

## Метод за автоматизирано програмиране на работи Изисквания към метода

Иван Станев

*Method for Automated Programming of Robots – Method Requirements: Three major trends in CASE systems are recognized – standardization, flexibility, and automation. Robot's features List is prepared. Robot Programming Tools features List is established. Advantages and disadvantages of the robot programming tools are identified. MAP features are defined.*

*Key words: Software Engineering, Automated Programming, CASE, Service Robots, Robot Programming.*

Четиридесет години след провеждане на първата конференция по софтуерно инженерство, разглеждайки дискутираните там теми ([1]) може да се приеме, че този клон на компютърната наука е един от силно влияещите на нейното развитие.

Изминатият през тези години път е съпроводен с по-големи или по-малки кризи ([2]). Като най-сериозна може да се посочи кризата, обхванала софтуерната индустрия в периода 1965-1985г. В наши дни също се появяват първи признаци на криза. Един от знаците за това е появата на статии от водещи в областта специалисти, като примерно тази на Том деМарко ([3]), озаглавена „Софтуерното инженерство: Една идея, чието време е дошло и е отминало?“. Все по-често в професионалната общност се дискутират въпроси, свързани с ефективността на програмирането и тестването, ефективността на управлението и контрола на софтуерния процес, намаляването на грешките при работа, както и възможността за многократно употреба на вече използвани в създаваните продукти компоненти.

За да се подобри ефективността на разработване на софтуерни продукти е необходимо при създаването на нови технологии и инструменти за разработка на софтуер да се търси подобряване на: степената на стандартизация на използваните софтуерни компоненти и архитектури, за да се подобри възможността за многократната им употреба в различни технически решения; степената на гъвкавост на създадените модели и разработените на тяхна база продукти, с цел да се увеличи живота на продуктите чрез съкращаване на времето за тяхното реконструиране и подобряване на качеството на реконструкциите; степената на автоматизация на процеса на разработка на софтуер, за да се увеличи производителността, чрез намаляване на големия обем ръчен и непродуктивен програмистки труд и да се подобри качеството на продуктите, като се доведе до възможния чрез автоматизацията на процеса минимум на допускане на грешки.

Като се има предвид, че техниките за програмиране на информационни системи и на работи се развиват по приблизително еднакъв начин и като се разгледа задоволително пълна извадка от работи и инструментални среди за програмиране на работи (ИСПР) (примерно [4], включваща 10 работи и 9 ИСПР), може да се каже, че при програмирането на работи отново стандартизацията, гъвкавостта и автоматизацията са основните направления, в които трябва да се развива процеса. За да бъдат определени характеристиките на метода за автоматизирано програмиране на работи (МАП) тук ще бъдат разгледани задачите, които решават роботите и характеристиките, които притежават ИСПР.

### **Задачи, решавани от роботите**

Разгледаните 10 работи (виж Таблица 1) са разположени в 5 групи, всяка от които притежава характеристиките, показани за съответната група в Таблица 2.

Групата работи с най-слаби възможности е разположена в края на двете таблици (работи 1 и 2 от Таблица 1 и характеристика 21 от Таблица 2). По правило по-ниска група не притежава характеристиките на по-високите.

Таблица 1 Роботи – типове и производители

No	Тип робот	Късо име на работ	Име на работ	Късо име на производител	Производител
10	Социален робот	Leo	Social robot Leo	MIT	Massachusetts Institute of Technology Boston
9	Обучаем робот	SR	Service robot	Ski	Skiligent LLC
8	Обучаем робот	Dexter	Dexter robot	UMASS	University of Massachusetts Amherst
7	Автономен - интелигентен	MULE	MULE (FCS)	LM	Lockheed Martin
6	Автономен - интелигентен	UGV	Unmanned Ground Vehicle (FCS)	GD	General Dynamics
5	Автономен - интелигентен	Stanley	Stanley	SVID	Stanford-Volkswagen-Intel-MohrDavidow
4	Автономен	IMR	Warehouse robots	KIVA	KIVA Systems
3	Автономен	IMR	Indoor mobile robots	MR	MobileRobots Inc
2	Дистанционно управляем	MAARS	Armed Talon MAARS robot	F-M	Foster-Miller
1	Дистанционно управляем	HTPR	Healthcare tele-presence robot	ITT	InTouch Technologies Inc

