

Universitatea Politehnică București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Tema de curs INGINERIE SOFTWARE

Solutii de modelare a aplicatiilor de inginerie software : SysML

Profesor îndrumător:

Conf. Dr. Ing. Stancescu Stefan

Studenti: Buculei Theodor Cristian

Marin Victor Ionut

Mieila Adrian

Grupa: 441A (CTI)

Anul universitar
2014-2015

Cuprins

1) Prezentare generala a ingineriei sistemelor bazata pe modele	3
<i>BUCULEI Theodor Cristian, 441A</i>	
2) SysML si concepte de limbaj	4
<i>MIEILA Adrian, 441A</i>	
3) Tranzitia la SysML.....	9
<i>BUCULEI Theodor Cristian, 441A</i>	
4) Modelare SysML ca parte a ingineriei software(exemple si aplicatii finale)	11
<i>MARIN Victor Ionut, 441A</i>	
5) Portabilitatea SysML-ului	15
<i>BUCULEI Theodor Cristian, 441A</i>	
6) Concluzii	16
<i>MIEILA Adrian, 441A</i>	
7) Bibliografie.....	17

Solutii de modelare a aplicatiilor de inginerie software : SysML

BUCULEI Theodor Cristian, 441A

1. Prezentare generala a ingineriei sistemelor bazata pe modele

Ideea de SysML apare prima data în Ianuarie 2001 în urma unei decizii a Consiliului Internațional a Ingineriei Sistemelor(International Council on Systems Engineering =INCOSE) . Împreună cu Object Management Group (OMG) s-a lucrat la realizarea unei proceduri general acceptate de efectuare a diagramelor UML a aplicațiilor ingineresti. Aceasta a fost finalizată în Martie 2003 [Systems Engineering Request for Proposal (UML for SE RFP; OMG document ad/03-03-41)].^[1]

În anul 2003 Cris Kobryn și Sanford Friedenthal au organizat și co-prezidat asociația a □ □ SysML Partners”. Din această asociație făceau parte liderii din industria software, precum și o serie de developeri software [] .S-au dezvoltat specificațiile unui proiect open-source de dezvoltare. Acestea au fost date publicului larg în anul 2004, iar în Noiembrie 2005 au fost lansat primul lor produs open-source SysML 1.0a .

Ingineria sistemelor bazată pe modele (MBSE) reprezintă o modalitate prin care putem aplica o serie de principii stricte de modelare grafica asupra ingineriei de sisteme software. Aceste activități necesită cerințe și verificare suplimentară, analize funcționale și alocări, analize de performanțe și indicații pentru arhitecturile de sistem.^[2]

Ingineria sistemelor bazată pe modele oferă următoarele privilegii:

- Se expune o înțelegere a cerințelor și a modelului de sistem pentru a putea fi validate cerințele și identifică pericolele.
- Asistență în progresarea sistemelor complexe printr-un model de organizare ierarhică a modelelor de sistem, analizarea schimbărilor de cerințe și de structură și suport pentru dezvoltare incrementală.

¹ http://en.wikipedia.org/wiki/Systems_Modeling_Language#cite_note-rfp-3

² <http://sysml.org/sysml-partners/>

³ <http://www.modelbasedsystemsengineering.com/>

- Calitatea îmbunătățită a modelelor, ce se concretizează în ezitarea erorilor și formarea unor reprezentări complete.
- Suport pentru validare și verificarea în primele procese ale produsului și în timpul vieții acestuia, pentru a reduce riscurile de funcționare necorespunzătoare sau defectare.

Modelarea sistemelor se desfășoară pe mai multe etape, așadar modele pot fi: modele operaționale, modele de sistem, modele de componente.

Ingineria sistemelor bazată pe modele este o soluție tehnologică pentru ingineri care caută o trecere de la procesele tradiționale din ingineria sistemelor, care se bazează pe documente și se centrează pe cod, la procese mult mai eficiente care se bazează pe cerințe și se centrează pe arhitectură.³

MIEILA Adrian, 441A

2. SysML si concepte de limbaj

SysML este un limbaj de modelare grafică care vine în urma UML-ului, dezvoltat de OMG, INCOSE și AP233. Poate fi definit ca un profil UML care reprezintă un subset al UML 2, cu extensii.

SysML suportă analiza, verificarea, modelarea, specificarea și validarea unei game largi de sisteme și subsisteme. Aceste sisteme pot fi hardware, software, informație, procese și facilități.⁴

³ A Practical Guide to SysML, Second Edition: The Systems Modeling Language (The MK/OMG Press)

⁴ A Practical Guide to SysML, Second Edition: The Systems Modeling Language (The MK/OMG Press)

