



**JOHN ATANASOFF SOCIETY  
OF AUTOMATICS AND INFORMATICS**

**JOHN ATANASOFF  
CELEBRATION DAYS**

*International Conference*

**AUTOMATICS  
AND  
INFORMATICS'11**

**PROCEEDINGS**

**Published by**

**JOHN ATANASOFF SOCIETY  
OF AUTOMATICS AND INFORMATICS**

**Bulgaria, Sofia, October 3 - 7, 2011**

## 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](mailto:sai@infotel.bg)

[www.sai.infotel.bg](http://www.sai.infotel.bg)

[www.sai.bg](http://www.sai.bg)

## APPROACHES OF SOFTWARE ENGINEERING TO IMPROVE THE DEVELOPMENT OF IEC-61499 BASED DISTRIBUTED CONTROL SYSTEMS

### ПОДХОДИ НА СОФТУЕРНОТО ИНЖЕНЕРСТВО ЗА ПОДОБРЯВАНЕ НА РАЗРАБОТКАТА НА IEC-61499 БАЗИРАНИ РАЗПРЕДЕЛЕНИ СИСТЕМИ ЗА УПРАВЛЕНИЕ

I. Batchkova, I. Antonova, J. Tellalyan

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

**Abstract:** One of the ways to tackle the challenges facing today's control systems, associated with achieving a high degree of interoperability, agility and reconfigurability, is the transition from a centralized architecture of information management and control systems to a distributed one. Promising trend in this direction is the implementation of the IEC61499 standard, offering reference architecture and models for development of distributed systems. The main purpose of this report is to summarize and compare different approaches and methods to improve the development of IEC-61499 based distributed control systems through their combination with different approaches and modern methods of software engineering.

**Keywords:** distributed control system, IEC-61499, UML, SysML, ANSI/ISA-S88, PLCopen

#### ВЪВЕДЕНИЕ

Световната конкуренция между предприятията и новите производствени стратегии, възприети от предприятията налагат необходимостта от нов тип информационно-управляващи системи, характеризиращи се както с висока степен на хоризонтална и верикална интеграция, така също и със самоорганизиращи се характеристики, с цел бърза адаптация към промените и смущенията в средата.

Един от начините за справяне с тези предизвикателства, продиктуван от бурното развитие на електрониката, информационните и комуникационни технологии, които доведоха до появата на редица нови технически средства и нововъведения като: полеви шини, интелигентни устройства (с вградени микропроцесори), размити програмируеми контролери и други, е подмяна на централизираната структура на информационно-управляващите системи с разпределена, която притежава значително по-големи предимства в сравнение с централизираните такива. Естествена реакция на тези промени е и новия стандарт IEC-61499 за изграждане на разпределени системи за измерване и управление на процеси [1, 2].

Цел на доклада е да обобщи и сравни различни подходи и методи за разработка на разпределени системи за управление в различни приложни области на базата на референтната архитектура и модели, залегнали в стандарта IEC-61499, комбинирани със съвременни подходи и методи на софтуерното инженерство. Предлаганите подходи се отличават по степента на формализация на моделите, използвания модел на жизнен цикъл на разработката, както и конкретните фази на жизнения цикъл, обхванати от подхода. За целите на изследването са подбрани тестови примери от разработката на разпределени системи за управление в различни производствени области – управление на непрекъснати процеси, периодични процеси и от областта на дискретните системи.

В следващата част на доклада ще бъде представен кратък обзор на стандарта IEC-61499, след което ще бъдат описани нак-

ратко различни подходи, разработени от колектива за разработката на разпределени системи за управление, подкрепени с конкретни тестови примери. Специално внимание е отделено на подходите, които са базирани на използването на съвременни методи и подходи на софтуерното инженерство, като: методите за формална спецификация и верификация, сигнално-интерпретиращи мрежи на Петри (SIPN), унифицирания език за моделиране UML и неговия профил за системно инженерство SysML, както и различни модели на процеса на разработка на софтуер или по-известни като модели на жизнения цикъл на разработката на софтуер. Накрая ще бъдат представени някои изводи и заключение, както и ще бъдат очертани насоките на бъдещото усъвършенстване и развитие на подходите.

