Eastern Academic Journal

ISSN: 2367-7384
Issue 4, pp. 62-75, December, 2017

ПОДХОД ЗА ПРИЛАГАНЕ НА ШАБЛОНИТЕ ПРИ РАЗРАБОТКАТА НА ИНФОРМАЦИОННИ СИСТЕМИ

Мария Армянова

Икономически универаитет, Варна, България

РЕЗЮМЕ — *Шаблоните са доказани решения, които подобряват процеса на разработка и спомагат създаването на успешни информационни системи. Шаблонът е добро решение, ако използването му носи повече ползи, отколкото недостатъци за проектния проблем. Статията предлага подход, съдържащ последователност от стъпките за комплексното прилагане на шаблоните при изграждането на информационна система. Той има за цел да постави на формална основа процеса и да спомогне за преодоляването на някои от проблемите, възникващи при използването им. Разгледан е начин за вписване на шаблоните в последователно създаваните модели на системата, за да може подходът да се включи в методологиите, базирани на подхода UML.*

Ключови думи: подход, прилагане, последователност от стъпки, шаблони за проектиране

S **Допълнителни материали / Supplementary materials**

APPROACH FOR DESIGN PATTERN' APPLICATION IN THE DEVELOPMENT OF INFORMATION SYSTEMS

Mariya Armyanova

University of Economics, Varna, Bulgaria

ABSTRACT— *The design patterns are proven solutions that improve the development process and help to create successful information systems. The design pattern is a good solution if its use brings more benefits than deficiencies to the project problem. The article provides an approach that contains a sequence of steps for the complex design patterns' implementation in the construction of an information system. It aims to put the process on a formal basis and help to overcome some of the problems that arise when using them. The design patterns are inserted into the consequently created system models so the approach can be included in the UML-based methodologies.*

Keywords: approach, implementation, sequence of steps, design patterns

Corresponding author at: University of Economics, Varna, Bulgaria, e-mail: **armianova@abv.bg**

ISSN: 2367-7384
Issue 4, pp. 62-75, December, 2017

1. ВЪВЕДЕНИЕ

Шаблоните са доказани решения, които подобряват процеса на разработка и спомагат създаването на успешни информационни системи. Шаблонът е добро решение, ако използването му носи повече ползи, отколкото недостатъци за проектния проблем. За да се гарантира получаването на качествен резултат от прилагането на шаблоните в процеса на разработката, е необходимо то да не зависи само от уменията и компетентността на специалистите, а да е повтаряем и систематизиран процес.

Статията предлага подход, съдържащ последователност от стъпките за прилагането на шаблоните при изграждането на информационна система. Той има за цел да постави на формална основа процеса и да спомогне за преодоляването на някои от проблемите, възникващи при използването им. Разгледан е начин за вписване на шаблоните в последователно създаваните модели на системата, за да може подходът да се включи в методологиите, базирани на подхода UML.

Необходимо е да се определят условията за успешно прилагане на шаблоните, като се сравнят предложени до момента методи на различни автори.

Според Коплин (Coplien, 1995) е необходимо е да се познават шаблоните. Усвояването на същността на всеки шаблон, за какво е предназначен и т.н., позволява шаблоните да се използват успешно при разработката. Необходимо е да се познават решенията предлагани от шаблоните, за да се прецени нуждата от отклонение или промяна на доказаните практики, капсулирани в тях.

Коплин разглежда обектноориентираните шаблони, които са ограничено множество. Те както казахме спомагат и за обучението на начинаещите разработчици, като познаването им подпомага усвояването на принципите на обектноориентираната парадигма. Но в момента развитието на информационните технологии и шаблоните, които ги прилагат е изключително динамично. Каталозите от шаблони постоянно се разрастват и е почти невъзможно да се познават всички. Филинг (Fehling, 2012) в подхода си за разработване на шаблони тръгва от презумпцията, че разработчиците се насочват към търсенето на шаблон, когато се сблъскат с конкретен проблем. За да могат да открият подходящия шаблон за конкретния проблем, разработчиците трябва да знаят какви алтернативи имат. Затова е подходящо да се смекчи изискването на Коплин. Необходимо е да се познават шаблоните, отнасящи се до разработваното приложение.

Разпространено мнение сред начинаещите разработчици е, че тъй като шаблоните предлагат успешни и доказали се решения на софтуерни проблеми, е добре да се ползват колкото се може повече при разработката на софтуера. Обаче, съществуват редица разработки, при които са използвани шаблони, но качеството на получените разработки не е добро. Това не означава, че разпространеното мнение е грешно. Противоречието идва от очакваното качеството при условие, че са използвани шаблони. Затова използването им не е самоцел, а е във връзка с решаване на открит проблем в точно определен контекст. Шаблоните не са универсално решение за всички проблеми при изграждането на системите. Те не могат да се използват винаги и навсякъде, но и пълното им игнориране е неправилно. Всеки проблем на разработката може да се реши по много начини. Разработчиците вземат решение, базирано на разбирането им за проблема и професионалните си умения и избират едно от възможните решения на проблема. Финалната разработка се получава, като резултат от взаимодействието на избраните проектни решения.

ISSN: 2367-7384
Issue 4, pp. 62-75, December, 2017

Можем да обобщим изискванията към разработчиците със следните условия за успешно прилагане на шаблоните:

- Познаване на шаблоните, отнасящи се до разработваното приложение;
- Шаблоните се използват за решаване на открит проблем в точно определен контекст;
- Проследяване на развитието на технологиите и познаване на настъпилите промени и шаблоните, които подпомагат въвеждането им;
- За всеки проблем трябва да се избере най-подходящата алтернатива на решение;
- Промяната внесена в шаблон не бива да противоречи на решението капсулирано в него;
- За системата трябва да се отчете взаимодействието на избраните решения по разработката.

