

## KBASE.T – технологична рамка

Иван Станев, Мария Колева

**Abstract: KBASE Technological framework.** Common Platform for Automated Programming (CPAP) is defined in details. Two versions of CPAP are described: Cloud based (including set of components for classic programming, and set of components for mixed programming); and KBASE based (including set of components for automated programming, and set of components for ontology programming). Four KBASE products (Modul for Automated Programming of Robots, Intelligent Product Manual, Intelligent Document Display, and Intelligent Form Generator) are analysed and its contributions to automated programming are described.

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

### 1. Въведение

**Общата технологична рамка за автоматизирано програмиране** предложена в [8], на база на направените в [5] и [8] анализи е изградена на като комбинация от облачни изчисления (CC), архитектури ориентирани към услуги (SOA), автоматизирано софтуерно инженерство базирано на знания (KBASE) и метод за автоматизирано програмиране на работи (MAP).

Предложената в [8] **Обща платформа за автоматизирано програмиране** (ОПАП) е изградена от четири типа компоненти, разположени в различните слоеве на ОПАП. Тези типове са както следва: (1) Компоненти за класическо програмиране (ККП); (2) Компоненти за смесено програмиране (КСП); (3) Компоненти за автоматизирано програмиране (КАП); (4) Компоненти за онтологично програмиране (КОП).

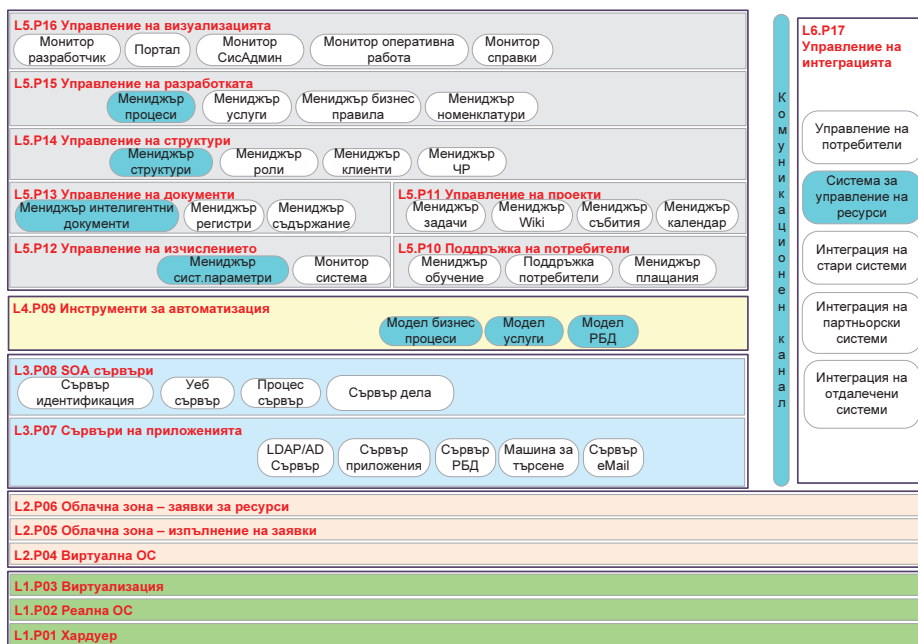
Тук ще бъдат описани накратко всички основни компоненти по техните слоеве (L3 - L6) и пакети (P07 – P17) съгласно въведената в [8] структура на ОПАП.

### 2. Компоненти на ОПАП за работа в облака

Технологичната рамка на ОПАП използвана за работа в Облака ([3], [4]) е показана на Фиг. 2, като са разгледани два типа компоненти – за класическо и за смесено програмиране.

**Компонентите за класическо програмиране**, позволяват реализацията на класическите техники за програмиране, свързани с трислойните архитектури, SOA ([4]) и CC ([2]) и представляват най-малък интерес от гледна точка на процеса на автоматизация. Тези компоненти са онова минимално ядро от технически средства, без които е практически невъзможно функционирането на стандартните софтуерни продукти. В групата са включени следните компоненти по слоеве и пакети: в слой L3, пакет P07 - компонентите: C1 LDAP / AD сървър, C2 Сървър на приложението, C3 Сървър на релационната база данни, C4 Машина за търсене, C5 eMail сървър; в пакет P08 - компонентите: C6 Сървър за управление на идентификацията, C7 Уеб сървър, C8 Процес сървър, C9 Сървър за управление на дела (папки от документи).

