

LATVIJAS UNIVERSITĀTE

VALDIS VĪTOLIŅŠ

**BIZNESA PROCESU MODELĒŠANA, IZMANTOJOT
METAMODELĒŠANAS PAŅĒMIENUS**

Promocijas darba kopsavilkums
datorzinātņu doktora (Dr. sc. comp.) zinātniskā grāda iegūšanai

Nozare: datorzinātnes
Apakšnozare: programmēšanas valodas un sistēmas

Zinātniskais vadītājs:
profesors, Dr. habil. sc. comp.
AUDRIS KALNIŅŠ

Rīga - 2007

Darbs ir izstrādāts ar Eiropas Sociālā fonda atbalstu. Projekts „Doktorantu un jauno zinātnieku pētniecības darba atbalsts Latvijas Universitātē”



Darba vadītājs:

*Profesors, Dr. habil. sc. comp. Audris Kalniņš
Latvijas Universitāte*

Oponenti:

*Profesors, Dr. habil. sc. comp. Jānis Bārzdiņš
Latvijas Universitāte*

*Asoc. Profesore, Dr. sc. ing. Mārīte Kirikova
Rīgas Tehniskā universitāte*

*Profesors, Ph. D. Olegas Vasilecas
Viļņas Gedimina Tehniskā Universitāte*

Darba aizstāvēšana notiks Latvijas Universitātes Datorzinātnes nozares promocijas padomes atklātā sēdē 11.09.2007 LU Matemātikas un informātikas institūtā (Raiņa bulvārī 29), 413. auditorijā.

Ar darbu un tā kopsavilkumu var iepazīties LU bibliotēkā (Kalpaka bulvārī 4).

Padomes priekšsēdētājs

Jānis Bārzdiņš

Saturs

Promocijas darba tēmas aktualitāte un iegūtie rezultāti	4
1 Promocijas darba vispārējais raksturojums	6
2 Biznesa modelēšana	6
2.1 Biznesa metamodeļi.....	6
2.2 Galvenie biznesa jēdzieni un to saistība	6
2.3 Jēdzienu kartēšana dažādās notācijās	6
3 Biznesa procesu mēri	6
3.1 Biznesa procesa modelis	6
3.2 Mēru agregācijas modeļa piemērs	6
3.3 Biznesa procesa metamodelis.....	6
3.4 Biznesa mēru metametamodelis	6
4 UML 2.0 aktivitāšu diagrammas biznesa modelēšanas semantika kā virtuālā mašīna	6
4.1 UML 2.0 aktivitāšu diagrammas apakškopa un ierobežojumi	6
4.2 Vispārīgs UML 2.0 AD un izstrādātās VM apraksts	6
4.2.1 Aktivitāšu diagrammas standarta semantika	6
4.2.2 Izstrādātās virtuālās mašīnas apraksts.....	6
4.2.3 Metamodeļa paplašināšana un modeļu kartēšana	6
5 UML un modeļu transformācijas biznesa procesu definēšanā	6
5.1 Darba plūsmas definēšanas valodas.....	6
5.2 UML aktivitāšu diagrammas pielāgošana darba plūsmas modelēšanai.....	6
5.3 BPMN diagramma kā otra valoda	6
5.4 AD transformācija uz BPMN	6
6 Noslēgums	6
7 Atsauces	6
7.1 Autora publikācijas recenzētos starptautisku konferenču materiālos.....	6
7.2 Citas autora publikācijas (tieši nesaitītas ar promocijas darba tēmu)	6
7.3 Citi darbā izmatotie avoti	6
1 Pielikums	6
1.1 Autora referāti par darba rezultātiem starptautiskās zinātniskās konferencēs vai semināros	6
1.2 Promocijas darbā iekļautās publikācijas un promocijas darba autora personiskais ieguldījums.....	6

Promocijas darba tēmas aktualitāte un iegūtie rezultāti

Tēmas aktualitāte. Darba tēma ir saistīta ar biznesa procesu vadības sistēmu izstrādi. Biznesa procesu vadības sistēmas ir attīstījušās no darba plūsmas pārvaldības sistēmām, tajās pakāpeniski iekļaujot arvien vairāk standarta informācijas sistēmu funkciju, piemēram, resursu un klientu pārvaldību.

Lai pārvaldītu biznesa procesu vadības sistēmu sarežģītību un kāpinātu izstrādes efektivitāti, ir nepieciešams pielietot uz modeļiem balstītu sistēmu izstrādi. Atšķirībā no klasiskām informācijas sistēmām, biznesa vadības sistēmu darbību augstā mērā nosaka to izpildītie procesu modeļi. Līdz ar to arī biznesa procesu modeļus var izmantot biznesa vadības sistēmu projektēšanā un izstrādē, ar secīgām modeļu transformācijām tos pielāgojot konkrētai izpildes platformai.

Biznesa vadības sistēmu funkciju paplašināšanās un nepieciešamība izmantot biznesa procesu modeļus programmatūras izstrādē ir izvirzījušas jaunas prasības gan biznesa modeļiem, gan procesu modelēšanas valodām. Šobrīd nav vienas labākās biznesa procesu modelēšanas valodas, tāpēc dažādus biznesa procesu vadības aspektus noteiktos izstrādes posmos apraksta ar dažādām modelēšanas valodām.

Piemēram, programmatūras izstrādes *de-facto* standartam – UML valodai darbību semantika nav pietiekami detalizēta, lai to lietotu darba plūsmas aprakstam, savukārt biznesa procesu modelēšanas valoda – BPMN nav pietiekami precīza programmatūras izstrādes vajadzībām. Esošo valodu procesu izpildes semantika nav pietiekami precīzi definēta procesu izpildei sadalītās biznesa vadības sistēmās. Tas apgrūtinā modeļu izmantošanu biznesa procesu vadības sistēmu izstrādē, un šobrīd modeļu bāzēta izstrāde ir pielietojama tikai specifiskās biznesa vadības jomās un atsevišķos izstrādes posmos.

Tāpēc ir nepieciešams izstrādāt paņēmieni, kas ļautu aprakstīt dažādus biznesa procesu vadības apgabalus dažādās modelēšanas valodās, kā arī nodrošināt pāreju no vienas valodas citā, gan virzoties no biznesa modeļa uz konkrētu izpildes platformu, gan pārejot uz alternatīvu valodu. Šāds paņemiens ļautu izveidot vienotu rīku platformu, kurā iespējams aptvert visus nepieciešamos biznesa procesu vadības sistēmas darbības aspektus un efektīvi izmantot dažādos sistēmas izstrādes posmos iegūto informāciju.

Šajā darbā biznesa procesu modelēšanas problēmas tiek piedāvāts risināt, izmantojot metamodelēšanas paņēmienus. Metamodelēšana ļauj dažādus biznesa modelēšanas aspektus aplūkot vienotā un vispārīgā formā, tai pat laikā nezaudējot precīzu jēdzienu nozīmi. Izmantojot metamodelēšanas paņēmienus, darbā ir izstrādāta metodika, ar kuru analizētas dažādas modelēšanas valodas, precīzi salīdzinot to jēdzienus, kā arī izstrādāti modeļu izpildes un izpildes laika mērījumu paņēmieni. Izmantojot izstrādāto metodiku, darbā ir izstrādāts ietvars ar dažādu valodu redaktoriem un implementētas modeļu transformācijas, ar kurām konkrētus biznesa modeļus var transformēt konkrētā izpildes platformā.

Darba galvenie rezultāti.

- Izstrādāts "notācijas neatkarīgais" biznesa procesu metamodelis, kas parāda biznesa modelēšanas jēdzienus un to saistību. Šis metamodelis visos turpmākajos pētījumos ir izmantots kā dažādu biznesa modelēšanas valodu "kanoniskā forma".
- Izstrādāta biznesa modeļu jēdzienu kartēšanas metode jēdzienu kartēšanai no viena domēna uz vairākām modelēšanas valodām, kas izmanto līdzīgus jēdzienus (uz "semantiski līdzīgām" valodām). Šis paņemiens ir izmantots par pamatu modeļu transformāciju būvei.

- Izstrādāta precīza UML (*Unified Modeling Language*) aktivitāšu diagrammas (AD) izpildes semantika, izmantojot virtuālo mašīnu. Izstrādāta uz metamodeli bāzēta metodika modeļu raksturlielumu mērījumu definēšanai un mērījumu apstrādei modeļa izpildes laikā. Šāda mašīna der par pamatu procesu imitācijas vai biznesa procesu vadības sistēmas būvei.
- Izstrādāti funkcionāli ekvivalenti populārāko biznesa modelēšanas valodu – UML AD apakškopas profila un BPMN (*Business Process Modeling Notation*) apakškopu metamodeļi, un uz to bāzes izveidoti šo valodu redaktori. Izstrādātas transformācijas valodā MOLA (*MOdeling LAanguage*), kas veic modeļu transformāciju no AD uz BPMN. Šāda transformācija ir viens no biznesa vadības sistēmu modeļu bāzētas izstrādes soļiem. Ar līdzīgu paņēmieni modeļus tālāk transformējot uz BPEL (*Business Process Execution Language*) valodu, tos var izpildīt reālās biznesa vadības sistēmās.

1 Promocijas darba vispārējais raksturojums

Promocijas darbs "**Biznesa procesu modelēšana, izmantojot metamodelēšanas paņēmienus**" ir izstrādāts no 2002. gada līdz 2007. gadam Latvijas Universitātes Fizikas un matemātikas fakultātē un Matemātikas un informātikas institūtā (MII) profesora Audra Kalniņa vadībā. Šis darbs turpina MII biznesa modelēšanas tradīcijas, kas aizsāktas jau 1986. gadā.

Promocijas darba pētījumu galvenie rezultāti ir atspoguļoti četrās publikācijās [1-4] un referēti četrās starptautiskās konferencēs. **Promocijas darbs veidots kā šo četru publikāciju krājums**, apkopojot autora pētījumu rezultātus dažādos biznesa procesu modelēšanas aspektos.

Pētījuma priekšmets – biznesa modeļi, dažādu procesu modelēšanas metodiku un modeļu salīdzināšana, modeļu izpildes analīze, izmantojot metamodelēšanas pieeju.

Pētījuma mērķis – izstrādāt vienotu un precīzu pieeju dažādu biznesa procesu aspektu modelēšanai, procesu izpildei, mērīšanai un pārveidošanai, izmantojot metamodelēšanas paņēmienus.

Pētījuma posmi – promocijas darba pētījumi ir veikti vairākos posmos, sākot ar vispārīgu procesu modelēšanas aspektu analīzi, turpinot ar precīzu valodu jēdzienu semantikas un izpildes analīzi. Tālāk, izmantojot valodu jēdzinu precīzo semantiku, ir izstrādātas transformācijas, ar kurām biznesa procesa modeļus var pārveidot uz biznesa vadības sistēmās izpildāmu valodu. Kopsavilkuma tālākajās nodaļās ir īsi izklāstītas pētījumu galvenās problēmu nostādnes un iegūtie rezultāti, kas publikācijās ir aprakstīti detalizēti.

□ Pirmajā nodaļā "**Biznesa modelēšana**" tiek piedāvāts vispārīgs biznesa procesu metamodelis, kas tiek izmantots jēdzienu kartēšanai valodās ar līdzīgiem jēdzieniem. Darbā tiek veikta esošo biznesa procesu modelēšanas valodu un biznesa modelēšanas ietvaru analīze. Izpētē ir ņemta vērā gan biznesa procesa darbināšanai nepieciešamā informācija, gan biznesa procesu apkārtne, kas nepieciešama biznesa procesa skaidrošanai un mērījumiem. Esošo metodiku biznesa procesu jēdzieni un to savstarpējā saistība ir analizēti, kā jaunu paņēmieni izmantojot metamodelēšanas metodiku. Izmantojot biznesa procesu metamodeli, tiek piedāvāta metode, kurā biznesa modelēšanas valodas, kas izmanto līdzīgus jēdzienus, attēlo kā "notācijas neatkarīgā" metamodeļa skatus. Izstrādātais vispārīgais biznesa procesu metamodelis visos turpmākajos pētījumos ir izmantots kā biznesa procesu definīcijas "kanoniskā forma".

□ Otrajā nodaļā "**Biznesa procesu mēri**" tiek piedāvāts jauns biznesa procesu mērījumu definēšanas paņēmiens un ir izstrādāts jauns, uz metamodeli balstīts, biznesa procesu mērījumu ietvars. Izstrādātais paņēmiens ļauj biznesa procesu mērus definēt integrēti kopā ar biznesa procesa definīciju un nosaka procesa elementu iespējamās mērus. Ir dots īss esošo biznesa procesu mērīšanas paņēmieni apraksts, un ir secināts, ka esošie mērīšanas paņēmieni nav universāli un ka tie ļauj lietotājam definēt mērus bez praktiskas nozīmes. Tālāk tiek piedāvāta metodika gan biznesa procesa elementu mērījumu definēšanai, gan iespējamo mēru ierobežojumiem, gan arī šo mērījumu automatizētai apstrādei (agregēšanai) procesa izpildes laikā. Mērījumu definēšanai ir izstrādāts UML AD profils, mērāmo lielumu apstrādes principi ir nodrošināti, papildinot UML metametamodeļi.

- Trešajā nodaļā "**UML 2.0 aktivitāšu diagrammas biznesa modelēšanas semantika kā virtuālā mašīna**" tiek piedāvāta UML standarta aktivitāšu diagrammas izpildes semantikas analīze, kā jaunu paņēmieni izmantojot virtuālo mašīnu. Tiek noteikti biznesa procesu definēšanai nepieciešamie AD elementi un ir precizēta diagrammu izpilde biznesa procesos. Ir izstrādāts jauns, vienkāršāks par oriģinālo, AD izpildes algoritms, kas tomēr nezaudē oriģinālo aktivitāšu diagrammas izpildes nozīmi. Tas ir panākts, izmantojot "grūdošos" un "velkošos" dzinējus, kas pārvieto marķierus pa aktivitāšu izpildes plūsmas ceļiem. Izstrādātais algoritms ir izmantojams aktivitāšu diagrammas imitācijas, kā arī biznesa procesu vadības sistēmas dzinēja izstrādei.
- Ceturtajā nodaļā "**UML un modeļu transformācijas biznesa procesu definēšanā**" tiek piedāvāta uz modeļu transformācijām balstīta biznesa vadības sistēmu izstrādes tehnika. Ir aplūkotas divas pasaulē populārākās biznesa modelēšanas valodas – UML aktivitāšu diagramma un biznesa procesu modelēšanas notācija BPMN (*Business Process Modeling Notation*). Apskatot esošās metodikas un biznesa vadības sistēmas, ir noteikti svarīgākie biznesa procesu modelēšanas aspekti. Uz to bāzes ir izstrādāti UML AD apakškopas profils un funkcionāli ekvivalenta BPMN apakškopa, kā arī izstrādāti šo valodu redaktori. Precīza valodu izpildes semantika ir noteikta, izmantojot jēdzienu pārveidi uz izpildāmu biznesa procesu valodu BPEL. Salīdzinot AD un BPMN valodu metamodeļus, ir pierādīts, ka modeļu pārveide no vienas valodas otrā nav triviāla un ka modeļus visefektīvāk var salīdzināt (un pārveidot), izmantojot modeļu transformācijas. Modeļu transformācijas ir izstrādātas MOLA transformāciju valodā un realizētas ar MOLA transformāciju rīka palīdzību.
- Nobeigumā ir dots pētījumu rezultātu novērtējums jaunāko notikumu kontekstā, kā arī ir aprakstīta ideju praktiskā realizācija modelēšanas rīkos.

