Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Általános célú szerkesztőfelület parciális modellekhez

Szakdolgozat

Készítette
Deim Péter Pál

Konzulens
Semeráth Oszkár

2016. december 9.

Tartalomjegyzék

Kivonat **i**

Abstract **ii**

1. Bevezetés **1**
1.1. Témamegjelölés . . . 1
1.2. Problémafelvetés . . . 1
1.3. Célkitűzés . . . 1
1.4. Kontribúció . . . 1
1.5. Hozzáadott érték . . . 1
1.6. Dolgozat felépítése . . . 1

2. Előismeretek **2**
2.1. Bemutatás egy példa segítségével . . . 2
2.2. Metamodell . . . 2
2.3. Részleges modell . . . 2
2.3.1. Szintaktika . . . 3
2.3.2. Szemantika . . . 3

3. Áttekintés **6**

4. Megvalósítás **7**
4.1. Szükséges eszközök . . . 7
4.1.1. Eclipse . . . 7
4.1.2. EMF . . . 7
4.1.3. Java . . . 7
4.1.4. Sirius . . . 7
4.1.5. Aql . . . 7
4.2. Metamodell szerkezete . . . 8
4.2.1. Általános modell . . . 8
4.2.2. Kiegészítés részleges modellé . . . 9
4.2.3. Példánymodell felülete . . . 9
4.2.4. Funkciók . . . 9

5. Összefoglalás és továbbfejlesztési lehetőségek **11**

Függelék **12**

Irodalomjegyzék **12**

HALLGATÓI NYILATKOZAT

Alulírott *Deim Péter Pál*, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2016. december 9.

Deim Péter Pál
hallgató

Kivonat

Abstract

1. fejezet

Bevezetés

1.1. Témamegjelölés

Modell-vezérelt tervezésnek nagy szerepe van az informatikában egy szoftver vagy rendszer létrehozásánál. Modellek segítségével sokkal jobban, strukturáltabban és megbízhatóbban lehet tervezni. Ehhez a rengeteg eszköz áll rendelkezésünkre, viszont ezek nem mindenre nyújtanak megoldást.

1.2. Problémafelvetés

Mai modellező eszközökkel nem lehetséges, hogy részleges, vagy hibás modelleket kezeljünk, elmentsünk. A modell készítésének közben lehetnek döntések, amik bizonytalanok, és nem is feltétlen fontos velük foglalkozni abban a tervezési szakaszban. Ezeket az elemeket célszerű lenne valamilyen módon megjelölni. Előfordulhat hogy a modellben egy elem megléte, vagy a multiplicitása kétséges. Esetleg, olyan, hogy nem lehet eldönteni egy elemről hogy az melyik másikkal áll kapcsolatban.

1.3. Célkitűzés

Kutatásom célja, olyan domain független modell elkészítése, aminek segítségével lehet részleges, vagy hiányos modelleket készíteni. Például, előfordulhat olyan, hogy nem tudjuk eldönteni egy attribútumról, hogy melyik elemhez tartozik. Ilyet általában a modellező eszközök nem támogatnak. Célszerű lenne egy olyan általános módszert kitalálni, amivel lehetséges az ilyen és ehhez hasonló esetek megjelölése, kezelése.

1.4. Kontribúció

Kutatásom során megismerkedtem a részleges modellezés leglényegesebb aspektusaival. Létrehoztam egy olyan metamodellt, ami képes részleges modellek készítésére. Ezután elkészítettem egy olyan vizuális szerkesztőfelületet, aminek a segítségével részleges modell példányokat lehet készíteni.

