

# Aspecte ale proiectării sistemelor embedded a microsateleților cu ajutorul diagramelor SysML

Levineț Nicolai, inginer la Centru Tehnologii Spațiale, *levinet.nicolai@gmail.com*

Gîrșcan Andrian, inginer la Centru Tehnologii Spațiale, *girsca@gmail.com*

Ilco Valentin, student anul 4 FIMET, UTM, *valentin.ilco@gmail.com*

**Abstract** — în lucrare se elucidează procesul de proiectare a softului pe baza limbajelor artificiale de modelare, care presupun elaborarea unor diagrame ce vor oferi o viziune de funcționare a fiecărui sistem din satelit. La etapa inițială are loc o pierdere de timp necesar pentru proiectarea diagramelor cu ajutorul limbajului de modelare. Necesitatea proiectării acestor diagrame se explica prin: fiabilitate înaltă a softului realizat pe baza diagramelor modelate, rapiditatea proiectării și întreținerii softului, se crează un cod compact datorită căruia sistemul va funcționa în regim real fără executarea sintaxelor inutile, complexitatea mare a programelor realizate, și nu în ultimul rând documentarea tehnică a diagramelor și a softului generat. Documentația generată în mod automat va oferi posibilitatea de întreținere a softului și depistarea erorilor apărute în urma testărilor. În articol sunt descrise principalele diagrame folosite la proiectarea modulelor unui satelit. Etapa inițială va da un start bun pentru realizarea softului utilizat la funcționarea sistemelor hardware a satelitului.

**Cuvinte cheie** — modelare, protocol, satelit, SysML, UML.

## I. INTRODUCERE

Obstacolele des 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 instrucțiuni. Fiabilitatea nu este unicul minus în proiectarea software, deoarece programul nu este structurat și aranjat scade rapiditatea lui, ceea ce este un dezavantaj enorm pentru sistemele ce funcționează în timp real.

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

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.

## II. CARACTERISTICA DIAGRAMELOR DE PROIECTARE A SISTEMELOR COMPLEXE

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 refolosesc 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 poate fi simplu înțeles de către comunitatea software datorită relației strânse cu UML2.

SysML face posibilă generarea specificațiilor într-un limbaj unic pentru echipele eterogene, ocupându-se cu realizarea hardware-ului sistemului și a blocurilor software. Informația este prin urmare stocată prin modele într-un repertoriu unic, îmbunătățind comunicațiile între echipe. Pe termen lung blocurile pot fi refolosite dacă specificațiile și modelele implementate sunt potrivite pentru proiecte viitoare.

SysML oferă semantică mai flexibilă și mai expresivă, reduce numărul tipurilor de diagrame și a construcțiilor, iar tabelele SysML permit: alocarea cerințelor; alocarea funcțională; alocarea structurală.

SysML definește următoarele diagrame:

- 1) Diagrame de structură:
  - a) Diagrama de definire a blocurilor (Block

Definition Diagram) înlocuiește diagrama claselor în UML2. Diagrama de definire a blocurilor furnizează o reprezentare “black box” a unui bloc al sistemului alături de ierarhia blocurilor componentelor, ea poate include blocuri de orice tip: software, hardware, etc. Această diagramă poate fi considerată un inventar al tuturor componentelor și cantităților lor. În comparație cu UML2, diagrama de definire a blocurilor din SysML redefiniște diagrama claselor prin înlocuirea claselor cu blocuri și introducerea de “flow ports”.

