

PRIMENA SysML NA POJEDNOSTAVLJEN MODEL OSMATRAČKOG RADARA

Miloš D. Jevtić^a, Siniša R. Marinković^b, Ivica B. Marjanović^c

^a Univerzitet u Beogradu, Elektrotehnički fakultet, Beograd, Republika Srbija,
e-mail: milos.jevtic@pupin.rs,

ORCID iD: <http://orcid.org/0000-0002-2358-0074>,

^b Univerzitet u Beogradu, Institut „Mihajlo Pupin”, Beograd, Republika Srbija,
e-mail: sinisa.marinkovic@pupin.rs,

ORCID iD: <http://orcid.org/0000-0002-3952-5432>,

^c Ministarstvo odbrane Republike Srbije, Vojna kontrola kvaliteta,
Beograd, Republika Srbija,

e-mail: ivicabeograd@gmail.com,

ORCID iD: <http://orcid.org/0000-0001-6945-7533>

DOI: 10.5937/vojtehg65-8697

OBLAST: informatika

VRSTA ČLANKA: stručni članak

JEZIK ČLANKA: srpski

Sažetak:

Systems Modeling Language (SysML) jeste profil Unified Modeling Language (UML) namenjen za upotrebu u sistemskom inženjerstvu. Ovaj tekst inspirisan je manjkom literature o SysML-u na srpskom jeziku, a ima za cilj upoznavanje inženjerske i akademske zajednice sa ovim interesantnim jezikom za modelovanje, pri čemu su posebno naglašene razlike u odnosu na UML. Opis je dat kroz konkretan originalan primer – pojednostavljen model osmatračkog radara. Ukratko su analizirani praktični aspekti SysML-a – koliko je prihvaćen, kakva je perspektiva njegove dalje primene, itd. Može se zaključiti da je SysML koristan, prihvaćen i perspektivan jezik za modelovanje.

Ključne reči: sistemsko inženjerstvo zasnovano na modelima, osmatrački radar, SysML, UML.

Uvod

Sistemsko inženjerstvo zasnovano na modelima (eng. Model Based Systems Engineering) savremen je pristup u sistemskom inženjerstvu koji

ACKNOWLEDGEMENT: Rad je finansijski potpomoglo Ministarstvo prosvete, nauke i tehnološkog razvoja Republike Srbije (Ugovor TR32051).

bi trebalo da zameni tradicionalan pristup zasnovan na dokumentima (Friedenthal, et al., 2008). Jedan od ključnih elemenata koji omogućuju primenu sistemskog inženjerstva zasnovanog na modelima je standardizovan i robustan jezik za modelovanje (Friedenthal, et al., 2008). Systems Modeling Language (SysML) jeste jezik za modelovanje opšte namene koji podržava specificiranje, analizu, projektovanje i verifikaciju sistema, a stvoren je kao odgovor na prethodno pomenute potrebe. SysML je proširenje jezika za modelovanje Unified Modeling Language (UML) (Friedenthal, et al., 2008), (Weilkiens, 2007).

Cilj ovog članka, za čije razumevanje je potrebno dobro poznavanje UML-a, jeste da se ukratko predstavi SysML i da se ispita njegova upotrebljivost u sadašnjoj i budućoj praksi. U sledećem odeljku navedene su osnovne informacije o sistemskom inženjerstvu (zasnovanom na modelima) i nastanku SysML-a. U trećem odeljku nalazi se kratak prikaz SysML-a sa težištem na razlikama u odnosu na UML, na primeru pojednostavljenog modela osmatračkog radara. Pomenuti primer nije preuzet iz postojeće literature već je namenski smišljen za potrebe ovog članka. U trećem odeljku predstavljen je kratak osvrt na praktične aspekte korišćenja SysML-a, na osnovu tuđih, ali i ličnih iskustava autora, dok je zaključak naveden u četvrtom odeljku.

Sistemsko inženjerstvo

Sistemsko inženjerstvo (eng. systems engineering) jeste „interdisciplinarni pristup i sredstvo koje omogućuje realizaciju uspešnih sistema” (INCOSE, 2004). Ovu definiciju preporučuje Međunarodni savet za sistemsko inženjerstvo (eng. International Council on Systems Engineering (INCOSE)). Slično tome, Friedenthal i ostali definišu sistemsko inženjerstvo kao „multidisciplinarni pristup kojim se skup potreba zainteresovanih strana transformiše u balansirano sistemsko rešenje koje zadovoljava date potrebe” (Friedenthal, et al., 2008).

Sistemsko inženjerstvo je skoncentrisano na definisanje i dokumentovanje zahteva, projektovanje sistema i verifikaciju sistema u smislu saobraznosti sa zahtevima (Weilkiens, 2007). Ono uzima u obzir celokupni životni ciklus sistema (uključujući faze koje nastupaju nakon isporuke sistema, kao što su obuka, podrška i eventualno povlačenje sistema iz upotrebe) (Weilkiens, 2007). Projektovanje i implementacija pojedinačnih komponenti nije predmet sistemskog inženjerstva već drugih inženjerskih disciplina (softversko inženjerstvo, elektrotehnika, mašinstvo itd.). Sistemsko inženjerstvo, međutim, mora definisati zahteve koje date komponente moraju zadovoljavati (bilo da se razvijaju ili kupuju kao gotovi proizvodi), način na koji se komponente integrišu u sistem, te procedure za

verifikaciju ispunjenosti sistemskih zahteva. Drugim rečima, sistemsko inženjerstvo je metadisciplina (Weilkiens, 2007), koja funkcioniše na višem nivou apstrakcije u odnosu na druge inženjerske discipline.

Sistemsko inženjerstvo zasnovano na modelima

Tradicionalan pristup u sistemskom inženjerstvu zasnovan je na dokumentima. Ovaj pristup može se okarakterisati stvaranjem specifikacija i projektnih dokumenata, koji se zatim razmenjuju među zainteresovanim stranama (eng. stakeholders). Sistemski zahtevi i projektne informacije su u ovim dokumentima izraženi u obliku teksta i crteža. Kod sistemskog inženjerstva zasnovanog na dokumentima, težište je na upravljanju dokumentacijom i osiguravanju da su dokumenti validni, kompletni i konzistentni, te da je razvijeni sistem saobrazan dokumentaciji (Friedenthal, et al., 2008). Iako rigorozan, pristup zasnovan na dokumentima ima fundamentalna ograničenja. Artefakti razvojnog procesa (zahtevi, projekat, rezultati analize, informacije vezane za testiranje, itd.) rasejani su na nekoliko dokumenata, što otežava procenu kompletnosti i konzistentnosti dokumentacije, kao i shvatanje odnosa među pomenutim artefaktima, što nadalje stvara poteškoće u razumevanju pojedinih aspekata sistema i postizanju sledljivosti (eng. traceability). Takođe, otežani su održavanje i ponovna upotreba sistemskih zahteva i projektnih informacija (Friedenthal, et al., 2008).

Termin inženjerstvo zasnovano na modelima (eng. model-based engineering) odnosi se na pristup u kojem centralnu ulogu u definisanju specifikacija, projektovanju, integraciji, validaciji i operativnoj upotrebi sistema imaju modeli (Estefan, 2008). Pristup zasnovan na modelima odavno je postao standardna praksa u mašinstvu i elektrotehnici, a postaje sve šire prihvaćen i u softverskom inženjerstvu, naročito nakon pojave UML-a (Friedenthal, et al., 2008). Očekuje se da će pristup zasnovan na modelima postati standardna praksa i u sistemskom inženjerstvu, na sličan način kao u pomenutim disciplinama (Friedenthal, et al., 2008).

Prema (Friedenthal, et al., 2008) i njihovim referencama, sistemsko inženjerstvo zasnovano na modelima (SIZM) jeste „formalizovana primena modelovanja kao podrške aktivnostima definisanja sistemskih zahteva, projektovanja, analize, verifikacije i validacije, počevši od faze idejnog projekta i nastavljajući se tokom razvoja i kasnijih faza životnog ciklusa”. Potencijalne koristi od primene SIZM su poboljšanja u: komunikaciji među zainteresovanim stranama, preciznosti izražavanja specifikacija i projekta, integraciji projekta i ponovnoj upotrebi artefakata razvojnog procesa.

Pomenuta poboljšanja trebalo bi da poprave kvalitet i produktivnost, te da smanje rizike (Friedenthal, et al., 2008). Kod SIZM težište je na upravljanju koherentnim modelom sistema i na korišćenju tog modela za specificiranje i projektovanje sistema (Friedenthal, et al., 2008). Model sistema evoluirao i inkrementalno raste tokom vremena (Friedenthal, et al., 2008), (Estefan, 2008).

O nastanku SysML-a

Inženjerstvo zasnovano na modelima podrazumeva postojanje jezika za modelovanje. Uprkos brojnim inicijativama za standardizaciju procesa u sistemskom inženjerstvu, do 2001. godine nije došlo do pojave jedinstvenog jezika za modelovanje (Weilkiens, 2007). Ovakva situacija dovela je do znatnih problema u interdisciplinarnim projektima (Weilkiens, 2007). U međuvremenu, UML je postao dominantan jezik za modelovanje u softverskom inženjerstvu (Weilkiens, 2007), a 1997. godine prihvaćen je kao standard od međunarodnog konzorcijuma za standardizaciju Object Management Group (OMG) (Tartalja, 2011).

