



MOF 2.0 Query/ View/ Transformation

7. Lekcija

Ievads MOF 2.0 QVT standartā ar akcentu uz
Relāciju kartēšanām

Ērika Asņina

Vēsture

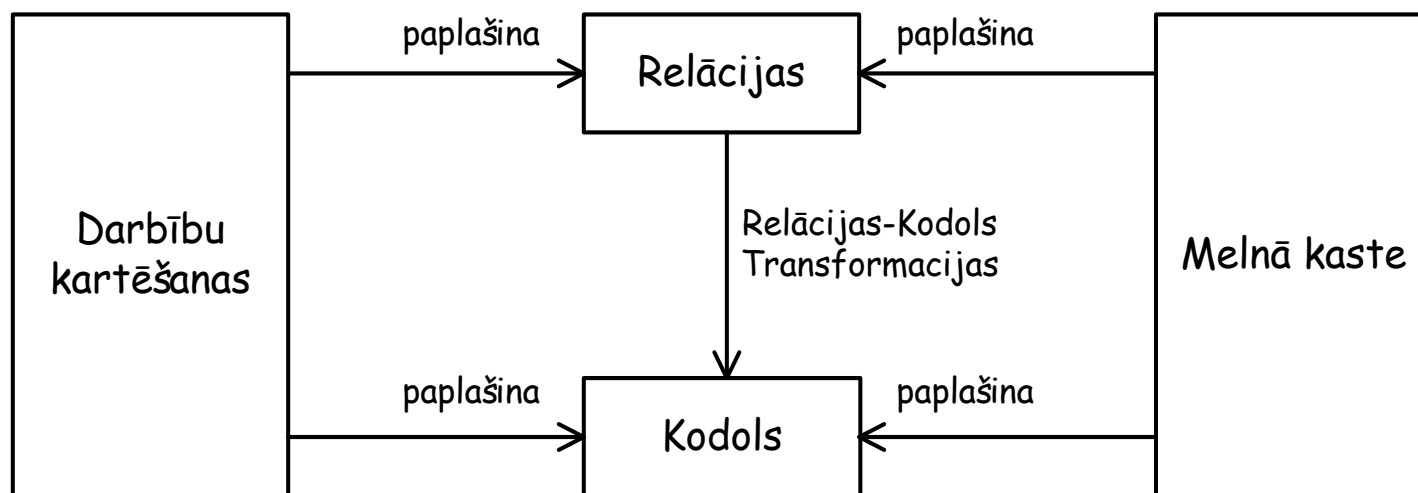
- QVT atšifrējas kā Query/Views/Transformations
- OMG grupas standarta valoda uz MOF modeļiem pamatotiem *vaicājumu uz modeļiem, skatu uz metamodeļiem*, un *modeļu transformāciju* izteikšanai
- OMG QVT Request for Proposals M2M transformācijām tika izsludināts 2002. gadā
- 8 dažādas grupas iesniedza savus priekšlikumus, kuru pirmais apvienošanas mēģinājums tika izdots 2005. gadā
- Tekošais statuss (2007. gada novembris): versija 1.0 pabeigta, saskaņošanā atrodas versijas 1.0 Beta 2 specifikācija

Prasības M2M valodai

- Transformāciju atsekošana
 - Iespēja apskatīt iepriekšējo transformāciju rezultātus
 - Transformāciju atklūdošanas iespēja
- Izmaiņu izplatīšana
- Attiecību starp avotu un mērķi verifikācija
- Ietekmju analīze (impact analysis)
 - Inkrementāla modifikācija (incremental update)
- Saglabāšanas politika (retainment policy)
- $N \times M$ transformācijas
- Divvirzienu transformācijas

MOF 2.0 QVT vispārīga arhitektūra

- Slāņota arhitektūra ar trīs transformācijas valodām
 - Relāciju valoda (Relations)
 - Kodols (Core)
 - Darbību kartēšanas (Operational Mappings)
- “Melnās kastes” mehānisms ārējo programmu izsaukšanai transformāciju izpildes laikā



QVT valodas

- Relācijas (Relations)
 - Deklaratīva transformāciju valoda
 - Modeļu elementu attiecību specifikācija
- Kodols (Core)
 - Deklaratīva transformāciju valoda
 - Relāciju valodas vienkāršošana
- Darbību kartēšanas (Operational Mappings)
 - Imperatīva transformāciju valoda
 - Paplašina Relāciju valodu ar imperatīvām konstrukcijām

