



СЪЮЗ ПО АВТОМАТИКА И ИНФОРМАТИКА
„ДЖОН АТАНАСОВ”



ФЕДЕРАЦИЯ
НА НАУЧНО-ТЕХНИЧЕСКИТЕ СЪЮЗИ

XX Юбилеен Международен симпозиум

**УПРАВЛЕНИЕ
НА ТОПЛОЕНЕРГИЙНИ
ОБЕКТИ И СИСТЕМИ**

8 – 9 ноември 2012 г.

Банкя

СБОРНИК ДОКЛАДИ

XX Юбилеен Международен симпозиум

**УПРАВЛЕНИЕ
НА ТОПЛОЕНЕРГИЙНИ ОБЕКТИ
И СИСТЕМИ**

СЪОРГАНИЗАТОРИ И СПОНСОРИ

METSO
ХАНИУЕЛ ЕООД
РИТБУЛ ЕООД
РИТАЛ ЕООД
ОСКАР-ЕЛ ЕООД
АМК ЕООД
ЕТ ТРАПЕН

ПРОГРАМЕН КОМИТЕТ

М. Хаджийски – председател
Е. Николов
Б. Бонев
И. Бачкова
К. Бошнаков
В. Петков
Т. Готев
Тр. Пензов

ОРГАНИЗАЦИОНЕН КОМИТЕТ

М. Николов – председател
Б. Ванев
В. Андреев
Н. Христова
Л. Дуковска
Ю. Божков
З. Георгиев
М. Божкова



ISSN 1313-2237

ФОРМАЛНА СПЕЦИФИКАЦИЯ И ИМПЛЕМЕНТАЦИЯ НА СИСТЕМИ ЗА УПРАВЛЕНИЕ С ИЗПОЛЗВАНЕ НА ВРЕМЕВИ ВХОДНО/ИЗХОДНИ АВТОМАТИ

FORMAL SPECIFICATION AND IMPLEMENTATION OF CONTROL SYSTEMS USING TIMED INPUT/OUTPUT AUTOMATA

П. Дойчева¹, М. Шмид², И. Бачкова¹

¹Химикотехнологичен и Металургичен университет - София, катедра „Автоматизация на производството”, idilia@uctm.edu

²Рурски Университет – Бохум, катедра „Автоматизация и процесна информатика“, schmidt@atp.rub.de

Abstract: The formal specification and verification is an important prerequisite for achieving the key features of control system software, such as functionality, reliability, usability, efficiency, portability and maintainability. The main goal of the report is to investigate and present an approach for development of control systems using timed I/O automata. The proposed approach is illustrated by a case study using Matlab/Simulink.

Keywords: formal specification, control systems, timed input-output automata, modeling, implementation

ВЪВЕДЕНИЕ

В съвременното гъвкаво и ориентирано към клиента автоматизирано производство разработката на софтуер за управление е съществен фактор, който изисква значително време и финансови средства. Нещо повече, все по-често се налага алгоритмите за управление да бъдат адаптирани към нови сценарии на производство. Нарастващата сложност на съвременния софтуер и повишаващите се изисквания на потребителя по отношение на сигурност и функционални възможности налага използването на нови подходи при разработката на софтуер в системите за управление. Съгласно стандарта ANSI/IEEE 610, софтуерът за управление на процеси е приложен софтуер и като такъв неговите качества съгласно стандарта ISO/IEC 9126 могат да бъдат оценени на базата на 6 ключови характеристики: функционалност, сигурност, използваемост, ефективност, преносимост и възможност за поддръжка [1]. Важна роля за постигане на тези характеристики оказва използването на формални методи при тяхната разработка. Системите за управление могат да бъдат формално описани с различни дискретно-събитийни модели като: различни модификации и разширения на времеви автомати [2], мрежи на Петри [3], Grafset/SFC [4], формални езици или мрежи от функционални блокове, базирани на стандартите IEC-61131 [5] и IEC-61499 [6].

Цел на доклада е да изследва и предложи подход за формална спецификация и верификация на системи за управление с използване на метода на времеви входно-изходни автомати (времеви I/O автомат), позволяващи моделиране на поведението на елементите на системата за управление, както и тяхната формална верификация. Предложеният подход е приложен за имплементацията на управляем дискретно събитийен процес в системата от три резервоара, намираща се в катедрата „Автоматизация и процесна информатика“ на Рурския Университет в Бохум. Трите резервоара преминават през зададена последователност от нива, симулирайки определен производствен процес. Имплементацията на контролера в средата на MATLAB/Simulink.

