



**JOHN ATANASOFF SOCIETY OF
AUTOMATICS AND INFORMATICS**

International Conference

**AUTOMATICS
AND
INFORMATICS'12**

PROCEEDINGS

Published by

**JOHN ATANASOFF SOCIETY
OF AUTOMATICS AND INFORMATICS**

Bulgaria, Sofia, October, 3 – 5, 2012

JOHN ATANASOFF SOCIETY OF AUTOMATICS AND INFORMATICS

Secretariat Address

Bulgaria
1000 Sofia
108 Rakovsky str.

tel. (+359 2) 987 61 69

fax (+359 2) 987 61 69

e-mail: sai@infotel.bg

www.sai.infotel.bg

www.sai.bg



PROCEEDINGS CD: ISSN 1313-1869

APPLICATION OF UML/MARTE PROFILE FOR MODELING OF BLADE SERVERS

ПРИЛОЖЕНИЕ НА UML/MARTE ПРОФИЛА ЗА МОДЕЛИРАНЕ НА МОДУЛАРНИ СЪРВЪРИ

I. Antonova, I. Batchkova, I. Gerdjikov, G. Elenkov

Dept. of Industrial Automation, University of Chemical Technology and Metallurgy, Bul. Kl. Ohridski 8, Sofia, Phone: +359 2 8163326, E-mail: {iskra.antonova, idilia}@uctm.edu

Abstract: The capabilities of UML for analysis and modeling of real-time systems are supported by the new profile MARTE. MARTE is tailored for systems in which real time constraints play a major role. The paper presents a blade server modelling example by UML MARTE possibilities. Timing requirements and constraints are added in different views and diagrams. Finally some conclusions are made.

Keywords: UML, MARTE, SysML, Blade server, time, clock

ВЪВЕДЕНИЕ

Съвременните информационни системи се характеризират с все по-сложни и динамични архитектури, съставени от различни хардуерни и софтуерни компоненти, които налагат използване на различни подходи и концепции за тяхната разработка. Новите тенденции за разработка на софтуерни системи за реално време налагат изисквания като намаляване времето за проектиране, висока надеждност и сигурност, ниска цена на разработка. Това води до нарастване, както на дела на софтуера в областта на системите за управление, така и до повишаване на неговата сложност и размер. Традиционният подход, използван за разработката на софтуер, базиран на алгоритми и програмен код, не е достатъчно ефективен, продуктивен и надежден. Основните тенденции в тази насока са свързани с проектиране и използване на базирани на стандарти, модулни, многократно използвани, отворени и независещи от конкретния производител приложения, осигурявайки тяхната оперативна съвместимост, преносимост и конфигурируемост. През последните години делят на използваните стандартизирани средства за разработка на разнообразни информационни системи значително нараства. Използването на стандарти насърчава оперативната съвместимост между различните софтуерни средства използвани от проектантите.

Многообещаващ подход за разработка на вградени системи и системи за реално време е използването на обектно-ориентиран подход и Унифицирания език за моделиране UML [1]. Разработката на приложения за реално време с използване на UML е възможна чрез създаването на профили, добавяйки нови мета-класове и мета-конструкции или чрез дефиниране на нови стереотипи, ограничения и тагови стойности [2, 3]. С новите UML профили: MARTE - за моделиране и анализ на вградени системи за реално време [4] и SysML профила за системно инженерство [5] посредством дефинирането на допълнителни разширения от стереотипи, диаграмни разширения и библиотека от модели се създава възможност за моделиране на широк кръг от системи, включващи хардуер, софтуер, данни, персонал, процедури и съоръжения, както и за дефиниране на разнородни

времеви изисквания и ограничения в софтуерните системи.

Както профилът SysML, така и профилът MARTE са подходящи за моделиране на вградени системи, но SysML се използва за моделиране на софтуерни приложения на високо ниво на абстракция, като осигурява възможност за дефиниране на изискванията към разработваните системи, докато профилът MARTE е подходящ за разработка на системи, в които времето играе съществена роля и е насочен към по-ниските нива на разработка като акцентира върху времевите параметри.