QVT ir trīs valodu kopa, kuras kopā piedāvā hibrīda valodu

Relāciju valodas apskats

- Deklaratīva valoda, kuras pamatā ir attiecības, definētas uz modeļa elementiem metamodelī
- Objektu šabloni, kuri var būt salīdzināti un kuriem var būt izveidoti eksemplāri
- Trasējamības saišu automātiska apstrāde
- Transformācijas ir potenciāli daudzvirzienu
- Atbalstāmie izpildes scenāriji
 - Tikai pārbaude: verificē vai dotais modelis ir saistīts kādā noteiktā veidā
 - Vienvirziena transformācijas
 - Daudzvirzienu transformācijas
 - Eksistējošo modeļu inkrementāla modifikācija

Kodola valodas apskats

- Deklaratīva valoda, kuras pamatā ir attiecības, definētas uz modeļa elementiem metamodelī
- Vienkāršākie objektu šabloni
- Trasējamības saišu manuālā apstrāde
- Izteiksmība ir vienā līmenī ar Relāciju valodu
- Daudzvārdīgs nekā Relāciju valoda
- Atbalsta tādu pašu izpildes scenāriju kopu kā Relāciju valoda
- Pielietošanas veidi
 - Vienkārša transformāciju valoda
 - Atsauces, kuras definē Relāciju valodas semantiku

Darbību kartēšanu valoda

- Imperatīva valoda, kuras pamatā ir domēnam specifiska imperatīva valoda modeļu elementu transformāciju aprakstam
- Vaicājumiem tiek izmantota OCL valoda, izskaitļošanai tā ir papildināta ar imperatīvām īpašībām
- Trasējamības saišu automātiska apstrāde
 - Palīgu (helpers) izmantošana
- Vienvirzienu transformācijas (!)
- Pielietošanas veidi
 - Pilnīgi aprakstīt transformācijas darbību kartēšanas valodā
 - Hibrīda režīms = Relāciju (Kodola) Valoda + Melnās kastes kartēšanas (Darbību kartēšanas valoda)

QVT Terminoloģija (daļēja)

■ Relācija (Relation)

- Definē *likumu*, pēc kura tiek noteikta modeļa elementu konkrētā apakškopa
- Noteic atbilstošo trasēšanas klašu eksistēšanu un atrastām domēna attiecībām trasēšanas eksemplāru atbilstošo skaitu

■ Relāciju transformācija (Relation Transformation)

- Formalizēta ar attiecību sarakstu

■ Relāciju domēns (Relation Domain)

- Tips, kas var būt par *parauga šablona* pirmsākumu

QVT Terminoloģija (daļēja)

■ Parauga šablons (Template Pattern)

- Literāļu kombinācijas, ar kuru var sakrist klašu eksemplāri jeb vērtības
- Balstās uz klašu *īpašību identificēšanas*, kuri ir definēti ar *atslēgām*

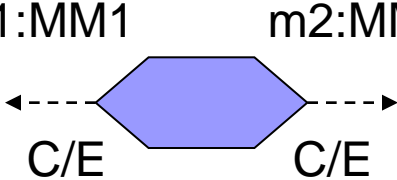
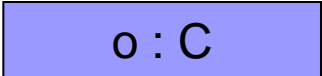
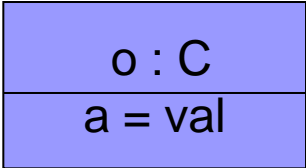
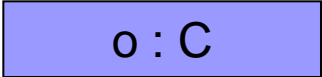
■ Trasēšanas klase (Trace Class)

- MOF klase, ar īpašībām, kuras atsaucas uz ar attiecību saistītiem objektiem transformācijā


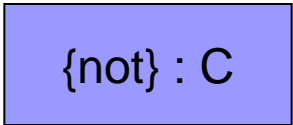

■ Trasēšanas eksemplārs (Trace Instance)

- Savienojums starp modeļiem izveidots transformācijas izpildes laikā

Relāciju valodas grafiskā sintakse

Notācija	Apraksts
	<p>Attiecība starp modeļiem m1, kura metamodelis ir MM1, un m2, kura metamodelis ir MM2. Iezīme C/E norāda vai domēns šajā virzienā ir pārbaudāms (C) vai izpildams (E)</p>
	<p>Objekta paraugs ar tipu C uz kuru atsaucas mainīgais o</p>
	<p>Objektu paraugs ar tipu C un ierobežojumu, ka īpašībai a jābūt vērtībai val. val var būt patvaļīga OCL izteiksme</p>
<p><<domain>></p> 	<p>Domēns attiecībā</p>