Тези изисквания към разработчиците са пожелателни и затова е добре да потърсим начин да ги включим в някакъв подход, който да улесни прилагането им.

2. МАТЕРИАЛИ И МЕТОДИ

За да се създаде представения шаблон са направени проучвания върху съществуващите в момента шаблони. Използвани са методи на системен, сравнителен и исторически анализ. За разработката на шаблона са ползвани методи за моделиране – UML.

3. РЕЗУЛТАТИ

Резултатът от направените проучвания е дефиниране на последователността от стъпки на подход за прилагане на шаблоните при разработката на информационни системи. Той е представен с диаграми, разработени в съответствие с UML подхода.

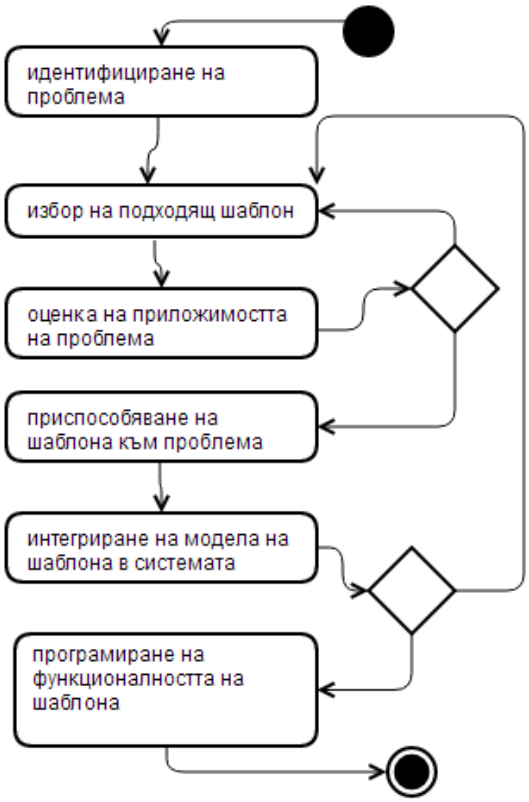
Съвременните методологии за изграждане на информационни системи не предлагат формална методика за прилагане на шаблоните. Считаме, че една от причините е, че не може да се изгради информационната система само от шаблони. Целта на разработчика е да открие оптимално решение на проектните проблеми, но не за всеки проблем има разработен шаблон. Като следствие от търсенето на решение чрез тях, разработчикът може да стигне до извода, че трябва да открие индивидуално решение на проблема, т.е не винаги има осезаем резултат. Те подпомагат процесите на разработка, а не ги реализират, затова няма специализирана методология, ориентирана към шаблоните. Според Одентал (Odenthal, 1997, с. 511-529) не може да се създаде методология за разработка основана единствено на шаблоните. Те са мисловни градивни единици, които подпомагат разработчика в основните етапи на жизнения цикъл на системите. Затова смятаме, че е от съществено значение е да се предложи подход за съвместното прилагането на шаблоните при изграждането на информационните системи, по който да се водят разработчиците при употребата им.

Редица учени са работили върху проблеми, свързани с използването им. Кумар (Kumar, 2013) и Филинг (Fehling, 2015) предлагат последователност от стъпки за използване на шаблоните при разработка на облачни приложения. Одентал (Odenthal, 1997, с. 511-529) предлага подход за прилагане на шаблоните, който е силно специализиран за обектноориентираните шаблони. Според Мили (Mili, 2002) първо трябва да се разпознае възможността, т.е да се открие подходящ шаблон и да се оцени, като потенциално решение на проблема. После би трябвало да се разбере шаблона, т.е. принципите, на които се

ISSN: 2367-7384
Issue 4, pp. 62-75, December, 2017

основава и структурата му. Накрая шаблонът се адаптира към проблема така, че да може да се интегрира в системата.

Въз основа на всички тези изследвания, може да определим последователност от стъпки, изпълнявани при прилагането на даден шаблон за изграждане на информационна система. Основните стъпки са: идентифициране на проблема, за който се търси подходящ шаблон, по време на анализа и проучването на проекта или съществуващата система; селектиране на подходящи шаблони и взаимоотношенията им; оценява се приложимостта на различните алтернативи и се избира шаблон; избраният шаблон трябва да се проучи, приспособи и документира така, че да може да се използва придобитото знание отново на по-късен етап; да се интегрира и програмиране. Познаването на шаблоните, използвани в програмите, улеснява разбирането им, а оттам и поддръжката им. Всеки шаблон има много възможни приложения. Екземплярът на шаблона може да се различава от описаните в каталога приложения, но да съответства на поставената цел за шаблона. На фиг. 1 е показана последователността от стъпките на подхода за прилагането на шаблони при разработката на информационните системи.



Фиг. 1. Диаграма на дейностите за последователността на стъпките в подхода за разработване на шаблони.