- b) Diagrama internă a blocurilor (Internal Block Diagram) specifică componentele și modul de funcționare intern al unui bloc și este de obicei construită pe baza Diagramei de definire a blocurilor pentru a reprezenta asamblarea finală a tuturor blocurilor și componentelor din blocul principal. În comparație cu UML2, Diagrama internă a blocurilor din SysML redefiniște structura componentelor diagramei, blocurile sunt create în diagrama internă ca piese. Piese sunt asamblate prin conectori, legându-i direct sau prin porturi.
  - c) Diagrama parametrică (Parametric Diagram) are ca scop sprijinirea analizei sistemului (performanță, fiabilitate etc.) prin definirea blocurilor de constrângeri. Un bloc de constrângeri conține o ecuație matematică și parametrii săi, dintre care unii pot corespunde unor proprietăți ale blocurilor din system.
  - d) Diagrama pachetelor (Package Diagram) sunt folosite pentru a reflecta organizarea de pachete și a elementelor prin organizarea diagramelor de caz și diagramei de clase construite pentru a reprezenta relații, fie fizice sau logice.
- 2) Diagrame dinamice:
- a) Diagramele de tip activitate au fost ușor modificate în SysML. Diagrama de activitate reprezintă pașii unui proces, de multe ori folosindu-se de „pini de ieșire sau de intrare” care corespund tipului de element. Dacă o acțiune sau activitate corespunde unei operații de bloc, se poate asigura ca tipurile intrării și ieșirii acestei operații sunt consecvente cu tipurile operației de bloc.
  - b) Diagramele tabel de stări sunt folosite ca și în cazul UML2, adică oferă o cale de definire a ciclului de viață a blocurilor, căruia trebuie să se supună toate componentele. Un ciclu de viață definește toate stările posibile ale unui bloc și evenimentele împreună cu condițiile ce definesc stările de tranziție. În realitate numai blocurile complexe sau importante din punctul de vedere al activității, au nevoie de o diagramă a stărilor.

Toate aceste diagrame se proiectează în softuri specializate și standardizate după regulile de elaborare a diagramelor.

Unul din instrumentarii ce oferă posibilitatea de proiectare SysML sunt programele din familia IBM Rational Rhapsody. Mediul este integrat pentru cerințele de modelare folosind diagramele standardizate UML și SysML și oferă o analiză de control și o analiză dinamică,

ceea ce duce la majorarea consistenței proiectării. În plus mediul oferă posibilitatea de compilare și de elaborare a documentației după șabloanele standardizate, astfel se reduce esențial timpul de proiectare.

### III. IMPLIMENTAREA SYSML LA PROIECTAREA SATELITULUI SATUM

Proiectul propus se află în concordanță cu prioritățile naționale de: monitorizare/proгноzare a alunecărilor de teren; monitorizare/proгноzare a formării și migrației norilor cu grindină; monitorizarea plantațiilor și terenurilor agricole; monitorizarea ecologică a masivelor de pădure și evoluției acvatică în râuri și lacuri; prestarea diverselor servicii hidrologice; obținerea informației foto-video 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 satelitului este nucleul oricărui satelit. Arhitectura calculatorului de bord este strict dependentă de misiunile satelitului, capabil să rezolve operativ o serie de probleme, dintre care:

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

Pentru a observa modelele sistemelor a fost elaborată diagrama cazurilor, 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 satelitului cu energie electrică.

Poziționarea satelitului se realizează cu ajutorul Attitude and Determination and Control System (ADACS), dar pentru a determina coordonatele de poziționare pe cele 3 axe are loc achiziția de date de la: senzorii solari, senzorii magneticii, GPS, și datele de la sistemul de poziționare după stele.

Iar pentru alimentarea satelitului se vor folosi acumulatele ce se vor încărca de la panourile solare. Astfel la realizarea diagramelor se va ține cont de starea acumulatelelor, și de sistemele ce necesită pornite după graficul stabilit.

