

ROLUL DIAGRAMELOR SYMML ÎN PROIECTAREA MICROSATELITELOR

Nicolai LEVINETȚ, Valentin ILCO, Andrian GÎRȘCAN, Nicolae SECRIERU

Universitatea Tehnică a Moldovei

Abstract: *This paper presents the process of software design based on artificial modeling languages which involve some diagrams that will provide a function vision for each satellite system. At the initial stage there is a small loss of time needed to design diagrams using modeling language. The need of design these diagrams can be explained by: high reliability of the software based on the modeled diagrams, fast software design and maintenance, compact code, because will be created a system that will work in real time without execution unnecessary syntax, high complexity of programs achieved and in the least, generation of technical documentation and software diagrams. Automatically generated documentation will offer the possibility of software maintenance and also will make possible to detect errors resulting from the tests. The article describes the main diagrams used in the design of a satellite module. The initial phase will give a good start to achieve software used in operation of the satellite hardware.*

Cuvinte cheie: *SysML, satelit, diagrame, UML*

1. Introducere

Obstacolele întâlnite la proiectarea “sistemelor înglobate” constau în proiectarea structurii sistemului, în modelarea scenariilor de utilizare, a interacțiunilor dintre procesele ce formează un model complex și de aranjarea în timp a evenimentelor unui model.

Astfel totul se reduce la proiectarea hardului și softului unui sistem. Și dacă în cazul hardului se proiectează schema electrică, iar pe baza ei se elaborează cablajul imprimat, la elaborarea softului se creează algoritmi iar pe baza lor programul de funcționare. Deseori se primesc multe procese care pot apărea concomitent, sau care trebuie să urmeze strict unul după altul, în așa cazuri este dificil de alcătuit un soft cu o fiabilitate sporită din cauza haosului în program. Fiabilitatea nu este unicul minus în proiectarea software, deoarece programul nu este structurat și aranjat scade și rapiditatea lui. Pentru a reduce aceste dezavantaje tehnologia modernă a elaborat UML (Unified Modeling Language). UML reprezintă limbajul de destinație general al modelării vizuale, care este elaborat pentru specificarea, vizualizarea, construirea și documentarea componentelor produsului soft, business-proceselor și altor sisteme. Necesitatea unei diagrame UML este simplu de justificat dacă este nevoie de elaborat o aplicație de dimensiuni mari. Aplicațiile mici și medii pot fi realizate fără a schița inițial un model, deoarece sunt elaborate de echipe mici, au o complexitate redusă și pot fi înțelese relativ ușor doar prin consultarea codului sursă și eventual a documentației. Însă, pentru o aplicație mare, cu zeci de module, pachete și clase acest lucru nu mai este valabil. O aplicație sau un sistem de dimensiuni mari conține interdependențe între pachete și clase, ierarhii de obiecte complexe și foarte probabil zeci de mii de linii de cod. Astfel, pentru o simplă modificare trebuie parcurse clasele și pachetele pentru a observa unde există implicații ale modificării anterioare. Pentru a evita aceste situații se folosesc diagramele UML care prezintă arhitectura aplicației, făcând-o astfel ușor de înțeles și de urmărit.

SysML(The Systems Modeling Language) este bazat pe UML și implică modelarea blocurilor în loc de modelarea claselor, și furnizează un vocabular mai potrivit pentru ingineria sistemelor. Un bloc cuprinde software, hardware, date, procese, personal și facilități. Diagramele SysML re folosesc un subset al UML2 și își definesc propriile extensii. Prin urmare SysML include noua diagrame în loc de treisprezece, făcându-l astfel un limbaj mai ușor de învățat și aplicat. SysML face posibilă generarea specificațiilor într-un limbaj unic pentru echipele eterogene, ocupându-se cu realizarea hardware-ului sistemului și a blocurilor software.

2. Implementarea sysml la proiectarea satelitelui SATUM

Elaborarea satelitelui se află în concordanță cu prioritățile naționale de: monitorizare/prognosticare a alunecărilor de teren; monitorizare/prognosticare a formării și migrației norilor cu grindină; monitorizarea plantațiilor și terenurilor agricole; monitorizarea ecologică a masivelor de pădure și evoluției acvatice în râuri și lacuri; prestarea diverselor servicii hidrologice; obținerea informației foto și pentru alte scopuri, etc. În limitele acestui program este propusă elaborarea unui aparat cosmic, numit SATUM (Satelit Universitar Moldova), care se referă la clasa sateliților de orbita inferioară (înălțimea orbitei 500-600 km).

