

Struktúra alapú modellezés

1. Bevezetés	1
2. Strukturális modell	2
3. Fogalmak	3
3.1. Gráf	3
3.1.1. Gráftípusok	3
3.1.2. Navigáció gráfokban	4
3.1.3. Kiterjesztések	4
3.1.4. Műveletek gráfokon	5
4. Tulajdonságmodellezés	5
4.1. Műveletek	6
4.1.1. Szűrt nézet	6
4.1.2. Vetített nézet	6
5. Kitekintés	7
5.1. Modellezési nyelvek	7
5.1.1. UML (Unified Modeling Language)	7
5.1.2. AADL (Architecture Analysis & Design Language)	8
5.1.3. SysML (Systems Modeling Language)	8
5.1.4. EMF (Eclipse Modeling Framework)	8
5.1.5. AUTOSAR (AUTomotive Open System ARchitecture)	8
5.2. Refaktoring	9
5.3. Eszközök	9
5.4. Relációk tulajdonságai	9
5.4.1. Tranzitivitás	9
5.4.2. További tulajdonságok	9
5.5. Ontológia	9
6. Ajánlott irodalom	10
7. Szótár	11
Hivatkozások	11

1. Bevezetés

A modellezés során egy rendszert jellemezhetünk annak statikus, ill. dinamikus jellemzői alapján. A statikus jellemzés esetén a rendszer felépítését, *struktúráját* írjuk le, míg dinamikus jellemzés esetén a rendszer *viselkedésével* foglalkozunk.

Jelene anyag témája a *struktúra alapú modellezés*. Célja, hogy bemutassa a strukturális modellezés matematikai alapjait, a hierarchikus modellezés ötletét stb.

2. Strukturális modell

1. definíció. A *strukturális modell* a rendszer felépítésére vonatkozó tudás.

A strukturális modell alkalmas arra, hogy az alábbi kérdésekre nyújtson valamilyen (nem feltétlenül kimerítő) választ:

- Milyen elemekből áll a rendszer?
- Hogyan kapcsolódnak egymáshoz az elemek?
- Milyen tulajdonságúak az elemek?

Egy rendszer strukturális modellje a rendszer dekompozíciójával állítható elő.

2. definíció. A dekompozíció ("faktoring") egy összetett probléma vagy rendszer kisebb részekre bontása, amelyek könnyebben érthetők, fejleszthetők és karbantarthatók.

Természetesen a dekompozíció során ügyelnünk kell arra, hogy az egyes részekből visszaállítható legyen az eredeti rendszer, különben a kapott strukturális modellünk hiányos.

3. definíció. Egy dekompozíció *helyes*, ha a dekompozícióval kapott rendszer minden elemének megfeleltethető az eredeti rendszer valamelyik eleme, és az eredeti rendszer minden eleméhez hozzárendelhető a dekompozícióval kapott rendszer egy vagy több eleme.

1. példa.

4. definíció. *Top-down* modellezés során a rendszert felülről lefelé építjük. A modellezés alaplépése a *dekompozíció*.

A top-down modellezés jellemzői:

- + Részrendszer tervezésekor a szerepe már ismert
- – „Félidőben” még nincsenek működő részek
- – Részek problémái, igényei későn derülnek ki

5. definíció. *Bottom-up* modellezés során a rendszert alulról felfelé építjük. A modellezés alaplépése a *kompozíció*.

A bottom-up modellezés jellemzői:

- + Alrendszer önmagában kipróbálható, tesztelhető
- + Részleges készülségnél is összeépíthető valami
- – Nem látszik előre a rész szerepe az egészben

3. Fogalmak

Az alábbiakban összefoglaljuk a szükséges fogalmakat és definíciókat. Nem mindenhol célunk a teljes matematikai precizitás, ill. az egyes fogalmakhoz tartozó különböző definíciók felsorolása (pl. a *gráf* fogalomra sok különböző, némileg eltérő jelentésű definíció létezik).

3.1. Gráf

6. definíció. A *gráf* egy olyan $G = (V, E)$ struktúra, ahol V halmaz a csomópontok, E az élek halmaza. Az élek csomópontok között futnak, *irányítatlan gráf* esetén E csomópontok rendezetlen $\{v_1, v_2\}$ párjaiból áll (tehát nem különböztetjük meg a $\{v_1, v_2\}$ és a $\{v_2, v_1\}$ párokat, míg *irányított gráf* esetén csomópontok rendezett (v_1, v_2) párjaiból.

