



СОФИЙСКИ УНИВЕРСИТЕТ “СВ. КЛИМЕНТ ОХРИДСКИ”
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

маг. Деница Цветанова Манова

МЕТОДОЛОГИЯ ЗА ТЕСТВАНЕ НА КОМПОЗИЦИИ ОТ УЕБ УСЛУГИ

АВТОРЕФЕРАТ

на дисертация за придобиване на образователна и научна степен “Доктор”,
професионално направление 4.6 “Информатика и компютърни науки”
(докторска програма „Софтуерни технологии - Тестване на софтуерни системи“)

Научен ръководител:

доц. д-р Десислава Петрова-Антонова

София

2017

Дисертационният труд е с обем 236 страници и се състои от списък на термините и съкращения, увод, четири глави, заключение, списък на използваната литература и приложения.

Дисертацията съдържа 50 фигури, 24 таблици и 2 приложения. Библиографията включва: 82 заглавия и 24 интернет адреса.

Дисертационният труд е обсъден в катедра „Софтуерни технологии” при Софийски университет „Св. Климент Охридски“ и е насочен за защита пред научно жури.

Авторката на дисертационния труд е била докторантка на самостоятелна подготовка към катедра „Софтуерни технологии” при Софийски университет „Св. Климент Охридски“.

Материалите по защитата са на разположение на заинтересованите лица в Учебен отдел на ФМИ и на интернет страницата на Софийски университет „Св. Климент Охридски“ и на Факултет по математика и информатика (<https://www.uni-sofia.bg/> и https://www.fmi.uni-sofia.bg/habil_disert_trudove).

Съдържание

Увод.....	3
Актуалност на проблема.....	3
Предмет и работни хипотези на научното изследване.....	4
Цел и задачи.....	4
Методи на изследване.....	4
Приложимост и полезност.....	4
Обем и структура на дисертацията.....	5
Основно съдържание на дисертационния труд.....	5
1. Анализ на предметната област.....	5
1.1 Типове тестване на приложения, реализиращи АОУ.....	5
1.2 Стратегия за извършване на систематизирано проучване на предметната област.....	8
1.3 Резултати от систематизираното проучване.....	10
1.4 Изводи.....	14
2. Методология за тестване на съставни услуги.....	14
2.1 Концептуално описание на методология за тестване на съставни услуги.....	15
2.2 Подходи за тестване на съставни услуги, подкрепящи методологията.....	20
2.3 Приложение на предложената методология.....	24
2.4 Изводи.....	24
3. Разработени подходи.....	24
3.1 Подход за изолиране от външни зависимости.....	24
3.2 Подход за инжектиране на неизправности.....	25
3.3 Подход за създаване на функционални тестове.....	26
3.4 Подход за анализ на даннови зависимости.....	26
3.5 Подход за генериране на тестови данни.....	28
3.6 Изводи.....	29
4. Валидация на подходите и методологията.....	30
Заклучение.....	32
Приноси.....	33
Публикации в тематиката на дисертационния материал.....	34
Цитирания.....	35
Благодарности.....	36
Библиография.....	37

Увод

Актуалност на проблема

Намаляването на разходите и увеличаването на приходите е в основата на успешни предприятия. За да се постигне това, се изискват гъвкави и бързи връзки между вътрешните системи и външни бизнес партньори. Информационните технологии позволяват такива бизнес взаимодействия чрез базирани на стандарти, оперативно съвместими и многократно използвани интерфейси за създаване на приложения. Тези модели, базирани на стандарти, за многократно употреба са в основата на Архитектурите ориентирани към услуги (АОУ). Те са една от най-популярните парадигми за изграждане на разпределени системи. Софтуерните решения, базирани на тях, са създадени, за да удовлетворят бизнес цели, включващи лесна и гъвкава интеграция със съществуващите системи, опростяване на бизнес процесите, предлагане на иновативни услуги за клиентите, редуциране на цената, гъвкава адаптация и реакция на възможностите и конкуренцията на пазара.

АОУ може да се дефинира като архитектурен стил, при който системите се състоят от потребители и доставчици на услуги. Самите уеб услуги може да се разглеждат като повторяеми бизнес задачи. Техните основни приложения са представени като компоненти, които се повтарят често и се използват от много системи и за осъществяване на връзка между системи, които иначе са оперативно несъвместими. Отделните уеб услуги имат много предимства, но още по-големи възможности предлагат съставните услуги, т.нар. композиции от уеб услуги. Те свързват повече от една услуга и използват нейните функции в комбинация с други услуги. За описание на взаимодействието между базовите услуги, изграждащи дадена съставна услуга, организацията OASIS публикува стандарта WS-BPEL [1]. Той дефинира модел и граматика за описание на оркестрацията на уеб услуги.

Големият потенциал на АОУ, в частност на съставните уеб услуги и многото предимства, които предлагат, са предпоставка за тяхното все по-широко прилагане. В следствие се появява и необходимостта от осигуряване на тяхното качество и свързания с това процес на тестване. Това обаче е трудоемка задача, свързана с редица предизвикателства. От гледна точка на АОУ инфраструктурата, предизвикателствата са свързани с ограничена техническа организация относно компонентите, сложна конфигурация на инфраструктурата, кратък цикъл на пускане на компонентите в инфраструктурата, липса на унифициран поглед и различия върху инфраструктурата. От гледна точка на уеб услугите тестването се усложнява поради липса на програмния код на уеб услугата, липса на информация относно контекста и средата на приложение, непредвидимо влияние на качеството на услугата, не добра съвместимост със стандарти. В допълнение към предизвикателствата за тестване на отделни услуги са налице и предизвикателства за тестване на композиция от услуги, които са свързани с промени в базовите услуги, координация на множество доставчици, липса на общ модел на данните и грешките, странични ефекти, липса на софтуерни инструменти и др.

За решение на някои от посочените предизвикателства е възможно да се използват конвенционални подходи и техники за тестване, за други случаи – трябва да се допълнят, променят или изцяло да се изготвят нови такива.

Множеството недостатъци и липсата на единен тестов процес, пораждаат необходимостта от стъпки, насоки, отговори, инструменти за тестването на АОУ.

За да отговори на тази необходимост дисертационния труд проучва, предлага и прилага цялостна методология и софтуерна платформа за тестване, съобразени със спецификата на АОУ и в частност на композициите от уеб услуги.

Предмет и работни хипотези на научното изследване

Обект на научното изследване са АОУ и в частност композиции от услуги.

Предмет на научното изследване са методологии, подходи и софтуерни инструменти за тестване на приложения, реализиращи АОУ.

В контекста на обекта и предмета на научното изследване е дефинирана следната основна хипотеза:

Хипотеза 1: Възможно е да се разработи методология за тестване на композиции от веб услуги, която систематизира всички ключови дейности в тестовия процес на системите, базирани на композиции от услуги.

В допълнение са формулирани следните хипотези, които подпомагат доказването на основната:

Хипотеза 2: Всички дейности, включени в методологията, могат да бъдат изпълнени чрез прилагане на конкретни подходи и техники за тестване.

Хипотеза 3: Възможно е автоматизиране на базовите дейности от предложената методология чрез набор от софтуерни инструменти и тяхното интегриране в единна платформа.

Цел и задачи

Целта на дисертационния труд е да се разработят методология и подкрепяща я платформа за тестване на съставни веб услуги. Тя изисква от една страна разработване на група от подходи за тестване на АОУ, и в частност на композициите от веб услуги, а от друга – реализирането на набор от софтуерни инструменти за автоматизиране на тестовия процес и последващо интегриране на тези инструменти в единна платформа.

Постигането на поставената цел изисква решаване на следните научноизследователски задачи:

- **Задача 1:** Проучване и сравнителен анализ на съществуващите методологии, подходи и софтуерни платформи за тестване на приложения, реализиращи архитектура, ориентирана към услуги;
- **Задача 2:** Изграждане на методология за тестване на композиция от веб услуги;
- **Задача 3:** Дефиниране на подходи за функционално тестване на композиция от веб услуги, които покриват базовите дейности от тестовия процес и разработване на прототипи на софтуерни инструменти, реализиращи предложените подходи;
- **Задача 4:** Валидиране и оценка на подходите;
- **Задача 5:** Валидиране на методологията в контекста на предложените подходи и платформа за тестване.

Методи на изследване

За изследването и анализирането на текущото състояние на предметната област (**Задача 1**) е използвано систематичното проучване по метода на Kitchenham [2], който е специално адаптиран за нуждите на софтуерните технологии и предоставя препоръки и насоки как да бъде проведено то. Следвайки ги, са специфицирани и дефинирани въпроси/задачи, които са обект на проучване в дисертацията, критерии за включване/ изключване на информацията, стратегия за търсене. Също така са извадени, синтезирани и описани намерените текстове в систематичното проучване.

За валидирането на предложената методология и подходи за тестване на композиция от веб услуги (**Задача 5**) е използван експерименталният метод, като са въведени критерии и метрики за оценка. На база отчетените резултати от експериментите и подбраните критерии е извършена съответната оценка.

Приложимост и полезност

Предложена в дисертацията методология е фокусирана върху тестване на композиции от веб услуги, описани чрез езика BPEL. Тя и подкрепящите я подходи са най-подходящи за тестване на качеството на бизнес процеси, реализирани с този стандарт. Методологията е подходяща и за

бизнес процеси, описани със сходни на WS-BPEL стандарти, единични уеб услуги, микроуслуги, някои модели облачни структури и други разпределени и слабо свързани системи.

Обем и структура на дисертацията

Дисертационният труд е структуриран в шест глави. **Уводът** представя кратко въведение в предметната област, актуалността и предизвикателствата на научното изследване.

Във **Втора глава** са разгледани типовете тестване, използвани в АОУ и са класифицирани литературните източници в изследваната област. Направено е систематично проучване на научните материали относно съществуващите методологии, подходи и инструменти за тестване на приложения, реализирани чрез уеб услуги (Задача 1).

В резултат на това е извършен обзор и сравнителен анализ на съществуващите решения в областта и са направени изводи.

В **Трета глава** е разработена нова методологията за тестване на композиции от уеб услуги (Задача 2). Разгледани са, класифицирани са и са анализирани подходи за тестване на съставни уеб услуги, които подкрепят методологията. Идентифицирани са най-подходящите за използване в предложената методология. В тази глава са описани и типовете тестване и видовете архитектури, в които разработената в дисертацията методология намира приложение.

В дисертацията са синтезирани нови подходи за тестване на съставни уеб услуги, които покриват основни дейности от предложената методология и са разработени прототипи на софтуерни инструменти за тях (Задача 3). Те са представени подробно в **Четвърта глава**.

В **Пета глава** са валидирани предложените подходи заедно със съпътстващите ги прототипи на софтуерни инструменти (Задача 4) и разработената методология за тестване на бизнес процеси, описани чрез езика BPEL (Задача 5). Дефинирани са критерии и метрики за оценка на методологията и реализираните подходи. Проведени са експерименти и на база на резултатите от тях са направени изводи за качествата на предложената в дисертацията методология и подходи за тестване на композиция от уеб услуги.

Заклучението обобщава основните изводи от осъщественото проучване и резултатите от създадената методология и подходи за тестване. Определени са насоки за бъдещо развитие. Изведени са научните и научно-приложните приноси на дисертационния труд и е предоставен списък с публикациите върху него.

В дисертацията са включени Съдържание, Речник на термините и съкращенията, Декларация за оригиналност, Приложения и Списък с използвани литературни източници.

Основно съдържание на дисертационния труд

1. Анализ на предметната област

В тази глава на дисертацията са проучени и анализирани съществуващите методологии, инструменти и платформи за тестване на АОУ във връзка със Задача 1.

Разгледани са типове тестване приложими за АОУ, класифицирани са литературните източници и е извършено задълбочено изследване на предметната област на дисертацията. Намерените, в резултат от направеното проучване, методологии и инструменти за тестване на АОУ са описани, сравнени и анализирани. В последната секция са дадени изводи от направения анализ на предметната област.

1.1 Типове тестване на приложения, реализиращи архитектура, ориентирана към услуги АОУ

Някои от подходи, приложими за традиционните системи могат да се приложат и за системи, базирани на услуги (СБУ). Основната идея е, че е необходима комбинация от тестване на всички нива (модулно, интегрирано и системно) и регресивно тестване за да се провери дали системата

отговаря на поставените към нея функционални изисквания. Независимо от това, динамичният и адаптивен характер на АОУ прави невъзможно директното прилагане на част от познатите методи за тестване.

Основните типове тестване, приложими за АОУ, които имат най-голямо значение за тяхното функциониране и биха могли да се използват в предложената методология са:

- **Модулно тестване** – тестването на единични, атомни услуги, може да се счита за еквивалент на тестването на компонент.
- **Модулно тестване на композиция от уеб услуги** – при тестването на WS-BPEL процеси също може да се използва модулно тестване. От гледна точка на разработчика, който има достъп до BPEL процеса, може да бъде използван методът за тестване като бяла кутия. Ако се гледа на композицията от услуги от външна гледна точка, то може да се използва тестване като черна кутия, тъй като процесът за външния наблюдател ще изглежда като услуга, предоставяща WSDL интерфейс. Такъв тип тестване в изолация обаче рядко се прави, защото това налага симулиране на поведението на всички партньори на процеса. От друга страна, възможността за изпитване на BPEL процес без наличието на услугите - партньори ще даде възможност за тестване още преди партниращите услуги да са разработени или даже без тяхно участие.
- **Интеграционно тестване** - АОУ измества перспективата за разработка от монолитни приложения към приложения съставени от услуги, разпределени по мрежата, които са разработени и се обслужват от различни организации и си сътрудничат на база централизирана оркестрация (напр., описани с помощта на WS-BPEL) или в напълно разпределен начин (както това се случва в партньор към партньор архитектура). Освен това, композициите на услуги може да се променят динамично и да се справят с различни ситуации, като например, липса на услуга, или публикуването на нова услуга с по-добро Качество на услугата (Quality of Service (QoS)). Под QoS се разбира възможността за предоставяне на различен приоритет за различни приложения, потребители, или потоци от данни, или да се гарантира определено ниво на изпълнение на потока от данни). При такъв сценарий е особено важно услугите да се тестват за оперативна съвместимост, т.е. тестване на интеграция/взаимодействие.
- **Регресионно тестване** – този тип тестване е трудно да се извърши при АОУ, поради липсата на контрол върху интегрираните услуги. Архитектът на системата избира уеб услуги и предполага, че те ще запазят своите функционални и нефункционални характеристики. Това обаче рядко се случва – уеб услугите по своята същност са системи, като всички други, които просто са обявени в интернет като услуги и при тях е напълно нормално да търпят промени (обновяване, поправка и др.). Следователно, променената услуга може да повлияе на други услуги, които я използват. Това прави системите, базирани на услуги, различни от компонентно-базирани системи: когато компонент се развива, това не засяга системи, които използват предишни версии на самия компонент. Все пак, този вид тестване е приложим за АОУ, но с по-голяма честота на изпълнение, за да може тестерът до известна степен да е убеден, че тестваната система все още отговаря на поставените към нея изисквания.
- **Нефункционално тестване** – тестването на нефункционалните характеристики на една АОУ е от сериозна важност, тъй като проблеми могат да настъпят в доста аспекти. Липсата на стабилност, на подходящи действия за възстановяване може да доведе до нежелани странични ефекти в композицията от услуги, обект на тестване. По принцип доставчикът на услуга я предоставя на потребителя с определени гаранции за качество (QoS). Когато обаче услугата е предоставена без обвързване с QoS договорите, т.е. без гаранция или клиентът желае да се увери, че са спазени нефункционалните изисквания, които той има

към услугата, то е препоръчително да се извърши тестване на нейните нефункционални характеристики. В СБУ трябва да се обърне внимание на следните видове подходи за нефункционално тестване:

- тестване при натоварване – е обвързано с проверка на времево поведение и работоспособност на системата при натоварване/ стрес . Наблюдават се характеристики като време за отговор, максимална пропускателна способност, наличие, достъпност и успеваемост и се отчита дали отговарят на специфицираните нефункционални характеристики на системата.
- тестване на сигурност – има за задача да провери дали уеб услугите и данните, които обменят са защитени от различни заплахи за конфиденциалност, интегритет и достъпност. За АОУ са дефинирани редица стандарти и механизми за сигурност, които спомагат нейното постигане и подходите за тестване трябва да проверяват за тяхното спазване.
- тестване на политиките на ниво услуги (Service Level Agreement, SLA) обхваща необходимостта за определяне на условия, за които дадена уеб услуга не може да предостави своята функционалност. Преди да предостави услугата на някой потребител, нейният доставчик е добре да ограничи възможностите за нарушения по време на използването ѝ.
- разрушителното тестване, на устойчивост също се използва като подход за тестване на уеб услуги. Проверява се какво е поведението на услугата, при неочаквани входни данни и влияние. Често, обаче възстановяването от грешки не се тества правилно. За да се извърши този тип тестване е необходимо да се предизвика „блокиране“ на услуга, не обработване на изключения, необичайно поведение и др.

