

Ш.Е. Бакытова<sup>1</sup>, Г.А. Камалова<sup>1</sup>

<sup>1</sup>Жеңгір хан атындағы Батыс Қазақстан аграрлық-техникалық университеті,  
Орал қ, Қазақстан

## ВЕБ-СЕРВИСТЕР МЕН СЕРВИСКЕ (ҚЫЗМЕТ КӨРСЕТУГЕ) БАҒЫТТАЛҒАН ҚОСЫМШАЛАРДЫ МОДЕЛЬДЕУ ТӘСІЛІНІҢ МҮМКІНШІЛІКТЕРІ

---

---

**Түйіндеме.** Әр саланың өз құрылымы, даму үдерісі, жаңалығы болатыны белгілі. Мақаламен жұмыс барысында веб-сервистер мен сервиске бағытталған қосымшаларды, веб-қызметтерді құруға арналған екі платформа талданды. J2EE платформасы (Java 2 Enterprise Edition) кез-келген сәулет пен операциялық жүйелер үшін Java қосымшаларына негізделген веб-қызметтерді құруға мүмкіндік береді. Microsoft. NET платформасы ASP.NET технологиясында Visual Basic, C++, C# сияқты бағдарламалау тілдері арқылы веб-қызметтерді құруға болады. Алынған екі платформаны талдау және салыстыру барысында сервиске бағытталған қосымшаларды модельдеу тәсілінің мүмкіншіліктері анық көрсетіледі және нақты мысалдармен жұмыс жасалады.

**Түйінді сөздер:** веб-қосымшалар, сервиске бағытталған сәулет (SOA, Service-Oriented Architecture), XML, SOAP, WSDL , UDDI, сервиске бағытталған қосымша.

• • •

**Аннотация.** Каждая отрасль имеет свою структуру, свой процесс развития, свою новизну. В ходе работы над статьей были проанализированы две платформы для создания веб-сервисов и сервис-ориентированных приложений, веб-сервисов. Платформа J2EE (Java 2 Enterprise Edition) позволяет создавать веб-сервисы на основе приложений Java для любой архитектуры и операционных систем Microsoft. Платформа NET ASP.NET технология позволяет создавать веб-сервисы через такие языки программирования, как Visual Basic, C++, C#. В ходе анализа и сравнения двух полученных платформ наглядно демонстрируются особенности подхода к моделированию сервисно-ориентированных приложений и работа на конкретных примерах.

**Ключевые слова:** веб-приложения, сервис-ориентированная архитектура (SOA, Service-Oriented Architecture), XML, SOAP, WSDL , UDDI, сервис-ориентированное приложение.

• • •

**Abstract.** Each industry has its own structure, its own development process, and its own novelty. While working on the article, we analyzed two platforms for creating web services and service-oriented applications, web services. The J2EE platform (Java 2 Enterprise Edition) allows to create web services based on Java applications for any architecture and operating systems. Microsoft the NET platform ASP.

NET technology allows to create web services using programming languages such as Visual Basic, C++, and C#. During the analysis and comparison of the two platforms we have obtained the features of the approach to modeling service-oriented applications and clearly demonstrated working on specific examples.

**Keywords:** web applications, service-Oriented Architecture (SOA, Service-Oriented Architecture), XML, SOAP, WSDL , UDDI, service-oriented application.

**Кіріспе.** Соңғы кездері сервиске бағытталған сәулет (SOA, Service-Oriented Architecture) негізінде қосымшаларды компоненттік құрастыру кеңінен танылып жүр. SOA қосымшасы көптеген бизнес қосымшаларындағы компоненттерге негізделген. Әрбір осындай компонент SOA (1-ші сурет) жүйесінің модулі болып табылады. Сервиске бағытталған сәулет тәсілін пайдалану ірі кәсіпорынға өзінің инфрақұрылымын одан әрі дамыту кезінде біршама артықшылықтарға қол жекізуге мүмкіндік береді.



1 сурет – SOA Құрамдас қосымшаларды құрастыру

Қызметке бағытталған қосымшаларды құру үшін веб-қызметтер технологиясын қолданудың болашағы зор. Веб-қызметтер технологиясы қосымшаларға және олар орналастырылған платформаға, сондай-ақ олар жазылған бағдарламалау тіліне қарамастан бір-бірімен өзара әрекеттесуге мүмкіндік береді. Веб-қызметте стандартталған XML хабарламалары арқылы қашықтан шақырылатын операциялар жиынтығын

сипаттайтын бағдарламалық интерфейсі болады. Дәстүрлі веб-қосымшалардан веб-қызметтің айырмашылығы сол, ол соңғы пайдаланушыларға қызмет көрсетпейді. Өйткені оның пайдаланушы интерфейсі жоқ. Оның басты міндеті – веб-қосымшалар, мобильді немесе консольдік қосымшалар сияқты басқа қосымшаларға қызмет көрсету [1].