Таблица 2 Роботи - характеристики

Техника на спецификация на заданието	No	Име работ	Производител	10	9	8	7	6	5	4	3	2	1
				Leo	Ski	UMASS	LM	GD	SVID	KIVA	MR	MAARS	HTPR
Работят с указания как да научат какво да правят	No	Характеристика на работата											
	1	Прогнозиране на намерения	Да	Не	Не	Не	Не	Не	Не	Не	Не	Не	Не
	2	Диалог с оператора	Да	Не	Не	Не	Не	Не	Не	Не	Не	Не	Не
	3	Разпознаване на свойствата на обекта	В	Да	Да	Не	Не	Не	Не	Не	Не	Не	Не
	4	Работа в динамична среда	В	Да	Да	Не	Не	Не	Не	Не	Не	Не	Не
Работят с указания какво да правят	5	Възможности за обучение	В	Да	Да	Да	Не	Не	Не	Не	Не	Не	Не
	6	Поделени данни	В	В	В	В	Да	Да	Да	Не	Не	Не	Не
	7	Умни сензори	В	В	В	В	Да	Да	Да	Не	Не	Не	Не
	8	Колективна работа	В	В	В	В	Да	Да	Да	Не	Не	Не	Не
	9	Планиране на мисии	В	В	В	В	Да	Да	Да	Не	Не	Не	Не
	10	Обща архитектура	В	В	В	В	Да	Да	Да	Не	Не	Не	Не
	11	Общо мрежово пространство	В	В	В	В	Да	Да	Да	Не	Не	Не	Не
	12	Алгоритми за разпознаване на образи	В	В	В	В	Да	Да	Да	Не	Не	Не	Не
	13	Предполагане на сложни препятствия	В	В	В	В	Да	Да	Да	Не	Не	Не	Не
	14	Автономна навигация в открити пространства	В	В	В	В	Да	Да	Да	Не	Не	Не	Не
	15	Навигационни карти	В	В	В	В	Да	Да	Да	Не	Не	Не	Не
	16	Компютърно зрение	В	В	В	В	В	В	В	Да	Да	Не	Не
	17	Инфрачервени ултразвукови локатори	В	В	В	В	В	В	В	Да	Да	Не	Не
	18	Лазерни локатори	В	В	В	В	В	В	В	Да	Да	Не	Не
	19	Предполагане на прости препятствия	В	В	В	В	В	В	В	Да	Да	Не	Не
	20	Автономна навигация в закрити помещения	В	В	В	В	В	В	В	Да	Да	Не	Не
	21	Дистанционно управление	В	В	В	В	В	В	В	В	В	Да	Да

**Легенда:** Роботът от съответната колона се съотнася към характеристиката от съответния ред както следва: **Да** – притежава, **не** – забранено е да притежава; **в** – възможно е, но не е задължително да притежава.

Групите са както следва: **Дистанционно управляеми работи** – те се управляват единствено отдалечено от оператор. Тяхна основна характеристика е, че те трябва да са високо надеждни, с отлично време за реакция на команди и просто управление; **Автономни работи** – работят автономно в закрити помещения. Средата, в която работят е относително статична. Имат вградени технологии за самостоятелна ориентация в затвореното пространство, разполагат с навигационни карти, притежават лазерни локатори и в някои случаи инструменти и алгоритми за компютърно зрение; **Автономни интелигентни работи** – те работят по правило на открити терени. При тях средата е динамично променяща се и значително по-трудна за идентифициране. Това налага тези работи да използват по-интелигентни алгоритми за разпознаване на различни ситуации, за планиране и дори за вземане на прости самостоятелни решения. Присъщо за тях е да разпознават и преодоляват в реално време сложни препятствия, да планират мисии, да работят по изпълнение на мисиите в група, да обменят в реално време данни и да поддържат в реално време изградена високонадеждна и скоростна комуникационна среда помежду си и с външни кореспонденти; **Обучаеми работи** – характерно за тях е, че програмирането им е заменено с инструменти за обучение и самообучение. Други използвани при тези работи инструменти са средствата за работа в силно динамични и преднамерено недружелюбни среди, както и необходимите за това средства за разпознаване на свойствата на обектите от динамично изменящата се среда; Социални работи – при които към свойствата от предходната група работите от тази група притежават още две свойства – богати възможности за различни видове интелигентно общуване с оператора (примерно чрез говор, мимики, жестове и действия) и прогнозираат бъдещи намерения на оператора в зависимост от контекста на разиграваната ситуация.

### **1. Характеристики на ИСПР**

Разгледаните 9 ИСПР (виж Таблица 3) са разположени в 3 групи, всяка от които притежава характеристиките, показани за съответната група в Таблица 4.