Статията е организирана в 4 части. В следващата част е представен кратък обзор на UML профила за системно инженерство SysML и профила за моделиране и анализ на вградени системи за реално време MARTE. Приложението на разглеждания профил за моделиране на модуларни сървъри е представено в третата част на доклада. Накрая ще бъдат представени някои изводи и заключение, както и ще бъдат очертани насоките на бъдещото развитие и усъвършенстване на моделите базирани на UML и профилът му MARTE.

ОБЗОР НА UML ПРОФИЛА MARTE

Обзор на UML профила SysML. SysML [5] е общоцелеви език за моделиране в областта на системното инженерство, който използва подмножество на UML2.1 нотации, като осигурява и допълнителни разширения от стереотипи, диаграмни разширения и библиотека от модели, с цел решаването на широк кръг от задачи в областта, като например специфициране на изисквания, структура, поведение, различни по вид разпределения и дефиниране на ограничения върху свойствата на моделираната система, подпомагащи нейния анализ, проектиране и имплементация.

Използваното подмножество на UML, известно още като UML4SysML включва диаграмите на взаимодействие, машините на състояние, случаите на приложение и профили. Проектирането на структурата на системата за управление се поддържа от четири типа диаграми: Диаграми за дефиниране на блокове (BDD), диаграми на вътрешни блокове (IBD), усилен от параметричните диаграми (PD) и диаграми на пакети. Диаграмите на

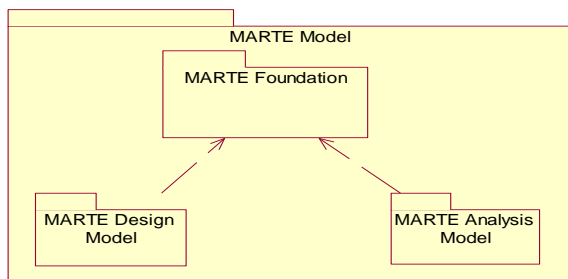
поведение също включват четири типа диаграми: функционални диаграми, диаграми на състояние, диаграми на последователност и диаграми на случаите на приложение. Диаграмите на изисквания могат да бъдат представени графично, или като дървовидни структура и се използват за дефиниране на различни ограничения свързани със системните изисквания и връзките между тях.

Обзор на UML профила MARTE. Профилът MARTE е спецификация на консорциума OMG за моделиране и анализ на вградени системи за реално време и е базиран на профила за условията на планиране, изпълнение и времева спецификация (SPT - Schedulability, Performance and Time profile) използващ стандартните нотации и семантиката на UML [4, 6]. MARTE представява независима методология, предлагаща съвместимо множество от стандартизирани нотации и семантика за проектиране на персонализирани потребителски софтуерни и хардуерни приложения.

Профилът дефинира рамка за анотация на нефункционални свойства в UML моделите на вградени системи чрез добавяне на различни разширения, като:

- възможност за добавяне на модели и ресурси за реално време;
- осигурява моделиране и поддръжка на нефункционални свойства;
- дефинира понятия за моделиране и изпълнение на хардуерни и софтуерни платформи;
- дефинира понятия за алокация на приложенията върху дадена платформа;
- осигурява поддръжка на количествен анализ.

На фиг. 1 е показан MARTE профилът, който се състои от три пакета. Пакетът с име „MARTE Foundation” дефинира основни понятия за проектиране и анализ на вградени системи за реално време. В другите два пакета тези понятия са подобрени. Вторият пакет с име „MARTE Design Model” осигурява понятия за дефиниране на характеристиките за реално време на вградени приложения на високо ниво на абстрактност, като осигурява детайлна спецификация на хардуерната и софтуерната платформи. Третият пакет „MARTE „Analysis Model” се използва като основа за анализ и е разделен на три части – първата дефинира понятия за количествен анализ, а втората и третата акцентират върху планиране и анализ на изпълнението. Профилът MARTE покрива целия етап на разработка на различни приложения - от дефиниране на изискванията до детайлното моделиране на хардуера и софтуера [7, 8].