#### СТАНДАРТ IEC-61499

*Кратък обзор на стандарта.* Стандартът IEC-61499 дефинира основните понятия и референтна архитектура за проектиране на многократно използваеми и компоненто-базирани разпределени системи за управление. На фиг.1 е представена базираната на стандарта референтна архитектура под формата на UML диаграма на класовете. В основата на стандарта, като базова структурна единица на приложението е понятието “функционален блок” (ФБ). Функционалният блок може да бъде използван за дефиниране на многократно използваеми софтуерни компоненти, които на базата на избрана методология да бъдат използвани за проектиране на сложни, децентрализирани, разпределени системи за управление. Понятието “Функционален блок” е дефинирано от Левис като: “абстрактен механизъм, който позволява капсулиране на промишлените алгоритми във форма, която може лесно да бъде разбрана и приложена от инженера, който не е специалист в реализацията на сложни алгоритми” [2]. Стандартът дефинира три основни типа функционални блока (фиг.1): основен, съставен и интерфейс за услуги.

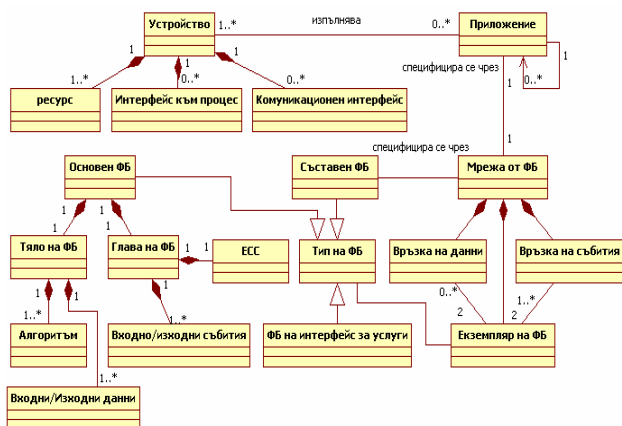
Основният функционален блок се състои от две основни части, условно наречени „главна част” и „тяло”. Главната част се представя посредством интерфейс на входните и из-

ходни събития, а вътрешността му включва “граф за изпълнение на управлението” (Execution Control Chart (ECC)). Графът за изпълнение на управлението е машина на състоянието, използвана да управлява изпълнението на алгоритмите, асоциирани към функционалния блок. Графът за изпълнение на управлението се състои от състояния, преходи и действия, които предизвикват изпълнението на алгоритми, асоциирани към състоянията на ECC, в отговор на събитийни входове. Едно от състоянията е т.н. начално състояние към което не могат да бъдат асоциирани никакви действия, докато към другите състояния на графа за изпълнение на управлението могат да бъдат асоциирани едно или няколко действия. Всяко действие може да бъде асоциирано с конкретен алгоритъм или изходно събитие. Придвижването на ECC машината на състоянието от едно състояние на изпълнение на управлението в друго се реализира чрез активирането на преходи. „Тялото” на основния функционален блок се представя чрез интерфейс на входните и изходни данни вътрешни данни и вътрешни алгоритми. Функционалният блок се характеризира чрез име на типа и името на екземпляра, които го идентифицират еднозначно. Входните и изходни данни и събития са необходими за връзка между отделните функционални блокове и изграждането на по-сложни системи за управление.

Съставният функционален блок служи за свързване и капсулиране на основни ФБ с цел изграждане на по-големи логически компоненти на една система. За разлика от основния ФБ, той не съдържа в себе си граф за управление на изпълнението, алгоритми и вътрешни данни, а се управлява от събития. Съставният ФБ е изграден като мрежа от ФБ, свързани посредством своите входно/изходни интерфейси.