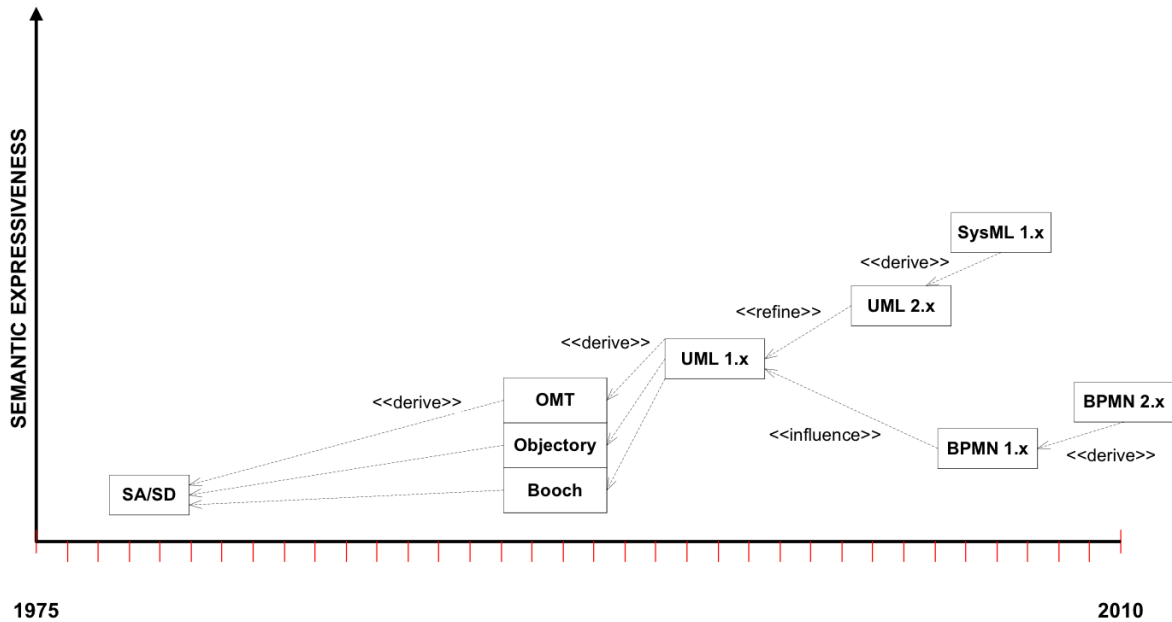


Fig 1. Evolutia limbajelor de modelare grafica (Sursa 5 din bibliografie)

SysML a fost prima data dezvoltat ca un proiect open source ca răspuns la UML, inițiat în 2003. Cea mai recentă versiune de SysML este v1.3. SysML și conține 9 tipuri de diagrame, majoritatea fiind comune cu limbajul UML2.

SysML conține patru diagrame esențiale: diagrama Bloc, Parametrică, de Cerințe și de Activitate. Aceste diagrame au fost alese, deoarece mai mult de 80% din limbajul SysML se bazează pe funcții ce folosesc aceste diagrame. Diagramele Bloc și de activitate sunt cele mai importante, deoarece sunt modificări ale diagramelor UML2, dovedit deja că sunt eficiente. Celelalte două diagrame sunt nou introduse.⁵

Cadrele diagramelor se împart în câmpurile header și conținut. În header este indicat contextul diagramei (tip, tip de model, nume de model, numele diagramei). Fiecare reprezintă un element model.⁶

⁵ <http://www.sysmlforum.com/sysml-faq/>

⁶ <http://www.sysmlforum.com/sysml-faq/>

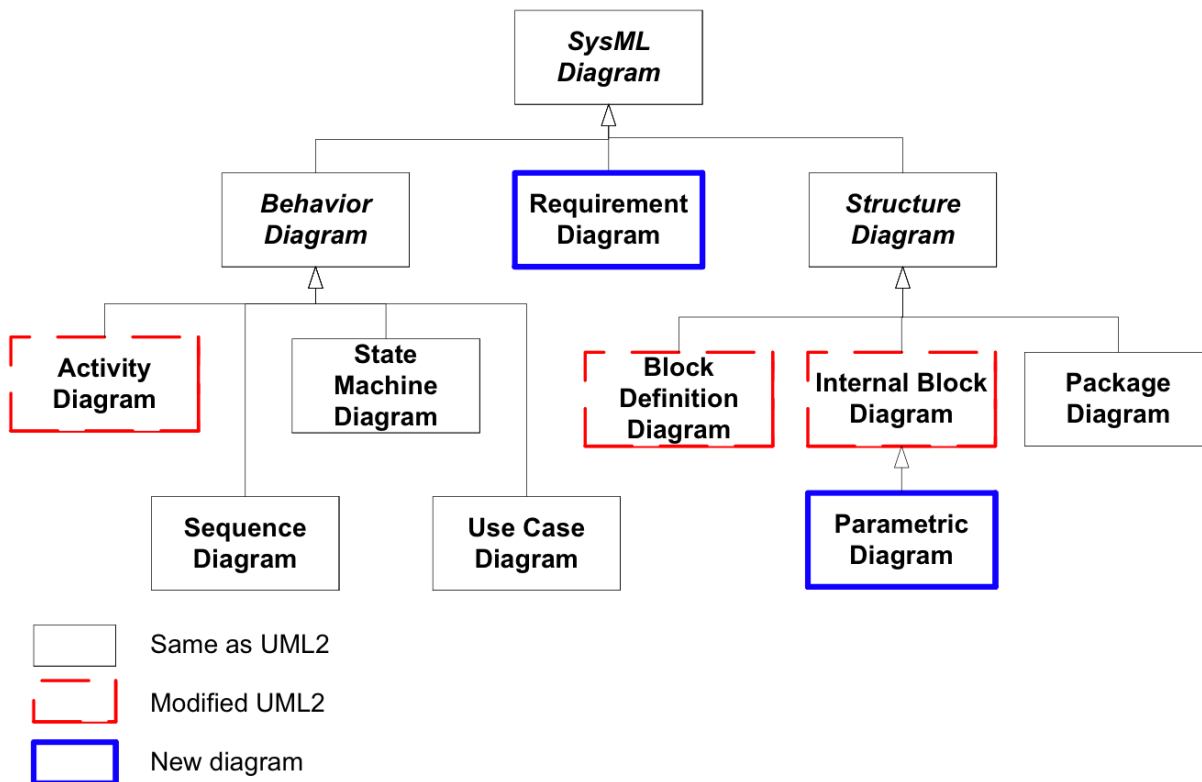


Fig 2. Diagrame SysML (Sursa 5 din bibliografie)

Modelarea grafică a unui sistem se bazează pe blocuri și relațiile dintre ele. Este obligatoriu să putem gestiona aceste blocuri, din punct de vedere al conținutului dar și din punct de vedere al legăturilor dintre ele. Diagrama de definire a blocului descrie relațiile între blocuri (compoziție, specializare, asociere), ea este „cutia neagră” a blocului.