Фиг. 1: Изглед на профила MARTE

Във всяка система времевите ограничения се представят като основни аспекти за анализ, необходими при определяне изпълнението на системи от различен вид. Чрез профила MARTE се дефинират два типа време – логическо и хронометрично, като в системите за реално време, факторът време е решаващ, тъй като системите

трябва да удовлетворяват както твърди (hard), така и меки (soft) времеве ограничения.

Задаването на времеве ограничения в моделите е възможно чрез дефинираните нови стереотипи за елементите на структурните диаграми, функционалните диаграми и диаграмите на състояния, представящи времевите единици заедно с времевите характеристики и ограничения като период и времезакъснение [9].

Допълнителните стереотипи, използвани за моделиране на време са <<Clocks>>, които са екземпляри на <<ClockType>>, <<ClockConstraint>>, <<TimedEvent>> и <<TimedProcessing>>. Чрез стереотипа <<ClockType>> се дефинират свойствата и вида на използвания часовник, функциите на часовника като резолюция, максимална стойност и др. Стереотипът <<ClockConstraint>> се използва, за да представи ограниченията на часовника чрез определяне на зависимостта на времевата структура в дадена времева област. Спецификацията на времевите ограничения се представя декларативно, посредством езика OCL [10]. В MARTE чрез стереотипа <<TimedEvent>> се специфицира насочването към точно определен часовник. В случаи на повтарящо се събитие, се дефинират периода и честотата на повторенията. Стереотипът <<TimedProcessing>> разширява мета-класовете на елементите представящи поведение, съобщения и действия [11].

Разширенията в профила MARTE се реализират чрез добавяне на нови стереотипи в библиотека от модели. Елементите от библиотеката с модели могат да бъдат използвани за моделиране на различни нива на мета-модели, профили и приложения.

Друга важна характеристика на профила MARTE, с която чистият UML не може да се справи, са възможностите за дефиниране на различни форми на време. С нея се осигурява възможност на потребителите да задават времеве изисквания и ограничения.

Както в SysML, така и в MARTE е въведено понятието алокация. В профила MARTE под алокация се разбира избор на платформа, на която разработваното приложение ще бъде реализирано. Алокацията обхваща както пространствените разпределения, така и аспектите на времевото планиране, за да разпредели различни алгоритми по изпълнението към съществуващите изчислителни и комуникационни ресурси и услуги. Времево планиране е необходимо в случаите, в които няколко приложения се разпределят на една и съща платформа или в случаите, когато се изисква синхронизация между различни елементи.

В профила MARTE етапът на анализ се базира на софтуерното поведение, както и на планирането и изпълнението, като дефинира основни моделиращи понятия, използвани за дефиниране на нефункционалните изисквания. Етапът на анализ е разделен на две фази – анализ на планирането и анализ на изпълнението.

ПРИМЕР: МОДЕЛИРАНЕ НА ПОВЕДЕНИЕТО НА ХАРД ДИСК

UML/MARTE профилът е използван за моделиране на твърд диска на модуларен сървър. Моделът на твърдия диска на модуларен сървър, реализиран със средствата на UML профила MARTE е разработен в среда на Rapyrus [12], който е инструмент с отворен код, разработен в рамките на проект на Eclipse, целящ да осигури лесен начин за създаване, редактиране, персонализиране и интегриране на различни модели, поддържащи UML2.0 и

свързаните с него профил за системно инженерство SysML и профила MARTE, използвайки стереотипи и мета-класове. В [13] е представена структурата на модулaren сървър посредством диаграма на класове.