**Веб-қызметтерге бағытталған тұжырымдамалар:** Сервистерді пайдалана отырып, ақпараттық жүйелерді құру веб-сервистерге (Web Services Architecture - WSA) бағытталған сәулет ұғымына алып келді. WSA қосымшаларын пайдалану аясында қолданылатын веб-қызметтерді дамыту және модельдеу мәселелерін зерттейді. Сондықтан болашақта “қызмет” сөзі веб – қызмет деген мағынада түсініледі. WSA қосымшаларының негізгі құрылымдық элементі – оның негізгі компоненті ретінде қарастыруға болатын қызмет. Сервиске бағытталған архитектураның және WSA - ның тағы бір негізгі құрамдас бөлігі – корпоративті қызмет көрсету шинасы (Enterprise Service Bus-ESB). ESB тұтынушыларға және қызмет өндірушілеріне тікелей өзара тәуелділікті орнатпай-ақ, бір-бірімен өзара әрекеттесуге мүмкіндік береді. Бұл барлық корпоративтік қызметтер мен қосымшаларды біріктіретін деректерді түрлендіру және беру, басқару және бақылау функцияларын жүзеге асыратын делдал десек те болады.

Осы қызметті жүзеге асыру тұрғысынан WSA веб-қызметтердің стандарттарына, құралдарына және технологияларына сәйкес келетін бағдарламалаудың моделі болып табылады. Егер SOA белгілі бір технологияға қосылмаса, көптеген технологияларды қолдана отырып жүзеге асырылуы мүмкін болса, онда WSA негізгі бағдарламалық жасақтама компоненттері ретінде веб-қызметтерді пайдалануға бағытталған. Осылайша, WSA бұл белгілі бір интерфейстері бар тәуелсіз қызметтерді пайдаланумен сипатталады. Олар өз міндеттерін орындау үшін белгілі бір стандартты жолмен шақырылуы мүмкін. Бұл жағдай қызметтер алдын-ала оларды шақыратын бағдарлама туралы ештеңе білмеген, бағдарлама қызметтердің өз міндеттерін қалай орындайтынынан хабарсыз болғанда да ғана орын алады.

WSA-ны ақпараттық жүйелер архитектурасының стилі ретінде қарастыруға болады. Ол төмен байланысқан және өзара әрекеттесетін веб-қызметтерді біріктіру арқылы жасалған қосымшаларды жасауға мүмкіндік береді. Бұл қызметтер қатаң анықталған платформалық-тәуелсіз интерфейс негізінде өзара әрекеттеседі. Интерфейс анықтамасы қызметтің жасалу тіліне тәуелді болудан арылтады. Мысалы, C# - та жазылған, .net платформаларында және Java платформаларында жұмыс істейтін Java қызметтері жалпы құрама қосымшамен қатар сәтті болуы мүмкін. Кейбір платформаларда жұмыс істейтін қосымшалар басқа платформаларда жұмыс істейтін қызметтерді шақыра алады. Бұл компоненттерді қайта пайдалануды жеңілдетеді.

WSA келесі сипаттамаларға ие:

- Веб-қызметтерді көптеген есептеу жүйелеріне таратуға болады. Жергілікті немесе ғаламдық желілерді қолдана отырып, өзара әрекеттесуге қабілетті;
- Веб-сервистердің интерфейсі оларды іске асырған технологияға немесе платформаға байланысты емес;
- Қажетті веб-қызметтерді іздеуге және қосуға болады [2,3].

**Веб-сервистер технологиясы:** Веб-қызмет – URI (Unified Resource Identifier) арқылы анықталатын бағдарламалық модуль. Оның сыртқы интерфейстері мен байланыстары XML арқылы анықталады және сипатталады. Бұл бағдарламалық жүйелер кейбір интернет протоколы арқылы жіберілген XML хабарламаларын қолдана отырып, веб-қызметпен оның сипаттамасында анықталған стильде өзара әрекеттесе алады. Осыған байланысты веб-қызметтер технологиясы XML технологиясына негізделген ашық стандартты екенін көруге болады. Осылайша, веб-қызметтер қосымшаларға, олар орналастырылған платформаға, сондай-ақ, олар жазылған бағдарламалау тіліне қарамастан бір-бірімен өзара әрекеттесуге мүмкіндік береді.

