

KBASE.T – изисквания към технологичната рамка

Иван Станев, Мария Колева

Abstract: KBASE Technological framework requirements. Automated software development issues are addressed in this paper. Layers and packages of a Common Platform for Automated Programming (CPAP) are defined based on Service Oriented Architecture, Cloud computing, KBASE and Method of automated programming. Tools of seven leading companies are analysed in the context of CPAP. Based on the results of the analysis CPAP requirements are formulated.

Key words: Knowledge Based Software Engineering, Automated Programming, Cloud Computing, Service Oriented Architecture, Requirements specification.

1. Въведение

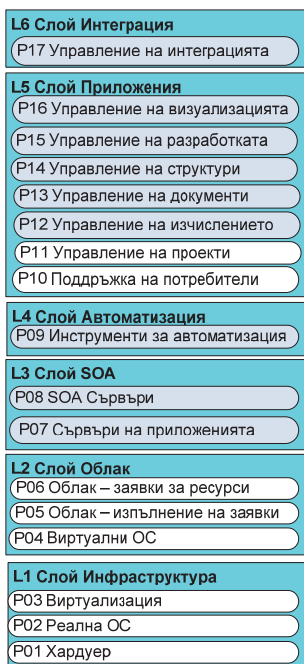
Бързо развиващият се процес на разработка на информационни системи е по-малко ефективен и по-ресурсоемък, отколкото софтуерните компании и техните клиенти биха желали. Широко използваните в рамките на целия жизнен цикъл на софтуерното инженерство методи и инструменти се характеризират с ниска степен на автоматизация, недостатъчни възможности за многократна употреба на разработваните компоненти и краен продукт, който трудно се адаптира към бързо променящите се изисквания на заобикаляния свят.

Сериозна стъпка към по-добра автоматизация на програмирането представляват двете широко разпространени индустриални технологии Service Oriented Architectures (SOA) и Cloud Computing (CC).

Тези технологии взаимно се допълват ([9]). Технологията SOA ([7]) дава добри резултати при автоматизирано съгласяване на нови софтуерни продукти от компоненти за многократна употреба. Технологията CC ([5]) притежава отлични инструменти за автоматично управление на ресурсите и разпределените изчисления.

Два въпроса обаче при тези технологии остават не достатъчно добре решени. Първият е как с по-малко ресурси и знания да бъде специфицирана непълно и неточно решаваната задача и как автоматично, с помощта на използваните инструменти да бъде достигната необходимата пълнота и точност на заданието. Вторият също толкова важен въпрос е как от вече пълното и точно задание да се генерира автоматично работещ софтуерен продукт. Възможен отговор на двата въпроса е даден в разработените Knowledge Based Automated Software Engineering (KBASE [11]) и Method for Automated Programming of Robots (MAP [10]).

Изграждането на обща технологична рамка за автоматизация на програмирането може да се постигне чрез комбиниране на методите SOA, CC, KBASE и MAP. Изискванията към тази рамка ще бъдат определени, като се анализират някои от най-използваните инструменти за разработване на CC сравнени през тук дефинираната Обща платформа за автоматизация на програмирането (ОПАП).



Фиг. 1 Словес и пакети на ОПАП

2. Структура на ОПАП

Моделът на ОПАП е разработен като са взети под внимание особеностите на някои основни инструментални среди за СС ([14]), на стандартите, дефиниращи SOA ([7]) и СС ([5]) и описанието на методите KBASE([11]) и MAP ([10]). Структурата на ОПАП е показана на Фиг. 1.

ОПАП е изграден от 6 слоя както следва:

L1 Слой Инфраструктура, който организира и управлява работата на протичащите в хардуерната и комуникационна инфраструктура процеси – на физическо ниво (**P01 Инфраструктура**), на ниво Операционна система (**P02 Реална операционна система**) и чрез инструментите за виртуализация (**P03 Виртуализация**).

L2 Слой Облак е предназначен автоматично да организира работата по изпълнението на заявените за обработка задания в облачната среда под управлението на виртуалната операционна система (**P04 Виртуална ОС**), чрез механизма за скалиране на ресурси (**P05 Облак – изпълнение на заявки**) и чрез запазване на необходимите за заданието физически и виртуални ресурси (**P06 Облак – заявки за ресурси**).

L3 Слой SOA, който е предназначен да изпълнява потребителските задания и да управлява процесите по тяхното изпълнение на ниво услуги (чрез **P07 Сървъри на приложенията**) и на ниво процеси (чрез **P08 SOA Сървъри**).