Описание на UML диаграми на състоянията. Диаграмата на състоянието представя поведението на системата. Тя описва последователност от състояния, които може да приеме един обект в продължение на своя жизнен цикъл, изхождайки от някакво първоначално състояние и някакво целево крайно състояние. Диаграмите отразяват чрез какви събития се предизвиква промяна на състоянието и при какви предпоставки това е допустимо. Те служат предимно за описание поведението на отделен обект по време на жизнения му цикъл. Диаграмата на състояние включва състояния и преходи, които се активират чрез събития. Състоянието на един обект се описва чрез определена стойност на атрибутите в определена ситуация, в която се намира. Ако обектът получи съобщение, например запитване, то се решава, когато всички условия за това са изпълнени (събития или операции). Изпълнението на операцията предизвиква преминаването в ново състояние, когато са изпълнени определени условия. Диаграмата на състояние е подходяща за представяне на поведението на обект в различни сценарии (случаи на приложение). При това поведението се дефинира точно и пълно [14, 15]. Тя може да бъде използвана за моделиране на хардуерни елементи, реализирани като електронна и/или интегрална схема. Прието е, че всяко събитие е свързано с област от дискретни времена. Преминаването от едно състояние в друго в дадена схема се извършва чрез преходи, наречени микростъпки. Поведението на всеки хардуерен елемент може да бъде описано с диаграма на състоянията, като състоянията от диаграмата съответстват на определено логическо устройство [16]. От друга страна полуформалният UML модел може да бъде формализиран използвайки трансформация на диаграмите на състоянието във времеви мрежи на Петри (Time Petri Nets) [17].

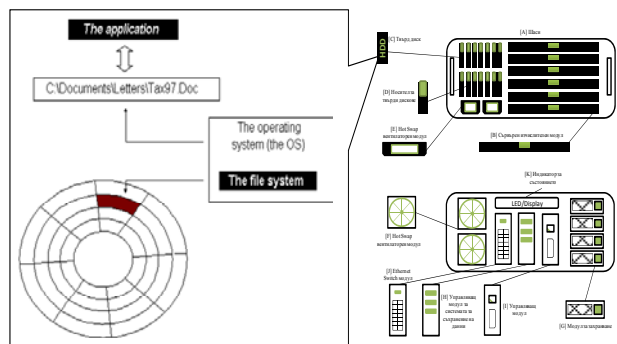
Описание на твърди дискове. Названието модулaren сървър се появява, когато в една компютърна платка се включват процесор, памет, входове изходи (I/O) и устройства за постоянно съхранение на програмите (flash памет или малък твърд диск). Тези, така наречени, модули могат да оперират независимо в едно общо шаси, изпълнявайки задачите на множество отделни сървъри, но с по-голяма ефективност [18, 19].

Твърдият диск (наричан още хард диск) е енергонезависима компютърна памет, в която информацията се съхранява по магнитен път върху плочи покрити със специален слой. Запаметяващото устройство използва набор от една или повече дискови плочи около обща ос в т.нар. дисков пакет (фиг.2). Данните се записват върху концентрични окръжности, които се наричат писти. Целият дисков пакет се върти с постоянна ъглова скорост около оста си, задвижван от електродвигател. За двете работни повърхности на всяка плоча има универсална глава или по-често блок от четяща, изтриваща и записваща глави. Блокът магнитни глави се задвижва чрез рамо, извършвайки операцията позициониране на главите чрез радиално преместване.

От гледна точка на оптимизацията на достъпа до данните върху пакета във физическата им организация е въведено понятието цилиндър, обединяващо мислено пистите с еднакъв диаметър от всички работни повърхности. Данните се запазват върху диска под формата на файлове. Файлът е низ от байтове. Байтовете могат да представляват ASCII кодове (знаци в текстов файл), но

могат да бъдат и част от кода на програма или база данни. Когато процесорът се нуждае от тези байтове, той подава заявка към контролера на твърдия диск и ги получава. Има два начина за измерване производителността на твърдите дискове:

- **Скорост на данните:** Скоростта на данните показва какво количество данни в секунда може да изпраща диска към системата.
- **Време за търсене:** Времето за търсене показва какво закъснение има между заявката, подадена от процесора за даден файл и времето в което първият байт от него ще бъде преведен по шината.



Фиг.2: Структура на твърд диск на модулaren сървър

Диаграма на състоянията за твърд диск. Диаграмата на състоянието за твърдия диск е показана на фиг.3 и описва неговите основни състояния и връзките при преминаване от едно състояние в друго чрез т.н. преходи. Твърдият диск има три състояния - "Read", "Write" и "Idle", като състоянията на четене "Read" и запис "Write" протичат на три етапа представени като подсъстояния с имена "Seek", "Rotation" и "Transfer".