**Веб-сервистің қызметтері мен мақсатары:** Әзірленген қызметтің негізгі мақсаты – өндірістік типтегі білім базасын құру процесін қолдау болып табылады. Олардың негізгі функциялары:

- Жеке пайдаланушылар мен белгілі бір аймақтың қол жетімділігі мен артықшылықтарын теңестіру мүмкіндігімен білім базаларының модельдерінің жобаларын жасау;
- Xmi MOF, XMI UML және XML FreeMind форматтарының тұжырымдамалық және танымдық модельдерін талдау негізінде пәндік аймақтың моделін автоматты түрде қалыптастыру;
- RVML (rule Visual Modeling Language) тілін пайдалана отырып, өнім түріндегі білім базаларын жобалау;
- CLIPS форматындағы білім базаларын өндірістік білім базаларының модельдері негізінде әзірлеу.

Белгілі бір функцияларды жүзеге асыру үшін қызмет архитектурасы жасалынып көрсетілді (2-сурет.) Берілген суретте серверлік және клиенттік бөліктер бөліп көрсетілген.

**Негізгі модульдер және олардың мақсаты:**

Пайдаланушылар интерфейсі – көмекші басқару тақталары, кеңестер арқылы пайдаланушылардың ыңғайлы, жылдам және тиімді өзара әрекеттесуін қамтамасыз етеді;

Сервистің әкімшілік модулі – қолжетімділік құқықтарын анықтау мүмкіндігін қамтамасыз етеді:

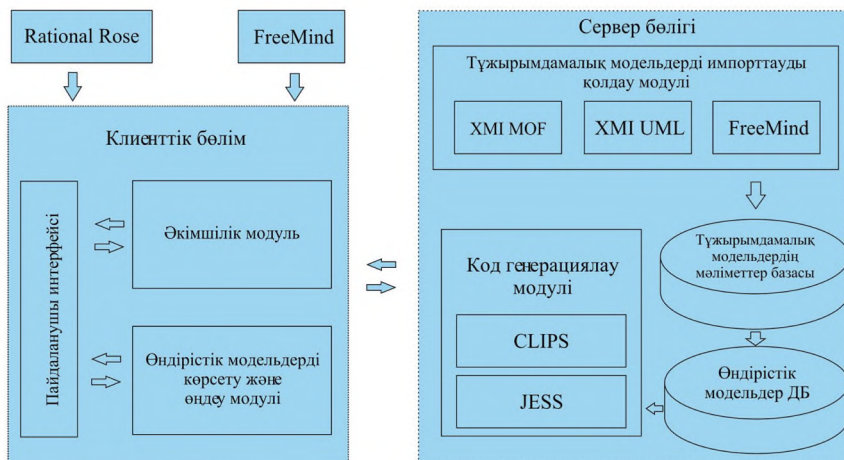
Қызметтің әртүрлі пайдаланушылары үшін

Тұжырымдамалық модельдерді импорттауды қолдау модульдері (XMI MOF, XMI UML, FreeMind);

FreeMind және Rational Rose-де жасалған XML форматындағы файлдарды импорттау мүмкіндігін қамтамасыз етеді. Егер жүктелетін XML файлы дұрыс болса, онда талдау нәтижесінде ұғымдар, байланыстар, атрибуттар, олардың параметрлері және т.б. бөлінеді;

Өнім модельдерін clips форматына бейнелеу модулі; өнім моделі негізінде CLP форматында файлды автоматты түрде қалыптастыруды қамтамасыз етеді;

Тұжырымдамалық модельдер туралы ақпаратты сақтауға арналған деректер базасы жобалау кезінде сақталатын ақпараттың сипаты мен шешілетін міндеттердің ерекшелігіне сәйкес деректер базасын үш бөлікке сегменттеу туралы шешім қабылданды. Олар жиынтығында сервистің жұмысын қамтамасыз етеді [4].



2 сурет - Веб-сервистің архитектурасы

**Веб-қызметтер технологиясының стандарттары:** Веб-қызметтер келесі негізгі веб-стандарттарға негізделген:

- XML (құрылымдық деректерді сақтауға және беруге арналған кеңейтілген белгілеу тілі);
- SOAP (XML негізіндегі хабар алмасу протоколы);
- WSDL (веб-қызметтерді сипаттау тілі);
- UDDI (әмбебап тану, сипаттау және интеграция интерфейсі).