1.5. Hozzáadott érték

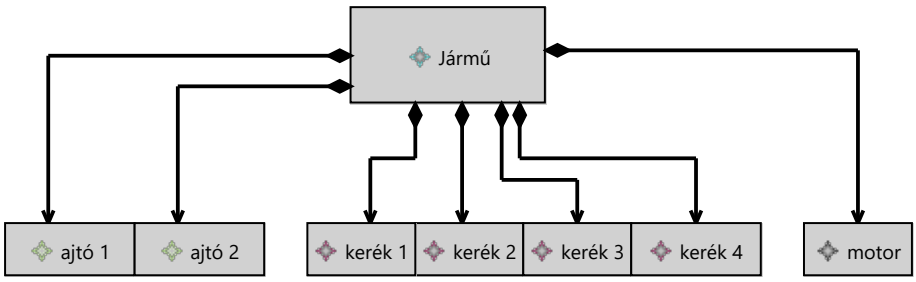
1.6. Dolgozat felépítése

2. fejezet

Előismeretek

2.1. Bemutatás egy példa segítségével

A részleges modellezést legegyszerűbben egy gyakorlati példán lehet bemutatni. Vegyük példának az UML objektumdiagramját. Az UML (Unified Modeling Language) egy szabványos, objektumorientált modellezési nyelv, ami a tervezést, fejlesztést és egyéb folyamatokat segíti. Ezt a nyelvet informatikában és egyéb üzleti területeken egyaránt alkalmazzák. Magában foglal több diagram típust is. Az objektumdiagram egy strukturális diagram, az osztálydiagramnak egy példánya, egy konkrét megvalósítás. Példának vegyük egy jármű objektum diagramját.



2.1. ábra. Jármű egyszerűsített objektumdiagramja

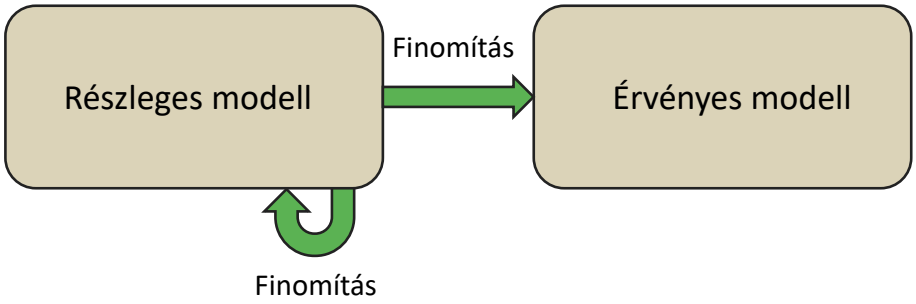
2.2. Metamodell

A metamodell az tulajdonképpen egy modellnek a modellje. Egy olyan alapséma, amire illeszkedik az összes hozzá tartozó magasabb absztrakciós szinttel rendelkező modell. Tegyük fel hogy M* modell M1 modellnek a metamodellje. Akkor M* metamodell meghatározza az M1 modellben lévő elemeket, attribútumokat és ezeknek lehetséges kapcsolatait. Ilyen metamodell például az objektumdiagramnak az osztálydiagram, ami leírja, hogy objektumok hogyan kapcsolódhatnak egymással és milyen tulajdonságai lehetnek.

2.3. Részleges modell

Részleges modell segítségével lehetőség van olyan modellezési döntések elodázására, aminek az akkori helyzetben még nincs relevanciája. Általában a modellező eszközökkel csak érvényes modelleket lehet készíteni. Így a modell készítője belekényszerül olyan döntések meghozatalába, amiket csak később kéne meghozni. Részleges modellben lehetőség van ezeket a kétséges helyzeteket már a modellezés szintjén kezelni. Így a modell nem csupán

strukturális információt tartalmaz, hanem a részlegességéről is, tehát hogy mennyire teljes a modell. Az előbbi példát tekintve, lehetséges az hogy a járműnek nem 2 hanem 4 ajtót szeretnénk. Erről információt az UML szabályai szerint nem tudunk tárolni. Vagy feljegyezzük a lehetséges változtatást, vagy pedig létrehozunk egy másik diagramot, amibe 4 ajtós a jármű. Ezek egyike sem tűnik jó megoldásnak. Egy ilyen picike diagram esetén még akár átlátható de egy nagyobb, akár több száz elemből álló modell esetén, ha máshol is van ilyen kétség a végleges modellel kapcsolatban, akkor már kezelhetetlenné válik. Lehetőség van a modell finomítására is. Finomítás során a részleges modellből kikerülnek bizonytalan elemek. Ez azt jelenti, hogy a modellben jelzett részlegesség mértéke csökkenthető és ennek eredményeképpen véges számú finomítás után a modellből teljesen eltűnnek a részlegességek.