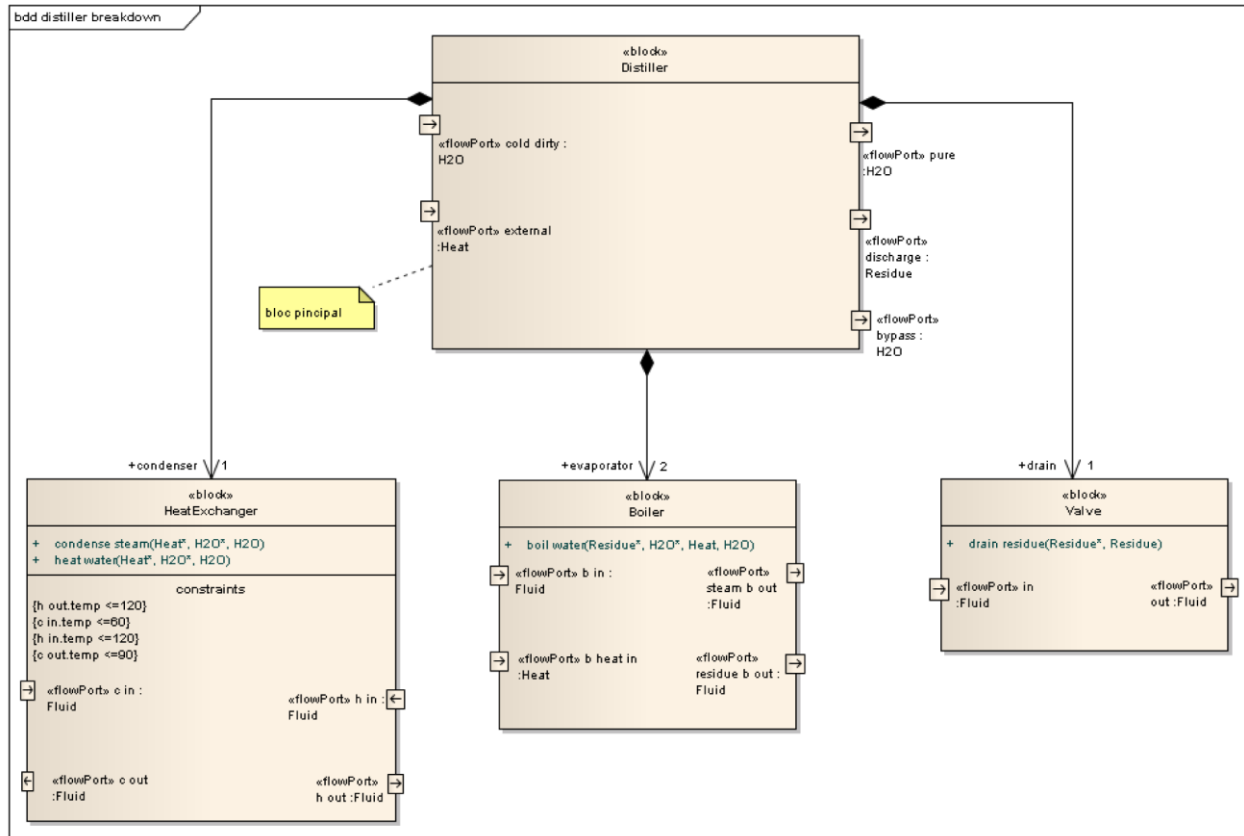
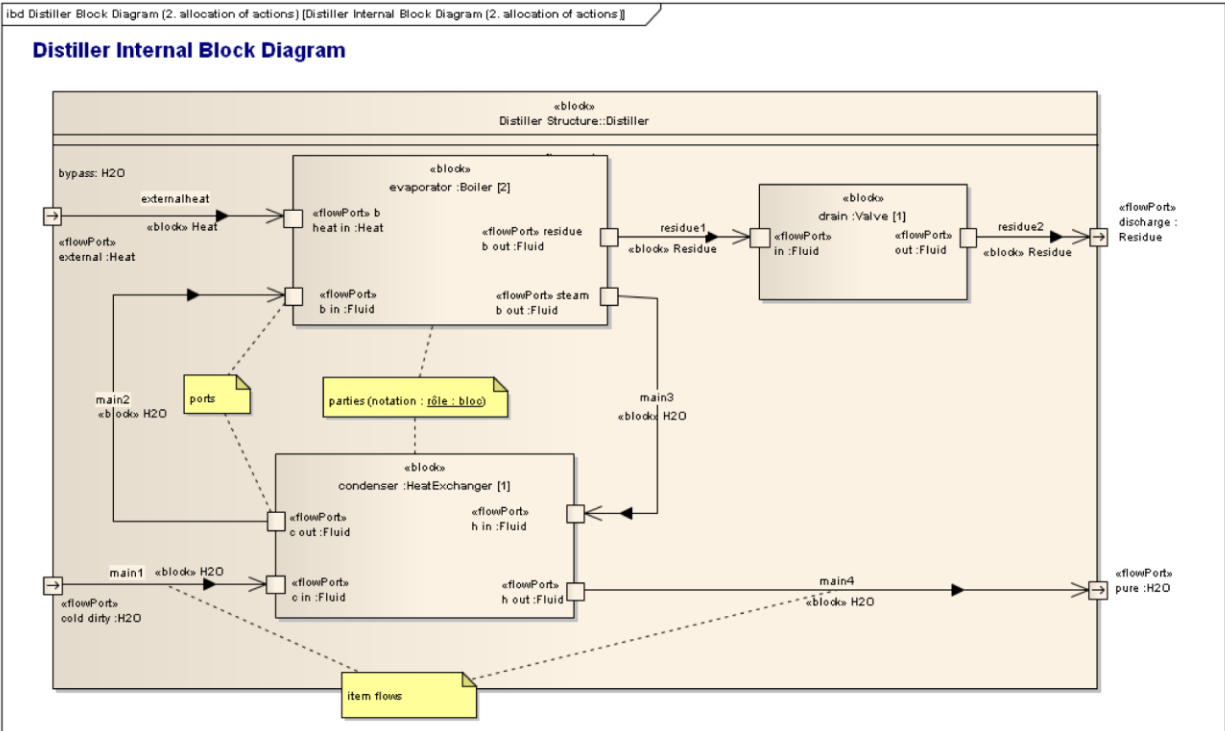


Diagrama internă a blocului descrie structură internă, în termeni de proprietăți. Este „cutia albă” a blocului, și arată cum sunt legate componentele interne la interfețele externe și între ele.



Porturile SysML sunt puncte de interacțiune ale componentelor. Porturile pot fi standard : specifică un set de operații, iar porturile de flux : specifică fluxul din sau în bloc.