**Семантикалық веб-қызметтер:** Семантикалық желі тұжырымдамасын W3C төрағасы Тим Бернерс-Ли 2001 жылы W3C консорциу-

мының сегізінші конференциясында сөз сөйлеп, «Scientific American» журналында тиісті трактат жариялады деген ақпарат В.А Вишняковтың «Использование интеллектуальных и блокчейн технологий в информационном управлении» атты мақаласында айтылған. Семантикалық веб-қызметтердің дәстүрлі веб-кеңістіктен айырмашылығы – семантикалық желінің әр бетінде екі тілде ақпарат бар: табиғи (шолғыш көрсететін) және арнайы (адамның көзінен жасырылған, бірақ зияткерлік бағдарламалық агенттер үшін түсінікті). БА пайдаланушылардың тапсырмалары бойынша ақпарат көздерін табады, деректерді сұрайды, оларды іздеу критерийлеріне сәйкестігін тексереді, содан кейін жауапты пайдаланушыларға ыңғайлы түрде береді. Семантикалық желі үш қағидаға негізделген агрегация, қауіпсіздік және логика. Біріктіру – деректермен бөлісу. Семантикалық веб-қызметтерде мәселені шешу кезінде кез-келген деректерді қолдануға болады, олардың негізінде семантикалық ақпарат жасалады. Семантикалық ақпарат семантикалық желінің негізін құрайды және белгілі бір пән саласы мен олардың арасындағы қатынастар ұғымдарының кейбір ресми тілдегі сипаттамасын білдіреді. Семантикалық желіге деген сенімділікті қамтамасыз ететін қауіпсіздік сандық қолтаңбаларға негізделген. Оны агенттер мен компьютерлер ақпараттың сенімді көзден алынғанын тексеру үшін қолдана алады.

Семантикалық веб-қызметтер стандартты веб-сервис негізінде құрастырылған. Семантикалық веб-қызметтердің көп деңгейлі архитектурасы (3-сурет.) көрсетілген [5].



3 сурет - Веб-қызметтер мен қызметтің семантикалық сипаттамасы

### Веб-қызметтерді дамытудың негізгі платформалары.

#### Enterprise Edition (J2EE™) платформасы платформасы.

Күрделі таратылған қосымшаларды қамтитын электронды бизнестің қазіргі қарқынды дамып келе жатқан өлемінде кәсіпорын қосым-

шаларының нарыққа мүмкіндігінше тез шығарылуы маңызды. Яғни, қашықтықтағы байланыс қызметтері, ат қою, тұрақты деректер, қорғау немесе транзакцияларды басқару сияқты жүйелік деңгейдегі қызметтерді дамытуға уақыт бөле алмайды. Жобаны әзірлеу тобы портативті қайта пайдалануға болатын компоненттерді жасап, енгізуі керек. Enterprise Edition (J2EE™) платформасы осы мақсатта арнайы жасалған. Ол Java-да таратылған, көп деңгейлі, компоненттерге негізделген қосымшаларды әзірлеу және іске қосу үшін дұрыс құжатталған әрі стандартталған ортаны ұсынады. Бұл орта автоматты түрде қосымшаны құру кезінде төменгі деңгейдегі жұмыстың көп бөлігін орындайды. Мысалы, қашықтан қосылу қызметтері, ат қою, тұрақты деректер, транзакцияларды қорғау және басқару әзірлеушілерге қосымшаның логикасына назар аударуға мүмкіндік береді.

J2EE технологиясы (Java 2 Enterprise Edition) корпоративтік таратылған көп буынды қосымшаларды құруға арналған стандарт болып табылады. Ол стандарттар жиынтығы мен маңызды жүйелік қызметтерді қамтитын айқын және икемді архитектураны ұсына отырып, жүйелік сәулетшілердің, бағдарлама жасайтындардың, таратылған АЖ жобалаушылары мен әзірлеушілерінің жұмысын айтарлықтай жеңілдетуге мүмкіндік береді. J2EE Enterprise JavaBeans (EJB) компоненттік моделі, Java Servlets және Java Server Pages (JSP)-пайдаланушылардың әрекеттеріне динамикалық жауап беруге арналған веб-қосымшалардың стандарттары және JDBC дерекқорына кіру стандарты сияқты стандарттарды біріктіреді.

J2EE технологиясының артықшылықтары:

- Объектінің орналасқан жерін анықтау және объектілерге қол жеткізу операцияларының жоғары ашықтығын қамтамасыз ететін икемді және дәйекті объект моделі;

- Нысандар күйге ие, оларды жаһандық түрде анықтауға болады және сілтемелер клиенттен клиентке және машинадан машинаға оңай беріледі;

- Объектілер бейрезидентті де, сақталған да болуы мүмкін;

- Танымал және қарқынды дамып келе жатқан Java бағдарлама-лау тілін қолдану арқылы құрылған;

- Нысандарды мәні бойынша беру мүмкіндігі;

- Іске асырудың салыстырмалы қарапайымдылығы;

- JDBC, JSP және серверлер, бөлінген қосымшалар сервері мен веб-сервермен таратылған жүйені құруға көбірек мүмкіндік береді;

CORBA технологиясымен үйлесімділік.