ТАБЛИЦА 1 НИВО НА ИЗПОЛЗВАЕМОСТ НА ТЕСТОВИТЕ ПОДХОДИ

Вид [Ф/Н]	Наименование	Ниво на приложение [КУ/СУ]	Недостатъци
Ф	Модулно	КУ, СУ	Комплексни типове данни. Множество допустими стойности. Зависимост от партниращите услуги.
Ф	Интеграционно	КУ	Множество възможни връзки, при динамичното свързване.
Ф	Регресионно	КУ, СУ	Нужда от често повторение на тестовете, поради възможни промени в услугите
Н	Натоварване	СУ	Зависимост от партниращите услуги. Нужда от наблюдение в продължителен период от време.
Н, Ф	Сигурност	КУ, СУ	Липса на видимост върху услугите, предоставени от други доставчици.
Н	SLA	СУ	Зависимост от външно влияние.
Н	Устойчивост	КУ,СУ	Възможен е срив в услугата и невъзможност за възстановяване

В Таблица 1 тестовите подходи са систематизирани според това на какво ниво могат да се използват: композиция от услуги [КУ] или самостоятелни услуги [СУ]. Според вида на тестването могат да се прилагат за функционално [Ф] или нефункционално [Н] тестване.

Наличните подходи не са достатъчни, за да се постигне добро покритие на типовете тестване и тестването в различните фазите на развитие на АОУ, но все още се работи за подобрене в тази област и създадената дисертация допринася именно за това.

1.2 Стратегия за извършване на систематизирано проучване на предметната област

В рамките на дисертационния труд е направено систематизирано проучване, като са следвани процедурите, описани в [2].

Най-важните стъпки, върху които е разработено проучването са:

- Проверка за необходимост от систематично проучване;
- Дефиниране на въпрос/проблем;
- Специфициране какво ще бъде направено, за да се реши проблемът, като се приложат критерии за включване и/или изключване и се вземат предвид всички намерени данни;
- Дефиниране на стратегия за търсене;
- Избиране на информацията, която да бъде извлечена от всяко първично проучване;
- Запазване на списъци с включените и изключените проучвания;
- Използване на напътствията за синтез на данните;
- Използване на насоките за отчитане на резултатите.

Направена е проверка в **електронните библиотеки** дадени в Таблица 2, за да се провери дали съществува вече изследване върху тестването на композиция от веб услуги.

ТАБЛИЦА 2 ЕЛЕКТРОННИ БИБЛИОТЕКИ

Име	Интернет адрес	Кратко описание
IEEE Xplore	http://ieeexplore.ieee.org	Библиотека IEEE Xplore е източник на научни и технически текстове публикувани от IEEE (Institute of Electrical and Electronics Engineers) и нейните партньори.
ACM	http://dl.acm.org	Тази интернет библиотека дава пълен достъп до публикациите на ACM (Association for Computing Machinery), за списания, конференции и бюлетини.
Google Scholar	http://scholar.google.bg	Google Scholar предоставя възможност за търсене на научна литература. Търсенето може да се извърши в много дисциплини и източници: статии, дисертации, книги, резюмета и съдебни становища, от академични издателства, професионални общества, онлайн хранилища, университети и други веб сайтове.

Проверката показва, че няма направен обзор по конкретния проблем на дисертационния труд, а само върху по-широк кръг от областта, към която принадлежи – тестване на АОУ. Ето защо е направено систематично проучване на методологиите за тестване на композиции от веб услуги.

За да се постигнат поставените цели на дисертационния труд, е поставен следният въпрос, който да идентифицира съществуващата основа от знания в сферата на АОУ - **Съществува ли**

методология, която да дава предложения, насоки, отговори, стъпки, инструменти за тестването на композиция от уеб услуги?

На база въведения въпрос се дефинира израз за търсене:

(methodology OR procedure) AND (testing OR test OR quality) AND (composition OR compositions OR BPEL) AND (web services OR webservices OR WS OR services OR service OR SOA),

което се извършва в дигиталните библиотеки отбелязани в Таблица 2.

Изборът на научни текстове свързани с обекта на проучване е извършено следвайки стъпките:

1. Автоматизирано търсене в електронните библиотеки за дефинирания израз;
2. Премахване на еднаквите, подобните резултати;
3. Премахване на публикации, на база тяхното заглавие;
4. Премахване на текстове, на база на съдържанието на резюмето (абстракта), където е възможно (за Google Scholar, не може да се извърши);
5. Търсене за основни фрази от дефинирания израз в програмите за търсене.

Трябва да се отбележи, че друг критерии, по който са селектирани материалите е езикът – търсенето е извършено за материали на английски език.

Резултатът от търсенето и приложените към намерената информация критерии за избор са дадени в Таблица 3:

ТАБЛИЦА 3 РЕЗУЛТАТИ ОТ ТЪРСЕНЕТО И ПРИЛАГЕНЕ НА КРИТЕРИИТЕ ЗА ВКЛЮЧВАНЕ И ИЗКЛЮЧВАНЕ НА НАУЧНИ МАТЕРИАЛИ

	IEEE Xplore	ACM	Google Scholar	Общо
1.Търсене израз	139	82	99	320
2.Без подобни	129	73	74	276
3.Според заглавие	51	20	73	144
4.Според абстракт	20	4	20	44
5.Търсене на всякъде	-	-	-	+12
ОБЩО (които представляват интерес)	56			

Направеното проучване **включва над 300 (триста) текста**, от които чрез прилагане на описаната стратегия са **избрани 56 научни публикации**.

Изграждането на таксономии е широко използван инструмент в науката за класификация на обекти по групи въз основата на определени признаци и критерии и обясняването на това подреждане. За целите на дипломната работа е разработена таксономия на литературните източници (статии, доклади на конференции, отчети от работа по проекти, спецификации на стандарти), във връзка с изследваната област. Литературните източници са класифицирани спрямо таксономията в три групи: базови източници, обзорни източници и специфични източници.

Базовите източници се класифицират в следните категории:

- Технологии и стандарти
 - АОУ
 - Уеб услуги
 - Съставни услуги
- Тестване на АОУ

Обзорните източници представят научноизследователски резултати, получени въз основа на преглед на съществуващото състояние в областта на АОУ. Разгледана е обзорна литература за:

- Предиизвикателства и проблеми при АОУ

- Предиизвикателства и проблеми при тестване на АОУ

В резултат на направеното проучване са идентифицирани следните специфични източници:

- Тестване на АОУ
 - Методологии за тестване на АОУ
 - Софтуерни платформи за тестване на АОУ
- Тестване на композиция от уеб услуги
 - Подходи за тестване на композиция от уеб услуги
 - Инструменти за тестване на композиция от уеб услуги
 - Методологии за тестване на композиция от уеб услуги
- Тестване на уеб услуги
 - Подходи за тестване на уеб услуги
 - Инструменти за тестване уеб услуги
 - Методологии за тестване уеб услуги

В дисертационният труд е използвана и реферирана информация от 106 броя литературни източници (статии, доклади на конференции, отчети от работа по проекти, спецификации на стандарти), които спадат към направената таксономия.

1.3 Резултати от систематизираното проучване

Избраните научни материали са разгледани подробно и **информацията в тях е синтезирана и изложена в дисертационния труд**. Публикациите, касаещи инструменти и платформи за тестване, са описани според това дали се отнасят за композиция от уеб услуги или единични такива. Методологиите са описани в една подсекция на дисертацията, като след това са сравнени и анализирани. А подходите за тестване на разпределените архитектури са подбрани и включени според тяхната използваемост и принос за разработената методология за тестване на композиция от уеб услуги.

За да се направи **сравнителен анализ** на всяка от трите групи – инструменти за тестване на единични услуги, инструменти за тестване на съставни услуги и методологии за тестване в областта на АОУ е необходимо да се въведат критерии за сравнение.

Критериите, които се използват за съпоставяне на инструментите за тестване на единични услуги, са следните:

- 1) Да подпомага тестването на уеб услуга на база нейното **WSDL описание**;
 - 2) Видове **тестове**, които могат да се реализират;
 - 3) **Отчитане** на други метрики и променливи, освен получения от отговор от заявката, при наблюдение на състоянието на услугата/системата обект на тестване;
 - 4) Да позволява **валидация**, проверка на резултата;
 - 5) Възможност за **параметризация**, на входни данни, очаквани резултати и настройки на средата.
 - 6) Свободен за използване без нужда от закупуване на комерсиален **лиценз**.
- Тъй като на композиция от уеб услуги, може да бъде представена като единична услуга, когато на нея се гледа като на черна кутия, то предложените критерии за тестване на една уеб услуга се отнасят и за композиция от услуги. Към последните, обаче са добавени и:
- 7) Начин на **представяне на композицията от уеб услуги** - чрез BPEL, чрез друг език, чрез друг език или модел, който се трансформира в BPEL.
 - 8) Да позволява **симулация** на услуги, когато те не са налични или тяхното често използване е скъпо.

Сравнението на инструментите за тестване на уеб услуги, които са резултат от направеното проучване е показано в Таблица 4. Ако съответният инструмент предоставя, т.е. има дадена характеристика, е използвано означение ДА, ако липсва информация - Н.И. (няма информация), а ако няма съответните възможности – НЕ.

ТАБЛИЦА 4 СРАВНИТЕЛЕН АНАЛИЗ НА ИНСТРУМЕНТИ ЗА ТЕСТВАНЕ НА УЕБ УСЛУГИ

Инстр.	Тестове	1)	3)	4)	5)	6)	7)	8)
[3]	Функционални тестове Регресионни тестове Тестове за натоварване Тестове за сигурност	ДА	ДА	ДА	ДА	НЕ	НЕ	ДА
[4]	Функционални тестове Тестове за натоварване Тестове за сигурност	ДА	ДА	ДА	НЕ	ДА	НЕ	ДА
[5]	Функционални тестове Тестове за натоварване	ДА	ДА	ДА	ДА	ДА	НЕ	НЕ
[6]	Функционални тестове	НЕ	НЕ	ДА	НЕ	ДА	НЕ	НЕ
[7]	Функционални тестове	НЕ	НЕ	ДА	ДА	ДА	НЕ	НЕ
[8]	Функционални тестове	ДА	НЕ	НЕ	НЕ	ДА	НЕ	НЕ
[9]	Тестове за натоварване Функционални тестове	НЕ	ДА	ДА	ДА	ДА	НЕ	НЕ
[10]	Функционални тестове	НЕ	ДА	ДА	НЕ	ДА	UML към BPEL	НЕ
[11]	Функционални тестове	ДА	ДА	ДА	ДА	ДА	VCL	НЕ
[12]	Тестове за натоварване	ДА	ДА	Н.И.	Н.И.	ДА	WSFL	ДА
[13]	Функционални тестове Регресионни тестове	ДА	ДА	ДА	ДА	НЕ	BPEL	ДА
[14]	Функционални тестове Регресионни тестове Тестове за натоварване	ДА	ДА	ДА	ДА	НЕ	WSFL, BPEL	ДА

За сравнение на тестовите методологии в АОУ са използвани **шест критерия**, които основно са свързани със спецификите на тестването при АОУ. Критериите сравнение на тестовите методологии са следните:

- **Нивата на тестване**, при АОУ те са – на ниво модул (в случаят модулът е услуга), на ниво интеграция или композиция и на ниво система (в случая това е целия бизнес процес). При някои методологии на операциите в услугата се гледа като на най-ниско ниво и това ниво е означено като „Компонент на услуга“ в Таблица 5;
- **Видове тестване**, които са приложими за АОУ са: тестване за съответствие със стандарти, функционално тестване, тестване при натоварване (също се среща и като производителност на системата, а в случаите на АОУ се свързва и с SLA), тестване на сигурност, тестване за оперативна съвместимост;
- В кои **Фази** от жизнения цикъл на АОУ може да се приложи дадена тестова методология.
- Да подпомага тестването и изпълнението на тестовата методология с **подходи**;
- Описана ли е възможност за **автоматизация** на методологията;
- Методологията използва ли модели, които спомагат тестването на АОУ.

В Таблица 5 е изведен сравнителният анализ на намерените, при проучването методологии за тестване на АОУ, а означенията са като тези в Таблица 4.

ТАБЛИЦА 5 СРАВНИТЕЛЕН АНАЛИЗ НА МЕТОДОЛОГИИ ЗА ТЕСТВАНЕ НА АОУ

Методология	Нива на тестване	Видове тестване	Фази на приложение	Подходи	Автоматизация	Използван модел
[15]	Компонент на услуга Услуга Интеграция/ оркестрация Система/ процес	Функционално Производителност Сигурност Оперативна съвместимост SLA	Проектиране Разработка Тестване Експлоатация	НЕ	НЕ	НЕ
[16]	Компонент на услуга Услуга Интеграция/ оркестрация Система/ процес	Функционално Производителност Сигурност Оперативна съвместимост Съответствие със стандарти	Проектиране Разработка Тестване	НЕ	НЕ	НЕ
[17]	Услуга Интеграция/ оркестрация Система/ процес	Функционално	Проектиране Разработка Тестване Експлоатация	ДА	ДА	STS (Symbolic Transition System)
[18]	Интеграция Система	Няма	Проектиране	ДА	ДА	C-LTS (Label Transition System) XRGs (XPath Rewriting Graphs)
[19]	Услуга Интеграция/ оркестрация Система/ процес	Функционално Производителност Сигурност Оперативна съвместимост	Няма	НЕ	НЕ	НЕ
[20]	Услуга Интеграция/ оркестрация	Функционално	Проектиране Разработка Тестване	НЕ	ДА	UML

Методология	Нива на тестване	Видове тестване	Фази на приложение	Подходи	Автоматизация	Използван модел
	Система/ процес		Експлоатация			
[21]	Услуга Интеграция/ оркестрация Система/ процес	Функционално	Проектиране Разработка Тестване Експлоатация	ДА	ДА	UML
[22]	Интеграция/ оркестрация	Функционално	Тестване	ДА	ДА	JPF(JavaPath Finder)
[23]	Услуга Интеграция/ оркестрация	Функционално	Тестване	ДА	НЕ	Methamorphic
[24]	Компонент на услуга Услуга	Функционално	Тестване Експлоатация	ДА	НЕ	GTS(Graph Transformation Systems) PSM(Protocol State Machines)
Предложена методология	Услуга Интеграция/ оркестрация Система/ процес	Функционално Производителност Сигурност SLA Съответствие със стандарти	Разработка Тестване Експлоатация	ДА	ДА	НЕ

1.4 Изводи

Направеното проучване, установи, че до 2015 г. липсва преглед на научните материали, относно методологии за тестване на АОУ. В резултат на това в дисертацията е осъществено задълбочено проучване на съществуващите литературни източници, фокусирано върху методологии и инструменти за тестване на композиция от уеб услуги, като парадигма за реализация на АОУ.

Използвайки стратегия за систематизирано проучване е формулиран израз за търсене в предметната област в 3 електронни библиотеки, избрани са 56 научни публикации и информацията от тях е синтезирана и класифицирана.