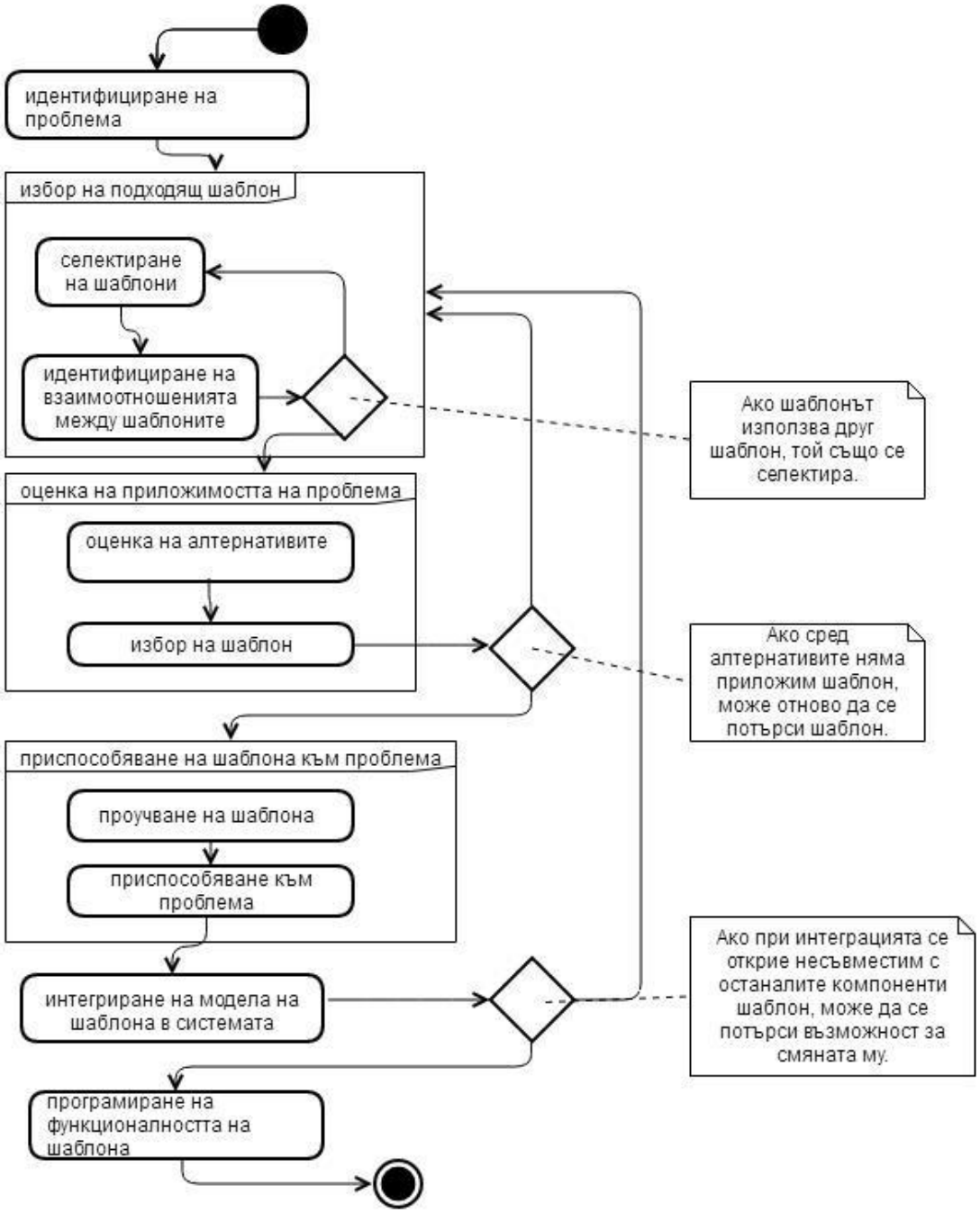
ISSN: 2367-7384
Issue 4, pp. 62-75, December, 2017

- Идентифициране на проблема
- Избор на подходящи шаблони
  - Селектиране на шаблони
  - Идентифициране на взаимоотношенията между шаблоните
- Оценка на приложимостта на шаблона
  - Оценка на алтернативите
  - Избор на шаблон
- Приспособяване на шаблона към проблема
  - Проучване на шаблона
  - Приспособяване към проблема
- Интегриране на модела на шаблона в системата
- Програмиране на функционалността на шаблона

Когато се добавят и под стъпките се получава цялостна диаграма, на дейностите на подхода. Представена е на фиг. 2.

Предложният подход се съчетава с останалите дейности в етапите на разработка на информационните системи. Първата стъпка от подхода е идентифицирането на проблема, за чието решаване се търси подходящ шаблон. Проблемите се идентифицират на различни етапи от разработката на системата (на етап анализ, проектиране или програмиране) и подходът може да се използва за откриване на шаблон от съответното архитектурно ниво.

ISSN: 2367-7384
Issue 4, pp. 62-75, December, 2017



Фиг. 2. Диаграма на дейностите за последователността на стъпките в подхода за разработване на шаблони.

На етап анализ въз основа на изискванията се разработва бизнес модел, който е независим от програмната реализация. Той описва единствено бизнес логиката на приложението. Подходящо е да се използва UML (Филипова, 2005, с. 216-261). При изграждането на информационната система се създава бизнес модел, за да се разбере в по-голяма дълбочина дейността, подлежаща на автоматизиране (Филипова, 2009). При създаването му могат да се идентифицират проблеми, касаещи корпоративната архитектура на информационните системи.

Проблеми могат да се открият и на етап проектиране, когато се разработва модел на приложението. Бизнес моделът на приложението се детайлизира, за да се създаде

ISSN: 2367-7384
Issue 4, pp. 62-75, December, 2017

функционално пълен модел на приложението. Той се декомпозира на отделните компоненти на системата. Реализацията на всеки може да се разглежда, като потенциален проблем, чието решение се оптимизира с използването на шаблони за проектиране. Така на всяко ниво от декомпозицията на системата може се открие потенциален проблем, за който да се потърси подходящ шаблон.

Втората стъпка от подхода е избор на подходящ шаблон за решаване на конкретния проблем. При нея се селектират потенциално подходящите за решаването на проблемите шаблони и се определят взаимоотношенията им.