ФБ на интерфейс за услуги (ИФБ) се използва основно за целите на комуникациите и мениджмънта. ФБ на комуникационни интерфейси реализират два основни вида транзакции – еднопосочни от типа „Publish-Subscribe” и двупосочни от типа “Client-Server”. За да се обособи едно приложение като ресурс, е необходимо да се дефинират подходящите за случая входно/изходни интерфейсни ФБ на комуникациите.

Освен моделите на функционалния блок, изброени по-горе, стандартът дефинира още 3 вида референтни модели, на базата на които може да бъде проектирана разпределената система за управление на различните етапи от нейния жизнен цикъл: системен модел, модел на устройствата и модел на ресурсите. Чрез системния модел се специфицира разпределението на дадено приложение между устройствата, като едно устройство може да изпълнява няколко приложения. Устройствата са контейнери на ресурси, които правят възможно независимото изпълнение на функционалните блокове в мрежата. Моделът на ресурсите специфицира мястото, където се изпълнява едно приложение, като обезпечава основните средства за това.



Фиг. 1: Референтна архитектура на стандарта IEC-61499

Програмни среди за проектиране на функционални блокове, съгласно стандарта. Понастоящем инженерните среди, подпомагащи процесите на разработка и симулация на разпределени системи за управление, съгласно новия стандарт IEC-61499, са в процес на активна разработка. Съществуват няколко инженерни среди, поддържащи разработката на системи за управление, базирани на стандарта IEC6199. Най-известната и най-често използвана в академичните разработки за момента отворена среда е FBDK (Function Block Development Kit) [3]. Продуктът е разработен от Rockwell Automation и позволява дефинирането на основни и съставни функционални блокове, ресурси, устройства и системни конфигурации. За целта се използва XML формат, който е специфициран в IEC-61499 и Java интерфейс, подпомагащ визуалното тестване на разработените модели. В средата на FBDK съществува библиотека от графични интерфейси и възможности за симулация.

Някои недостатъци на стандарта. Основните недостатъци на стандарта могат да бъдат обобщени, както следва:

- Стандартът не се поддържа от формални методи, въпреки че целият инженерингов процес при проектирането на разпределени системи предлаган в стандарта IEC-61499 е подобрен по отношение на надеждност, преизползваемост и интероперабилност.
- Стандартът IEC-61499 е чисто функционален и не засяга жизнения цикъл на разработка на системите, както и не предлага методология за разработка на разпределени приложения за управление. Нейният избор е предоставен на разработчика.
- Дефинирането на изисквания към системите за управление и етапите на анализ не са засегнати в стандарта, което ограничава възможността за ранно откриване на грешки, допуснати във фазите на проектиране на приложенията.

Един от начините да се преодолеят тези недостатъци на този етап от развитието на стандарта е неговото комбинирано прилагане със съвременни методи и подходи на софтуерното инженерство други съществуващи средства. Някои от тези комбинирани подходи са предмет на кратко представяне в следващата част на доклада.

#### СИСТЕМАТИЗАЦИЯ НА ПОДХОДИТЕ ЗА РАЗРАБОТКА НА РАЗПРЕДЕЛЕНИ СИСТЕМИ ЗА УПРАВЛЕНИЕ БАЗИРАНИ НА СТАНДАРТА IEC-61499

Систематизираните подходи за разработка на разпределени системи за управление на базата на стандарта IEC-61499 са дело на колектив от катедра „Автоматизация на производството” на ХТМУ-София под ръководството на доц. И. Бачкова, като някои от подходите са разработени съвместно и в сътрудничество с колективи от ТУ-София (катедра ТМММ, проф.Г.Попов) и Университета в Заарбрюкен, Германия (катедра „Автоматизация на производството”, проф. Г. Фрай).