Забелязан е пик на научните материали през 2010 г. и същевременно е установено, че до 2015 г. липсва утвърдена, стандартизирана тестова методология и широко приложима, автоматизирана платформа, която да я реализира.

Резултатите от направеното проучване, свързано с инструменти за тестване на единични уеб услуги, инструменти за тестване на съставни уеб услуги и методологии за тестване са представени в 3 подсекции на дисертацията. А публикациите, които представят подходи за тестване на АОУ, са разгледани в следващите глави, предвид необходимостта да бъдат идентифицирани тези, които са подходящи за реализацията на предложената методология. Разгледаните инструменти и методологии за тестване на АОУ са сравнени и анализирани.

От направения анализ може да се заключи, че софтуерните инструменти за тестване на АОУ предлагат основно модулно тестване, от тип „черна кутия”, подходящо за тестване на единични услуги. Инструментите, фокусираните върху композиции от услуги, които се появиха последните години, все още имат недостатъци - извършват преобразуване на бизнес процеса от един език (например UML) в друг (BPEL), при което често се губи част от бизнес логиката; липсва автоматизация или е частична – често се налага намеса от тестера, например при специфициране на данни; имат слаба проследимост и проверка за покритие на пътищата в процеса; липсва възможност за мониторинг на състоянието на системата, и на метрики за отчитане на нефункционални характеристики; отчитат само крайният, получен резултат в следствие на изпълнение на даден бизнес процес.

От направения анализ на софтуерни инструменти, може да се изведе, че автоматизирането на тестовия процес при АОУ е на ниво, при което се предлагат базови възможности за тестване на уеб услуги, но все още има място за подобрене и комбинирането им в единна платформа осигуряваща тестване от край до край (end-to-end testing).

След обстойно, систематизирано разглеждане на съществуващите методологии за тестване на АОУ, се наблюдава **липса на конкретна методология, която да осигури високо ниво на качеството, покривайки всички аспекти на тестовия процес при наличие на автоматизация.**

Изследваните методологии включват разнообразен спектър от подходи за тестване на АОУ. Някои от тях са иновативни в решението си да представят бизнес логиката на композицията от услуги чрез формални тестови модели и да се възползват от вече съществуващи и утвърдени начини за тестване им. Други методологии се основават на традиционните видове тестове, имайки за цел да обхванат колкото се може повече нива на тестване и фази от жизнения цикъл на СБУ.

2. Методология за тестване на съставни услуги

Според една от дефинициите – „**Методология е система от методи или група от методи, правила и постулати използвани в дадена дисциплина**“ [25]. Следвайки тази дефиниция методология за тестване на композиция от уеб услуги трябва да представлява група методи, правила и постулати даващи предложения, насоки, отговори, стъпки, инструменти за нейното тестване. Всъщност най-общо казано методологията обяснява как да се направи нещо.

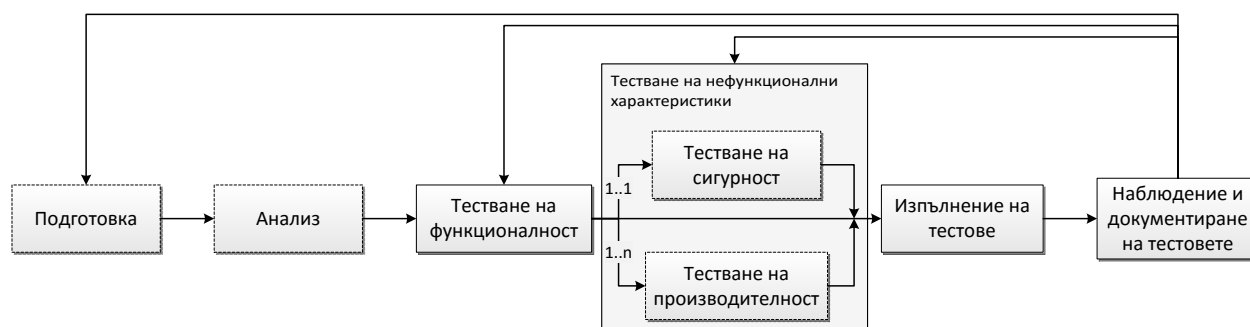
В тази глава на дисертацията е предложена методология (Задача 2), която комбинира подходи, инструменти, процедури, практики, умения и опит, за да се отговори на въпроса

„Как да се тестват композиции от уеб услуги, за да се осигури покриване на изискванията за качество?“.

2.1 Концептуално описание на методология за тестване на съставни услуги

За да се валидира дадена разпределена, хетерогенна система, каквато е СБУ, спрямо изискванията за функционалност, производителност и сигурност, както и поведението ѝ при промяна в реализацията ѝ, настоящата дисертация предлага методология представена на Фиг. 1, която се състои от следните основни стъпки:

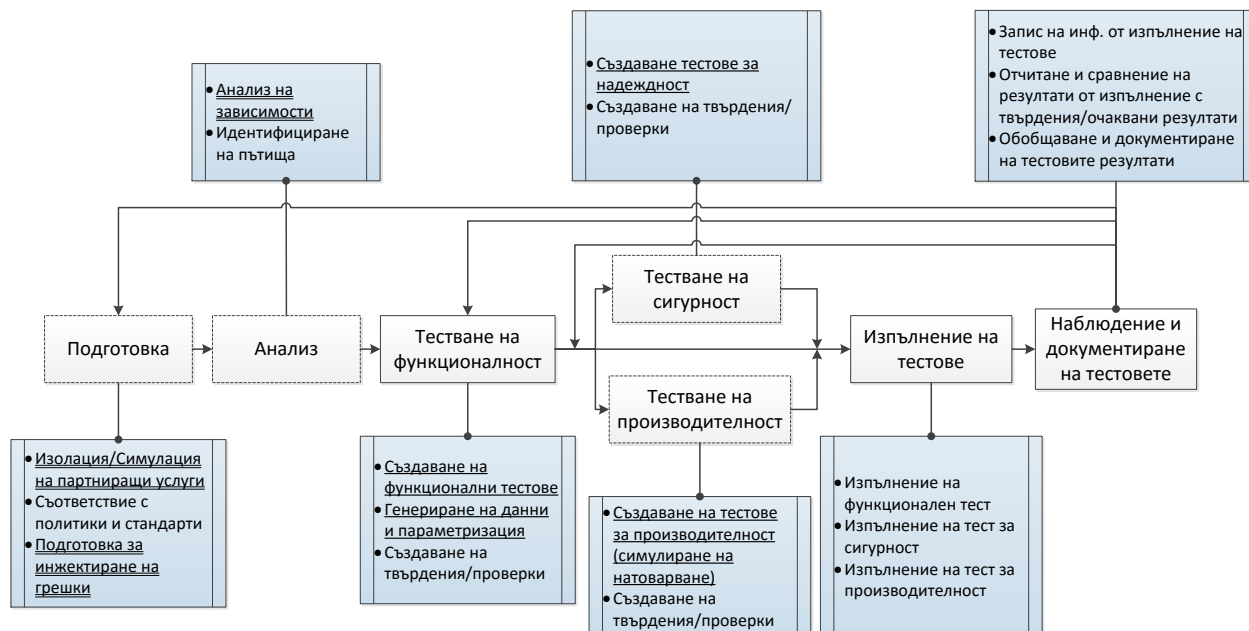
- 1) Подготовка на VPEL процесът, който ще се тества, т.е. композицията от уеб услуги, която е обект на тестване.
- 2) Определяне на тестова цел (покритие на всички пътища, избор на определен тестов път), анализ на бизнес процеса и съответните променливи, за да се идентифицират тестови сценарии.
- 3) **Дефиниране на функционални тестове.**
- 4) Дефиниране на нефункционални тестове, в две групи, според това дали се изисква паралелно или последователно изпълнение:
 - Тестване на сигурността на база вече създаден функционален тест, т.е. използва се заявка с входни данни към съответната услуга от функционален тест, надградена с необходимите артефакти за тестване на сигурността.
 - Тестване на производителност и натоварване, при което се изпращат множество заявки към бизнес процеса, обект на тестване.
- 5) **Изпълнение на тестовите сценарии** (процесът) в сървъра за приложения.
- 6) Резултатите от тестовете се записват и анализират.



ФИГ. 1 КОНЦЕПТУАЛНА СХЕМА НА МЕТОДОЛОГИЯ ЗА ТЕСТВАНЕ НА КОМПОЗИЦИЯТА ОТ УСЛУГИ

Маркираните с пунктир стъпки на Фиг. 1 са опционални и може да не се изпълняват в зависимост от конкретният процес, от поставените изисквания към него и др. Също така трябва да се отбележи, че тестването на сигурността и на производителността са независими едно от друго, и потребителят може да изпълни само един от тях или в избран от него ред.

За да се добие задълбочена представа за разработената методология, на Фиг. 2 е представена подробна схема, в която са показани основните дейности и задачи от нея. За задачите с двойно подчертаване са разработени нови подходи в рамките на дисертационния труд, а именно подход за изолиране на една или няколко уеб услуги, които се използват от процеса, подход за инжектиране на неизправности и подход за анализиране на даннови зависимости. Те са описани подробно в следващата глава. За задачите с единично подчертаване е използван утвърден подход, който е допълнен или изменен за нуждите на тестването на композиция от уеб услуги. За реализация на останалите аспекти от предложената методология са разгледани и избрани всеобщо утвърдени подходи, доказали се като ефективни.



ФИГ. 2 ДЕТАЙЛНА СХЕМА НА МЕТОДОЛОГИЯТА ЗА ТЕСТВАНЕ НА КОМПОЗИЦИЯ ОТ УЕБ УСЛУГИ

Предложената методология за тестване на композиция от уеб услуги, въпреки че включва множеството задачи подлежи на автоматизация и използва малък набор от тестови артефакти, а именно – BPEL файл (от него автоматично може да се създадат или извличат данните за/от WSDL и XSD файловете), входно/изходни данни и твърдения/проверки. Последователността на работа и взаимодействие между различните стъпки от методологията, както и информацията, която си обменят е дадена на Фиг. 3.

Предложената методология за тестване на композиция от уеб услуги в детайли има следния начин на работа:

- 1) Подготовка на BPEL процесът, който ще се тества, т.е. композицията от уеб услуги, която е обект на тестване.

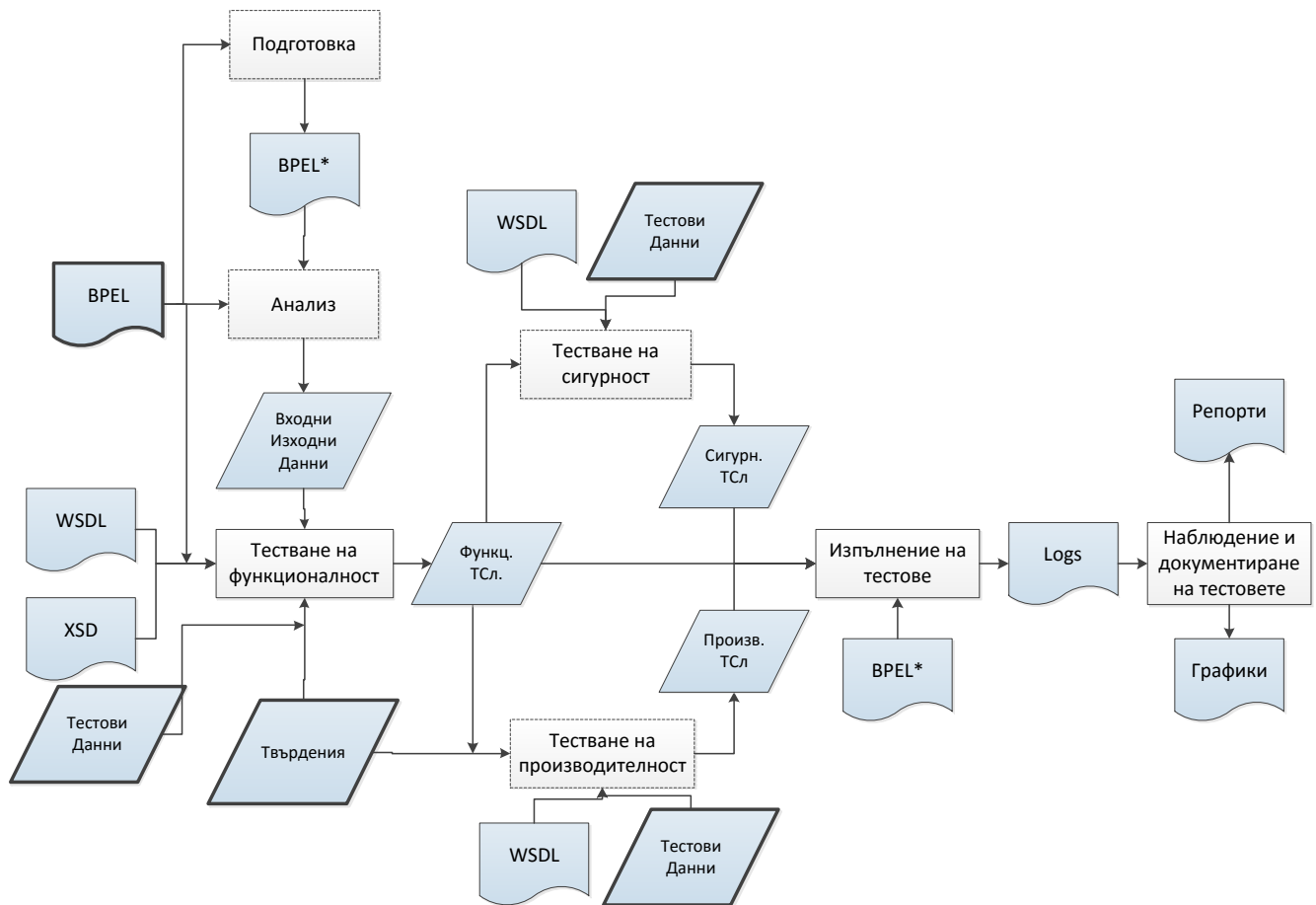
Вх. информация: BPEL (WSDL, XSD, SOAP)

Изх. информация: BPEL* (версия на оригиналния BPEL)

Задачи: Съответствие с политики и стандарти; Изолация на партниращи услуги;

Промяна на връзката между бизнес процес и партньор с цел инжектиране на грешки;

- Бизнес процесът (описан в BPEL файл) и съпътстващите го документи (WSDL, XSD файлове) се валидират/проверяват дали съответстват на стандартите зададени от World Wide Web Consortium (W3C) [26] и ако има грешки при валидацията те се представят на потребителя и следва да се коригират.
- Изолация на услуги се извършва като се генерират данни, които да заместят отговора на услугата или се създава така наречената Stub услуга (“олекотена” версия на реалната услуга, симулираща поведението ѝ). В методологията е избран да се използва първият метод, при който отговорът на партниращата услуга се симулира, като се задават стойности, отговарящи на дефинираните в XSD схемата на процеса/услугата тип данни и се създава трансформирана версия на оригиналния BPEL файл.
- Инжектирането на грешки става, като се внесе промяна в комуникационния канал между тествания бизнес процес и партнираща услуга. Между тях се включва услуга “посредник“, която ще инжектира грешките. Бизнес процесът се променя, така че да се обръща към услугата посредник.



ФИГ. 3 ВЗАИМОДЕЙСТВИЕ МЕЖДУ СЪПКИТЕ ОТ МЕТОДОЛОГИЯТА И ПОТОК НА ДАННИТЕ

2) Анализ на бизнес процеса и идентифициране на пътища и съответните променливи, за създаване на функционални тестове.

Вх. информация: BPEL

Изх. информация: тестови данни

Задачи: Анализ на данните зависимости; Идентифициране на пътища.