Bár a *gráf* meglehetősen egyszerű fogalom, rendkívül sok szakterület modelljei jól ábrázolhatók benne. Ennek egyik fő oka, hogy az emberek gondolkodás közben a világot gyakran úgy írják le, mint “dolgok” (csomópontok), amelyek egymással valamilyen “viszonyban” vannak (élek).

2. példa. Irányítatlan gráffal ábrázolhatjuk emberek között az *ismerőse* relációt, ilyen pl. a Facebook *friendship* relációja.

3. példa. Irányított gráffal ábrázolhatjuk emberek között a *követi* relációt, ilyen pl. a Twitter *follow* relációja. Szintén irányított gráffal ábrázolhatók az interneten egymásra mutató weboldalak.

Természetesen irányítatlan gráfok ábrázolhatók irányítottként is, ha minden $\{v_1, v_2\}$ irányítatlan élet lecserélünk. (Érdekesség: megfelelő transzformációval irányított gráfok is ábrázolhatók irányítatlan gráfként [18].)

Gyakran nincs szükségünk a teljes gráfra.

7. definíció. Egy $G = (V, E)$ gráfnak $G' = (V', E')$ *részgráfja*, ha G' megkapható úgy, hogy G -ből elhagyunk néhány csomópontot és azok éleit, ill. néhány további élet. Formálisan: ha $V' \subseteq V$ és $E' \subseteq V' \times V'$.

3.1.1. Gráftípusok

Tulajdonságaik alapján többféle gráftípust különböztethetünk meg, pl. a *teljes*, a *páros*, a *perfekt*, az *intervallum*- és a *síkbarajzoló* gráfok. Ezekkel bővebben a *Bevezetés a számításhalmazelméletbe* 2. tantárgy foglalkozik.

A strukturális modellezés szempontjából kiemelt szerepet játszanak a *fa* gráfok.

8. definíció. A körmentes, összefüggő gráfokat *fának* nevezzük.

9. definíció. A körmentes gráfokat *erdőnek* nevezzük.

Az *erdő* gráfot egyes források *ligetnek* nevezik.

A fák esetén gyakran kiemelt szerepet tulajdonítunk egy csomópontnak.

10. definíció. *Gyökér csomópont:* a fa egy megkülönböztetett csomópontja.

11. definíció. *Gyökeres fa:* olyan fa, ami rendelkezik gyökér csomóponttal.

12. definíció. *Gyökeres, szintezett fa:* a fa csomópontjaihoz hozzárendeljük a gyökértől vett távolságukat.

3.1.2. Navigáció gráfokban

Irányított és irányítatlan gráfokban is gyakran szükség van arra, hogy az egyes élek mentén “lépkedjünk”.

13. definíció. *Séta:* szomszédos csúcsok és élek váltakozó sorozata.

14. definíció. *Út:* olyan séta, amely nem metszi önmagát, valamint első és utolsó csúcsa különbözik.

15. definíció. *Kör:* olyan séta, amely nem metszi önmagát, valamint első és utolsó csúcsa megegyezik.

3.1.3. Kiterjesztések

A gráf adatmodellnek többféle kiterjesztése is létezik. Ezek közül az alábbiak ismeretére építünk:

- *Címkézett gráf:* a gráf elemeit (csomópontjait és/vagy éleit) *címkékkel* láthatjuk el. A címkézés célja lehet *egyedi azonosító* hozzárendelése vagy bizonyos tulajdonság leírása (pl. csomópontok kapacitása, élek típusa).
 - *Csúscsímkézett gráf:* az egyes csomópontokat címkézzük.
 - *Élcímkézett gráf:* az egyes (irányított vagy irányítatlan) éleket címkézzük.
- *Típusos gráfok:* a gráf elemei *típusokkal* rendelkeznek. A típusos gráfok a *címkézett gráfok* speciális esetei.
- *Tulajdonsággráf:* az egyes csomópontokat és éleket tulajdonságokkal látjuk el. A tulajdonságok kulcs-érték párok, pl. “neve = Gipsz Jakab”, “életkora = 20”, “színe = zöld”.

