

Proiectarea Arhitecturii Software a Calculatorului de Bord a Microsatelitului SATUM cu Ajutorul Limbajului SYSML

Levineț N., Ilco V.

Centrul Tehnologii Spațiale
Universitatea Tehnică a Moldovei
Chișinău, Republica Moldova
levinet.nicolai@gmail.com, valentin.ilco@gmail.com

Abstract — This paper presents the experience of software architectural design process for the microsatellite board computer based on formal language SysML. The design is achieved by top down method, which aims to set events and processes for satellite systems work by drawing SysML diagrams. In the design environment thanks to diagrams is generated program structure, which provides reliability and high-speed operation. Acquired experience shown that due to the reverse design (reverse engineering) may be reduced time testing and the design environment will allow correction and automatic generation of electronic documentation.

Keywords — SYSML, board computer, diagrams, telemetry.

I. INTRODUCERE

Calculatorul de bord reprezintă un dispozitiv de conducere și control cu periferiile "slave" a unui sistem automatizat. Pentru un satelit, calculatorul de bord conduce cu: sistemul de poziționare, comunicare, alimentare, sarcina utilă (camera de fotografiat). La etapa inițială se elaborează arhitectura generală a softului. Din start se vor evidenția sarcinile și problemele ce pot apărea la etapa de elaborare a softului. Arhitectura soft poate fi proiectată cu ajutorul Unified Modeling Language (UML) - un limbaj standard pentru descrierea modelelor și specificațiilor soft. UML a fost la bază dezvoltat pentru reprezentarea complexității programelor orientate pe obiect, al căror fundament este structurarea programelor pe clase, și instanțe (numite și obiecte). Cu toate acestea, datorită eficienței și clarității în reprezentarea unor elemente abstracte, UML este utilizat dincolo de domeniul IT. Pentru elaborarea softului calculatorului de bord, se modelează diagramele ce reflectă cerințele și modul de structurare a subprogramelor pe axa timpului. Diagramele date vor reprezenta principiile de funcționare a softului pentru calculatorul de bord.

Un microcontroller va substitui calculatorul de bord și va realiza operațiile ce vor fi înscrise în memoria internă. La modelarea diagramelor de clasă și sequence diagram, mediul de proiectare va permite de creat în mod automat diagramele de activitate. Aceste diagrame vor fi modificate conform algoritmilor de funcționare. Diagramele de activitate și cele de clasă vor permite de generat scheletul programului. Programul va fi completat cu sintaxele corespunzătoare, iar în final se va

realiza proiectarea inversă, pe baza softului proiectat se vor restructuriza diagramele.

Documentația și diagramele prezentate în formă electronică vor permite de depistat localizarea erorii. După corectarea erorii, cu ajutorul mediului de proiectare a diagramelor SYSML va fi posibil de regenerat codul programului. Documentația tehnica a diagramelor proiectate va fi necesară la etapa de testare preliminară, și la etapa de validare.

II. SARCINILE CALCULATORULUI DE BORD

Calculatorul de bord este autopilotul ce conduce cu sistemele satelitului. Funcțiile principale a calculatorului este transmiterea comenzilor către dispozitivele periferice și controlul procesului de realizare a acestor comenzi.

Pentru încărcarea acumulatorilor este necesar de poziționat satelitul spre soare. Iar pentru captarea și transmiterea imaginii, are loc poziționarea antenelor în direcția stației terestre. În dependență de locația satelitului, calculatorul transmite coordonatele de poziționare către Attitude Determination and Control System (ADCS). Datele de poziționare sunt stocate într-un algoritm în memoria calculatorului de bord. Sistemul de captare se conectează doar când satelitul se află în zona de cartografiere. Calculatorul de bord conectează sistemul de captare și transmite comandă ca imaginea să fie transmisă spre stația terestră.

Ca datele să se recepționeze de la orice stație terestră, telemetria poate fi transmisă fără întrerupere. Acest proces este oprit de calculatorul în cazul când acumulatorii ating nivelul minim de energie stocată.

Astfel la etape stricte calculatorul de bord transmite comenzi și recepționează răspunsuri, după care analizează dacă sarcina a fost executată corect. În memoria calculatorului de bord este stocată grila de comenzi și răspunsuri pentru fiecare sistem. Pentru a transmite comenzile este necesar de stabilit un algoritm de selectare și transmitere a comenzii conform protocolului de comunicare. Următorul algoritm analizează corectitudinea executării comenzilor. Dacă apare o eroare, calculatorul încearcă să rezolve situația dată. Rezolvarea se limitează la căutarea și acționarea algoritmilor de funcționare a satelitului pentru cazurile excepționale.