Съществуват два различни подхода за реализацията на процеса за подбор на шаблони. Единият е отдолу-нагоре (bottom-up), а другият отгоре-надолу (top-down). Освен това съществуват и автоматизирани средства за откриване на шаблони, например ANTLR (ANother Tool for Language Recognition за C++ или Java) (Freitas, 2004), който чрез последователност от задаване на въпроси на разработчика препоръчва подходящи шаблони. Повечето такива средства обаче са разработени само за шаблоните на GoF и не включват цялото многообразие от възможни шаблони.

Традиционният метод за откриване на шаблоните отдолу-нагоре предполага създаване на модел на системата или проблемния компонент и на база на взаимодействието на елементите на модела, да се потърси шаблон, който да е близък по структура до разработения модел. На база познанията си разработчикът открива съответствие между елементите на UML диаграмите на системата и на шаблона. При това разпознаване на шаблона има опасност структурата на модела на системата, просто да прилича на структурата на шаблона, но да няма същата цел. Освен това е възможно разработчикът да не познава добре шаблоните и като резултат погрешно да избере неподходящ или да не може да открие подходящ, защото не го познава достатъчно добре, за да го разпознае. Донякъде тези проблеми могат да се избегнат с разработените инструментални средства за разпознаване на шаблони в модели. Такива са MARPLE (Metrics and Architecture Reconstruction plug-in for Eclipse) (Arcelli, 2008) , DesPaD на Орук (Oruc, 2016, с. 115-121) и др. Проблемът е, че тези инструменти са предназначени единствено за откриване на шаблоните на GoF. Затова този метод е приложим за ограничено множество от шаблони – на GoF, или ако се откриват от разработчика, само до шаблоните, които познава в детайли.

Нека да видим как се прилага този метод при разработване на модели, базирани на обектноориентираната парадигма. Започва се с определяне на класовете. Те се идентифицират според поставените изисквания към средата. После се определят отговорностите и поведението на всеки клас. На тази база се дефинират методите им. Следващият момент е откриване на взаимодействията между класовете. За целта могат да се използват интерактивните, колаборативните или диаграмите на последователността на метода UML, за да се представи взаимодействието между класовете и обектите. После се определят групите от класовете. Според Рам (Ram, 1997) за група се определят шаблоните които са свързани помежду си. Един клас може да принадлежи на повече от една група. След това се определя взаимодействията между класовете в групите с оглед на функционалността, която се постига от групата. Най-подходящи са диаграмите на последователността на UML, които представят в абстрактен вид само съществените взаимодействия.

Следва откриването на шаблоните, като се съпоставят диаграмите на шаблона с абстрактните диаграми на групите от класове. Разработчикът определя подходящи шаблони, които да съответстват на целите на групите от класове. На тази основа се усъвършенстват диаграмите на тези групи така, че да се получат екземпляри на съответните шаблони. После се откриват възможните недостатъци и тесни страни от

ISSN: 2367-7384
Issue 4, pp. 62-75, December, 2017

използването на шаблоните. За да се преодолеят проблемите, те се модифицират, или се предпочита друг шаблон. След това се преминава към детайлно разработване на модела на системата. Тъй като шаблоните са абстрактни, се налага да се добавят всички детайли за съответната система, за да се получи пълен модел.

Този метод обаче не отговаря на всички определени условия за успешното прилагане на шаблоните. Има опасност особено за начинаещите разработчици да прекалят с използването на шаблоните. Често се стремят да открият всички възможни шаблони в модела, а не търсят начин за решаване на конкретен проблем. Това може да е причина използването на шаблоните да не подобри разработката. Считаме, че този подход не е подходящ и предпочитаме другия подход отгоре-надолу.

Такава процедура е предложен от Бушман (Buschmann, 2001). При нея се започва с определянето на проблема, за който се търси шаблон. Под проблем се разбира проектна ситуация, за която се търси решение. За да може да се открие подходящ шаблон обаче проблемът трябва да е строго дефиниран, т.е. ясно да са поставени всички изисквания и ограничения към решението. Ако основният проблем има няколко различни аспекта, се налага разделянето му на подпроблеми. Поотделно се дефинира всеки подпроблем и влияещите му фактори. Процесът се повтаря итеративно, докато се дефинират ясно определени подпроблеми. За решаването на всеки от тях може да се търси шаблон измежду колекциите с подобна функционалност и от същото ниво на абстракция. По този начин се ограничава претърсваното множество от шаблони. След като се набележат подходящите за проблема шаблони, те трябва да се проучат подробно.

Тази процедура се вписва в дефинирания подход за прилагане на шаблоните при разработка на информационна система. Те трябва да се използват в съответствие с контекста, за който са разработени, а той се променя, когато се съчетае с другите цели на сложните системи. От тази гледна точка започвайки от отгоре-надолу с цялостен бизнес модел на системата е по-лесно да се открие точния контекст на всеки компонент, за който се използва шаблон.

Разработчикът започва да търси шаблони измежду публикуваните каталози, които се отнасят до средата, в която ще работи приложението. Целта е да се открият всички шаблони, които са приложими за решаването на даден проблем.

На втора стъпка от подхода за използването на шаблоните се налага да се определят и взаимоотношенията между шаблоните, тъй като те не са изолирани, а се съчетават с останалите елементи на системата. В рамките на каталога те са описани с термини, като свързани, алтернативи или участващи в състава на друг шаблон и т.н. Към пряко избраните от разработчика шаблони се добавят и шаблоните, които се използват в съчетание с тях. За тях също се разглеждат взаимоотношенията с останалите. Освен това се определят и шаблоните, които са несъвместими.