További kiterjesztések:

- *Súlyozott gráfok*: a gráf elemeit (csomópontjait és/vagy éleit) súlyokkal látjuk el, pl. kapacitást vagy költséget rendelünk hozzájuk. A *súlyozott gráfok* a címkézett gráfok speciális esetei.
- *Multigráfok*: a csomópontok között *többszörös élek* is futhatnak.
- *Hipergráfok*: egy él több csomópont között is futhat. A hipergráf éleit *hiperéleknek* nevezzük.

3.1.4. Műveletek gráfokon

Elérhetőségvizsgálat Egy v_i csomópontból egy v_j csomópont elérhető, ha létezik v_i -ből v_j -be irányított út.

Ez elérhetőségvizsgálatra gyakran elemi bejáróalgoritmusokat használunk:

- Szélességi bejárás (BFS)
- Mélységi bejárás (DFS)

Útkereső algoritmusok

- Dijkstra-algoritmus
- A* keresés
- Bellman–Ford algoritmus
- Floyd algoritmus

A legrövidebb út keresése kipróbálható az alábbi példán: <http://gist.neo4j.org/?6d2a1a1c6325043d09a0>.

Transzformációk

- *Szűrés éltípusra*: gyakran csak bizonyos éltípus(ok)ra van szükségünk. Egy $G = (V, E)$ gráf adott c_1, \dots, c_n élcímkékre szűrt részgráfja $G' = (V, E')$, ahol V a csomópontok változatlan halmaza, míg E' azon e élek halmaza, ahol e címkéje eleme a c_1, \dots, c_n halmaznak.

4. Tulajdonságmodellezés

Egy modellelem (gráfok esetén tipikusan a modell csomópontjai) tulajdonságait *jellemzőkkel* vagy más néven *attribútumokkal* jellemezzük. Különböző modellelmekre az egyes jellemzők más-más vehetnek fel (de nem szükségszerűen térnek el).

A jellemzők között gyakran kitüntetünk egyet:

16. definíció. A *típus* egy kitüntetett jellemző, amely tipikusan állandó egy modellelem életciklusa során. A többi jellemzőt *tulajdonságnak* hívjuk.

Azonos *típussal* rendelkező modellelemek *közös tulajdonságokkal* bírnak.

A tulajdonságokat gyakran ábrázoljuk ún. *relációkban*. A reláció pontos halmazelméleti definíciójával és a relációkon értelmezett műveleteket definiáló relációalgebrával bővebben a 3. féléves *Adatbázisok* c. tárgy foglalkozik. Most egy egyszerűbb, kevésbé formális definíció alkalmazunk:

17. definíció. A reláció a modellelemeket és azok jellemzőit tárolja. A relációkat gyakran táblázatos formában ábrázoljuk, ekkor a táblázat egy sora a modell egy elemének, egy oszlopa egy jellemzőjének felel meg.

Példaként használjuk az alábbi táblázatot!

név	fénykard színe	nem	holdak száma	keringési idő
Alderaan	piros	férfi hímnős nő	1	364
Coruscant			4	368
Darth Vader				
Jabba				
Leia Organa	zöld	férfi	3	304
Luke Skywalker	zöld	férfi		
Mace Windu	lila	férfi		
Tatooine	zöld	férfi		
Yoda	zöld	férfi		

Látható, hogy a táblázatnak nem minden celláját töltöttük ki. Ezeket a cellákat gyakran az *NA* (not applicable, not available) rövidítéssel vagy (különösen az adatbázisokban) *null* értékkel jellemezzük.

4.1. Műveletek

A relációkon most két alpműveletet definiálunk, ezek a *szűrés* (relációalgebrában *szelekció*) és a *vetítés* (relációalgebrában *projekció*). Ezekkel a műveletekkel a modellünknek egy-egy nézetét állíthatjuk elő.

4.1.1. Szűrt nézet

Szűrés során egy feltétel mentén bizonyos sorokat elhagyunk a relációból:

Szűrés a **fénykard színe = "zöld"** feltételre:

név	fénykard színe	nem	holdak száma	keringési idő
Luke Skywalker	zöld	férfi		
Yoda	zöld	férfi		

Szűrés a **holdak száma > 2** feltételre:

név	fénykard színe	nem	holdak száma	keringési idő
Coruscant			4	368
Tatooine			3	304

4.1.2. Vetített nézet

Vetítés során a modell egyes jellemzőit elhagyjuk a relációból. Fontos, hogy az esetleg létrejövő üres sorokat is elhagyjuk.