От начално състояние твърдият диск преминава в две нови състояния - четене "Read" или запис "Write". Преминаването е възможно след активиране на преходите "Start Read" или "Start Write" съответно за четене и запис. В случай, че дискът е в състояние "Read" се активира преходът "Start Read", като се изпълнява действие по преобразуване на физическия адрес в блок от логически адреси (physical location→LBA), в противен случай дискът е в състояние "Write" и се активира прехода "Start Write", където се изпълнява действие за преобразуване на блока от логически адреси във физически адрес (LBA→physical location). Дискът може да бъде изведен от състояние "Read" при настъпване на събитието "Data Read", при което се преминава в състояние "Idle", а извеждането от състояние "Write" става при настъпване на събитието "Data Write", при което дискът отново преминава в състояние "Idle". В състояние "Idle" се изпълняват действията по запис или четене на части от файла във файлова таблица (Write/Read in file table). Дискът може да бъде изведен от това състояние по два начина: или при появата на събитие "ContinueReading" и изпълнение на условието на условието за прехода [period=true], при което се преминава в състояние "Read" или при появата на събитие "ContinueWriting" и изпълнение на условието на условието за прехода [period=true], при което се преминава в състояние "Write".

И състояние "Read" и състояние "Write" са съставни и в тях са вложени състоянията "Seek", "Rotation" и "Transfer". В състояние "Seek" при четене на данни се извършва действие за позициониране върху подходящ сектор (Get track), докато при запис, сектори се позиционират във файловата таблица (Position over track). При удовлетворяване на условието на прехода

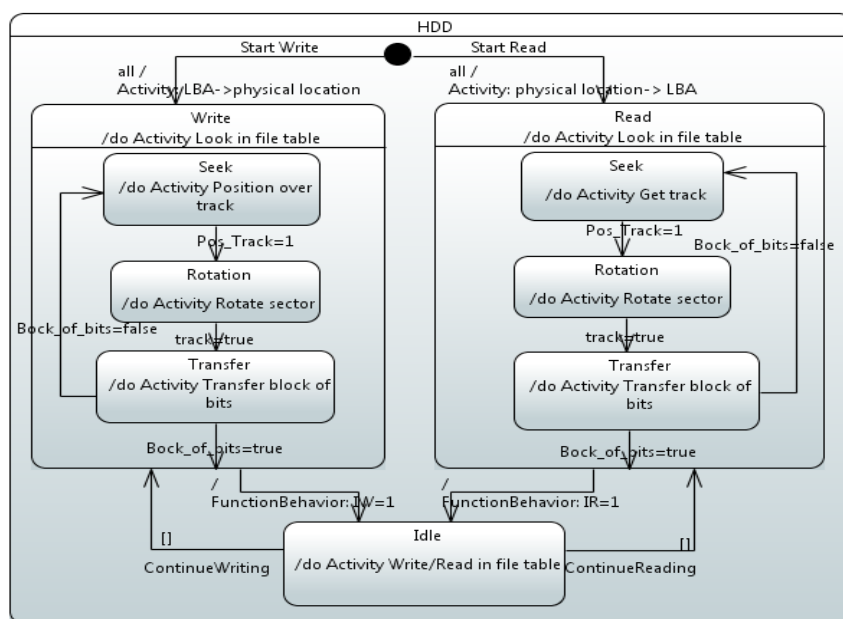
[Pos_Track=1] се преминава в състояние "Rotation", където се извършва действие по завъртане на секторите за запис или четене (Rotate sector). Извеждането от това състояние е възможно след удовлетворяване условието за прехода [track=true] и се преминава в състояние "Transfer". Действието, което се изпълнява в това състояние е прехвърляне на блока от битове, съответно за четене и запис (Transfer block of bits). Това действие се извършва докато целият файл бъде прочетен или записан, свързано с удовлетворяване на условието [Block_of_bits=true], в противен случай при удовлетворяване на условието [Block_of_bits=false] файлът не е прочетен/записан и се преминава в състояние "Seek".