Početakom 2001. godine INCOSE je doneo stratešku odluku da se UML prilagodi za primenu u sistemskom inženjerstvu. Zajedničkim naporima INCOSE i OMG postavljeni su zahtevi za novi jezik za modelovanje, koji su objavljeni u dokumentu „UML for Systems Engineering Request for Proposal (UML for SE RFP)”, 2003. godine (OMG, 2012). SysML je jezik za modelovanje koji je nastao kao odgovor na pomenute zahteve. Verzija 1.0 SysML-a (zasnovana na verziji 2.1.1. UML-a) usvojena je kao zvanični OMG standard 2007. godine (Weilkiens, 2007). Nakon toga SysML je doživeo nekoliko revizija, a verzija 1.3 (zasnovana na verziji 2.4.1. UML-a), koja je opisana u ovom članku, usvojena je kao standard 2012. godine (OMG, 2012).

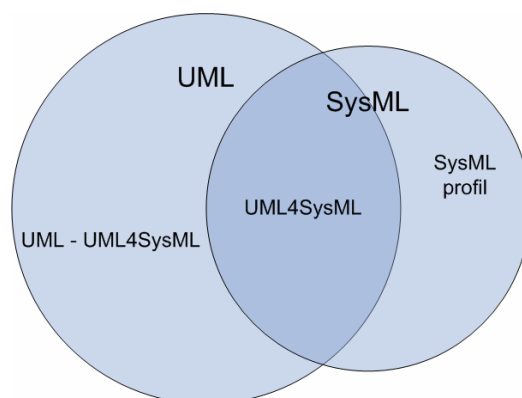
Postoji više razloga zbog kojih je SysML zasnovan na UML-u. Naime, UML je popularan jezik koji je u širokoj upotrebi, može se prilagoditi specifičnim potrebama (zahvaljujući svojim mehanizmima za proširivanje), a postoji i veliki broj raspoloživih alata za modelovanje, kao i organizacija koje pružaju obuku ili savetovanje (Weilkiens, 2007). Takođe, sistemski inženjeri koji koriste SysML i softverski inženjeri koji koriste UML moći će da sarađuju na modelima softverski intenzivnih sistema. Zahvaljujući tome, poboljšaće se komunikacija među zainteresovanim stranama koje učestvuju u razvojnem procesu i promovisaće se interoperabilnost među alatima za modelovanje (OMG, 2012).

Može se postaviti pitanje da li je stvaranje novog jezika za modelovanje bilo zaista neophodno i opravdano, ili zašto UML u postojećoj

formi nije mogao biti upotrebljen u sistemskom inženjerstvu? Iako je UML veoma ekspresivan jezik, nedostaju mu neki elementi koji su bitni za sistemsko inženjerstvo, kao što je modelovanje zahteva (Weilkiens, 2007). Pored toga, UML je usmeren na modelovanje softverskih sistema i jako je povezan sa objektnim modelom razvoja (Weilkiens, 2007). Sa druge strane, od jezika za modelovanje u sistemskom inženjerstvu očekuje se da podrži interdisciplinarno modelovanje (Weilkiens, 2007) i da bude agnostičan u odnosu na primenjenu metodologiju (OMG, 2012).

Kratak prikaz SysML-a

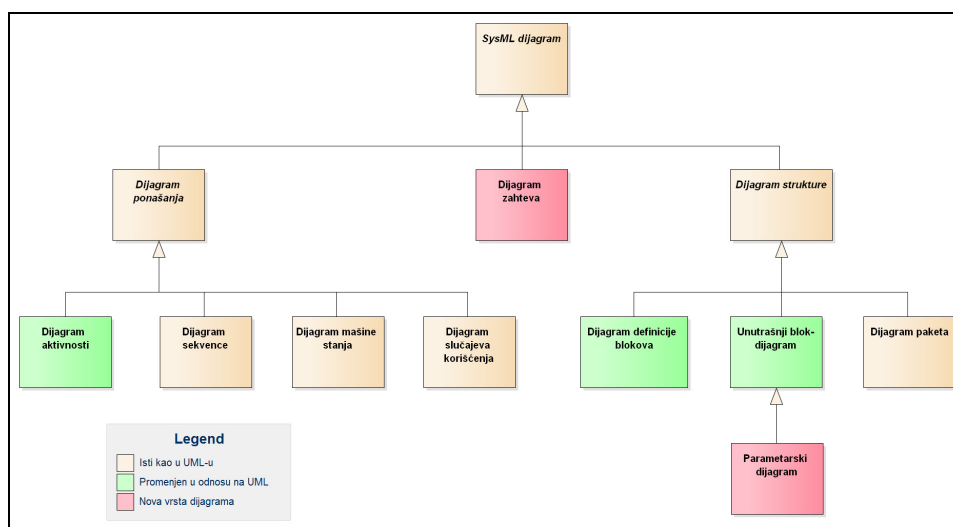
SysML ponovo upotrebljava (eng. reuse) podskup UML-a i dodaje proširenja koja su neophodna da bi se ispunili zahtevi iz „UML for SE RFP” (OMG, 2012). Veza između UML-a i SysML-a prikazana je Venovim dijagramom na slici 1, po uzoru na (OMG, 2012). Krugovi označeni sa „UML” i „SysML” označavaju skupove jezičkih konstrukcija koje čine UML i SysML, respektivno. Presek ova dva skupa, označen sa „UML4SysML” predstavlja deo UML-a koji je ponovo upotrebljen u SysML-u. Region označen sa „SysML profil” predstavlja nove jezičke konstrukcije definisane u SysML-u, a koje ili ne postoje u UML-u ili menjaju odgovarajuće UML konstrukcije. Region označen sa „UML – UML4SysML” odnosi se na deo UML-a koji nije potreban za implementaciju SysML-a.



Slika 1 – Veza između UML-a i SysML-a
 Рус. 1 – Связь между UML и SysML
 Figure 1 – Relation between UML and SysML

Klasifikacija SysML dijagrama prikazana je na slici 2, po uzoru na (OMG, 2012). Dijagram sekvence (eng. sequence diagram), dijagram mašine stanja (eng. state machine diagram), dijagram slučajeva korišćenja

(eng. use case diagram) i dijagram paketa (eng. package diagram) jesu ponovo upotrebljeni iz UML-a, bez izmena (OMG, 2012). Dijagrami koji su nastali modifikacijom odgovarajućih UML dijagrama su: dijagram aktivnosti (eng. activity diagram), dijagram definicije blokova (eng. block definition diagram) koji je zapravo modifikacija UML dijagrama klasa (eng. class diagram) i unutrašnji blok-dijagram (eng. internal block diagram) koji je, u stvari, modifikacija UML dijagrama složene strukture (eng. composite structure diagram) (OMG, 2012).



Slika 2 – Klasifikacija SysML dijagrama
 Рис. 2 – Диаграмма классификации SysML
 Figure 2 – SysML diagram classification

SysML uvodi dve nove vrste dijagrama. Dijagram zahteva (eng. requirement diagram) namenjen je modelovanju zahteva predstavljenih u tekstualnom obliku, uključujući veze sa drugim elementima modela koji date zahteve zadovoljavaju ili definišu način njihove verifikacije (OMG, 2012). Parametarski dijagram (eng. parametric diagram) opisuje ograničenja (eng. constraint) koja postoje među svojstvima (eng. property) blokova (OMG, 2012) (blok je modifikovana UML klasa, o čemu će više reči biti kasnije). Parametarski dijagram uveden je radi integracije modela strukture i ponašanja sa analitičkim modelima, kao što su model performansi ili model pouzdanosti (OMG, 2012).

Pojedini UML dijagrami se ne koriste u SysML-u. Dijagram raspoređivanja (eng. deployment diagram) isključen je iz SysML-a, jer se raspoređivanje softvera na hardverske resurse može predstaviti pomoću unutrašnjeg blok-dijagrama (OMG, 2012). Pored toga, smatralo se da su

SysML dijagrami ponašanja, prikazani na slici 2, sasvim adekvatni i dovoljni za predstavljanje ponašanja, te su dijagram komunikacije (eng. communication diagram) i dijagram pregleda interakcije (eng. interaction overview diagram) isključeni iz SysML specifikacije (OMG, 2012). Vremenski dijagram (eng. timing diagram) nije našao mesto u SysML-u, jer je u pitanju relativno nova vrsta UML dijagrama koja još nije do kraja razvijena, a postoji i pitanje da li je ova vrsta dijagrama prikladna za potrebe sistemskog inženjerstva (OMG, 2012). Ipak, rečenica iz (OMG, 2012) koja glasi: „Iako UML 2 vremenski dijagram nije uključen u ovu verziju SysML-a, on može da dopuni SysML dijagrame ponašanja ...” ukazuje na potencijalno uključivanje vremenskog dijagrama u buduće specifikacije SysML-a. Dijagram profila (eng. profile diagram) nije uključen u SysML, jer se definicije profila mogu predstaviti i na dijagramu paketa (OMG, 2012). Konačno, dijagram objekata (eng. object diagram) i dijagram komponenta (eng. component diagram) ne postoje u SysML-u.

Svaki SysML dijagram mora imati okvir (eng. frame), za razliku od UML-a gde su okviri dijagrama opcioni (OMG, 2012). Okvir je pravougaonik čiji je veći deo površine predviđen za prikaz sadržaja dijagrama, dok je manji deo površine u gornjem levom uglu rezervisan za zaglavlje (eng. heading). Zaglavlje je niska sa sintaksom *vrstaDijagrama [tipElementaModela] imeElementaModela [imeDijagrama]*.