- Анализ на данни зависимости. За тази задача е създаден нов подход, при който се идентифицират условните изрази, от които зависи преминаването по конкретна последователност от активности (път) и съответните данните (константи, променливи), от които зависят условията. Този анализ спомага идентифицирането на подходящи тестови данни, такива, че изпълнението на процеса да премине по желания път, т.е. да се изпълни желана тестова цел. Тестовите цели се определят от тестера на база изискванията към системата и тези от тях, които касаят избор на определен път от бизнес процес или покритие на всички пътища в него, се отразяват върху този избор.
- Покритие на пътища с тестови сценарии е един от най-често използваните методи за тестване на бизнес процеси. Обикновено за да се изследват пътищата в процес описан, чрез BPEL се извършва трансформация към друго представяне/език. Подходът, който се използва в методологията е сходен с [27] и [28] и представя процесът, който ще се тества, чрез междинен BFG (BPEL Flow Graph) модел, проследява BFG, за да намери набор от пътища, които удовлетворяват определени тестови цели, премахва неосъществимите пътища и комбинира пътищата и данни за да се създадат тестови сценарии.

3) Функционалното тестване на бизнес процеса, на тази стъпка е тип „черна кутия“, т.е. на бизнес процесът вече се гледа като на уеб услуга.

Вх. информация: BPEL, WSDL, XSD, тестови данни, твърдения

Изх. информация: тестови сценарии (функционални тестове)

Задачи: Създаване на функционални тестове; Генериране на тестови данни и параметризация; Създаване на проверки/твърдения.

- Функционалните тестове се създават като заявки, с входни тестови данни и очаквани отговори, към бизнес процеса на база неговото WSDL описание. Тези тестове по своята същност представляват модулно тестване на уеб услуга (WS-Unit Tests), което е добре познат и често реализиран подход за тестване на уеб услуги и композиция от уеб услуги. За да се надгради този метод и да допълни нуждите за тестване конкретно на композиция от услуги се дава възможност за проверка на обходения път в бизнес процеса, т.е. идентифицират се изпълнените активности и се проверяват вътрешни за процеса данни.
 - Генерирането на данни и параметризацията на тестовете се извършва по добре познат метод, при който на база XSD схемата на входните и изходните данни се генерират коректни тестови данни, като се добавя и възможност за тяхното параметризиране, чрез използване на променливи, функции и четене от файл и база данни. Ако изпълнението на методологията премине през стъпка „2) Анализ“ то за генерирането на тестови данни се следва описанието от тази стъпка.
 - Създаване на проверки/твърдения се осъществява, като се определят очакваните резултати от изпълнението на теста. Твърденията, които са приложими за уеб услуги, извършват проверка на получения резултат спрямо точна стойност, регулярен израз, тип на стойност, тип на файл, големина на файл, HTTP статус код на отговор, удовлетворяване на съответен стандарт и др. Важна проверка, която е препоръчително да се включи, е редът на изпълнение на активностите в бизнес процеса, което всъщност основно извършва тестването на композицията от услуги.
- 4) Тестване на сигурността на уеб услугата се извършва на база вече създаден функционален тест, т.е. да има заявка с входни данни към съответната услуга, която да се надгради със съответните артефакти за тестване на сигурността. Проверява се, че когато съответната заявка и входни данни отговарят на механизмите за сигурност се получава и връща коректен отговор от услугата, а когато не отговаря на изисквания механизъм за сигурност се връща съответният отговор за нарушение на сигурността (например забранен, неоторизиран достъп, не е намерена информация, частично съобщение, лоша заявка, услугата не е на разположение и др.).
- Вх. информация: BPEL, WSDL, XSD, тестови данни, твърдения
Изх. информация: тестови сценарии (тестове за сигурност)
- Задачи: Създаване на тестове на ниво съобщения; Създаване на тестове на ниво транспорт; Създаване на тестове за устойчивост; Създаване на твърдения/проверки.

- Тестване на сигурността на ниво съобщения се състои в изпращането на заявки със „сигурни“ съобщения и отчитане на получените отговори. Поради голямото разнообразие от механизми за сигурност този подход включва валидиране по отношение на WS-Security включително X509, SAML, Username security tokens, XML Encryption и XML Digital Signature, както и QoS политики, за които се изисква изпращане на специфична информация в заявката или отговора - WS-RM, WS-Addressing, MOTM, т.е. механизми за прикачване на файлове, редактиране на заглавната част на съобщението (Head), криптиране и декриптиране.
- Тестове на сигурността на ниво транспорт включват тестове за автентификация, като се осъществяват връзки със съответните сървъри и се предават сертификати като SSL, Kerberos.
- За да се реализира задачата за тестване на устойчивостта на системата е предложен нов подходът, който изисква трансформация на BPEL файла, която е направена в стъпка 1) на методологията, а на този етап се въвеждат тестови данни, определят се очаквани резултати и се избира начин за компрометиране на

- надеждността на бизнес процеса (забавяне на отговор, недостъпна услуга, грешка в семантиката или граматиката на съобщението).
- Начинът за създаване на проверки, при тестване на сигурността не е много различен от този за създаване на проверки за функционални тестове. Тук основния вид проверки са свързани с получени файлове, текст в съобщението на отговора и код за грешка.
- 5) Тестване на производителността или издръжливостта на процеса (услугата), се извършва като се изпратят множество заявки към него и се отчетат стойностите за нефункционалните метрики.
- Вх. информация: BPEL, WSDL, XSD, тестови данни, твърдения
 Изх. информация: тестови сценарии (тестове за производителност)
 Задачи: Задаване на виртуални потребители; Създаване на профили; Създаване на графици; Създаване на твърдения/проверки.
- Задаване на виртуални потребители включва симулиране на определен брой изпълнения (виртуални потребители) за всеки функционален тест или за група функционални тестове. Под изпълнение се има предвид изпращане на брой заявки за определено време с тестовите данни, описани във функционалния тест. В повечето случаи се създават т. нар. нишки (threads).
 - Графиките представляват описание на начина на управление на виртуалните потребители (нишките). Описва се кога да се стартира изпращането на определена заявка към услугата/бизнес процеса и кога изпращането да приключи. Във връзка с това се използват допълнителни указания като време на стартиране, ред на изпълнение, брой изпълнения, времеви интервал между тестовете и др.
 - Създаване на твърдения/проверки при тестовете за натоварване надгражда твърденията при функционалните тестове. Въвеждат се проверки за времеви интервали, използваемост на ресурси, големина на файлове и други характеристики, които дават информация за натовареността на системата, мрежата, софтуерните и хардуерните ресурси. Тук се включва и създаване на проверка, специфична за тестването на услуги, а именно дали са спазени SLA договорите.
- б) Изпълнение на тестовите сценарии (процесът, заедно със съответните заявки, тестови данни и твърдения) в сървъра за приложения.
- Вх. информация: BPEL, Тестове
 Изх. информация: Логове
 Задачи: Изпълнение на функционален тест; Изпълнение на тест за сигурност; Изпълнение на тест за производителност; Разпределение на тестове на различни машини
- Изпълнение на един функционален тест се извършва като се публикува (deploy) и стартира инстанция на бизнес процес, описан чрез BPEL, на сървър за изпълнение и към него се създаде т.нар. клиент, т.е. бизнес процесът се „консумира“ като услуга, към която се изпращат заявки –съответстващият тестови сценарии, изпълняват се и се връщат отговори. Механизмът за изпълнение е добре познат и изисква BPEL Engine.
 - Изпълнението на тест за сигурност е до голяма степен идентично с метода използван при изпълнение на функционален тест с тази разлика, че тук се изискват настройки на сигурността (на мрежа, сървър и др.), поради което за да се изпълнят коректно тези тестове трябва сървърът/мрежата, на които се изпълнява веб услугата обект на тестване да са конфигурирани правилно.
 - Изпълнение на тест за бързодействие и производителност представлява в своята същност създаване на нишки (threads) според описаните тестове за производителност към инстанция на бизнес процес. В този случай се надгражда методът за изпълнение на функционален тест като се стартират, изпълняват, спират нишки, по начина, описан от тестовете.

- Разпределение на тестове на различни машини е включено в методологията за тестване, за да се покрийт случаите, когато се очаква голямо натоварване на композицията от уеб услуги или хардуерните ресурси на тестовия сървър са ограничени. Това включва управление и разпределяне на „тестващи клиенти/агенти“ на различни компютри. Стартирането на тестовете се синхронизира и резултатите се записват, отчитат от един главен сървър.
- 7) Записване и анализиране на резултатите от тестовете.
- Вх. информация: Логове и тестове
Изх. информация: Отчети, Графики
- Задачи: Запис на информацията от изпълнение на тестове; Отчитане на резултати от изпълнение на тестовете въз основа на дефинираните твърдения; Обобщаване и документиране на тестовите резултати.
- Информацията от изпълнение на всички тестове, изпълнени на даден сървър се записва в лог файлове, като същевременно по-важната (тази, която е изисквана) се записва в база данни за по-лесна обработка. Заедно с информацията от изпълнение на тестовете се записват и метрики, свързани с бизнес процеса, параметрите на сървъра, комуникационните канали, мрежата и др. Метриците, които се отчитат са прости и е възможно тяхното комбиниране за изчисляване на т.н. комплексни/сложни метрики.
 - Получаването на информация за статус на твърдения/проверки, се извършва като се записват очакваният резултат/ стойност и получения резултат/ стойност с цел сравнение между тях. След сравнението се отчита дали статусът на теста е успешен (pass) или неуспешен (fail). При неуспешен тест се отбелязва, каква точно е причината за получения неуспех.
 - Отчитане на резултати от изпълнение на тестовете въз основа на дефинираните твърдения – заявките, изпратени към конкретен процес, и отговорите, в това число и съобщения за грешки и повреди, се записват и се предоставя възможност на тяхна база да се създадат нови тестове, които да се използват за регресионно тестване.
 - Графици и документи с информация се създават на база записаната в лог файлове и база данни информация. В АОУ се използва основно XML формат на данните, за който има създаден език за стилизация - XSL (EXtensible Stylesheet Language), с помощта на който информацията от проведените тестове може да бъде представена.

2.2 Подходи за тестване на съставни услуги, подкрепящи методологията

Разгледани са подходи, които са приложими за реализация на предложената в дисертацията тестова методология. След подробен анализ на съществуващите подходи в областта са избрани най-подходящите от тях или са предложени нови подходи.

- **Подход за проверка на съответствие със стандарти**

Подходът проверява дали XML документи, които са стандартизирани, отговарят на правилата, дефинирани в съответните стандарти. Така например, BPEL е XML базиран език, който използва стандарти за уеб услуги, за да опише композиция от уеб услуги, изпълняваща дадена бизнес задача.

В предложената методология е избран подходът, реализиран от Eclipse BPEL Validator [29]. При него BPEL описанието на бизнес процеса се разглежда като DOM модел и за всеки възел, характерен за BPEL езика (например assign, copy, from, to и т.н.) се създават правила, с които да се провери дали отговарят на ограниченията определени от съответния стандарт. Правилата са написани с код на JAVA и има възможност да се добавят и променят.

- **Подход за изолиране от външни зависимости**

Същността на този подход се състои в прекъсването на зависимостта на даден бизнес процес от реалното извикване и изпълнение на услуга, която му партнира. Възможността за тестване на BPEL процес без наличието на услугите партньори ще

даде възможност за тестване преди партниращите услуги да са разработени. Този подход позволява модулно тестване на композиция от услуги, при което услугата е в изолация.

Изолацията на външни зависимости е известна още като виртуализация на уеб услуги. Сред най-популярните подходи за виртуализация е създаването на „олекотена“ версия на симулираната услуга (stub, mockup) [30][31][32][33], чрез запис и възпроизвеждане на съобщенията обменяни с нея [34] [30] [31]. Принос на дисертацията е създаденият нов подход за изолиране на партниращи уеб услуги, който е използван в методологията.

- **Подход за инжектиране на неизправности**

За да се провери надеждността на бизнес процеса, комуникиращ с външни услуги е необходимо да се извърши негативно тестване чрез предизвикване на неизправности както в самия процес, така и при комуникацията с партниращите услуги.

Що се отнася до тестване на надеждността, съществуващите подходи предлагат инжектиране на повреди в реализацията на услугата и по този начин извършване на тестване чрез покритие на кода, отговарящ за възстановяване от грешки [35] или анализиране на композицията от услуги от гледна точка на отказоустойчивост [36]. Сред откритите научни трудове само два разглеждат тестването на отказоустойчивостта на композиции, описани чрез BPEL [37], [38]. В рамките на дисертационния труд е предложен нов подход, който е реализиран чрез няколко дейности в описаната методология (Подготовка за инжектиране на грешки, Създаване тестове за надеждност, Създаване на проверки/твърдения и Изпълнение на тест за сигурност/надеждност).

- **Подход за създаване на функционални тестове**

Подходите за генериране на функционални тестове за композиция от уеб услуги може да се разделят в две основни групи – тестване тип „черна кутия“ и тестване тип „бяла/прозрачна кутия“. Техниките за тестване тип „черна кутия“ разглеждат BPEL процеса като единична услуга с известни операции и тип съобщения (дадени в WSDL описанието на процеса). При подходите за тестване като „бяла кутия“ се разглежда композицията, описана чрез BPEL езика (дейности, поток от данни, управляващ поток).

При тестването на приложения реализиращи АОУ най-често разглежданите подходи в научните трудове са тези, свързани с функционално тестване и по-конкретно генериране на тестови сценарии. Сред тях са, подходите описани в [39], [27], [40], [41], [42]. В предложената тестова методология е избран комбиниран подход, който използва техники от метода на „черната кутия“ и на „стъклената кутия“.

- **Анализ на зависимости**

В компютърните програми съществуват два основни вида зависимости – зависимости породени от управляващ поток (при управлението на потокът от данни) и зависимости породени от даннов поток (между самите данни). В системите, реализиращи АОУ, са разработените методи за анализ на тези зависимости са в четири направления – методи базирани на граф, методи с формално моделиране, методи с количествено моделиране и такива базирани на изкуствен интелект.

Сред по-популярните подходи, които могат да се приложат за композиция от услуги са: трансформиране в граф на управляващ поток [27], [39], [43], [44]; представяне с мрежи на Петри [45]; трансформиране в граф на състоянията и евристично търсене [42]; трансформиране и анализиране към други по-популярни езици за описание на бизнес процеси като UML [41] [46]; базирани на матрици на зависимостите [47]. В дисертационния труд е синтезиран нов подход за анализ на зависимостите, използващ представяне на бизнес процеса чрез граф на управляващ поток (Control Flow Graph, CFG).

- **Подход за генериране на тестови данни**

Подходите за генериране на тестови данни в изследваната област, са базирани на тестване тип „черна кутия“ и използват само обхвата и типа на входните и изходните данни. В АОУ, комуникацията между различните услуги се осъществява посредством XML съобщения. Поради тази причина една голяма част от подходите за генериране на тестови данни използват WSDL спецификациите на услугите в комбинация с информацията за типовете данни, описани в XML схема [48][49][50][51][52]. XSD схемите се използва за определяне на ограниченията върху типовете, в резултат на което могат да се генерират тестови данни.

Подходът за генериране на входни данни според дефинирана схема XSD, вече е познат и се използва в инструменти за тестване на уеб услуги [53][54][55][56]. За нуждите на методологията е необходим такъв подход, при който се приема като вход XML схема и спрямо нея се създават стойности, удовлетворяващи типа данни, зададен от схемата или стойности, които не отговарят на типа данни, зададен в схемата. Предложен е подход, който е реализиран в софтуерен инструмент, надграждащ функционалността на вече съществуващите инструменти, като реализира техните основни характеристики и добавя нови, за да спомогне генерирането на тестови данни [57].

- **Подход за създаване на проверки/твърдения**

Подходите за създаване на твърдения обикновено са породени и се фокусират върху една от следните цели – да проверят дали даден обект отговаря на съответните стандарти, да проверят дали отговаря на функционалните изисквания поставени към обекта и дали отговаря на нефункционалните (за сигурност или за бързодействие) изисквания отправени към него.

Авторите на [58] представят предимствата на модела за тестови твърдения предложен от OASIS в [59].

Schematron [60] е схема език за проверка на шаблони открити в XML. Той използва XPath функции и изрази, и може да бъде реализиран чрез XSLT. Твърденията в Schematron позволяват да се провери, че даден документ отговаря на съответната схема, а също така валидират и XPath изрази.