Fiecare bloc are relații de constrângere cu proprietățile sale interne. SysML introduce o soluție pentru a vizualiza aceste constrângeri, în formă unei noi diagrame, numită diagrama Parametrică. Aceste diagrame se folosesc pentru a arată cum o schimbare a unei proprietăți structurale a unui sistem poate afecta valorile celorlalți parametri. Sunt folosite pentru a specifică constrângeri de modele/arhitecturi de sisteme.⁷

Diagramele prezentate mai sus fac parte din categoria diagramelor structurale ale SysML. De asemenea, există și diagrame comportamentale, care arată în ce mod se comportă anumite sisteme.

Activitatea unui sistem reprezintă activitatea transformării intrărilor în ieșiri, printr-o secvență controlată de acțiuni. Descrierea unui scenariu pentru un sistem este data de diagrama de activitate. Ca și în UML, acesta diagramă conține elemente sugestive: noduri inițiale, finale, de reuniune, de decizie, etc.⁸

⁷ T. Weilkiens, Systems Engineering with SysML/UML: Modeling, Analysis, Design, 2008, OMG Press

⁸ <http://www.sysmlforum.com/sysml-faq/>

Unul din cele mai importante lucruri în proiectarea unui sistem este atingerea scopului definit și respectarea cerințelor impuse. Este necesară o diagrama care să specifice și să analizeze cerințele unui astfel de sistem. Se introduce diagrama de necesități, diagrama specifică, într-o structură ierarhică. Relațiile de necesitate se aplică între blocuri și sunt de tipul: copy, verify, satisfy, trace, etc.⁹

Pentru a putea fi îndeplinite, cerințele trebuie validate. Pentru a valida cerințele se caută calități ce sunt legate de corectitudine, claritate, consistență, necesitate, prioritate, fezabilitate etc.

Maparea unui element la altul se face prin alocări ce sunt relații generale. Alocările pot fi comportamentale, structurale, software la hardware, etc.¹⁰

BUCULEI Theodor Cristian, 441A

3. Tranzitia la SysML

Unul dintre scopurile SysML a fost să aibe asemănarea cu UML în proporție de aproximativ 80%. Acest procent a fost atins, și chiar depășit. O mare parte din UML2 este refolosită în SysML, în timp ce multe alte părți sunt omise. SysML adaugă câteva idei proprii.

Motivele principale pentru care este justificată tranziția de la UML la SysML sunt legate de îmbunătățirea descrierii diagramelor, folosirea blocurilor în locul claselor și introducerea diagramelor requirements și parametric.¹¹

Dezavantajul UML este neutilizarea noțiunii de "requirement". SysML a adăugat acest element și un nou tip de diagramă pentru a putea reprezenta necesitățile și relațiile taxonomice între ele.

⁹ <http://www.visual-paradigm.com/product/vpuml/features/sysml.jsp>

¹⁰ A Practical Guide to SysML, Second Edition: The Systems Modeling Language (The MK/OMG Press)

¹¹ <http://www.sysmlforum.com/sysml-faq/>

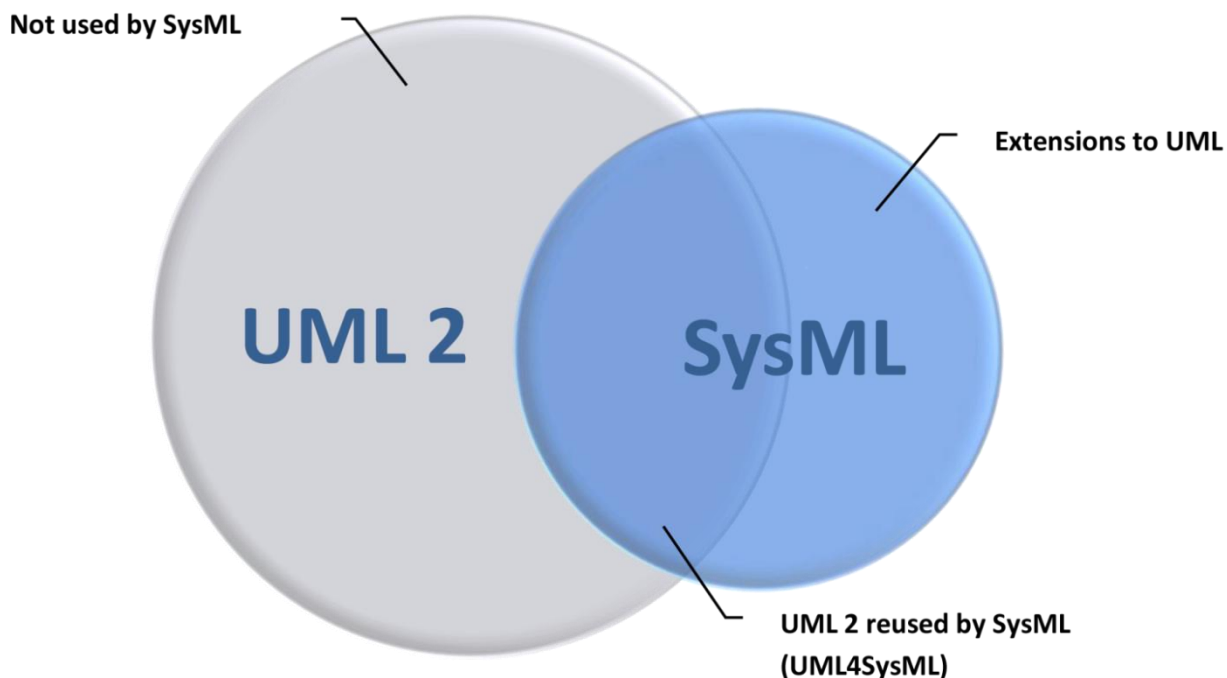


Figura 3: Relații între UML and SysML (Sursa 8 din bibliografie)

Cea mai importantă extensie a UML-ului, pe care o putem găsi și la SysML, este capacitatea de a modela comportamentul continuu. Diagramele de activități din UML2 se bazează pe execuția token-urilor. O activitate se va efectua atunci când are un token care îi permite să ruleze. O activitate are un token de execuție atunci când există date la toți pinii de intrare. Pinii de intrare corespund parametrilor într-o funcție sau procedură. Când o activitate se termină, pune toate datele pe pinii de ieșire. Aceștia corespund cu parametrii de ieșire ai funcțiilor sau procedurilor. Acest comportament este discret.