Третата стъпка от подхода е оценка на приложимостта на шаблона. За целта се преценяват алтернативите и се прави окончателния избор на шаблоните, които се използват в системата. Има две възможности за протичането на тази стъпка в зависимост от типа на изградената информационна система. Когато компонентите ѝ са малки и относително самостоятелни (например се реализират, като микро услуги в облачна среда) тази стъпка се прилага само до един компонент на системата.

След като се изберат шаблони за компоненти, за които това е възможно, се изследват и възможните конфликти между компонентите на системата, внесени от използваните в тях шаблони. Независимостта на компонентите предполага избягване на подобни проблеми, но ако се открият, може да се повторят стъпките по селектиране и избор на шаблон до

ISSN: 2367-7384
Issue 4, pp. 62-75, December, 2017

отстраняването на конфликтите.

Обаче разработването на система в повечето случаи предполага създаване на сложни компоненти с много взаимосвързани шаблони в тях. Подходящ е друг процес за избор на шаблон. Трета стъпка започва след като на предходната се създаде списък от всички шаблони, които могат да решат някакъв проблем на системата, заедно със свързаните с тях и алтернативите им. Тази стъпка има за цел да открие конфликтите между шаблоните решаващи различни проблеми на системата. Отбелязват се всички шаблони, които не могат да се прилагат съвместно. Проверява се за алтернативни решения на проблемите с други непротиворечиви шаблони. Ако няма такива разработчикът трябва да прецени, кои проблеми са по-важни. За част от проблемите може да се търси индивидуално решение. Итеративно могат да се променят избраните шаблони, докато се отстранят конфликтите в цялата система.

Като резултат се получава списък от всички шаблони, които са подходящи за съвместно решаване на проблемите на системата.

Четвъртата стъпка от подхода е приспособяване на шаблона към проблема. За целта разработчикът трябва да се вникне в документацията на шаблона. Това се налага, тъй като приехме хипотезата на Филинг, че не е необходимо разработчиците предварително да познават детайлно шаблоните, а го правят, при възникване на проблем, който могат да решат с тях. Това е принцип на гъвкавите методологии.

Прилагането на шаблон към конкретната система изисква да се пригоди към нейните изисквания. Приспособяването на шаблона може да изисква промени в оригиналната му структура. Например може да е необходимо да се намали оригиналната гъвкавост на шаблона. Ако изискванията се променят, например се изисква увеличаване на гъвкавостта, по оригиналната документация на шаблона, гъвкавостта може лесно да се възстанови, като шаблонът се върне към оригиналния си вид.

При проектирането на системата за всеки компонент се разработват модели, базирани на определен език за моделиране. По принцип е препоръчително да е UML, тъй като документацията на повечето шаблони е разработена чрез него. За да стане моделът на шаблона, част от моделите на системата е нужно да се съобрази със спецификите на средата на работа на системата. След всяка трансформация на модел на шаблон, трябва да се следи за качеството му. За целта трябва да се провери дали шаблонът запазва принципната си функционалност след въвеждането на спецификите.

Петата стъпка е интегрирането на отделните модели на шаблоните в пълен модел на системата. Могат да се разработят различен брой модели, ако се използват различни инструменти. Отново се проверя за съвместимост между компонентите на системата. Самостоятелни разработените елементи трябва да се съгласуват с вложените шаблони. Ако това е невъзможно се налага да се потърси друг шаблон, т.е. да се върнем към втора стъпка от подхода, или самостоятелно да се разработи елемента.

Шестата стъпка е програмиране на функционалността на шаблона. Той може да се кодира на подходящия за системата програмен език.

Има разработени програмни средства за генериране на код на база на модели основани върху шаблони. Адмодисастро прави сравнителен анализ на няколко такива инструмента (Admodisastro, 2002, с. 94-101). Проблемът е, че почти всички съвременни инструменти са ориентирани към шаблоните на GoF, така, че в общия случай са неприложими. Разработчиците трябва сами да напишат кода, опирайки се на моделите и примерната реализация представена в документацията на шаблона. Създаването на кода се улеснява, ако се използва примерният код на шаблоните. Макар за част от дейностите да има

ISSN: 2367-7384
Issue 4, pp. 62-75, December, 2017

средства за разработка, е необходимо екипът от разработчици да реализира специфичните изисквания към кода на шаблона и да интегрира отделните елементи, реализирани от различни шаблони в една цялостна система. Както и изцяло сам да разработи кода на елементите, за които няма подходящ шаблон.

Определяне на мястото на стъпките на подхода при създаването на моделите на информационната система

За да стане обаче предложената последователност от стъпки подход е нужно да се открие как прилагането на шаблона се вписва в етапите и моделите на системата. Целта е предложените стъпки да се впишат в етапите на жизнения цикъл на системите, за да може подходът да се включи в методологиите, базирани на подхода UML.

На фиг. 3 е показана последователността на създаването на моделите на системата, което да позволи използването на шаблоните.

Разработката на приложението започва със създаването на бизнес модел, който отразява изискванията към системата. За разработването му могат да се използват шаблоните на Фаулър (Fowler, Rice, 2002), (Fowler, 1996). Тези шаблони са свързващото звено между концептуалните шаблони и софтуера, т.е. те превръщат концептуалните/бизнес модели в софтуер. Те показват как се включва този софтуер в архитектурата на голяма информационна система. Подпомагат етапа на анализ с различни бизнес модели, позволяващи по-пълно отразяване на бизнес, системните и технологичните изисквания. Участват и в проектирането, като предлагат примерни модели за конкретната предметна област. Ако няма подходящ шаблон разработчиците сами трябва да преобразуват бизнес моделите (Филипова, 2010).