Докладът е систематизиран в 5 части. След въведението, в част 2 са представени накратко времевите I/O автомати. Част 3 е предложен подход за формална спецификация и имплементация на системи за управление на базата на времевите I/O автомати. В част 4 е представена илюстрация на подхода за спомената по-горе система от три резервоара. Накрая са направени някои изводи относно предимствата и недостатъците на предложения подход и резултатите от неговото тестване.

КРАТКО ВЪВЕДЕНИЕ ВЪВ ВРЕМЕВИТЕ ВХОДНО-ИЗХОДНИ АВТОМАТИ

Недетерминираният времеви I/O автомат A_t е система, представляваща наредена последователност от пет елемента [7].

$$A_t = (Z, V, W, L_t, z_0) \quad (1)$$

където:

Z - множеството на състоянията

V - множеството на входните символи

W - множество на изходните символи

L_t - характеристична функция

z_0 - начално състояние

Динамиката на I/O автомат се представя чрез характеристичната функция (2, 3), която приема стойност единица, ако в сила $(z', w, z, v, \tau)!$, което означава, че преходът е дефиниран и системата преминава от състояние z с вход v , достигнато за време t в състояние z' като генерира изхода w за време $(t + \tau)$. Престоят в състоянието може да се дефинира не само с определена стойност, а и чрез времеви интервал $[T_{min}, T_{max}]$.

$$L_t = Z \times W \times Z \times V \times \mathbb{R} \rightarrow \{0, 1\} \quad (2)$$

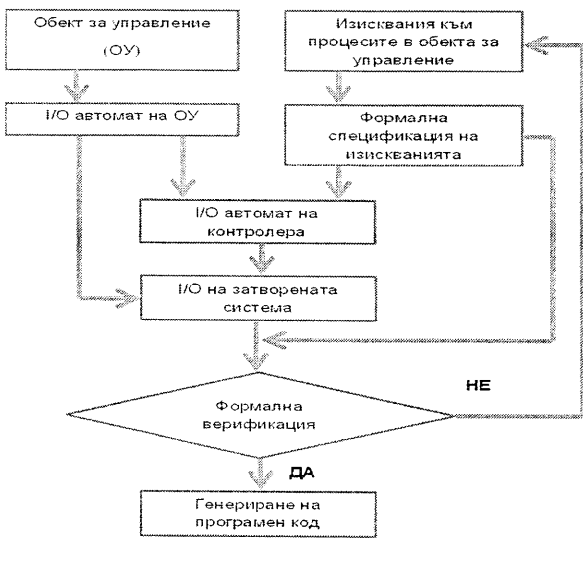
$$L_t = (z', w, z, v, \tau) = \begin{cases} 1, & \text{if } (z', w, z, v, \tau) \\ 0, & \text{else} \end{cases} \quad (3)$$

Формализацията на базата на I/O автомати може да бъде използвана за моделиране като система от преходи както на управляемата система, така също и на нейните

елементи - обекта за управление и контролера. Важно предимство на този подход е възможността за моделиране на реално-времеви, хибридни и стохастични аспекти на системите за управление, както и за подпомагане на процеса на ранна верификация на управляващия алгоритъм и поведението на управляваната система.

ОПИСАНИЕ НА ПОДХОДА

На фиг.1 е представена графична илюстрация на предлагания подход, в чиято основа са 3 вида формални модели: на обекта за управление, на контролера и на затворената система. Формалната верификация се базира на модела на затворената система и формалната спецификация на изискванията към системата (желаното поведение). При успешна верификация на модела следва автоматичното генериране на код.