XMLUnit [61] е програмна реализация на JAVA и .NET и предлага класове за проверка на валидността на XML схема, за проверка на стойностите на XPath заявки (изрази), а също така дава възможност за сравняване на цял XML документ с очакван резултат/текст.

Подходящ подход за реализация на покритие и валидиране на изискванията в методологията, е предложението от Durand и Green [58]. Освен него, практическо приложение в тестовата методология намира и подходът описаният в [61], поради добрата си програмна реализация (автоматизация).

- **Подход за тестване на бързодействието/ производителността на композиция от уеб услуги**

За да се тестват нефункционалните характеристики, като производителност и натоварване конкретно на композиция от уеб услуги, а не на единична услуга сама по себе си, трябва да се обърне внимание на две неща – изпълнение на бизнес процесите и измерване на нефункционални характеристики за композицията.

Съществуват два основни модела за конкурентност, използвани за изпълнението на софтуерни приложения – базирани на нишки (threads) и базирани на събития (events). Програми, изпълняващи BPEL процесите и заявките свързани с тях (BPEL Engines), по модела базиран на нишки са ActiveBPEL Engine [62], Oracle BPEL Process Manager[63], IBM WebSphere Process Server [64] и др. Модела базиран на събития е реализиран в [65], в BPWS4J [66] и BPEL-Mora [67]. В Apache ODE [68] пък е използвана комбинация от нишковия модел и модел с актьори.

Освен изпълнението на композицията от услуги, при тестването трябва да може и да се отчете нейното бързодействие. Организацията OASIS има утвърден стандарт за „Фактори на качеството на уеб услугите“ [69], на база на които може по-лесно да

се направи избор, да се систематизират и групират различните подходи за тестване на производителността на услугите в АОУ.

Подходите за тестване на производителност на ниво уеб услуга са обект на разглеждане от значително по-дълго време, за разлика от тези, които тестват на ниво композицията от уеб услуги, описана чрез WPEL. Като допълнение при тестването на композиции от уеб услуги спрямо тези за тестване на единична услуга се взема под внимание начинът, по който партниращите уеб услуги са оркестрирани в самия бизнес процес, т.е. тяхната композиция. При този подход се пресмятат факторите на качество [69] на всяка от партниращите уеб услугите, за да може да се агрегира съответния фактор за цялата композиция от услуги и да се идентифицира, от кои партниращи услуги зависи. Този подход е дефиниран в няколко литературни източника [70],[71],[72].

При валидацията на предложената методология е използвана комбинация от подходи, при която WPEL процесът се изпълнява на ApacheODE [68]. Генерирането на виртуални потребители е от външна програма, която извършва отчитане на нефункционалните характеристики на принципа на „черната кутия“. Тестовите данни захранващи виртуалните потребители следват подхода за генериране на данни описан в дисертацията. Отчитането на съответните метрики се базира на получените изходни данни от генератора на виртуални потребители и се комбинира с подходите за агрегиране на показателите за измерване на качеството на ниво услуга.

- **Подход за тестване на сигурността**

Според OASIS [69] „Типични заплахи за АОУ, са неоторизиран достъп, излагане, фалшифициране и унищожаване на уеб услуги“.

В част от проучените източници [73], [74], [75] се разглеждат подходи, при които се създават модели за достъп на информацията в и от единични услуги, и на база на тези модели се оценява сигурността при тяхната композиция. Други подходи за тестване на сигурността на уеб услуги са описани в [76], [77], [78], [79], [80].

В предложената методология, тестовете за сигурност се състоят в това да се провери дали съответният механизъм или стандарт (електронен подпис, криптиране, политика) е приложен или не. Тестовете в този случай може да са на две нива – ниво транспорт и ниво съобщение. Но тъй като предложената методология е насочена към композиция от уеб услуги, е необходимо да се обърне внимание и на тестове, които са на ниво бизнес процес. За да се провери дали няма нарушения на изискванията за сигурност в потока на работа и на данните на WPEL процеса, се използва подходът предложен в [80].

- **Подход за верификация и отчитане на резултатите от тестовете**

В компютърните системи един от най-често използваните начини за проследяване на поведението на една система е чрез т.нар. наблюдение или още верификация по време на работа (Runtime monitoring). Подходите за наблюдение в АОУ могат да се класифицират по различни начини, като едни от най-приложимите отчитат: характеристиките свързани с начина на събиране на данни, степента на вмешателство, времето за отчитане, техники за валидация.

Авторите на [81], [82] и [83] използват популярен подход, при който WPEL процесът се „инструментира“. Инструментирането представлява вмъкване на допълнителен код в самия WPEL файл, чиято цел е да спомогне мониторинга, но не и да промени основния работен и даннов поток на бизнес процесът.

Една голяма група научни изследвания за наблюдение в режим на изпълнение на софтуерни системи се фокусират върху връзката между начина на задаване на изискванията към системата и начина на верификация, че нейното поведение удовлетворява тези изисквания [51], [84], [85].

Като най-подходящ подход за отчитане на резултатите при тестване на композиция от услуги в предложената методология е избран този, при който се използват инструменти/ техники за наблюдение от използвания WPEL Engine. При нужда от

допълнителна информация при тестването на нефункционалните характеристики е препоръчително в допълнение да се използва подход за наблюдение на обменните съобщения между доставчиците и клиентите, чрез „поставяне на проби“, където е възможно.

2.3 Приложение на предложената методология

Предложената методология е с относително високо ниво на абстракция и гъвкавост (голяма част от стъпките и са незадължителни), което позволява нейното използване и при тестване на бизнес процеси, описани чрез различни стандарти (BPEL4WS, UML, BPMN); единични веб услуги; разновидности на АОУ (Microservices, Event-Driven Architectures, Cloud Architectures); други разпределени и слабо свързани системи.

2.4 Изводи

Предложената методология се състои от 6 основни стъпки като 2 от тях са задължителни – дефиниране на функционални тестове и тяхното изпълнение. Тя е **систематизирана като последователност от действия и гъвкава**, поради това е лесна за разбиране и използване. Незадължителното извършване на някои дейности, например дефиниране на нефункционални тестове, **позволява фокусиране на тестовия процес върху конкретните нужди на потребителя и извършване на поетапно тестване**.

Предложената методология, може да се използва от **различни специалисти, за различни тестови цели**. Специалистите по сигурността, могат да я използват за тестване на сигурността, докато друга група специалисти може да се фокусират само върху тестване на натоварването.

Друго предимство на предложената методология, е че тя използва **малък набор от тестови артефакти**. За да се извърши тестване на композиция от веб услуги, дефинирана чрез BPEL, следвайки въведената методология са необходими BPEL файл, тестови данни и тестови твърдения. При реализация на тестове за пълно покритие, базирани на методологията, е **възможно да се постигне високо ниво на автоматизация**, при което е нужно да се осигури единствено тестваният бизнес процес (BPEL файлът).

Осъществяването на всяка от шестте дейности в методологията е свързано с решаването на редица по-малки задачи. Поради спецификата на АОУ и големия брой предизвикателства, при нейното тестване, конвенционалните подходи и методи за тестване в други сфери имат ограничено приложение. Ето защо са **проучени, анализирани и подбрани подходи за тестване на композиция от веб услуги**, подкрепящи предложената методология и описаните в нея задачи.

3. Разработени подходи

В дисертацията са разработени нови или разширени подходи, заедно с реализиращите ги софтуерни инструменти, чието предназначение е да подкрепят методологията за тестване на АОУ (Задача 3). Предложени са нови подходи за изолиране от външни зависимости, за инжектиране на неизправности и за анализ на данните зависимости. Съществуващи подходи за генериране на тестови данни и за създаване на функционални тестове са разширени по подходящ начин, за да бъдат приложени в предложената методология.

3.1 Подход за изолиране от външни зависимости

В бизнес процес, описан чрез BPEL, зависимостта от външни услуги може да се прояви чрез активностите Invoke, Invoke+Receive, Reply, Pick/OnMessage, Pick/OnAlarm и HumanTask.

Invoke активността може да се представи чрез следния израз:

$$o = f(i_1, i_2, \dots, i_n, R) \quad (1)$$

С f е отбелязана функционалността, която се реализира от операцията, предоставена от външната услуга, чрез i_1 до i_n са входните данни, които се подават на операцията, чрез o е резултатът от изпълнението, и R са допълнителни параметри на активността, които не са

пряко свързани с изпълнението на операцията (например име на активността, координати на съответния блок в BPEL диаграмата и т.н.).

За да премахването на зависимостта от f са възможни два подхода:

1. да се замени f с друга операция \bar{f} , която е под собствен контрол на тестера и която има същият синтаксис (т.е. получава същите входни данни и връща същият тип изходни данни):

$$o = \bar{f}(i_1, i_2, \dots, i_n, R) \quad (2)$$

2. да се замени f с константа, чийто тип съответства на типа изходни данни на f :

$$o = C \quad (3)$$

Подход 1) е известен и прилаган във всички проучени среди за тестване. В тези среди той е известен като “Емулация” и се състои в замяната на извикването на същинската операция с извикване на т.нар. stub операция, която има същия синтаксис (т.е. не се налага промяна на модела на данните).

Намаляването на необходимите тестови артефакти е от особено значение за ефективното премахване на зависимостите. Поради тази причина в предложената методология е възприет подход 2), а именно замяната на извикването на операцията f с константа от съответния тип данни, отговарящ на типа на резултата на f . При този подход (наричан по-долу “Изоляция”), се налага модификация на процеса, като съответната Invoke активност се замени с Assign активност, която присвоява на изходната променлива o конкретни стойности, определени от потребителя. Към останалите зависимости се подхожда по подобен начин – напр. при асинхронното извикване на операция (Invoke + Receive), Invoke активността се заменя с Empty активност (тъй като не оказва влияние върху процеса), а Receive активността се заменя с Assign. Таблица 6 показва начина, по който процесът се изолира от външни зависимости.

ТАБЛИЦА 6. ИЗОЛАЦИЯ НА ПРОЦЕСА ОТ РАЗЛИЧНИ ВИДОВЕ АКТИВНОСТИ, ВОДЕЩИ ДО ЗАВИСИМОСТИ

Зависимост (същинска активност в BPEL)	Замяна с
Invoke	Assign
Invoke+Receive	Empty+Assign
Pick/OnMessage	Assign
Pick/OnAlarm	Empty
Reply	Empty
HumanTask	Assign

Проучванията показват, че предложеният подход не се среща в известните програмни среди за разработка на софтуерни приложения и следователно създаването и приложението му представлява принос в тематичната област.

Софтуерната реализация на Изоляционния инструмент, реализиращ подхода, е чрез уеб услуга, наречена SimulateWS. За нея има разработен интерфейс за команден ред, графичен интерфейс за средата за разработка NetBeans и графичен интерфейс за средата за разработка Eclipse. Конзолното приложение, предава конфигурационни параметри в необходимия формат на съответния метод на уеб услугата SimulateWS. Тя прочита подадения входен BPEL файл, извършва замяна на активности със зададените стойности и създава нова версия на BPEL файла. Тази нова версия се записва на файловата система, според указания път или ако липсва такъв, се изписва в конзолата.

3.2 Подход за инжектиране на неизправности

Предложен е нов подход, който извършва нефункционално тестване на АОУ и по точно разрушително тестване и тестване на устойчивост, чрез инжектиране на грешки, симулиране на забавяне и липса на услуги. За да се симулират тези повреди първо се променя активността (Invoke) в BPEL процеса, която изпраща и приема съобщенията между тествания процес и партниращата услуга. Съобщенията се изпращат и приемат не

към/ от реалната партнираща услуга, а към/ от услуга посредник (proxy). Посредникът трябва да е универсален – т.е. да може да замести произволно извикване на операция от произволна услуга. Поради тази причина се налага допълнително кодиране/декодиране на входно/изходните данни на активностите, които се модифицират. За целта могат да се използват вградените в WPEL функции за маршализация и демаршализация. Преди извикване на посредника XML структурите, които съставляват аргументите на истинския получател, се преобразуват в символен низ (маршализация). Този символен низ се подава като аргумент на посредника, който преди да изпрати съобщението на истинския получател извършва обратната операция – демаршализация – и преобразува символния низ в оригиналната XML структура, която се подава на истинския получател. При получаване на отговор се действа по аналогичен начин – получената структура се нарежда/ събира в символен низ, който се връща на WPEL процеса, където, преди да се продължи с оригиналната бизнес логика, низът се пренарежда/ разлага и се попълват изходните променливи на активността. Формално погледнато, се извършва следната трансформация:

$$o = Invoke(i_1, i_2, \dots, i_n) \Rightarrow o = UnMarshal(ProxyInvoke(Marshal(i_1, i_2, \dots, i_n), R)); \quad (4)$$

където с i_1, i_2, \dots, i_n са отбелязани оригиналните аргументи на модифицираната активност Invoke; с o – оригиналните изходни данни; с *Marshal* и *UnMarshal* са отбелязани вградените в WPEL функции за маршализация и демаршализация, а с *ProxyInvoke* – извикването на посредника с параметризацията относно исканите повреди R .

Трябва да се отбележи, че така избраният подход е приложим само към Invoke активности, тъй като те са единствените активности, при които обмяната на съобщения се инициира от бизнес процеса, който се тества. Това е необходимо условие за реализация на подхода, тъй като се налага поставянето на активности за маршализация и демаршализация около инициатора на обмяната, а когато инициаторът не е под собствен контрол, това е невъзможно.

На база описания подход е разработен инструмент наречен Инжекционен инструмент. Той представлява уеб услуга, за която са създадени: интерфейс за команден ред, графичен интерфейс за средата за разработка NetBeans и графичен интерфейс за средата за разработка Eclipse. Реализираният по този начин подход има добра оперативна съвместимост и приложение в АОУ.

3.3 Подход за създаване на функционални тестове

В тази секция е предложен подход за създаване на функционалните тестове, който използва съществуващи техники и подходи, комбинирани по такъв начин, че да реализират функционални тестове, приложими за композиция от уеб услуги, описана чрез WPEL. Функционалните тестове се базират на четири елемента – WPEL описанието на бизнес процес(а), входни данни (б), изходни данни (в) и проверки(г).

Използва се един от най-популярните подходи за създаване на функционални тестове в различни среди и програмни езици. Бизнес процесът, описан чрез WPEL, се изпълнява на сървър за приложения, към който се изпращат заявки (съответните входни данни) и се отчитат техните отговори (получените резултати). Получените отговори се валидират спрямо очакваните изходни данни и проверки.

Специфичното, при създаването на функционални тестове за композиция от уеб услуги, се състои в механизма за подбор на четирите елемента и автоматизация, на тези тестове. В настоящия дисертационен труд са създадени следните подходи, които да отговорят на нуждите и проблемите при създаване на функционални тестове – „Анализ на даннови зависимости“, „Изолиране от външни зависимости“, „Генериране на тестови данни“ и „Създаване на проверки/твърдения“.

3.4 Подход за анализ на даннови зависимости

Предложен е подход, който за даден път в WPEL процес, намира всички условни активности, участващи в пътя, специфицира, кои променливи указват влияние на тези условия и какви стойности трябва да имат.

За формалното описание на подхода, са въведени следните дефиниции:

Дефиниция 1: D – множеството от всички процесни променливи, независимо от техния тип. Всяка процесна променлива може да участва в описанието на условията и всяко условие работи само с процесни променливи и константи.

Дефиниция 2: F – множеството от всички булеви функции върху една или няколко процесни променливи. Функциите в F се изграждат от елементарните операции, достъпни в BPEL. В BPEL спецификацията е дефинирана стандартна библиотека от елементарни операции в различни области – математика, работа със символни низове, преобразование на дати и др., като тези елементарни операции се описват посредством XPath.

