



*Юбилейна научна конференция
„40 ГОДИНИ КАТЕДРА АВТОМАТИЗАЦИЯ
НА ПРОИЗВОДСТВОТО”, 18 март 2011*

Anniversary Scientific Conference with International Participation

40 Years Department of Industrial Automation

PROCEEDINGS

BULGARIA, SOFIA
18.03.2011



INTERNATIONAL PROGRAM COMMITTEE (IPC)

Chairmen: M. Hadjiski

Vice Chairmen: K. Boshnakov

Members:

Batchkova I.	Mladenov M.
Bratengeyer E.	Nenov T.
Velev K.	Nikolov E.
Voutchkov I.	Petkov P.
Vachkov, G.	Petkov M.
Garipov E.	Popchev I.
Damyantov Ch.	Richalet, J.
Djambov P.	Sgurev V.
Elenkov G.	Stoilov T.
Iliev Z.	Stoyanov S.
Yonchev, H.	Uyar E.
King, R.	Frey G.
Kocijan, J.	Fahri O.
Madjarov N.	Tzotchev V.

NATIONAL ORGANIZING COMMITTEE (NOC)

Chairmen: V. Tzotchev

Vice Chairmen: I. Batchkova

Secretary: I. Antonova

Members:

Elenkov G.
Christova N.
Gocheva D.



APPROACHES OF INDUSTRIAL INFORMATICS FOR DEVELOPMENT OF DISTRIBUTED CONTROL SYSTEM

ПОДХОДИ НА ИНДУСТРИАЛНАТА ИНФОРМАТИКА ЗА РАЗРАБОТКА НА РАЗПРЕДЕЛЕНИ СИСТЕМИ ЗА УПРАВЛЕНИЕ

I. Batchkova¹, I. Antonova¹, D. Ivanova²

University of Chemical Technology and Metallurgy, Dept. of Industrial Automation, Bul. Kl. Ohridski 8, Sofia, Bulgaria, {idilia|iskra.antonova}@uctm.edu

Abstract: One of the main trends in the field of automation and control is associated with the development and implementation of distributed control systems which support the achievement of high degree of flexibility, openness, modularity, vendor independence and interoperability. The main purpose of this report is to analyze and summarize the most suitable approaches and methods for development of distributed control systems and the main tasks and problems, connected with their application. Two combined approaches are proposed, illustrated and discussed. Finally some conclusions are made.

Key words: distributed control systems, IEC-61499, UML, Petri nets

ВЪВЕДЕНИЕ

Една от основните тенденции в областта на автоматизацията и управлението е свързана с разработката и внедряването на разпределени системи за управление, състоящи се от множество нейерархично разположени модули, свързани помежду си посредством различни по вид комуникационни системи. Разпределените системи притежават редица предимства в сравнение с централизираните системи по отношение на разходи, гъвкавост, отвореност, модулност, независимост от производителя, оперативната съвместимост, поддръжка. За да бъде управлявана сложността на разпределените системи за управление, трябва да бъдат изпълнени редица изисквания, които най-общо могат да бъдат обобщени по следния начин: надеждна концепция за декомпозиция и постигане на модулност, възможности за разширения в случаите на нови продукти, машини и съоръжения, модел на общата структура на системата, който не трябва да бъде зависим от конкретния производител, както и използване на капсулирани, многократно използвани компоненти.

Функционалността в разпределените системи за управление може да бъде моделирана използвайки различни средства за моделиране на дискретно-събитийни системи, като времеви или хибридни автомати, мрежи на Петри, Grafcet/SFC, различни формални езици или базирани на стандарта IEC 61499 функционални блокове.

Основната цел на доклада е да обобщи най-удачните подходи и методи в областта на разработката на разпределени системи за управление и свързаните с тяхното прилагане задачи и проблеми. В тази връзка и на базата на натрупания в областта опит се предлага използването на комбинирани подходи с оглед на преодоляване на недостатъците на „чистите“ подходи. Предложени са два примерни подхода, базирани на комбинираното използване на предложени в стандарта IEC61499 компонентен подход за разработка на разпределени системи за измерване и управление на процеси с инструментариума на Сигнално

Интерпретиращите Мрежи на Петри (СИМП) предоставящи възможност както за формална верификация на моделите за управление, така също и подпомагаща валидацията на затворената система за управление. В основата на втория подход е унифицирания език за моделиране UML, осигуряващ реализацията на имплементиращ обектно-ориентиран подход, покриващ пълния жизнен цикъл на разработка на разпределени приложения за управление и позволяващ ранно откриване на грешки и несъответствия.

