

Програм привредног развоја Војводине
Назив програма **еВојводина**

Пројекат
СОФТВЕРСКА АРХИТЕКТУРА ЗА Е-ВОЈВОДИНУ
ЗА ПРОГРАМ Е-ВОЈВОДИНА

Уговор бр. 01-1688/1 од 12. децембра 2005. године

Наручилац: ИЗВРШНО ВЕЋЕ АУТОНОМНЕ ПОКРАЈИНЕ ВОЈВОДИНЕ,
ПОКРАЈИНСКИ СЕКРЕТАРИЈАТ ЗА НАУКУ И ТЕХНОЛОШКИ РАЗВОЈ,
са седиштем у Новом Саду, на адреси Булевар Михајла Пупина 16.

Носилац: ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА, са седиштем у Новом Саду,
на адреси Трг Доситеја Обрадовића 6

Руководилац: Мр Мирослав Зарић, дипл. инж. ел.

Нови Сад, јун 2006. године

Програм привредног развоја Војводине
Назив програма **еВојводина**

Пројекат
СОФТВЕРСКА АРХИТЕКТУРА ЗА Е-ВОЈВОДИНУ
ЗА ПРОГРАМ Е-ВОЈВОДИНА

Уговор бр. 01-1688/1 од 12. децембра 2005. године

Пројектантски тим:
Мр Мирослав Зарић, дипл. инж. ел.
Др Милан Видаковић, дипл. инж. ел.
Др Бранко Милосављевић, дипл. инж. ел.
Мр Ђорђе Обрадовић, дипл. инж. ел.
Горан Сладић, дипл. инж. ел.
Срђан Комазец, дипл. инж. ел.

Нови Сад, јун 2006. године

Програм привредног развоја Војводине
Назив програма **еВојводина**

Менаџер програма еВојводина
Проф. Др Душан Сурла
Заменик менаџера програма еВојводина
Проф. Др Зора Коњовић

Назив пројекта
**СОФТВЕРСКА АРХИТЕКТУРА ЗА Е-ВОЈВОДИНУ
ЗА ПРОГРАМ Е-ВОЈВОДИНА**

Савет пројекта
Др Раде Ћирић, Помочник покрајинског секретара за науку
Др Душан Сурла, ред. проф. Природно-математички факултет, Нови Сад
Др Саша Бошњак, ванр. проф. Економски факултет, Суботица
Др Бранко Милосављевић, доцент, Факултет техничких наука, Нови Сад
Мр Мирослав Зарић, асистент, Факултет техничких наука, Нови Сад
Мр Милан Парошки, Служба за опште и заједничке послове покрајинских органа

Аутори пројекта
Мр Мирослав Зарић, асистент, Факултет техничких наука, Нови Сад
Др Милан Видаковић, доцент, Факултет техничких наука, Нови Сад
Др Бранко Милосављевић, доцент, Факултет техничких наука, Нови Сад
Мр Ђорђе Обрадовић, асистент, Факултет техничких наука, Нови Сад
Горан Сладић, дипл. инг. асистент, Факултет техничких наука, Нови Сад
Срђан Комазец дипл. инг. асистент, Факултет техничких наука, Нови Сад

РЕДЕФИНИСАЊЕ ПРЕДМЕТА УГОВОРА

о изради пројекта *Софтверска архитектура за Е-Војводину* за програм *Е-војводина*, који је заведен на Факултету техничких наука у Новом Саду под бројем 01-1688/1 од 12.12.2005. године

На четвртој седници савета пројеката, која је одржана 20. јуна 2006. године, закључено је следеће:

1. да се део предмета овог уговора који се односи на пример модела: *Кориснички јавни сервис Фонда за развој Војводине* премести у Уговор о изради пројекта *Спецификација информационих захтева јавних сервиса система Е-Војводина* за програм *Е-војводина*, заведен на Факултету техничких наука у Новом Саду под бројем 021-1687 од 12.12.2005. године.
2. да се у овом пројекту дода у предмет уговора део који се односи на избор софтверских компоненти за Е-војводину.