Групата ИСПР с най-слаби възможности е разположена в началото на двете таблици (ИСПР в редове от 1 до 4 от Таблица 3 и колони от 1 до 4 на Таблица 4).

Групите са както следва: **Класически ИСПР** – които са така разработени, че да следват стил на програмиране, възможно най-близък до класическо програмиране. Типично за този тип продукти е: наличието на добре развити ИСПР обикновено с наличието на удобни за работа графични интерфейси, липсата на интелигентни услуги, както и липсата на инструменти за колективна работа; **ИСПР с елементи на интелект** – които представляват комбинация между класически стил на програмиране и интелигентни функции. Обикновено интелигентните функции са реализирани на ниво услуга или компонент (най-често компютърно зрение или модули за навигация, както по-рядко и модули за преодоляване на препятствия). В тази група по правило отсъстват интелигентни функции, реализиращи възможности за колективна работа, за определяне на стратегии и за решаване на противоречия; **Интелигентни ИСПР** – които използват като архитектурни решения Системите за обработка на знания. Основна характеристика на разработения „Софтуер” е, че това не са програми, описващи каква работа да се свърши, а алгоритми за обучение, самообучение и автономно решаване на задачи.

Друга основна характеристика на тези среди е възможността за разпределена работа в група, като интелигентни услуги и компоненти се използват както на ниво локални приложения, така и на групово ниво, където се извършва стратегическо и тактическо планиране, колективна работа и групово вземане на решения. Типично за този тип продукти е: отсъствието на развойни среди и графични интерфейси; наличието на силно развити интелигентни услуги; добре развити инструменти за колективна работа.

Таблица 3 ИСПР – типове и производители

No	Тип ИСПР	Късо име на ИСПР	Име на ИСПР	Късо име на произ.	Производител
1	Класическа	MRDS	Microsoft Robotics Developer Studio	Ms	Microsoft
2	Класическа	URBI	Universal Real-Time Behavior Interface	Gosai	Gosai
3	Класическа	Orcoss	Open Robot Control Software	OrcOcons	EU IIST-2000-31064 Open Robot Control Software Consortium
4	Класическа	Player	Player - Open Software for robot programming	Stacocons	Stanford Univ. USA, Univ. of Southern California USA, Simon Fraser Univ. Canada Consortium
5	С елементи на интелект	ERSP	Evolution Robotics Software Platform	ER	Evolution Robotics
6	С елементи на интелект	AWARE	AWARE Robot Intelligence Software	iRobot	iRobot
7	Интелигентна	ARNL	Advanced Robotic Navigation and Localisation	MR	Mobile Robotics Inc.
8	Интелигентна	RLBCS	Robot Learning and Behavior Control System	SKI	Skilgent LLC
9	Интелигентна	FTCF	Fault-Tolerant Control Framework	SKI	Skilgent LLC

Таблица 4 ИСПР - характеристики

Категория	No	1	2	3	4	5	6	7	8	9	
Общи характеристики	1	Архитектура, базирана на събгиги към услуги	Архитектура, базирана на събгиги	Свойна архитектура	Свойна архитектура	Архитектура, базирана на събгиги към услуги	Модел Публикуваей - Абонрай	Свойна архитектура	Система за обучение на задачи	Система за обучение на задачи	
	2	не	ч	да	да	не	ч	не	ч	не	
	3	ч	ч	да	да	не	ч	не	ч	не	
	4	ч	ч	да	да	не	ч	не	ч	не	
	5	MSRS	URBILive/urbiMove+urbiLab	URBI	Real Time Framework	не	ERSP	да	ARAM	RLBCS	FTCS
	6	да	да	да	не	не	да	не	TrainingFactory	не	не
	7	VPIL	C++, Java, Matlab	C++, Java, Matlab	C++	Езикова независими (C++, Tcl, Java, Python)	Python	Python, C++	ARCL	обучение с примери	обучение с примери
	8	да	да	да	да	не	не	да	MobileSm	не	не
	9	да	да	да	да	да	не	да	да	да	да
Библиотеки	10	не	от външни производители	от външни производители	Open Robot Control Source	Open Robot Control Source	VRP + vSLAM	ACTS	да	да	
	11	да	да	да	да	да	да	да	да	да	
	12	не	не	не	не	не	не	не	не	да	
	13	не	не	не	не	не	не	не	не	да	
	14	не	не	не	не	не	не	не	не	да	
	15	да	да	да	да	да	да	да	да	да	
	16	да	да	да	да	да	да	да	да	да	
	17	не	не	не	не	не	не	не	не	да	
	18	не	не	не	не	не	не	не	не	не	
	19	DSS	urbi Runtime Engine	urbi Runtime Engine	Task manager	всичка с TCP протокол	RCC	Scripted Component	ARNL	RLBCS	FTCS
Средствa за реална работа	20	да	да	RTAI	RTAI	RTAI	BEEL	ARIA	JAVIS	JAVIS	
	21	HAL	HAL	?	HAL	HAL	HAL	?	?	?	
	22	DSSP	urbiScript	CORBA (IOP)	TCP/IP	Поддръжките от ОС	Поддръжките от ОС	Поддръжките от ОС	TOPIP, RS232 CAN, Ethernet	JAVIS OCUI JAUSOCUI	
	23	да	да	да	да	да	да	да	да	Windows, Linux	
	24	Windows	Windows, Linux	Linux	Windows (part), Linux	Windows (part), Linux	Windows, Linux	?, Linux	Windows, Linux	Windows, Linux	