Aceste erori împreună cu datele telemetrice colectate de la dispozitivele satelitelui se stochează în memorie de către calculatorul de bord.

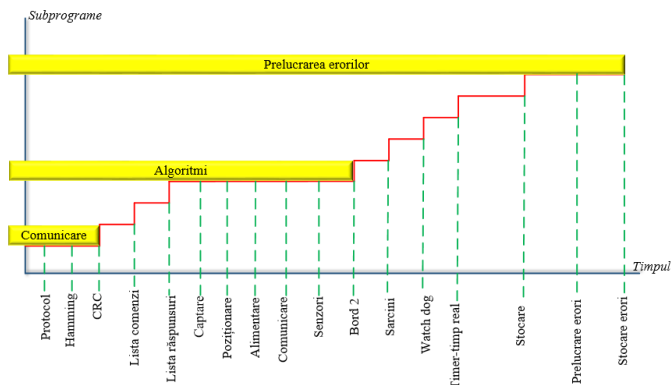


Fig. 1. Diagrama algoritmilor pentru calculatorul de bord.

În Figura 1 sunt prezentate sarcinile care sunt realizate de calculatorul de bord. La o proiectare arhitecturală, are loc o descriere a sarcinii în forma de diagramă. La un nivel mai înalt calculatorul de bord acționează aceste funcții la diferite etape în dependență de punctul de poziționare pe orbită.

III. ARHITECTURA SOFTWARE A CALCULATORULUI DE BORD

Pentru fiecare algoritm se elaborează diagramele SYSML, ce reflectă procesul de funcționare a fiecărei sarcini. În Figura 2 este prezentată diagrama ce arată procedeul de selectare și transmitere a comenzilor din grila de comenzi stocate.

Lista de comenzi se elaborează în baza documentației tehnice a fiecărui sistem microsatelitar. După un algoritm soft se realizează căutarea comenzii pentru fiecare sistem, după care se extrag datele într-un buffer temporar. Aceste date vor fi folosite la realizarea mesajului ce va fi transmis către sistem.

După același principiu poate fi modelată diagrama de “Răspunsuri” recepționate de la fiecare sistem. Aceste date se prelucrează conform algoritmilor proiectate în baza metodelor de funcționare a fiecărui sistem. Astfel diagramele date pot fi reprezentate prin intermediul altei diagrame, în cazul în care a avut loc încapsularea.

Telemetria se colectează de la sistemele periferice, după care se află media mărimilor achiziționate la intervale diferite de timp.