Vrsta dijagrama i ime elementa modela su obavezni u zaglavlju. Ime elementa modela ukazuje na element modela koji je podrazumevani prostor imena (eng. namespace) za elemente modela prikazane unutar okvira. Elementi modela koji ne pripadaju podrazumevanom prostoru imena moraju biti prikazani sa kvalifikovanim imenima. Vrsta dijagrama je jedna od skraćenica iz sledeće liste: *act* (dijagram aktivnosti), *bdd* (dijagram definicije blokova), *ibd* (unutrašnji blok-dijagram), *pkg* (dijagram paketa), *par* (parametarski dijagram), *req* (dijagram zahteva), *sd* (dijagram sekvence), *stm* (dijagram mašine stanja), *uc* (dijagram slučajeva korišćenja).

Primer okvira dijagrama može se videti na slici 2 (koja je zapravo SysML dijagram definicije blokova). U nastavku ovog odeljka biće detaljnije opisani elementi SysML profila. Dijagrami i jezičke konstrukcije koji čine SysML biće ilustrovani na konkretnom primeru veoma pojednostavljenog modela osmatračkog radara.

Dijagram definicije blokova i unutrašnji blok-dijagram

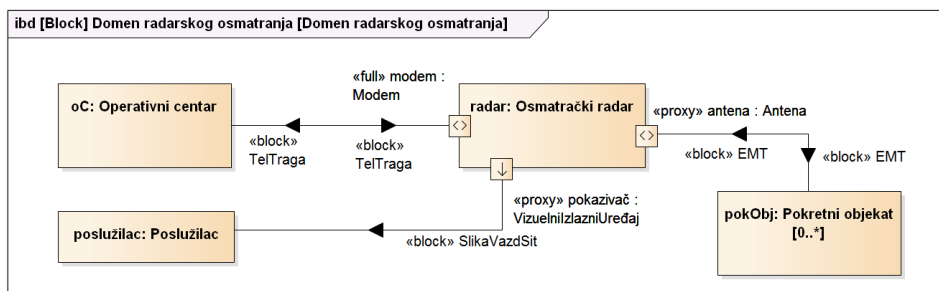
U SysML-u blok (eng. block) podrazumeva modularnu jedinicu opisa sistema. On definiše skup karakteristika (eng. features) koje opisuju sistem ili neki drugi značajan element. Formalno, blok je UML klasa sa stereotipom «*block*» (OMG, 2012).

Dijagram definicije blokova zasnovan je na UML dijagramu klasa uz određena proširenja i ograničenja. Namenjen je za definisanje karakteristika blokova i relacija među blokovima (OMG, 2012). Karakteristike blokova uključuju svojstva i operacije. Svojstva su strukturne karakteristike, npr. delovi i portovi poznati iz UML-a, ali i nove vrste svojstava o kojima će biti reči kasnije. Operacije su karakteristike bloka koje se odnose na ponašanje. Među blokovima mogu postojati svi tipovi relacija koji mogu postojati između UML klasa: zavisnost, asocijacija, agregacija, kompozicija, generalizacija i ugnježđivanje. Dijagrami definicije blokova uglavnom se koriste za prikaz hijerarhijske dekompozicije nekog elementa na podelemente („je deo” hijerarhija) ili za prikaz stabla klasifikacije elemenata („je vrsta” hijerarhija). Zbog toga su na ovim dijagramima najzastupljenije relacije kompozicije i generalizacije.

Unutrašnji blok-dijagram zasnovan je na UML dijagramu složene strukture uz određena proširenja i ograničenja. Na ovom dijagramu prikazuje se unutrašnja struktura bloka pomoću svojstava koja su povezana konektorima (OMG, 2012). Dok se dijagram složene strukture povezuje sa naprednim korišćenjem UML-a, unutrašnji blok-dijagram je esencijalni element strukturnog modelovanja u SysML-u.

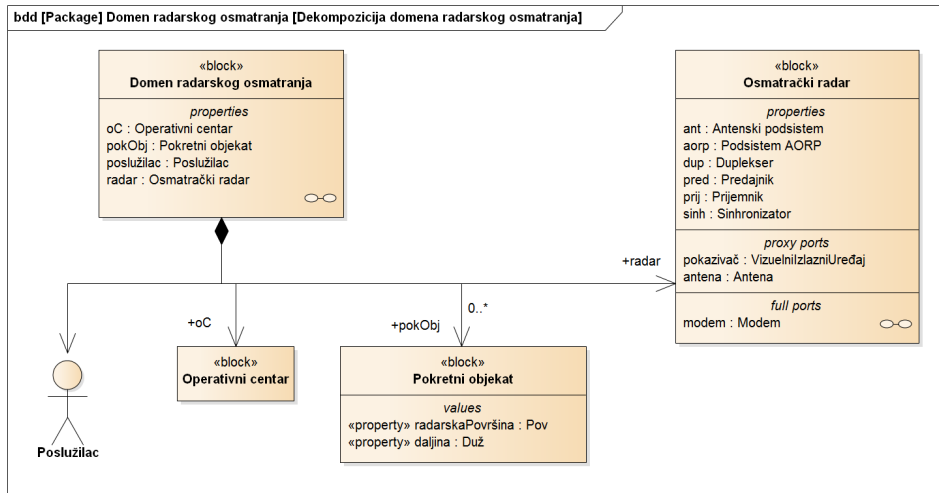
Sledi ilustracija specifičnosti dijagrama definicije blokova i unutrašnjeg blok-dijagrama na ranije pomenutom primeru osmatračkog radara. Na slikama 3, 4 i 5, dati su SysML dijagrami koji opisuju domen problema, tj. domen radarskog osmatranja. Unutrašnji blok-dijagram na slici 3 prikazuje delove domena radarskog osmatranja (radar, pokretne objekte, poslužioca i operativni centar) i način na koji su oni povezani. Radar emituje elektromagnetne talase (EMT) putem antene. EMT se odbijaju od pokretnih objekata nazad prema radaru koji ih prima putem antene. Obradom primljenih signala formira se slika vazdušne situacije koja se prezentira poslužiocu putem pokazivača. Primenom tehnika automatske detekcije i praćenja (ADP) generišu se tragovi koji se putem modema šalju operativnom centru (OC). Od OC se putem modema dobijaju tragovi generisani na drugim radarima ili dobijeni tehnikama fuzije podataka u samom OC.

Na osnovu zaglavlja dijagrama na slici 3 vidi se da se dati unutrašnji blok-dijagram odnosi na blok *Domen radarskog osmatranja*. Ovaj blok i njegova dekompozicija na delove definisani su pomoću dijagrama definicije blokova na slici 4. Ovde su iskorišćeni označeni odeljci (eng. labeled compartments) za grupisanje karakteristika iste vrste, što je dozvoljeno (ali ne i obavezno) u SysML-u. SysML definiše određene standardne odeljke, a dozvoljeno je da korisnici definišu dodatne.



Slika 3 – Domen radarskog osmatranja
 Рис. 3 – Диапазон радиолокационного наблюдения
 Figure 3 – Radar surveillance domain

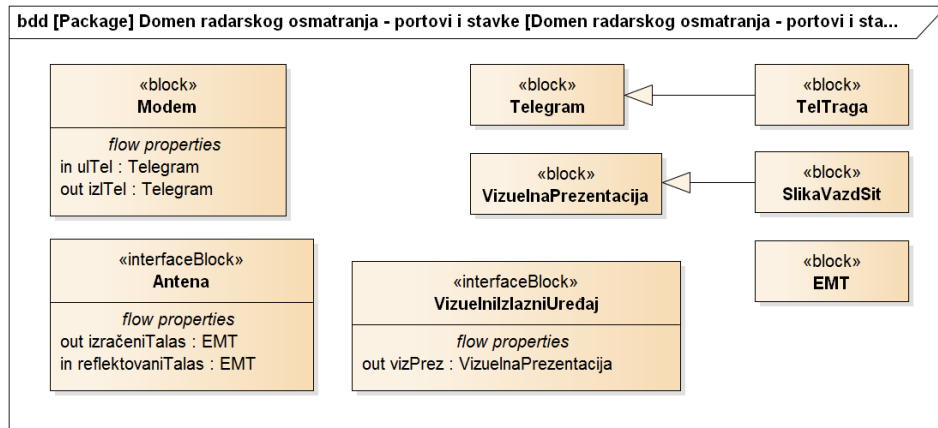
Dijagram definicije blokova na slici 5 definiše tipove portova i stavki (eng. items) u domenu radarskog osmatranja. Jedno od novih vrsta svojstava koja uvodi SysML je svojstvo toka (eng. flow property). Svojstva toka specificiraju vrste stavki koje mogu „teći” (tj. razmenjivati se) između bloka i njegove okoline (OMG, 2012). Stavke mogu biti podaci, ali i materijalni objekti ili energija. Jedno svojstvo toka specificira jednu vrstu stavki koja može teći u blok ili iz bloka. Smer toka se specificira prefiksom koji može imati vrednost *in*, *out* ili *inout*. Blok *Antena* na slici 5 ima dva svojstva toka. Svojstvo toka *izračeniTalas* je tipa *EMT* i ima prefiks *out*, što znači da iz bloka *Antena* mogu izlaziti stavke tipa *EMT*. Svojstvo toka *reflektovaniTalas* definiše da u blok *Antena* mogu ulaziti stavke tipa *EMT*.