У складу са овим закључком, редефинише се члан 1 уговора тако да гласи:

Предмет уговора је израда пројекта **Софтверска архитектура за Е-Војводину** који обухвата:

1. Полазне основе за израду софтверске архитектуре.
2. Модел софтверске архитектуре за Е-Војводину.
3. Препоруке за избор софтверских компоненти за Е-Војводину.

Предговор

У Пројекту је разматрана неопходна софтверска архитектура за увођење система електронске управе на територији АП Војводине.

Пројектна документација садржи следећа поглавља:

1. Полазне основе за израду пројекта
2. Елементи софтверске архитектуре – решења у свету
3. Препоруке за избор софтверских компоненти за еВојводину

У **првом** поглављу дат је приказ основне архитектуре софтверских система за подршку еУправи, кратак преглед постојећег стања инфраструктуре у ИВ и Скупштини АПВ (покрајинским органима управе), републичким органима и локалним самоуправама, као и предлог поступка анализе захтева.

У **другом** поглављу извршена је детаљнија анализа појединих софтверских компоненти, као што су оперативни системи и апликације на радним станицама, серверски оперативни системи, системи за управљање базама података, као и развојних платформи, апликационих сервера и CMS система. Као посебан одељак дат је преглед истраживања Европске уније по питању коришћења OpenSource софтвера.

Треће поглавље садржи завршне анализе на основу претходно изнесених података, као и предлог за избор одређених врста софтвера у свакој од посматраних група.

У прилогу пројекта дата су два илустративна примера имплементације софтверске архитектуре сервиса еУправе.

Као прилог пројектној документацији приложен је CD који садржи овај пројекат и документа наведена у литератури која се могу слободно преузети са Интернета.

Захваљујемо се члановима Савета пројекта на активном учешћу у формирању коначног текста пројектне документације. Посебно се захваљујемо проф. др Зори Коњовић, проф. др Душану Сурли, мр Милану Парошком и господину Ненаду Маурићу који су својим примедбама и сугестијама допринели да коначан текст буде јаснији и обухватнији.

Аутори

Садржај

1.	Полазне основе за израду пројекта -----	1
1.1	Софтверска архитектура за еВојводину -----	1
	Софтверске компоненте -----	2
1.2.	Преглед постојећег стања у Војводини -----	6
1.3.	Предлог поступка анализе софтверских захтева -----	8
2.	Елементи софтверске архитектуре – решења у свету -----	11
2.1.	Десктоп оперативни системи и апликације -----	11
2.2.	Серверски оперативни системи -----	16
2.3.	Системи за управљање базама података – RDBMS системи -----	23
	Основне особине RDBMS система -----	24
	Додатни типови табела и погледи (<i>views</i>) -----	26
	Индексирање -----	27
	Остали објекти -----	28
	Партиционисање -----	29
	Подршка за рад са XML документима -----	30
2.4.	Апликациони сервери и CMS системи -----	40
2.4.1.	Апликациони сервери -----	40
2.4.2.	.NET и J2EE платформе – основни концепти -----	42
	Поређење .NET и J2EE технологија -----	43
	Кључне разлике -----	44
	Какав је утицај .NET приступа? -----	45
2.4.3.	CMS -----	52
	WCMS -----	52
	Употреба CMS система у пројектима еУправе -----	54
	CMS системи у <i>eGovernment</i> пројектима -----	55
	Подела CMS система -----	55
	Анализа CMS система -----	58
	Apache Lenya -----	58
	Magnolia -----	58
	Open CMS -----	59
2.5.	Смернице и искуства у ЕУ -----	61
3.	Препоруке за избор софтверских компоненти за еВојводину -----	71
3.1.	Оперативни системи и десктоп апликације на радним станицама -----	71
3.2.	Серверски оперативни систем -----	73
3.3.	Базе података -----	76
3.4.	Апликациони сервери и CMS системи -----	80
	Литература -----	83
	Прилог -----	