Pētījumu teorētiskā un praktiskā nozīme.

Darbā ir izstrādāti teorētiskie principi, kā, izmantojot procesu modelēšanas valodu metamodeļus, šīs valodas var aprakstīt un salīdzināt un kā šo valodu modeļus var pārveidot un izpildīt. Izstrādātais biznesa procesu metamodelis un modeļu kartēšanas paņēmieni ir izmantojami precīzai "semantiski līdzīgu" modelēšanas valodu modeļu pārveidei, bet modeļu transformācijas ļauj iegūt modeļa ekvivalentu arī ļoti sarežģītu atbilstību gadījumā. Izstrādātais uz metamodeli balstītās virtuālās mašīnas modeļu izpildes paņēmieni ir izmantojami dažādu procesu modelēšanas valodu imitācijas vai izpildes sistēmas izstrādei, ļaujot veikt procesu raksturlielumu mērīšanu izpildes laikā. Darbā izstrādātie metamodelēšanas paņēmieni un atziņas ir izmantoti modelēšanas rīku ietvara un uz tā balstītu modelēšanas redaktoru izstrādē, apliecinot metamodelēšanas paņēmieni efektivitāti.

Darba pētījumu rezultāti izmantoti GMF (*Generic Modeling Framework*, [87]) rīka biznesa modeļu redaktoru izstrādē. Izmantojot GMF rīku, ir izveidoti jauni UML AD un BPMN modelēšanas valodu redaktori. Šajos redaktoros izveidotos modeļus var pārveidot no vienas vienas valodas uz otru ar MOLA transformācijas rīka palīdzību.

2 Biznesa modelēšana

2.1 Biznesa metamodeļi

Biznesa procesu modelēšana (saīsināti – BPM) aizsākās kā daļa no biznesa procesu reinženierijas pagājušā gadsimta 90-tajos gados. Sākotnēji biznesa procesu modeļi bija konceptuāli, un tie tika izmantoti programmatūras apraksta vietās, kur diagramma bija uzskatāmāka par tekstu. Lai nodrošinātu viennozīmīgu modeļu izpratni, tika izstrādātas pirmās modelēšanas valodas, modeļu veidošanai tika izstrādāti atbilstoši modelēšanas rīki.

Viena no pirmajām modelēšanas valodām tika definēta t.s. Zahmana ietvarā [21], kas tika ieviests Popkin Software System Architect rīkā. Rīkos tika implementētas arī citas modelēšanas valodas, piem., GRAPES BM – GRADE rīkā [78], ARIS tāda paša nosaukuma rīkā [32], kā arī tādas valodas kā IDEF 3 un UML 1.0 aktivitāšu grafi. Izstrādājot precīzas modelēšanas valodas, parādījās arī modeļu validācijas un imitācijas iespējas (piem., ARIS, System Architect [33], GRADE), un tika izstrādātas pirmās darba plūsmas pārvaldības sistēmas (FileNet, MQ Workflow). Līdz ar to biznesa modelis vairs nebija tikai programmatūras specifikācija, bet gan izpildāma programma, un modelēšanas valodu precīza izpildes semantika kļuva par aktuālu problēmu.

2002. gadā, kad tika uzsākts šis pētījums, biznesa procesu modelēšana bija kļuvusi par plašu nozari un ar BPM sāka apzīmēt biznesa procesu pārvaldību (*business process management*), tādējādi uzsverot gan procesu analīzes, gan izpildes aspektus. Procesu modeļu izpildes nepieciešamība savukārt uzlika stingras prasības modelēšanas valodu semantikā. Daudzās procesu modelēšanas valodas un modelēšanas ietvari biznesa modelēšanu skaidroja katrs savā veidā, tāpēc integrētai uzņēmuma biznesa procesu apstrādei vienota biznesa modelēšanas pieeja bija absolūti nepieciešama.

Pētījuma mērķis bija izstrādāt vienotu biznesa procesu modelēšanas metodiku, kā jaunu paņēmieni izmantojot metamodelēšanu, ko izmantoja modelēšanas valodā UML [11]. Tā laika izplatītākie modelēšanas ietvari bija izstrādāti pirms UML ieviešanas (piem., Zachmana satvars, ARIS) un tāpēc nebija izstrādāti kā metamodeļi. Līdz ar to bija nepieciešams analizēt esošos ietvarus un izstrādāt vispārīgu un saskaņotu biznesa procesu metamodeli.

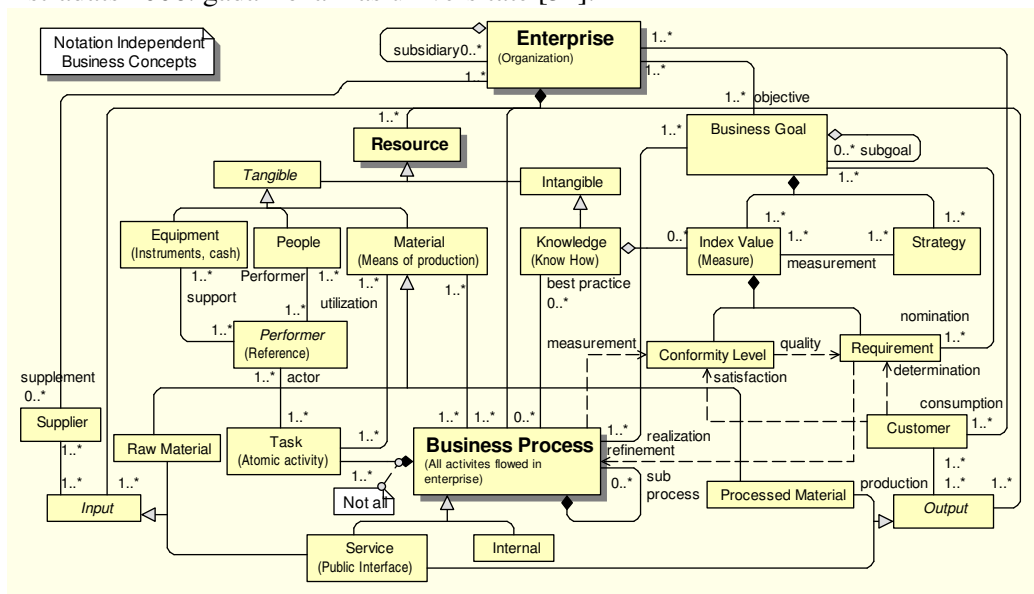
Izmantojot vienotu metamodeli, tiek piedāvāta metode, kurā valodas, kas izmanto līdzīgus jēdzienus ("jēdzieniski līdzīgas" valodas), ir iespējams attēlot kā harmonizēta metamodeļa skatus.

2.2 Galvenie biznesa jēdzieni un to saistība

Lai aprakstītu biznesa procesu, ir nepieciešami divi skati. Viens skats ir nepieciešams procesa izpildei. Tas apraksta pašu procesu – kādas darbības, kādos apstākļos un kādā kārtībā procesu izpilda dažādi veicēji. Otrs skats ir nepieciešams procesa skaidrošanai, kas ir svarīgi biznesa procesu integrēšanai vienā vai vairākos uzņēmumos. Tas parāda biznesa procesa apkārtni – procesa ieejas (piegādātājus) un izejas (pasūtītājus), procesa saistību ar citiem procesiem, tā mērķus, nozīmi un kvalitātes mērus.

Izpētot esošos, gan uz metamodeli bāzētos, gan citus modelēšanas ietvarus [21,22,20,23], tika konstatēts, ka tie ir fragmentāri un ka neviens no tiem nav pietiekami aptverošs un vispārīgs. Tāpēc ir izstrādāts jauns biznesa procesu metamodelis, kurā ir apvienotas izplatītākās ietvaros sastaptās biznesa procesu pārvaldības metodikas. Šajās metodikās sastaptie jēdzieni ir īpaši harmonizēti, apvienotajā metamodelī parādot tikai svarīgākos jēdzienus, kas ir kopīgi visām apskatītajām biznesa procesu pārvaldības metodikām.

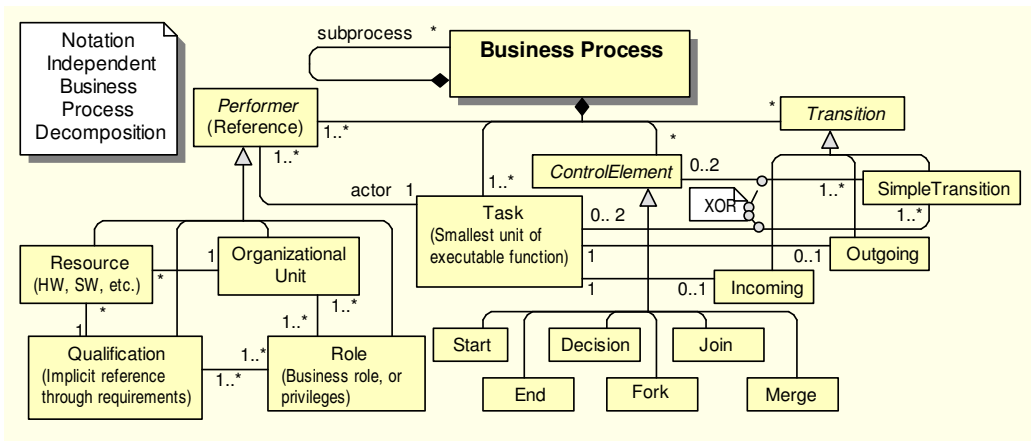
Att. 1 ir parādīts izstrādātais biznesa procesu apkārtnes metamodelis. Tas atbilst vairākiem pasaules vadošajiem biznesa pārvaldības standartiem (piem., pievienotās vērtības ķēdei, ISO kvalitātes standartam [24]). Metamodelis ir līdzīgs OMG izstrādātajam Biznesa motivācijas modelim [49]. Šī standarta izstrādi Biznesa likumu grupa (*Business Rules Group*) uzsāka 2000. gadā, bet metamodelis tika izstrādāts 2005. gadā, kad standarta izstrādi pārņēma OMG, un tas vēl joprojām ir projekta stadijā. Jāpiezīmē, ka procesa resursi, ieejas un izejas autora pētījumā ir aprakstīti detalizētāk kā topošajā OMG standartā. Kā ontoloģija līdzīgs biznesa modelis ir izstrādāts 2006. gadā Lozannas universitātē [54].



Att. 1 Biznesa procesa apkārtnes metamodelis

Att. 2 ir parādīts izstrādātais biznesa procesa sastāvdaļu metamodelis, kas nepieciešams procesa definēšanai un izpildei. Procesā galvenā sastāvdaļa ir uzdevums, kas attēlo vienu nedalāmu darbību. Dažādās biznesa modelēšanas valodās to sauc dažādi: GRAPES BM un BPMN valodās to sauc par uzdevumu (*Task*), UML 1 – par darbības stāvokli (*Action State*), UML 2 – par darbību (*Action*), IDEF 3 – par uzvedības vienību (*Unit of Behavior*). Uzdevumu izpilda veicējs (*Performer*). Pārejas (*Transition*) nosaka, kādā kārtībā uzdevumi tiek izpildīti. Uzdevumi un pārejas ir galvenie biznesa procesa jēdzieni, kas parādās visās procesu modelēšanas valodās.

Vairumā modelēšanas valodu ir arī vadības elementi (*Control Elements*), kas nosaka procesa zarošanos. Izvēle (*Decision*) norāda zarošanās sākumu, sadalīšanās (*Fork*) – paralēlu plūsmu sākumu, savienošanās (*Merge*) – zarojumu apvienošanās un saplūšana (*Join*) – paralēlu zaru apvienošanu. Var būt arī tieši norādīti procesa sākuma (*Start*) un beigu (*End*) punkti. Tomēr vadības elementi, kā arī pāreju savienojumi ar vadības elementiem dažādās valodās atšķiras diezgan krasi. Tāpēc, lai viennozīmīgi attēlotu noteiktu vadības elementu atbilstību analizētajām UML 2.0 AD un GRAPES BM valodām, ir ieviestas papildus pāreju apakšklases – vienkāršā (*SimpleTransition*), ieejošā (*Incoming*) un izejošā (*Outgoing*) pāreja. Šo apakšklašu izmantošana jēdzienu kartēšanai (*mapping*) ir parādīta 2.3. nodaļā.



Att. 2 Biznesa procesa sastāvdaļu metamodelis

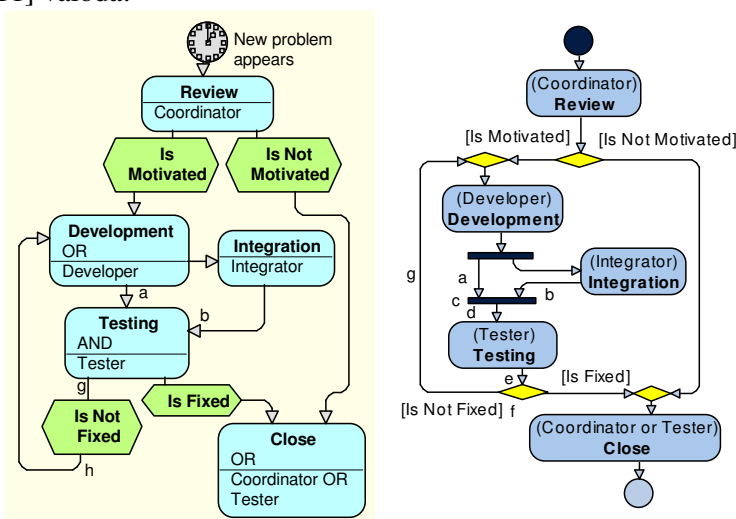
Šis biznesa procesa sastāvdaļu metamodelis ir līdzīgs vēlāk izstrādātajam OMG Processu definīcijas standarta projektam [48]. Jāpiezīmē, ka uzdevuma veicēju modeļa daļa autora izstrādātajā variantā ir detalizētāka kā OMG piedāvātajā projektā.

Izstrādātais harmonizētais jeb "notācijas neatkarīgais" biznesa procesa apkārtnes un sastāvdaļu metamodelis (Att. 1, Att. 2) visos turpmākajos pētījumos ir izmantots kā biznesa procesu metamodeļa "kanoniskā forma" (3., 4. un 5. nodaļā). Lai arī izstrādātais metamodelis ir samērā neliels, tas ir pietiekami vispārīgs, jo parāda ikviena uzņēmuma biznesa procesu galvenos jēdzienus. Tāpēc tas labi saskan ar vēlāk izstrādātajiem starptautiskajiem standartiem.

2.3 Jēdzienu kartēšana dažādās notācijās

Izmantojot "notācijas neatkarīgo" metamodēli, ir izstrādāta metode, kurā "semantiski līdzīgas" biznesa modelēšanas valodas var definēt kā specifiskus vispārīgā metamodeļa skatus.