Slika 4 – Dekompozicija domena radarskog osmatranja
 Puc. 4 – Декомпозиция поля радиолокационного наблюдения
 Figure 4 – Radar surveillance domain decomposition

Port čiji je tip blok koji ima svojstva toka može unutar svog grafičkog simbola imati strelicu koja ukazuje na smer svojstava toka u odnosu na blok koji je vlasnik porta (OMG, 2012). Tip porta *pokazivač* bloka *Osmatrački radar* je blok *VizuelniIzlazniUređaj* koji ima jedno svojstvo toka sa smerom *out*. Zato na slici 3 port *pokazivač* ima strelicu čiji je smer od bloka *Osmatrački radar* ka spoljašnjosti. Ukoliko je tip porta blok koji ima svojstva toka sa različitim smerovima ili svojstva toka sa smerom *inout*, strelica unutar grafičkog simbola porta ima oblik kao kod portova *modem* i *antena* na slici 3.

Tok stavki (eng. item flow) specificira stavke koje se prenose preko konektora ili asocijacije (OMG, 2012). Formalno, tok stavki je stereotip *«ItemFlow»* toka informacija (eng. information flow) iz UML-a. Notacija za tok stavki je slična kao za tok informacija: na konektor se dodaje popunjeni vrh strelice u smeru toka pored kojeg stoji ime tipa stavke koja se prenosi (slika 3). Dok svojstva toka definišu šta može da teče u blok ili iz bloka, tokovi stavki definišu šta zaista teče između blokova u konkretnom kontekstu (OMG, 2012). Iz definicije bloka *Modem* (slika 5) sledi da u port *modem* (koji je tipa *Modem*) mogu ulaziti i iz njega izlaziti stavke tipa *Telegram*. Međutim, tok stavki na konektoru koji spaja port *modem* i deo *oC* (slika 3) specificira da u port *modem* ulaze i iz njega izlaze stavke tipa *TelTraga*, koji predstavljaju jednu specijalizovanu vrstu telegrama.



Slika 5 – Portovi i stavke u domenu radarskog osmatranja
 Рус. 5 – Порт и запись в зоне радиолокационного наблюдения
 Figure 5 – Ports and items within the radar surveillance domain

Pored standardnih portova koji su preuzeti iz UML-a, u SysML-u postoje još dve vrste portova: potpun port (eng. full port) i port zastupnik (eng. proxy port). Port zastupnik identifikuje karakteristike bloka koji je vlasnik porta (ili karakteristike unutrašnjih delova tog bloka), a koje su raspoložive spoljašnjim blokovima. Akcija nad portom zastupnikom ekvivalentna je akciji nad blokom ili delom koji taj port zastupa. Tip porta zastupnika mora biti interfejs-blok (eng. interface block). Interfejs-blok je posebna vrsta bloka koja ne može imati ponašanje i unutrašnje delove. Sa druge strane, potpun port predstavlja zaseban deo sistema koji može imati svoje ponašanje i unutrašnje delove. Korišćenje pomenutih specijalizovanih vrsta portova nije obavezujuće. Korisnik može izabrati da li će na nekom mestu koristiti potpun port, port zastupnik ili „običan” port, u zavisnosti od primenjene metodologije (OMG, 2012). Notacija za potpune portove i portove zastupnike može se videti na slikama 3 i 4, dok se na slici 5 može videti notacija za interfejs-blokovne.

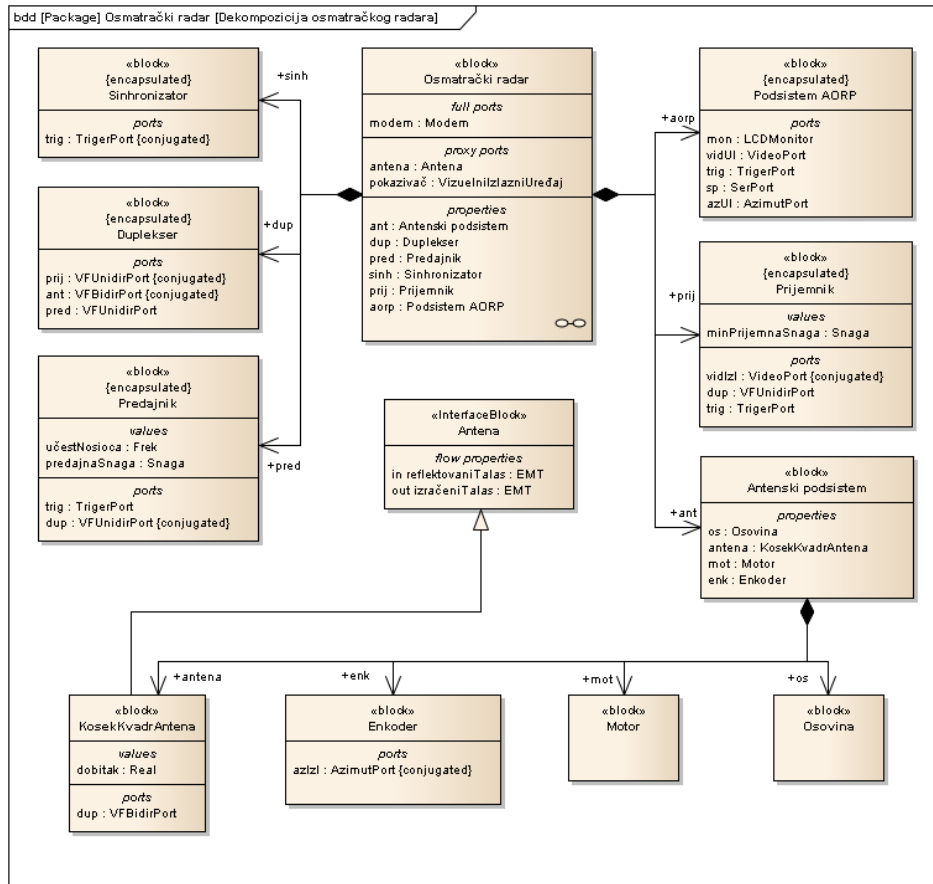
Na slikama 6 i 7 predstavljeni su dijagrami koji prikazuju dekompoziciju, i unutrašnju strukturu bloka *Osmatrački radar*, respektivno. Treba napomenuti da ovde nije bio cilj da se radar modelira potpuno tačno i sa svim detaljima već da se ilustruje SysML notacija.

Stereotip «*block*» uvodi atribut *isEncapsulated* tipa *Boolean*. Ako za neki blok ovaj atribut ima vrednost *true*, dati blok je kapsuliran (eng. encapsulated) i posmatra se kao crna kutija – veza sa ovim blokom može se ostvariti isključivo putem njegovih portova ili direktno sa njegovom spoljnom ivicom. Ako pomenuti atribut ima vrednost *false* ili vrednost uopšte nije definisana, blok nije kapsuliran i mogu se ostvarivati veze i sa njegovim unutrašnjim delovima (OMG, 2012). Na slici 6 blokovi

Sinhronizator, Duplekser, Predajnik, Prijemnik i Podsystem za akviziciju obradu razmenu i prikaz (AORP) kapsulirani su (uočiti notaciju), dok ostali blokovi nisu kapsulirani. Na slici 7 može se videti kako su delovi *dup* i *aorp* povezani konektorima sa unutrašnjim delovima dela *ant* čiji je tip blok *Antenski podsystem* koji nije kapsuliran.

Vezivni konektor (eng. binding connector) jeste konektor koji specificira da svojstva na njegovim krajevima imaju istu vrednost. Ukoliko su tipovi tih svojstava blokovi, vezivni konektor označava da se instance ovih svojstava odnose na istu instancu bloka (OMG, 2012). Vezivni konektor prikazuje se kao konektor sa ključnom reči «*equal*». Tako se, na primer, na slici 7 instance porta zastupnika *antena* i instance dela *ant.antena* odnose na istu instancu, čime je zapravo definisano da port *antena* zastupa deo *ant.antena* (sve njegove karakteristike stavlja na raspolaganje spoljašnjim blokovima). Isto važi za port zastupnik *pokazivač* i port *aorp.mon*.

Tip vrednosti (eng. value type) jeste tip koji se može koristiti za izražavanje informacija o sistemu, ali čije vrednosti nemaju identitet (osim same vrednosti) i ne može im se pristupiti putem reference (OMG, 2012). Formalno, tip vrednosti je stereotip «*ValueType*» tipa podatka (eng. data type) iz UML-a. Specifikacija SysML definiše osnovne tipove vrednosti u paketu *PrimitiveValueTypes*. Tip vrednosti omogućava da se vrednosti pridruže vrsta veličine (eng. quantity kind) i jedinica (eng. unit). U paketu *Tipovi*, čiji je sadržaj prikazan na slici 8, definisani su tipovi vrednosti korišćeni u modelu osmatračkog radara. U ugnježdenom paketu *Fizičke veličine* definisane su vrste veličina, a u ugnježdenom paketu *Jedinice* definisane su jedinice mere vezane za ove veličine. Za svaki od definisanih tipova vrednosti specificirana je jedinica (pomoću atributa *unit*), dok vrste veličina nisu eksplicitno navedene, jer se implicitno mogu izvesti iz jedinica. Primera radi, vrednosti čiji je tip *Brz* odnose se na brzinu izraženu u metrima u sekundi. Vrednosno svojstvo (eng. value property) jeste jedna od novih vrsta svojstava koje uvodi SysML. Tip vrednosnog svojstva mora biti tip vrednosti. Vrednosna svojstva blokova svrstavaju se u standardni odeljak *values* (slike 4 i 6).