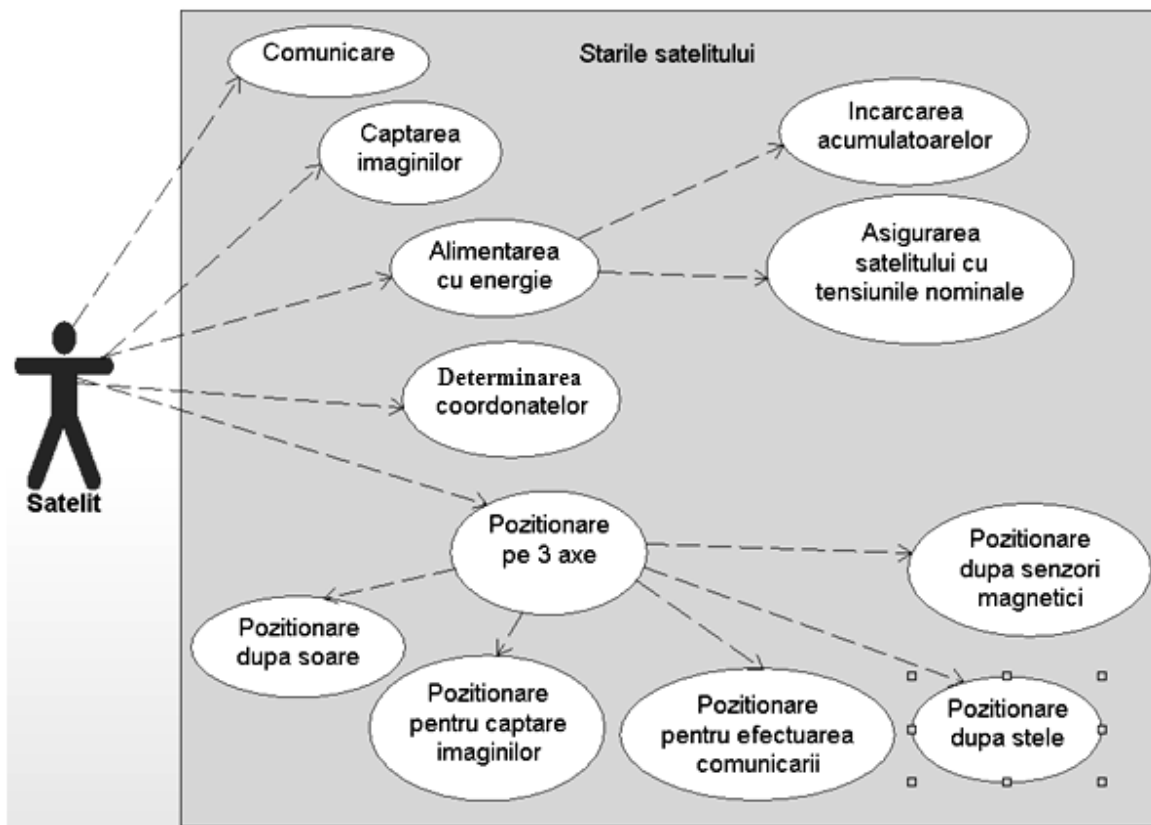


Fig.1. Diagrama cazurilor a calculatorului de bord

Modelarea vizuală în UML poate fi reprezentată ca un oarecare proces de lansare pe niveluri de la cel mai general și abstract model conceptual al sistemului inițial către model logic și mai apoi fizic, ce corespunde unui sistem de program. Pentru atingerea acestui scop de la început se creează un model în formă de diagrama cazurilor de

utilizare (use case diagram) care descrie destinația funcțională a sistemului sau cu alte cuvinte descrie ceea ce sistemul va executa în procesul său de funcționare.

Pe baza acestei diagrame se poate alcătui schema de structura.