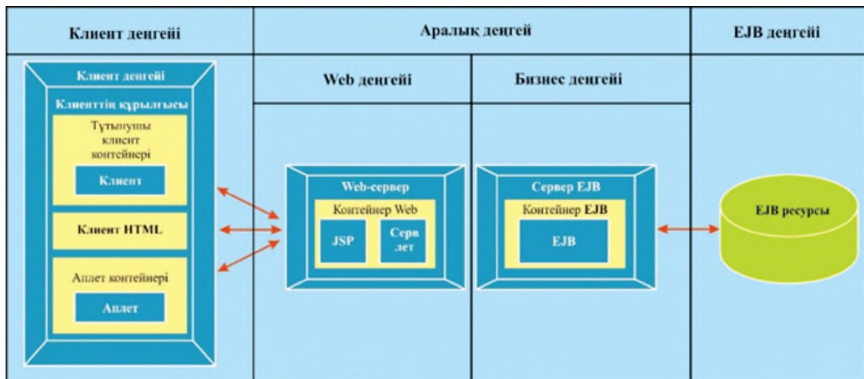
J2EE технологиясының кемшіліктері:

- Төмен өнімділік;

- Тек бір Java тілін қолдау.

Жоғарыда аталған артықшылықтар мен кемшіліктер таратылған ақпараттық жүйелерді құру үшін J2EE технологиясы өзінің EJB компоненттік моделімен өте қолайлы екенін көрсетеді.

J2EE-бұл клиенттік деңгейден, орта деңгейден және EIS деңгейінен немесе базалық деңгейден тұратын таратылған қосымшалардың көп деңгейлі архитектурасы деп айтуға болады. J2EE платформасының көп деңгейлі архитектурасы, сонымен қатар J2EE компоненттерін қолдайтын әртүрлі J2EE контейнерлері көрсетілген. (4-сурет) [6,7].



4 сурет - J2EE көп деңгейлі архитектурасы

**NET платформасы.** NET платформасы әзірлеу құралдарын, қоршаған ортаны, серверлік инфрақұрылымды және қосымшаларды құруды қамтамасыз ететін ақылды бағдарламалық жасақтаманы ұсынады. Ол HTTP, XML және SOAP сияқты жалпы қабылданған стандарттарды қолдану арқылы әртүрлі қосымшалар мен құрылғыларды біріктіруге ықпал етеді. NET бағдарламалық жасақтама индустриясының басты мәселелерінің бірін шешеді, әртүрлі тілдерде және әртүрлі ортада жазылған қосымшалар арасында мәліметтер алмасуды қамтамасыз етеді.

NET келесі құралдар мен операциялық жүйелерді қолдайды:

- Smart Client бағдарламалық жасақтамасы клиентке, компьютерге немесе мобильді құрылғыға веб-сервистер арқылы кез-келген жерден және кез-келген құрылғы арқылы деректерге қол жеткізуге мүмкіндік береді.
- NET Server инфрақұрылымы NET қосымшаларын орналастыру үшін жоғары қауіпсіз және масштабталатын платформаны ұсынады.
- XML веб-қызметтері NET технологиясымен қоршалған қосымшаларды біріктірудің негізі болып табылады. Олар интернет пен инт-



ранетке әртүрлі тілдерде жазылған және әртүрлі платформаларда жұмыс істейтін қосымшаларға HTTP, XML және SOAP сияқты стандартты протоколдар арқылы мәліметтер алмасуға мүмкіндік береді;

- Visual Studio. Net және. NET Framework – XML веб-қызметтерін құруға, хостинг пен қолдануға арналған толық шешім. Visual Studio. Net қоршаған орта мен бағдарламалау тілдерінің алуан түрлілігін қолдайды, барлық қажетті құралдарға қол жеткізуге мүмкіндік береді. Мұның бәрі Visual Studio. Net-ті қазіргі уақытта қол жетімді ең өнімді әзірлеуші құралдарының бірі етеді.

ASP.NET – веб-қосымшалар мен веб-сервистерді құруға арналған технология. Бұл Microsoft. NET платформасының ажырамас бөлігі және ескі Microsoft ASP технологиясының дамуы. Microsoft толығымен ASP.NET бағдарламаның орындалу ортасына негізделіп қайта құрылды (CLR ортасы). Ол барлық Microsoft қосымшаларының негізі болып табылады. Әзірлеушілер кодты ASP.NET жиынтығына кіретін кез-келген бағдарламалау тілдерін қолдана отырып жаза алады [8].

Кіретін кез-келген бағдарламалау тілдерін қолдана отырып жаза алады.