За разработката на архитектурата на системата могат да се ползват различни архитектурни шаблони, например предложените от Еле и Крипс (Eeles, 2009). Повечето каталози от шаблони са за съответна предметна област и включват както архитектурни, така и шаблони за проектиране. Например каталозите на Филинг за облачните системи (Fehling, 2014, 2012), Леа и Шмид (Lea, 1999), (Schmidt, 2000) за мрежовите системи или на Волтер (Volter, 2002) за компонентите на сървърите и т.н.

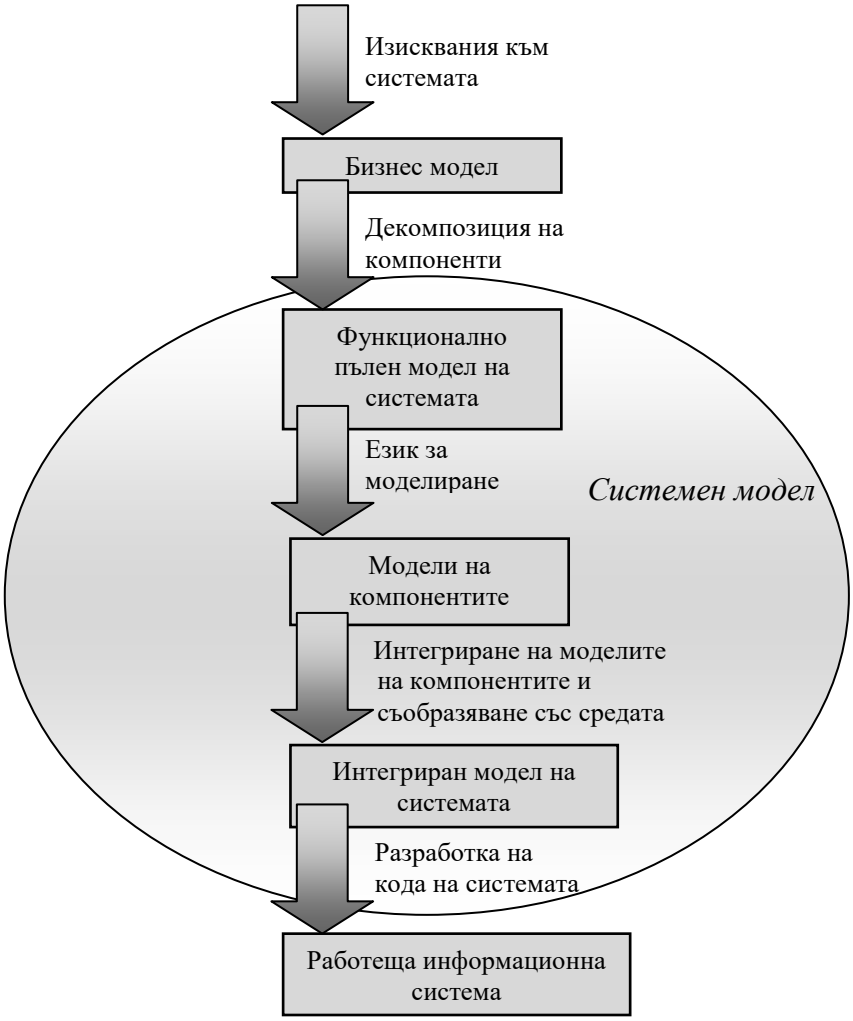
След това на етап проектиране се разработва модел на системата. За да се детайлизира разработеният бизнес модел, първо се определят основните елементи, които трябва да реализира. На тази база системата може да декомпозира на отделни компоненти, за да се създаде функционално пълен модел. Декомпозиция може да се определи от използвания архитектурен шаблон.

След това за всеки определен системен елемент, се разработва модел, базиран на определен език за моделиране. По принцип е препоръчително да е UML, тъй като документацията на повечето шаблони е разработена чрез него. Ако моделите са разработят чрез UML е много по-лесно да се направи съпоставка между наличните шаблони и нуждите на системата, за да се избере най-подходящият шаблон за дадена разработка. Откриването на подходящ шаблон би трябвало да следва втората, трета и четвъртата стъпка от представения подход. Ако такъв не е наличен, то съответната функционалност се проектира чрез UML модели.По-този начин се създават модели на всички компонентите на приложението, които са независими от реализацията си. Те са част от системния модел.

Нужно е отделните модели да се интегрират според избраната архитектура да се съобразят със спецификите на средата на работа на системата. Така чрез избор на подходящи инструменти и конкретни шаблони, които ще се използват в бъдещата разработка, могат пълно да се въведат особеностите на технологията. Получава се пълна съвкупност от модели за етапа на проектиране.

ISSN: 2367-7384
Issue 4, pp. 62-75, December, 2017

Има разработени инструменти (Tankovic, 2010) за трансформацията на един модел в друг или пък за генериране на програмен код от модел. Подобни инструменти се основават на дефиниции за трансформация, определени от правилата за трансформации. Обаче те не могат да заменят разработчика, а само облекчават работата му.