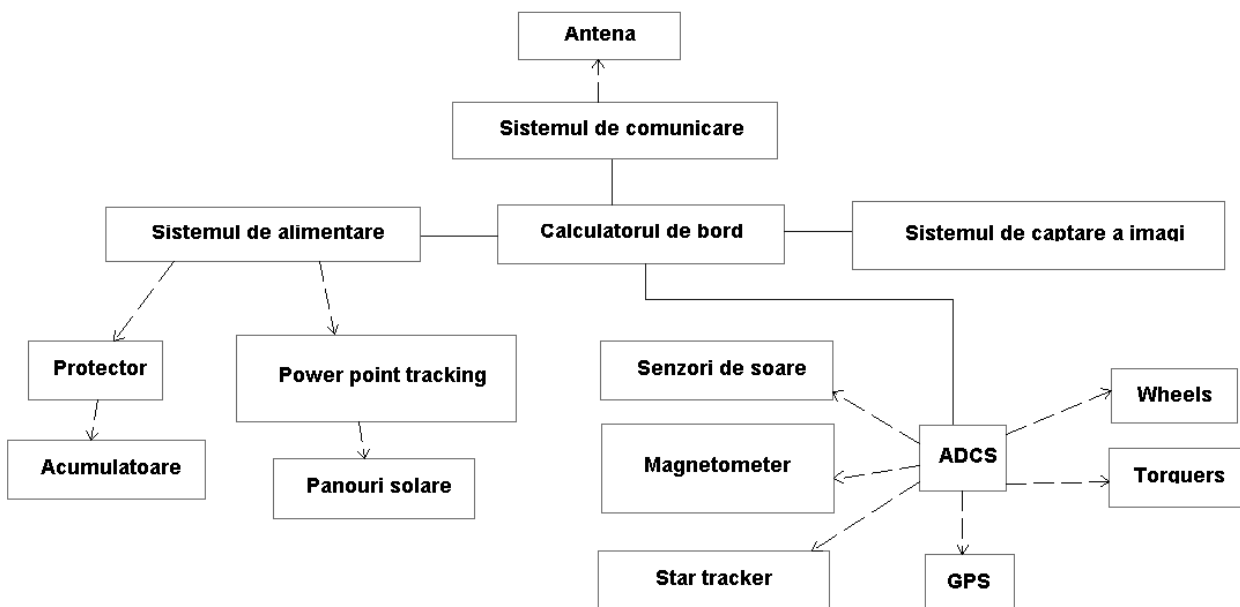


Fig.2. Schema de structură a satelitelui

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 modifică diagramele din IBMRR. Iar la finalizarea proiectării IBMRR generează documentația tehnică a proiectului.

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. Mai jos este reprezentată diagrama ce sugerează toate aceste procese de conducere.

IV. EXEMPLU DE PROIECTARE A UITĂȚII DE COMANDĂ CU ALIMENTARE SATELITULUI  
Datorită condițiilor în care funcționează microsatelitul, necesitatea controlului alimentării crește în raport cu

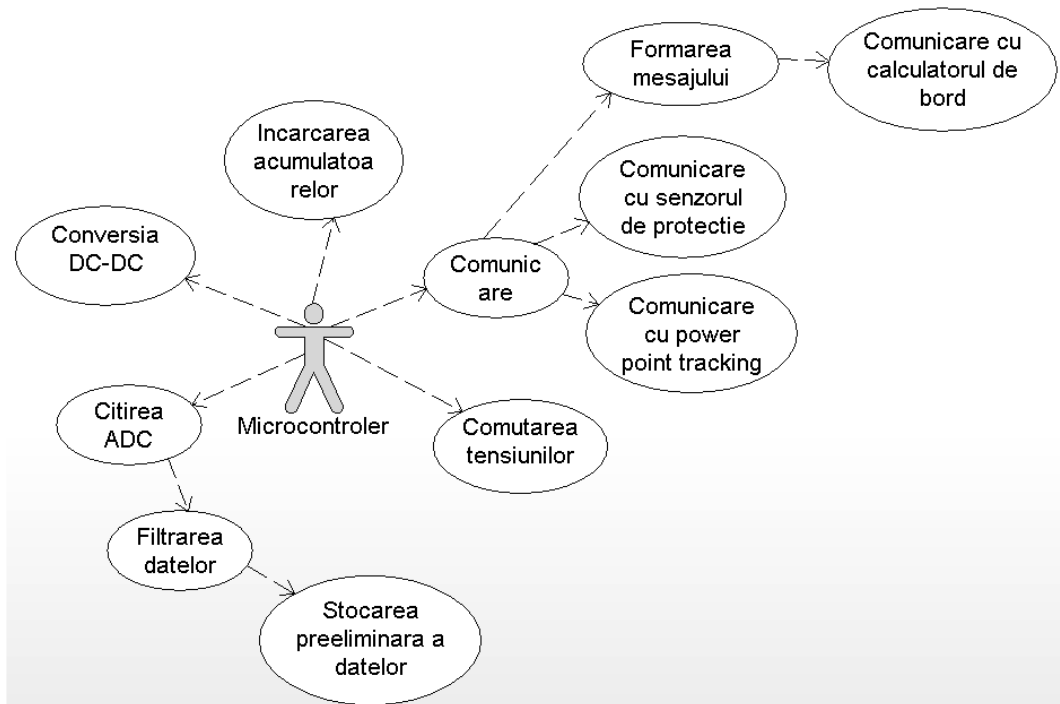


Fig.3. Diagrama cazurilor pentru sistemului de alimentare

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.