În SysML, diagramele de activitate sunt extinse pentru a suporta comportamentul continuu, prin adăugarea de obligații fluxurilor dintre activități. Aceste fluxuri pot fi discrete, de stream sau de control. Fluxurile de stream permit modelarea mișcării continue a materialului, fluxurile de control permit oprirea sau pornirea activităților, iar fluxurile discrete sunt fluxuri de diagrame de activitate standard UML.

Toate aceste aspecte reprezintă motive pentru care tranziția la SysML aduce avantaje în modelarea sistemelor.¹²

¹² <http://www.drdoobs.com/architecture-and-design/sysml-the-systems-modeling-language/192700757>

4. Modelare SysML ca parte a ingineriei software(exemple si aplicatii)

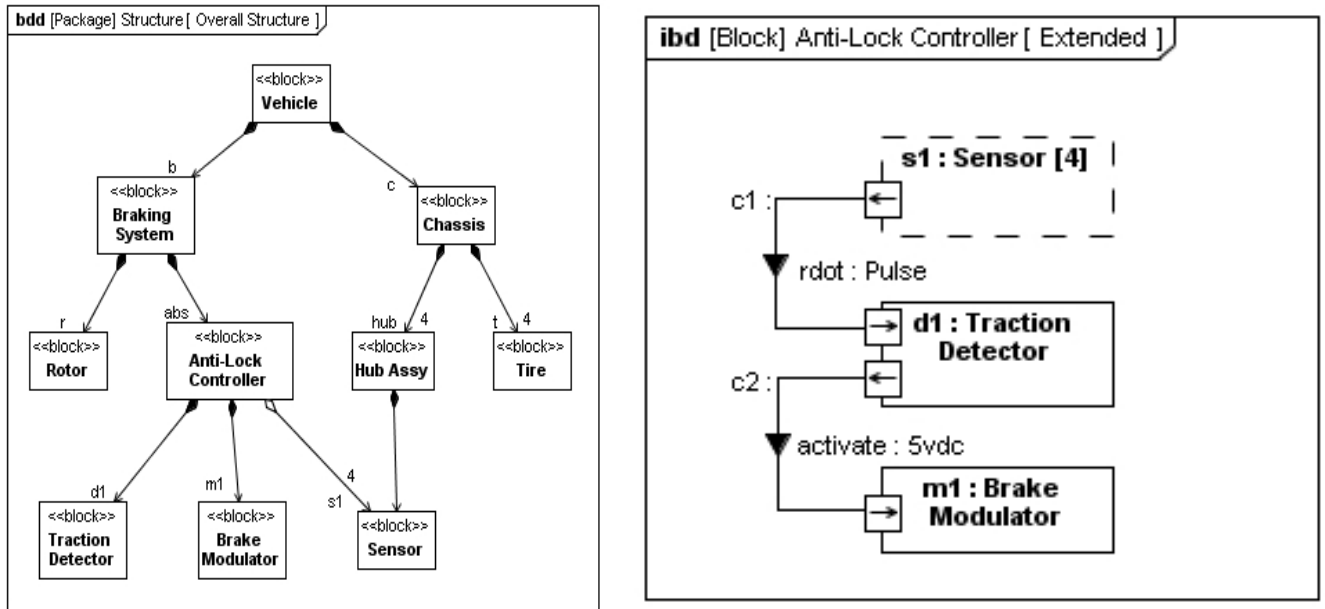


Fig. 4: Sursa 8 din bibliografie

Diagramele din figură sunt exemplificative pentru domeniul de electronică auto. Prima diagrama este diagrama bdd (block definition diagram), ce descrie structura sistemului mașinii și funcționalitatea lui. Senzorii oferă informații despre starea fiecărei roți a mașinii. În caz de blocaj, trebuie să anunțe controller-ul care va limita forța de frânare. Cea de-a doua diagramă este diagrama ibd (internal block diagram), care arată conținutul blocului, adică a controller-ului.

Următoarea diagrama este Parametrică, care descrie dinamică mașinii la mersul în linie dreaptă. Fiecare bloc reprezintă o ecuație și are o serie de parametri. Prin schimbarea lor, se modifică mărimea descrisă de ecuație. Se observă cum schimbarea poziției modifică succesiv ecuația vitezei, a accelerației și a forței de frânare.