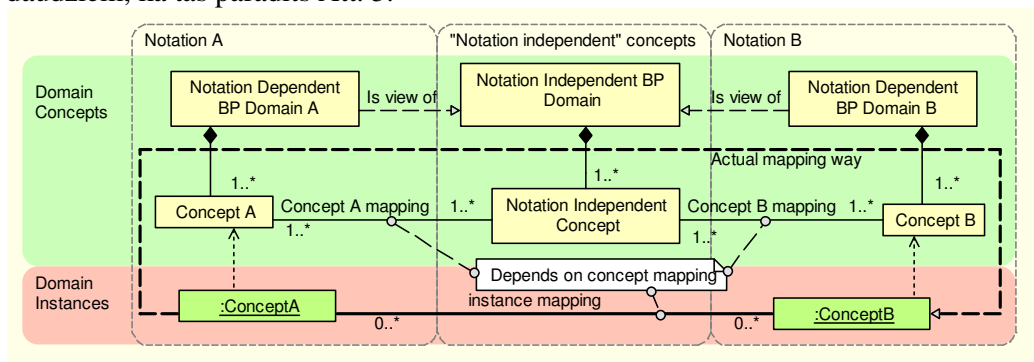
Par piemēriem jēdzienu kartēšanas idejas izklāstam ir izmantotas UML 2.0 aktivitāšu diagramma (tajā laikā kā projekts) un GRAPES BM valoda. Att. 3 ir parādīts viens un tas pats biznesa process abās valodās, parādot šo valodu atšķirības – piem., nosacījumu virsotnes GRAPES BM [78] valodā attēlojas kā nosacījumi pie šķautnēm UML AD [11] valodā:



Att. 3 Biznesa process GRAPES BM un UML AD valodās

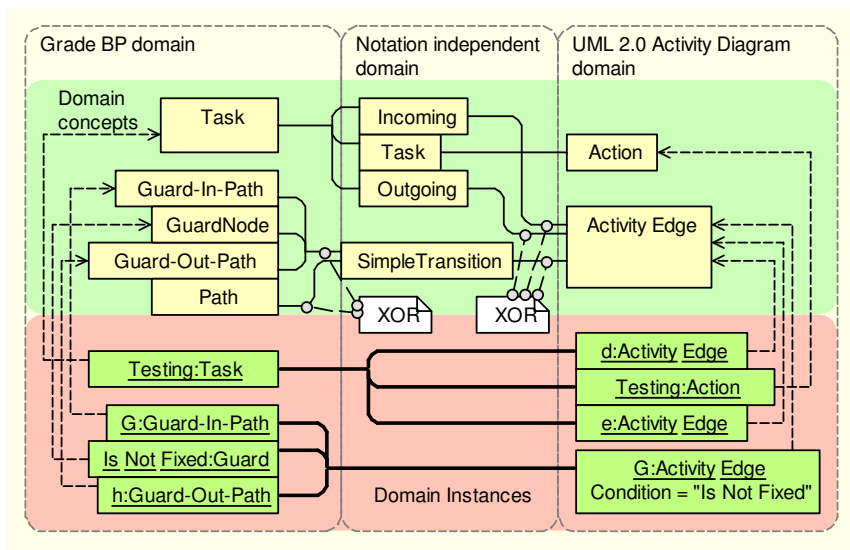
Jēdzienu kartēšanas vispārīgā shēma ir parādīta Att. 4. Attēla augšējā daļa parāda biznesa procesa aprakstīšanai izmantotos jēdzienus (*Domain concepts*). Jēdzieni valodā A (*Notation A*) tiek kartēti uz jēdzieniem valodā B (*Notation B*), kā starpniekus izmantojot "notācijas neatkarīgos" (*Notation independent*) jēdzienus. Daudzos gadījumos saistība starp dažādu valodu jēdzieniem ir daudzi-pret-daudziem, un tā nav pietiekami noteikta un izsekojama. Lai šādu atbilstību "normalizētu", tiek ieviesti papildus starposma jēdzieni, kas daudzi-pret-daudziem atbilstību starp valodu jēdzieniem parāda kā viens-pret-daudziem atbilstību pāri starp abām valodām un starposma jēdzieniem.

Kad kartējums, izmantojot starposma jēdzienus, ir viennozīmīgi definēts, jēdzienu instances vienā notācijā tiek pārvērstas par jēdzienu instancēm otrā notācijā, kā tas parādīts attēla apakšējā daļā. Atkarību līnijas parāda <<instance of>> saistību starp jēdzienu klasēm un to instancēm. Lai arī vispārīgā gadījumā instanču atbilstība ir daudzi-pret-daudziem, konkrētām instancēm šī atbilstība parasti ir viens-pret-daudziem, kā tas parādīts Att. 5.



Att. 4 Metamodeļu kartēšanas vispārīgā shēma

Att. 5 ir parādīts konkrēts kartējuma fragments starp GRAPES BM un UML AD valodām. Attēla augšējā daļa parāda domēna jēdzienus ar GRADE metamodeļa klasēm kreisajā pusē, UML AD metamodeļa klasēm labajā pusē un notācijas neatkarīgā domēna klasēm pa vidu. Vispārīgā daudzi-pret-daudziem atbilstība starp GRAPES BM un UML AD nosacījumu virsotnēm un šķautnēm tiek novērsta, izmantojot vairākas viens-pret-daudziem attiecības starp valodu jēdzieniem un starposma jēdzieniem (*Transition* apakšklases – *Incoming*, *Outgoing* un *SimpleTransition*). Att. 5 apakšējā daļa parāda konkrētas domēna jēdzienu instances atbilstoši Att. 3 parādītajai procesa diagrammai. Kā tas redzams Att. 5, konkrētām instancēm abās valodās atbilstība ir viens-pret-daudziem, tātad, vienkārši izsekojama.



Att. 5 Kartējuma definīcijas un instanču fragments

Lietotāju patiesībā interesē abu modelēšanas valodu grafiskā prezentācija, tāpēc pēdējais solis ir atbilstošo grafisko elementu (šķautņu un virsotņu) izveide diagrammā. Tā kā domēna elementi ar grafisko prezentāciju parasti ir saistīti kā vienspret-vienu, tas ir salīdzinoši vienkāršs uzdevums. Kā tas redzams Att. 5 parādītajā kartējuma fragmentā, kaut arī jēdzienu kartēšana tiek veikta starp "semantiski līdzīgām" valodām, to tehniskie elementi atšķiras pietiekami, lai atbilstošo jēdzienu kartējums būtu diezgan sarežģīts.

Jēdzienu kartēšanas paņēmieni šobrīd izmanto OMG Domēna darba grupa (*Domain task force*), apvienojot UML 2.0 AD un BPMN 1.0 valodas [55]. Tā kā UML AD un BPMN ir vēl lielākas atšķirības, kā UML un GRAPES BM (piem., AD nav ekvivalenta BPMN biznesa procesam, kas satur vairākus procesus, savukārt BPMN nav precīza ekvivalenta AD objektu plūsmi), jēdzienu atbilstību ar kartēšanas paņēmieni vai nu nevar parādīt viennozīmīgi, vai arī ir nepieciešams ieviest daudz starpposma jēdzienu. Tomēr gadījumos, kad "semantiski līdzīgas" valodas izmanto arī līdzīgus tehniskos elementus, jēdzienu kartēšana ir ērts vienkāršojums. Pēdējā laikā šādā veidā ir gūti zināmi panākumi, kartējot BPMN, BPEL, XLANG un WSFL valodas [57].

Vēlākā pētījumā (aprakstīts 5. nodaļā) ir pierādīts, ka sarežģītu jēdzienu atbilstību var efektīvi risināt, izmantojot modeļu transformācijas. Tādā gadījumā kartēšanas asociācijas ir noderīgas divējādi – definīcijā tās uzskatāmi parāda vispārīgo kartēšanas shēmu (kas gan nav viennozīmīga), bet konkrētām instancēm, katrai mērķa instancei parādot avota instanci, tās parāda transformācijas rezultātu, kas ir nepieciešams transformāciju trasējamībai.

3 Biznesa procesu mēri

Biznesa pasaulei globalizējoties, konkurence kļūst arvien asāka, bet biznesa procesi ģeogrāfiski kļūst arvien sadalītāki. Lai noteiktu kopējo procesa efektivitāti, ļoti svarīgi ir noteikt katra atsevišķā biznesa procesa soļa izmaksas, laiku u.c. parametrus. Biznesa procesu efektivitātes mērījumi ir svarīgi uzņēmuma kopējo izmaksu un investīciju atpelnīšanās noteikšanā. Kā to parāda pētījumā veiktā esošo biznesa vadības sistēmu analīze, procesu mērīšanas iespēja ir to neiztrūkstoša sastāvdaļa [44]. Daudzas kvalitātes un biznesa procesu pārvaldības metodikas biznesa procesu stipro un vājo pušu noteikšanai izmanto skaitliskas metodes [28,29,30,31,24]. Metodikas atbalsta dažādi rīki [32,33,34,35], tomēr katrs no tiem piedāvā tikai "savā klasē labāko" metodiku kādam noteiktam biznesa procesa aspektam, nenodrošinot vienlaicīgu dažādu metodiku pielietošanu. Mēru definēšana un to vērtības aprēķināšana, it sevišķi vērtību agregēšana (piem., summēšana, vidējās vērtības, minimuma un maksimuma noteikšana) esošajās darba plūsmas pārvaldības sistēmās un imitācijas rīkos ir sarežģīts uzdevums, kas prasa pamatīgas tehniskās zināšanas. Pētījuma mērķis bija izstrādāt mēru definēšanas un apstrādes ietvaru, kas noteiktu algoritmus un ierobežojumus, kas biznesa procesa elementiem piedāvātu tikai tādus mēru veidus un to agregēšanu, kuriem ir praktiska nozīme. Šādā ietvarā mērus varētu norādīt pašā biznesa procesa modelī. Biznesa procesu analītiķiem tas nodrošinātu ērtu procesa mēru definēšanu un ļautu domāt biznesa jēdzienos, izvairoties no procesa izpildes īpatnībām.

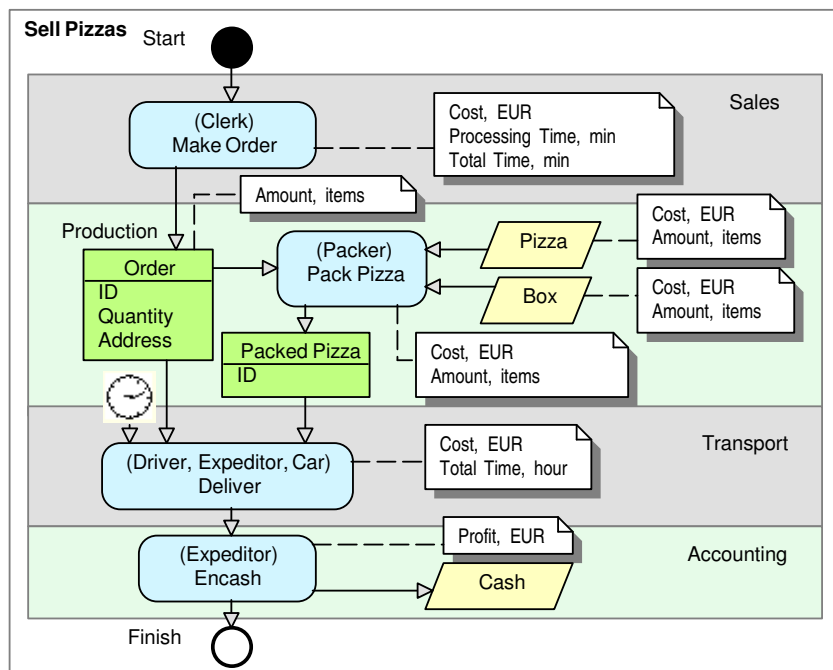
Darbā ir praktiski pierādīts, kā šādu ietvaru iespējams izveidot, izmantojot metamodelēšanas metodiku. Mēru definēšanas valoda ir izstrādāta ar UML [11] aktivitāšu diagrammas profila palīdzību, bet mēru apstrādes ietvars ir izveidots, ar "smagsvara paplašināšanu" paplašinot UML metametamodēli [12].

Izstrādātā pieeja ir aprakstīta, sākot ar konkrētu procesa un mēru definīcijas piemēru un beidzot ar vispārīgu mēru definēšanas ietvaru.

3.1 Biznesa procesa modelis

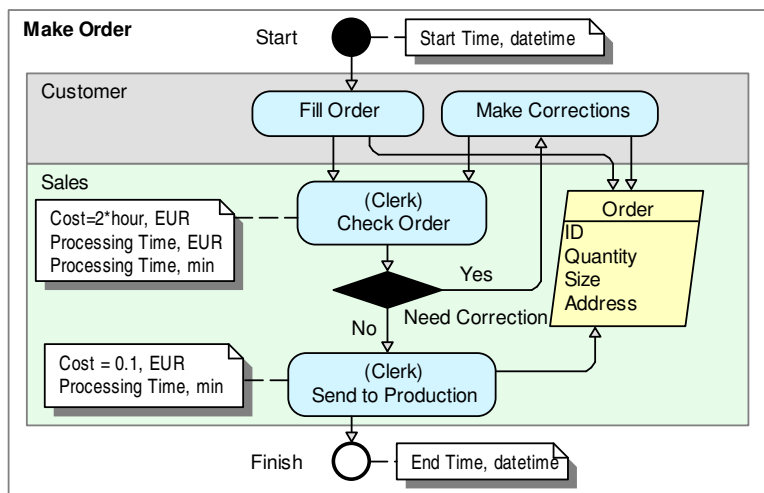
Procesa mēru apstrādes principu demonstrācijai tiek izmantots vienkāršs biznesa process, kas aprakstīts, izmantojot UML aktivitāšu diagrammu (AD) [11]. Att. 6 ir redzams biznesa process veikalam, kas piegādā picas uz pasūtītāju mājām. Picu pārdošana (*Sell Pizzas*) ir biznesa process (aktivitāte AD valodā), kas sastāv no vairākām darbībām (noapaļoti taisnstūri). Organizatoriskā struktūra nav atsevišķi parādīta, to var saprast no procesa apraksta. Organizatoriskās struktūrvienības ir attēlotas ar procesa joslām, un tajās esošie procesu veicēji (amati/lomas un resursi) ir norādīti darbības nodalījumā iekavās. Ar amatiem/lomām norādītie veicēji (*People*) ir aktīvi – tie veic manuālas darbības, resursi ir pasīvi, tie ir nepieciešami uzdevuma izpildei un tiek aizņemti vai iztērēti izpildes laikā. Objektu plūsma ir attēlota ar objektu virsotnēm (taisnstūri), krājumi ir attēloti kā paralelogrami. Procesu mēri ir parādīti ar piezīmēm (taisnstūri ar nolocītu stūri), kas ar raustītu līniju pievienotas mērāmajam objektam.

Saskaņā ar MOF [14] šāds biznesa procesa modelis ir visu faktisko biznesa procesu instanču abstrakcija, tātad atbilst M1 abstrakcijas slānim.



Att. 6 Picu pārdošanas biznesa process attēlots ar UML AD profilu (MI)

Make Order darbība izsauc biznesa apakšprocesu (aktivitāti), kas ir attēlota Att. 7.