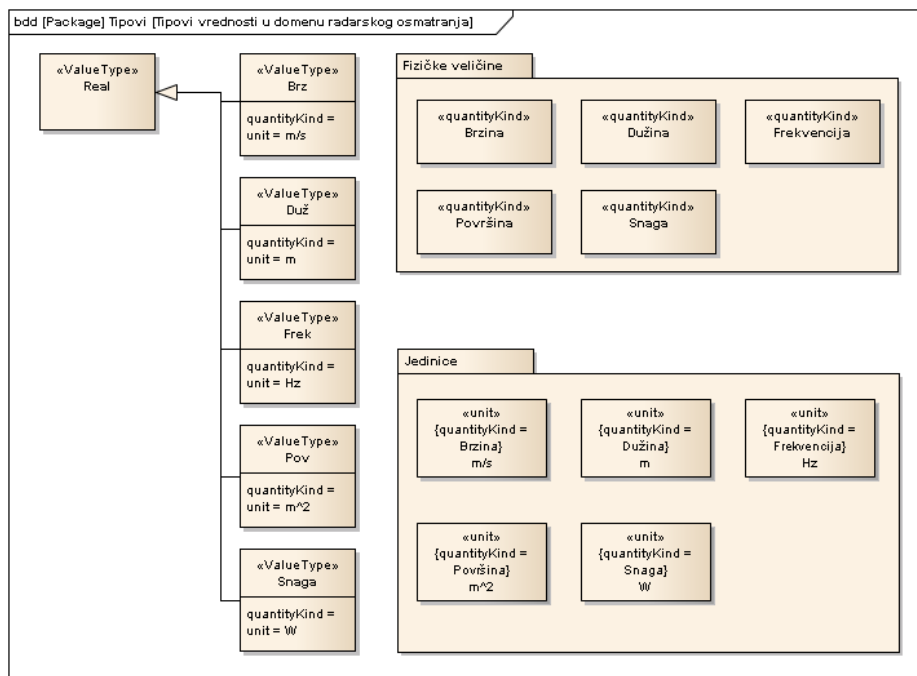
Slika 6 – Dekompozicija bloka Osmatrački radar

Рис. 6 – Декомпозиция блока радиолокационной станции кругового обзора
 Figure 6 – Decomposition of the Osmatrački radar (surveillance radar) block

Parametarski dijagram

U SysML-u, ograničavajući blok (eng. constraint block) jeste vrsta bloka koja predstavlja osnovu za integraciju analitičkih modela (npr. model performansi ili pouzdanosti) sa drugim vrstama modela. Ograničavajući blok uključuje ograničenje (matematički ili logički izraz proizvoljne složenosti) i parametre ograničenja (tj. parametre koji se koriste u pomenutom izrazu), a koristi se radi ograničavanja svojstava drugih blokova (OMG, 2012). Pod uslovom da je dovoljno uopšten, dati ograničavajući blok može se ponovo upotrebiti u mnogim različitim kontekstima.

svojstva u standardni odeljak *parameters*. Ograničavajuće svojstvo (eng. constraint property) jeste svojstvo čiji je tip ograničavajući blok. Ograničavajuća svojstva bloka svrstavaju se u standardni odeljak *constraints*. Blok *KontekstMaksimalnogDometa* predstavlja kontekst za pomenutu analizu maksimalnog dometa. Ovaj blok sadrži dva ograničavajuća bloka: blok *RadarskaJednačina* koji definiše zavisnost snage primljenog signala od snage predajnog signala i različitih parametara radarskog sistema i osmatranog objekta i blok *JednačinaTalasneDužine* koji definiše međusobnu zavisnost talasne dužine, učestanosti i brzine prostiranja talasa. Blok *KontekstMaksimalnogDometa* ima i referencu na blok *Domen radarskog osmatranja*.

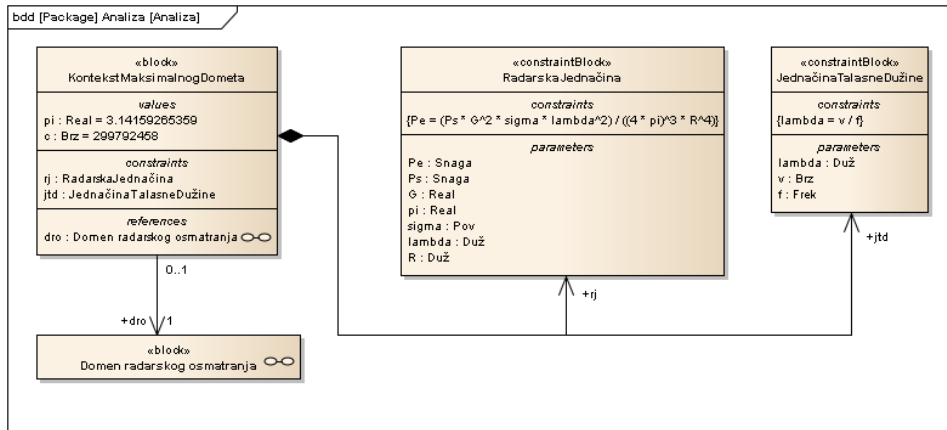


Slika 8 – Tipovi vrednosti u domenu radarskog osmatranja

Рис. 8 – Типы значений в зоне радиолокационного наблюдения

Figure 8 – Value types in the radar surveillance domain

Na slici 10 prikazan je parametarski dijagram koji se odnosi na blok *KontekstMaksimalnogDometa*. Ovde se vidi kako su parametri ograničavajućih blokova povezani sa svojstvima bloka *KontekstMaksimalnogDometa* i ugnježenim svojstvima blokova u „je deo” hijerarhiji bloka *Domen radarskog osmatranja*, radi modelovanja matematičkih relacija koje postoje između maksimalne daljine otkrivanja pokretnog objekta (*dro.pokObj.daljina*) i drugih relevantnih parametara.



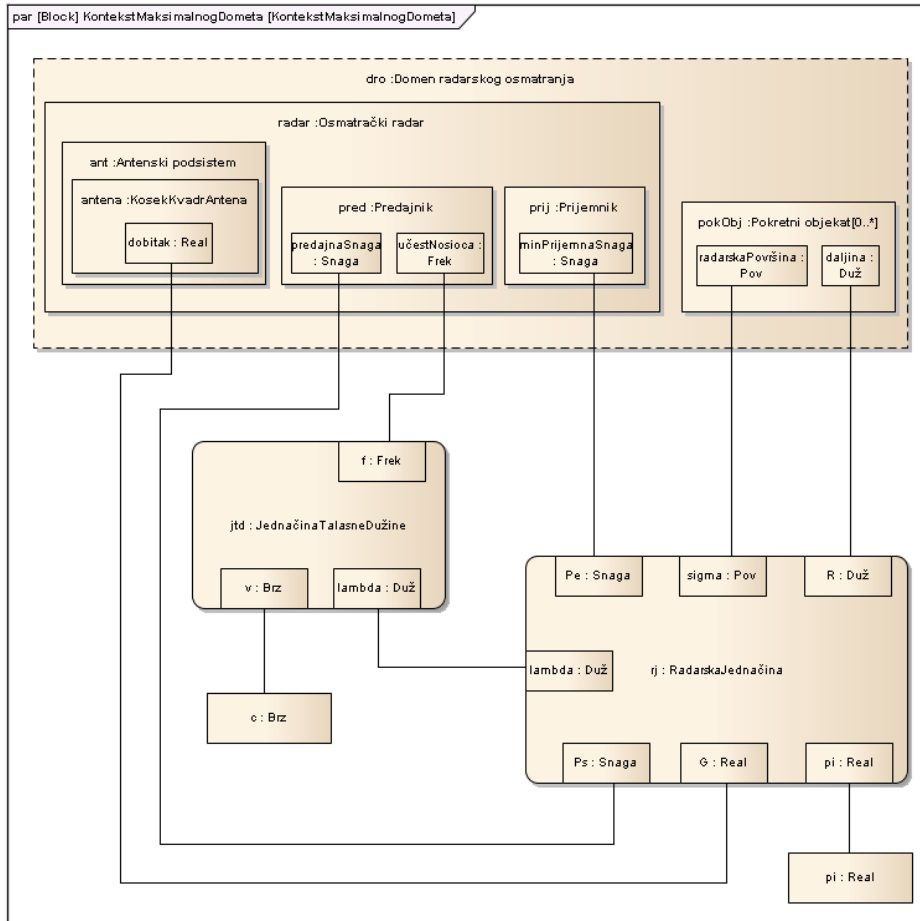
Slika 9 – Definicija konteksta za analizu maksimalnog dometa radara
 Рис. 9 – Определение контекста для анализа максимального диапазона радиолокационного наблюдения
 Figure 9 – Maximal range analysis context

Dijagram aktivnosti

U UML-u aktivnosti su klase čije su instance izvršenja aktivnosti (OMG, 2012). Samim tim, aktivnosti se mogu pojaviti na dijagramima definicije blokova i mogu učestvovati u relacijama generalizacije i asocijacije. SysML uvodi pojašnjenje semantike kompozicije između dve aktivnosti i između aktivnosti i tipa objektnog čvora. Kreiranjem instance aktivnosti započinje se izvršavanje aktivnosti, a uništavanjem instance aktivnosti izvršavanje se zaustavlja. Ako jedna aktivnost poziva drugu aktivnost onda ove dve aktivnosti mogu biti povezane relacijom kompozicije, pri čemu je pozivajuća aktivnost na strani celine, a pozivana aktivnost na strani dela. Gornja granica multiplikativnosti na strani dela ograničava broj konkurentnih izvršavanja pozivane aktivnosti. Donja granica multiplikativnosti na strani dela uvek je nula, jer tokom izvršavanja pozivajuće aktivnosti postoji vremenski interval u kojem se pozivana aktivnost ne izvršava. Na slici 11 prikazan je dijagram definicije blokova koji prikazuje relacije između aktivnosti *Osmatranje* i njenih podaktivnosti. Ovakav prikaz podseća na klasičnu funkcionalnu dekompoziciju.