На фиг.2 са обобщени предлаганите подходи, които са насочени към повишаване на потенциала за прилагане на стандарта и повишаване на ефективността и качеството на разработките, като могат да бъдат обособени в две групи:

- Подходи, представляващи комбинация на IEC-61499 базирани модели със съвременни подходи и средства на софтуерното инженерство, като формалните методи за спецификация и верификация, използването на пъргави модели в процесите на софтуерна разработка, използване на UML/SysML за разширяване на етапите на разработка или като алтернативен вариант при проектирането на IEC-61499 базирани системи за управление;
- Подходи, комбиниращи концепцията на стандарта с постигнатите успехи в областта на управлението на конкретни производствени области, като управлението на пери-

одични процеси (ANSI/ISA S88 - стандарт ISO-61512 [9]) и инициативата PLCopen за управление на движенията на базата на стандарта IEC-61131-3 [10].

В следващата част на доклада ще бъдат представени идеите само на подгрупата подходи, използващи постиженията на софтуерното инженерство, които в последната част на доклада ще бъдат илюстрирани с конкретен пример.



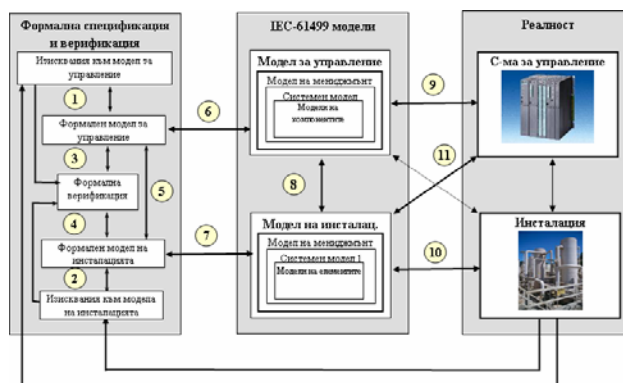
Фиг.2: Систематизация на подходите за разработка на IEC-61499 базирани системи за управление

### IEC-61499 БАЗИРАНИ ПОДХОДИ, КОМБИНИРАНИ С МЕТОДИ НА СОФТУЕРНОТО ИНЖЕНЕРСТВО

*Комбиниран подход с използване на IEC-61499 и формализация на моделите.* Съчетаването на концепцията на стандарта IEC-61499 с възможностите, които предоставят формалните подходи за спецификация и верификация води до значително повишаване на ефективността на разработката, поради факта, че на сравнително ранни етапи може да се верифицира специфицираната система за управление, на базата на прилагането на различни формални методи и подходи. На фиг.3 е илюстрирано в обобщен вид идеята за включване към разработката на етапи на формалната спецификация и верификация на моделите с използване на метода „model checking”. Трябва да се отбележи, че този подход може да бъде реализиран с различни формални методи и програмни средства. Така например в [11, 12] за целите на формалната спецификация на системата са избрани SIPN мрежи на Петри, а за нейната верификация се използват възможностите за трансформация на мрежата във времеви автомат и продукта NuSMV, реализиращ метода „model checking” с използване на BDD диаграма и SAT метод. Друга възможност е подмяната на NuSMV с UPPAAL, както е демонстрирано в [13]. Този продукт може да се използва и самостоятелно, като за целта моделът на системата за управление се специфицира с времеви автомати, а изискванията се дефинират с използване на темпорална логика [14].

*Комбиниран подход с използване на IEC-61499 и UML/SysML.* Унифицираният език за моделиране UML [15] и профила му за системно инженерство SysML [16] могат да бъдат използвани по разнообразни начини в етапите на разработка на IEC-61499 базирани системи за управление. Подробен обзор на тези подходи е представен в [17]. Предложените от авторите подходи се отличават помежду си по степента на използваните в рамките на разработката UML модели, средата за разработка, както и използваната версия на UML или допълнителното включване на SysML. Така например в [18] е предложен подход, използващ версията UML1.4, като се базира на програмните среди Rational Rose на IBM и профила UML-MASTI, който цели моделиране на реално-времеви характеристики на системата. В [19, 20] също са използвани старите версии на UML, но в тези случаи за целите на разработката се предлага използването на инженеринговата среда CORFU, свързана непосредствено с