Att. 7 Pasūtījuma izveidošanas apakšprocess

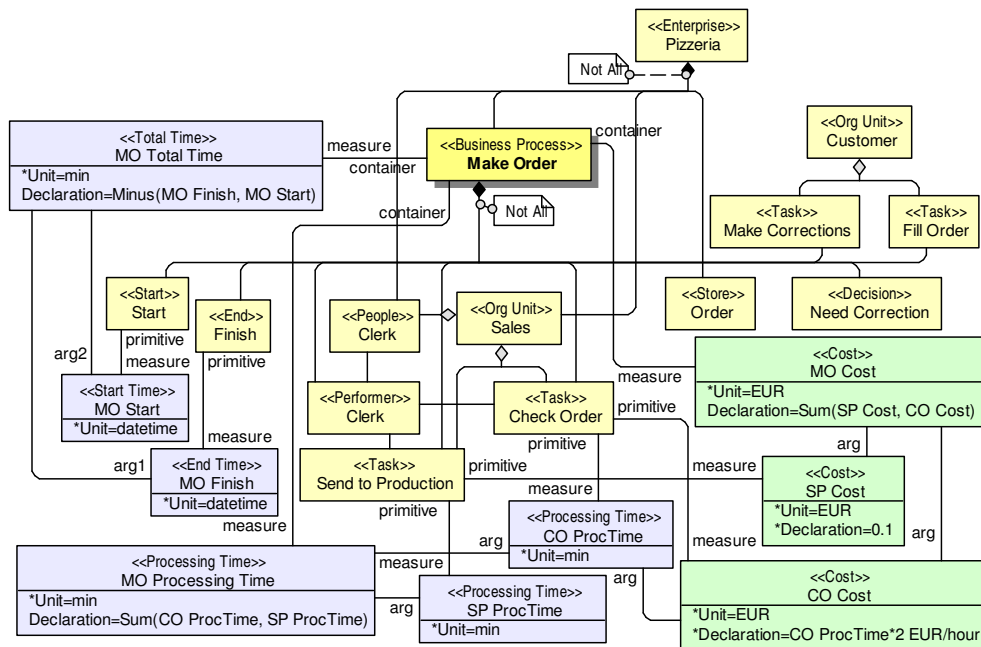
Lai AD nodrošinātu mēru definēšanas iespēju, tā saskaņā ar UML standartu ir paplašināta ar procesa mēru profilu. Procesa mēri faktiski ir metaklases stereotipa atribūti, kurus saskaņā ar UML standartu var attēlot piezīmēs. Attēlojuma uzskatāmībai vairāki viena objekta mēri ir apvienoti vienā piezīmē. Katra mēra deklarācija ir veidota ar sekojošu sintaksi: Veids[=deklarācija], mērvienība (piem., $Cost=2*hour, EUR$). Objektam pievienotais mērs nozīmē, ka procesa izpildes vai imitācijas laikā ir jānosaka dotā objekta parametra skaitliskā vērtība.

Šis piemērs parāda, ka procesu mēru definēšanai (mērīšanas specifikācijai) ir iespējams izmantot ar klašu stereotipiem paplašinātu UML aktivitāšu diagrammu, līdzīgi kā to dara esošajās imitācijas sistēmās (piem., ARIS [32]).

3.2 Mēru agregācijas modeļa piemērs

Ar UML standartam atbilstošu aktivitāšu diagrammas profilu var definēt biznesa procesa mērus, tomēr mēru apstrādei ar to vien nepietiek. Lai parādītu, kā tieši procesa izpildes laikā tiek apstrādātas mēru skaitliskās vērtības un kā tiek veiktas noklusētās darbības, ir nepieciešamas papildus klases un asociācijas. Tāpēc faktiski UML AD ir tikai ietvara ārējā specifikācija jeb saskarne, bet tā implementācija ir krietni sarežģītāka. Ietvara iekšienē aktivitāšu diagramma jātransformē uz īpašu "iekšējo modeli", kurā UML AD bez mēru profila tiek paplašināta ar jaunām klasēm un asociācijām (t.s. "smagsvara" paplašināšana). Tā kā MOLA [80] transformāciju valoda tajā laikā vēl nebija izstrādāta, šādu aktivitāšu diagrammas transformācija ir parādīta ar UML klašu diagrammas līdzekļiem, izmantojot īpašus sintakses paņēmienus.

Att. 8 diagrammā ir attēlota daļa no Att. 6 un Att. 7 definētā biznesa procesa kā ietvara mēru aprēķina modelis. Šis modelis ir "izpildes neatkarīgs", jo neatkarīgi no implementācijas parāda, kādi elementi, asociācijas un darbības ir nepieciešami, lai veiktu uzdoto mēru vērtību apstrādi. Formāli šāds modelis būtu jāattēlo ar divām diagrammām – ar klašu diagrammu, kura apraksta visas noteiktā procesa izpildes vai imitācijas laikā radušās instances (M0), un ar instanču diagrammu tālāk aprakstītajam ietvara metamodelim (M2, Att. 9). Tā kā abas diagrammas parādītu vienu un tā paša objekta dažādas īpašības, attēlojuma ekonomijas dēļ tās ir apvienotas vienā klašu diagrammā, kur šādas duālas "instanču klases" ir parādītas ar īpašu sintaksi. Lai parādītu, ka kāda klase (M1) ir vispārīgākas klases (M2) instance, "instanču klases" ir attēlotas, izmantojot stereotipus. Vispārīgākās klases (M2) vārds ir attēlots kā konkrētās zemākās abstrakcijas slāņa (M1) "instances klases" stereotips.



Att. 8 Mēru agregācijas modeļa piemērs (M1)

Gaišās klases ir UML AD profila stereotipi, kam no metamodela "smagsvara" paplašinājuma parādās papildus agregācijas asociācijas (piem., *Sales* agregācija). Procesu mēri (dažāds tonējums attēlo dažādus mēru tipus), kas ar asociācijām *primitive/container-measure* ir pievienoti pie procesu elementiem, ir AD stereotipu atribūti, kas attēloti kā "instanču klase". Uzskatāmībai mēriem ir piešķirti vārdi, kas sastāv no saistītā elementa nosaukuma vārdu pirmajiem burtiem un mēra tipa

(apstrādei tos nevajag). Ja kāds mērs izmanto citu mēru, atbilstošā asociācija ir ar lomas vārdu *arg*.

No M2 slāņa mantotā procesa mēru kompozīcija M1 slānī ir parādīta saskaņā ar MOF tradīcijām, izmantojot klašu nodalījumus (*compartments*, piem., *unit=EUR*, *declaration= Minus(MO_Finish,_MO_Start)*). Tie nodalījumi, kas ir skaidri definēti biznesa procesa modelī (Att. 6), ir atzīmēti ar zvaigznīti. Atbilstošās mēru asociācijas un ar zvaigznīti neatzīmētie nodalījumi tiek mantoti netieši, veidojot mēru definīcijas metamodeļa instances. Piem., *MakeOrder* biznesa procesa *MOTotalTime* mēra deklarācija automātiski tiek noteikta kā starpība starp procesa *MOSStart* un *MOEnd* mēru vērtībām ar atbilstošām asociācijām.

Lai modelis būtu lasāmāks, pārējām M2 slāņa klasēm kompozīcija arī M1 slānī ir attēlota kā kompozīcija. Piem., *Business Process* un *Enterprise* M2 kompozīcijas instances M1 ir parādītas kā kompozīcija ar atbilstošām klasēm un to stereotipiem.

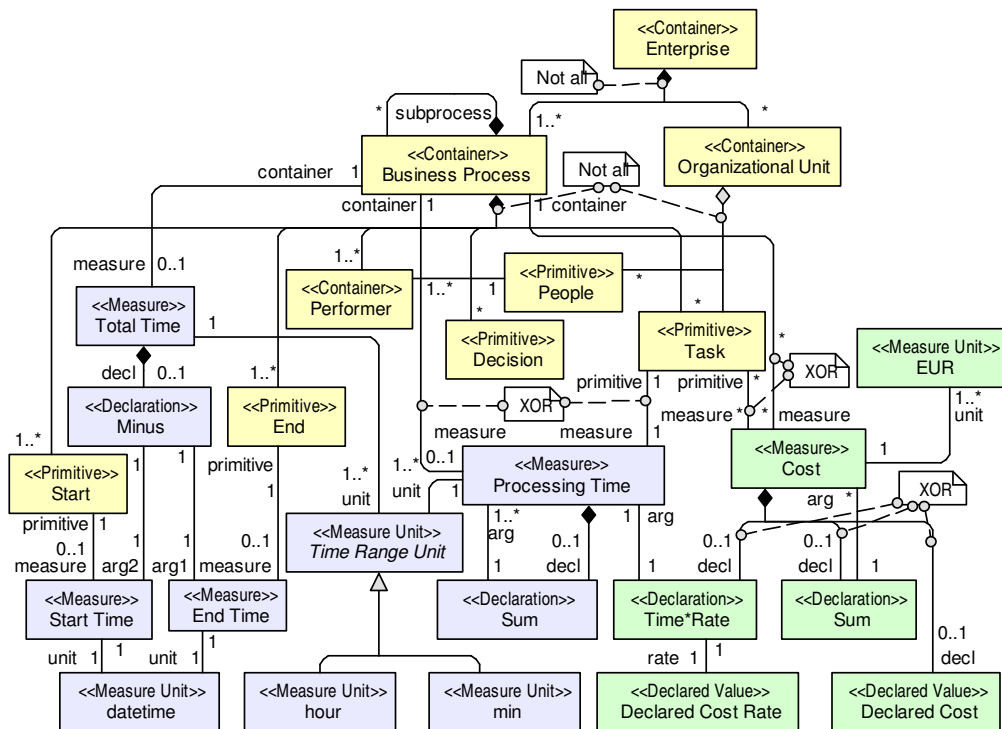
3.3 Biznesa procesa metamodelis

Lai definētu procesa elementu mēru definēšanas un agregēšanas noteikumus, ir izstrādāts īpašs metamodelis (M2), kas kalpo kā ietvars visām metamodeļa instancēm jeb procesu definēšanas modeļiem (M1). Šis metamodelis ir attēlots Att. 9. Metamodelī ir definēta mēru savstarpēja saistība, un kā tie ir saistīti ar biznesa procesa elementiem. Arī šajā attēlā ir sapludinātas divas diagrammas – visu iespējamo modeļu (M1, Att. 8) klašu diagramma un konkrēta metametamodeļa (M3, Att. 10) instanču diagramma.

Šis biznesa procesu mēru metamodelis atbilst iepriekš izstrādātajam biznesa procesu metamodelim (1. nodaļa), tomēr ir arī atšķirības:

- Visas klases ir papildinātā metametamodeļa (M3) instances.
- Lai arī gaišās klases atbilst UML AD profila stereotipiem (piem., *BusinessProces* kā *Activity* stereotips), bez UML definētajām asociācijām šīm klasēm ar papildināto metametamodeli (M3) ir iespējamās papildus agregācijas asociācijas.

UML definētās un papildus izveidotās agregācijas asociācijas tiek izmantotas, lai noteiktu, kuriem elementiem ir iespējama mēru agregācija. Tam varētu izmantot pilnīgi jaunas binārās asociācijas, bet, izmantojot esošās UML asociācijas, tiek samazināts papildus vajadzīgo asociāciju skaits, un metamodelis ir pārskatāmāks. Agregācijas (kompozīcijas) asociācija norāda, ka atbilstošai "saimnieka" elementa instancei iespējams veikt viena veida mēru agregāciju (piem., summa, vidējais, minimums, maksimums) pa visām tās "bērnu" instancēm.



Att. 9 Mēru definīcijas metamodeļa fragments (M2)

<<Measure>> klases (dažādi toņi norāda dažādus mēru tipus) ir klašu metaatribūti (formāli UML 2.0 īpašības – *properties*), kas iznesti kā atsevišķas klases ar asociācijām. Arī šī diagramma ir gan M1 metaslāni aprakstoša klašu diagramma, gan M3 metaslāņa instanču diagramma, tāpēc metaatribūti saskaņā ar ieviesto duālo notāciju ir attēloti kā klases. Šie metaatribūti parāda, kādi mēri var tikt definēti katram procesa elementam (gaišā klase).

Asociācijas starp stereotipiem un metaatribūtiem no papildinātā metametamodela manto īpašas *primitive/container-measure* lomas. Ja mēra deklarācija ir izteiksme (netieši mantota no metamodeļa vai arī konkrēti norādīta modelī), tā ir parādīta kā funkcijas argumentu kopa ar <<Declaration>> stereotipu.

Att. 9 ir parādīti daži iespējamie biznesa procesa elementu mēri. Piem., biznesa procesam (*Business Process*) vai uzdevumam (*Task*) izmaksas var noteikt kā konkrētu vērtību (*Declared Cost*), kā apstrādes laika un izmaksu koeficienta reizinājumu (*Processing Time*, *Declared Cost Rate*) vai arī kā summu vairākām izmaksām.

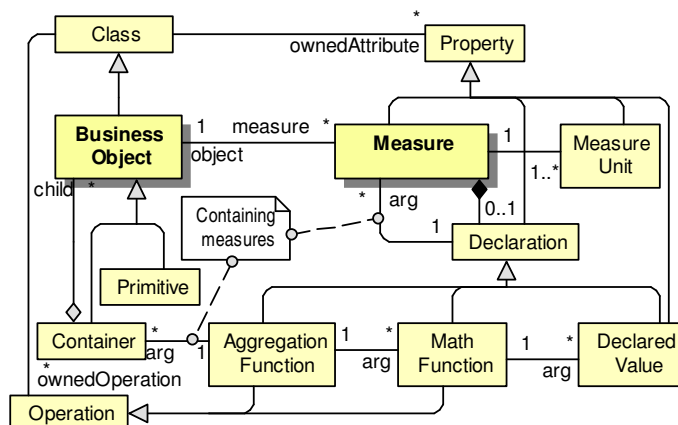
Izstrādātais metamodelis uzskatāmi ar klasēm un asociācijām parāda biznesa procesa modeļu elementu iespējamus mērus, kā arī to, kā mērus iespējams definēt un agregēt, lai mērījumu vērtībai būtu praktiska jēga. Uz šī metamodeļa bāzes veidojot konkrētus modeļus, definētās klases nosaka, kādas mēru instances var definēt procesa elementu instančēm, bet asociācijas nosaka mēru instanču atļautās saites un automātiskās apstrādes veidus.

3.4 Biznesa mēru metametamodelis

Saskaņā ar MOF tradīcijām metametamodelis (M3) tiek veidots pēc iespējas vienkāršāks, un visa konkrētās jomas specifika tiek atspoguļota metamodelī (M2). Tomēr, lai ar vispārīgu metametamodeli (M3) parādītu konkrētā domēna metamodeļa (M2) nozīmi un specifiku, nepieciešams izmantot daudzus OCL ierobežojumus. Tāpēc tiek izmantota cita (pilnīgi legālu saskaņā ar MOF standartu) pieeja – UML

metametamodeļa paplašināšana ar specializētām klasēm un asociācijām, kas saprotami parāda biznesa procesu mērus un to saistību.

Izstrādātais metametamodelis ir redzams Att. 10. Tā galvenā īpatnība ir jaunas biznesa objektu (*BusinessObject*) un mēru (*Measure*) metaklases, kas specializē *InfrastructureLibrary::Constructs* pakotnes *Class* metaklasi. Šāds specializēts metametamodelis atvieglo biznesa procesu mēra metamodeļa izveidošanu, jo, kā redzams Att. 9, pat neliels M2 fragments arī ar specializētu M3 ir pietiekami sarežģīts. Ar standarta UML MOF un OCL [13] ierobežojumiem M2 izveidot būtu daudz grūtāk, un tas būtu daudz nepārskatāmāks.