Relāciju valodas grafiskā sintakse

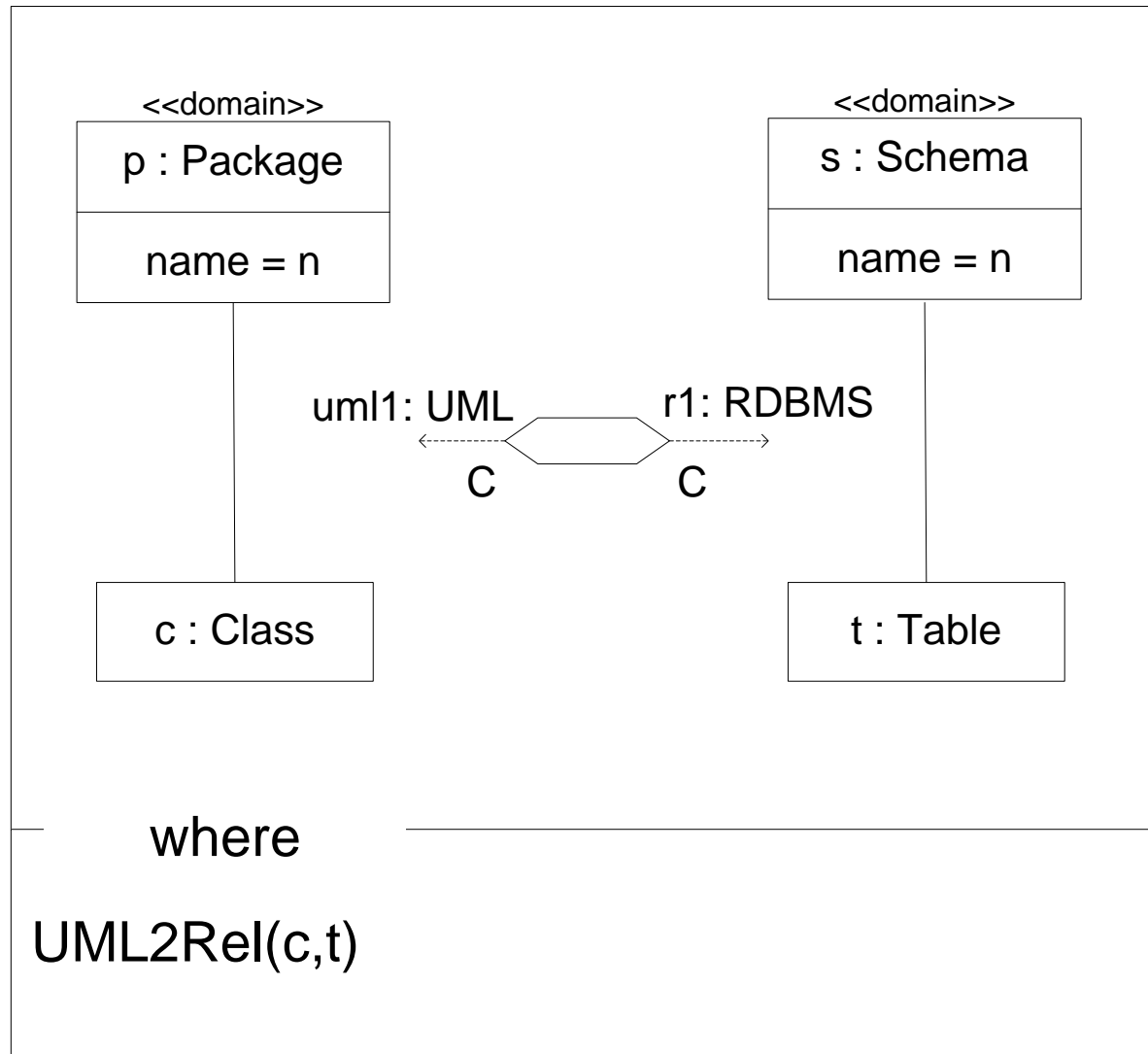
Notācija	Apraksts
	<i>oset</i> ir objektu paraugs kas sakrīt ar tipa C objektu kopu
	<i>not</i> paraugs, kas sakrīt tikai, ja nav tipa C objektus, kuri apmierina ar to asociētam ierobežojumam
	Ierobežojums, kas var būt piesaistīts vai nu domēna, vai nu objekta paraugam