2.2. ábra. Finomítás menete

2.3.1. Szintaktika

Részleges modellek esetén annotációkkal lehet megjelölni a modellt, illetve annak elemeit. Modellezés során 4 fajta részlegességet jelölhetünk meg. Annotációk

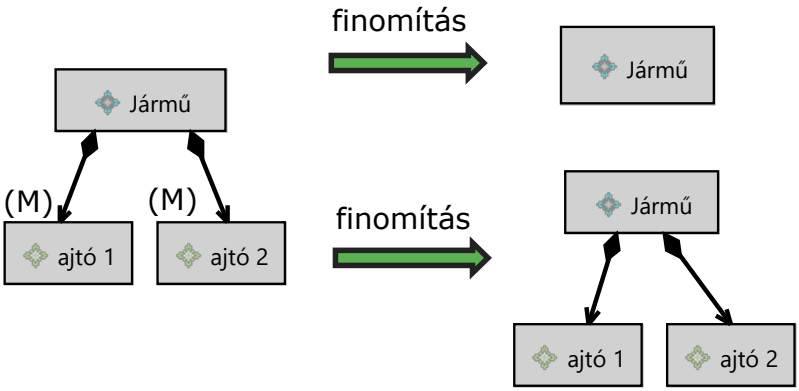
2.3.2. Szemantika

May

Annotációkkal láthatjuk el a modellt, az alapján, hogy egy modellelem biztosan benne lesz a végleges modellben, vagy pedig még bizonytalan a léte. 'M' May exist(lehet, hogy benne lesz a végleges modellben), 'E' Must exist(biztosan benne lesz a végleges modellben). A modell finomításával lépésről lépésre egyre kevesebb 'M' lesz. 'M' az vagy átvált 'E' –re, vagy teljesen kikerül a modellből. Akkor tekinthető a modell véglegesnek, ha már nem szerepel benne 'M'-el megjelölt elem. Tegyük fel, hogy nem tudjuk milyen járművet akarunk még modellezni a kezdeti fázisban. Ezért a kiindulási objektummodellben elláttuk May annotációkkal a járműnek az ajtó elemeit. Így finomítás során ezek az elemek később eltűnhetnek de akár meg is maradhatnak. Az a jármű aminek nincs ajtaja lehet akár egy motor, a két ajtós változat pedig egy autó.

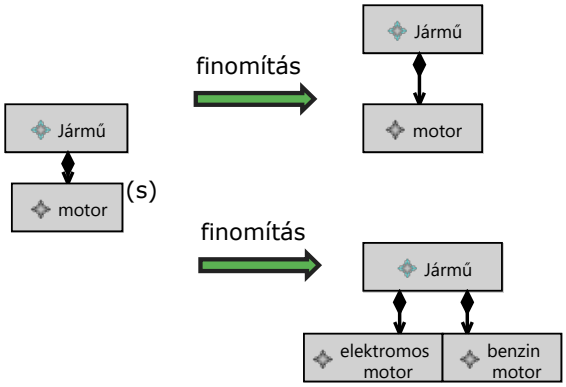
Abs

Ennek is két különböző annotálási módja lehet: 'P', Particular (egyedi elem) és 'S', azaz Set (több elemet jelölhet). Particular az olyan elem, ami már biztosan benne lesz a végleges modellben, azonban a Set olyan elemet jelöl, ami lehet, hogy a végleges modellben több elemként fog megjelenni. Finomítások során az 'S'-el megjelölt tagokat szétbontjuk több részre, de a végleges esetnél már csak particular, egyedi elemek szerepelhetnek. Az alábbi ábra szerint első lépésben még nincs eldöntve, hogy az autónak egy vagy több motorja lesz. Ezt meg is jelöltük a megfelelő annotációval ('S'). Így a finomítás során lehetséges, hogy



2.3. ábra. May részlegesség feloldása

marad az egy motoros jármű, de lehet hogy szétbontjuk a motort és a járműbe rakunk egy benzinmotort és egy elektromos motort is.