**NET және Enterprise Edition (J2EE™) платформаларына салыстырмалы талдау.** NET және Enterprise Edition (J2EE™) технологиялары бір-біріне өте бәсекелес болып табылады. Олар ерекше айырмашылыққа ие бола отырып, әрқайсысы веб-қызметтерді құруға мүмкіндік береді. Айырмашылықтарын атап айтсақ, көп платформалық және көптілді қолдауды жүзеге асыру технологиялары болып табылады. Көп платформалылық – NET және J2EE екеуі де веб-қызметтерді құруда маңызды рөл атқаратыны белгілі. Осы уақытқа дейін J2EE көптеген платформаларға қолдау көрсетті, бірақ Microsoft корпорациясының пікірінше, бұл J2EE құзыреті емес. Microsoft.NET-ті екі сатылы компиляция платформасы ретінде орналастырады, бұл Java сияқты кез-келген платформа үшін жұмыс ортасын құруға мүмкіндік береді. Танымал операциялық жүйелер үшін CLR жасау бірнеше жылға созылуы мүмкін. Сондықтан, бүгінгі таңда J2EE-бұл көп платформаны қолдайтын жалғыз даму ортасы деп айтуға болады. Көптілді қолдау – J2EE-дің жалғыз тілдік негізі – Java, ол NET-тен мүлдем өзгеше. Онда Fortran, COBOL, C++ және Visual Basic сияқты оннан астам тілге қолдау көрсетіледі. Бұл NET Java-дан басқа тілдерді қолданатын әзірлеушілер үшін жақсы жағдай жасайды дегенді білдіреді. NET қарапайым тілде веб-қызметтерді құруға мүмкіндік береді. Оны қайта оқытуға аз шығындар жұмсалады [9].

**Веб-қызметтің функционалдығын жүзеге асыратын қосымшаны құру.** Қосымшаны жобалау кезінде тиісті әдістер қолданылуы қажет. Объектіге бағытталған қосымшаларды құру кезінде кем

дегенде кейбір Uml диаграммалар жасалады. Microsoft Visual Studio құралы веб-қызметтерді ASP.net Web Service жобасы аясында және Visual Basic. Net-пен бірге екі объектіге бағытталған бағдарламалау тілдерінде жүзеге асыруға мүмкіндік береді. Жобаны жасағаннан кейін оған веб-қызметтің қажетті функционалдығын жүзеге асыратын кодты қосу керек.

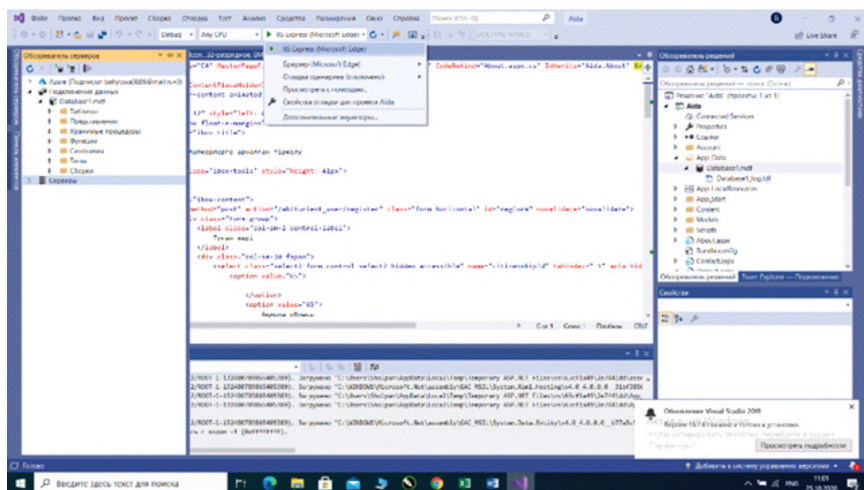
Әрбір жасалған Visual Studio веб-қызметі үшін. NET әдет бойынша келесі файлдарды жасайды;

- AssemblyInfo.cs-жобаны құру туралы жалпы ақпаратты сақтайды. Құрастыру-бұл жалпы тілдік орындау ортасының қатысуымен бірнеше рет қолданылатын кодтың функционалды бірлігі;