Proiectul propus prevede elaborarea următoarelor sisteme: sistemul de comunicație radio; Sistemul de captare a imaginilor; Sistemul de poziționare; Sistemul de comandă; Sistemul de alimentare. Sistemul de comanda sau calculatorul de bord al satelitelui este nucleul oricărui satelit. Arhitectura calculatorului de bord este strict dependentă de misiunile satelitelui, capabil să rezolve operativ o serie de probleme, dintre care:

- asigurarea comunicației între satelit și stația terestră;
- controlul parametrilor telemetrici ai satelitelui;
- controlul și reglarea alimentării cu energie electrică a subsistemelor satelitelui;
- controlul atitudinii (stabilizării și orientării satelitelui în spațiu);
- controlul sarcinii utile (a subsistemului de captare a imaginilor).

Pentru a observa modelele sistemelor a fost elaborata diagrama cazurilor(fig.1), această diagramă oferă o viziune simplificată, dar anume ea va permite o proiectare mai avansată. Diagrama cazurilor este cea mai simplă diagramă posibilă care descrie natura obiectelor ce nu se afla în sistemul modelat, prezentat în formă de “actors”, și procesele ce se execută. În calitate de “actors” va fi satelitul ca un corp material întreg care execută anumite funcții, de comunicare cu stația terestră, de captare a imaginilor, de poziționare prin intermediul sistemului ADCS și de alimentare a modulelor satelitelui cu energie electrică.

Astfel fiecare sistem poate fi descris aparte prin diagrame, iar toate diagramele și metodele de interacțiune a sistemelor între ele vor forma o rețea de modele a satelitelui. După realizarea schemei de structură a satelitelui se determină interfețele de comunicare dintre blocurile sistemului, și se stabilește protocolul de comunicare. Mai apoi, după ce scheletul a fost proiectat are loc descrierea tuturor evenimentelor și cercetarea tuturor cazurilor posibile de comportament al satelitelui, iar pe baza lor se pot elabora diagramele de activitate și de secvență. În baza algoritmilor se proiectează softul pentru toate blocurile satelitelui, softul structurat se generează din diagramele proiectate, el conține variabilele, bibliotecile, și condițiile descrise în diagrame. Mediul de proiectare IBM Rational Rose poate face conexiunea cu mediul de proiectare a softului Eclipse, la schimbarea softului în mediul Eclipse se modifica diagramele din IBMRR. Iar la finisarea proiectării IBMRR generează documentația tehnică a proiectului.

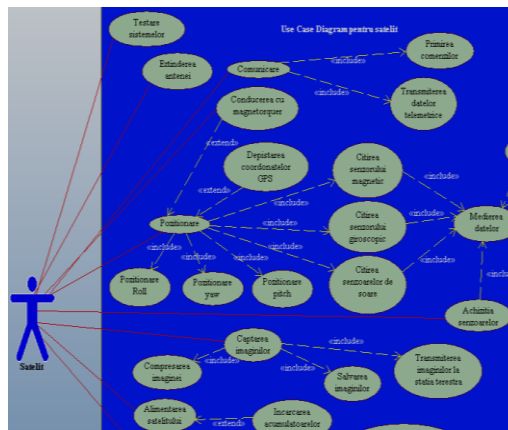


Figura1. Use Case Diagram de funcționare a satelitelui

3. Exemplu de proiectare a unității de comandă cu alimentarea satelitelui

Datorită condițiilor în care funcționează microsatelitul, necesitatea controlului alimentării crește în raport cu numărul de sarcini. Având orbită de nivel jos, satelitul doar parțial este alimentat de baterii solare, restul timpului este alimentat de acumuloare. Astfel o funcție este controlul încărcării acumuloarelor cu ajutorul panourilor solare, dar funcția principală este asigurarea alimentării stabile cu tensiuni nominale necesare sistemelor din satelit. Pentru aceasta este nevoie de convertoare specializate, conduse de un microcontroler. Pentru acumuloare se va folosi un protector ce va comunica cu microcontrolerul prin interfața I2C. Iar prin interfața SPI va avea loc comunicarea cu modulul de conducere cu panourile solare. Microcontrolerul va acționa cu aceste procese, iar comenzile le va primi de la calculatorul de bord.