Изследването е представено в 3 части. След въвеждането в част 2 са представени и анализирани накратко основните подходи при разработката на разпределени системи за управление. В част 3 са описани и илюстрирани с конкретни примери накратко двата подхода, В заключение са направени някои изводи.

АНАЛИЗ НА ОСНОВНИТЕ ПОДХОДИ ПРИ РАЗРАБОТКА НА РАЗПРЕДЕЛЕНИ СИСТЕМИ ЗА УПРАВЛЕНИЕ

Обектно-ориентиран подход за разработка на разпределени системи за управление. Обектно-ориентираният подход гарантира три основни свойства на моделираната система и нейните елементи: капсулиране, наследяване и полиморфизъм. Капсулирането позволява разработката на софтуер на модулния принцип, наследствеността подпомага неговото автоматично многократно използване, чрез което се повишава робастността и значително се улесняват процесите на разработка. Полиморфизмът улеснява разработката на независими модули по конструктивен начин. Основните недостатъци на обектно-ориентираното моделиране са свързани с липсата на абстрактност или подпомагане на връзките и взаимодействието между обектите. Сложните приложения обикновено изискват оперативна съвместимост между няколко обекта. При проектирането и създаването на обекти не винаги е известно как и кога да се използва тяхната функционалност и промените в

състоянието им, което може да доведе до неочаквани грешки.

Едни от най-обещаващите обектно-ориентирани подходи са базиран на използване на Унифицирания език за моделиране (UML). Макар че е проектиран специално за разработката на софтуерни системи, UML се разглежда като визуален език за моделиране с общо предназначение, базиран на единни нотации и обектно-ориентиран мета-модел [1]. Едно от основните предимства на езика е възможността за създаване на UML профили за определена приложна област чрез дефинирането на специфична терминология и структури.

Профилът на UML за системно инженерство подпомага процесите на моделиране на широк кръг от системи, които могат да включват хардуер, софтуер, данни, персонал, процедури и съоръжения. Езикът за системно моделиране (SysML) е първият UML профил за системно инженерство, изграден на базата на OMG RFP (Request Object Management Group за подаване на предложение) [2] и представлява унифициран език за системно инженерство за целите на анализ, спецификация, проектиране, и верификация на сложни системи, насочени към подобряване на качеството системите, подобряване на обмена на информация между системите на разнородни инженерни средства, както и съдейства за преодоляването на семантичната празнина между системи, софтуер и други инженерни дисциплини [3].

Проектирането на софтуера в системите за разпределено управление на производствени системи е сложна задача и за да могат да бъдат анализирани и отразени различните аспекти на тези системи е необходимо проектът да включва графичното представяне на различни изгледи на системата. Някои от тези изгледи се отнасят до абстрактни аспекти на проекта, докато други касаят неговата физическа реализация или организацията на софтуера. Тези изгледи не могат да бъдат реализирани с използването на една единствена методология за проектиране. От областта на софтуерното инженерство е широко известно, че за целта са необходими поне четири различни изгледа, както и съвкупност от сценарии, които формират архитектурата, известна като модел на изгледите от типа "4+1" [4]. Изгледите включени в модела са: логически, изглед на процесите, на разработката и физически изглед. При проектирането на разпределени системи логическият изглед отразява функционалните изисквания към системата и включва дефиниране на основните функционални блокове и интерфейсите помежду им. Този изглед се покрива изцяло от стандартите IEC-61131-3 и IEC-61499. Изгледът на процесите представя системата за управление като логическа мрежа от комуникиращи си програми, разпределени на различни хардуерни ресурси и се покрива изцяло от методологията, залегнала в стандарта IEC-61499. Изгледът "разработка" отразява организацията на използваните в системата софтуер, библиотеки и модули, както и съществуващите връзки, и ограничения между тях. Физическият изглед отразява, използваните в системата за управление, физически устройства и контролери и показва разнородните комуникационни връзки между тях. Сценариите отразяват основните взаимодействия между елементите на системата, като всеки един сценарий разглежда взаимодействието между различни функционални части на софтуера. Сценариите могат да показват както аспекти на логическия, така и на физическия изглед. На този етап на развитие стандартът IEC-61499 не поддържа последните три изгледа.