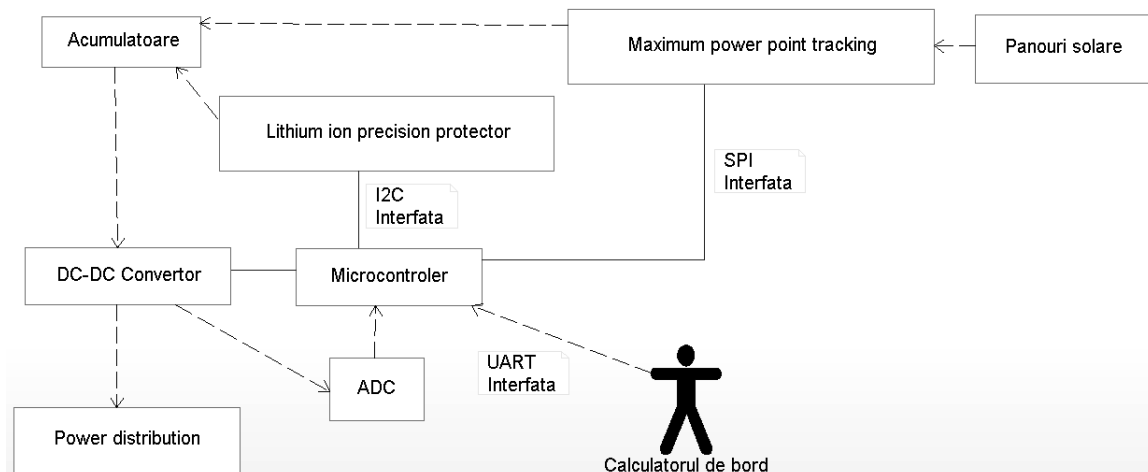


Fig.4. Schema de structură a sistemului de alimentare

În schema de structură a sistemului de alimentare, comunicarea cu periferia se realizează prin câteva interfețe, cu ajutorul unor protocoale. La interfețele I2C și SPI protocolul este dictat de documentația microcircuitelor, iar pentru comunicarea cu calculatorul de bord se va stabili un protocol cu o fiabilitate înalta de depistare a erorilor, cu

posibilitatea de adresare, și cu posibilitatea de majorare a rapidității comunicării datorită lungimii flotante a mesajului. Depistarea erorilor de comunicare se realizează cu ajutorul la CRC(Cyclic redundancy check), alcătuit din 16 biți.

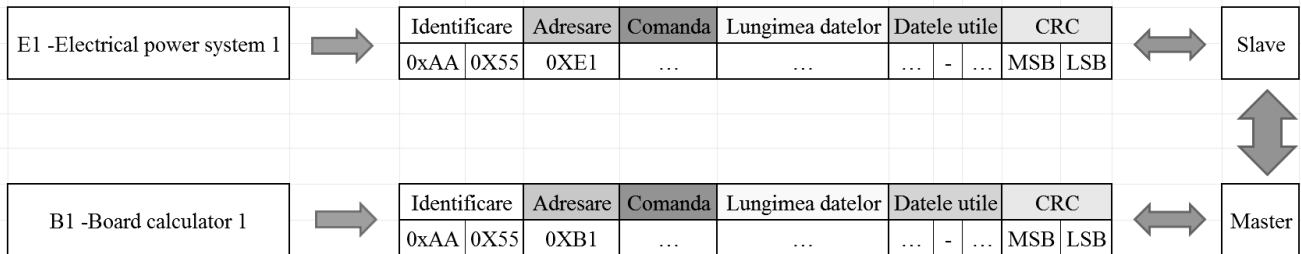


Fig.5. Protocolul de comunicare dintre calculatorul de bord și sistemul de alimentare

Cu ajutorul protocolului de comunicare se poate realiza dialogul dintre master și slave, pentru aceasta este nevoie de alcătuit grila de comenzi pentru calculatorul de bord și răspunsurile sistemului de alimentare. Această grilă se poate crea în dependență de sarcinile sistemului de alimentare, cu ajutorul diagramei de secvență.