Na slici 11 prikazane su i relacije između aktivnosti *Osmatranje* i klasifikatora koji su tipovi objektnih čvorova u ovoj aktivnosti (prikazani ružičastom bojom). Relacija kompozicije ukazuje na to da objektni čvorovi ne mogu postojati nakon završetka izvršavanja aktivnosti. Gornja granica multiplikativnosti na strani dela (tj. objektnog čvora) ograničava broj instanci koje istovremeno mogu postojati u objektnom čvoru. Donja

granica multiplikativnosti na strani dela uvek je nula, pošto tokom izvršavanja aktivnosti postoje intervali u kojima je objektni čvor prazan.

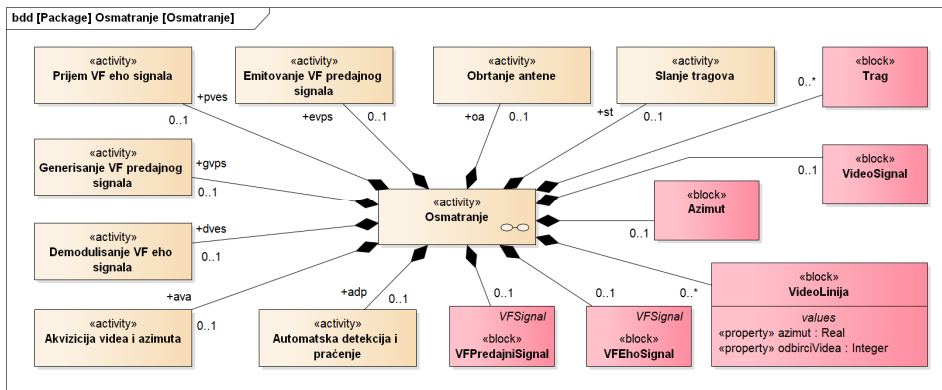


Slika 10 – Matematičke relacije vazane za analizu maksimalnog dometa radara
 Рис. 10 – Математические отношения, связанные с анализом максимального
 диапазона радиолокационного наблюдения
 Figure 10 – Mathematical relations for the maximal range analysis

Na slici 12 prikazan je dijagram aktivnosti za aktivnost *Osmatranje*. Ovde najpre treba uočiti da su imena akcija pozivanja ponašanja (eng. call behavior action) ista kao imena uloga pozvanih aktivnosti na slici 11. To je primer primene jednog od pravila konzistentnosti koja uvodi SysML.

Takođe, treba uočiti (iako nije u pitanju specifičnost SysML-a) da u aktivnosti *Osmatranje* postoji nekoliko paralelnih tokova kontrole čije se

izvršavanje može prekinuti jedino izlaskom iz prekidljivog regiona aktivnosti (eng. interruptible activity region) pojavom signala *Isključenje*.



Slika 11 – Dekompozicija aktivnosti Osmatranje
 Puc. 11 – Декомпозиция деятельности Наблюдения
 Figure 11 – Decomposition of the Osmatranje (surveillance) activity

Stereotip *«rate»* može se primeniti na granu ili na parametar i omogućava da se pomoću njegovog atributa *rate* specifikira brzina toka (eng. rate of flow), tj. broj objekata ili vrednosti koje prelaze preko grane, odnosno teku u/iz parametra u jedinici vremena (OMG, 2012). Tok može biti diskretan ili kontinualan. Stereotip *«discrete»* je specijalizacija stereotipa *«rate»* i služi za označavanje diskretnih tokova (vremenski interval između prolazaka stavki koje teku je različit od nule). Stereotip *«continuous»* takođe je specijalizacija stereotipa *«rate»*, a namenjen je za predstavljanje kontinualnih tokova kao što su signali kontinualni u vremenu ili kontinualni tok energije. Na slici 12 prikazano je kako se pomoću kontinualnih tokova može modelovati tok kontinualnih signala (*VFPredajniSignal*, *VFEhoSignal* i *VideoSignal*) između aktivnosti. Tok vrednosti azimuta antene modelovan je diskretnim tokom, pošto se azimut najčešće određuje pomoću optičkog enkodera (u ovom slučaju navedena je brzina toka 819.2 Hz, što odgovara slučaju kada se koristi 13-bitni enkoder, a perioda obrtanja antene je 10 sekundi). Paketski prenos video linija i tragova takođe je modelovan diskretnim tokom.

U UML-u, ako je token koji pristiže u objektni čvor odbijen od izlazne grane (ili akcije kojoj je objektni čvor ulazni pin), token se zadržava u čvoru sve dok se ne steknu uslovi za njegov odlazak. SysML uvodi stereotip *«noBuffer»* koji, ako se primeni na objektni čvor, menja prethodno opisano ponašanje, tako da se token koji ne može da napusti

čvor odbacuje. Jedna od tipičnih primena ovog stereotipa javlja se kod modelovanja električnih signala (OMG, 2012), što je primenjeno i na slici 12.

SysML uvodi i stereotip «*overwrite*». Kada token pristigne na pun objektni čvor na koji je primenjen ovaj stereotip (pun objektni čvor sadrži maksimalan dozvoljen broj tokena), dati token zameniće jedan od tokena koji su već u čvoru. Na slici 12 objektni čvor tipa *Azimuth* ima stereotip «*overwrite*», što uzevši u obzir da je gornja granica multiplikativnosti za ovaj čvor 1 (slika 11), osigurava da će se u ovom čvoru čuvati samo vrednost azimuta koja je poslednja pristigla.

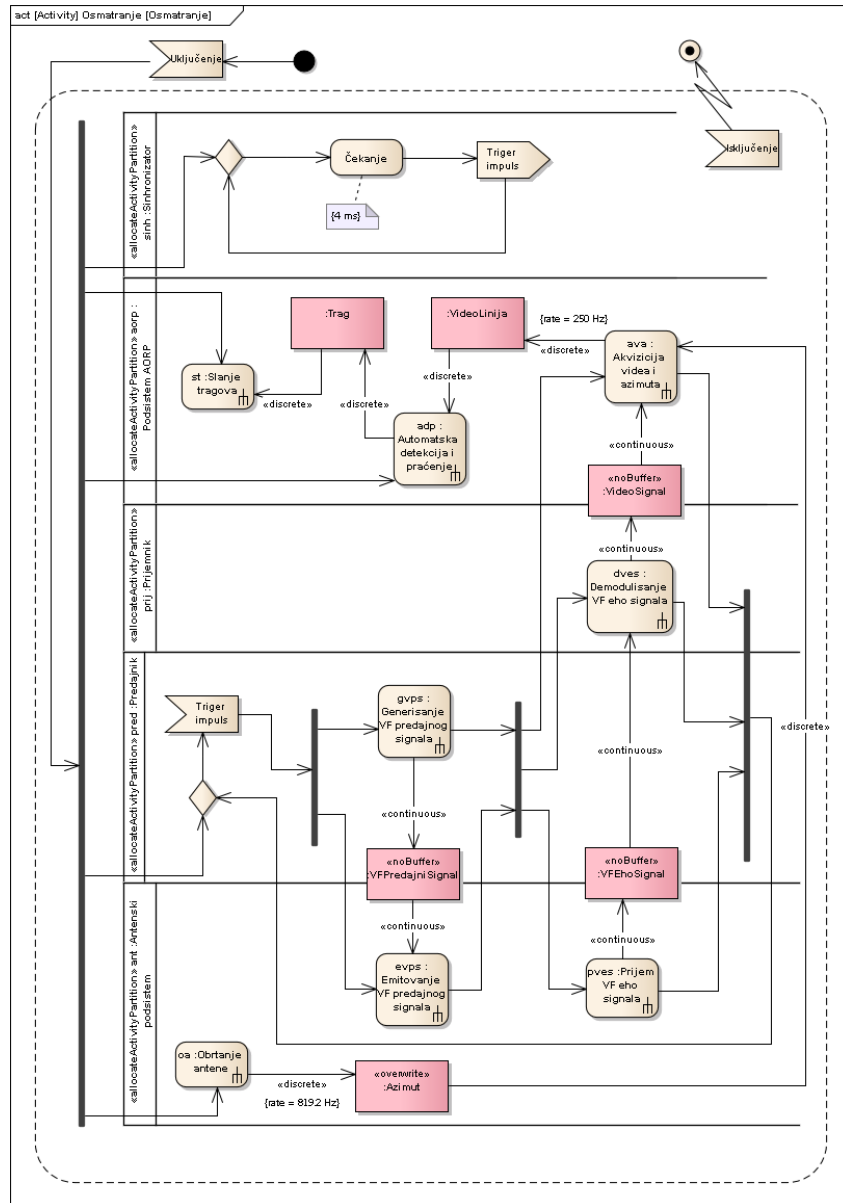
Na slici 12 prikazano je kako se na dijagramu aktivnosti može označiti alokacija ponašanja (eng. behavior allocation) pomoću stereotipa «*allocateActivityPartition*» primenjenog na pregradu aktivnosti (eng. activity partition). Alokacija je termin koji se u sistemskom inženjerstvu koristi za označavanje organizovane asocijacije ili mapiranja među elementima koji pripadaju različitim strukturama ili hijerarhijama u modelu (OMG, 2012). Alokacija ponašanja je jedna od tipičnih vrsta alokacije u sistemskom inženjerstvu, a odnosi se na mapiranje između elemenata funkcionalnog i strukturnog modela. Tako je, na primer, na slici 12 ponašanje opisano aktivnošću *Demodulisanje VF eho signala* asocirano sa strukturnim elementom (svojstvom) *prij* tipa *Prijemnik*, itd.