1. Полазне основе за израду пројекта

На крају овог текста наведена је коришћена литература за израду овог Пројекта. У тој литератури разматране су софтверске архитектуре за електронску комуникацију између *грађана и влада* и између *компанија и влада*. У овој области уведени су појмови као што су *електронска управа*, *електронска влада*, *дигитално друштво* или скраћено *еВлада*, *еУправа* (*eGovernment*, *e-gov*) [*wikipedia-eGov*]. Ти појмови описују електронски начин пословања или комуникације између грађана и владе *G2C* (*Government to Consumer*), владе и компанија *G2B* (*Government to Business*) и владе и владе *G2G* (*Government to Government*). У наставку ће се наведени појмови равноправно користити, а појам *еВојводина* описиваће систем електронске управе Аутономне Покрајине Војводине. Појам *еУправе* подразумева коришћење информатичких и комуникационих технологија с циљем повећања ефикасности законодавне, извршне и судске власти, као и за побољшање и већу транспарентност у комуникацији са грађанима. У овом светлу појам *еВојводина* представља интегрисани систем електронске управе на покрајинском нивоу и нивоу локалних самоуправа на територији АП Војводине. Овај систем обухвата како техничка средства (хардверске компоненте, комуникациону опрему, софтверске системе), тако и организацију одређених служби. Крајњи циљ овог система јесте успостављање скупа електронских сервиса који ће омогућити грађанима и пословним корисницима електронску комуникацију и обављање одређених послова из надлежности управе електронским путем. Основу оваквог система електронске управе чини дистрибуирани, робустан и проширив информациони систем кога чине интероперабилне апликације имплементирани у одговарајућим покрајинским органима (Скупштини АП Војводине, ИВ АП Војводине), као и органима локалне самоуправе и државне управе на целокупној територији АП Војводина.

1.1. Софтверска архитектура за еВојводину

При анализи софтверске архитектуре информационог система за потребе еВојводине, извршено је груписање на:

- **Софтвер на десктоп рачунарима** (десктоп апликације - користе их запослени у јавној администрацији - служе за припрему интерних докумената, за припрему докумената који ће бити публиковани путем јавних сервиса, помоћу њих се обрађују примљени документи од клијената јавних сервиса и сл). У ову групу се могу убројати и разне клијентске апликације (*web*, *ftp*, *email*), као и сам оперативни систем на радним станицама.
- **Серверски оперативни системи** – инфраструктурни софтвер који омогућава коректно функционисање целокупне инфраструктуре која је саставни део информационог система.
- **Базе података** – релационе базе података које служе за трајно смештање и архивирање свих докумената (и метаподатака којима су описани) који се јављају у процесу рада сервиса еВојводине.
- **Web апликације** – апликације које се налазе у међуслоју између база података и клијентских апликација. Web апликације омогућавају приступ јавним сервисима дефинисаним у оквиру еВојводине. Истовремено *web*

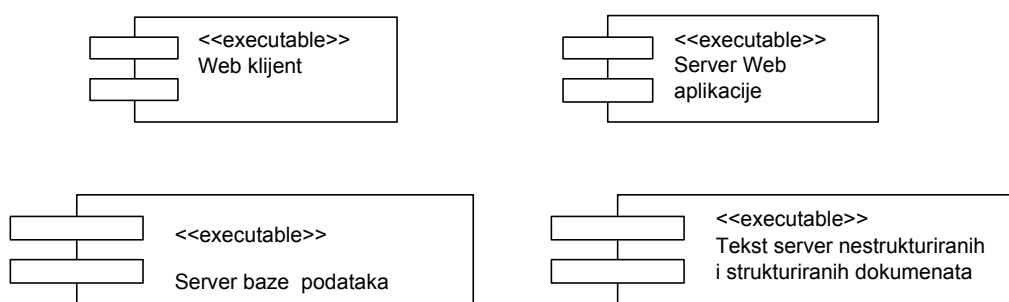
апликацијама се приступа и од стране запослених у јавној администрацији, који их користе као средство за припрему садржаја јавних сервиса еВојводине (припрема и публиковање докумената који су доступни путем ових јавних сервиса).

У овом одељку дата је глобална архитектура заснована на трослојном моделу организације информационих система. Први слој је предвиђен за интеракцију са корисником. Намена другог (средњег) слоја је извршавање пословне логике самог система, а трећег слоја физичко руковање подацима.