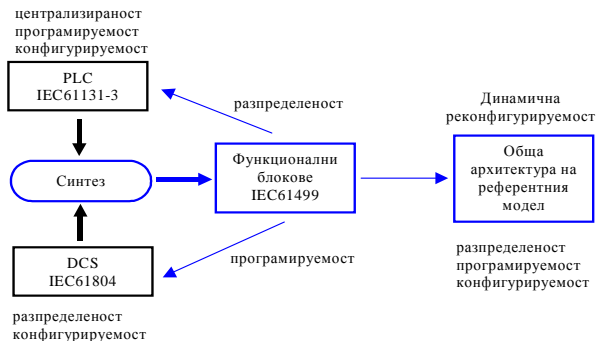
Компонентно-базиран подход за разработка на разпределени системи за управление. Компонентно-базираната разработка прави стъпка напред чрез въвеждането на повече абстрактност по отношение на самоописателните характеристики, които позволяват дефинирането на връзки към други обекти и използването на външна функционалност. Това улеснява процесите на разработка, но за съжаление не допринася значително за постигането на многократно употреба и модификация на интерфейса в реално време, т.е. използваните компоненти са пасивни и могат да бъдат описвани или използвани отвън. Важна стъпка в това направление е новият стандарт за моделиране на разпределени системи за измерване и управление на процеси IEC-61499 [5]. Стандартът е разработен на базата на съществуващите стандарти IEC-61131-3 [6] за програмиране на програмируемите логически контролери (PLC) и IEC-61804 [7] в областта на използваните в непрекъснатите производства разпределени системи за управление (DCS). Основните недостатъци на базираните на стандарт IEC-61131 софтуерни модели за разпределени приложения, могат да бъдат обобщени както следва [8]:

- Липсва ясно дефиниран ред за изпълнение на функционалните блокове;
- Приложенията в базираните на стандарта IEC-61131-3 модел не са разпределими на различни ресурси;
- Липсва необходимата гъвкавост при назначаване на задачи към програми и функционални блокове;
- Начинът на изпълнение на базираните на стандарта IEC-61131-3 функционални блокове не може да бъде използван за свързани функционални блокове, разположени на различни ресурси.

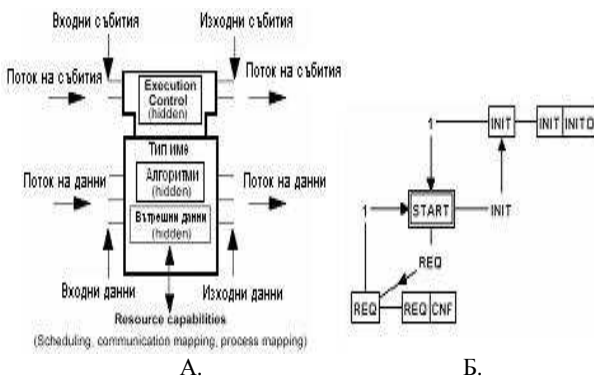
Новият стандарт IEC-61499 има за цел да обедини и доразвие положителните страни на стандартите IEC-61131-3 и IEC-61804 за целите на проектирането и разработката на разпределени системи за управление. Както е показано на Фиг.1, стандартът изхожда от необходимостта за създаването на по-общ модел, който да може да бъде използван както при проектиране на PLC, така и на разпределени системи за управление като гарантира едновременно програмируемост, разпределеност и конфигурируемост на приложенията [9]. Стандартът IEC-61499 дефинира референтна архитектура, основни понятия и модели за разработка на модулни, преизползваеми, компонентно-базиран и отворени разпределени приложения за управление на процеси [5]. Функционалният блок, представен на Фиг.2 може да бъде използван за дефиниране на многократно използваеми софтуерни компоненти, които на базата на предлаганата методология да бъдат използвани за проектиране на сложни децентрализирани системи за управление. Стандартът дефинира три основни типа функционални блока: основен, съставен и интерфейс за услуги. Системите за управление могат да се моделират като логически свързани функционални блокове чрез техните входно-изходни данни и събития.