Relāciju valodas papildjēdzieni

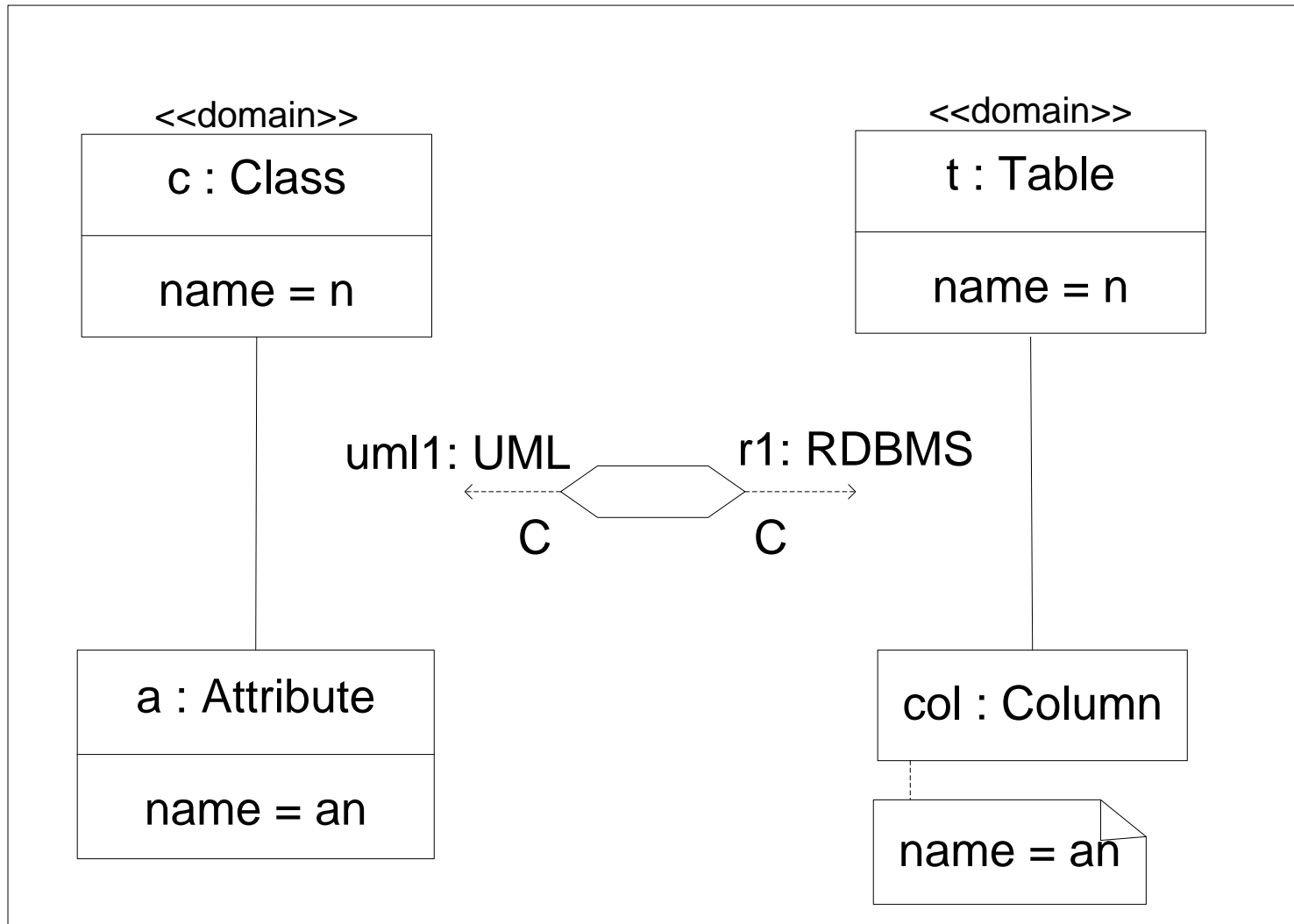
- Pārbaudāms (check-only, C) un izpildāms (enforced, E)
- Where/When nosacījums
 - When – nosacījums zem kura pastāv *attiecība*
 - Where – nosacījums kuru jāapmierina visiem modeļa elementiem, kuri piedalās attiecībā