Dijagram zahteva

U SysML-u zahtev (eng. requirement) jeste stereotip klase namenjen za modelovanje zahteva predstavljenih u tekstualnom obliku. Osnovni atributi koje uvodi ovaj stereotip su *id* (jedinstveni identifikator zahteva) i *text* (tekst zahteva). Dijagram zahteva namenjen je za prikaz zahteva i njihovih relacija sa drugim elementima modela. Zahtevi se mogu pojavljivati i na drugim dijagramima radi prikazivanja njihovih relacija sa drugim elementima. Na slici 13 prikazan je dijagram zahteva koji prikazuje deo specifikacije radara koji se odnosi na uslove okoline prilikom upotrebe.

Stereotip «*deriveReq*», primenjen na relaciju zavisnosti između dva zahteva, označava da je zahtev na klijentskoj strani relacije izveden iz drugog zahteva, npr. zahtev na nižem nivou apstrakcije koji je izveden iz sistemskog zahteva (OMG, 2012). Na slici 13, iz zahteva *Podsistemi unutar kabine – uslovi okoline (upotreba)*, koji predstavlja sistemski zahtev vezan za uslove okoline, izveden je zahtev *Prijemnik – uslovi okoline (upotreba)* koji definiše uslove okoline za jedan konkretan podsistem. Pored same klase uslova okoline, standard EN 300 019 definiše i specifikaciju testova kojima se proverava da li neki uređaj zadovoljava zahtev za funkcionisanjem u uslovima okoline opisanim datom klasom. Ovde je to modelirano izvođenjem zahteva

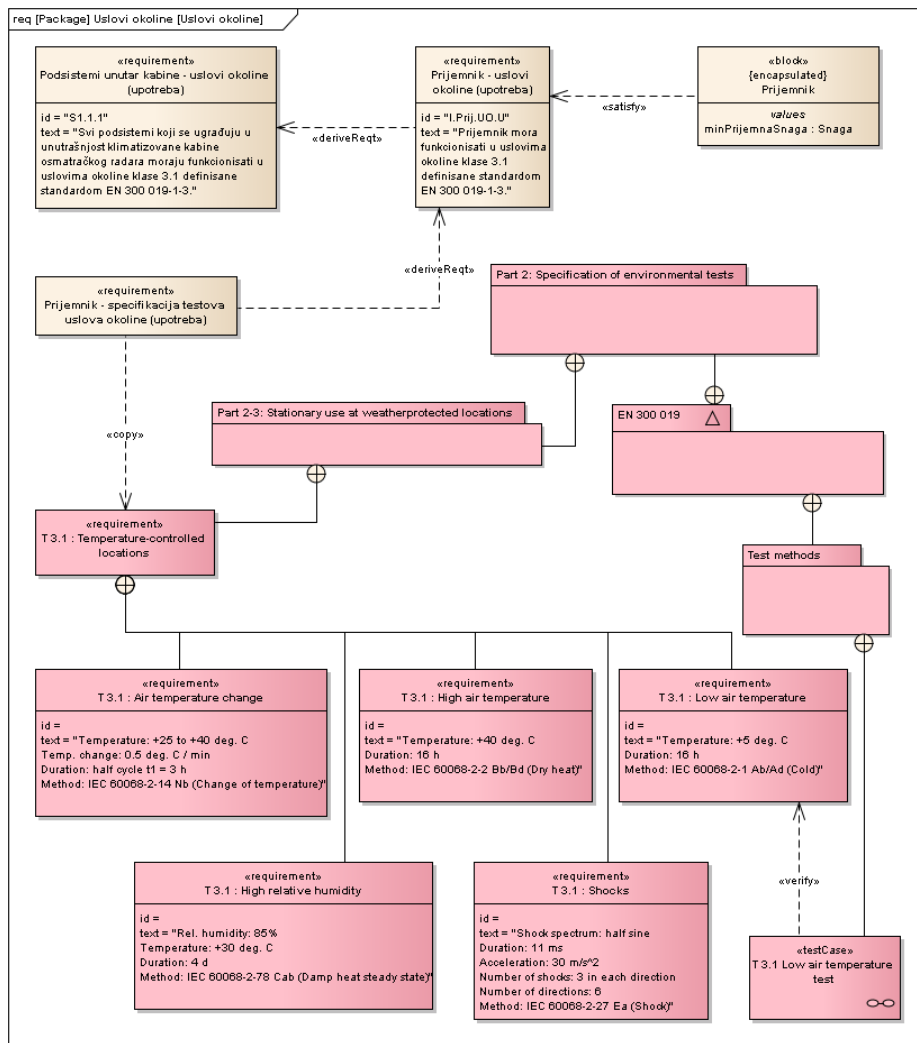
Prijemnik – specifikacija testova uslova okoline (upotreba) iz zahteva
 Prijemnik – uslovi okoline (upotreba).



Slika 12 – Aktivnost Osmatranje
 Рис. 12 – Действия Наблюдения
 Figure 12 – Osmatranje (surveillance) activity

Jevtić, M., et al, Primena SysML na pojednostavljen model osmatračkog radara, pp. 160-185

Stereotip «copy» može se primeniti na relaciju zavisnosti između dva zahteva kako bi označio da je tekst zahteva na klijentskoj strani relacije nepromenljiva (eng. read-only) kopija teksta drugog zahteva. Svrha ove relacije jeste da omogući ponovnu upotrebu zahteva u različitim kontekstima. Sa slike 13 se vidi da je zahtev *Prijemnik – specifikacija testova uslova okoline (upotreba)* zapravo kopija zahteva T3.1 – *Temperature-controlled locations*, koji pripada modelu EN 300 019.



Slika 13 – Relacije vezane za zahteve
 Рис. 13 – Связь между установленными требованиями
 Figure 13 – Relations which pertain to requirements

Zahtev *T3.1 – Temperature-controlled locations* može se ponovno upotrebiti u drugim projektima, jer predstavlja univerzalnu specifikaciju primenljivu na bilo koji uređaj. Sa slike 13 takođe se vidi kako se složen zahtev može razložiti na jednostavnije, korišćenjem relacije ugnježđivanja.

Slučaj testiranja (eng. test case) označava metod kojim se proverava da li je zahtev zadovoljen (OMG, 2012). Relacija zavisnosti sa stereotipom «*verify*» može postojati između zahteva i slučaja testiranja ili drugog elementa modela koji može da odredi da li sistem zadovoljava zahtev (OMG, 2012). Na slici 13 je pomoću ove relacije označeno da se zahtev *T3.1 Low air temperature* može proveriti pomoću slučaja testiranja *T3.1 Low air temperature test*. Formalno, slučaj testiranja je stereotip «*testCase*» operacije ili ponašanja, pa se može modelovati pomoću dijagrama ponašanja.

Relacijom zavisnosti na koju je primenjen stereotip «*satisfy*» može se označiti element modela koji ispunjava neki zahtev. Ovom relacijom je na slici 13 označeno da blok *Prijemnik* ispunjava zahtev *Prijemnik – uslovi okoline (upotreba)* (prijemnik je projektovan tako da zadovolji zahtev za funkcionisanjem u određenim uslovima okoline).

SysML u praksi

Sistemske inženjer, suočen sa dilemom da li da uvede SysML u svoju praksu, najverovatnije bi postavio i sledeća dva pitanja: 1) da li je SysML „zaživeo”, tj. kako ga je prihvatila stručna i naučna zajednica i kakva je perspektiva njegove primene i 2) kakvi alati za SysML modelovanje su raspoloživi? U ovom odeljku pokušaćemo odgovoriti na ta pitanja.

Prihvaćenost i perspektiva SysML-a

Godine 2009. OMG je sproveo onlajn anketu sa ciljem da se korisnicima SysML-a i proizvođačima alata omogući da daju povratne informacije koje će pomoći da se unaprede buduće verzije SysML-a. Detaljni rezultati ankete navedeni su u (Cloutier, Bone, 2010.), dok su ključne informacije sumirane u (Bone, Cloutier, 2010.). Anketa je bila otvorenog tipa, a u njoj je učestvovalo 128 ispitanika iz 45 organizacija iz 16 zemalja (Bone, Cloutier, 2010.). Među anketiranim najviše je bilo sistemskih inženjera (Bone, Cloutier, 2010.). Anketa je pružila puno informacija o tekućem statusu SysML-a i SIZM (Bone, Cloutier, 2010.). Jedno od pitanja odnosilo se na to u kojoj meri se u organizacijama anketiranih lica planira upotreba SysML-a u budućnosti. Više od dve

trećine anketiranih (71,3%) odgovorilo je da njihova organizacija već koristi SysML ili planira da ga uvede u svoju praksu (Bone, Cloutier, 2010.). To pokazuje da je stručna zajednica dobro prihvatila SysML i da on predstavlja perspektivan jezik za modelovanje.