IEC-61499 базираната развойна среда FBDK. Развитието на езика UML и създаването на профила за системно инженерство SysML подобряват степента на използваемост на езика по отношение на разработката на системи за реално време, като дават възможност за покриване на целия жизнен цикъл на разработката. В тази връзка са изследвани различни подходи със и без помощта на профила SysML в средата на Rhapsody на Telelogic понастоящем собственост на IBM [21, 22]. Трябва да се отбележи, че предлаганите подходи са приложими, както за дискретни, така и за непрекъснати системи. Значителен е и успеха в областта на формализацията на UML и възможността за верификация и валидация на предлаганите модели на различни фази от тяхната разработка, което е предизвикателство за по-нататъшно подобряване и усъвършенстване на предложените подходи. Показателен е и опитът целящ, на базата на IEC-61499 базиран подход с използване на UML/SysML, да бъде проектирана мултиагентна система за управление [23].



Фиг.3: Илюстрация на IEC-61499 базиран подход, комбиниран с формални методи за спецификация и верификация

*IEC-61499 базиран подход с използване на Harmony модел на жизнения цикъл на разработката на системи за управление.* Както UML така и SysML не предоставят методология за моделиране, което налага необходимостта от избор на методология, която да бъде приложена в процеса на разработка на системата за управление. В тази връзка с оглед на добро структуриране на разработката и избягване на възможността за допускане на двусмислие е предложен подход, базиран на модифициран вариант на методологията Harmony-SE, която е част от цялостна методология за разработка на софтуер и интегриране на системи [24]. Процесът на разработка, илюстриран на фиг.4, включва следната последователност от задачи: Анализ на изискванията; Анализ на функционалността на системата; Проектиране на архитектура; Спецификация на проектирания хардуер/софтуер. Етапът на анализ на изискванията започва с анализ на входовете за процеса. Изискванията на потребителите се трансформират в множество от изисквания, дефиниращи какво трябва да прави системата и нейните качествени показатели. За целта, първоначално се създава таксономия на диаграмите на изисквания (RD), изобразяваща използваните изисквания и случаи на приложение на системата (SUC).

Етапът на системен функционален анализ включва трансформиране на дефинираните функционални изисквания към системата в съгласувани системни функции, като за целта се използват една или повече диаграми на случаите на използване. Всеки случай на използване се анализира поотделно, като за целта се използват BDD и IBD, които представят съставната структура на системата. За моделиране на функционалността на системата на високо ниво на абстрактност се използват различни диаграми от типа “черна кутия”, като функционална диаграма (BB-AD) и диаграми на последователност (BB-SD).



Третият етап на методологията е проектиране на архитектурата и включва проектиране структурата на подсистемите и поведението базирано на диаграми, използвани на предходния етап, но използващи варианта на “бялата кутия” - (WB-UCD, WB-AD, WB-SD). Дефинирането на портове и интерфейси са другите важни задачи, както и описване на поведението на системата, използвайки диаграми на състояния.



Фиг.4: Илюстрация на комбиниран обектно-ориентиран подход

Последният етап, свързан с имплементацията на системата, включва спецификация на проектирания хардуер/софтуер. Всички етапи от методологията включват задачи за верификация и валидация, които се базират на моделите, разработени в предходните етапи и на дефинираните изисквания.

#### ПРИМЕР НА КОМБИНИРАН ПОДХОД ЗА РАЗРАБОТКА НА IEC-61499 БАЗИРАНА СИСТЕМА ЗА УПРАВЛЕНИЕ НА ЖИЛИЩНИЯ КОМФОРТ.

В тази част на доклада ще бъде представен тестови пример, илюстриращ комбинация от предложените по-горе подходи (фиг.5). По този начин се създава възможност за поэтапна разработка на системата за управление, съгласно избран модел на жизнения цикъл на разработката (в случая е използван модифициран *Harmony* модел), поддържане на модели на всяка фаза на разработката посредством използване на UML/SysML, както и използване на формални методи за спецификация, моделиране и верификация на разработката на базата на SIPN мрежи на Петри и NuSMV.