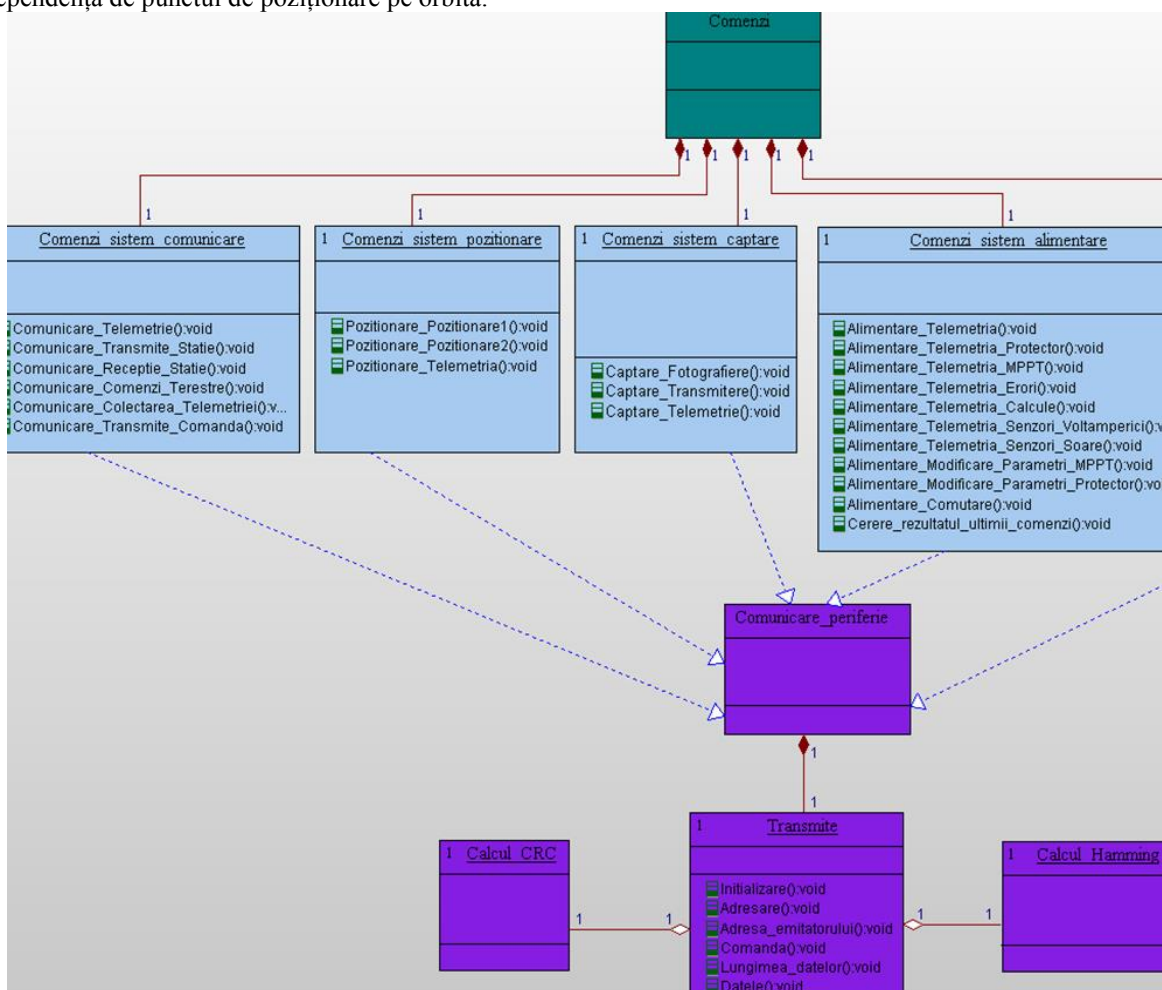


Fig. 2. Class diagram a comenzilor pentru sistemele microsatelitare.

În baza diagramelor de structură se proiectează diagramele de activitate și se elaborează programul de funcționare a calculatorului de bord.

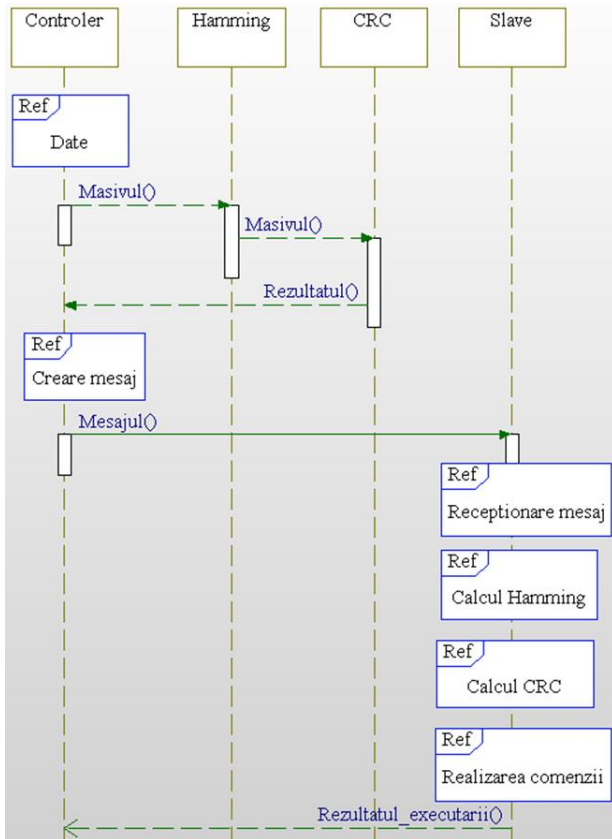


Fig. 3. Sequence diagram de comunicare a calculatorului de bord cu sistemele satelitului.

La etapa elaborării diagramelor de structură, nu se observa starea executării a unei sarcini în timp. Astfel următorul pas consta în elaborarea diagramelor de secvență și a diagramelor de stare.

Din diagramele de clasă s-a constatat că comunicarea, și algoritmi de funcționare a sistemelor vor fi cele mai apelate subprograme.