Piemērs 1

Package2Schema



UML2Rel



Kartēšanas likums

Ar dabisko valodu:

Ja klasifikators ir klase ar nosaukumu “Norekinu konts” un tam ir atribūts ar nosaukumu “debets”, kura tips nav “Currency”, **tad** piešķirt šim atribūtam tipu “Currency”

IF

{S: ?x īpašība datatype ir “UML::Class”} and

{S: ?x īpašība name ir “Norekinu konts”} and

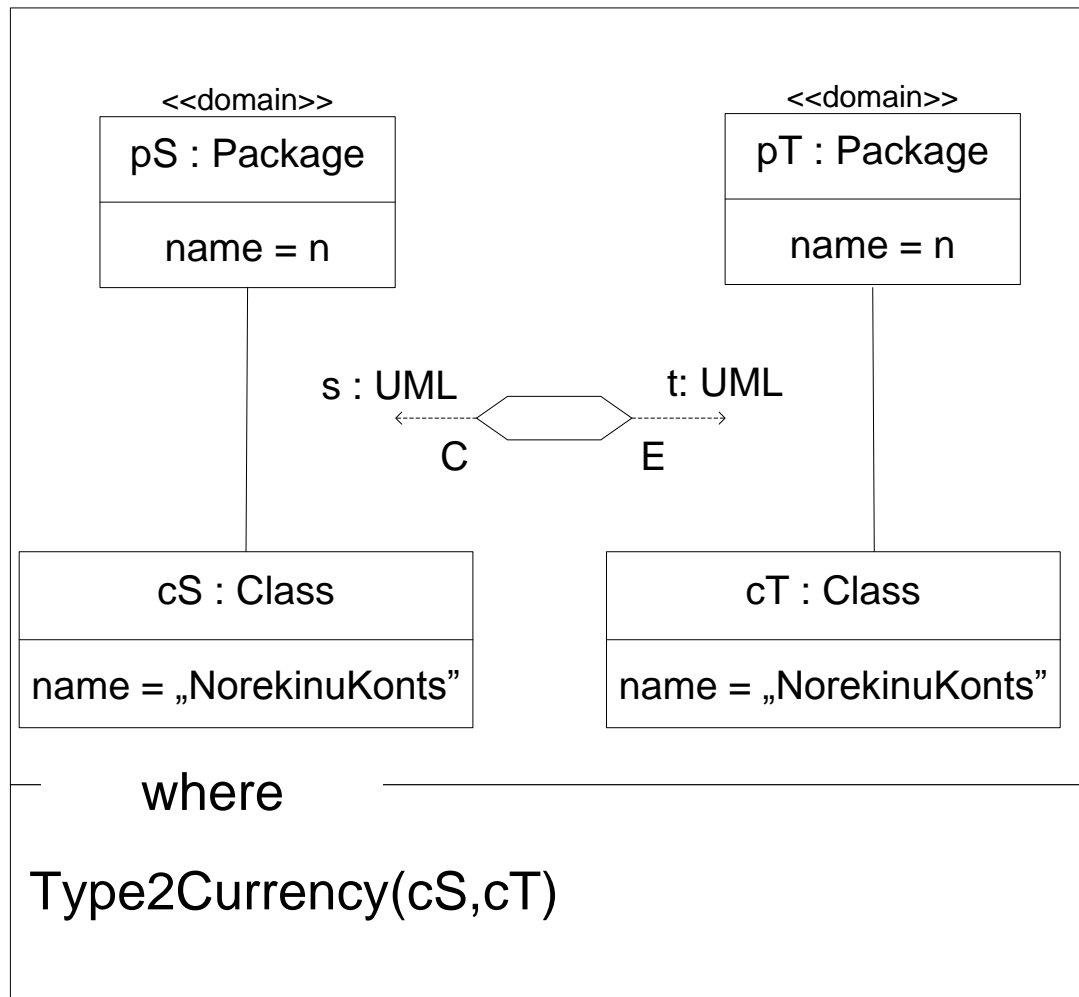
{S: ?x īpašības ownedAttribute īpašība name = “debets” and
īpašība datatype nav “Currency”}