Microcontrolerul execută comenzile după o ordine anumită, care este dictată de caietul de sarcini. Tot odată ordinea executării comenzilor poate fi prezentată pe o axă a timpului cu posibilitatea selectării proceselor în dependență de condițiile stabilite. Aceasta poate fi descrisă cu ajutorul diagramei de secvență.

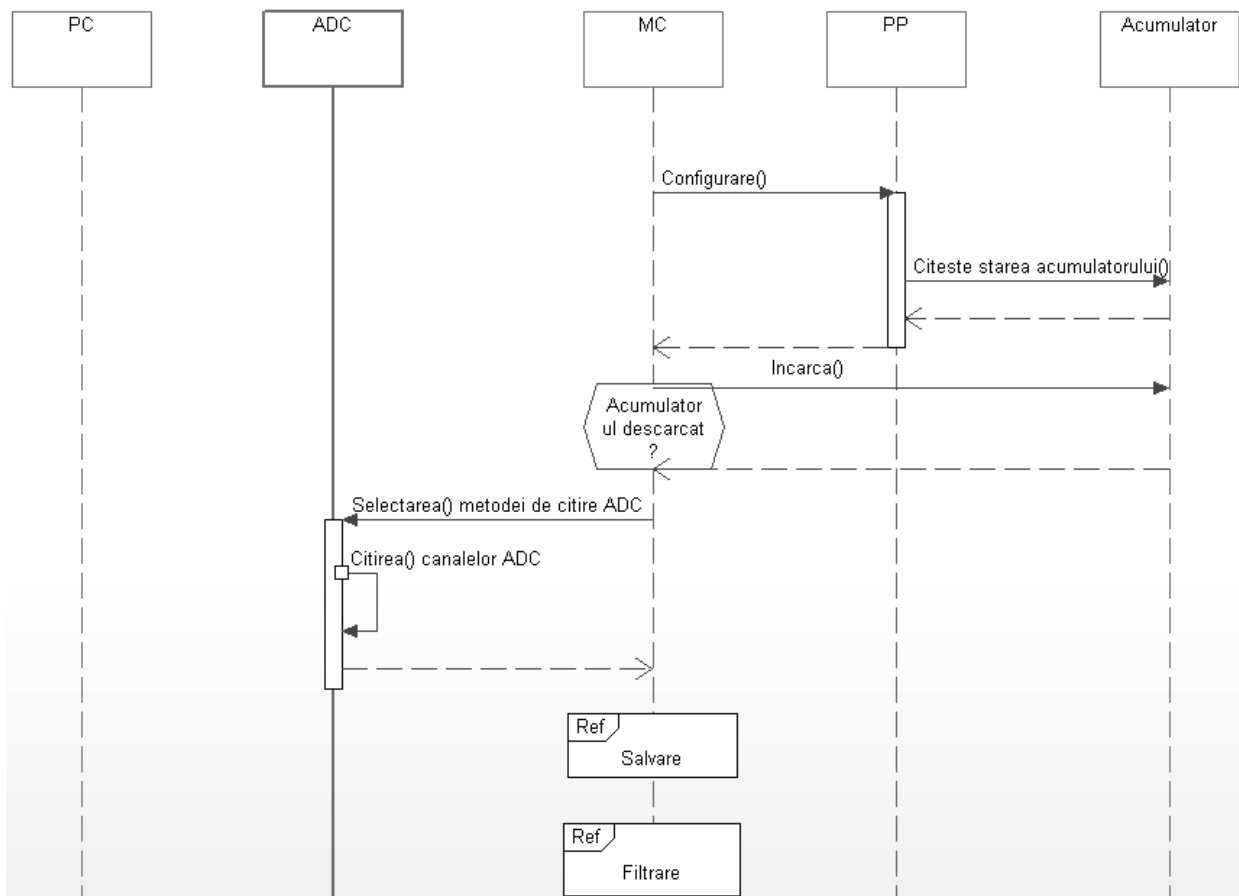


Fig.6. Fragment din diagrama de secvență a unității de alimentare

În diagrama de secvență se reprezintă procesul de încărcare a acumulatorului și de achiziție a datelor. După ce s-a controlat starea acumulatorului are loc procesul de citire a tensiunilor de pe canalele Convertorului Analog Digital (ADC), după care urmează stocarea și filtrarea preliminară a datelor. În fig.6 este reprezentat doar începutul diagramei,

avantajul acestei diagrame este posibilitatea de extindere fără întrerupere pînă la ultimul proces, formând o axa a evoluției proceselor în timp. Iar la necesitate diagramele se pot documenta în diferite formate fie pdf, doc, sau html cu posibilitatea de navigare între diagrame.