Comunicarea dintre calculatorul de bord și sistemele satelitului se realizează conform unui protocol stabilit. Datele transmise sau recepționate se transmit la algoritmul de calcul al codurilor corectoare de erori Hamming, și la algoritmul de calcul al sumei de control. Aceste date sunt încapsulate în mesajul final. Algoritmi returnează variabile ce descriu corectitudinea mesajului recepționat. Dacă sunt depistate erori, calculatorul transmite mesaj că a recepționat greșit mesajul, după care mesajul este retransmis de către sistemul „slave”.

În Figura 4 este reprezentat procedeul de control al datelor recepționate. Conform la diagrama de activitate s-a elaborat programul ce va fi înscris pe controlerul MSP430F5529. La primirea datelor prin interfața UART, byte se înregistrează în masiv unidimensional, și se incrementează contorul de date.

Dacă contorul depășește lungimea transmisă de date, datele se neglijează și contorul se resetează. Fanionul de recepție este setat după ce se verifică câteva condiții. Lungimea de date trebuie să fie egală cu cifra contorului. Inițializarea și adresarea trebuie să corespundă cu adresa sistemului. În caz că fanionul se setează în unitate, pachetul de date trece la etapa de verificare a corectitudinii de recepție a datelor.

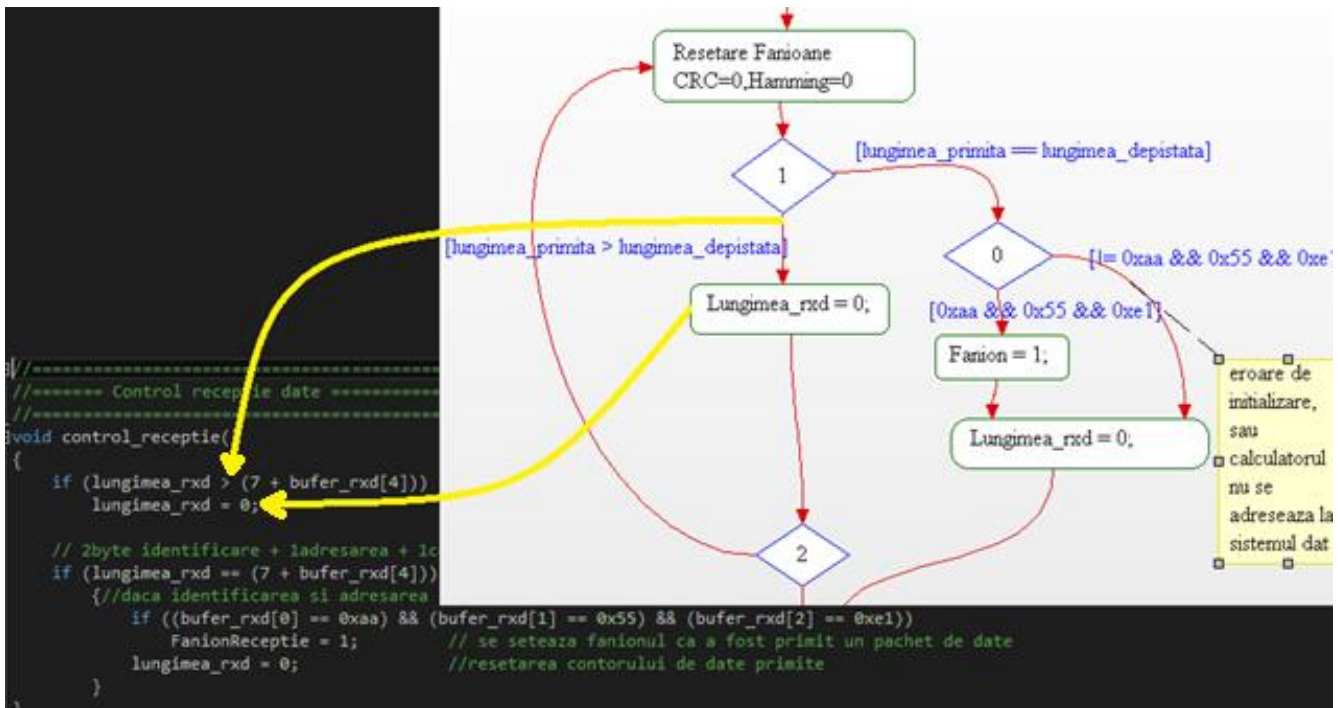


Fig. 4. Subprogramul de control a pachetelor de date la recepție.

