Information Technology in Asia (2011) 1–4.

[4] Dutta B., DeBellis M.: *CODO: An Ontology for Collection and Analysis of COVID-19 Data*. arXiv preprint arXiv:2009.01210 (2020).

[5] Kachaoui J., Larioui J., Belangour A.: *Towards an Ontology Proposal Model in Data Lake for Real-Time COVID-19 Cases Prevention*. (2020).

[6] Malviya N., Mishra N., Sahu S.: *Developing University Ontology Using Protégé Owl Tool: Process and Reasoning*. International Journal of Scientific & Engineering Research, 2/9. (2011) 1–8.

[7] Gherabi N., Bahaj M.: *A New Method for Mapping UML Class into OWL Ontology*. Spec. Issue Int. J. Comput. Appl.(0975 8887) Soft. Eng. Databases Expert Syst. SEDEXS, (2012) 5–9.

[8] Codd E. F.: *Relational Database: A Practical Foundation for Productivity*. In: *Readings in Artificial Intelligence and Databases* (1989) 60–68.

[9] OWL Guide: <https://www.w3.org/TR/owl-guide/> (Hozzáférés dátuma: 2021. 11. 09.)

[10] SoftwareTechnology. (Hozzáférés: 2021. 11. 09.) <https://github.com/detnavillus/rdf-owl-ontologies.git>