L4 Слой Автоматизация – предназначен чрез пакета си **P09 Инструменти за автоматизация** да генерира заявени от потребителя софтуерни продукти, по направени от него спецификации на BPMN, UML, чрез графични интерфейси, на естествен език и други.

L5 Слой Приложения включва богат набор от готови инструменти, които се използват по заявка на потребителя за сглобяване на неговия софтуер. Сглобяването в общи софтуерни решения може да се управлява от L4, но може да се управлява и от отделни компоненти на L5. Този слой включва следните пакети: **P10 Поддръжка на потребителите**, отговорен за указване на помощ при работата с активния в момента продукт или относно действия в контекста на ОПАП; **P11 управление на проекти**, отговорен за организация на работата по управление на проекти; **P12 Управление на изчислението**, отговорен за управление на диагностиката и самодиагностиката на софтуерните продукти по време на тяхното изпълнение в реално време; **P13 Управление на документи**, отговорен за създаването и работата с документи, както и управлението на документопотоците в ОПАП; **P14 – Управление на структури**, отговорен за създаване и поддръжка на структурата на организацията собственик на ОПАП; **P15 – Управление на разработката**, с отговорност за управление на процеса на създаване на нови софтуерни продукти в контекста на ОПАП и тяхното актуализиране по заявка на потребителите (възможно и в реално време); **P16 Управление на визуализацията** отговорен за управлението на комуникацията между ОПАП и всички нейни роли.

L6 Слой Интеграция включва в своя пакет **P17 Управление на интеграцията** инструменти за интегриране на продукти, процеси и услуги, разработени в рамките на организацията, от партньори, от трети страни.

3. Определяне на изискванията към ОПАП

За провеждане на сравнителния анализ са избрани от водещи компании следните инструментални среди (ИС) за СС: **T1 AWS** на Amazon ([12]), **T2 Azure** на Microsoft ([8]), **T3 App Engine** на Google ([2]), **T4 vCloud** на VMWare ([13]), **T5 Bluemix** на IBM ([1]), **T6 Helion** на HP ([3]) и **T7 OCPaaS** на Oracle([6]).

Резултатите от сравнителния анализ са отразени в

Таблица 1. В редовете на таблицата са включени пакетите на ОПАП, чиито функционалности тук са разглеждани като важни характеристики на ОПАП. С цел подобряване на качеството на анализа някои от пакетите (P07, P08, P09, P16 и P17) са представени с всичките си компоненти. Номерът на слоя, в който се намира всеки пакет е отбелязан в колона Слой. Номерът на пакета – в колона Пакет. Типът на съответния пакет – в колона Тип характеристика. Типовете пакети са: **класически** – не съдържа компоненти, поддържащи автоматизация на програмирането, **автоматизиран** – съдържа само компоненти, позволяващо автоматизирано програмиране, **комбиниран** – съдържа както класически, така и автоматизирани компоненти, **онтологичен** – позволява работа както с бази данни, така и с бази знания, действия по извличане на тези знания, самообучение и различни видове интелигентни търсения. В колони от 5 до 11 на тази таблица са включени участващите в сравнението инструментални среди и компании. В пресечните точки на колони и редове е поставен някой от следните символи – **1** – „инструментът притежава дадена характеристика“, **празен символ** – „инструментът не притежава дадена характеристика“, **?** – „не е открита информация дали инструментът притежава дадената характеристика“.

Таблица 1.
Резултати от сравнителен анализ на Инструментални среди за СС

Слой	Пакет	Тип	Платформа	Т1	Т2	Т3	Т4	Т5	Т6	Т7
L6			Слой Интеграция							
	P17	Класически	Управление на потребители	1	1	1	1	1	1	1
		Комбиниран	Система за управление на ресурси						1	
		Класически	Интеграция на стари системи	1	1			1	?	1
		Класически	Интеграция на партньорски системи	1	1			1	?	1
		Класически	Интеграция на отдалечени системи	1	1			1	?	1
		Класически	Комуникационен канал	1	1	1	1	1	?	1
		Онтологичен	Семантична комуникация							
L5			Слой Приложения							
	P16	Класически	Монитор разработчик	1	1	1	1	1	1	1
		Класически	Портал	1	1		1	1	1	1
		Класически	Монитор СисАдмин	1	1	1	1	1	1	1
		Класически	Монитор оперативна работа	1	1	1	1	1	1	1
		Класически	Монитор справки	1	1		1	?	1	
		Онтологичен	Монитор онтологии							
	P15	Комбиниран	Управление на разработката		1	1	1	1	1	1
	P14	Комбиниран	Управление на структури		1					
	P13	Комбиниран	Управление на документи	1		1		1		1
	P12	Комбиниран	Управление на изчисленията	1	1		1	1	1	1
	P11	Класически	Управление на проекти			1		1	1	1
	P10	Класически	Поддръжка на потребители	1	1	1	1	1	1	1
L4			Слой Автоматизация							
	P09	Автоматизиран	Модел ОБД				1	1	1	1
		Автоматизиран	Модел дела			1		1		
		Автоматизиран	Модел интерфейси							
		Автоматизиран	Модел справки		1			1	1	1
		Комбиниран	Модел бизнес процеси		1		1	1	1	1
		Комбиниран	Модел услуги		1	1	1	1	1	1
		Комбиниран	Модел РБД				1	1	1	1
		Онтологичен	Модел онтологии							