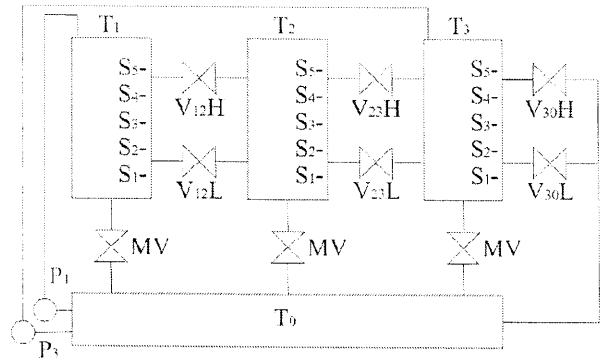
Фиг.1: Илюстрация на предлагания подход

ПРИЛОЖЕНИЕ НА ПОДХОДА

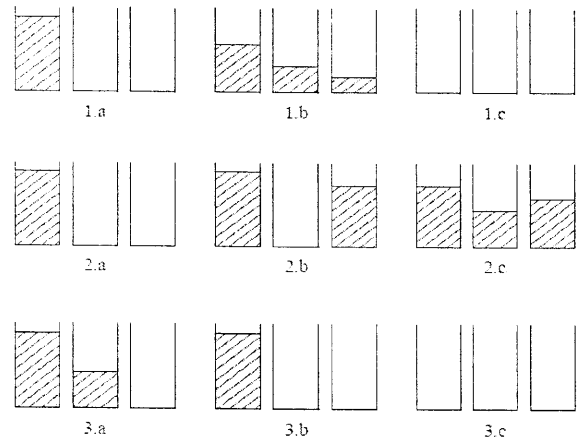
Описание на обекта за управление. Обектът за управление представлява система от три основни резервоара (T_1 , T_2 и T_3) и един спомагателен резервоар T_0 , девет вентила и две помпи (фиг.2). Спомагателният резервоар съдържа водата за пълнене на основните резервоари посредством помпите P_1 и P_2 , захранващи резервоари T_1 и T_3 , респективно. Основните три резервоара са свързани с автоматични вентили на ниско и високо ниво. Към всеки от основните резервоариса монтирани по пет дискретни сензора за ниво, означени с S_i , където $i=1,2,\dots,5$, които са номерирани във възходящ ред. Автоматичните вентили са обозначени с $V_{km}H$ или $V_{km}L$, където k и m са номерата на резервоарите, между които се намира съответния вентил, H показва, че вентилът е монтиран на високо ниво, а L - на ниско ниво. С MV са означени ръчно управляваните вентили в системата. Те могат да бъдат използвани за симулиране на теч в системата и свързват основните резервоари със спомагателния.

Дефиниране на изискванията към системата за управление. В системата трябва да се реализира процес, работещ в три режима - "Почистване", "Производство" и "Край", всеки от тях включващ три фази. Задачата в режим

"почистване" е резервоар T_1 да бъде напълнен догоре, след което водата трябва да премине през другите два резервоара и системата трябва да бъде изпразнена. В режим "производство" резервоар T_1 трябва да бъде напълнен догоре с вещество "А", а резервоар T_3 трябва да бъде напълнен $\frac{3}{4}$ -та с вещество "Б", след което двете течности трябва да бъдат смесени в резервоар T_2 . Тук е прието, че резервоари T_1 и T_3 се пълнят с две различни вещества, а тяхната смес е продуктът на процеса. В последната част, резервоарът T_3 трябва да бъде изпразнен, продуктът от резервоар T_2 трябва да бъде бутилиран, а остатъците от инсталацията трябва да бъдат източени. На Фиг.3 процесът е изобразен графично по режими и фази.



Фиг.2: Схема на системата от три резервоара



Фиг.3: Илюстрация на процеса

Модел на обекта за управление. Моделът на управляваната система е представен в две форми - с характеристичната функция от Табл.4. и чрез диаграмата на преходите на състоянията на I/Oавтомат, представена на Фиг.4. Моделът показва всяка фаза от процеса, като състоянията на модела са представени чрез нивата в основните резервоари, входовете представляват сигналите, определящи положението на регулиращите органи, а изходите представляват най-високите достигнати нива във всеки резервоар за всяка фаза на процеса. Множеството на състоянията е представено в Табл.1 и представлява множеството на всички стойности, на състоянията на обекта за управление. В общия случай той може да премине през 216 различни състояния, тъй като във всеки резервоар течността може да бъде на 6 различни нива. Съгласно дефинираните изисквания към управляемата система, обаче, тя може да приема само през 14