**Компонентите за смесено програмиране**, които представляват техники за класическо програмиране, надстроени с елементи на автоматизация (примерно динамично реконфигуриране на структури данни или управление на изчислителния процес чрез спецификации на бизнес правила, бизнес процеси и други). Компонентите от този тип са: в слой L4, пакет P09 компонентите: C10 Модел бизнес процеси, C11 Модел услуги, C12 Модел РБД; в слой L5, пакет P12 компонент C13 Мениджър системни параметри; в пакет P13 компонент C14 Мениджър интелигентни документи; в пакет P14 компонент C15 Мениджър структури, в пакет P15 компонент C16 Мениджър процеси; в слой L6, пакет P17 компоненти: C17 Система за управление на ресурси, C18 Комуникационен канал.



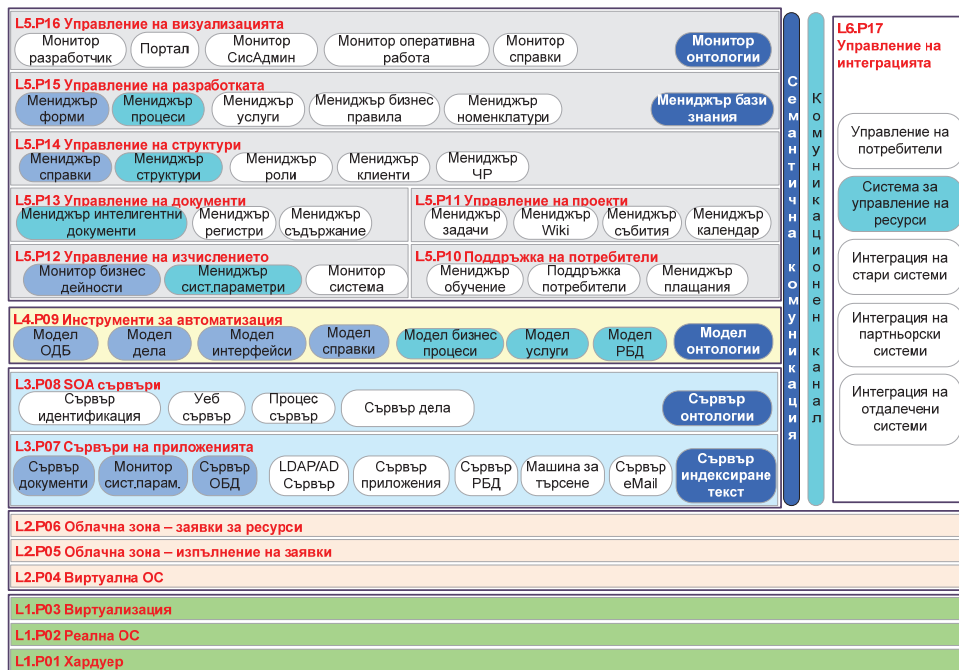
Фиг. 2 Технологична рамка на ОПАП за работа в облака

### 3. Компоненти на ОПАП за автоматизирано програмиране

Технологичната рамка на ОПАП използвана за автоматизация на програмирането (Фиг. 3), освен посочените в предходната точка компоненти съдържа още два типа компоненти ([1], [9], [11]) – за автоматизация на програмирането и за онтологично програмиране.

**Компонентите за автоматизирано програмиране**, които представляват инструменти за автоматизация, базирани на интерпретацията на формални модели (примерно директно генериране на софтуерни продукти от UML, BPMN, EPC, езици 4 поколение, формални методи и други). Този тип компоненти включва: в слой L3, пакет P07 - компоненти: C19 Сървър документи, C20 Монитор системни параметри, C21 Сървър ОБД; в слой L4, пакет P09 - компоненти: C22 Модел ОБД, C23 Модел дела, C24 Модел интерфейси, C25 Модел справки; в слой L5, пакет P12 - компонент: C26 Монитор бизнес дейности; в пакет P14 - компонент: C27 Мениджър справки; в пакет P15 - компонент: C28 Мениджър форми.

**Компонентите за онтологично програмиране**, които осигуряват автоматизация чрез интерпретация на знания, обучение и самообучение (примерно генериране на софтуер по онтологични описания, размити интерпретатори на непълни и неточни задания, генератори на код от естественоезикови спецификации и други). Този тип компоненти включва: в слой L3, пакет P07 - компонент: C29 Сървър индексирание текст; в пакет P08 - компонент: C30 Сървър онтологии; в слой L4, пакет P09 - компонент: C31 Модел онтологии; в слой L5, пакет P15 - компонент: C32 Мениджър бази знания; в пакет P16 - компонент: C33 Монитор онтологии; в слой L6, пакет P17 - компонент: C34 Семантична комуникация.



Фиг. 3 Технологична рамка на ОПАП за автоматизирано програмиране

#### 4. Приноси на ОПАП към индустриалното програмиране

За да бъде оценено какви са приносите при използване на технологичната рамка на ОПАП към индустриалното програмиране са анализирани по избрани количествени и качествени показатели (редовете в Таблица 2) следните софтуерни разработки (стъбловете в Таблица 2): Модул за автоматизирано програмиране на работи ([6], [7], [9]), Интелигентна потребителска документация ([10]), Интелигентен дисплей за документи ([1]), Интелигентен генератор на форми ([12]).