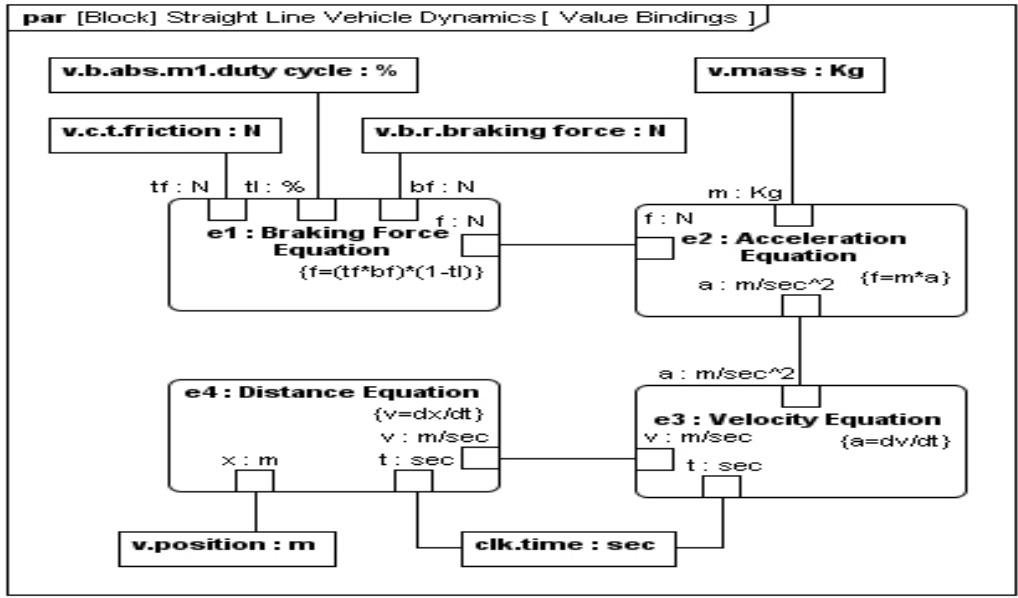


Fig. 5: Sursa 8 din bibliografie

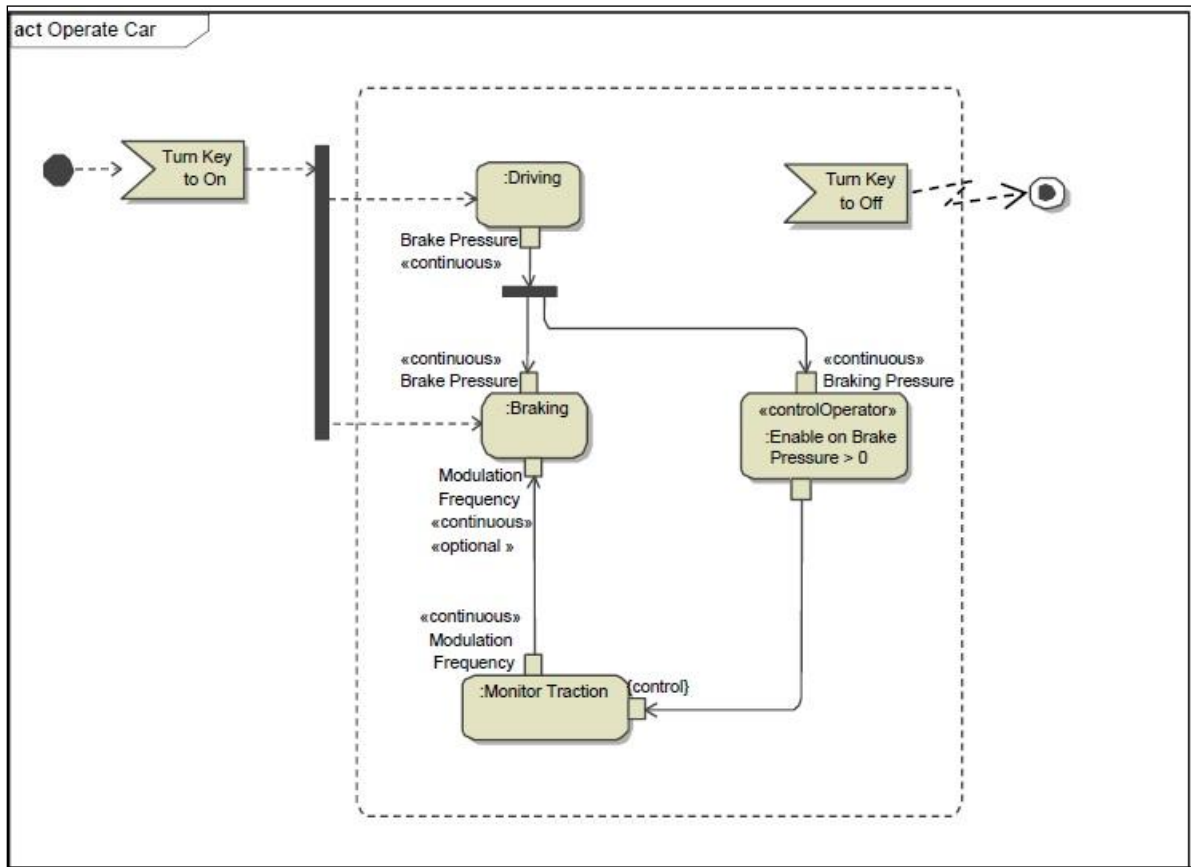


Fig. 6: Sursa 8 din bibliografie

Diagrama de activitate de mai sus descrie procesul de operare a unui automobil. Se observă folosirea structurilor de tip join, care arată că după pornirea mașinii, procesele de conducere și frânare se pot desfășura simultan. Așadar, apăsarea pedalei de frână determină o modulare de frecvență, schimbătoare în funcție de presiunea cu care se acționează frână. Astfel, mașină se oprește brusc sau treptat. Introducerea și scoaterea cheii din contact reprezintă punctele de start și stop din diagrama de activitate, iar blocurile reprezintă acțiunile.

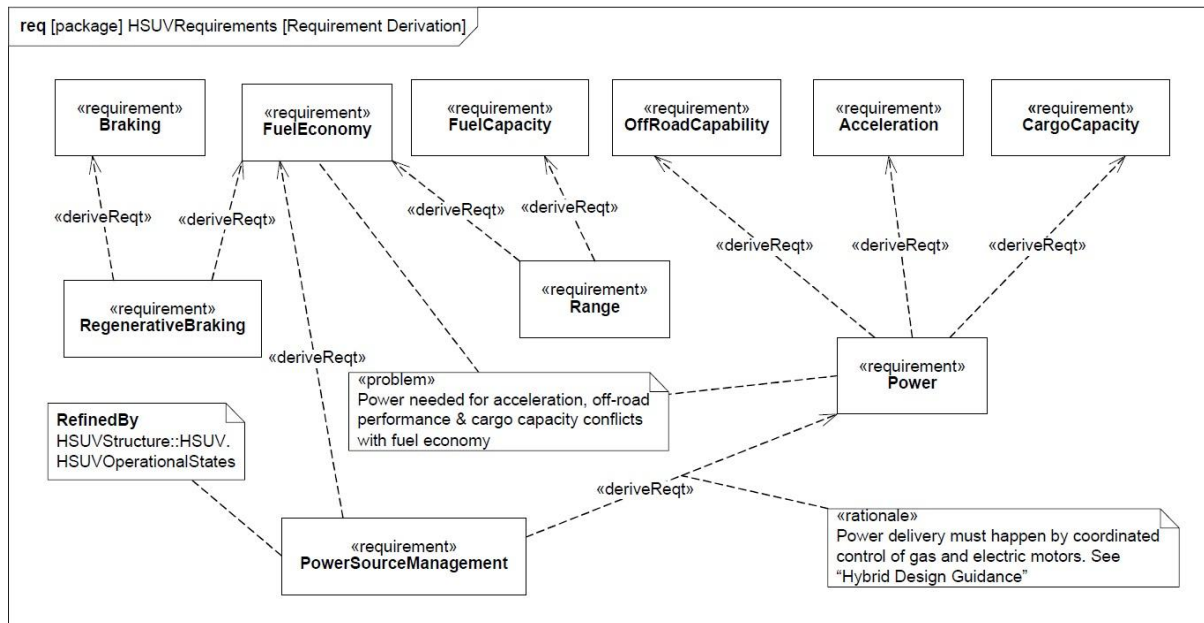


Fig. 7: Sursa 8 din bibliografie

Exemplul de mai sus ilustrează o diagrama de necesitate, care arată cerințele pentru consum mic și performanță mare. Accelerarea puternică și obținerea performanței duce la un consum ridicat de putere. Pe de altă parte, economia de benzină se realizează prin cerințe mici de putere. Astfel cele două situații sunt conflictuale. Se observă mesajele de cerere de tip derive request, mesaje ce reprezintă necesități ce sunt transformate din necesități de nivel superior.