Слой	Пакет	Тип	Платформа	Характеристика	T1	T2	T3	T4	T5	T6	T7
L3				Слой SOA							
	P08	Класически		Сървър идентификация	1	1	1	1	1	1	1
		Класически		Уеб сървър	1	1	1	1	1	1	1
		Класически		Процес сървър	1	1	1	1	1	1	1
		Класически		Сървър дела					1		1
		Онтологичен		Сървър онтологии							
	P07	Автоматизиран		Сървър документи					1		1
		Автоматизиран		Монитор системни параметри	1	1		1	1	1	1
		Автоматизиран		Сървър ОБД					1		
		Класически		LDAP/AD Сървър	1	1	1	1	1	1	1
		Класически		Сървър приложения	1	1	1		1	1	1
		Класически		Сървър РБД	1	1	1	1	1	1	1
		Класически		Машина за търсене	1	1	1		1	1	1
		Класически		Сървър eMail	1	1	1	1	1	1	1
		Онтологичен		Сървър индексирание текст			1				
L2				Слой Облак							
	P06	Класически		Облак - заявки за ресурси	1	1	1	1	1	1	1
	P05	Класически		Облак - изпълнение на заявки	1	1	1	1	1	1	1
	P04	Класически		Виртуална ОС	1	1	1	1	1	1	1
L1				Слой Инфраструктура							
	P03	Класически		Виртуализация	1	1	1	1	1	1	1
	P02	Класически		Реална ОС	1	1	1	1	1	1	1
	P01	Класически		Хардуер	1	1	1	1	1	1	1

От прегледа на таблицата могат да се направят следните изводи: (1) Всички ИС имат отлично развити инструменти за програмиране и автоматизация на програмирането в слоеве L1, L2, L5 P16 и L6. (2) Всички ИС имат добре развити инструменти за разработване на SOA приложения. (3) Наблюдава се недостиг на средства за автоматизация в слой L3 пакет P07 и слой L5 пакети P12, P13, P14 и P15. (4) Практически силен недостиг на развити инструменти за автоматизация на процесите в слой L4. (5) Практически липсват инструменти за работа с Онтологии, както и други инструменти за обработка на знания.

На база на направения анализ и произтичащите от него изводи могат да бъдат определени следните изисквания към ОПАП:

- **R1** ОПАП трябва да съчетава техники за автоматизация на програмирането, използвани в методите SOA, CC, KBASE и MAP.
- **R2** ОПАП трябва да позволява непълно и неточно специфициране на заданието с език и инструменти близки до крайни потребители от типа „човек от улицата“.
- **R3** платформата трябва да позволява работа със знания и тяхната интерпретация (системи за обработка на знания, онтологии и размити множества) при автоматизираното отстраняване на непълноти и неточности в спецификацията и при генериране на софтуерните продукти.
- **R4** платформата трябва да гарантира автоматизирано генериране на софтуерни продукти от пълната и точна спецификация на задачата.
- **R5** ОПАП трябва да позволява автоматизирано използване на техники от типа „Contract Testing“ и „Quality of Service Testing“ за донастройка на софтуерните продукти, за проверка на коректността на изпълняваните от тях действия и за коректно присъединяване на софтуерни компоненти от трети страни.

- **R6** Трябва да бъде обърнато специално внимание на автоматизираното генериране на компоненти за идентификация (специално авторизация), сигурност и интеграция на данните между вътрешни и външни за ОПАП потребители.