позволените състояния. Множеството на входните символи е множеството от всички стойности, които могат да заемат входовете, а това са всички възможни комбинации от положенията на регулиращите органи. За конкретните изисквания тези комбинации са 7 и са показани в Табл.2. Множеството на изходните символи (Табл.3) е множеството от всички стойности, които приемат изходите. В общия случай обектът за управление може да има 125 различни изходни символа, тъй като всеки от трите резервоара има по 5 бинарни сензора за ниво. Наложените изисквания върху управляемата система ограничават възможните изходни символи до 11 вектора, всеки с по 3 елемента. Стойността на елементите на вектора показва номера на най-високия сензор, който е достигнат от течността във всеки резервоар. В някои случаи промяната на състоянието зависи не само от входа, но и от времеви интервал, който е известен и се добавя към съответния преход. В случаите, когато времето не оказва влияние върху прехода, времеви интервал се дефинира като $\tau^* = [0; \infty]$.

Диаграмата на I/O автомата (фиг.4) показва преходите между състоянията. Всеки преход е означен с v/w, което означава, че преходът от текущо състояние z към следващото състояние z' се осъществява с вход v и изход w. За някои от преходите е използвано време, съгласно Табл.4. Състоянията в диаграмата са означени с техните номера, а съответстващите им вектори са добавени за по-добра илюстрация. Входовете и изходите са представени с техните кодирани стойности. Диаграмата показва, че процесът образува цикъл, при който началното състояние съвпада с последното състояние на процеса. Преходът между състояния 5 и 6, например, показва, че необходимия вход е v=4, което означава отваряне на вентилите на ниско ниво на трите резервоара. Генерираният изход има стойност w=2, а съответният вектор е [1,0,0], което показва, че само сензорът S₁ в резервоар T₁ е достигнат от течността. Този преход представя изпразването на резервоар T₁, а времето, необходимо за пълното източване на течността е добавено към диаграмата ($\tau=207s$).

Табл.1: Кодирани на състоянията

z	[l ₁ ; l ₂ ; l ₃]
1	[0; 0; 0]
2	[5; 0; 0]
3	[3; 2; 0]
4	[2; 1; 1]
5	[1; 0; 0]
6	[0; 0; 0]
7	[5; 0; 0]
8	[5; 0; 4]
9	[4; 1; 3]
10	[4; 1; 1]
11	[4; 2; 0]
12	[4; 1; 0]
13	[4; 0; 0]
14	[1; 0; 0]

v	входен сигнал
1	включи P1
2	отвори V12L
3	отвори V12L и V23L
4	отвори V12L, V23L и V30L
5	включи P3
6	отвори V23L и V30L
7	отвори V30L

2	[1; 0; 0]
3	[2; 1; 1]
4	[3; 2; 0]
5	[4; 0; 0]
6	[4; 1; 0]
7	[4; 2; 0]
8	[4; 2; 1]
9	[4; 2; 3]
10	[5; 0; 0]
11	[5; 0; 4]

Табл.4: Характеристична функция

z ₀	w	z	v	τ^*
2	1	1	1	[0;1]
3	10	2	2	[0;1]
4	4	3	3	[0;1]
5	3	4	4	[0;1]
6	2	5	4	207s
7	1	6	1	[0;1]
8	10	7	5	[0;1]
9	11	8	3	[0;1]
10	9	9	7	[0;1]
11	8	10	7	49s
12	7	11	6	[0;1]
13	6	12	6	110s
14	5	13	4	[0;1]
1	2	14	4	210s
-	1	1	-	[0;1]

Табл.2: Кодирани на входовете

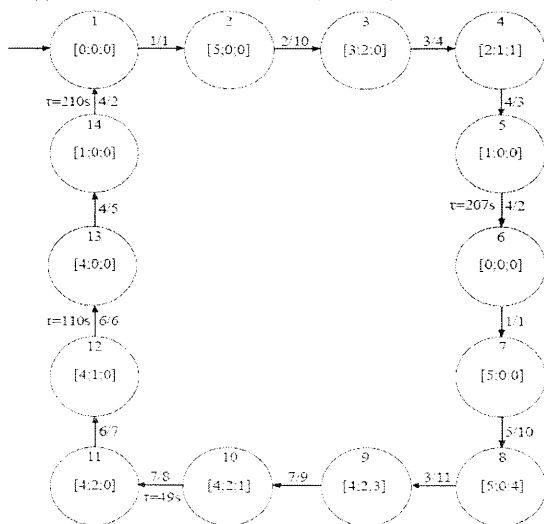


Табл.3: Кодирани на изходите

w	[S _{ij} ; S _{ij} ; S _{ij}]
1	[0; 0; 0]

Модел на управлението. Управляващият алгоритъм е реализиран от функция, която свързва входовете и изходите на контролера. Входовете на функцията са времето t , входният вектор u и променливата z_{Cur} , а изходите са изходният вектор y и променливата z_{Next} .