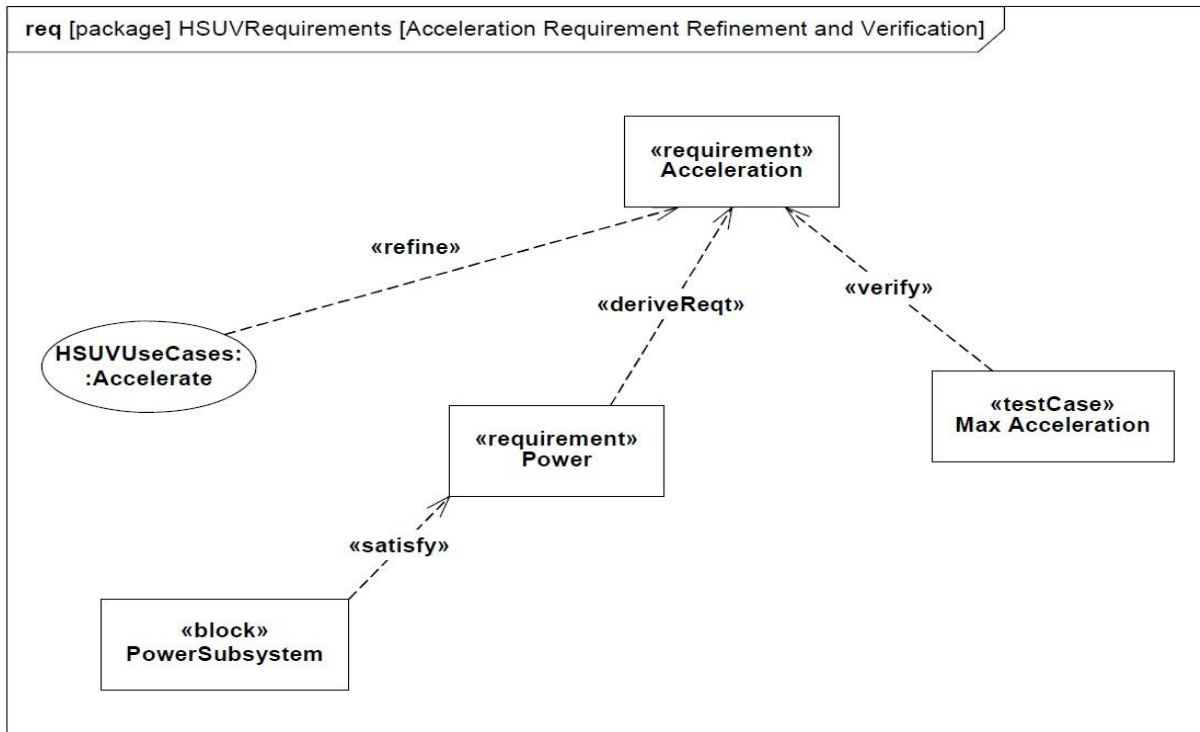


Fig. 8: Sursa 8 din bibliografie

Diagrama detaliază necesitățile pentru accelerația mașinii. Se poate observa și mesaje de tipul verify, satisfy și refine.

Diagrama parametrică

Diagrama parametrică ajută la analiza sistemului (performanță, fiabilitate etc.) prin definirea unor blocuri de constrângeri.

Blocul de constrângeri conține o ecuație matematică și parametrii săi, dintre care unii pot corespunde unor proprietăți ale blocurilor din sistem.

Constrângerile de bloc sunt definite într-o diagrama de clasă. După crearea acestei clase, poate fi creată o diagramă parametrică:

- Constrângerile sunt definite drept proprietăți și moștenesc proprietățile din constrângerea de bloc
- Proprietățile sistemului pot și adăugate și pot fi asociate cu proprietățile blocului
- Conectorii sunt folosiți pentru a uni toate proprietățile sistemului și proprietățile constrângerilor