Дефиниция 3: G – Граф на управляващ поток (Control Flow Graph), който отговаря на тествания BPEL процес. Представява двойката $G = (V, E)$, където V е множеството от върхове, а E – множество от насочени условни бинарни релации между върховете $E \subseteq \{(u, f \times v) / u, v \in V, f \in F\}$ (дъги). В графа на управляващия поток, върховете съответстват на активности в BPEL процеса, а дъгите – на реално възможен преход от активността u в активността v . Това, дали даден преход ще се реализира в действителност, зависи от условието на прехода f .

Дефиниция 4: Път (в графа на управляващия поток) $p = [v_1, v_2, \dots, v_n]$ – последователност от върхове в G , които отговарят на последователност от активности, които се изпълнява по време на работата на BPEL процеса.

Дефиниция 6: $cdr(p)$ – десен под-път на пътя p . Представява пътят p , без първата активност: ако $p = [v_1, v_2, \dots, v_n]$, то $cdr(p) = [v_2, \dots, v_n]$.

Дефиниция 5: $P \subseteq 2^V$ – множеството на валидните пътища – такива, които реално могат да се изпълнят.

Лема 1: Ако $u \in V$ и $e = (u, f \rightarrow v) \in E$, то $[u, v] \in P$. (5)

Доказателство: Съгласно Дефиниция 3, преходът между u и v е възможен поради наличието на дъгата e . „Възможен преход“ означава „може реално да се измине по време на изпълнението“.

Лема 2: Ако $w \in V$, $e = (w, f \rightarrow v_1) \in E$ и $p = [v_1, v_2, \dots, v_n] \in P$, то $[w, v_1, v_2, \dots, v_n] \in P$ (6)

Доказателство: Съгласно Дефиниция 3, преходът между w и v_1 е възможен поради наличието на дъгата e . Преходите между v_1 и v_2 , както и от v_2 до ... и от там до v_n са възможни, тъй като са част от валидния път p . Следователно, всички преходи между последователните активности в пътя $[w, v_1, v_2, \dots, v_n]$ са възможни, т.е. той може да бъде реално изминат по време на изпълнението и съответно е валиден.

Следствие от Лема 2: За всеки път $p = [v_1, v_2, \dots, v_n]$ е вярно твърдението:

$$cdr(p) \in P \wedge \exists e = (w, f \rightarrow v_1) \in E \Rightarrow p \in P \quad (7)$$

или с други думи – ако десния под-път на p е валиден и в графа на изпълнението има дъга между първите два върха в p , то p е валиден път.

Задачата, която подходът решава може да се раздели на 2 подзадачи:

1. Намиране на всички условни активности, включени в даден път в процеса и за всяко условие, намиране на променливите, които участват в него;
2. При зададени стойности на променливите, открити в т. 1, се проверява дали изпълнението на процеса ще премина по исканото разклонение, дефинирано с условната активност.

Подзадача 1 се решава по **Алгоритъм 1** представен по долу.

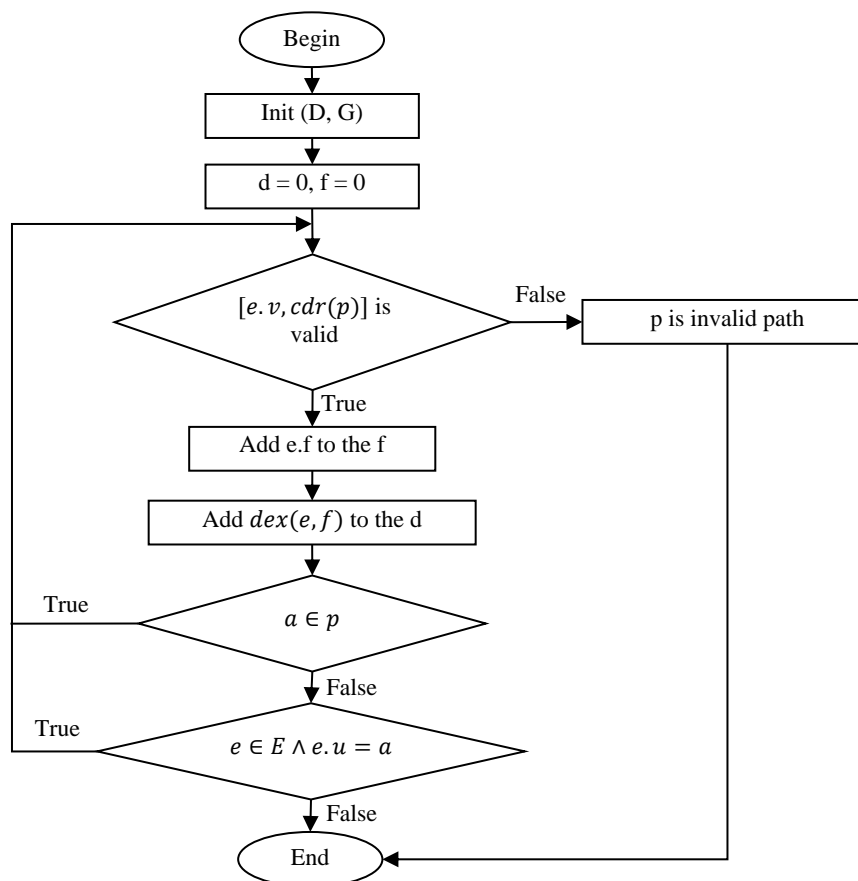
Вход: пътят $p \in P$, който се изследва.

Изход: множество от променливи d и условните изрази f , в които те участват.

Псевдо-кодът на Алгоритъм 1, разработен за целите на подхода, е както следва:

1. $init(D, G)$
2. $d = \emptyset, f = \emptyset$
3. $\forall a \in p$
 - 3.1. $\forall e \in E \wedge e.u = a$
 - 3.1.1. ако $match(e.v, cdr(p))$, то
 - 3.1.1.1. $f = f \cup e.f$
 - 3.1.1.2. $d = d \cup dex(e.f)$
 - 3.1.1.3. изход от цикъла 3.1 чрез преход към следващата итерация на стъпка 3
 - 3.2. грешка - невалиден път. Край.
4. Край.

На Фиг. 4 е показана блок схема на Алгоритъм 1.



ФИГ. 4 БЛОК СХЕМА НА АЛГОРИТЪМ ЗА АНАЛИЗ НА ЗАВИСИМОСТИТЕ

При подзадача 2 при зададени стойности на променливите, открити в подзадача 1, се проверява дали процесът ще премине по исканото разклонение, през дадена условна активност – представлява функционалност, която е налична в средите за изпълнение на бизнес процеси описани чрез BPEL, тъй като е необходима за самото изпълнение на процесите. Следователно, подзадача 2 може да се реши посредством извикването на подходящи API функции на конкретната среда.

Предложеният и описан подход за анализ на условни зависимости е разработен в Инструмент за анализ на зависимости. Той представлява уеб услуга, за която са създадени: интерфейс за команден ред и графично представяне в средата за разработка NetBeans.

3.5 Подход за генериране на тестови данни

По своята същност, подходът за генериране на тестови данни не е нов, но е необходимо неговото детайлизиране и приспособяване за нуждите на BPEL процеси.

Данните в АОУ са в XML вид и отговарят на XSD дефиниция. Това налага използването на подход, които се състои поне от 3 стъпки – 1) извличане на XML схемата, 2) създаване на структура на XML документ според схемата и 3) генериране на стойности за този документ.

Информация от XML схемата е необходима за идентифициране на структурата на тестовите данни. Самата информация може да се съдържа в отделен XSD файл съдържащ самата XML схема или в WSDL или BPEL файл с реферирана (import) или включена (inline) XML схема. Ето защо първата стъпка в подхода е да се добие схемата за XML от файла, която обикновено е заградена със <schema> таг, а в случаите на реферирана схема в WSDL или BPEL файл – с <import> таг, който има атрибут “location” завършващ със “.xsd” или “?xsd”. По този начин схемата е разграничена и е напълно реализуемо нейното откриване. Втората стъпка от подхода е създаването на XML документ отговарящ на ограниченията, описани във вече извлечената XML схема. Генерират се съответните XML елементи според структурните описания в схемата, но без стойности - стойностите за тях се въвеждат на следващата стъпка от подхода. Когато XML схемата има повече от един главен (root) елемент, потребителят трябва да определи, за кой точно от главните елементи ще се генерират данни.

След като структурата на XML документа е създадена се определя начинът за генериране на стойности. За автоматизираното тестване е добре да може да се генерира повече от един набор тестови данни (в случая повече от един XML файл). Ето защо подходът включва създаването на повече от една инстанция на XML документа, като необходимите стойности се параметризират и се попълват със стойности чрез заместване от външен източник на данни, заместване с регулярен израз или заместване с извикване на функция.

За извършване на негативно тестване е предвидено генериране на некоректни данни. Имайки предвид, че данните са в XML формат, то тестовите данни са некоректни, ако нарушават структурата на XML документ или типът зададен от XML схемата.

Подходът е реализиран чрез интерфейс за команден ред (CLI), предвиден най-вече за използване от другите инструменти и посредством които методологията може да се автоматизира.

3.6 Изводи

В тази глава са дефинирани пет подхода за функционално тестване на съставни уеб услуги и са представени четири прототипа на софтуерни инструменти, реализиращи предложените подходи. Три от описаните подходи са нови в областта на АОУ, а именно: подход за изолиране от външни зависимости, подход за инжектиране на неизправности и подход за анализ на даннови зависимости;

Останалите два подхода – за създаване на функционални тестове и за генериране на тестови данни, допълват съществуващи подходи по начин подходящ за тестваните АОУ и предложената, в дисертацията методология.

Предимства на подхода за изолиране от външни зависимости са, че при него се намалява броя на тестваните артефакти (няма stub услуги) и версиите на BPEL процеса могат лесно да се управляват.

Реализираният подход за симулиране на неизправности, чрез Инжекционния инструмент, има добра оперативна съвместимост и приложение в АОУ. С негова помощ може да се провери до каква степен тестваната композиция от уеб услуги или самостоятелна услуга, реагира с правилна обработка на изключения и действия за възстановяване след некоректни входни съобщения.

В дисертацията е предложен подход за анализ на даннови зависимости и софтуерен инструмент, който го реализира. Чрез анализа, който този подход извършва, става ясна връзката между път, условия и променливи в бизнес процес, описан чрез BPEL. Това позволява, чрез задаване на конкретни стойности за променливите в бизнес процеса, да се изпълни точно определен път в него. Това прави предложения подход за анализ на даннови зависимости, приложим при функционално тестване на BPEL процеси, за покритие на пътища.

Надграден е подходът за генериране на тестови данни в XML формат по зададена XSD схема и е програмно реализиран. Важни негови характеристики са, че взема предвид всички ограничения наложени от XSD схемата, предлага създаване на множество XML

инстанции, има възможност за създаване на тестови данни, които са в разрез с XML схемата (за негативно тестване), и предлага ръчно и автоматично специфициране на стойностите за XML инстанциите.

За да се повиши ефективността и да се намали броя на тестовете спрямо тяхното създаване на случаен принцип, е специфициран подход за създаване на функционални тестове, който използва подходите за анализ на даннови зависимости, изолиране от външни зависимости, генериране на тестови данни и създаване на проверки/твърдения. Следвайки описания подход, се създават функционални тестове за покритие на всички пътища в бизнес процес, за изпълнение на конкретни пътища свързани с конкретни тестови цели или изисквания, както и позитивни и негативни тестове, що се отнася до ограниченията наложени от XML схемата на входните данни.

4. Валидация на подходите и методологията

Дейностите по валидиране в дисертация са ориентирани към постигането на две основни цели – оценка и подобрене на постигнатите резултати в предметната област, както по отношение на разработените подходи и прототипи на инструменти, така и на тестовата методология. За да се направи тяхната оценка са въведени критерии и метрики.

Тъй като предложената методология е свързана с тестване на композиция от услуги, то за нейното валидиране е приет подход, чрез който да се докаже, че тя покрива следните критерии:

- Приложимост за СБУ;
- **Пълнота при тестването**, както по отношение на класификацията, свързана с жизнения цикъл на тестваната система (проектиране, разработка и експлоатация), така и по отношение на компонентите, изграждащи архитектурата (операции, услуги, композиции от услуги);
- Позволява автоматизация на тестовия процес;
- **Повишава ефективността на тестовия процес** и качеството на системата, от гледна точка на възможността за тестване на нейните функционални и нефункционални характеристики.

Метриците, които се използват за оценка на предложената методологията са:

- Метрики, които проверяват, дали чрез предложената методология осигурява **функционално тестване**. За целта от една страна се ползват метрики базираните на изисквания, а от друга страна - програмно базирани метрики за покритие. Тестовете за покритие в композиция от уеб услуги, могат да са базирани на управляващия поток или на данновия поток. В [86] Bertolino дефинира така наречената метрика „Социално покритие“, която се изчислява по следния начин:

$$\text{Социално покритие} = \frac{\text{Брой покрити обекти}}{\text{Брой целеви обекти}} \quad (8)$$

Където Социално покритие е персонализирана информация за покритие в даден контекст въз основа на данни за покритие, събрани от подобни потребители; *Брой покрити обекти* са обектите тествани до момента; *Брой целеви обекти* са обектите, които представляват интерес за подобни потребители на системата (използват се коефициенти за подобие).

- Метриците, показващи, че методологията осигурява **тестване на нефункционални характеристики** са разделени основно в две групи – за сигурност и устойчивост и за време

Предложените в дисертацията подходи осигуряват цялостния процес за тестване на композиция от уеб услуги в АОУ. Те са неразделна част от тестовата методология и поради тази причина, валидирането ѝ е тясно свързано и основано на валидирането на включените в нея подходи. За валидирането на предложените подходи и тяхната софтуерна реализация са избрани следните метрики:

- Приложимост на подхода: за тестване в различни сфери, технологии (Оценка:4); за тестване на АОУ (Оценка:3); за тестване на бизнес процеси в АОУ (Оценка:2); за тестване на бизнес процеси, описани с езика BPEL (Оценка:1).
- Пълнота на решението: напълно решава поставената му задача (Оценка:3); решава до голяма степен поставената му задача (Оценка:2); решава само от части поставената му задача (Оценка:1).
- Ниво на автоматизация: може напълно да се автоматизира (Оценка:3); необходима е човешка намеса в начален етап (Оценка:2); необходима е човешка намеса на различни етапи от процеса на реализация на подхода (Оценка:1).
- Сравнение с други подобни подходи: подходът е нов (Оценка:3); съществуват подобни подходи, но с по-ограничена функционалност (Оценка:2); подобен е на други подходи. (Оценка:1).

Предложената в дисертацията методология и подходи се валидират върху СБУ, като се използват софтуерни инструменти, интегрирани в обща платформа за тестване и се оценява тяхната приложимост и ефективност (Задача 5). Технологичната платформа, в която са интегрирани разработените инструментите, описани в предходната глава се нарича TASSA [87]. Софтуерните инструменти се използват като средство за автоматизация и доказване на работоспособността на предложените в дисертацията подходи. С помощта на инструментите, интегрирани в платформата TASSA, е създаден и изпълнен подходящ набор тестовете. На база отчетените резултатите от тестовете и подбраните критерии е извършена оценка на предложените подходи и методология за тестване на композиция от уеб услуги.

Предложената методология и подходи са валидирани върху 3 синхронни (Order Music, Check Payment, Wine Producer) и 1 асинхронен (Travel Reservation Service) бизнес процес. За валидацията на подхода за изолиране на външни зависимости са изпълнени 6 тестови сценария. На база на въведените метрики за валидация и направените експерименти, подходът за изолиране на външни зависимости има обща оценка 9 от 13. Той е нов (Оценка 3), приложим е за тестване на бизнес процеси, описани с езика BPEL (Оценка:1), напълно решава поставената му задача (Оценка:3) и е необходима човешка намеса в начален етап от неговата реализация (Оценка:2).