Sa druge strane, relativno kratkom pretragom nama dostupnih indeksnih baza uspehi smo da pronađemo preko 40 radova koji se u nekoj meri dotiču SysML-a. U mnogim od ovih radova „SysML” se pojavljuje u naslovu ili u listi ključnih reči. Pregled pomenutih radova prevazilazi obim ovog teksta, ali samo njihovo postojanje ukazuje na to da SysML ima značajan uticaj i u akademskim okvirima.

SysML alati

Alati za SysML modelovanje sa komercijalnom licencom uključuju: *IBM Rational Rhapsody*, *NoMagic MagicDraw*, *Sparx Systems Enterprise Architect*, *Artisan Studio* i *IBM Rational Tau*. Treba napomenuti da se kod pojedinih komercijalnih alata podrška za SysML modelovanje dobija ili kupovinom posebnog dodatka ili kupovinom specijalne verzije alata („systems engineering edition” i sl.). Besplatni SysML alati uključuju: *TOPCASED* i *Papyrus for SysML*. Postoji i alat *Modelio*, koji ima i komercijalnu i besplatnu verziju. Verzija *Modelio*-a otvorenog koda ne podržava modelovanje zahteva.

Tokom upoznavanja sa SysML-om testirali smo jedan besplatan (*TOPCASED*) i jedan komercijalan (*Sparx Systems Enterprise Architect*) alat. Besplatan alat je u verziji koju smo koristili imao grešku koju nije bilo moguće prevazići (nije bilo moguće prikazati relaciju ugnježđivanja na dijagramu zahteva). Kod komercijalnog alata, pomoću kojeg su izrađeni modeli i dijagrami korišćeni u ovom tekstu, takođe su postojali propusti, koji su ili bili superficijalni ili ih je bilo moguće prevazići. Ako uzmemo u obzir i prethodno pomenutu nemogućnost modelovanja zahteva u drugom besplatnom alatu (*Modelio*), stiče se utisak da besplatni SysML alati još uvek nisu dostigli nivo zrelosti komercijalnih konkurenata. Nadamo se da će se to u budućnosti promeniti.

Zaključak

Jezik za modelovanje SysML je, pre svega, namenjen za primenu u SIZM, ali je generalno primenljiv svuda gde je potrebno modelovati složene sisteme. Glavne razlike u odnosu na UML odnose se na dve nove vrste dijagrama koji omogućuju modelovanje tekstualnih zahteva i

integraciju analitičkih modela sa drugim vrstama modela, kao i na brojne izmene u strukturnim dijagramima i u dijagramu aktivnosti. Te izmene omogućavaju modelovanje najrazličitijih sistema ili elemenata sistema, kao što su elektronski ili mehanički sistemi/elementi, postrojenja, organizacije, itd. U ovom članku SysML je ilustrovan na tipičnom primeru složenog sistema čiji razvoj bi zahtevao interdisciplinarni pristup – primeru osmatračkog radara. Ovaj primer nije preuzet iz postojeće literature već je smišljen za potrebe ovog članka.

U praksi je SysML dobro prihvaćen, kako od stručnjaka – sistemskih inženjera, tako i od naučne zajednice. Mnoge organizacije već primenjuju SysML ili planiraju njegovu primenu u budućnosti. Na raspolaganju su i brojni alati za SysML modelovanje, kako komercijalni, tako i besplatni.

Na osnovu svega što je rečeno, kao i na osnovu ličnog iskustva, može se zaključiti da je SysML ekspresivan, koristan i perspektivan jezik za modelovanje, čija bi primena mogla pospešiti komunikaciju među zainteresovanim stranama u interdisciplinarnim projektima i unaprediti proces razvoja složenih sistema.

Literatura

Bone, M., & Cloutier, R. 2010. The Current State of Model Based Systems Engineering: Results from the OMG SysML Request for Information 2009. . In: 8th Conference on Systems Engineering Research CSER, Hoboken, NJ., pp.225-232

Cloutier, R., & Bone, M. 2010. *Compilation of SysML RFI - Final Report*. Stevens Institute of Technology, School of Systems & Enterprises.

Estefan, J.A. 2008. *Survey of Model-Based Systems Engineering (MBSE) Methodologies, Rev. B*. INCOSE MBSE Initiative.

Friedenthal, S., Moore, A., & Steiner, R. 2008. *A Practical Guide to SysML: The Systems Modeling Language*. Burlington, MA: Morgan Kaufmann Publishers.

INCOSE. 2004. *Systems Engineering Handbook, Version 2a*.

OMG. 2012. *Systems Modeling Language (OMG SysML) Version 1.3*.

Tartalja, I. 2011. *Skripta za predmet Projektovanje softvera*. Beograd: Elektrotehnički fakultet Univerziteta u Beogradu, Katedra za računarsku tehniku i informatiku.

Weilkiens, T. 2007. *Systems Engineering with SysML/UML: Modeling, Analysis, Design*. Burlington, MA: Morgan Kaufmann Publishers.

ПРИМЕНЕНИЕ ЯЗЫКА SYSML НА УПРОЩЕННОЙ МОДЕЛИ РАДИОЛОКАЦИОННОГО НАБЛЮДЕНИЯ

Милош Д. Евтич^а, Синиша Р. Маринкович^б, Ивица Б. Марьянович^в

^а Университет в Белграде, Электротехнический факультет, г. Белград, Республика Сербия

^б Университет в Белграде, Институт «Михайло Пупин», г. Белград, Республика Сербия

^в Министерство обороны Республики Сербия, Военный контроль качества, г. Белград, Республика Сербия

ОБЛАСТЬ: информатика

ВИД СТАТЬИ: профессиональная статья

ЯЗЫК СТАТЬИ: сербский

Резюме:

Systems Modeling Language (SysML) представляет собой профиль Unified Modeling Language (UML), предназначенный для применения в системной инженерии. Недостаток литературы о SysML на сербском языке подтолкнул автора данной статьи к ее написанию, с целью представления этого интересного языка моделирования инженерному и академическому сообществу, при этом подчеркиваются различия между языками SysML и UML. Описание профиля приведено на основании конкретного примера – упрощенной модели радиолокационного наблюдения. Проведен краткий анализ практических аспектов SysML о его внедрении, перспективах дальнейшего применения и пр. В заключении приведены выводы о полезности применения языка SysML, его употребляемости и благоприятных перспективах в связи с его внедрением.

Ключевые слова: системная инженерия основанная на моделях, радиолокационное наблюдение, SysML, UML.

APPLICATION OF SYSML ON A SIMPLIFIED MODEL OF AN AIR SURVEILLANCE RADAR

Miloš D. Jevtić^а, Siniša R. Marinković^б, Ivica B. Marjanović^в

^а University of Belgrade, School of Electrical Engineering, Belgrade, Republic of Serbia,

^б University of Belgrade, Institute „Mihajlo Pupin”, Belgrade, Republic of Serbia,

^в Ministry of Defence of the Republic of Serbia, Military Quality Control, Belgrade, Republic of Serbia

FIELD: IT
ARTICLE TYPE: Professional Paper
ARTICLE LANGUAGE: Serbian

Summary:

Systems Modeling Language (SysML) is a Unified Modeling Language (UML) profile intended for application in systems engineering. This text is inspired by lack of literature on SysML in Serbian language. Its goal is to introduce this interesting modeling language to the engineering and academic community, with a special emphasis on the differences from UML. An overview is given, using an original example – a simplified model of an air surveillance radar. A short analysis of practical aspects of SysML is made – its scope of acceptance, the prospects of its further application, etc. We concluded that SysML is a useful, accepted and promising modeling language.

Key words: *Model-based systems engineering, Air surveillance radar, SysML, UML.*

Datum prijema članka / Дата получения работы / Paper received on: 13. 07. 2015.
Datum dostavljanja ispravki rukopisa / Дата получения исправленной версии работы / Manuscript corrections submitted on: 06. 11. 2015.
Datum konačnog prihvatanja članka za objavljivanje / Дата окончательного согласования работы / Paper accepted for publishing on: 08. 11. 2015.

© 2017 Autori. Objavio Vojnotehnički glasnik / Military Technical Courier (www.vtg.mod.gov.rs, vtg.mo.upr.srb). Ovo je članak otvorenog pristupa i distribuira se u skladu sa Creative Commons licencom (<http://creativecommons.org/licenses/by/3.0/rs/>).

© 2017 Авторы. Опубликовано в «Военно-технический вестник / Vojnotehnički glasnik / Military Technical Courier» (www.vtg.mod.gov.rs, vtg.mo.upr.srb). Данная статья в открытом доступе и распространяется в соответствии с лицензией «Creative Commons» (<http://creativecommons.org/licenses/by/3.0/rs/>).

© 2017 The Authors. Published by Vojnotehnički glasnik / Military Technical Courier (www.vtg.mod.gov.rs, vtg.mo.upr.srb). This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/rs/>).