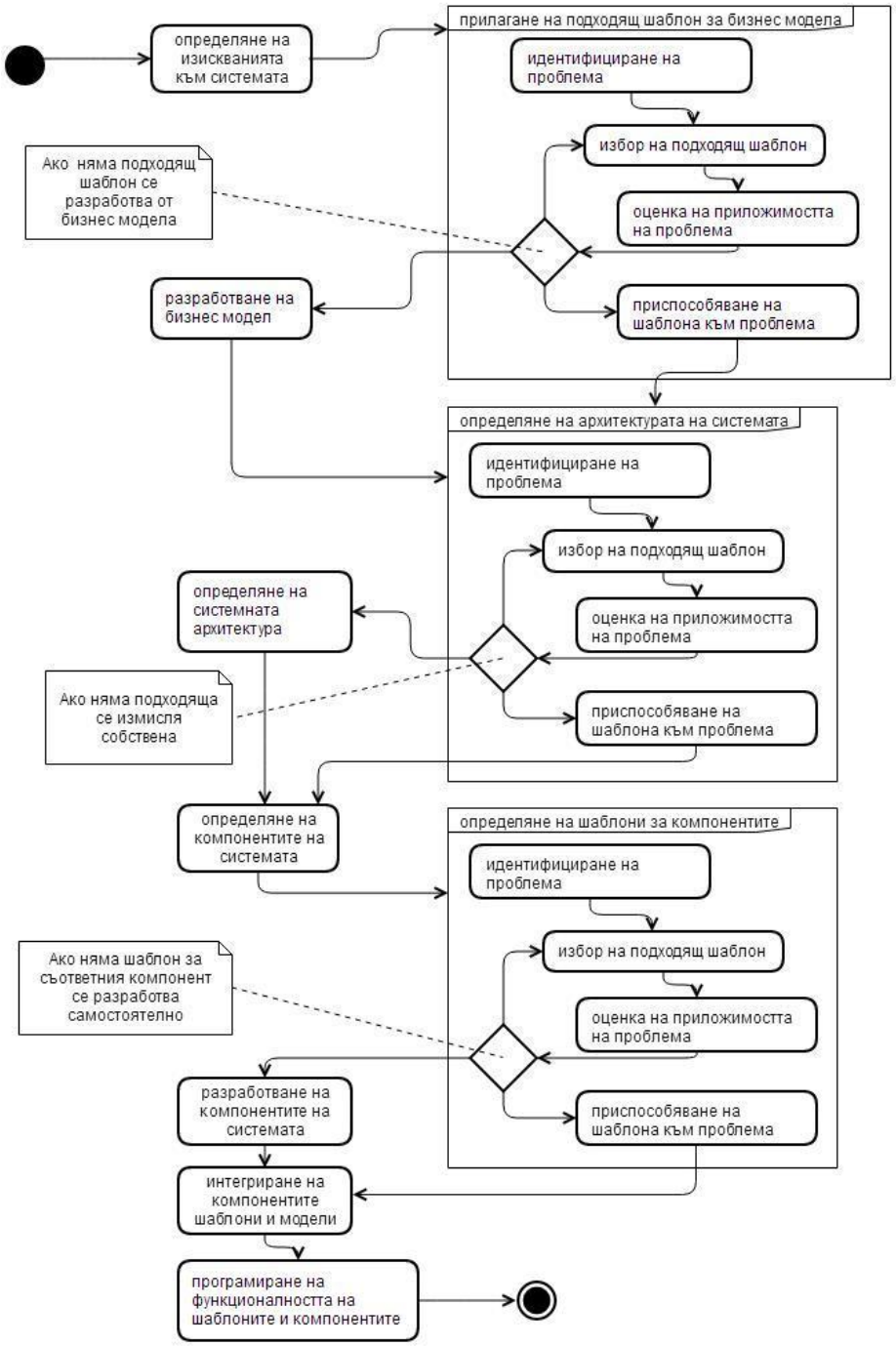
Фиг. 3. Последователност на моделите при разработка на информационна система с помощта на шаблони.

На етап програмиране се изпълнява последната стъпка от предложения подход и моделите се кодират с помощта на указанията в шаблоните. Получава се прототип на информационната система. След като е избран езикът за програмиране, за реализацията на определени функции и алгоритми могат да се ползват програмнозависимите шаблони.

Шаблоните участват и в етапа на внедряване и на съпровождане, като подпомагат внасянето на промени. Те са добре дефинирани и дори за някои от тях има описани различни вариации, които могат да се използват при промяна в средата на работа или изискванията към системата.

Прилагане на шаблоните в процеса на разработката зависи само от уменията и компетентността на специалистите. За да прерасне използването им в повтаряем процес, който гарантира определено ниво на качество, е нужно да се въведе някаква степен на формализация на процеса. Нужно е резултатите от процеса по прилагането им да намерят място сред моделите на системата. Затова на фиг. 4 е показана примерна последователност на стъпките на подхода при създаването на моделите на системата.

ISSN: 2367-7384
Issue 4, pp. 62-75, December, 2017



Фиг. 4. Диаграма на дейностите за процеса на разработка с прилагане на шаблони.

Дефинираният подход има за цел да представи последователност от стъпки, които да позволят въвеждане на някаква формална основа за процеса и да спомогнат за преодоляването на някои от проблемите при прилагането им. Той спомага за използването на шаблоните при решаване на конкретен проблем, а не като самоцел. Не изисква от разработчиците подробно познаване на огромното множество от шаблони, а само при необходимост, което е и основен принцип на гъвкавите методологии. Още повече, че голямото количество шаблони е пречка пред опознаването и прилагането им. Подходът позволява шаблоните да се ползват като средство за обучение и въвеждане в особеностите на новите технологии, тъй като при проблем разработчикът проучва заложения в шаблоните експертен опит.

ISSN: 2367-7384
Issue 4, pp. 62-75, December, 2017

Представеният шаблон за проектиране на двустъпков изчислителен процес е разработен самостоятелно от автора. Той е резултат от проучване на проблемите в областта на облачните системи, паралелните обработки и шаблоните. Проучени са съвременните технологии за разработка, реализация и поддържане на софтуерните системи. Шаблонът подпомага реализирането на продължителните обработки за приложения, работещи в облачна среда или базиращи се на многонишковото програмиране.