Предлаганият подход покрива основните етапи на жизнения цикъл на разработката на системи за управление, включващ дефиниране на изискванията към системата за управление чрез използване на UML диаграма на приложението (фиг.6(1)) и SysML базирана диаграма на изискванията (фиг.6(2)). В етапите на проектиране на системата са използвани два типа SysML диаграми – диаграма за дефиниране на блокове (Block Definition Diagram (BDD)) и вътрешно-блокова диаграма (Internal Block Diagram (IBD)), поз-

воляващи както моделирането на основната структура на системата, така и комуникациите между отделните елементи на системата. Поради сравнително не сложната задача не са използвани други системни UML/SysML модели. За моделиране на поведението на контролера е използвана функционална диаграма, която лесно може да бъде трансформирана в сигнално-интерпретираща мрежа на Петри (фиг.6(5), фиг.6(6)), която да бъде верифицирана чрез генерирането на SMV код посредством който се извършва проверката на модела, с използване на метода „Model Checking”. Следващата стъпка е създаване на модел на управлението, представляващ мрежа от функционални блокове, съгласно стандарта IEC-61499 (фиг.6(4)). Използваните функционални блокове са или част от библиотеката на стандарта IEC-61499, или са новосъздадени такива, като по този начин се разширява съществуващата до момента библиотеката в рамките на стандарта. Целта е преизползване на изградените вече компоненти за управление в етапа на разработка на различни приложения с идентична функционалност, което спестява време и финансови средства.

Верифицираният формален модел за управление може да бъде имплементиран в програмен код. Съществува възможност и за директна имплементация на мрежата от функционални блокове чрез използване на JAVA базираният контролер „Netmaster” на фирмата "ELSIST". Наред с директната имплементация, изграденият модел, разработен в среда на SIPN, автоматично генерира кодовете - Structure Text (ST) и Instruction List (IL), които са част от стандарта IEC-61131, както е показано на фиг.5.

#### ЗАКЛЮЧЕНИЕ

В доклада са систематизирани авторски подходи в областта на разработката на IEC-61499 базирани разпределени системи за управление. Основната цел на предложените подходи е да подобрят процесите на разработка, чрез използване на съвременни методи, подходи и средства на софтуерното инженерство. Използвани са редица такива подходи като: формални методи за спецификация и верификация на моделите, както и набиращите скорост MDD подходи, представени от езика за моделиране UML и профила за системно инженерство SysML, както и модифицираната методология Harmony.

Важна стъпка към подобряване на предложените подходи е тяхната интеграция, както е илюстрирано в последната част на доклада, както и разработката на специализирани за целите на областта интегрирани среди за разработка и използването на специални UML/SysML профили.