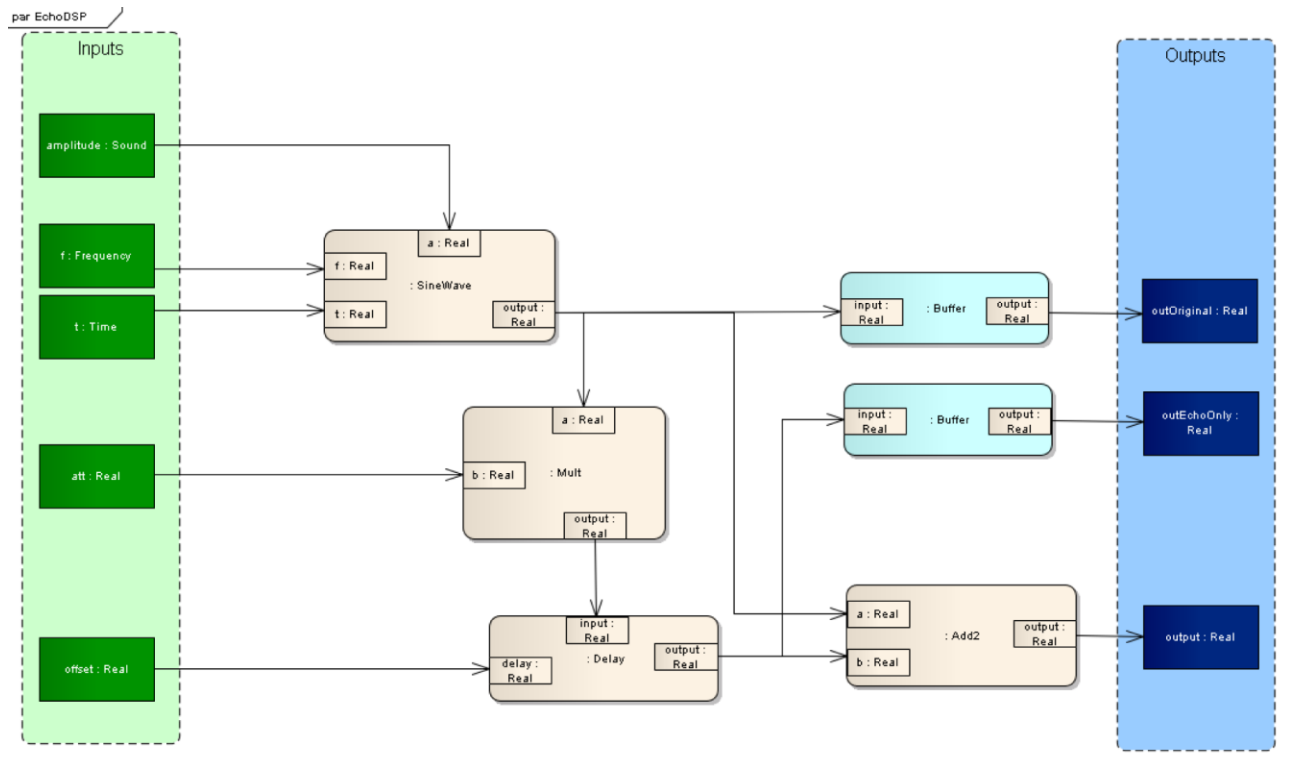


Fig. 9: Diagrama parametrica a mp3 player-ului lansata pe piata de Sparx Systems

Diagrama conține șase proprietăți de constrângeri de tipul SineWave, Mult, Delay, Add2 și Buffer. Parametrii de sistem sunt împărțiți între intrări și ieșiri și sunt asociați cu parametri proprietate ai constrângerii. Unii dintre acești parametri de constrângere sunt legați între ei.

Limbajul SysML are o gamă mare de aplicații, domeniul auto fiind doar unul dintre ele. Scopul modelării sistemelor auto este obținerea unei mai bune înțelegeri a subsistemelor și funcționalităților sale.

BUCULEI Theodor Cristian, 441A

5. Portabilitatea SysML-ului

Limbajul SysML este bine suportat de majoritatea instrumentelor comerciale și open source utilizate în acest domeniu.

6. Concluzii

SysML este sponsorizat de INCOSE/OMG ce oferă un limbaj de modelare de uz general pentru care se oferă suport pentru specificarea, analiză, arhitectură și verificarea sistemelor complexe. Este un subset al UML2. Cei 4 stâlpi ai SysML oferă modelarea cerințelor, comportamentului, structurii și parametrilor. Se recomandă tranziția la SysML, ca parte apropiată de modelarea grafică a sistemelor, în principal pentru că îmbunătățește comunicațiile, interoperabilitatea instrumentelor și calitatea proiectării. SysML continuă să evolueze, evoluție bazată pe feedback de la clienți și avansare a tehnologiei.

Versiunea de SysML actuală este 1.3. Această nu aduce îmbunătățiri considerabile față de 1.2, însă 1.2 aduce câteva schimbări importante: sincronizare cu schimbările din UML2.3, model de porturi conjugate și notații, asocierea de nume regiunilor de activitate neîntreruptă, includerea de specificații UML de instanță, de noduri de activitate structurate, îmbunătățiri pentru suportul de valori de tipul Unit și QuantityKind, și introducerea unui model pentru a defini sistemele de unități și cantități.

Deși SysML este practic o variantă îmbunătățită a UML, acestea se pot folosi simultan într-un proiect, în care unii ingineri de software folosesc SysML pentru analiză de sistem și de cerință, iar alții folosesc UML pentru proiectare. Problemele întâlnite în utilizarea acestei combinații sunt de diferență de limbaj și de translatarea diagramelor din UML în SysML. În cazul diagramelor de activitate, translatarea nu este o mare problemă, din moment ce folosesc aceleași convenții de nume, iar diferențele date de extensiile SysML sunt rezolvate de profilurile și stereotipurile UML.

Probleme de diferență de limbaj apar în cazul diagramelor block definition/internal block și cele class/internal structure/component. Acestea probleme pornesc de la folosirea de convenții de nume diferite, până la adăugarea constructorilor SysML cu semantica ambiguă.

Alte probleme de incompatibilitate apar și la diagramele parametrice. Introducerea unor paradigme de constrângeri logice, care nu sunt integrate total în paradigma obiect-componentă a UML-ului, produce dificultăți de înțelegere. De asemenea, relațiile de alocare ale SysML și relații trace/refine ale UML sunt ambigue, deci este frecvent neclar cum se mapează constructorii pt analiză de sistem SysML la constructorii UML.

Din cauza acestor probleme, echipele care plănuiesc să folosească SysML și UML într-un proiect ar trebui să fie conștiente de apariția acestora, atât subtile, cât și substanțiale de translație și trasabilitate și să ia măsuri suplimentare.

7. Bibliografie

1. Note de curs Cpnf. Dr. Ing. Stefan Stancescu:
<http://stst.elia.pub.ro/programa/is.htm>
2. A Practical Guide to SysML, Second Edition: The Systems Modeling Language (The MK/OMG Press)
3. T. Weilkiens, Systems Engineering with SysML/UML: Modeling, Analysis, Design, 2008, OMG Press
4. <http://www.omg.sysml.org/>
5. <http://www.visual-paradigm.com/product/vpuml/features/sysml.jsp>
6. <http://www.sysmlforum.com/sysml-faq/>
7. <http://www.modelbasedsystemsengineering.com/>
8. www.omg.org/ocsmg/Hsuv.pdf
9. <http://www.drdoobbs.com/architecture-and-design/sysml-the-systems-modeling-language/192700757>