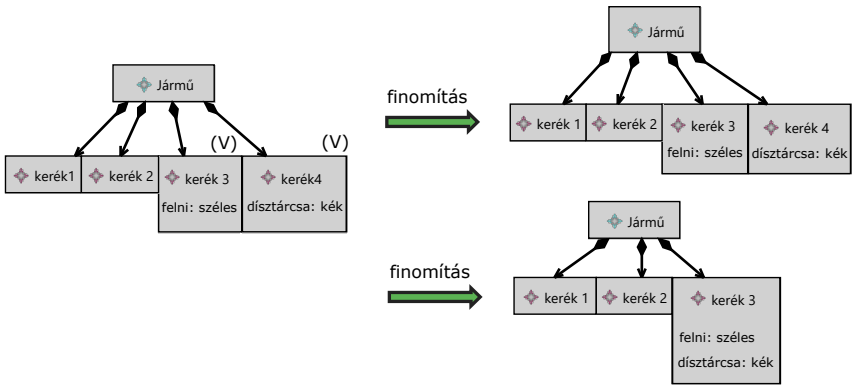


2.4. ábra. Abs részlegesség feloldása

Kétféleképpen lehet annotálni az elemeket: 'C' Constant (konstans elem) és 'V' Variable (változó elem). Bizonyos értelemben az Abs fordítottjának lehet tekinteni. Felveszünk több elemet, ami lehet, hogy későbbiekben összeolvasztunk egy elemmé, tehát fenn áll a lehetősége annak, hogy két elem megegyezik. Amikor elkezdünk egy modellt, akkor nem biztos, hogy meg tudjuk mondani elemekről, hogy azok a későbbiekben azonosak-e vagy különbözőek. A végleges modellben már csak konstans elemek lehetnek. Itt a példában látható, hogy a kezdeti állapotban még két külön kereke van a járműnek egyik kerekén kék dísztárcsa van a másik kereke pedig széles felnivel rendelkezik. Ezek meg lettek jelölve 'V' annotációval. Finomítás után ez megmaradhat ilyen különálló formában, de akár összeolvaszthatjuk ezt a két kereket és a végeredményben egy kerék marad, ami mind a kettő kerék tulajdonságát magában hordozza.

OW

Modellezés folyamán lehet megjelölni azt, hogy a modell már végleges e vagy sem. Ez a részlegesség nem a modell elemeire vonatkozik, hanem a teljes modellről árul el információt.