Softul calculatorului de bord stochează telemetria și erorile colectate de la sistemele periferice. Statistica erorilor apărute este necesară pentru determinarea modului de funcționare a satelitelui în spațiu, și pentru corectarea parametrilor de funcționare a satelitelui.

Corectarea parametrilor se realizează prin schimbarea variabilelor din programele de funcționare a fiecărui bloc. Erorile pot apărea la comunicare, la apariția restului de la câinele de pază, sau erori de funcționare a fiecărui modul.

Telemetria se colectează de la sistemele periferice, după care se afla media mărimilor achiziționate la intervale diferite de timp.

CONCLUZII

Diagramele SYSML sunt necesare pentru modelarea conceptului de structură și comportamentul softului. Datorită algoritmilor elaborați, la etapa proiectării programului se primește o structură fiabilă și o lungime minimală a codului. După detectarea erorilor, cu ajutorul proiectării inverse pot fi regenerate diagramele în baza programului corectat. Astfel documentația electronică se modifică automat și în termeni scurți.

Documentația este necesară pentru a face schimb de idei cu membrii unei echipe, sau cu echipelor ce proiectează periferiile "slave". La necesitate documentația realizată poate fi utilizată pentru proiectarea altor tipuri de sisteme, deoarece primele diagrame oferă o viziune de blocuri. Realizarea diagramelor pe blocuri duce la restrângerea problemelor și la axarea atenției către unele aspecte.

Actualitatea problemei de elaborare a arhitecturii softului se exprimă prin avantajele ce oferă aceasta metoda:

1. Micșorarea timpului de proiectare, testare și validare a sistemului;
2. Majorarea fiabilității software;
3. Generarea documentației în mod automat;
4. Schimb de idei între membrii unei echipe. Gândurile sunt exprimate în mod clar cu ajutorul diagramelor, care sunt evidente din punct de vedere intuitiv. Timpul de asimilare a programului de către altă persoană este mai scurt, datorită studierii blocurilor și nu a codului programului.
5. Se accentuează punctele slabe a sistemului și se delimitează problemele sistemului. Deoarece programul este împărțit pe blocuri, are loc o restrângere a atenției asupra

anumite aspecte a detaliilor fără a lua în considerare restul problemelor. Astfel din haosul de întrebări și probleme pas cu pas se abordează arhitectura software.

6. Datorită instrumentarului, proiectantul se întoarce la diagrama precedentă și are posibilitatea de a efectua modificările necesare.

7. Generarea scheletului programului în baza diagramelor proiectate. Dacă codul programului este structurat, atunci are loc o micșorare a numărului de erori.

8. În baza programului se modifică diagramele proiectate. Datorită funcției de revers engineering are loc perfecționarea documentației în regim automat și micșorarea timpului de testare.

9. După validare în baza documentației electronice poate fi efectuată susținerea tehnică. Adică se studiază problemele și erorile sistemului, și cu ajutorul diagramelor se modifică modul de funcționare a sistemului.

10. Utilizarea arhitecturii proiectului pentru elaborarea altor tipuri de sisteme. Acest avantaj va duce la micșorarea timpului de proiectare pentru alte tipuri de sisteme.

BIBLIOGRAFIE

- [1] Addison Wesley. "Enterprise Patterns and MDA - Building Better Software with Archetype Patterns and UML". Pub Date: December 22, 2003. 528 pagini.
- [2] R.J. Wieringa. Design Methods for Reactive Systems: Yourdon, StateMate, and the UML. Morgan Kaufmann Publishers 2003. 457 pagini.
- [3] 14. Lenny Delligatti. SysML Distilled: A Brief Guide to the Systems Modeling Language. Paperback – November 18, 2013. 265 pagini.
- [4] Cheol-Han Kima, R.H. Westonb, A. Hodgsonb, Kyung-Huy Lee. „The complementary use of IDEF and UML modelling approaches”.
- [5] Jens Eickhoff. „Onboard Computers, Onboard Software and Satellite Operations”. 282 pagini.
- [6] Jonas Solvhoj. „Onboard Computer for Pico Satellite”. Technical University of Denmark January 2002. 63 pagini.
- [7] M. Eng. Toshinori Kuwahara. „FPGA-based Reconfigurable On-board Computing Systems for Space Applications”. October 29, 2009. 200 pagini.