ЗАКЛЮЧЕНИЕ

Докладът е посветен на приложението на UML профила за моделиране и анализ на вградени системи за реално време

MARTE. Чрез него се осигурява възможност за моделиране на различни времеви характеристики и ограничения на моделираната система под формата на допълнително дефинирани стереотипи.

Друго основно предимство от приложението на UML профила MARTE е възможността за анализ на моделираната система, както и за детайлно проектиране на хардуерните и софтуерните платформи на моделираното приложение. Тази възможност е използвана при моделиране поведението на твърд диск на модулaren сървър. За целта е използвана работната среда на Papyrus, която е среда с отворен код, осигуряваща възможност за моделиране на приложения, както в среда на UML/SysML, така и на UML/MARTE.



Фиг.3: Диаграма на състоянията за твърд диск

ЛИТЕРАТУРА

- <http://www.omg.org/>
- Антонова И., И. Бачкова (2005), Сравнителен анализ на използването на UML в приложения за реално време, "Развитие, бъдеще, перспективи и иновации в науката - 2005", стр.103÷113, София, 22 юни.
- Antonova I. (2011), Application of Object-Oriented Approach in Control Systems, Anniversary Scientific Conference „40 YEARS DEPARTMENT OF INDUSTRIAL AUTOMATION”, University of Chemical Technology and Metallurgy, pp. 189÷192, 18 March 2011, Sofia.
- OMG MARTE, Profile for Modeling and Analysis of Real-Time and Embedded (MARTE) systems, Beta 1, 2007.
- OMG. UML for Systems Engineering, Request for Proposals - RFP ad/2003-3-41, Object Management Group, Needham, September 2003
- Mallet F., Simone R., MARTE: A Profile for RT/E Systems Modeling, Analysis —and Simulation?, 1st Int. Conf. on Simulation Tools and Techniques for communications, Networks and Systems SIMUTools'08, pp. 1-8
- Hagner M., Huhn M., Tool Support for a Scheduling Analysis View, in Design, Automation and Test in Europe (DATE 08), 2008.
- Andre C., Mallet F., Khan A., Simone R., Modeling SPIRIT IP-XACT with UML MARTE, In: Proc. DATE Workshop on Modeling and Analysis of Real-Time and Embedded Systems with the MARTE UML profile, pp. 35-40, 2008.
- Peraldi-Frati M., Sorel Y., From high-level modelling of time in MARTE to realtime scheduling analysis, In MoDELS'08 W. on Model Based Architecting and Construction of Embedded Systems on ACESMB, pp. 129–143, Toulouse, France, September 2008
- OMG, Object Constraint Language, version 2.0, May 2006, OMG document number: formal/06-05-01.
- Andre C., Mallet F., Simone R., Modeling time(s). In Engels, G., Opdyke, B., Schmidt, D.C., Weil, F., eds.: MoDELS. Volume 4735 of Lecture Notes in Computer Science, Springer (2007) 559_573
- <http://www.eclipse.org/modeling/mdt/papyrus/>
- Antonova I., Gerdjikov I., Batchkova I., Elenkov G., Информационно моделиране на модулари сървъри с използване на UML, IX международен конгрес "МАШИНИ, ТЕХНОЛОГИИ, МАТЕРИАЛИ", 19-21.09.2012, Варна
- Boggs W., Boggs M, Sybex Mastering UML wirh Rational Rose 2002, Sybex, 2002.
- Duby C., Implementing UML Statechart Diagrams, Embedded System Conference, (September 2001), USA
- Labiak G., From UML statecharts to FPGA — the HiCoS approach, Forum on Spec. & Design Langs - FDL '03, Frankfurt 2003
- Ge N., Pantel M., Time Properties Verification Framework for UML-MARTE Safety Critical Real-Time Systems, Proceeding ECMFA'12 Proceedings of the 8th European conference on Modelling Foundations and Applications Pages 352-367
- http://en.wikipedia.org/wiki/Blade_server
- Blade Server Benefits. Evaluating new server technology in an IT environment, Intel Information Technology White paper, May, 2004.