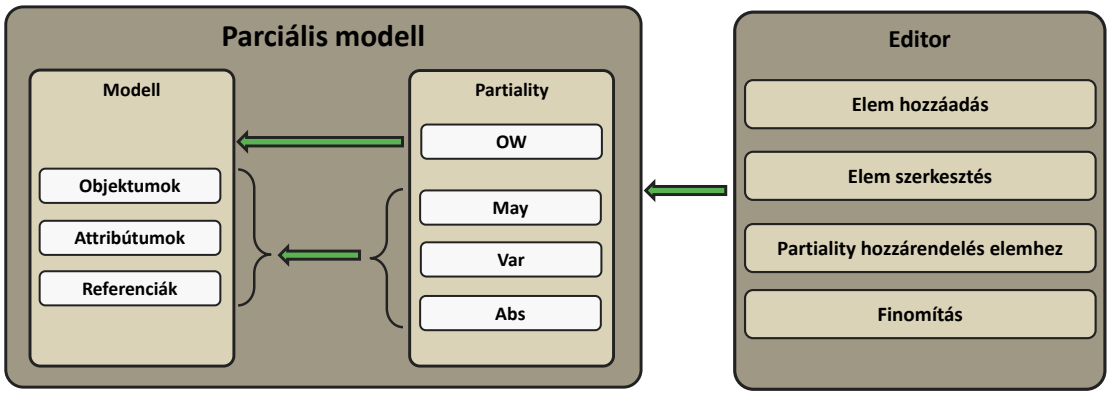


2.5. ábra. Var részlegesség feloldása

3. fejezet

Áttekintés

A cél egy olyan metamodell elkészítése, ami segítségével lehetséges részleges modelleket készíteni. Ehhez olyan vizuális szerkesztőfelület társul, ami megkönnyíti ezt a folyamatot. Ahhoz, hogy ez generikusan működjön egy olyan modell szükséges, aminek segítségével a lehető legtöbb egyéb modell kifejezhető. Így ez a metamodell tartalmazni fog objektumokat, attribútumokat és az ezek közti kapcsolatot kifejező referenciákat. Ezen felül a részlegesség kifejezésére minden ilyen elemhez lehetséges rendelni *May*, *Var* vagy *Abs* részlegességet. Magához a modellhez pedig *OW* részlegességet lehet rendelni. A kapcsolódó editor képes részleges modellt létrehozni és manipulálni. Lehetőséget ad új objektumok, attribútumok, referenciák létrehozására. Ezen elemekhez a már fent említett részlegességek rendelhetők. A szerkesztő a részlegességek feloldására, tehát finomításra is biztosít eszközöket.



3.1. ábra. Részleges modell és a rajta végezhető műveletek

4. fejezet

Megvalósítás

4.1. Szükséges eszközök

4.1.1. Eclipse

A feladat elkészítéséhez Eclipse-et használtam. Ez egy szabadon bővíthető nyílt forráskódú szoftverkeretrendszer. Eclipse-et a bele integrált fejlesztői környezetek segítségével nagyon sok mindenre lehet használni. Azért ezt választottam mert a modellezés általam használt részeihez is biztosít megfelelő keretrendszereket.

4.1.2. EMF

EMF az Eclipse Modelling Framework rövidítése. Ez egy olyan keretrendszer, mely az eclips-be beépülve teszi lehetővé modellek készítését. Segítségével könnyedén grafikus eszközökkel lehet metamodelleket alkotni, melyből később konkrét példánymodellt készíthetünk. Továbbá lehetőséget biztosít metamodellből java kódot generálni, ami leképezi a modell elemeit osztályokba és a hozzá tartozó kapcsolatokat és attribútumokat is.

4.1.3. Java

Objektum orientált programozási nyelv. Szerepe a kódgenerálásnál van. Az EMF modellből kigenerált kódot futtatva egy új Eclipse alkalmazás indítható, melyben már lehetőség van a metamodellben definiált modell példányosítására.

4.1.4. Sirius

Ez egy szintén Eclipse-be épülő keretrendszer. Ez lehetővé teszi hogy EMF modellekhez vizuális megjelenítő és szerkesztő felületet készítsünk. Rendkívül sokrétű lehetőséget nyújt. Lehetőség van új objektumok létrehozására szerkesztésére és biztosít validációs lehetőségeket is. Az egyes elemek módosítását megszorításokhoz lehet kötni. Egy úgynevezett ”viewpoint specification” projektet kell létrehozni, majd ezen belül egy ”odesign” kiterjesztésű fájlt. Ki kel választani milyen modellel szeretnénk dolgozni és azután lehet a modellhez tartozó editor működését és megjelenését definiálni.