Реализация на системата за управление. Елементите на автоматизираната система са: компютър, контролер, полева шина PROFIBUS, преобразувател, регулиращи органи и сензори и са описани накратко по-долу. Структурата на системата е показана на фиг.5.

- Компютър, на който, в среда на Simulink, се намира интерфейсът, за връзка с контролера. На него е имплементиран управляващия алгоритъм.

- Контролер - предава сигналите между компютъра и системата от три резервоара. Той е директно свързан към компютъра и косвено свързан към обекта за управление и преобразувателя, чрез комуникационна шина. Контролерът е използван само да предава сигналите от компютъра към инсталацията. Управляващият алгоритъм е инсталиран на компютъра.

Фиг.4: Диаграма на преходите на състоянията

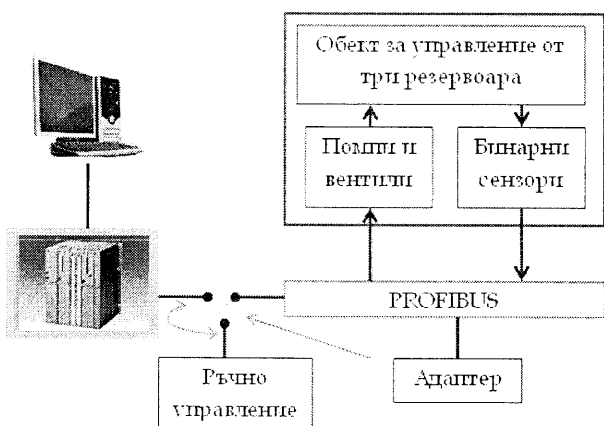
- Полева шина PROFIBUS - свързва контролера, системата от три резервоара и адаптера. Тя преобразува

сигналите, предавани между контролера и обекта за управление.

- Адаптер - осигурява захранване за регулиращите органи.
- Панел за ръчно управление - служи за ръчно управление на помпите. Когато този панел е превключен на ръчен режим, управлението от компютъра се прекъсва.
- Помпите и вентилите се управляват от контролера като получават входните си сигнали от него. Сензорите генерират изходите на системата от три резервоара, които респективно са входове за контролера.

Програмата, използвана за имплементацията на контролера е разработена в среда на MATLAB/Simulink. Тя осъществява връзка с инсталацията и извършва конвертирането на сигналите от обекта. Блоковете, включени в Simulink диаграмата са следните:

- „Source“ блок, който представя системата от три резервоара и преобразува сигналите от сензорите в булеви стойности. Тези сигнали преминават през „Bus“ блок и могат да бъдат визуализирани посредством „Scope“ блокове.
- „Bus“ блок е свързан към „Score“ блокове
- „Constant“ блокове, съответстват на регулиращите органи, като промяната на булевата стойност на тези блокове, превключва статуса на съответния регулиращ орган.
- „Sink“ блок, преобразува булевите стойности от „Constant“ блоковете в сигнали за регулиращите органи.
- „Scope“ блокове изобразяват сигналите от сензорите.



Фиг.5: Структура на автоматизираната система

Крайният вид на блок диаграмата в Simulink е показана на Фиг.6. Добавени са блокове, използващи управляващия алгоритъм с цел изпълняване на дадената задача. Всеки добавен блок е описан по-долу:

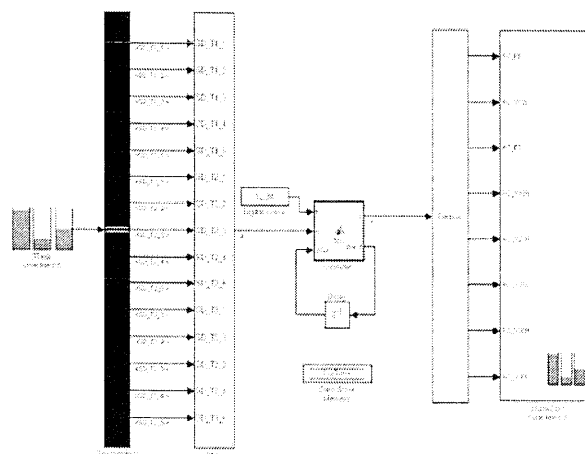
- “Controller” блок представя контролера, който е имплементиран под формата на MATLAB функция. Входовете му са:
 - u - входен вектор. Неговите стойности се зареждат директно от системата и представят сигналите от дискретните сензори.
 - t - текущо време на експеримента.
 - zCur - променлива, която представя стойността на текущото състояние. Изходите са:

- u - изходен вектор- неговите стойности управляват PO на системата.
- zNext - променлива, която представя стойността на следващото състояние.
- “Delay” съхранява стойността на zNext и осъществява необходимата връзка между zCur и zNext.
- Блокът “Mux” свързва всички изходи на системата за улеснено използване в блока “Controller”, като елементи на входния вектор u.
- Блокът “Demux” за разлика от блока “Mux”, разделя елементите на изходния вектор u от блока “Controller” за изпращане към съответните помпи и вентили.
- Блокът “Digital Clock” осигурява текущото време на експеримента.
- Блокът “Data Store Memory” е необходим за глобалната променлива CurTime, дефинирана във функцията, тъй като тя изисква собствена памет.

ЗАКЛЮЧЕНИЕ

Докладът представя подход за формална спецификация и имплементация на системи за управление с използване на времеви I/O автомати. Подходът е илюстриран с подходящ пример. На този етап на разработка проверката на модела за управление е извършена на базата на проследяване на реакцията на системата при внасяне на смущения, чрез отваряне на един или всички ръчни вентили, като техният дебит може да се променя, така че да се наблюдават различни реални случаи. При симулацията са възможни следните случаи:

- Когато се симулира състояние, което е обхванато от модела, процесът се изпълнява съгласно предварително дефинираните изисквания.
- В случаите когато системата попадне в състояние неохванато от модела, процесът нарушава изискванията.



Фиг.6: Блокова диаграма за управление на процеса

Тези резултати налагат извода, че за да се избегнат случаите от втората ситуация е необходимо разработката и прилагането на стратегия за реконфигурация на контролера в реално време при появата на неизправности в обекта за управление. Независимо от този недостатък, използването на I/O автомати за формална спецификация и имплементация на системи за управление притежава и

безспорни предимства, които могат да бъдат обобщени, както следва:

- Използването на времеви I/O автомати подпомага процесите на формална верификация на разработките, чрез което се гарантират едни от основните изисквания към софтуера за управление, като: функционалност сигурности преносимост.
- Създадената чрез подхода възможност за многократно автоматична имплементация на разработения софтуер за управление гарантира ефективността и лесната поддръжка на софтуера.
- Осигурява прозрачност на контролера, която включва възможността за лесно и ясно проследяване на поведението на контролера в момента или в близко бъдеще, както и реинтерпретирането на алгоритъма.

ЛИТЕРАТУРА

1. Frey G., 2000, Automatic Implementation of Petri Net based Control Algorithms on PLC, Proceedings of the American Control Conference, ACC 2000, Chicago (IL), June 2000.
2. Alur R., D. Dill, A Theory of Timed Automata, 17th International Colloquium on Automata, Languages, and Programming, ICALP. LNCS 443, 1990.
3. Peterson J. L. (1977), Petri Nets, Computing Surveys, vol.9, No 3, September, pp.224-252.
4. Lhoste P., Panetto H., Roesch M. (1993), GRAFCET: from syntax to semantics, APII-AFCET/CNRS, Ed. Hermes, Vol. 27- No1, pp 127-141.
5. John K.H., M. Tiegelkamp, IEC-61131-3: Programming Industrial Automation Systems, Springer-Verlag, Berlin-Heidelberg-New York, 2001.
6. IEC-61499-1, Function Blocks for Industrial-Process Measurement and Control Systems – Part 1: Architecture, 2005.
7. Lunze, J.(2006) Ereignisdiskrete Systeme. München Wien: Oldenbourg Verlag, ISBN 3-486-58071-X.