Pe baza proceselor, se creează blocuri ce vor prelua funcțiile: de protecție, convertire de tensiuni, transformarea tensiunii analogice în digitală, etc. Rețeaua de blocuri reprezintă schema de structură a modelului proiectat. Microcontrolerul va fi centrul ce va realiza comunicarea cu modelele periferice, cu senzorii de protecție sau cu calculatorul de bord. El va citi nivelele de tensiune pe canalele ADC, va filtra datele și în dependență de datele primite va lua anumite decizii. Aceste evenimente se pot structura în una din diagramele de proiectare: collaboration, sequence, object diagram.

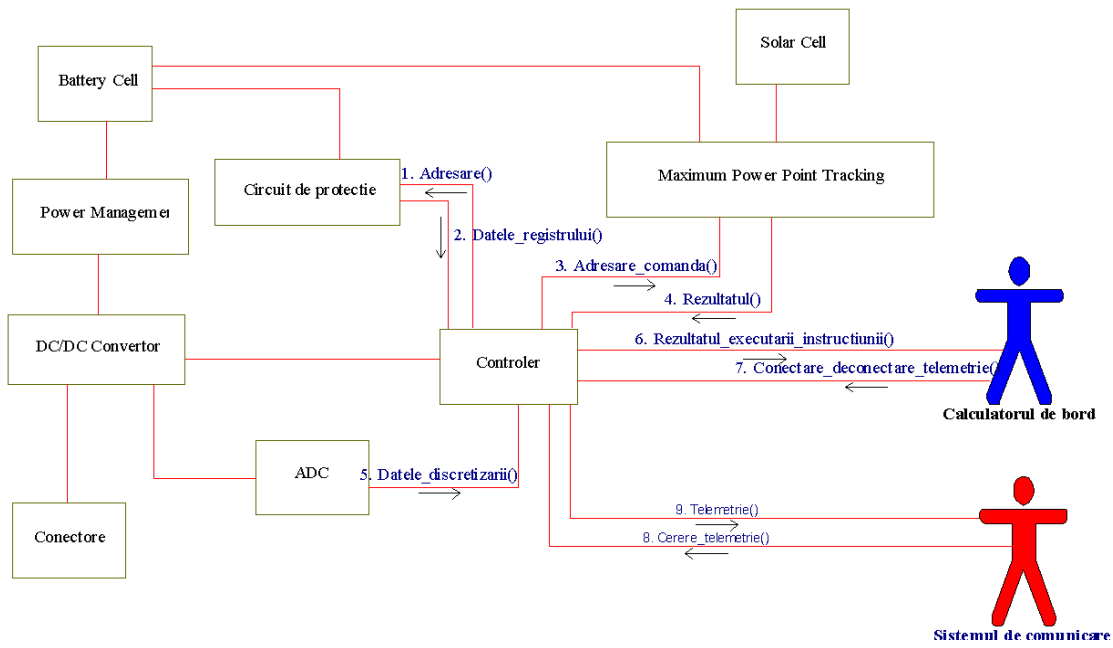


Figura 2. Collaboration Diagram de functionare a blocului de alimentare

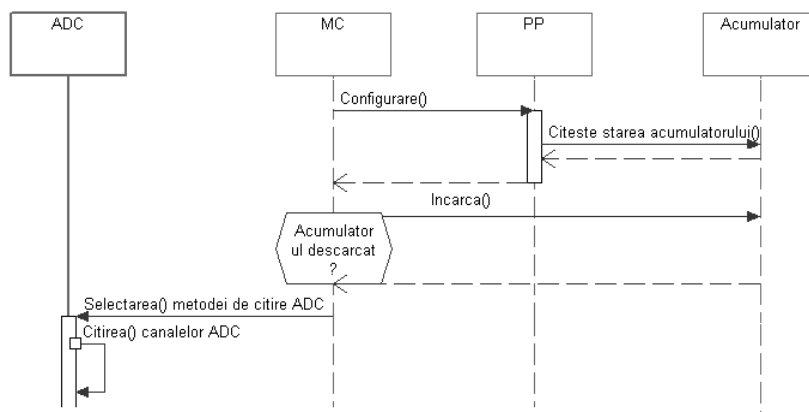


Figura 3. Sequence Diagram de functionare a blocului de alimentare

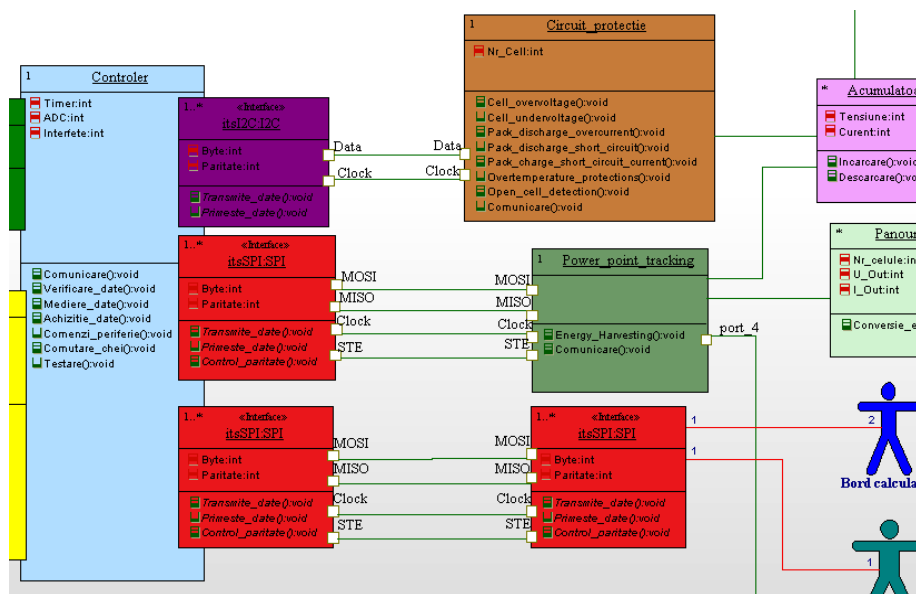


Figura 3. Object Diagram de functionare a blocului de alimentare

4. Concluzie

Limbajul UML este un mijloc de modelare simplu dar puternic, care poate fi utilizat efectiv pentru construirea modelelor conceptuale, logice și grafice ale sistemelor complexe de diferită destinație.

Acest limbaj conține cele mai bune calități ale metodelor ingineriei de program care au fost utilizate cu succes pe parcursul ultimilor ani la modelarea sistemelor complexe. Limbajul de modelare nu este doar un set de simboluri grafice, fiecare din ele se bazează pe semantica respectivă, ce înseamnă că modelul creat de un elaborator poate fi uniform interpretat de altul și nu neapărat de alt