4.1.5. Aql

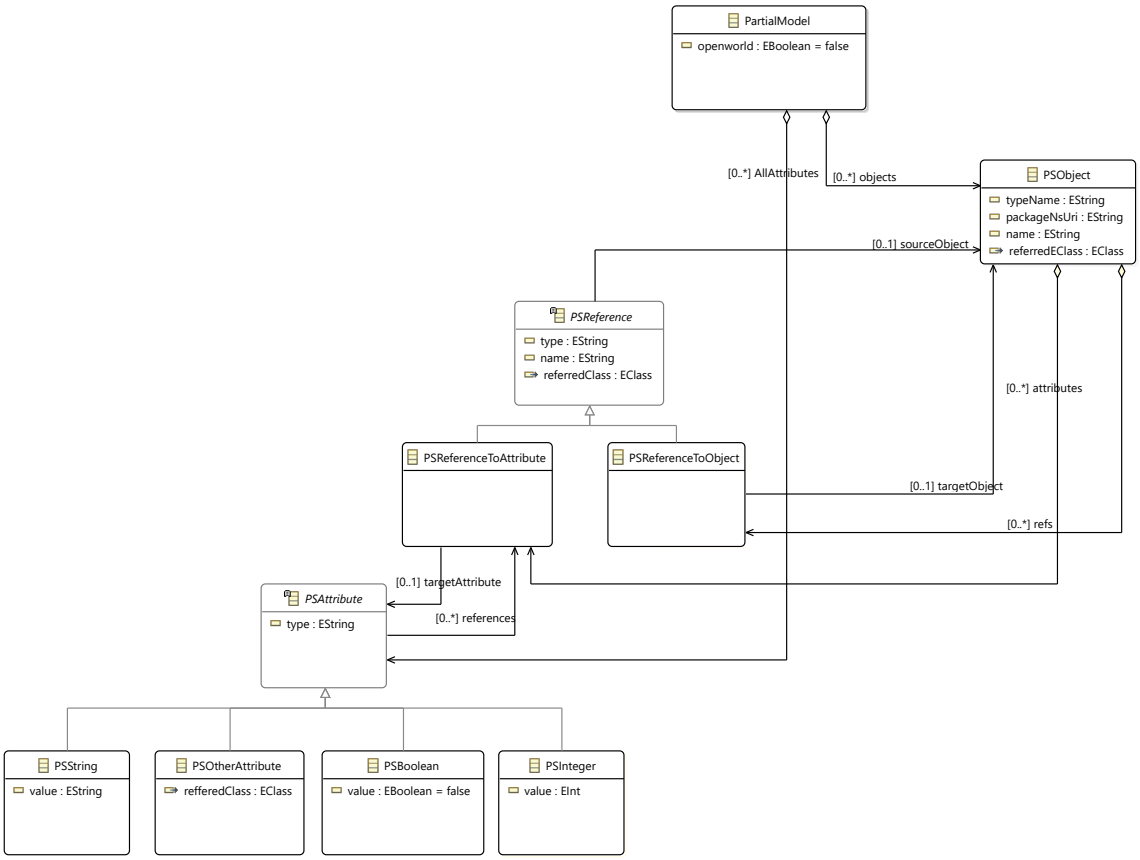
Az Annotation Query Language rövidítése. Lamda kifejezésekhez hasonlóan lehet alkalmazni. Az editor elkészítésébe van nagy szerepe. Az egyes funkciókat előfeltételeit lehet leírni ezen a nyelve.

4.2. Metamodell szerkezete

4.2.1. Általános modell

Jellemzés

Egy általános modell elemekből áll, amik tulajdonságokkal rendelkeznek, az elemek között pedig kapcsolatok vannak. Tehát olyan meta modellt kell készíteni, ami mindezeket magába foglalja. A metamodell elkészítéséhez EMF-et használtam.



4.1. ábra. Általános modell, ami legtöbb modell metamodelljének tekinthető

PartialModel

A fenti modellben a PSObject-ek reprezentálják az általános modell elemeit. Ez bármilyen típusú lehet. A PSObject attribútumokat (PSAttribute) tartalmaz, amik szintén bármilyen típusban és számosságban megjelenhetnek.

PSReference

Az elemek közötti kapcsolatokat reprezentálják. Fontos, hogy ezek is külön elemként jelenjenek meg a modellben, hisz ehhez is szeretnénk majd részlegességet társítani annotációk formájában. A PSReference egy absztrakt objektum két leszármazottja van a PSReferenceToAttribute és PSReferenceToObject. Erre azért volt szükség mert későbbiekben a szerkesztő elkészítésénél más funkcionalitások vonatkoznak rájuk.