Основният функционален блок се представя посредством интерфейс на входните и изходни данни и събития (фиг.2-А). Вътрешността му включва "граф за изпълнение на управлението" (Execution Control Chart (ECC)), вътрешни данни и вътрешни алгоритми. ECC е представен на фиг.2-Б и представлява машина на състоянието, която управлява изпълнението на алгоритмите, асоциирани към функционалния блок. Функционалният блок се характеризира чрез име на типа и името на екземпляра, които го идентифицират еднозначно. ECC се състои от

състояния, преходи и действия, които предизвикват изпълнението на алгоритмите, асоциирани към състоянията на ЕСС, в отговор на събитийни входове. Едно от състоянията е т.н. начално състояние, а към другите състояния от изпълнението на управлението могат да бъдат асоциирани едно или няколко действия. Всяко действие може да бъде свързано с изпълнението на алгоритъм или генериране на изходно събитие. Преминването на машината на състоянието от едно състояние на изпълнение на управлението в друго се реализира чрез активирането на преходи.



Фиг.1. Диалектика на архитектурата на функционалните блокове [9]



Фиг.2. Модел на основен функционален блок [5]

Стандартът дефинира 4 вида модели, на базата на които може да бъде проектирана разпределената система за управление: системен модел, модел на устройствата, модел на ресурсите и модел на функционалния блок. IEC-61499 не осигурява програмна среда за реализация на разпределената система за управление, а само определя нейната архитектура и дефинира основните понятия и правила на включените в нея елементи. Основен недостатък на стандарта е, че 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 интероперативност.

Комбиниран подход за разработка на базата на стандарта IEC61499 и UML. Представените по-горе подходи притежават както своите силни страни, така и редица недостатъци, които затрудняват отделни етапи и аспекти на разработката. Нарастват изследванията, предложенията и разработките, които са в посока на интеграцията на възможностите на обектно-ориентирания език за моделиране UML с тези на стандарта IEC-61499 за целите на проектиране на разпределени системи за управление. Едни от най-перспективните идеи в тази насока са свързани с имената на В. Вяткин и Дубинин

[10], Г. Фрай [11], К. Трамболидис [12], и др. Подробен обзор на тези подходи е представен в [13, 14].

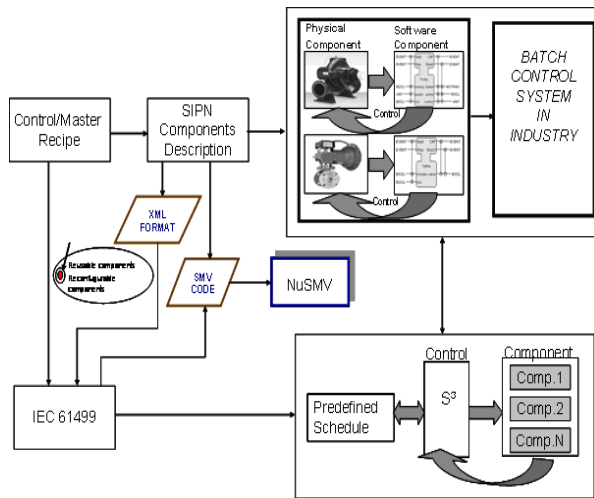
Агентно-базиран подход. Друго перспективно направление за компенсиране на недостатъците на обектно-ориентирания и компонентно-базирания подход е използването на концепцията на мулти-агентните системи чрез прилагане на принципите на агентно-ориентираното софтуерно инженерство (AOSE). Изследователската общност в областта на AOSE изследва различни разширения на UML за подпомагане на процесите на моделиране на основните агентно-базирани концепции, като агенти, онтология и протоколи за взаимодействие. От гледна точка на софтуерното инженерство, агентните системи са специализация на обектно-базираните системи, в които отделните обекти имат свои собствени пътеки на управление и свои собствени цели, или чувство за цел. С развитието на UML и въвеждането на активни обекти, разликите между обектите и агентите намалява, благодарение на подобряване мета-модела на езика и създаването на нови означения и формализацията на различни диаграми, използвани от езика. Това позволява постигането на по-висока степен на покритие на основните свойства на агентите и техните взаимодействия. Освен това основно предимство на използването на UML при разработката на MAS в сравнение с други алтернативи е, че използването на стандартни механизми и спецификации, е от решаващо значение за осигуряване на оперативна съвместимост между автономни единици в една отворена и разпределена агентно-базирана интегрирана среда за управление.