Валидацията на подхода за инжектиране на неизправности, е извършена чрез 18 тестови сценария, разделени в 5 основни групи (без инжектиране на грешки, симулиране на закъснение на съобщение, симулиране на прекъсване в комуникацията между партнираща услуга и бизнес процес, инжектиране на шум в съобщението от партниращата услуга и промяна в данните изпращани от партниращата услуга). Чрез въведените метрики за валидация, за подхода за инжектиране на неизправности се изчислява обща оценка 10 от 13. Предложеният подход напълно решава поставената му задача (Оценка:3), подходящ е за тестване на АОУ (Оценка:3), предлага повече функции от други подобни подходи (Оценка:2), а за неговата работа е необходима човешка намеса само в начален етап на конфигурация (Оценка:2).

Експериментите, за валидация на подхода за анализ на даннови зависимости показваха, че той е нов (Оценка:3), предполага човешка намеса на различни етапи (Оценка:1) и решава само от части поставената му задача (Оценка:1). Недостатък на предложения начин за анализ на даннови зависимости е, че е подходящ само за тестване на бизнес процеси, описани с езика BPEL (Оценка:1).

Инструментът за генериране на тестови сценарии е валидиран чрез 4 примерни тестови сценария, приложени върху процеса *CheckPayment*. Създадените тестови сценарии осигуряват пълно покритие на пътища в бизнес процеса, а възможностите за дефиниране на твърдения, които инструментът предоставя са напълно достатъчни, за извършване на функционално тестване. На база направената валидация на подхода за създаване на функционални тестове, се изчислява неговата обща оценка, като сбор от метриците: Приложимост на подхода за тестване в различни сфери, технологии (Оценка:4), Пълнота на решението - напълно решава поставената му задача (Оценка:3), Ниво на автоматизация - необходима е човешка намеса на различни етапи от процеса на реализация на подхода

(Оценка:1) и Сравнение с други подобни подходи - Подобен е на други подходи. (Оценка:1). В резултат, общата оценка на валидирания в тази секция подход е 9 от 13. Проведените 8 експеримента, с предложения подход за генериране на тестови данни и съответната му софтуерна реализация, показват, че той предлага автоматично генериране на коректни тестови данни. По този начин се автоматизира процес, който изисква значително време и усилия, а именно създаването на данни в XML формат, отговарящи на редица ограничения описани в съответната XSD схема. Използвайки въведените метрики за валидация, подходът за генериране на тестови данни се определя като приложим за тестване на АОУ (Оценка:3), решаващ до голяма степен поставената му задача (Оценка:2), изискващ човешка намеса в начален етап (Оценка:2) и надграждащ съществуващите подходи (Оценка:2).

Предложената методология е валидирана, чрез тестовата платформа TASSA, в две среди за разработка – NetBeans и Eclipse. Изпълнени са редица тестове и трансформации, които показват различни аспекти на тестването, които методологията покрива.

Валидацията показва, че **предложената в дисертацията методология успешно осигурява функционално тестване**. Чрез него може да се провери до колко системата обект на тестване (композиция от уеб услуги, единична уеб услуга или операция на уеб услуга), отговаря на поставените ѝ функционални изисквания. Освен това, следвайки стъпките и подходите в методологията, създадените функционални тестови сценарии осигуряват пълно покритие на изпълнимите пътища в бизнес процес.

Чрез формулата за Социално покритие (8), може да се изчисли **социалното покритие на предложената методология**, като за целеви обекти може да се вземат изискванията поставени към бизнес процеса, пътищата в него или активностите в неговото VPEL описание. Направените експерименти показаха, че и в трите случая социалното покритие на методологията е максимално, т.е. е равно на 1, тъй като всички обекти са покрити чрез тестове.

В Таблица 7 е представена обобщена оценка на валидираните подходи и базирана на тях оценка на методологията за тестване на композиция от уеб услуги. Стойностите за всеки критерий са представени в проценти спрямо максималната възможна стойност. Използвани са метриците „иновация“, „пълнота“, „автоматизация“ и „приложимост“, които отговарят на въведените, за всеки подход, метрики за оценка. Оценката на методологията варира според използваната комбинация от подходи, при различни нейни приложения.

ТАБЛИЦА 7 ОЦЕНКА НА ПРЕДЛОЖЕНИТЕ В ДИСЕРТАЦИЯТА МЕТОДОЛОГИЯ И ПОДХОДИ

	Иновация	Пълнота	Автомат.	Приложимост	ОБЩО
Изолиране на от външни зависимости	100%	100%	66%	25%	73%
Инжектиране на неизправности	66%	100%	66%	75%	77%
Анализ на даннови зависимости	100%	33%-66%	33%-100%	25%	48%-73%
Създаване на функционални тестове	33%	100%	33%	100%	66%
Генериране на тестови данни	66%	66%-100%	66%	75%	68%-77%
Методология	100%	80%-93%	33%-100%	25%-100%	66%-77%

Заключение

В дисертационния труд е предложена **нова методология за тестване на композиция от уеб услуги**. Тя включва насоки, отговори, стъпки, инструменти за тестване, които да се справят с предизвикателствата свързани с тестването на съставни уеб услуги. Предложената методология надгражда вече съществуващите в литературата в три основни направления - пълнота на тестване, високо ниво на автоматизация и гъвкавост. За част от

дейностите на методологията са създадени нови подходи и реализиращи ги софтуерни инструменти в случаите, за които не е открито адекватно решение. За останалата част от дейностите са разширени или преизползвани по подходящ начин съществуващи подходи и софтуерни инструменти. В резултат е удовлетворена и основната цел на дисертационния труд, а именно **изграждане на единна, стандартизирана тестова методология и широко приложима автоматизирана платформа**, която я реализира.

Предложената методология се състои от **шест основни дейности**: 1) **подготовка** на композиция от уеб услуги, описана като бизнес процес, чрез езика BPEL, която е обект на тестване; 2) **определяне на тестова цел, и анализ на бизнес процеса** и съответните променливи, за да се идентифицират тестови сценарии; 3) **дефиниране на функционални тестове**; 4) **дефиниране на нефункционални тестове**, в това число - тестове за сигурност на база вече създадени функционални тестове, надградени с необходимите артефакти за тестване на сигурността и тестове за натоварване и бързодействие, чрез многократно, паралелно изпълнение на създадени функционални тестове; 5) **изпълнение на тестовите сценарии**; 6) наблюдение и документиране на **резултатите от тестовете**. Дадени са и задачите, които се решават с тях, както и съответните входни и изходни данни, необходими за изпълнението им.

За реализацията на предложената методология са разгледани множество подходи. Подбрани са най-подходящите от тях, **създадени са нови и са допълнени съществуващи подходи** по начин подходящ за тестваните СБУ спрямо предложената методология. **Проучени са десет подхода**, а именно: подход за проверка на съответствие със стандарти; подход за изолиране от външни зависимости; подход за инжектиране на неизправности; подход за създаване на функционални тестове; подход за анализ на зависимости; подход за генериране на тестови данни; подход за създаване на твърдения; подход за тестване на бързодействие; подход за тестване на сигурност и подход за верификация и отчитане на тестови резултати.

Съществен принос на дисертационния труд са **предложените нови подходи и софтуерни инструменти**, които реализират премахване на зависимости от външни услуги, инжектиране на неизправности и анализ на даннови зависимости.

Два **популярни подхода, са разширени и специфицирани** в дисертацията. Това са подход за създаване на функционални тестове и подход за генериране на тестови данни.

Предложената методология и подходи са **валидирани върху четири бизнес процеса** от различен тип и сложност, за които кодът на част от партниращите уеб услугите е наличен, а за други - липсва. Използвани са софтуерни инструменти, интегрирани в **обща тестова платформа**, наречена TASSA. Тя интегрира разработените инструментите според предложените подходи и методология. Чрез нея е реализирано и валидирано функционално тестване на съставни уеб услуги, описани чрез BPEL. Резултатите от направените в дисертацията изследвания потвърждават основната изследователска хипотеза, че е възможно да се разработи методология за тестване на композиции от уеб услуги, която покрива и систематизира всички ключови дейности в тестовия процес и я превръщат в изследователска теза. Потвърждават се и двете допълнителни хипотези.

На база на въведените критерии и метрики в пета глава, е направена оценка на новата методология и включените в нея подходи. Оценката показва, че **предложената методология е иновативна, предлага пълнота при тестването, високо ниво на автоматизация и успешно приложение, както за тестване на композиции от уеб услуги, описани с BPEL, така и за тестване АОУ като цяло.**

Приноси

Научните и научно-приложните приноси могат да бъдат обобщени както следва:

Принос 1. Предложена е нова таксономия за систематизиране и класифициране на литературните източници в предметната област на дисертационния труд;

Принос 2. Предложена е **нова методология за тестване** на бизнес процеси, описани с езика WS-BPEL, която систематизира всички основни дейности в тестовия процес, при наличие на автоматизация;

- Принос 3.** Предложена концепция за цялостна автоматизирана платформа, приложима при тестване на софтуерни приложения, базирани на услуги, както във фазата на проектирането им, така и по време на тяхното изпълнение, при която човешката намеса е минимална;
- Принос 4.** Предложен нов подход и софтуерна реализация на инструмент за премахване на зависимости от външни услуги на бизнес процеси, описани с езика WS-BPEL, при който се извършва симулация - замяната на извикването на операция на външна услуга с константа;
- Принос 5.** Предложен нов подход и софтуерна реализация на инструмент за инжектиране на грешки в бизнес процеси, описани с езика WS-BPEL, който симулира четири основни групи проблеми, характерни за реалната работа на АОУ, има ниска степен на вмешателство и извършва негативно тестване;
- Принос 6.** Предложен нов подход и софтуерна реализация на инструмент за анализ на даннови зависимости в бизнес процес описани с езика WS-BPEL, който представя бизнес процеса чрез граф на BPEL активности и идентифицира условните активности в графа, с цел покритие на пътища;
- Принос 7.** Предложен е нов инструментариум за оценка на ефективност и степен на приложимост, на методологии и съпътстващите ги подходи за тестване на софтуерни системи.

Публикации в тематиката на дисертационния материал

Научните публикации свързани с дисертационния труд са общо 11, от които 3 са публикувани в международни научни списания, а 7 от тях – в сборници на международни конференции, сред които Service Oriented System Engineering и Engineering of Computer-Based Systems. **Принос 1** е представен във **Втора глава** на дисертацията и публикуван в [5] и [7]. **Принос 2** и **Принос 3** са изведени в **Трета глава**, резултатите от която са публикувани в [1], [5], [7], [8], [10] и [11]. **Принос 4**, **Принос 5** и **Принос 6** са представени в **Четвърта глава**, резултатите от която са включени в [1], [2], [3], [4], [5], [6] и [9]. **Принос 7** и резултатите от **Пета глава** са представени в публикации [3], [4], [5], [7], [10] и [11].

Дисертантът е единствен автор на 1 публикация, първи съавтор – на 4, втори съавтор – на 2 и трети съавтор – на 2 публикации.

Публикациите са:

- [1] Ilieva, S., Pavlov, V., Manova, I., & **Manova, D.** (2011). A Framework for Design-Time Testing of Service-Based Applications at BPEL Level. *Serdica Journal of Computing*, 5(4), 367p-384p.
- [2] Manova, I., **Manova, D.**, Pavlov, V., Ilieva, S., & Petrova-Antonova, D. (2011). Fault injection testing of web service business processes. *International Journal on Information Technologies & Security*, Year, 3, 3-12.
- [3] **Manova, D.**, Ilieva, S., Lonetti, F., Bertolino, A., & Bartolini, C. (2011, June). Towards automated robustness testing of BPEL orchestrators. In *Proceedings of the 12th International Conference on Computer Systems and Technologies* (pp. 659-664). ACM.
- [4] **Manova, D.**, Manova, I., Ilieva, S., & Petrova-Antonova, D. (2011, September). faultInjector: A Tool for Injection of Faults in Synchronous WS-BPEL processes. In *Engineering of Computer Based Systems (ECBS-EERC), 2011 2nd Eastern European Regional Conference on the* (pp. 99-105). IEEE.
- [5] Ilieva, S., **Manova, D.**, Manova, I., Bartolini, C., Bertolino, A., & Lonetti, F. (2011, December). An automated approach to robustness testing of BPEL orchestrations. In *Service Oriented System Engineering (SOSE), 2011 IEEE 6th International Symposium on* (pp. 193-203). IEEE.
- [6] Petrova-Antonova, D., Ilieva, S., Manova, I., & **Manova, D.** (2012). Towards Automation Design Time Testing of Web Service Compositions. *e-Informatica*, 6(1), 61-70.
- [7] **Manova, D.**, Ilieva, S., & Petrova-Antonova, D. (2013, June). Testing web service's compositions following TASSA methodology. In *Proceedings of the 14th International Conference on Computer Systems and Technologies* (pp. 185-192). ACM.
- [8] **Манова, Д.**, (2013) Методология за тестване на композиция от веб услуги. *Национална младежка школа и борса за научни идеи в областта на информационните и комуникационните технологии*.
- [9] **Manova D.**, D. Petrova-Antonova, S. Ilieva, I. Manova, (2014, September) XML schema based generation of test data. In *Proceedings of the International Conference on Information Technologies (InfoTech-2014)* 18-19 September 2014.

- [10] Petrova-Antonova, D., Ilieva, S., & **Manova, D.** (2015, May). TASSA: Testing Framework for web service orchestrations. In *Proceedings of the 10th International Workshop on Automation of Software Test* (pp. 8-12). IEEE Press.
- [11] Petrova-Antonova, D., Ilieva, S., & **Manova, D.** (2016). TASSA: A Testing as a Service Framework for Web Service Compositions. TASSA: A Testing as a Service Framework for Web Service Compositions. In *Proceedings of the International Workshop on domain specific Model-based Approaches to vErification and validaTiOn, 33-42, 2016, Rome, Italy.*

Цитирания

Открити са **общо 11 цитирания**. От тях 4 са в международни списания, 5 в сборници от международни конференции и 2 статии са цитирани в дисертационни трудове. Публикация [5] е цитирана в пет научни текста, публикация [3] – в два научни текста, а всяка от публикации [2], [6], [7] и [10] – в по един научен материал.

Благодарности

Изказвам своята благодарност и признателност на научния си ръководител доц. д-р Десислава Петрова-Антонова, за професионалното отношение и ценните напътствия, за съдействието при изготвянето на дисертацията и публикациите свързани с нея.

Благодаря на проф. Силвия Илиева, доц. д-р Александър Димов и колегите от катедра „Софтуерни технологии“ на СУ „Св. Кл. Охридски“ за съдействието и проявената отзивчивост.

Благодаря на колегите от „Рила Солюшънс“ за полезните съвети, за проявеното разбиране и търпение, за предоставената технологична среда.

Благодаря на семейството си за търпението и моралната подкрепа, която ми указаха.