Према трослојној архитектури, уочене су софтверске компоненте које представљају извршне програме и везе зависности између њих. Са клијентске стране се налази као извршна компонента *Web browser*, у другом слоју се налази *Сервер Web апликација*, а у трећем слоју су представљени сервери база структурираних и неструктурираних докумената.

Софтверске компоненте

На слици 1.1 приказане су софтверске компоненте система.



Слика 1.1. Софтверске компоненте

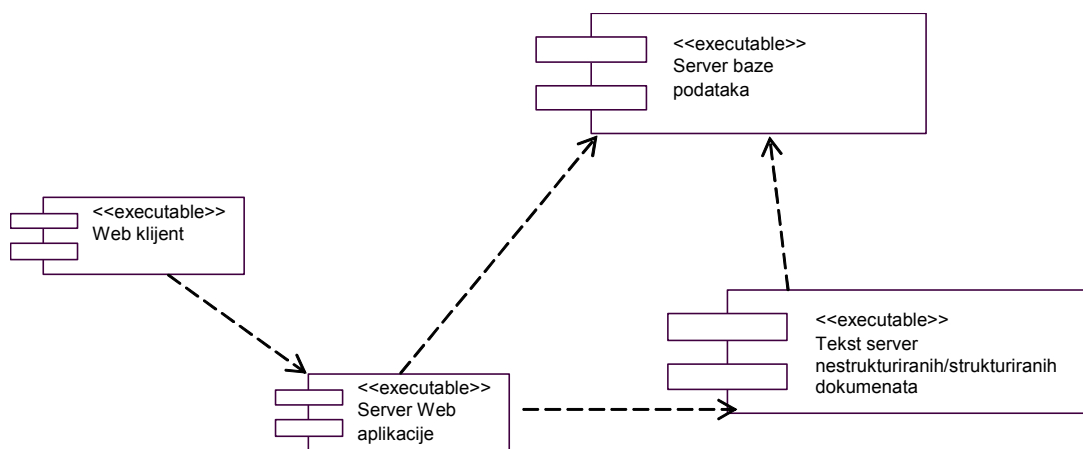
Web клијент је компонента која се извршава на клијентској страни. Кориснички интерфејс апликације се са клијентске стране позива са сервера Web апликације, најчешће у облику сервлета. Осим Web browser-а као клијентска апликација може се у одређеним случајевима користити и посебно развијена самостална апликација (у случају да је потребно обезбедити неку сложенију функционалност на клијентској страни). Када се као начин комуникације са сервером користе Web сервиси као клијентска страна се јавља програмски модул који је креиран на основу спецификације сервиса и може бити део апликације на десктоп или серверском систему.

Сервер Web апликације садржи имплементацију пословних процеса посматраног информационог система, тј. пословну логику система. Под овим серверима би радиле и апликације са којима се интеракција обавља посредством web сервиса.

Сервер базе података. Сервер базе података служи за трајно чување свих података који настају у процесу рада система еВојводине. У суштини ова компонента представља релациону базу података. Кључни садржај ове базе јесу сами документи (структурирани и неструктурирани) који се користе у систему еВојводине, као и метаподаци којима се овакви документи описују, а који омогућавају лакше проналажење докумената као и праћење њиховог животног циклуса.