4. Заключение

Въпреки бързото развитие на софтуерната индустрия и използваните през последните години високотехнологични решения за разработване на софтуерни продукти, продължава търсенето на неестествено голям брой програмисти. Направеният в материала анализ показва, че все още автоматизацията на програмирането представлява индустриална ниша, която дори най-големите ИТ компании не успяват да запълнят. Анализът също показва, че един възможен път за автоматизация на програмирането е съчетаването на технологии като SOA, CC, KBASE и MAP. За успешното реализиране на тази комбинация от технологии е необходимо да се разработи стандартизиран инструмент от типа на тук предложената Обща платформа за автоматизация на програмирането.

Благодарности: Работата е финансирана по договор ДФНИ - И02/13 с ФНИ.

ЛИТЕРАТУРА

- [1] Fred, A.M., et all. IBM Bluemix Architecture Series: Web Application Hosting on IBM Containers Leveraging best practice and reference architectures for cloud. IBM. Redbook. P.94. 2015.
- [2] Google. Architecture: Web Application on Google App Engine. <https://cloud.google.com/solutions/architecture/webapp> (Accessed 1 Oct 2015).
- [3] HP. Understanding the HP Cloud System Reference Architecture. White Paper. HP. P.20. June 2011
- [4] Liu, F., et. all. NIST Cloud Computing Reference Architecture. Gaithersburg: National Institute of Standards and Technology Special Publication 500-292. US Department of Commerce. P.35. 2011.
- [5] Mell, P., T.Grance. The NIST Definition of Cloud Computing. Gaithersburg: National Institute of Standards and Technology NIST Special Publication 800-145 US Department of Commerce. P.7. 2011.
- [6] Oracle. Oracle Cloud Computing. White Paper. Oracle. P.22. 2010.
- [7] Organization for the Advancement of Structured Information Standards (OASIS). Reference Model for Service Oriented Architecture. OAZIS. P.31. 2006.
- [8] Pedigari Bala, G. Phadke. Microsoft Windows Azure – The Cloud Computing Platform. White Paper. Tata Consultancy Ltd. P.25. 2011.
- [9] Piprani, B., D.Sheppard, A.Barbir. Comparative Analysis of SOA and Cloud Computing Architectures Using Fact Based Modeling. Springer-Verlag Berlin Heidelberg: OTM 2013 Workshops Volume 8186 of the series Lecture Notes in Computer Science, pp. 524–533. 2013.
- [10] Stanev I. Method for Automated Programming of Robots. In Knowledge Based Automated Software Engineering. Cambridge Scholars Press. Cambridge. Pp.67 – 85. 2012.
- [11] Stanev, I., K.Grigorova. KBASE Unified Process. Knowledge Based Automated Software Engineering. Cambridge Scholars Publishing. Cambridge Pp. 1 – 19. 2012.
- [12] Varia, J. Architecting for the Cloud: Best Practices. White Paper. Amazon. P.23. 2011.
- [13] VMWare. VMWare vCloud Architecting vCloud. Technical White Paper. VMWare. P.100. 2010.

[14] Woods, J. Google, Microsoft, VMware, and Amazon: How Cloud Computing Changes the Vendor Landscape. Gartner The Future of IT Conference Mexico. P.25. 2011.

За контакти:

Доц. д-р Иван Николаев Станев, Катедра “Информатика и информационни технологии”, Русенски университет “Ангел Кънчев”, тел.: +359882117345, e-mail: instanev@gmail.com

Маг. Мария Петкова Колева, Катедра “Компютърна информатика”, Софийски университет “Св.Климент Охридски”, тел.: +359882555491, e-mail: marie.koleva@gmail.com

**РУСЕНСКИ УНИВЕРСИТЕТ „АНГЕЛ КЪНЧЕВ”
UNIVERSITY OF RUSE „ANGEL KANCHEV“**

ДИПЛОМА

**Програмният комитет на
Научната конференция RU&SU'15
награждава с КРИСТАЛЕН ПРИЗ**

“THE BEST PAPER”

**доц. д-р ИВАН СТАНЕВ
и МАРИЯ КОЛЕВА**

автори на доклада

**“ KBASE – изисквания към
технологичната рамка”**

DIPLOMA

**The Programme Committee of
the Scientific Conference RU&SU'15
awards the Crystal Prize "THE BEST PAPER"
to Assoc. Prof. PhD IVAN STANEV and**

MARIA KOLEVA,

authors of the paper

**“KBASE – Requirements to the
Technological Framework”**

**РЕКТОР Чл.-кор. проф. д.т.н. Христо Белоев, DHC, mult.
RECTOR COR MEM Prof. DSc. Hristo Beloev, DHC, mult.**

10.10.2015