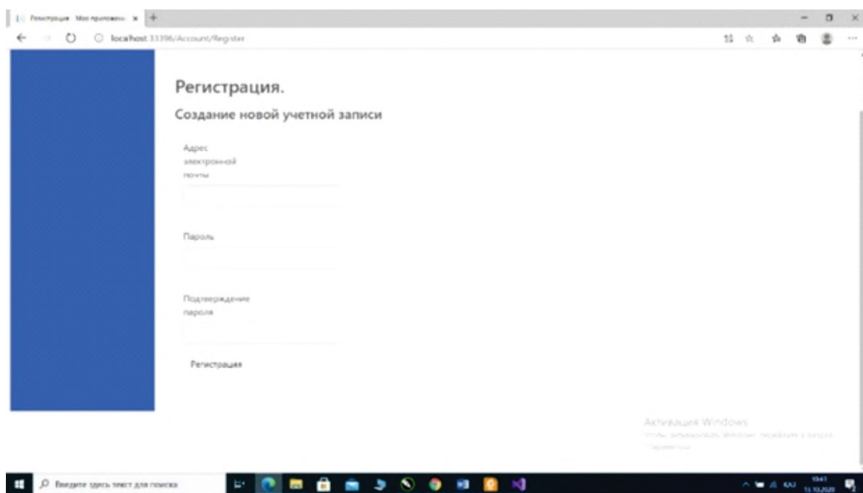
- Service1.asmx және Service1.asmx.cs-веб-қызмет интерфейсін құрайды. Service.asmx @WebService директивасынан басталады. Бұл директивада веб-қызметті құрайтын сыныпты анықтайтын class тәлсіпаты болуы керек. Service1 класс файлы.asmx.cs-Web Service-ке тәуелді жасырын файл.asmx; веб-қызметтің бөлінген код класын қамтиды;

- Web.config-веб-жобаның конфигурациясы туралы ақпаратты қамтиды. Мысалы, күйін келтіру режимінің сипаттамасы және аутентификация әдісі, сондай-ақ осы жоба үшін стандартты емес қате туралы хабарламаларды анықтайды. Web.config веб-қызметтердің конфигурациясы туралы ақпаратты сақтай алады.

Құрылған веб-қосымшаны іске қосуды жүзеге асыра отырып (5-сурет), осындай нәтижеге қол жеткіземіз.(6-сурет)



5 сурет - Веб-қосымшаны іске қосу



6 сурет - Веб-қосымшаны құру барысында тіркеу бөлімін жүзеге асыру

Осындай бағытта тіркеу бөлімін Microsoft Visual Studio құралы веб-қызметтерді ASP.net Web Service жобасы аясында жасап, нәтежеге қол жеткізуге болады. Бұл көрсетілген веб-қосымша .net платформасының тиімділігі мен ынғайлылығын айқындай түседі [10-12.]

**Қорытынды.** Қазіргі уақытта семантикалық веб-қызметтерге көп көңіл бөлінеді. Қызмет WSDL-де анықталған операцияларды онтология түрінде берілген ғаламдық ақпараттық ресурстармен байланыстыратын семантикалық сипаттаманың WSDL-ге үстірт құрылымы арқылы семантикалық болады. Семантикалық веб-қызметтер қызметтердің құрамын айтарлықтай жеңілдетеді және қызметке тек мекен-жайы бойынша ғана емес, мазмұны бойынша да қоңырау шалу мәселесін көрсетеді. Қазіргі уақытта веб-қызметтермен жұмыс істеудің статикалық парадигмасы жиі қолданылады - қосымша веб-қызметке қосымшаны әзірлеу кезінде таңдалған және жауапты пассивті күтуге мәжбүр болған нақты мекен-жай (URL) бойынша жүгінеді. Алайда, бірдей функционалдығы бар веб-қызметтер көп болуы мүмкін, өйткені оларды дамыту, қолдау және орналастыру, сайып келгенде, пайда табу арқылы анықталады. Сонымен, егер белгілі бір серверде орналасқан белгілі бір веб-сервистің қызмет көрсету сипаттамалары Клиентті қанағаттандырмаса, онда таңдалған сервердегі басқа қызметке немесе сол қызметке, бірақ басқа серверде орналасқан қызметке жүгінген жөн. Код деңгейінде бұл қосымшаның міндеті шешілетін болады. Егер құрылатын болса веб-сервисі ғана емес, нақ-

ты URL-ға жүгінуге мүмкіндік береді. Веб-қызметтермен жұмыс істеудің динамикалық парадигмасында бұл функцияларды кәсіпорынның сервистік шинасы орындайды. Ол веб-қызметтің қоңырауларына мониторинг жүргізеді және мониторинг ақпаратына сүйене отырып, нақты URL мекен-жайы бойынша байланыс жасайды.

Зерттеуде “қарапайым” және композиттік қызметтерді дамыту процестерін талдау және веб-қызметтермен жұмыс істеудің динамикалық парадигмасын модельдеу міндеттері қойылған және жұмыс істеу бағыты мен баптаулары туралы нақты ақпарат алынды. Microsoft Java - ны бағдарламалау тілдерінің бірі ретінде қарастырады. Java – бағдарламалау тілі емес, платформа деген пікір қалыптасады. Талданған платформалар веб-қызметтерді құру құралдарының ең толық тізімін ұсынады. J2EE платформасы (Java 2 Enterprise Edition) кез-келген сәулет пен операциялық жүйелер үшін Java қосымшаларына (Java сыныбы, EJB компоненті) негізделген веб-қызметтерді құруға мүмкіндік береді. Microsoft .NET платформасы бірнеше Visual Basic, C++, C# бағдарламалау тілдері арқылы веб-қызметтерді құруға мүмкіндік береді. Мақалада “Тіркеу бөлімі” деп аталатын веб-қосымшаның даму әдісі қарастырылған. Ол .NET платформасынан толық қолдау табады және қолданыстағы бағдарламалық қосымшалардан веб-қызметтерді құруға ыңғайлы.