Подходи базирани на формална спецификация и верификация на разпределени системи за управление. Разпределените системи са сложна област, тъй като грешките (дефектите) в програмното осигуряване нарастват експоненциално с нарастване броя на взаимодействащите си компоненти. Статистически анализи показват, че 40 – 50% от софтуерното обезпечаване съдържа нетривиални грешки, които не могат да бъдат отстранени нито в етапа на β -тестване, нито посредством обикновен анализ. Математиката осигурява солидна база от методи за справяне с този проблем чрез спецификация и анализ на системите и използването на формални методи. Тяхната математическа основа дава възможност за по-прецизно последователно и еднозначно специфициране на системите. Нещо повече, могат да бъдат използвани методи за формален анализ за да се провери дали системата притежава желаните свойства. Предимствата от използване на формалните методи в областта на проектиране на разпределени системи за управление могат да се обобщят, както следва:

- Водят до значително съкращаване на времето за разработка на системите за управление и свързаните с тях разходи;
- Създават възможност за преизползване на вече разработените софтуерни модули;
- Задоволяват потребността от високо качествени решения вследствие на възможностите за формална верификация на модела за доказване на специфични статични и динамични свойства на системите, като например жизненост, недвусмисленост или време на отговор на контролера, преди да бъдат имплементирани.

ПРИМЕРНИ КОМБИНИРАНИ ПОДХОДИ ЗА РАЗРАБОТКА НА РАЗПРЕДЕЛЕНИ ПРИЛОЖЕНИЯ ЗА УПРАВЛЕНИЕ

Приложение на комбиниран компонентно-базиран подход. Предлаганият подход касае разработката на общи функционални компоненти за разпределените системи за управление и е илюстриран на фиг.3. Подходът се базира на комбинираното използване на стандарта ISA SP88 (на високо ниво) и стандарта IEC 61499 (на ниско ниво). Основната идея е да се използват знанията на технологиите по отношение на управляваните процеси и системи с използване на стандарта ISA SP88, като в същото време на базата на стандарта IEC-61499 съществува възможност за проектиране и имплементация на системата за управление. Целта е да се осигурят многократно използвани компоненти и лесен начин за реконфигуриране на тяхното изпълнение. В резултат на това се постига гъвкавост в стратегията за управление.



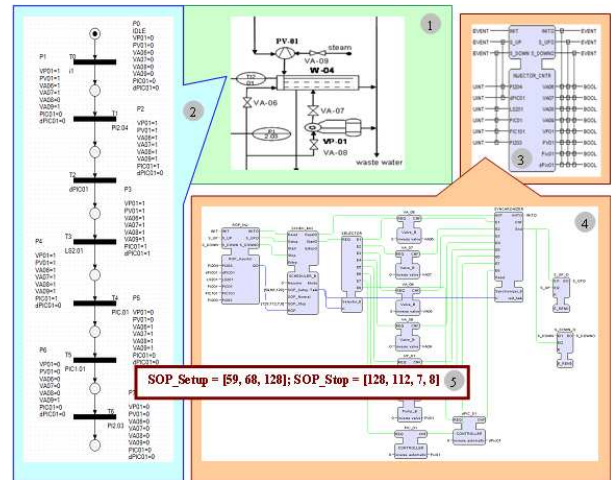
Фиг.3: Илюстрация на комбиниран компонентно-базиран подход

На базата на този подход, след като са изградени основните функционални компоненти, техните управляващи функции могат да бъдат използвани многократно за компоненти с подобна функционалност. По този начин значително се редуцира времето за разработка. Предложеният подход е разделен на три вида дейности: разработка на функционални компоненти, моделиране на управляващата рецепта с използване на СИМП и трансформацията на модела в базирано на IEC-61499 приложение. На фиг.4 е представено приложение на подхода при разработката на разпределена система за управление на ежекторната система на инсталация за периодична ректификация на сулфатен терпентин.