Използването на шаблоните намалява разходите, времето и трудността при разработката на системите. Създадените с тях приложения са с гарантирано ниво на качество, по-дълъг жизнен цикъл, гъвкави, преносими, могат динамично да се мащабират и персонализират според нуждите на потребителите.

Предложеният подход цели да подпомогне процеса на ефективно използване на шаблоните и да позволи включването им в създаването на последователността от модели на разработваната система. Той показва последователността от стъпките при прилагането им. Той поставя на формална основа процеса и спомогна за преодоляването на някои от проблемите при използването на шаблоните. Разгледан е начин за вписване на шаблоните в последователно създаваните модели на системата, за да може подходът да се включи в методологиите, базирани на подхода UML.

4. СЪТРУДНИЧЕСТВО

Диаграмите са разработени с безплатна среда Gliffy. Шаблонът е апробиран на облачната платформа на Google, в рамките на безплатно предоставеното облачно пространство и софтуер.

5. ДОПЪЛНИТЕЛНИ МАТЕРИАЛИ

Подходът е апробиран за разработката на облачна информационна система за анализ на клиентите при маркетингови кампании и част от кода на отделните микросървиси е изпратен в приложение1.docs.

6. ЛИТЕРАТУРА

Статия

Admodisastro, N. I., Palaniappan, S. (2002). A code generator tool for the Gamma design patterns. Malaysian Journal of Computer Science. Vol. 15 No. 2, December 2002, pp. 94-101.

Arcelli, F., Tosi, C., Zanoni, M., Maggioni, S. (2008). The marple project - a tool for design pattern detection and software architecture reconstruction. in Proceedings of the International Workshop on Advanced Software Development Tools and Techniques (WASDeTT 2008). Paphos. Cyprus.

Fehling, C., Ewald, T., Leymann, F., Pauly, M., Schumm, D. (2012). Capturing Cloud Computing Knowledge and Experience in Patterns.. In IEEE 5th International conference on cloud computing. pp 726-733. IEEE inc.

Freitas, A.,L. (2004). The Essence of Design Patterns In Automatic Identification Tool, IV Congresso Brasileiro de Computação – CBComp. Engenharia de Software.

Kumar, R., Bopaiah, J., Jain, P., Nalini N., Sekaran, Ch. (2013). Model Driven Approach For Developing Cloud Application. International Journal of Scientific & Technology Research. v.2. iss. 10.

Odenthal, G., Quibeldey-Cirkel, K. (1997). Using patterns for design and documentation. ECOOP'97. LNCS#1241. pp 511-529.
Oruc, M., Akal, F., Sever, H. (2016). Detecting Design Patterns in Object-Oriented Design Models by Using a Graph Mining Approach. 4th International Conference in Software Engineering Research and Innovation (CONISOFT). pp. 115-121.
Tankovic, N. (2010). Model Driven Development Approaches: Comparison and Opportunities www.fer.unizg.hr/_download/repository/N.Tankovic,_rad_za_KDI.pdf [21.09.17]

Книга

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal. (2001). M., Pattern-Oriented Software Architecture – A System of Patterns, John Wiley & Sons. 2nd Edition.
Coplien, J. O., Schmidt, D. C. (1995). Pattern Languages of Program Design. Addison-Wesley Professional.
Eeles P., Cripps, P. (2009). The Process of Software Architecting. Addison-Wesley.
Fehling, C., Leymann, F., Retter, R., Schupeck, W., Arbitter, P. (2014). Cloud computing patterns. Springer-Verlag Wien. ISBN 978-3-7091-1567-1.
Fehling, C., Barzen, J., Breitenbücher, U., Leymann, F. (2015). A Process for Pattern Identification, Authoring, and Application. Institute of Architecture of Application Systems. University of Stuttgart, Germany.
Fowler, M. (1996). Analysis Patterns. Addison-Wesley Professional.
Fowler, M., Rice, D., Foemmel, M., Hieatt, E., Mee, R., Stafford, R. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley.
Lea, D. (1999). Concurrent Programming in Java, in Concurrent Programming in Java: Design Principles and Patterns. Second Edition. Addison-Wesley.
Mili, H., Mili, A., Yacoub, S., Addy, E. (2002). Reuse-Based Software Engineering: Techniques, Organization, and Control. John Wiley & Sons. ISBN 0-471-39819-5.
Ram, Anantha,Guruprasad. (1997). A Pattern-Oriented Technique for Software Design.
Schmidt, D. C., Stal, M., Rohnert, H., Buschmann, F. (2000). Pattern-Oriented Software Architecture – Patterns for Concurrent and Networked Objects. John Wiley & Sons.
Volter, M., Schmid, A., Wolff, E. (2002). Server Component Patterns – Component Infrastructures Illustrated with EJB. John Wiley & Sons.
Гама, Е., Хелм. Р., Джонсън, Р., Влисидес, Дж. (2004). Шаблони за дизайн. СофтПрес.
Филипова, Н. (2005). Възможности на обектноориентирания метод UML за информационно бизнес моделиране. Годишник ИУ-Варна. 216-261с. ISSN: 08616752.
Филипова, Ф., Филипов, Ф. (2009). Бизнес моделиране, изд. Наука и икономика, ИУ-Варна.
Филипова, Н., Филипов, Ф. (2010). Насоки за преобразуване на UML бизнес моделите. Известия ИУ-Варна. 31-45с. ISSN: 13100343.

7. СЪКРАЩЕНИЯ

1. UML - Unified Modeling Language, специализиран графичен език за моделиране

2. GoF – Gang of Four, голямата четворка – Гама, Хелм, Джонсън и Влисидес (Гама, 2004)