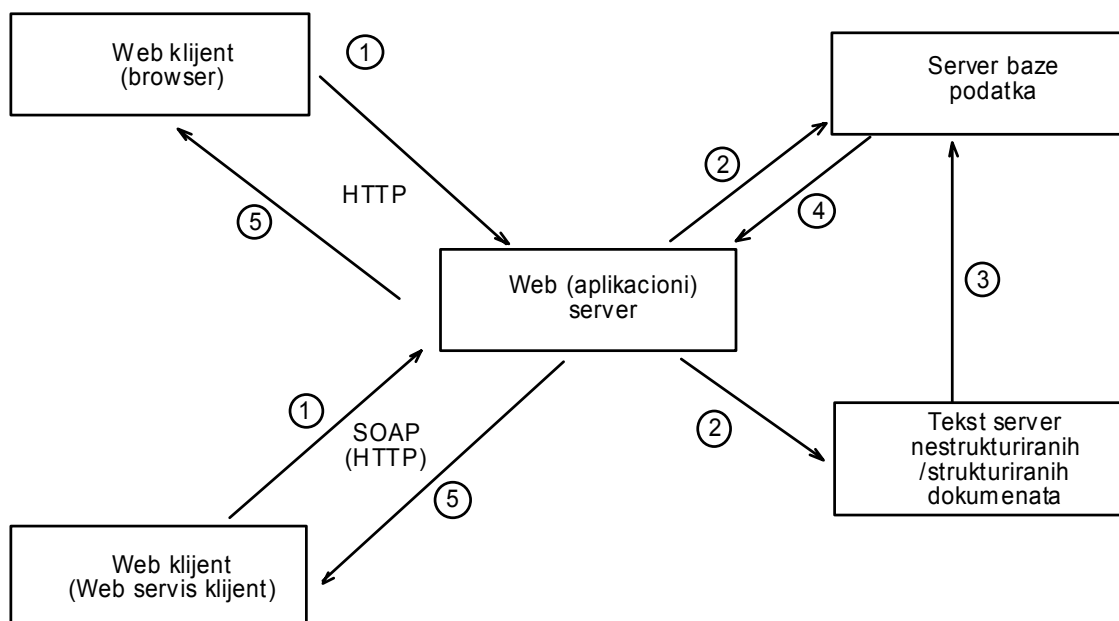
Текст сервер неструктурираних и структурираних докумената је софтверска компонента која обезбеђује индексирање и претраживање докумената из базе неструктурираних/структурираних докумената. Скоро сви познати произвођачи система за управљање базама података уз базу података нуде и текст сервер. На пример, код Oracle-а је то *ConText Server*, а код Microsoft-а је *Microsoft Index Server*. Постоји и одређен број *open source* пројеката који нуде софтверска решења која омогућавају претраживање текстуалних докумената различитог формата (нпр. Apache Lucene). Више информација о текст серверима и њиховој употреби у системима за руковање документима може се наћи у пројекту¹ који представља интегрални део пројекта система еВојводине.

На слици 1.2 приказан је дијаграм наведених софтверских компоненти. *Сервер базе неструктурираних докумената* и *Текст сервер неструктурираних/структурираних докумената* су софтверске компоненте које обезбеђују складиштење, индексирање и претраживање неструктурираних докумената. Структурирани документи се могу складиштити у бази структурираних докумената која подржава и њихово индексирање и претраживање. Могуће је и за структуриране документе користити посебан текст сервер за њихово индексирање и претраживање. Везе између софтверских компоненти представљају везу зависности (испрекидана усмерена линија). На пример, веза од компоненте *Текст сервер неструктурираних докумената* до компоненте *Сервер базе неструктурираних докумената* има значење да је за извршавање прве компоненте неопходно постојање друге компоненте.



Слика 1.2. Дијаграм софтверских компоненти

¹ Пројекат Спецификација информационих захтева интранет система ИВ АПВ, Нови Сад, 2006.



Слика 1.3. Упрости́ена шема комуникације између компоненти

На слици 1.3 дата је упрости́ена шема процеса комуникације појединих компоненти. Без обзира да ли се ради о захтеву за преузимање одређених докумената, попуњавању одређених образаца или неком сличном поступку, клијентска апликација иницира приступ одређеним садржајима (1). На основу захтева клијентске стране web сервер апликације иницира претраживање (или смештање података) у базу података смештену на одговарајућем серверу (2). У цео овај процес најчешће је укључен и текст сервер који има улогу да индексира све структуриране и неструктуриране документе (2,3). Приликом захтева за претраживање базе, текст сервер се користи за лоцирање тражених докумената, док се приликом додавања нових садржаја користи за индексирање. на основу упућених захтева база података враћа одговарајући резултат web серверу апликације (4), који се преко одговарајућег транспортног протокола враћа клијенту (5).

Системски софтвер

Системски софтвер треба одабрати тако да се у потпуности подржи предложена трослојна архитектура обраде података и интранет/Интернет модел дистрибуираног информационог система. У складу са постављеним принципом, потребно је за серверске и клијентске станице одабрати један или више оперативних система који, као битну карактеристику, поседују одговарајући комуникациони систем са потпуном и квалитетном подршком TCP/IP протоколу који је основни комуникациони протокол рачунарске мреже.