THEN

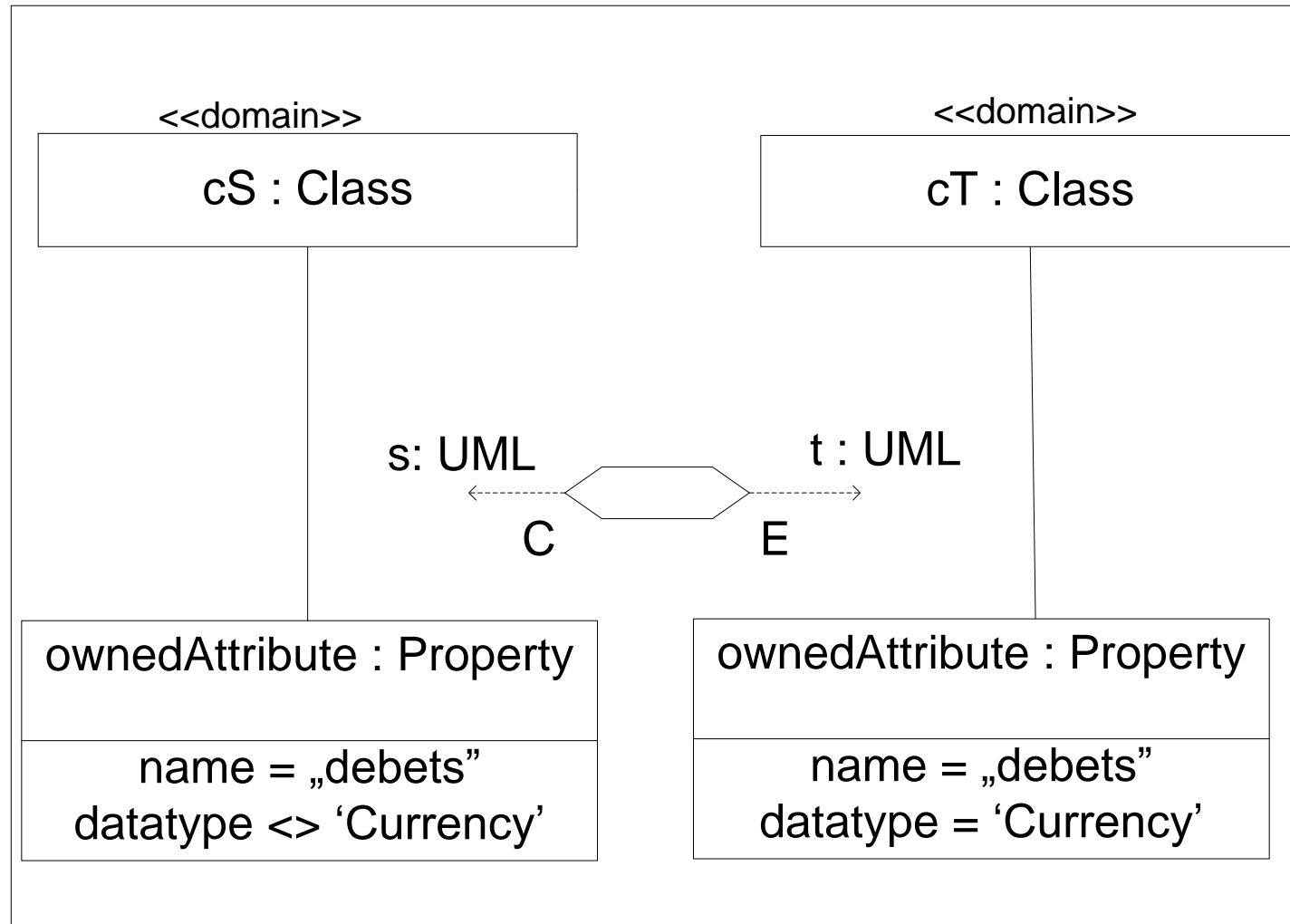
{T: set ?x īpašības ownedAttribute īpašība datatype = “Currency”}

Piemērs 2

SModel2TModel



Type2Currency



Izteiksmju ierobežojumi when/where nosacījumos un domēnā

■ Avota domēns

- $\langle \text{object} \rangle . \langle \text{property} \rangle = \langle \text{variable} \rangle$
- $\langle \text{object} \rangle . \langle \text{property} \rangle = \langle \text{expression} \rangle$

■ Mērķa domēns

- $\langle \text{object} \rangle . \langle \text{property} \rangle = \langle \text{expression} \rangle$