Приложение на комбиниран обектно-ориентиран подход. Подходът, който е използван за целите на разработката, се базира на използване на профила на UML за областта на системното инженерство SysML, който преизползва част от нотациите на UML, а също така и дефинира допълнителни диаграми, нотации и стереотипи. Като основни изграждащи конструкции са дефинирани различни стереотипи като: стереотип на основен функционален блок, стереотип на интерфейс на услуги, стереотип на ресурс и др.

UML2.x и SysML предлагат разнообразни възможности за моделиране на различните изгледи на проектираната система и представяне на разнообразни сценарии на различни нива на абстрактност на проекта. Предложеният

подход използва методологията Harmony-SE която представлява част от методологията за разработка интегрирани системи и софтуер [15]. Жизненият цикъл на разработката включва следните основни етапи (фиг.5): анализ на изискванията; функционален анализ на системата; проектиране на архитектурата; и хардуерно / софтуерно специфициране на проекта.

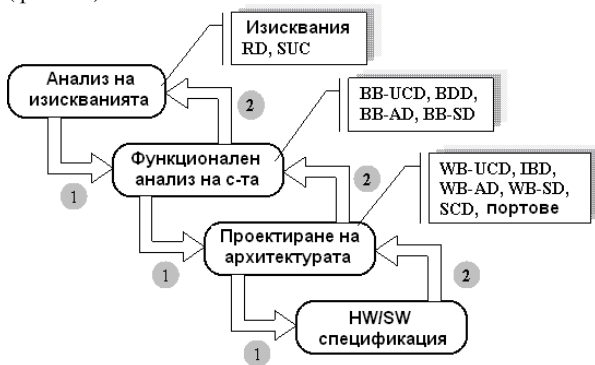


Фиг.4: Разработка на управлението на ежекторната система

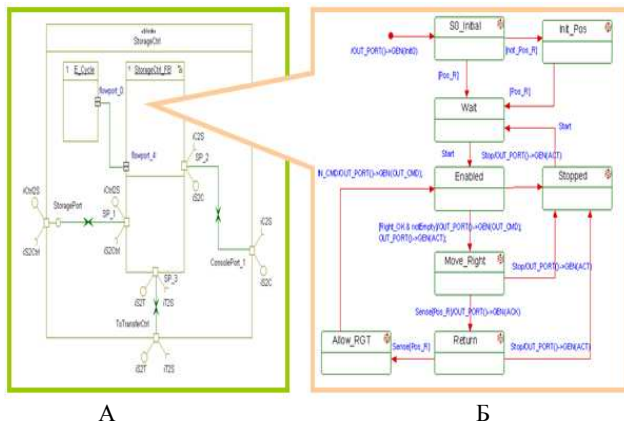
Структурните диаграми и блоковете са основни конструктивни елементи на разработваните модели. Фазата на анализ на изискванията стартира с анализ на входовете. Изискванията на клиентите се превръщат в набор от изисквания, като за целта се използват диаграми на изискванията (RD) и диаграми на случаите на използване (UCD). Фазата на функционалния анализ на системата представлява трансформация на дефинираните изисквания в съгласувана функционалност. В рамките на тази фаза се анализират всички случаи на използване, като за целта се използват диаграми от типа „дефинирани на блокове” (BDD) и „вътрешни блокове” (IBD). За отразяване на функционалността на системата на високо ниво се използват диаграми на последователност и функционални диаграми от типа „черна кутия” (BB-SD, BB-AD). Проектирането на архитектурата е третият етап, който включва проектиране на структурите и поведението на подсистемите, основаващо се на използване на по-горе изброените диаграми, но на този етап от тип „бяла кутия” (WB-UCD, WB-AD, WB-SD). Друга важна задача е дефинирането на портове и връзки и описание на поведението на подсистемите на базата на състоянията с използване на диаграми на състоянията (BCC). Последният етап е хардуерна/софтуерна спецификация на проекта и е тясно свързан с имплементацията и внедряването на системата. Всички етапи на методологията включват задачи по верификация и валидация, въз основа на моделите, разработени в предходните етапи и на вече дефинираните изисквания.