Att. 10 Biznesa procesu mēru metametamodelis (M3)

Izstrādātajā procesa mēru metodikā ir parādīts, kā iespējams efektīvi izmantot divus UML paplašināšanas veidus. Biznesa procesu mēru definēšanai ir izmantota UML aktivitāšu diagramma, papildināta ar biznesa procesu mēru profilu, attiecīgos mērus parādot kā stereotipu atribūtus. Šādā veidā ir nodrošināta procesa mēru definīcijas savietojamība ar UML. Savukārt mēru apstrādes ietvars pārskatāmi un efektīvi ir izveidots, paplašinot UML metametamodeli. Tādējādi ir panākts, ka mēru ierobežojumus un to agregēšanas loģiku var definēt kā pārskatāmas klases un asociācijas starp mēriem un mērāmajiem elementiem procesa metamodelī.

Paši par sevi biznesa procesu mēri ir tikai definīcijas, kas nosaka, kā konkrēts biznesa procesa elements tiks mērīts. Faktiskās mēru vērtības ir zināmas tikai procesa izpildes laikā. Lai procesa definīciju izpildītu, ir nepieciešams noteikt precīzu mēru nozīmi neatkarīgi no procesa izpildes veida. Procesu vadības sistēmu izpilde ir detalizēti analizēta nākošajā pētījumu posmā, kas aprakstīts 4. nodaļā (gan nedetalizējot procesa mērīšanas aspektu).

4 UML 2.0 aktivitāšu diagrammas biznesa modelēšanas semantika kā virtuālā mašīna

2004. gadā UML 2.0 standarts [11] bija nobeiguma apstiprināšanas stadijā un tā aktivitāšu diagramma (AD) tika ieteikta arī biznesa procesu modelēšanai. Tāpēc precīza AD izpildes semantika biznesa procesos bija aktuāls temats.

Pētījuma mērķis bija padarīt AD lietojamu precīzu (t.i., izpildāmu) biznesa procesu modelēšanai. Lai to veiktu, bija nepieciešams noteikt precīzu AD izpildes semantiku un izvēlēties atbilstošu izmantojamo elementu apakškopu, kas būtu pietiekama biznesa procesu modelēšanai, modeļu validācijai, imitācijai un izpildei.

Pētījumā ir parādīts, ka UML standartā definētā oriģinālā AD semantika izpildāma biznesa modeļa definēšanai nav aprakstīta pietiekami detalizēti un ir ļoti sarežģīta. Tāpēc ir izstrādāta jauna pieeja, skaidrojot esošo AD semantiku ar aktivitāšu diagrammas virtuālo mašīnu (ADVM). Semantikas skaidrojumam ir izvēlēta minimāla AD elementu apakškopa, kas ir nepieciešama biznesa procesu definēšanai. UML AD formālais definīcijas modelis ir pārveidots uz vienkāršāku un daudz ērtāku izpildes modeli, kas tai pat laikā atbilst oriģinālajai AD izpildes semantikai izvēlētajā elementu apakškopā.

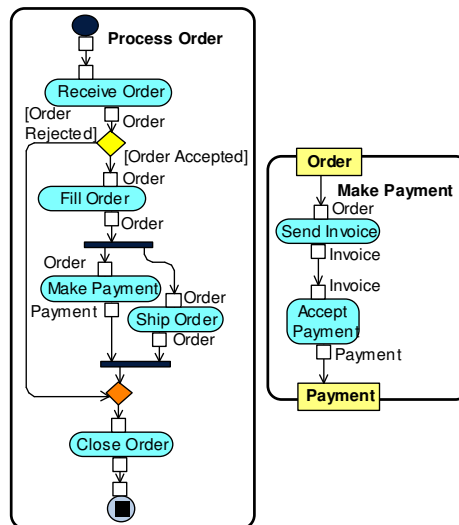
Šāda maksimāli pietuvināta pieeja oriģinālajai AD semantikai, lai arī nenodrošina formālu matemātiskās analīzes iespēju kā pilnīgi formālas algebriskās metodes (piem., Petri tīkli [37]), tomēr dod iespēju pilnīgi precīzi analizēt modeļa uzvedību. Papildus izveidotā virtuālā mašīna var kalpot par pamatu praktiskai imitācijas rīka vai darba plūsmas pārvaldības sistēmas būvei.

4.1 UML 2.0 aktivitāšu diagrammas apakškopa un ierobežojumi

Tā kā faktiski UML 2.0 aktivitāšu diagrammā ir ļoti daudz jēdzienu, kuru izpildes semantika ir optimizēta iegulto sistēmu projektēšanai, ir izvēlēti tikai tie, kas nepieciešami AD darbības skaidrošanai tieši biznesa modelēšanā. AD modeļa darbību nosaka marķieri, kas plūst no virsotnes uz virsotni caur šķautnēm no aktivitātes sākuma uz beigu punktu. Dažādi marķieri tiek izmantoti arī datu pārsūtīšanai no vienas aktivitātes darbības uz citu. Tāpēc, nosakot nepieciešamos elementus, ir izvēlēti tikai tie, kas tieši ietekmē marķieru kustību. Praktiski tās ir darbības un kontroles virsotnes, kā arī šķautnes ar nosacījumiem, kas tās savieno.

Ir vairāki darbi, kuros ir izvērtēti tikai AD izpildes vadības plūsmas (*control flow*) aspekti [36,37,38]. Apskatot AD vadības plūsmu atsevišķi no datu plūsmas [39], ir secināts, ka AD nav "darba plūsmas pilna". Tomēr autora pētījumā ir praktiski pierādīts, ka AD datu un vadības plūsmu apskatot integrēti, AD ir pielietojama precīzai un formālai biznesa procesu modelēšanai.

Att. 11 ir parādītas divas aktivitāšu diagrammas, kas ilustrē visus izvēlētos apakškopas diagrammas elementus. Galvenais AD process (*Process Order*) izsauc apakšprocesu (*Make Payment*). Galvenais process sākas ar sākuma virsotni, tālāk process iet caur izvēles, sadalīšanās, darbības, apvienojuma un saplūšanas virsotnēm un beidzas beigu punktā. *MakePayment* darbība izsauc apakšaktivitāti, kas sākas un beidzas ar aktivitātes parametru virsotnēm. Visas šķautnes (vadības un objektu plūsmās), kas iet no darbības uz darbību, no sākuma uz beigu virsotnēm ir pievienotas piespraudēm (*pins*) – kā tas ir pieprasīts izvēlētajā apakškopā. Ieguvums, izmantojot piespraudes, ir tāds, ka šajā gadījumā ir labi zināma marķieru atrašanās vieta, kā tas ir arī Petri tīklos.



Att. 11 Aktivitāšu diagrammas piemērs, kur "Process order" aktivitāte izsauc "Make Payment" aktivitāti

Tā kā zarošanās diagrammās tiek parādīta atklāti ar izvēles un sadalīšanās virsotnēm, no piespraudēm drīkst iziet un tajās drīkst ieiet tikai viena šķautne. Izvēles virsotņu izejošo šķautņu nosacījumi ir savstarpēji izslēdzoši un laikā nemainās. Bez tam tiek aizliegti sekojoši (nestandarta) slēgumi:

- Vadības virsotnes izejošā šķautne nevar būt tās pašas virsotnes ieejošā šķautne. Tas novērš vadības virsotņu strupsaķeres (*deadlocks*), kur vadības virsotne ieejā gaida marķieri, kas tai pašai ir jārada izejā.
- Starp darbību izsaukuma, sākuma, beigu un aktivitāšu parametru virsotnēm nedrīkst būt ceļi, kas satur gan sadalīšanās, gan saplūšanas virsotnes. Šis ierobežojums ir dabīgs no praktiskā viedokļa, jo nav vajadzības veidot paralēlus zarus, ja tie tiek vienkārši sapludināti bez jebkādam darbībām šajos zaros (t.i., zaros nav darbību izsaukuma virsotņu).

Šie ierobežojumi principā neierobežo dabīgu biznesa procesu veidošanu, bet to rezultātā starp dažādām darbībām nenotiek "cīņa par marķieriem" un ir izslēgta nedeterminēta diagrammas izpilde. Līdz ar to AD marķieru kustības likumi tiek būtiski vienkāršoti.

4.2 Vispārīgs UML 2.0 AD un izstrādātās VM apraksts

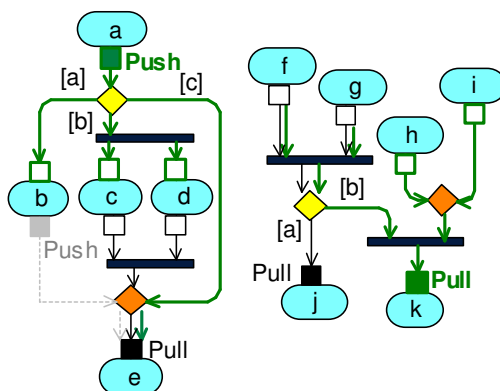
4.2.1 Aktivitāšu diagrammas standarta semantika

UML standartā aktivitāšu diagrammas izpildes semantika ir skaidrota ļoti sadalītā veidā, kur katrs AD elements pilda savu lomu [11,41,42]. Katra sadalīšanās, izvēles, apvienošanas un saplūšanas virsotne kopā ar šķautņu nosacījumiem nosaka, kuri no marķieriem tiek "piedāvāti" darbībām. Apkopojot atsevišķo kontroles elementu darbības aprakstu, marķieru "piedāvāšana" ir skaidrojama tā, ka vadības virsotnes marķierus padara darbībām "redzamus" un darbība tiek izpildīta tikai tad, "kad visām tā ieejas piespraudēm tiek piedāvāti marķieri, darbība tos saņem visus vienlaicīgi, neļaujot tos izmantot kādai citai darbībai" [11]. Praktiski tas nozīmē, ka darbības marķierus apstrādā ar vilkšanas paņēmieni un patiesi aktīvie diagrammas elementi ir "darbību dzinēji", kas cenšas savākt "redzamos" marķierus savās piespraudēs un izmantot tos definētajās darbībās. Skaidrs, ka ir iespējams uzbūvēt virtuālo mašīnu, kas precīzi atbilst UML AD standartam (ADVM), ar darbību dzinējiem kā aktīvajiem

elementiem, kas pārvieto marķierus, un vadības virsotnēm, kas strādā kā "marķieru redzamības slēdži". Tomēr tas būtu daudz sarežģītāk, jo šajā gadījumā marķieru redzamības uzstādīšana būtu acīmredzami nelokāla (sadalīta) darbība ar daudziem atkarīgiem iesaisītājiem elementiem.

4.2.2 Izstrādātās virtuālās mašīnas apraksts

Pētījumā ir izstrādāts atšķirīgs AD virtuālās mašīnas variants, kurā vadības virsotnes ("nestabilās vietas", kurās marķieri nevar atrasties) un šķautnes tiek apvienotas ceļos. Ceļi tātad savieno virsotnes, kurās marķieri var atrasties ("stabilās vietas") – parasti ieejas un izejas piespraudes (darbības, sākuma un beigu virsotnēm), kā arī aktivitāšu parametru virsotnes (aktivitātēm). Katram ceļam ir nosacījums, kas tiek veidots no apvienotiem šķautņu ierobežojumiem šajā ceļā. Augšminētie diagrammas veidošanas ierobežojumi nodrošina to, ka ceļi ir savstarpēji izslēdzoši. Pat tad, ja kāda piespraude ir sākums vairākiem ceļiem, katram konkrētam marķierim ir atļauts tikai viens ceļš.



Att. 12 Velkošo un grūdošo ceļu un dzinēju izveide dažādiem slēgumiem

"Stabilās vietas" apkalpo aktīvie elementi – marķieru dzinēji. Ir ieviesti divi dzinēju veidi – velkošie (*pull*) un grūdošie (*push*) marķieru dzinēji, kā arī grūdošie un velkošie ceļi. Grūdošie ceļi satur tikai izvēles, apvienošanas un saplūšanas virsotnes vai vispār nesatur vadības virsotnes. Grūdošos ceļus apkalpo grūdošie marķieru dzinēji, kas tiek piesaistīti ceļa sākumam – izejas piespraudei vai aktivitātes ieejas parametram. Izvēlētajā apakškopā marķieri no izejošajām piespraudēm var tikt pārvietoti pa grūdošajiem ceļiem pilnīgi neatkarīgi viens no otra līdz pat mērķim, ja vien ceļa nosacījumi to pieļauj. Līdz ar to grūdošajos ceļos marķieru kustība ir vienkārša un acīmredzama.

Velkošie ceļi ir tādi, kuros ir kaut viena saplūšanas (*join*) virsotne, kā arī izvēles un apvienošanas virsotnes. Velkošos ceļus apstrādā velkošie dzinēji, kas tiek piesaistīti ceļa mērķim – ieejas piespraudei (vai aktivitātes izejas parametram). Saskaņā ar AD semantiku, marķieri ar kopīgu mērķi pa velkošajiem ceļiem ir jāsaplūš – saplūšanas virsotni var šķērsot tikai saskaņota marķieru grupa.

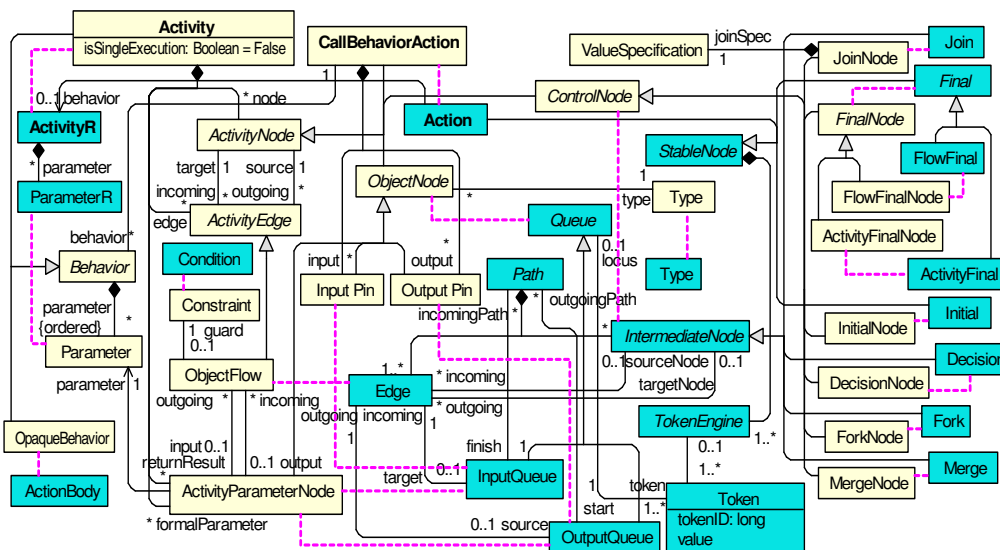
Darbības dzinējs ir daudz vienkāršāks par tā oriģinālās semantikas līdzinieku. Tā uzdevums ir savākt visus marķierus no visām ieejas piespraudēm vai vienu marķieru grupu no "velkošās piespraudes", tos "izmantot" un piedāvāt izejas piespraudēs.

Galvenā izstrādātās semantikas atšķirība no oriģinālās ir tā, ka marķieri vai to grupas tiek pārvietoti pa ceļiem uz ieejas piespraudi neatkarīgi, kamēr oriģinālajā semantikā darbības dzinējs savāc visus marķierus no izejas piespraudēm "visus vienlaicīgi". Tomēr tas nerada būtiskas procesa izpildes izmaiņas, jo darbību izvēlētajā apakškopā darbību dzinēju "cīņa par marķieriem" nav iespējama.