## ЛИТЕРАТУРА

1. IEC-61499 (2005), International Standard IEC-61499, Function Blocks, Part 1 - Part 4, International Electrotechnical Commission (IEC), Technical Committee TC65/WG6, IEC Press, Jan.
2. Lewis R. (2001), *Modelling Control Systems using IEC 61499*, The Institution of Electrical Engineers", London, United Kingdom.
3. Holobloc Inc. (2011), Function Block Development Kit (FBDK). available: <http://www.holobloc.org/>
4. 4DIAC (2011), Framework for Distributed Industrial Automation (4DIAC) Available: <http://www.fordiac.org>
5. K. Thramboulidis (2002), Development of Distributed Industrial Control Applications: The CORFU Framework, in: Proc. of 4th IEEE International Workshop on Factory Communication Systems, Sweden.
6. FBench. (2011), Open Source Function Block Engineering Tool., available: <http://ooneida-fbench.sourceforge.net/>
7. ICSTriplex. (2011), ISAGRAF v. 5.0. available: [www.isagraf.com](http://www.isagraf.com)
8. NxtControl, (2011), nxtStudio, available: [www.nxtcontrol.com](http://www.nxtcontrol.com)
9. ISA, Batch Control Part 1: Model and terminology, The International Society for Measurement and Control, ISA Press, ISA – S88.01-1995.
10. John K.H., M. Tiegelkamp, IEC 61131-3: Programming Industrial Automation Systems, Springer-Verlag, Berlin-Heidelberg-New York, 2001.
11. D. Dimitrova, G. Frey, I. Batchkova (2007), *Sequential control at the supervisory level of batch plant using Signal Interpreted Petri Nets*, International Conference "Automatics and Informatics'07", 3-6 October, 2007, Sofia, pp. V-17÷V-20.
12. Dimitrova D., G. Frey, I. Batchkova (2005), *Formal approach for modeling and verification of IEC 61499 Function Blocks*, Proceedings of the International Conference AMTECH 2005 "Advanced Manufacturing Technologies", University of Russe, Bulgaria, Volume 44, Book 2, pp. 731-736, Nov. 9-11.
13. Dimitrova D., I. Batchkova, G. Frey (2005), Component based Automation, Technical Report, Technical University, Kaiserslautern, September.
14. Димитрова Д., Бачкова И. (2006), Спецификация и верификация на компонентно базирани системи за управление с времеви автомати, Международна научна конференция "Развитие – бъдещи перспективи и иновации в науката'2006", 12.12.2006, София, стр.23-33.
15. OMG - UML, Unified Modeling Language UML (2011), <http://www.omg.org/>
16. OMG SysML: The OMG Systems Modeling Language, <http://omgsysml.org/index.htm>, May 2006.
17. Антонова И., Бачкова И. (2007), Проектиране на Разпределени Системи за Управление на Базата на UML/SysML и Стандарта IEC61499, International Scientific Conference, AMTEX'07, 23-24 November, Gabrovo, Bulgaria, pp. II-41 ÷ II-47.
18. Бачкова И., И. Антонова (2005), Обектно-ориентирано проектиране на децентрализирани системи за управление с използване на UML-MAST, Научна конференция с международно участие "60 години катедра Неорганична химия", 11.11.2005, София.
19. Бачкова И., Г. Попов, Г. Стамболов, И. Антонова (2006), Проектиране на отворени разпределени системи за управление на базата на стандарта IEC 61499, Journal of the Technical University at Plovdiv "Fundamental Sciences and Applications", Vol. 13 (6), pp. 93-102, Anniversary Scientific Conference' 2006, Пловдив.
20. Антонова И., Бачкова И. (2006), Обектно-ориентирано проектиране на системи за управление на базата на UML и Corfu-FBDK, Международна научна конференция "Развитие – бъдещи перспективи и иновации в науката'2006" 12.12.2006, София, стр.5-16.
21. Antonova I., Batchkova I. (2008), Design and analysis of IEC61499 based distributed control systems using UML and Rhapsody, Материали от Национална научно-техническа конференция с международно участие „Автоматизация в минната индустрия и металургията” БУЛКАМК'08, 27-28.11., София, стр.173-178.
22. Antonova I., Batchkova I., Popov G. (2008), Development of IEC61499 based feedback control using UML/SYSML, Proceeding of the International Conference „Automatics and Informatics”, 01-04 October, Sofia, pp. V-11 ÷ V-14.
23. Antonova I., Batchkova I. (2008), Development of Multi-Agent Control Systems using UML/SysML, In Proceedings of the 4-th International IEEE Conference on Intelligent Systems, Vol.1, pp.6-26: 6-31, Varna, Bulgaria, September 6-8.
24. Hoffman H.-P. (2006), Harmony-SE/SysML Deskbook Model-based Systems Engineering with Rhapsody, Rev.1.51, Telelogic/I-Logix whitepaper, Telelogic AB, May 24.