На фиг.6 е представен пример на използване на описания подход за целите на изграждане на разпределена система за управление на пилотна разпределителна станция на FESTO. Станцията е декомпозирана в два модула „Storage” и „Transfer”. Двата мехатронни модула се управляват отделно от разпределени контролери, които комуникират помежду си. Разпределените контролери "StorageCtrl" и "TransferCtrl" са моделирани като "системни блокове", като първият от тях се състои от две основни "части": "StorageCtrl_FB" и "E_Restart", както е видно от фиг.6-A. Вътрешната структура на контролера е моделирана с използване на диграма от „вътрешни

блокове” (IBD), а поведението на отделните функционални блокове – чрез диаграми на състоянието (фиг.6-Б).



Фиг.5: Илюстрация на комбиниран обектно-ориентиран подход



Фиг.6: Дефиниране на основен функционален блок в среда на UML2.x /SysML

ЗАКЛЮЧЕНИЕ

Индустрията се насочва все повече към използване на системи от разпределени интелигентни устройства, които комуникират посредством цифрови потоци от данни. Тези системи изместват мониторинга и управлението все по-плътнo към производствения цикъл, адаптират и непрекъснато повишават ефективността, като в същото време съкращават оперативните и инсталационни разходи както и времето, и разходите за поддръжка. В тази връзка предлаганите подходи биха намерили своето подходящо място, както в посока на създаване на стандартизирани библиотеки от компоненти, така и в посока на разработката и използване на инженерни среди за проектиране и анализ на разпределени системи за управление.

Съществуват достатъчно предизвикателства към усъвършенстване на представените подходи по отношение на етапите на разработка, степента на формализация, методите за симулация и верификация на решенията, както и създаването на специален UML2.x /SysML профил за разработка на разпределени системи за управление.

БЛАГОДАРНОСТИ

Авторите благодарят за финансовата подкрепа по проект ВУИ-307 с НФНИ.

ЛИТЕРАТУРА

1. OMG-UML (2010). Unified Modelling Language Specification, June: <http://www.omg.org/>
2. OMG-UML (2003). UML for Systems Engineering, Request for Proposals - RFP ad/2003-3-41, Object Management Group, Needham, September.
3. OMG-SysML (2006). The OMG Systems Modelling Language, May <http://omgsysml.org/index.htm>
4. P. Kruchten. The 4+1 View of Architecture. IEEE Software 12(6) November 1995 pp.45-50.
5. IEC-61499 (2005). International Standard IEC-61499, Function Blocks, Part 1 - Part 4, International Electrotechnical Commission (IEC), Technical Committee TC65/WG6, IEC Press, Jan.
6. John K.H., Tiegelkamp M., IEC 61131-3: Programming Industrial Automation Systems, Springer-Verlag, Berlin-Heidelberg-New York, 2001.
7. IEC-61804 IEC/TS 61804-1. Function blocks (FB) for process control - Part 1: Overview of system aspects, Ed. 1.0, 2003.
8. Lewis R., Modelling control systems using IEC 61499, The Institution of Electrical Engineers London, United Kingdom, 2001.
9. J. H. Christensen. Basic concepts of IEC 61499. Proc. of Conference "Verteile Automatisierung" (Distributed Automation), pp. 55-62, Magdeburg, Germany, 2000.
10. Dubinin, V. & Vyatkin, V. (2004). UML-FB – A Language for Modelling and Implementation of Industrial-Process Measurement and Control Systems on the Basis of IEC 61499 Standard, Proc. of the 6-th International Conference of Science and Technology "New Information Technologies and Systems (NITS'2004)", pp.77-83, Penza, Russia, Part 2, June 17-19.
11. Panjaitan S. D., Frey G. (2007), Development process for distributed automation systems combining UML and IEC61499, Int. Journal of Manufacturing Research, Inderscience Publisher, Vol.2, No1, pp.1-20.
12. K. Tramboulidis (2001), Using UML for the Development of Distributed Industrial Process Measurement and Control Systems. Proc. of the IEEE Conference on Control Applications (CCA), September, Mexico.
13. Antonova I., Batchkova I. (2010), Development of Multi-Agent Control Systems using UML/SysML, book chapter, InTech, ISBN: 978-953-307-568-6
14. Batchkova I., I. Antonova (2011), Improving the software development life cycle in process control using UML/SysML, 18th IFAC World Congress, Milan, Italy, 28. 08-02.09, accepted.
15. Hoffman, H.-P. (2001). Harmony-SE/SysML Deskbook Model-based Systems Engineering with Rhapsody, Rev.1.51, Telelogic/I-Logix whitepaper, May 24.