Vetítés a {**holdak száma**, **keringési idő**} jellemzőkre.

holdak száma	keringési idő
1	364
4	368
3	304

Vetítés a {név, fénykard színe} jellemzőkre.

név	fénykard színe
Alderaan	piros
Coruscant	
Darth Vader	
Jabba	
Leia Organa	zöld
Luke Skywalker	
Mace Windu	lila
Tatooine	zöld
Yoda	

5. Kitekintés

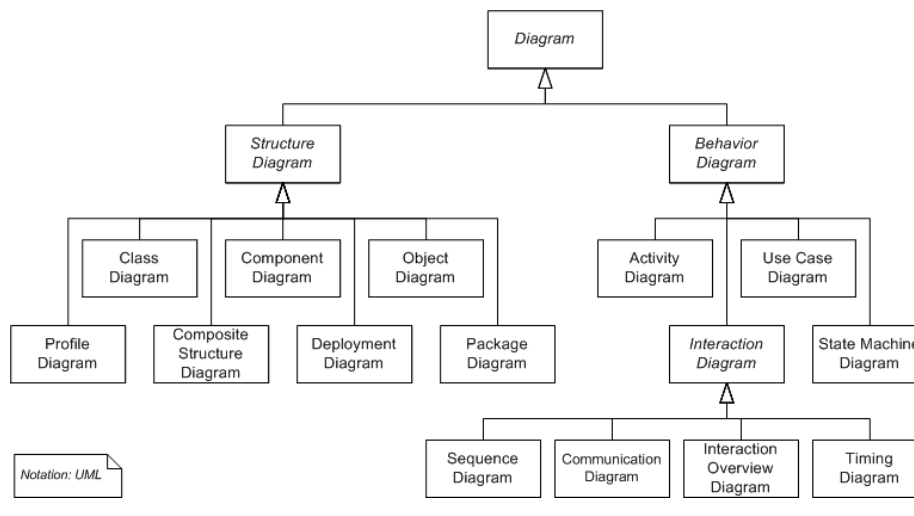
Az alábbiakban bemutatunk néhány, a strukturális modellezés témaköréhez kapcsolódó gyakorlati technológiát, specifikációt és eszközt. Az itt felsorolt fogalmak nem részei a számonkérésnek – viszont gyakran előkerülnek a későbbi tanulmányok és munka során, ezért mindenképpen érdemes legalább névről ismerni őket.

5.1. Modellezési nyelvek

5.1.1. UML (Unified Modeling Language)

Az UML egy általános célú modellezési nyelv az [13]. Az UML három fő diagramtípust definiál:

- *Structure Diagram*: strukturális modellek leírására. A *Class Diagram* az osztályok (metamodell), míg az *Object diagram* a példányok (modell) leírásra alkalmas. A *Composite Structure Diagram* egy rendszer struktúráját és a rendszer komponenseinek kapcsolatát mutatja be.
- *Behaviour Diagram*: viselkedési modellek leírására, pl. a *State Machine Diagram* segítségével állapotgépek, az *Activity Diagram* folyamatok ábrázolhatók.
- *Interaction Diagram*: a *Behaviour Diagram* altípusa. Ennek megfelelően szintén a viselkedés leírása a célja, de a hangsúly a vezérlés- és adatáramlás bemutatásán van. Ilyen pl. a *Sequence Diagram* (szekvenciadiagram), amely az egyes objektumok közötti interakciót mutatja be üzenetek formájában.



1. ábra. Az UML diagramjai egy UML osztálydiagramon ábrázolva
Az UML nyelvvel részletesen foglalkozik a *Szoftvertchnológia* c. tárgy.

5.1.2. AADL (Architecture Analysis & Design Language)

Az AADL eredetileg repülőipari célokra fejlesztett architektúraleíró nyelv.

5.1.3. SysML (Systems Modeling Language)

A SysML egy UML-alapú általános modellezési nyelv rendszertervezési célokra. A SysML az UML egy részhalmazát bővíti ki és új diagramtípusokat is bevezet, pl. *Requirement Diagramot* követelmények és viszonyaik ábrázolására.

5.1.4. EMF (Eclipse Modeling Framework)

Az Eclipse fejlesztőkörnyezet [5] saját modellezési keretrendszerrel rendelkezik, ez az *Eclipse Modeling Framework (EMF)*. Az EMF metamodellező nyelve, az Ecore lehetővé teszi saját, ún. szakterület-specifikus nyelv (domain-specific language, DSL) definiálását. Az EMF mára több területen is *de facto* modellezési keretrendszer, sikere nagyban hozzájárul az Eclipse népszerűségéhez.