Sistemul de alimentare condus de calculatorul de bord, va recepționa datele la nivel fizic prin interfața UART cu ajutorul protocolului stabilit. La transmiterea datelor pot apărea câteva probleme: identificarea poate fi eronată, lungimea mesajului nu este constantă și se pot pierde

datele, CRC calculat nu corespunde cu CRC primit. Toate aceste stări pot fi depistate cu ajutorul unor condiții. "Activity diagram" permite prezentarea acestor condiții și implementarea lor în formă de program de prelucrare a datelor recepționate.

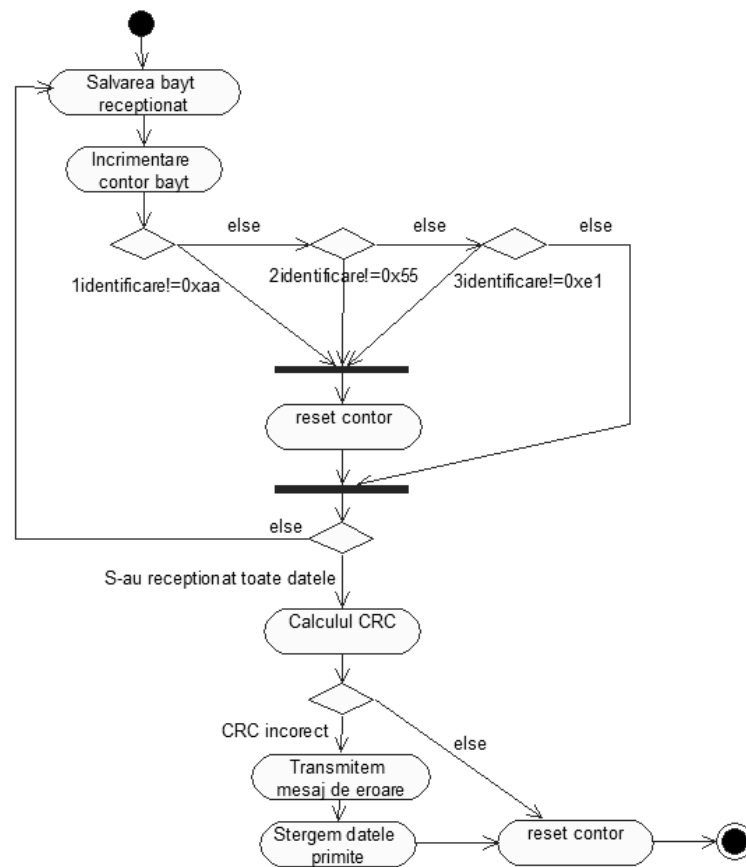


Fig.7. Prelucrarea mesajului recepționat prezentat sub forma diagramei de activitate

## 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 proiectarea 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 diagramei, dar acest timp se răscumpără la proiectarea softului. Un alt avantaj reprezintă generarea structurii programului pe baza diagramei, 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 diagramei, și cu economisirea timpului în cazul când se cere de revenit la etapa precedenta de proiectare.

## REFERENCES

- [1] Lenny Delligatti, SysML Distilled: A Brief Guide to the Systems Modeling Language, pag 301.
- [2] Andreea Orza, Sisteme informaționale pentru afaceri. Accesat 28.08.2014, disponibil: <http://www.infoec.ro/fisiere/Lucrare-7.pdf>
- [3] Uml. Accesat 05.09.2014, disponibil: <http://ru.scribd.com/doc/144420315/Uml>
- [4] Introducere în UML. Accesat 06.09.2014, disponibil: <http://inf.ucv.ro/~giurca/courses/CB3105/resources/Introducere%20in%20UML.pdf>
- [5] Продукты семейства IBM Rational Rhapsody. Data publicării: 22.01.2013. Accesat 07.09.2014, disponibil: <http://www.interface.ru/home.asp?artId=31829>