Таблица 2 Приноси на ОПАП към индустриалното програмиране

Тип	Име на подобрение	KBASE приложения			
		MAP	IPM	IDD	IFG
колич.	намалява времето за специфициране	да			
колич.	намалява времето за програмиране	да	да	да	да
колич.	намалява времето за присъедияване на COTS компоненти			да	
колич.	намалява времето за тестване	да	да	да	да
колич.	намалява времето за тестване на COTS компоненти			да	
колич.	намаляване на членовете на ИТ екипа	да			да
кач.	адаптиране към различни потребителски области	да	да	да	да
кач.	адаптиране към различни крайни потребители	да	да		
кач.	адаптиране на презентациите към различни стандарти		да		

Тип	Име на подобрение	КBASE приложения			
		MAP	IPM	IDD	IFG
кач.	адаптиране на представянето към различни медии	да	да		
кач.	синхронизиране на кода в реално време		да		да
кач.	синхронизиране на документите в реално време		да		
кач.	предотвратяване на аварийни спирания			да	да
кач.	подобряване на бързодействието в реално време			да	да

Като резултат от анализа могат да бъдат направени следните изводи: (1) безспорно намаляват времето за програмиране и тестване на разработваните приложения; (2) След разделянето на две фази на процеса на проектиране само първата, еднократно изпълнявана фаза за подготовка на знанията изисква увеличен ресурс за проектиране, докато втората, многократно изпълнявана фаза за решаване на задачата изисква чувствително намален ресурс за работа; (3) в задоволителна степен намалява времето за присъединяване и тестване на COTS компоненти; (4) наблюдават се значителни подобрения в качеството на разработваните продукти, в това число – адаптиране към нови предметни области и различни крайни потребители, реструктуриране за ефективно използване на качествата на различни медии, подобряване на процеса на версионирание и други.

### ЗАКЛЮЧЕНИЕ

Създадената технологична рамка на ОПАП предлага възможности за поевтиняване, по-кратък цикъл на разработка и качествени подобрения на софтуерните продукти в резултат от съчетаването на положителните страни на различните комбинирани в ОПАП технологии за автоматизация на програмирането.

Усилията, необходими за въвеждане на инструментите за автоматизирано програмиране имат задоволителна икономическа възвращаемост.

### Благодарности:

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

### ЛИТЕРАТУРА

[1] EU IST-1999-20162 Development and Applications of New Built-in-Test Software Components in European Industries. Software Architecture. 2003

[2] Liu, F., et. all. NIST Cloud Computing Reference Architecture. Gaithersburg: National Institute of Standards and Technology Special Publication 500-292. US Department of Commerce. Pp. 35 2011.

[3] 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. Pp. 7 2011.

[4] Organization for the Advancement of Structured Information Standards (OASIS). Reference Model for Service Oriented Architecture. OASIS. Pp. 31. 2006.

[5] 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.

[6] Stanev I. A Bulgarian Linguistic Processor Based on the Formal Model Control Networks - General Concepts. In proceedings of the CompSysTech'2002. Sofia. Bulgaria. June. 2002. Pp.. III.7-1 – III.7-5.

[7] Stanev I. Formal Programming Language Net. Part I – Conception of the Language. In proceedings of the CompSysTech'2001. Sofia. Bulgaria. June. 2001. Pp. I.16-1 – I.16-5.

[8] Stanev I. M. Koleva, KBASE – Requirements for Technical Framework. Proceedings of University of Ruse. 2015. (In Press).

[9] Stanev I. Method for Automated Programming of Robots. In Knowledge Based Automated Software Engineering. Cambridge Scholars Press. Cambridge. Pp.67 – 85. 2012.

[10] Stanev I., Ch. Vezirov, R. Kozlev, R. Kozhuharov, S. Kalinova. Intelligent Product Manual - Definitions, Structure, and Application in Agricultural Engineering. Proceeding of the XVI International Conference on "Material Flow, Machines and Devices in Industry" - ICMFMDI'2000, 07 -08 December 2000, Belgrad, Yugoslavia, pp. 1-153 , 1-156.

[11] Stanev, I., K.Grigorova. KBASE Unified Process. Knowledge Based Automated Software Engineering. Cambridge Scholars Publishing. Cambridge Pp. 1 – 19. 2012.

[12] Д-26/30.05.2012 (EU OPAC. Program K10-31-1 / 07-09-2010) Realisation of Priority Municipality Electronic Administrative Services. Software Architecture. Ministry of Transport Information Technologies and Communications. Sofia. Bulgaria. 2013.

**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