PSAttribute

Az objektumok tulajdonságai külön egy PSAttribute nevű elemben vannak hozzárendelve az objektumokhoz, így lehetőség van az attribútumokhoz is részlegességet rendelni. Lehetséges továbbá az is hogy egy attribútum több objektumhoz is tartozzon, ugyanis az attribútumokat nem közvetlenül a PSObjeckt-ek tárolják hanem csak hivatkoznak rájuk. PSAttribute is absztrakt ezért több fajtája lehet:

- PSString
- PSBoolean
- PSInteger
- PSOtherAttribute

4.2.2. Kiegészítés részleges modellé

OW részlegesség

Az *OW* részlegességet a modell gyökerében egy boolean változóval tudjuk szemléltetni. Fenti metamodell alapján lehetőségünk van részleges modellek készítésére.

PSType

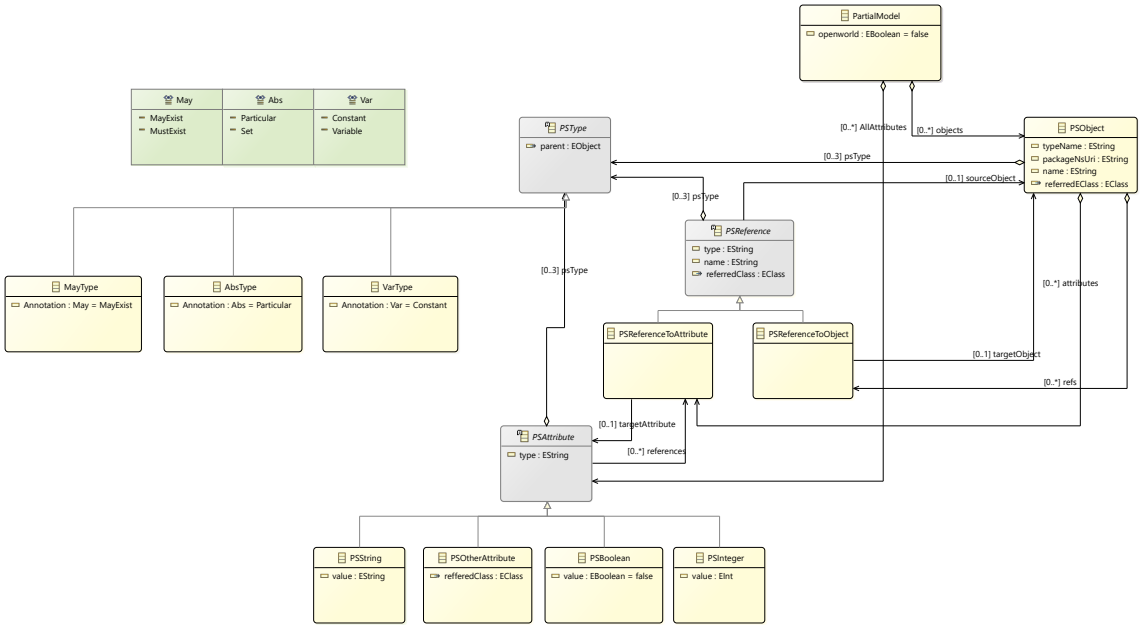
Minden egyes modellbeli elemhez tudnunk kell társítani annotációkat. Ezt a PSType-al tehetjük meg, amiből leszármaznak a már említett részlegesség fajták (*May*, *Var*, *Abs*):

- MayType
- AbsType
- VarType

4.2.3. Példánymodell felülete

Editort Sirius segítségével készítettem el. A PSObject-ek

4.2.4. Funkciók



4.2. ábra. Általános modell kiegészítve részleges modellekre jellemző tulajdonságokkal

5. fejezet

Összefoglalás és továbbfejlesztési lehetőségek

Függelék