## ИЗВОДИ

От анализа могат да се констатират следните по-важни факти: **(1)** Два основни стила на програмиране се използват при ИСПР – класическо (обикновено обектно ориентирано) програмиране и програмиране с елементи на изкуствен интелект; **(2)** Най-често се използват три софтуерни архитектури - слойни, ориентирани към услуги и тези за обработка на знания; **(3)** Използвани са два основни подхода за интеграция – чрез работни процеси и чрез комуникационни протоколи; **(4)** Използваните инструменти са платформено и езиково независими; **(5)** Практически липсват инструментите за неформално специфициране на решаваната задача; **(6)** Повечето от ИСПР не разполагат със средства за извличане на сложни семантични връзки от използваните формални спецификации на решаваната задача; **(7)** Широко се използват компоненти за многократна употреба; **(8)** Повечето от средите генерират код за работа в реално време с незадоволително бързодействие; **(9)** Всички среди са отворени за използване на външни компоненти; **(10)** Стандартизацията в предметната област не позволява задоволително автоматизиране на програмирането. Значително по-добре са стандартизирани комуникационните интерфейси. Доста по-слаба е стандартизацията на функционалността, структурата и диалозите за връзка с компоненти и услуги.

Характерно за разгледаните ИСПР е, че имат богат набор от библиотеки с класически компоненти за управление на работ. Те притежават добре развити средства за присъединяване на компоненти и услуги от външни производители. Имат разработени механизми за многократна употреба на собствените си компоненти и услуги. Имат и добре развити инструменти за неавтоматизирано проектиране.

ИСПР обаче имат и някои недостатъци. Примерно инструментите им за интегриране на класически и интелигентни системи в хибридни крайни решения не са достатъчно добре развити. Проблеми съществуват и при автоматизираното сглобяване от хардуерни елементи, софтуерни компоненти и услуги на системите за работа в реално време. Не е задоволителна платформената и езиковата независимост на разгледаните продукти. Относително ниска е степента на автоматизация при разработката на софтуер, като се чувства осезаемо липсата на инструменти за неформална спецификация на решаваните задачи. Наличните решения за автоматизиране на процеса не разполагат със задоволително бързодействие на генерирания за работа в реално време код.

Установените при анализа характеристики на ИСПР ще бъдат основните изисквания към метода, като се акцентира, върху отстраняването на възможно най-голям брой недостатъци на ИСПР.

## ЛИТЕРАТУРА

- [1] Bauer F.L. Chair. Report on a conference sponsored by the NATO SCIENCE COMMITTEE. Garmisch. Germany. 7th to 11th October 1968.
- [2] History of Software engineering. Aug 2009.  
[http://en.wikipedia.org/wiki/History\\_of\\_software\\_engineering](http://en.wikipedia.org/wiki/History_of_software_engineering).
- [3] DeMarko T. Software Engineering: An Idea Whose Time Has Come and Gone? IEEE. IEEE Software. July/August 2009. Pp. 95-96.
- [4] Somy M. Software Platforms for Service Robotics. Linux.Devices.com. Aug 2008.  
<http://linuxdevices.com/articles/AT9631072539.html>.

### За контакти:

Гл.ас инж. Иван Станев, Катедра “Информатика и информационни технологии”, Русенски университет “Ангел Кънчев”, тел.: 082-888-326, e-mail: [IStanev@ami.ru.acad.bg](mailto:IStanev@ami.ru.acad.bg)

**Докладът е рецензиран.**