Az Eclipse-szel és az EMF-fel foglalkozik az *Eclipse-alapú technológiák* c. szabadon választható tárgy.

5.1.5. AUTOSAR (AUTomotive Open System ARchitecture)

Az AUTOSAR [2] nagy autóipari gyártók és beszállítók által fejlesztett szabványos architektúran nyelv, amely az egyes hardver- és szoftverkomponensek együttműködése magas szinten definiálható. Az AUTOSAR konzorciumnak a Méréstechnika és Információs Rendszerek Tanszék is tagja [3].

5.2. Refaktoring

A dekompozícióhoz, azaz faktoringhoz szorosan kapcsolódik a *refactoring* (refactoring) fogalma [12]. Refaktoring során létező rendszerek, programkódok (vagy modellek) olyan átalakítását értjük, amelynek sorána megfigyelhető működés változatlan marad, de a kapott programkód (modell) olvashatóbb, érthetőbb, karbantarthatóbb lesz.

5.3. Eszközök

Gráfok automatikus megjelenítésére alkalmas pl. a GraphViz [6] programcsomag. Gráfok feldolgozására gyakran alkalmazzák az igraph [7] programcsomagot. Manapság több gráfadatbázis-kezelő rendszer is elterjedt, pl. a Neo4j [8] és a Titan [9] rendszerek.

Gráfok manuális rajzolására szintén több eszköz elterjedt. Egyszerűen használható online felületet biztosít a draw.io [4] és az Arrow Tools [1]. Komolyabb célokra a yEd [10] eszközt érdemes használni. Sok információt tartalmazó gráf esetén érdemes lehet vektorgrafikus rajzoló-, ill. prezentáló eszközök, pl. Microsoft Visio vagy Microsoft PowerPoint alkalmazása.

5.4. Relációk tulajdonságai

A *transzitiv lezárás* fogalom megértését segítheti, ha ismerjük a relációkon értelmezett tulajdonságokat, köztük a *transzitivitást*.

5.4.1. Transzitivitás

Egy S halmazon értelmezett r (kétféltváltozós) reláció *transzitiv*, ha bármely $a, b, c \in A$ esetén $r(a, b)$ és $r(b, c)$ teljesülése esetén $r(a, c)$ is teljesül.

4. példa. Az *egyenlőség* ($=$) és a *kisebbség* ($<$) reláció transzitivak, mert $a = b$ és $b = c$ esetén $a = c$, ill. $a < b$, $b < c$ esetén $a < c$. Az emberek közötti *őse* reláció transzitiv, mert ha $őse(a, b)$ és $őse(b, c)$, akkor $őse(a, c)$ is fennáll. A *nemegyenlőség* (\neq) reláció nem transzitiv, mert $a \neq b$ és $b \neq c$ esetén $a \neq c$ nem mindig áll fenn, pl. $1 \neq 2$ és $2 \neq 1$ esetén $1 \neq 1$ nem teljesül. Az emberek közötti *ismerőse* reláció nem transzitiv, mert *ismerőse*(a, b) és *ismerőse*(b, c) esetén nem garantált, hogy *ismerőse*(a, c) fennáll.

5.4.2. További tulajdonságok

A Rendszermodellezés tárgynak nem témája, de érdemes megismerni a relációk további tulajdonságait: *reflexivitás*, *szimmetria*, *rendezés* stb. [21].

5.5. Ontológia

Az *ontológia* egy nem egyértelmű nehezen definiálható kifejezés. Az informatikában általában az alábbihoz hasonló módon definiálják:

18. definíció. Az *ontológia* egy olyan taxonómia, amely tartalmazza a benne szereplő fogalmak közötti viszonyokat.

6. Ajánlott irodalom

A tulajdonsággráfokról egy jól érthető cikk Marko Rodriguez és Peter Neubauer munkája [19]. Rodriguez a Titan elosztott gráfadatbázis-kezelő rendszer fő fejlesztője [9], míg Neubauer a Neo4j gráfadatbázis-kezelőt fejlesztő cég alapítója [8]. Az elosztott gráfadatbázis alkalmazását, elméleti és gyakorlati kihívásait egy prezentáció is bemutatja [17].