4.2.3 Metamodeļa paplašināšana un modeļu kartēšana

Lai formāli definētu ADVM, ir izveidots AD izpildes metamodelis, kurā ir visas nepieciešamās klases un operācijas. Katrai konkrētai aktivitāšu diagrammai izpildes modelis tiek radīts, veicot modeļa transformāciju.

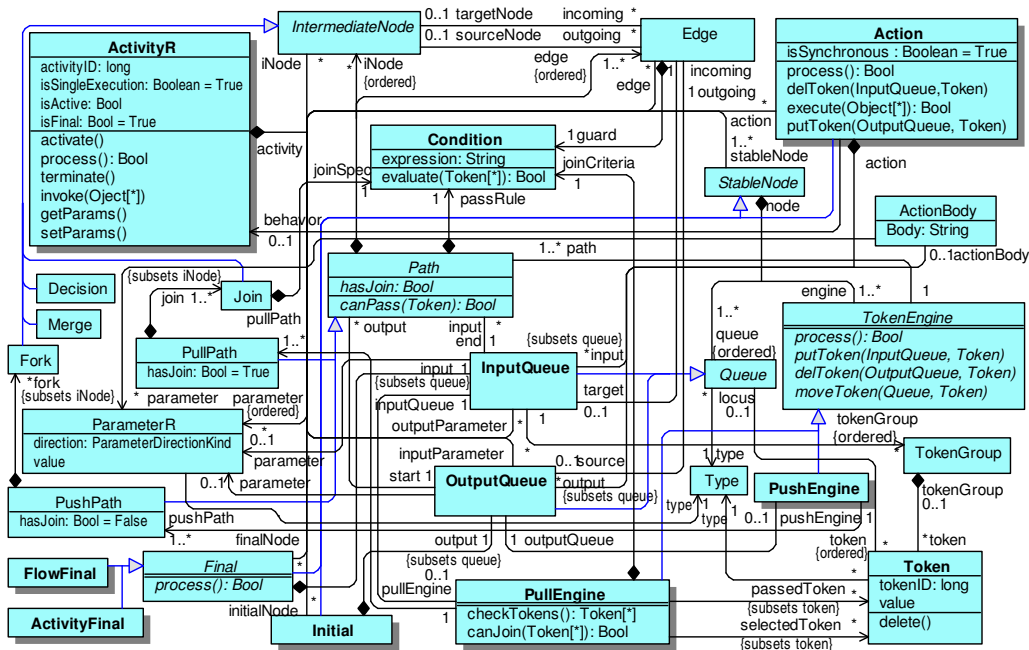
Att. 13 parāda vienā diagrammā gan AD oriģinālās metamodeļa klases, gan izpildes metamodeļa (virtuālās mašīnas) klases – oriģinālās klases ir gaišas, izpildes klases ir tumšas. Kur vien iespējams, atbilstošās klases abos metamodeļos ir savienotas ar īpašām divvirzienu asociācijām (2.3. nodaļā minētās kartēšanas asociācijas, raustītās līnijas).



Att. 13 UML aktivitāšu diagrammas apakškopa un tās saistība ar izpildes klasēm

Brīdī, kad aktivitāte tiek izsaukta, tās instancei ar visām sastāvdaļām tiek izveidotas atbilstošas izpildes klašu instance. Respektīvi, izpildes instance darbojas kā virtuālā mašīna, kas izpilda doto aktivitāti. Att. 13 attēlo "vispārīgo transformācijas shēmu", kur kartēšanas asociācijām ir formāla transformācijas jēga. Virzienā no definīcijas klases uz izpildes klasi asociācija norāda, ka transformācijas procesā katrai definīcijas klases instancei ir jāizveido atbilstoša izpildes klases instance. Pretējā virzienā asociācija norāda, no kuras definīcijas klases instance izpildes klases instance ir izveidota. Šī informācija ir nepieciešama, lai jaunizveidotajām instancēm iestatītu specifiskas īpašības, kuras iespējams iegūt tikai no avota (definīcijas) modeļa.

Att. 14 ir parādīta izstrādātā virtuālā mašīna. Šī diagramma attēlo citu skatu uz to pašu metamodeli, kas attēlots Att. 13, tikai šajā diagrammā nav avota (definīcijas) klašu, savukārt izpildes klases parādītas detalizētāk. Diagrammā ir parādītas pilnīgi visas izpildes klases, to asociācijas un operācijas, kas ir nepieciešamas virtuālās mašīnas darbināšanai.



Att. 14. Aktivitāšu diagrammas virtuālās mašīnas metamodelis

Katra izpildes elementa izveides apraksts (faktiski transformācija) un tā darbības pilnīgi precīzi ir parādīti publikācijā. Java pseidokods ir izmantots tur, kur ērtāka procedurāla pieeja, bet OCL nosacījumi – kur ērtāka deklarātīva pieeja. Kā tas ir aprakstīts 5. nodaļā, šādu pārveidi var definēt ar specializētu transformācijas valodu, piem., MOLA [80].

Tā kā izstrādātā virtuālā mašīna marķieru kustības līmenī darbojas citādāk kā oriģinālā, ir jāpierāda, ka oriģinālā un izstrādātā AD darbojas vienādi darbību izpildes secības līmenī. Darbā ir pierādīts, ka izstrādātajā virtuālajā mašīnā, lai arī atsevišķu marķieru kustības laiki var atšķirties, marķieru ceļi un marķieru izsaukto darbību sākums un beigas izvēlētajā elementu apakškopā pilnībā sakrīt. Pierādījums balstās uz sekojošo:

- Ar izvēlēto elementu kopu un diagrammas izveides ierobežojumiem gan izstrādātā, gan oriģinālā AD mašīna strādā determinēti.
- Lai arī izstrādātajā mašīnā atsevišķi marķieri var nonākt mērķa "stabilajā vietā" ātrāk kā oriģinālajā, tomēr pēdējais izstrādātās mašīnas marķieris vienmēr tiek saņemts tajā pašā brīdī, kad oriģinālā AD mašīna savāc visus nepieciešamos marķierus.

Tātad aktivitātes darbību izpildes līmenī abas mašīnas darbojas vienādi, bet izstrādātā mašīna ir daudz vienkāršāka. (Detalizēts pierādījums dots publikācijā, kur tas aizņem veselu nodaļu.)

Pseidokods un OCL nosacījumi uzskatāmi un precīzi parāda arī aktivitāšu diagrammas virtuālās mašīnas darbību (marķieru apstrādi). Šādā veidā uzbūvētā mašīna faktiski ir modeļu validācijas vai imitācijas sistēmas dizains. Darbā izmantotā pieeja var kalpot par pamatu arī darba plūsmas pārvaldības sistēmas izveidei.

5 UML un modeļu transformācijas biznesa procesu definēšanā

Modeļu bāzēta arhitektūra (*model driven architecture* – MDA), izmantojot modeļu bāzētu izstrādi (*model driven development* – MDD), ieņem arvien nozīmīgāku lomu arī biznesa vadības programmatūras izstrādē. Savukārt automatiska modeļu transformācija būtiski atvieglo sistēmu izstrādi ar MDD paņēmieni dažādos sistēmas izstrādes dzīves cikla posmos. Modeļu bāzētai biznesa vadības sistēmu izstrādei ir nepieciešama precīza biznesa modelēšanas valoda. Tā kā pasaulē nav "vienīgās labākās" biznesa modelēšanas valodas, modeļu bāzētā biznesa vadības programmatūras izstrādē bieži ir nepieciešama ātra modeļu transformācija no vienas valodas citā.

Pētījuma mērķis bija pierādīt, ka iespējams definēt precīzas un automatiski izpildāmas modeļu transformācijas, kas ļauj biznesa procesa modeli pārveidot no vienas modelēšanas valodas citā, nezaudējot modeļu semantiku.

Pētījumā ir apskatīti praksē nepieciešamie darba plūsmas modelēšanas aspekti, un uz to bāzes ir izveidots AD profils un tam atbilstoša BPMN apakškopa. AD profila un BPMN apakškopas izvēle ir balstīta uz iepriekšējos pētījumos veikto atziņu pamata (1. un 3. nodaļas), valodu līdzekļus papildinot ar B2B (*business-to-business*) sadarbības līdzekļiem. Abu valodu semantika ir analizēta, pievēršot uzmanību datu plūsmai sadalītos biznesa procesos, procesu veicēju aprakstīšanai un modeļu izpildes nosacījumiem.

Izstrādātais transformāciju paņēmieni ir ilustrēti, izmantojot divas izplatītas biznesa modelēšanas valodas – UML aktivitāšu diagrammu (AD) [11] un biznesa procesu modelēšanas notāciju (*Business Process Modeling Notation* – BPMN [15], abu valodu standartus šobrīd uztur OMG grupa). Transformācijas ir realizētas MOLA modeļu transformācijas valodā. Izstrādātā transformāciju pieeja nav ierobežota tikai ar aprakstītajām modelēšanas valodām. Līdzīgi iespējams veikt gan pretēju transformāciju, gan arī pavisam savādākas transformācijas, izmantojot citas modelēšanas valodas.

Kā jau tika minēts 2.3. nodaļā, šobrīd ar AD un BPMN valodu apvienošanu nodarbojas OMG domēna darba grupa (*Domain task force*) [70], kas valodu jēdzienus salāgo ar kartēšanas paņēmieni [55]. Tomēr, ņemot vērā valodu jēdzienu sarežģīto attiecību, pētījumā ir parādīts, ka jēdzienu salāgošana ar modeļu transformācijām ir efektīvāka un jēdzienu attiecību parāda precīzāk.

5.1 Darba plūsmas definēšanas valodas

Darba plūsmas definēšanai ir nepieciešama viegli lasāma grafiska valoda ar precīzu izpildes semantiku [64,65]. No vairākām pētījumā apskatītajām modelēšanas valodām pieejas ilustrācijai ir izvēlētas UML AD un BPMN, jo tās vispilnīgāk apmierina darba plūsmas definēšanai nepieciešamos nosacījumus. Tālāk ir analizēta šo valodu izpildes semantika, pievēršot uzmanību biznesa procesu specifikai un tam, kā valodu grafiskie līdzekļi to spēj attēlot.

Tā kā faktiski UML AD elementu semantika ir precizēta tā, ka labāk atbilst iegultajām sistēmām, biznesa procesu modelēšanai ir nepieciešama AD pielāgošana. Citiem vārdiem sakot, ir nepieciešams izveidot domēnam specifisku valodu (*domain specific language* – DSL) kā AD profilu, kur UML ieviestie stereotipi novērstu pamanītās AD darba plūsmas definēšanas nepilnības. Kā transformācijas mērķa valoda ir izvēlēta BPMN. Arī tā ir guvusi zināmu rīku atbalstu [63]. Tomēr arī šai valodai ir savas nepilnības, īpaši neformālā semantika un nepietiekams atbalsts datu struktūru

veidošanai. Tāpēc īsumā ir apskatītas arī BPMN izmantošanas iespējas, un ir izvēlēta sadalītu biznesa procesu modelēšanai nepieciešamā apakškopa.

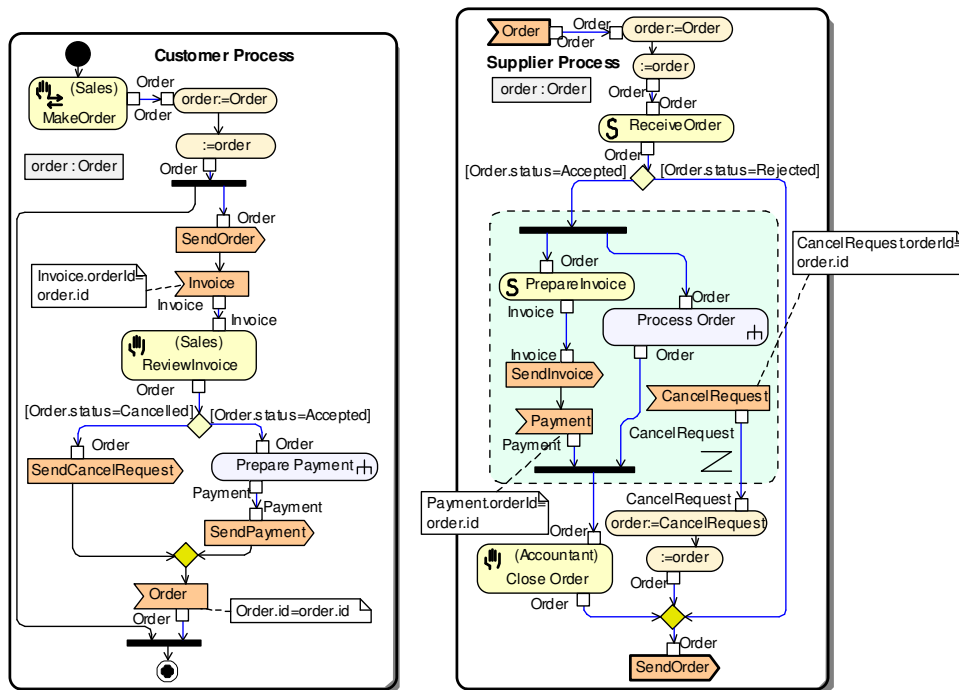
Lai precīzi noteiktu abu valodu izpildes semantiku sadalītos biznesa procesos, tās ir kartētas uz pagaidām vienīgo praktiski izpildāmo biznesa procesu modelēšanas valodu BPEL (*Business Process Execution Language*) [16]. Tajās vietās, kur standartā noteiktie valodas izteiksmes līdzekļi ir nepietiekami, tā ir paplašināta ar valodu implementējošo firmu *de-facto* rūpnieciski izmantotiem līdzekļiem.

5.2 UML aktivitāšu diagrammas pielāgošana darba plūsmas modelēšanai

Lai arī ir zināmi vairāki AD profili darba plūsmas definēšanai [50,51,52], neviens no tiem nenodrošina visas pētījumā izvirzītās darba plūsmas modelēšanai nepieciešamās prasības – sadalītu procesu sadarbību ar ziņojumiem, datu apstrādi, manuālu procesu veicēju aprakstīšanu un modeļa izpildes iespēju.

Tāpēc pētījumā ir veikta jauna AD profila izstrāde. No AD elementiem ir izvēlēti tie, kas nepieciešami darba plūsmas modelēšanai, saglabājot oriģinālo AD elementu nozīmi un ieviešot stereotipus tur, kur tiem nepieciešamas papildu īpašības vai arī elementu nozīmi ir nepieciešams precizēt sadalītas darba plūsmas vajadzībām.

Att. 15 parādītais biznesa process ar divām aktivitātēm ilustrē izstrādāto AD profilu. (Šī un citas diagrammas ir zīmētas GMF (*Generic Modeling Framework*) jeb EBM (*Exigen Business Modeler*) rīkā [87], kurā ir izveidoti papildus redaktori UML AD profilam un BPMN valodai). Diagrammā ir attēloti praktiski visi izvēlētie darba plūsmas definēšanas elementi. Lielākā daļa elementu ir izvēlēti jau no iepriekšējiem pētījumiem (1., 3., un 4. nodaļa), pievienojot tikai tos elementus, kas raksturīgi B2B sadarbībai.