## Әдебиеттер

1 Ключкин В. От традиционных способов интеграции – к SOA для небольших проектов: [сервисноориентированная архитектура // Банковские технологии, 2008. - №9. - с. 28-30/ [Klyuchkin.V. Ot traditsionnykh sposobov integratsii – k SOA dlya nebol'shikh proektov: [servisnoorientirovannaya arkhitektura] / V. Klyuchkin // Bankovskie tekhnologii, 2008. - №9. - s. 28-30]

2 Савельев А.О., Алексеев А.А. HTML5. Основы клиентской разработки. 2-е изд., испр. – М.: Национальный Открытый Университет «ИНТУИТ». 2016. - с. 272, [Savel'ev A.O., Alekseev A.A. HTML5. Osnovy klientskoj razrabotki. 2-e izd., ispr. – М.: Natsional'nyj Otkrytyj Universitet «INTUIT». 2016. - s. 272,]

3 Климов В.В., Чернышов А.А., Баландина А.И., Косткина А.Д. Лабораторный практикум по семантическим технологиям и вебсервисам / — М.: НИЯУ МИФИ. 2016, [Klimov V.V., CHernyshov A.A., Balandina A.I., Kostkina A.D.. Laboratornyj praktikum po semanticheskim tekhnologiyam i vebservisam / — М.: NIYAU MIFI. 2016,]

4 Типы HTTP-запросов и философия REST. URL: <https://habrahabr.ru/post/50147/> (дата обращения: 17.06.2017), [Типы HTTP-zaprosov i filosofiya REST. URL: <https://habrahabr.ru/post/50147/> (data obrashheniya: 17.06.2017)]

5 Gribova V., Kleschев A., Krylov D., Moskalenko P., Timchenko V., Shalfeeva E. A. Cloud computing platform for lifecycle support of intelligent multi-agent internet-services. Proc. Intern. Conf. PEEEE. Hong Kong. 2015. - vol.20. - s. 231-235.

6 Yurin A.Yu., Grishchenko M.A. Knowledge base editor for CLIPS. *Programmnye produkty i sistemy* [Software & Systems]. 2012. –no.4. - s. 8-87 (in Russ).

7 The Semantic Web / T. Berners-Lee, J. Hendler, O. Lassila // *Scientific American*, May. 2001. – s . 56-89.

8 Вишняков, В.А. Развитие интеллектуального управления с использованием облачных технологий / В. А. Вишняков // *Информатика*. 2016. - №2. - с.113-120., [Vishnyakov, V.A. Razvitie intellektual'nogo upravleniya s ispol'zovaniem oblachnykh tekhnologij / V. A. Vishnyakov // *Informatika*. 2016. - №2. - s.113-120]

9 Исакова А.И. Информационный менеджмент : учебное пособие – Томск : ФДО, ТУСУР. 2016 . - с 305., [Isakova A.I. Informatsionnyj menedzhment : uchebnoe posobie – Tomsk : FDO, TUSUR. 2016 . - s 305.]

10 Мак-Дональд М. WPF: Windows Presentation Foundation в .NET 4.5 с примерами на C# 5.0 для профессионалов. М.: Вильямс. 2013., [Мак-Дональд М. WPF: Windows Presentation Foundation v .NET 4.5 s primerami na C# 5.0 dlya professionalov. M.: Vil'yams. 2013.]

11 Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5. М.: Вильямс. 2015. – с 1311., [Troelsen E.H. YAzyk programmirovaniya C# 5.0 i platforma .NET 4.5. M.: Vil'yams. 2015. – s 1311.]

12 Рихтер Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C#. СПб: Питер. 2015. – с 896 ., [Rikhter D. CLR via C#. Programmirovaniye na platforme Microsoft.NET Framework 4.5 na yazyke C#. SPb: Piter. 2015. – s 896 ]

**Бакытова Ш.Е.**, ORCID ID <https://orcid.org/0000-0002-0815-5926>,  
e-mail: bahytova0806@mail.ru

**Камалова Г.А.**, ORCID ID <https://orcid.org/0000-0002-5252-4573>.  
e-mail: gokhakam@gmail.com