Barabási-Albert László magyar fizikus nemzetközileg elismert kutatója a komplex hálózatok elméletének. Behálózva c. könyve közérthető stílusban mutatja be a hálózatok elemzésének elméleti kihívásait a kutatási eredmények gyakorlati jelentőségét [11]. A szerzővel több interjú is készült [14–16].

Az osztályok és prototípusok közötti elvi különbséget mutatja be Antero Taivalsaari, a Nokia Research fejlesztőjének 1996-os cikke [20].

7. Szótár

magyar kifejezés	angol kifejezés
(de)kompozíció	(de)composition
gráf	graph
csomópont	node, vertex
él	edge, relationship
címke	label
gyökér	root
fa	tree
erdő	forest
körmentes	acyclic
irányított	directed
irányítatlan	undirected
hipergráf	hypergraph
reláció	relation
jellemző, attribútum	property, attribute
vetítés, projekció	projection
szűrés, szelekció	selection
típus	type
osztály	class
objektum	object
referencia	reference
mező	field
metódus	method
példány	instance
a instance of b	a példánya b -nek
t type of c	c típusa t -nek
öröklés	inheritance
alosztály, leszármazott	subtype, descendant
ősosztály, ő	supertype, ancestor
tranzitív	transitive
tranzitív lezárás	transitive closure
taxonómia	taxonomy
ontológia	ontology

Hivatkozások

- [1] Arrow Tool. <http://www.apcjones.com/arrows/>.
- [2] AUTOSAR, a.
- [3] Attendees: AUTOSAR. <http://www.autosar.org/partners/current-partners/attendees/>, b.
- [4] draw.io. <https://www.draw.io/>.

- [5] Eclipse – The Eclipse Foundation open source community website. <http://eclipse.org/>.
- [6] GraphViz, Graph Visualization Software. <http://www.graphviz.org/>.
- [7] igraph – The network analysis package. <http://igraph.org/>.
- [8] Neo4j, The World’s Leading Graph Database. <http://neo4j.org/>.
- [9] Titan: Distributed Graph Database. <http://thinkaurelius.github.io/titan/>.
- [10] yWorks – The Diagramming Company. <http://www.yworks.com/>.
- [11] Barabási Albert-László. *Behálózva – A hálózatok új tudománya*. Helikon Kiadó, 2013.
- [12] M. Fowler, K. Beck, J. Brant, W. Opdyke, and D. Roberts. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Object Technology Series. Pearson Education, 2012. ISBN 9780133065268. URL <http://books.google.hu/books?id=HmrDHwgkbPsC>.
- [13] Object Management Group. Information technology – object management group unified modeling language (omg uml) – part 2: Superstructure. Technical Report ISO/IEC 19505-2:2012, Object Management Group, 2012. URL <http://www.omg.org/spec/UML/ISO/19505-2/PDF/>.
- [14] Index. Mostantól tudjuk megérteni az emberi viselkedést. <http://index.hu/tudomany/bal080429/>, May 2008.
- [15] Index. Az első cikk után majdnem leharapták a fejünket. http://index.hu/tudomany/2010/06/02/az_elso_cikk_utan_majdnem_leharaptak_a_fejunket/, June 2010.
- [16] Index. Az NSA primitíven használta a begyűjtött adatokat. http://index.hu/tudomany/2015/02/20/az_nsa_primitiven_hasznalta_a_begyujtott_adatokat/, February 2015.
- [17] Marko Rodriguez. Titan: The Rise of Big Graph Data. <http://www.slideshare.net/slidarko/titan-the-rise-of-big-graph-data>, June 2012.
- [18] Marko A. Rodriguez. Mapping Semantic Networks to Undirected Networks. *arXiv preprint arXiv:0804.0277*, 2008.
- [19] Marko A. Rodriguez and Peter Neubauer. Constructions from dots and lines. *CoRR*, abs/1006.2361, 2010. URL <http://arxiv.org/abs/1006.2361>.
- [20] Antero Taivalsaari. Classes vs. Prototypes – Some Philosophical and Historical Observations. *Journal of Object-Oriented Programming*, 10 (7): 44–50, 1996. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.56.4713&rep=rep1&type=pdf>.

- [21] Wikipedia. Reláció. <http://hu.wikipedia.org/wiki/Rel%C3%A1ci%C3%B3>, March 2015.