Att. 15. Procesa piemērs UML aktivitāšu diagrammās

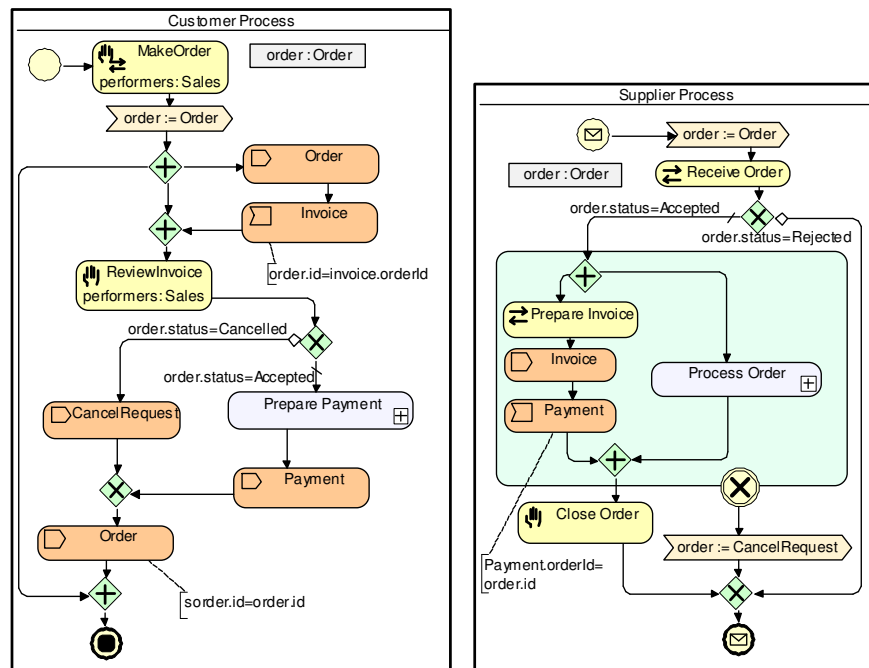
Jaunie elementi ir signāla nosūtīšanas (*SendSignal*) un notikuma saņemšanas (*AcceptEvent*) darbības (attiecīgi apzīmētas ar ieliektu un izliektu karodziņu), kā arī stereotipi dažādiem darbību tiem. Elements `order : Order` ir mainīgā `order`

definīcija ar tipu `Order`, kura redzamības apgabals ir aktivitāte `Supplier Process`. Mainīgo operāciju darbības ir parādītas kā standarta OCL operācijas.

Notikuma saņemšanas darbība ar nosacījumu izejošajai šķautnei ir īpaša konstrukcija, kas nepieciešama procesa instances noteikšanai un kas ir ekvivalenta BPEL korelācijas kopai.

Ar stereotipiem ir precizēta esošo AD elementu nozīme sadalīta biznesa procesa modelēšanas vajadzībām. Att. 16 attēlotais metamodelis parāda AD ieviestos stereotipus:

- *Galvenais process (MainProcess)* ir aktivitātes stereotips, kas norāda, ka aktivitāte ir atsevišķs darba plūsmas process, ko izpilda atsevišķs darba plūsmas dzinējs. Grafiski tas parādīts kā aktivitāte ar ēnu.
- *Veicējs (Performer)* ir sadaļas (*Partition*) stereotips, kas norāda manuāli veicamas darbības veicēju. Tā `represents` asociācija norāda uz klasi ar *Position* vai *OrgUnit* stereotipu. Veicējs ir parādīts kā darbības nodalījums (*compartment*).
- *WebServiss (WebService)* ir komponentes (*Component*) stereotips, kas tiek izmantots, lai norādītu web servisu atribūtus. *CallServiceTask* darbības, kas izsauc dotā web servisa operācijas, iegūst nepieciešamos parametrus no šī stereotipa.
- *Starpposma signāla nosūtīšanas darbība (IntermSSAction)* nozīmē vienkāršu signāla nosūtīšanu web servisam (parādīts kā izliekts karodziņš), kamēr *beigu signāla nosūtīšanas darbība (EndSSAction)* papildus signāla nosūtīšanai nozīmē arī aktivitātes beigas (izliekts karodziņš ar ēnu).
- *Katram (ForEach)* ir stereotips cilpas virsotnei (*LoopeNode*). Papildu `ForEach.collection` asociācija uz `ValuePin` ir ieviesta iteratoram (iteratora nepieciešamību biznesa modelēšanas valodās norāda arī [64]). Tā kā papildus asociāciju nevar ieviest ar stereotipu, šī ir vienīgā "spēcīgās paplašināšanas" vieta, lai maksimāli saglabātu metamodeļa savietojamību.



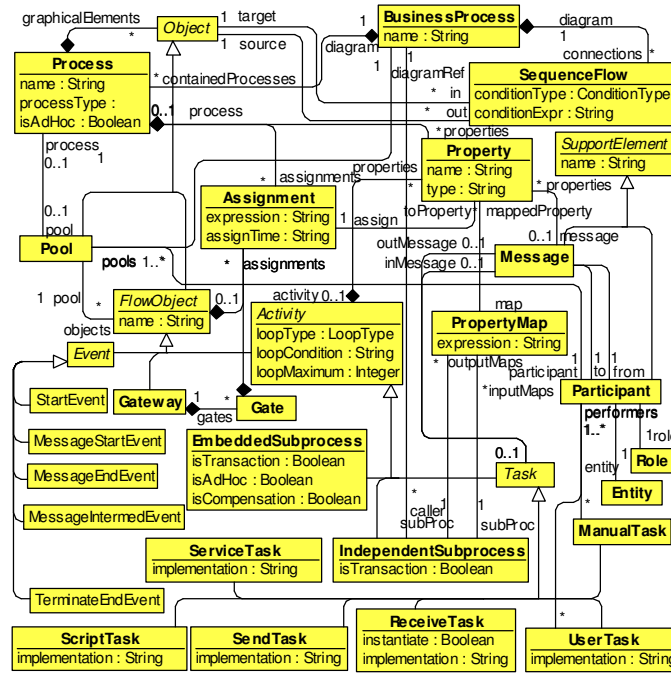
Att. 17. Procesa piemērs BPMN valodā

Ziņojumu nosūtīšanai un saņemšanai "nepārtraukumu" gadījumos tiek izmantoti nosūtīšanas un saņemšanas uzdevumi (noapaļoti taisnstūri ar karodziņiem iekšpusē).

Lai arī BPMN metamodelis ir pieņemams, tomēr vairākiem vajadzīgajiem elementiem nav grafiskās prezentācijas. Grafiskā prezentācija ir ieviesta sekojošiem elementiem:

- Uzdevumu tipiem tiek izmantoti stereotipi ar ikonām uzdevuma iekšpusē, uzdevuma nodalījumā ir minēti uzdevumu veicēji.
- Īpašības (*properties*) ir parādītas kā čertstūri formā `v rds:tips`.
- Lai padarītu redzamu datu apstrādi, piešķiršanas (*assignments*) ir parādītas ar lielām bultām un tekstuālu izteiksmi tajās.

Līdzīgi kā UML AD Att. 18 parāda BPMN fragmentu ar izvēlēto elementu apakškopu. Šo elementu instances parādās modeļa transformācijas rezultātā.

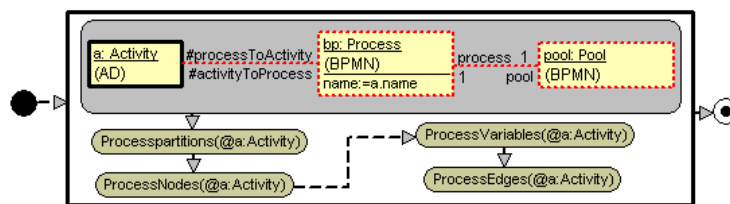


Att. 18. BPMN metamodela fragments (transformācijas mērķis)

5.4 AD transformācija uz BPMN

Lai arī vairāki darbi ir veltīti formālām biznesa modeļu transformācijām [60,65], netika atrasts darbs, kurā būtu aprakstīta AD transformācija uz BPMN, iekļaujot visus ar darba plūsmu definēšanu saistītos aspektus. Tāpēc ir izmantotas esošās AD kartēšanas shēmas uz BPMN [66,67,68,69,74,75], papildinot tās ar datu plūsmām, mainīgajiem, piešķiršanām un uzdevumu veicējiem.

Lai detalizētu jēdzienu kartējumu, ir parādīti arī daži fragmenti no modeļu transformācijām, kas izveido BPMN modeli no AD modeļa. Transformācijas ir rakstītas MOLA modeļu transformācijas valodā [80]. Att. 16 un Att. 18 parāda attiecīgi transformāciju avota un mērķa metamodelus, bet Att. 19 ilustrē vienu no transformācijām. Šajā transformācijā katra aktivitāte tiek transformēta BPMN procesā (*Process*) un pūlā (*Pool*), un tiek izsauktas transformāciju apakšprogrammas, kas apstrādā pārējos aktivitātes elementus.



Att. 19. Transformācijas piemērs no AD uz BPMN MOLA valodā

MOLA transformāciju programma pilnīgi precīzi apraksta atbilstību starp aktivitāšu diagrammas un BPMN elementiem. Kā redzams Att. 19, šādi atbilstībai ir nepieciešams izmantot vairākas klases un konteksta nosacījumus, kas ir pietiekami sarežģīti, lai tos nevarētu viennozīmīgi aprakstīt tikai ar kartējuma asociācijām. Modeļu transformācija ir labākais veids kā definēt sarežģītu atbilstību starp divām dažādām valodām, ņemot vērā visas ar valodu nozīmi saistītās īpatnības.

Transformācija ir realizēta GMF rīkā, izstrādājot jaunus AD un BPMN valodu redaktorus, un izmantojot MOLA valodas transformācijas vidi ar MySQL datu bāzi

[89]. Realizācija pierādīja paņēmienu efektivitāti pētnieciskiem nolūkiem – nelieli modeļi ar vairākiem desmitiem klašu tiek transformēti dažū sekunžu laikā uz standarta personālā datora. Uz šo pētījumu pamata šobrīd tiek izstrādāta jauna MOLA transformāciju valodas izpildes vide ar operatīvajā atmiņā glabājamu datu repozitoriju, kas ir izmantojams transformāciju izpildei rūpnieciskos mērogos.

6 Noslēgums

Šo pētījumu mērķis bija novērtēt un izmantot metamodelēšanas paņēmienus precīzu (izpildāmu) biznesa procesu definēšanā. Dažas autora izstrādātās biznesa procesu modelēšanas idejas jau ir ieviestas esošajos modelēšanas rīkos, bet citas var izmantot turpmākai modelēšanas rīku attīstībai. Turpmāk dots autora ideju vērtējums jaunāko biznesa procesa modelēšanas notikumu kontekstā:

- Pētījuma sākumā izstrādātais "notācijas neatkarīgais" biznesa procesu metamodelis parāda biznesa modelēšanas jēdzienus un to saistību un ir izmantots kā visu turpmāko pētījumu bāze. Izvirzītās idejas sasaucas ar OMG vēlāk izstrādātajiem Biznesa motivācijas modeļa [49] un Procesu definīcijas standarta [48] projektiem. Tomēr autora izstrādātajā biznesa procesu metamodelī procesa resursu, ieeju un izeju, kā arī uzdevuma veicēju apraksts ir detalizētāks. Uz nedaudz modificēta biznesa procesu metamodeļa pamata autors ir izveidojis UML AD un BPMN modelēšanas valodu redaktorus GMF rīkā [87].
- Autora izvirzītā ideja biznesa modeļu jēdzienu kartēšanā no viena domēna uz vairākām prezentācijām ir izmantota GMF rīkā, kurā viena domēna modeļi iespējams parādīt vairākās līdzīgās modelēšanas valodās. Ideja parādīt līdzīgas modelēšanas valodas kā "kanoniskās formas" skatus pēdējā laikā ir atspoguļojusies arī OMG grupas aktivitātēs, kas vienotā domēnā mēģina apvienot UML AD un BPMN modelēšanas valodas [55].
- Darba gaitā izstrādātie MOLA [80] valodas transformāciju piemēri biznesa modelēšanas vajadzībām ir devuši lielu ieguldījumu MOLA valodas un rīka validācijā, testēšanā un demonstrēšanā. MOLA rīkā realizētā ideja, transformējot biznesa modeļus no vienas modelēšanas valodas uz citu, ļauj vairākiem lietotājiem izmantot ērtāko modelēšanas valodu dažādās biznesa vadības sistēmu izstrādes fāzēs.
- LU MII šobrīd tiek izstrādāta jaunas paaudzes uz metamodeļiem un modeļu transformācijām balstīta rīku platforma TTF (*Transformation Tool Framework* [88]), kurā modeļu transformācijas tiek būvētas MOLA valodā. Kā viens no šīs jaunās paaudzes rīku platformas pielietojumiem ir dažādu specializētu modelēšanas rīku būve domēna specifiskām (DSL) valodām. Piemēram, jaunās paaudzes rīku platformā būs iespējams būvēt dažādu biznesa modelēšanas rīku redaktorus. Autora izstrādāto UML AD profilu un BPMN valodas apakškopu būs iespējams izmantot par pamatu redaktoru izstrādei šajā jaunās paaudzes modelēšanas rīku platformā. Savukārt autora izstrādāto aktivitāšu diagrammas virtuālo mašīnu un procesa mēru apstrādes satvaru būs iespējams izmantot modeļu imitācijas dzinēja izveidei. Tādējādi tiks nodrošināta visa biznesa procesu modelēšanai un imitācijai nepieciešamā funkcionalitāte.

7 Atsauces

7.1 Autora publikācijas recenzētos starptautisku konferenču materiālos

1. Vitolins Valdis, Audris Kalnins, *Modeling Business*. Modeling and Simulation of Business Systems, Kaunas University of Technology Press, edited by H. Pranevicius, E. Zavadskas, B. Rapp, Vilnius, May 13-14, 2003, pp. 215.-220.
2. Vitolins Valdis, *Business Process Measures*. Computer Science and Information Technologies, Databases and Information Systems Doctoral Consortium, University of Latvia Scientific Papers Vol. 673, edited by R. Freivalds, University of Latvia, 2004, pp. 186.-197.
3. Valdis Vitolins, Audris Kalnins, *Semantics of UML 2.0 Activity Diagram for Business Modeling by Means of Virtual Machine*, Proceedings Ninth IEEE International EDOC Enterprise Computing Conference, edited by Danielle C. Martin, IEEE, 2005, pp. 181.-192.
4. Audris Kalnins, Valdis Vitolins, *Use of UML and Model Transformations for Workflow Process Definitions*, Databases and Information Systems, BalticDB&IS'2006, edited by O. Vasilecas, J. Eder, A. Caplinskas, Vilnius, Technika, 2006, pp. 3.-15.

7.2 Citas autora publikācijas (tieši nesaistītas ar promocijas darba tēmu)

5. Vītoliņš Valdis, *Darba plūsmas pārvaldība*, e-pasaule (2001) Nr. 5, 6
6. Vītoliņš Valdis, *Microsoft dokumentu pārvaldības rīks SharePoint*, e-pasaule (2001), Nr. 7
7. Vītoliņš Valdis, *IT projektu pārvaldība*, e-pasaule (2001), Nr. 9
8. Vītoliņš Valdis, *Zināšanu pārvaldības sistēmas*, e-pasaule (2002), Nr. 9
9. Vītoliņš Valdis, *Organizācijas modelis biznesa procesu automatizācijā*, e-pasaule (2002), Nr. 10
10. Vītoliņš Valdis. *Biznesa modelēšanas rīki*, e-pasaule, (2003), Nr. 9.