om, în calitate de al doilea elaborator poate fi și un anumit mijloc instrumental. Unele trăsături ale sistemului sunt mai bine modelate ca textuale, alte – ca grafice. Numărul de diagrame oferă posibilitatea de proiectare a sistemelor complexe prin descrierea lor din toate punctele de vedere, fie prin diagrama cazurilor sau prin aranjarea funcțiilor ce necesita executate în timp, cu ajutorul diagramei de secvență.

La proiectare apare o problemă, din start se pierde timp pentru proiectarea diagramelor, dar acest timp se răscumpără la proiectarea softului. Un alt avantaj reprezintă generarea structurii programului pe baza diagramelor, după care doar se adaugă codul necesar și are loc trecerea la stadia de testare.

Datorită structurii bine aranjate a softului în final se va primi o fiabilitate înaltă a satelitelui proiectat, cu posibilitatea de modificare a diagramelor, și cu economisirea timpului în cazul când se cere de revenit la etapa precedentă de proiectare.

5. Bibliografie

1. Andreea Orza. *Sisteme informaționale pentru afaceri*. <http://www.infoec.ro/fisiere/Lucrare-7.pdf>
2. *Продукты семейства IBM Rational Rhapsody*. <http://www.interface.ru/home.asp?artId=31829>
3. *Introducere în UML*.
<http://inf.ucv.ro/~giurca/courses/CB3105/resources/Introducere%20in%20UML.pdf>
4. Шмоллер Дж. *Освой самостоятельно UML за 24 часа*. Москва. 2005.p 401.
5. Дж.Рамбо, М.Блаха. *UML 2.0. Объектно-ориентированное моделирование и разработка*. (2-е изд.). Санкт-Петербург. 2007. p 545.
6. *Telelogic Rhapsody User Guide*. IBM. p 1610.