Библиография

- [1] Jordan, D., Evdemon, J., Alves, A., Arkin, A., Askary, S., Barreto, C., ... & Guizar, A. (2007). Web services business process execution language version 2.0. OASIS standard, 11(120), 5.
- [2] Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004), 1-26.
- [3] SOAtest API Testing - Parasoft. (n.d.). Retrieved 10 April 2016, from <<https://www.parasoft.com/product/soatest/>>
- [4] Kankanamge, C. (2012). Web services testing with soapUI. Packt Publishing Ltd.
- [5] TestMaker Test Deployment - Cloud and Grid Testing. (n.d.). Retrieved 10 April 2016, from <<http://www.pushtotest.com/cloud-and-grid>>
- [6] Storm. (n.d.). Retrieved 10 April 2016, from <<https://storm.codeplex.com/Wikipage?ProjectName=storm>>
- [7] Postman. (n.d.). Retrieved 10 April 2016, from <<https://www.getpostman.com/products>>
- [8] Dai, N., Mandel, L., & Ryman, A. (2007). Eclipse Web tools platform: developing Java Web applications. Pearson Education.
- [9] Erinle, B. (2013). Performance Testing with JMeter 2.9. Packt Publishing Ltd.
- [10] Cavalli, A., Cao, T. D., Mallouli, W., Martins, E., Sadovykh, A., Salva, S., & Zaïdi, F. (2010, July). Webmov: A dedicated framework for the modelling and testing of web services composition. In *Web Services (ICWS), 2010 IEEE International Conference on* (pp. 377-384). IEEE.
- [11] JOpera for Eclipse | Process Support for More than Web Services. (n.d.). Retrieved February 17, 2016, from <<http://www.jopera.org/>>
- [12] Chandrasekaran, S., Miller, J. A., Silver, G. S., Arpinar, B., & Sheth, A. P. (2003). Performance analysis and simulation of composite web services. *Electronic Markets*, 13(2), 120-132.
- [13] Bradshaw, D., Kennedy, M., & West, C. (2005). Oracle BPEL Process Manager Developer's Guide. Oracle Corporation, California.
- [14] IBM - Rational Service Tester for SOA Quality. (2016, January 1). Retrieved April 12, 2016, from <<https://www.ibm.com/software/products/en/servicetest>>
- [15] Harris, T. (2007). Soa test methodology. *White paper*.
- [16] AppLabs. (2007, May). Approach to Testing SOA Applications. *White paper*.
- [17] Bertolino, A., De Angelis, G., Frantzen, L., & Polini, A. (2009). The plastic framework and tools for testing service-oriented applications. In *Software Engineering* (pp. 106-139). Springer Berlin Heidelberg.
- [18] Mei, L., Chan, W. K., & Tse, T. H. (2009, August). Data flow testing of service choreography. In *Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering* (pp. 151-160). ACM.
- [19] Manoja Nara. Testing Web Services based SOA. *White paper*.
- [20] Lenz, C., Chimiak-Opoka, J., & Breu, R. (2007, May). Model driven testing of SOA-based software. In *Proceedings of the Workshop on Software Engineering Methods for Service-Oriented Architecture (SEMSOA2007)* (pp. 99-110).
- [21] Felderer, M., Zech, P., Fiedler, F., & Breu, R. (2010, August). A tool-based methodology for system testing of service-oriented systems. In *Advances in System Testing and Validation Lifecycle (VALID), 2010 Second International Conference on* (pp. 108-113). IEEE.
- [22] Corradini, F., De Angelis, F., De Angelis, G., Polini, A., & Polzonetti, A. (2010, June). A model checking strategy to test services in orchestrations. In *Future Network and Mobile Summit, 2010* (pp. 1-8). IEEE.
- [23] Chan, W. K., Cheung, S. C., & Leung, K. R. (2005, September). Towards a metamorphic testing methodology for service-oriented software applications. In *Quality Software, 2005.(QSIC 2005). Fifth International Conference on* (pp. 470-476). IEEE.
- [24] Khan, T. A., & Heckel, R. (2009, September). A methodology for model-based regression testing of web services. In *Testing: Academic and Industrial Conference-Practice and Research Techniques, 2009. TAIC PART'09*. (pp. 123-124). IEEE.
- [25] Definition of METHODOLOGY. (n.d.). Retrieved February 17, 2016, from <<https://www.merriam-webster.com/dictionary/methodology>>

-
- [26] Standards - W3C. (n.d.). Retrieved February 17, 2016, from <<https://www.w3.org/standards/>>
- [27] Yuan, Y., Li, Z., & Sun, W. (2006, October). A graph-search based approach to BPEL4WS test generation. In *Software Engineering Advances, International Conference on* (pp. 14-14). IEEE.
- [28] Lertphumpanya, T., & Senivongse, T. (2008). Basis path test suite and testing process for WS-BPEL. *WSEAS Transactions on Computers*, 7(5), 483-496.
- [29] Team, B. (n.d.). BPEL Validator. Retrieved February 17, 2016, from <<https://eclipse.org/bpel/developers/validator/>>
- [30] CA Service Virtualization - CA Technologies. (n.d.). Retrieved February 17, 2016, from <<https://www.ca.com/us/products/ca-service-virtualization.html>>
- [31] Service Virtualization: Application & Data Simulation Software | Hewlett Packard Enterprise. (n.d.). Retrieved February 17, 2016, from <<http://www8.hp.com/us/en/software-solutions/service-virtualization/>>
- [32] IBM - Rational Test Virtualization Server. (2016, January 1). Retrieved February 17, 2016, from <<http://www.ibm.com/software/products/en/rtnvs>>
- [33] Web Services API Mocking Overview | SOAP Mocking. (n.d.). Retrieved February 17, 2016, from <<https://www.soapui.org/soap-mocking/service-mocking-overview.html>>
- [34] Fletcher, R. (2011). Betamax - Record & replay HTTP traffic. Retrieved February 17, 2016, from <<http://betamax.software/>>
- [35] Fu, C., Ryder, B. G., Milanova, A., & Wonnacott, D. (2004). Testing of java web services for robustness. *ACM SIGSOFT Software Engineering Notes*, 29(4), 23-34.
- [36] Fugini, M. G., Pernici, B., & Ramoni, F. (2007, September). Quality analysis of composed services through fault injection. In *International Conference on Business Process Management* (pp. 245-256). Springer Berlin Heidelberg.
- [37] Kuk, S. H., & Kim, H. S. (2009, December). Robustness testing framework for web services composition. In *Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific* (pp. 319-324). IEEE.
- [38] Bessayah, F., Cavalli, A., Maja, W., Martins, E., & Valenti, A. W. (2010). A fault injection tool for testing web services composition. In *Testing-Practice and Research Techniques* (pp. 137-146). Springer Berlin Heidelberg.
- [39] Yan, J., Li, Z., Yuan, Y., Sun, W., & Zhang, J. (2006, November). BPEL4WS unit testing: Test case generation using a concurrent path analysis approach. In *Software Reliability Engineering, 2006. ISSRE'06. 17th International Symposium on* (pp. 75-84). IEEE.
- [40] Zheng, Y., Zhou, J., & Krause, P. (2007, April). A model checking based test case generation framework for web services. In *Information Technology, 2007. ITNG'07. Fourth International Conference on* (pp. 715-722). IEEE.
- [41] Cao, H., Ying, S., & Du, D. (2006, September). Towards model-based verification of BPEL with model checking. In *Computer and Information Technology, 2006. CIT'06. The Sixth IEEE International Conference on* (pp. 190-190). IEEE.
- [42] Daghighzadeh, M., & Babamir, M. (2013, October). An ABC based approach to test case generation for BPEL processes. In *Computer and Knowledge Engineering (ICCKE), 2013 3th International eConference on* (pp. 272-277). IEEE.
- [43] Lertphumpanya, T., & Senivongse, T. (2008). Basis path test suite and testing process for WS-BPEL. *WSEAS Transactions on Computers*, 7(5), 483-496.
- [44] Li, Z. J., Tan, H. F., Liu, H. H., Zhu, J., & Mitsumori, N. M. (2008). Business-process-driven gray-box SOA testing. *IBM Systems Journal*, 47(3), 457-472.
- [45] Van der Aalst, W. M., Dumas, M., Ouyang, C., Rozinat, A., & Verbeek, E. (2008). Conformance checking of service behavior. *ACM Transactions on Internet Technology (TOIT)*, 8(3), 13.
- [46] Guangquan, Z., Mei, R., & Jun, Z. (2007, December). A business process of web services testing method based on uml2.0 activity diagram. In *Intelligent Information Technology Application, Workshop on* (pp. 59-65). IEEE.
- [47] Wang, S., & Capretz, M. A. (2009, July). A dependency impact analysis model for web services evolution. In *Web Services, 2009. ICWS 2009. IEEE International Conference on* (pp. 359-365). IEEE.
- [48] Bai, X., Dong, W., Tsai, W. T., & Chen, Y. (2005, October). WSDL-based automatic test case generation for web services testing. In *Service-Oriented System Engineering, 2005. SOSE 2005. IEEE International Workshop* (pp. 207-212).
- [49] Ma, C., Du, C., Zhang, T., Hu, F., & Cai, X. (2008, December). WSDL-based automated test data generation for web service. In *Computer Science and Software Engineering, 2008 International Conference on* (Vol. 2, pp. 731-737). IEEE.

-
- [50] Offutt, J., & Xu, W. (2004). Generating test cases for web services using data perturbation. *ACM SIGSOFT Software Engineering Notes*, 29(5), 1-10.
- [51] Sneed, H. M., & Huang, S. (2006, September). WSDLTest-a tool for testing web services. In *Web Site Evolution, 2006. WSE'06. Eighth IEEE International Symposium on* (pp. 14-21). IEEE.
- [52] Bartolini, C., Bertolino, A., Marchetti, E., & Polini, A. (2009, April). WS-TAXI: A WSDL-based testing tool for web services. In *Software Testing Verification and Validation, 2009. ICST'09. International Conference on* (pp. 326-335).
- [53] Eclipse XML Editor Help - Eclipse Platform. (n.d.). Retrieved February 17, 2016, from <<http://help.eclipse.org/neon/index.jsp?topic=%2Forg.eclipse.wst.xmleditor.doc.user%2Ftopics%2Fcxwxmledt.html>>
- [54] Sorens, M. (2009). Taking XML Validation to the Next Level: Introducing CAM. *DevX*. Retrieved from <<http://www.devx.com/xml/Article/41066>>
- [55] DataGen - XML Test Data Generation Tool (2012). Retrieved February 17, 2016, from <<http://iwm.uni-koblenz.de/datagen/index.html>>
- [56] Bergmann, V. (2012). Databene Benerator 0.7. 6 manual.
- [57] Petrova-Antonova, D., Kuncheva, K., & Ilieva, S. (2015, July). Automatic generation of test data for XML schema-based testing of web services. In *Software Technologies (ICSOFIT), 2015 10th International Joint Conference on* (Vol. 1, pp. 1-8). IEEE.
- [58] Durand, J., Green, S. D., Kulvatunyou, S., & Rutt, T. (2009). Test Assertions on steroids for XML artifacts. In *Proceedings of Balisage: The Markup Conference* (Vol. 3).
- [59] Green, S. D., Kostovarov, D. Test Assertions Guidelines Version 1.0. (2013). OASIS TAG Technical Committee, Retrieved February 17, 2016, from <<http://docs.oasis-open.org/tag/guidelines/v1.0/testassertionsguidelines.html>>
- [60] Tennison, J. (2004). Validating XML with Schematron. In *Beginning XSLT* (pp. 626-660). Apress.
- [61] Bacon, T., & Martin, J. (2016). XMLUnit - Unit Testing XML for Java and .NET. Retrieved February 18, 2016, from <<http://www.xmlunit.org/>>
- [62] Active Endpoint. (2007). ActiveBPEL. Retrieved 2013, <www.active-endpoints.com/active-bpel-engineoverview.htm>
- [63] Jamen, L., & Ghosh, S. Oracle Fusion Middleware Performance and Tuning Guide 11g (2011) Retrieved February 18, 2016, from <https://docs.oracle.com/cd/E21764_01/core.1111/e10108/bpel.htm#ASPER99175>
- [64] Johnson, C., & Newport, B. (2006). Develop high performance J2EE threads with WebSphere Application Server. Retrieved from <http://www.ibm.com/developerworks/websphere/techjournal/0606_johnson/0606_johnson.html>
- [65] Chen, W., Wei, J., Wu, G., & Qiao, X. (2008, November). Developing a concurrent service orchestration engine based on event-driven architecture. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 675-690). Springer Berlin Heidelberg.
- [66] Curbera, F., Khalaf, R., Nagy, W. A., & Weerawarana, S. (2006). Implementing BPEL4WS: the architecture of a BPEL4WS implementation. *Concurrency and Computation: Practice and Experience*, 18(10), 1219-1228.
- [67] Gunarathne, T., Premalal, D., Wijethilake, T., Kumara, I., & Kumar, A. (2007). BPEL-Mora: lightweight embeddable extensible BPEL engine. In *Emerging Web Services Technology* (pp. 3-20). Birkhäuser Basel.
- [68] Apache ODE Team. Apache ODE BPEL Engine - Architectural Overview. (2013). Retrieved February 18, 2016, from <<http://ode.apache.org/developerguide/architectural-overview.html>>
- [69] OASIS. (2011). Web Services Quality Factors Version 1.0. OASIS Committee Specification 01. <<http://docs.oasis-open.org/ws-qm/WS-Quality-Factors/v1.0/cs01/WS-Quality-Factors-v1.0-cs01.html>>
- [70] Li, B., Tang, X. Y., & Lv, J. (2005, August). The research and implementation of services discovery agent in web services composition framework. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on* (Vol. 1, pp. 78-84). IEEE.
- [71] Hwang, S. Y., Wang, H., Tang, J., & Srivastava, J. (2007). A probabilistic approach to modeling and estimating the QoS of web-services-based workflows. *Information Sciences*, 177(23), 5484-5503.
- [72] Mukherjee, D., Jalote, P., & Nanda, M. G. (2008, December). Determining QoS of WS-BPEL compositions. In *International Conference on Service-Oriented Computing* (pp. 378-393). Springer Berlin Heidelberg.
- [73] Bartoletti, M., Degano, P., & Ferrari, G. L. (2006). Plans for service composition. In *Workshop on Issues in the Theory of Security (WITS)*.

-
- [74] Yuan, E., & Tong, J. (2005, July). Attributed based access control (ABAC) for web services. In *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on*. IEEE.
- [75] Haibo, S., & Fan, H. (2005, October). A context-aware role-based access control model for web services. In *e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on* (pp. 220-223). IEEE.
- [76] Babamir, S. M., Babamir, F. S., & Karimi, S. (2011, February). Design and evaluation of a broker for secure web service composition. In *Computer Networks and Distributed Systems (CNDS), 2011 International Symposium on* (pp. 222-226). IEEE.
- [77] Anisetti, M., Ardagna, C. A., Bezzi, M., Damiani, E., Kaluvuri, S. P., & Sabetta, A. (2014). A Certification-Aware Service-Oriented Architecture. In *Advanced Web Services* (pp. 147-170). Springer New York.
- [78] Biskup, J., Carminati, B., Ferrari, E., Muller, F., & Wortmann, S. (2007, July). Towards secure execution orders for compositeweb services. In *Web Services, 2007. ICWS 2007. IEEE International Conference on* (pp. 489-496). IEEE.
- [79] Zhou, B., Llewellyn-Jones, D., Shi, Q., Asim, M., Merabti, M., & Lamb, D. (2012, December). A Compose Language-Based Framework for Secure Service Composition. In *Cyber Security (CyberSecurity), 2012 International Conference on* (pp. 195-202). IEEE.
- [80] Martinelli, F., Matteucci, I., Petrocchi, M., & Wiegand, L. (2012, August). A formal support for collaborative data sharing. In *International Conference on Availability, Reliability, and Security* (pp. 547-561). Springer Berlin Heidelberg.
- [81] Roth, H., Schiefer, J., & Schatten, A. (2006, June). Probing and monitoring of WSBPEL processes with web services. In *E-Commerce Technology, 2006. The 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services, The 3rd IEEE International Conference on* (pp. 30-30). IEEE.
- [82] Yee, C. K. (2007). *Design and Implementation of Test Case Generation Tool for BPEL Unit Testing* (Doctoral dissertation, thesis).
- [83] Baresi, L., Ghezzi, C., & Guinea, S. (2004, November). Smart monitors for composed services. In *Proceedings of the 2nd international conference on Service oriented computing* (pp. 193-202). ACM.
- [84] Baresi, L., Bianculli, D., Ghezzi, C., Guinea, S., & Spoletini, P. (2007). Validation of web service compositions. *IET software*, 1(6), 219-232.
- [85] Beeri, C., Eyal, A., Milo, T., & Pilberg, A. (2007, September). Monitoring business processes with queries. In *Proceedings of the 33rd international conference on Very large data bases* (pp. 603-614). VLDB Endowment.
- [86] Miranda, B., & Bertolino, A. (2014, May). Social coverage for customized test adequacy and selection criteria. In *Proceedings of the 9th International Workshop on Automation of Software Test* (pp. 22-28). ACM.
- [87] Petrova-Antonova, D., Ilieva, S., & Manova, D. (2015, May). TASSA: Testing Framework for web service orchestrations. In *Proceedings of the 10th International Workshop on Automation of Software Test* (pp. 8-12). IEEE Press.