7.3 Citi darbā izmatotie avoti

11. *Unified Modeling Language: Superstructure version 2.0*, OMG, 2005, <http://www.omg.org/cgi-bin/doc?formal/05-07-04>
12. *Unified Modeling Language (UML) Specification: Infrastructure, Version 2.0*, OMG, 2005, <http://www.omg.org/cgi-bin/doc?formal/05-07-05>
13. *OCL 2.0 Specification, Version 2.0*, OMG, 2005, <http://www.omg.org/cgi-bin/doc?ptc/05-06-06>
14. *MOF2 Versioning Final Adopted Specification*, OMG, 2006, <http://www.omg.org/cgi-bin/doc?formal/2006-01-01>
15. *Business Process Modeling Notation (BPMN), Final Adopted Specification*, OMG, 2006, <http://www.omg.org/cgi-bin/doc?dtc/2006-02-01>
16. *Business Process Execution Language for Web Services*, Version 1.0, IBM, 31 July 2002 <ftp://www6.software.ibm.com/software/developer/library/ws-bpel1.pdf>
17. Grady Booch, James Rumbaugh, *The Unified Modeling Language User Guide*, Addison-Wesley Longman, 1999
18. *Business Process Specification Schema v1.01*, Business Process Team, 11 May 2001
19. Osterwalder, S. Ben Lagha, Y. Pigneur, *An Ontology for Developing e-Business Models*, INFORGE, Encole des HEC, 2002
20. *ESPRIT project ADDE*, <http://www.fast.de/ADDE>
21. *The Zachman Institute for Framework Advancement (ZIFA)*, <http://www.zifa.com>
22. *System Architect, Tutorial*, 2001, Popkin Software, http://www.popkin.com/products/product_overview.htm
23. Andreas Dietzsch. *Adapting the UML to Business Modelling's Needs - Experiences in Situational Method Engineering*, UML 2002. LNCS 2460, pp 73-83
24. International Standard, *ISO/DIS 9001 Quality management systems — Requirements, International Organization for Standardization*, <http://www.iso.com>
25. Sandy Tyndale-Biscoe, *Business Modelling for Component Systems with UML*. Proceedings of the Sixth EDOC Conference (2002)
26. Osterwalder, *An e-Business Model Ontology for the Creation of New Management Software Tools and IS Requirement Engineering*, Ecole des Hec, Université de Lausanne, 2002

27. **Business Process Modeling Notation. Working Draft (0.9)** November 13, 2002, Business Process Management Initiative (BPMI), <http://www.bpmi.org>
28. Mark C. Paulk, et al., **The Capability Maturity Model : Guidelines for Improving the Software Process**, Addison Wesley Professional, 1995
29. Douglas T. Hicks, **Activity-Based Costing : Making It Work for Small and Mid-Sized Companies**, 2nd Edition, John Wiley & Sons, 2002
30. Robert S. Kaplan, David P. Norton, **The Strategy-Focused Organization: How Balanced Scorecard Companies Thrive in the New Business Environment**, Harvard Business School Press, 2000
31. Farok J. Contractor, **Valuation of Intangible Assets in Global Operations**, Quorum Books, 2001
32. **ARIS 6 – Collaborative Suite**, System White Paper, IDS Scheer, 2003, <http://www.ids-scheer.com/sixcms/media.php/1186/ARIS+6-2+SWP+en+2003-07.pdf>
33. **System Architect, Tutorial**, 2001, Popkin Software, http://www.popkin.com/products/product_overview.htm
34. **QPR Process Guide White Paper**, QPR ScoreCard White Paper, 2002, QPR Software Plc, http://www.qpr.com/protected/whitepapers/QPR_ScoreCard_WhitePaper.pdf
35. **Casewise Corporate Modeler Product Info**, Casewise, <http://www.casewise.com/products/corporate-modeler/corporate-modeler.php>
36. Stephen A. White, **Process Modeling Notations and Workflow Patterns**, March, 2004, <http://www.omg.org/bp-corner/pmn.htm>
37. R. Eshuis, R. Wieringa, **Comparing Petri Net and Activity Diagram Variants for Workflow Modelling - A Quest for Reactive Petri Nets**, Lecture Notes in Computer Science: Petri Net Technology for Communication-Based Systems: Advances in Petri Nets, Volume 2472. 2003, Heidelberg, Germany: Springer-Verlag. pp. 321 - 351.
38. Harald Störrle, Jan Hendrik Hausmann, **Towards a Formal Semantics of UML 2.0 Activities**, 2004, <http://www.pst.informatik.uni-muenchen.de/~stoerrle/V/AD-11-Limits.pdf>
39. Harald Störrle, **Semantics and Verification of Data Flow in UML 2.0 Activities**, 2004, <http://www.pst.informatik.uni-muenchen.de/~stoerrle/V/AD2b-DataFlow.pdf>
40. **Rational Software Architect**, <http://www-306.ibm.com/software/awdtools/architect/swarchitect/>
41. Conrad Bock, **UML 2 Activity and Action Models Part 4: Object Nodes**, in Journal of Object Technology, vol. 3, no. 1, pp. 27-41. http://www.jot.fm/issues/issue_2004_01/column3
42. Conrad Bock, **UML 2 Activity and Action Models Part 2: Actions**, in Journal of Object Technology, vol. 2, no. 5, pp. 41-56. http://www.jot.fm/issues/issue_2003_09/column4
43. Tom Baeyens, **The State of Workflow**, May 2004, <http://www.theserverside.com/articles/content/Workflow/article.html>
44. Derek Miers, Paul Harmon, **The 2005 BPM Suites Report, Version 1.0**, March 15, 2005, http://www.bptrends.com/reports_toc_01.cfm
45. Curtis Hall, Paul Harmon, **The 2005 Enterprise Architecture, Process Modeling & Simulation Tools Report**, Version 1.0, April 28, 2005, http://www.bptrends.com/reports_toc_02.cfm
46. Petia Wohed, et al., **Pattern-based Analysis of UML Activity Diagrams**, 2004, http://is.tm.tue.nl/research/patterns/download/uml2patterns_BETA_TR.pdf
47. WebSphere Business Integration Modeler, IBM, <http://www-306.ibm.com/software/integration/wbimodeler/>
48. **Business Process Definition Metamodel, Revised Submission**, Object Management Group, 2004, <http://www.omg.org/docs/bei/04-01-02.pdf>
49. **Business Motivation Model (BMM) Specification, Adopted Specification**, Object Management Group, 2004, <http://www.omg.org/docs/dtc/06-08-03.pdf>
50. Behzad Bordbar, Athanasios Staikopoulos, **On behavioural model transformation in Web services**, 5th International Workshop on Conceptual Modeling Approaches for e-Business eCOMO'2004, November 8-12, 2004
51. Jean Bezivin, et al., **Applying MDA Approach to B2B Applications: A Road Map**, Work-shop on Model Driven Development (WMDD 2004) at ECOOP 2004, Springer-Verlag, LNCS, vol. 3344, June 2004
52. Tracy Gardner, **UML Modelling of Automated Business Processes with a Mapping to BPEL4WS**, IBM, 2004, <http://www-128.ibm.com/developerworks/rational/library/4593.html>
53. Stephen A. White, **Using BPMN to Model a BPEL Process**, BPTrends, March 2005
54. A. Osterwalder, **An e-Business Model Ontology for the Creation of New Management Software Tools and IS Requirement Engineering**, Ecole des Hec, Université de Lausanne
55. Andrew Watson, **OMG's new modeling specifications**, ECMDA-FA 2005, Nuremberg, Germany, November 7-10, 2005, keynote speech

56. Armin Haller, Eyal Oren, Paavo Kotinurmi, *An Ontology for Internal and External Business Processes*, WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
57. *Web Services Business Process Execution Language Version 2.0, Primer Initial Draft*, 13th September, 2006
58. *BPELJ: BPEL for Java technology*, IBM Developerworks, 2004, <http://www-128.ibm.com/developerworks/library/specification/ws-bpelj/>
59. *Oracle BPEL Process Manager*, <http://www.oracle.com/technology/products/ias/bpel/index.html>
60. Draft *UML 1.4 Profile for Automated Business Processes with a Mapping to BPEL 1.0*, IBM, 2004, <http://www-128.ibm.com/developerworks/rational/library/4593.html>
61. *The Emerging Technologies Toolkit (ETTK)* <http://www.alphaworks.ibm.com/tech/ettk>
62. *MagicDraw UML 10.5* <http://www.magicdraw.com/>
63. *BPMI, Current Implementations of BPMN*, http://www.bpmn.org/BPMN_Supporters.htm#current
64. Mike Havey, *Essential Business Process Modeling*, O'Reilly, 2005, ISBN: 0-596-00843-0
65. Ken Beck, Joshy Joseph, Germán Goldszmidt, *Learn business process modeling basics for the analyst*, IBM Developerworks, 2005, <http://www-128.ibm.com/developerworks/library/ws-bpm4analyst/>
66. Alexandre Alves, *BPEL4WS 1.1 To WS-BPEL 2.0 - An SOA Migration Path*, 2005, http://webservices.sys-con.com/read/155617_1.htm
67. Wil van der Aalst, *Workflow Patterns*, <http://is.tm.tue.nl/research/patterns/>
68. Petia Wohed, et al., *Pattern-based Analysis of UML Activity Diagrams*, *BETA Working Paper Series, WP 129*, Eindhoven University of Technology, Eindhoven, 2004
69. Petia Wohed, et al., *Pattern-based Analysis of BPMN - an extensive evaluation of the Control-flow, the Data and the Resource Perspectives*, BPM Center Report BPM-05-26, BPMcenter.org, 2005
70. *Business Modeling & Integration Domain Task Force*, <http://bmi.omg.org/>
71. Bill Moore, et al., Using *BPEL Processes in WebSphere Business Integration Server Foundation Business Process Integration and Supply Chain Solutions*, IBM, 2004
72. Martin Keen et al., *BPEL4WS Business Processes with WebSphere Business Integration: Understanding, Modeling, Migrating*, IBM, 2004
73. Peter Swithinbank et al., *Patterns: Model-Driven Development Using IBM Rational Software Architect*, IBM, 2005
74. David Skogan, Roy Grønmo, Ida Solheim, *Web Service Composition in UML*, The 8th International IEEE Enterprise Distributed Object Computing Conference (EDOC), Monterey, Ca. Sept 2004
75. Roy Grønmo, Michael C. Jaeger, *Model-Driven Semantic Web Service Composition*, 12th Asia-Pacific Software Engineering Conference (APSEC), Taipei, Taiwan. December 2005
76. Chun Ouyang, et al., *Translating Standard Process Models to BPEL*, BPM Center Report BPM-05-27, BPM, 2005
77. *Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification*, OMG, 2005, <http://www.omg.org/docs/ptc/05-11-01.pdf>
78. J.Bārzdiņš, J.Tenteris, Ē.Viļums, *Biznesa modelēšanas valoda GRAPES-BM 4.0 un tās lietošana*, Rīga, 1998
79. Celms E., A. Kalnins, L. Lace. *Diagram definition facilities based on metamodel mappings*. OOPSLA'2003 (Workshop on Domain-Specific Modeling), Anaheim, California, USA, October 2003, pp. 23-32.
80. *MOLA Modeling language*, Institute of Mathematics and Computer Science University of Latvia, <http://mola.mii.lu.lv/>
81. A. Kalnins, J. Barzdins, E.Celms. *Efficiency Problems in MOLA Implementation*. 19th International Conference, OOPSLA'2004, Vancouver, Canada, October 2004
82. Kalnins, J. Barzdins, E. Celms. *Model Transformation Language MOLA*. Proceed-ings of MDFAFA 2004 (Model-Driven Architecture: Foundations and Applications 2004), Linköping, Sweden, June 10-11, 2004. pp.14-28.
83. *ProGuide*, <http://www.proformacorp.com/>
84. *xBML*, <http://www.xbmlinnovations.com/>
85. Satish Thatte, *XLANG Web Services for Business Process Design*, Microsoft Corporation, 2001
86. A. Ledecz et al., *The Generic Modeling Environment*, Workshop on Intelligent Signal Processing, Budapest, Hungary, May 17, 2001

87. L.Lace, E.Celms, A.Kalnins. *Diagram definition facilities in a generic modeling tool*, 2003,- Proceedings of International Conference on Modelling and Simulation of Business Systems, Vilnius, 2003, pp. 220-224.
88. A. Kalnins, J. Barzdins, *MDA Support by Transformation Based Tool*, Proceedings of First International Workshop MoRSe 2006, Warsaw, Poland, October 2006, pp. 21-24.
89. A. Kalnins, E. Celms, A. Sostaks, *Simple and Efficient Implementation of Pattern Matching in MOLA Tool*, Proceedings of the 7th International Baltic Conference on Databases and Information Systems (Baltic DB&IS'2006) , Vilnius, Lithuania, July 3-6, 2006, pp. 159-167.

1 Pielikums

1.1 Autora referāti par darba rezultātiem starptautiskās zinātniskās konferencēs vai semināros

1. Vitolins Valdis, Audris Kalnins, Modeling Business, Modeling and simulation of business systems, May 13-14, 2003, Vilnius, Lithuania.
2. Vitolins Valdis, Business Process Measures, 6th International Baltic Conference on Databases and Information Systems 2004, June 6-9, 2004, Riga, Latvia.
3. Valdis Vitolins, Audris Kalnins, Semantics of UML 2.0 Activity Diagram for Business Modeling by Means of Virtual Machine, Proceedings Ninth IEEE International EDOC Enterprise Computing Conference, 19-23 September 2005, Enschede, The Netherlands.
4. Audris Kalnins, Valdis Vitolins, Use of UML and Model Transformations for Workflow Process Definitions, Databases and Information Systems, BalticDB&IS'2006, 7th International Baltic Conference on Databases and Information Systems, July, 3-6 2006, Vilnius, Lithuania

1.2 Promocijas darbā iekļautās publikācijas un promocijas darba autora personiskais ieguldījums

Autori	Publikācija	Ieguldījums %	Promocijas darba autora ieguldījuma apraksts
Valdis Vītoliņš Audris Kalniņš	Modeling Business	90	<ul style="list-style-type: none">• Esošo modeļu apkopošana un analīze• Vienotā metamodeļa izstrāde• Jēdzienu kartēšanas idejas izstrāde
Valdis Vītoliņš	Business Process Measures	100	<ul style="list-style-type: none">• Esošo procesu mērīšanas metodiku analīze• Uz metamodeli balstītas mēru metodikas izstrāde• Mēru metamodeļa un metametamodeļa izstrāde
Valdis Vītoliņš Audris Kalniņš	Semantics of UML 2.0 Activity Diagram for Business Modeling by Means of Virtual Machine	90	<ul style="list-style-type: none">• UML aktivitāšu diagrammas darbības analīze un virtuālās mašīnas izveide• Vienkāršotās virtuālās mašīnas izveide, marķieru dzinēju darbības izstrāde
Valdis Vītoliņš Audris Kalniņš	Use of UML and Model Transformations for Workflow Process Definitions	90	<ul style="list-style-type: none">• UML aktivitāšu diagrammas analīze, BPMN valodas metamodeļa izstrāde, metamodeļu kartējuma izstrāde• UML AD un BPMN valodu redaktoru izstrāde• Transformāciju izstrāde no AD uz BPMN