Такође је потребно одабрати систем за управљање релационом базом података који подржава интранет/Интернет дистрибуирану базу, функције управљања базом структурираних и неструктурираних докумената, функције управљања текстуалним базама, а у делу техничког система и функције управљања мултимедијалним подацима (статичке слике, видео записи). У складу са предложеним концептом информационог система, потребно је одредити се и за одговарајући основни програмски језик чија основна карактеристика треба да буде квалитетна подршка објектно-оријентисаном приступу и мрежном раду.

Коначно, за развој софтвера потребно је одабрати одговарајуће развојно окружење којим се подржава основна оријентација информационог система и постиже висок степен продуктивности.

Интероперабилност

Наведене софтверске компоненте могу бити реализоване различитим технологијама, али кључна особина коју морају да обезбеде јесте интероперабилност.

Интероперабилност представља способност информационих и комуникационих система, и пословних процеса да подрже пренос података и омогуће размену информација и знања. Интероперабилност се осигурава како на техничком нивоу (стандарди за повезивање система и сервиса), семантичком нивоу (усаглашено значење података), и на нивоу пословних процеса. Постизање интероперабилности различитих система обезбеђује се поштовањем како националних тако и међународних стандарда и норми. За подручје еУправе од критичне је важности поштовање стандарда:

- за израду Интернет страница и успостављање web сервиса (HTML, XML, XHTML, SOAP, WSDL - ове стандарде дефинише World Wide Web Consortium) [W3C],
- Интернет техничких стандарда које одржава Радна група за Интернет инжењеринг (Internet Engineering task Force) [IETF]. Ова група представља веома бројну, отворену групу стручњака из области рачунарских мрежа, оператера, произвођача опреме и редовно одржава већину техничких норми која омогућава несметано функционисање и развој Интернета (већина техничких спецификација се може наћи у форми RFC – *Request for Comment* – докумената),
- стандарда за електронско пословање које израђује, одржава и промовише Организација за унапређење стандарда за структуриране информације (Organization for Advancement of Structured Information Standards) [OASIS]. Ова организација се бави развојем стандарда везаних за размену различитих врста структурираних информација. Међу најзначајнијим стандардима које је развила ова организација су ebXML, UDDI, OpenDocument, UBL.

Како би обезбедила интероперабилност система еУправе, Европска унија је покренула програм IDABC (Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens) [IDABC]. Као резултат рада на овом програму усвојен је Европски оквир за интероперабилност (European Interoperability Framework) [EIF], усвојене су Смернице за архитектуру система (Architecture Guidelines) који омогућавају интероперабилност, Стратегија за интероперабилност садржаја (Content Interoperability Strategy) која има за циљ да обезбеди јединствену семантику. Као један од резултата рада овог програма јесте и прихватање и промоција OpenDocument формата са циљем обезбеђења формата електронских докумената који ће омогућити њихову размену између јавних служби, компанија и појединаца, а да се при томе не фаворизује нити захтева употреба тачно одређеног софтвера.

Један од битних услова које сервиси јавних служби тј. еУправе морају да задовоље јесте доступност информација свим заинтересованим странама под једнаким условима. Како би се ово постигло, неопходно је да Интернет странице и документи који се на њима нуде поштују отворене стандарде, тј. да нису посебно прилагођене било којој појединачној апликацији за читање садржаја, која није јавно и бесплатно доступна. Такође добро реализовани сервиси еУправе морали би водити рачуна и о

доступности садржаја за особе са смењеним способностима. Ово подручје је обрађено у Смерницама за доступност веб садржаја (Web Content Accessibility Guidelines 1.0)

Софтверски системи развијени на основу наведених стандарда, без обзира у којој технологији рађени, гарантоваће интероперабилност са другим системима који поштују исте стандарде. Ово омогућава да се софтверски системи еУправе на нивоу покрајинске управе и нивоу локалних самоуправа разликују по технологији и начину имплементације, али да успешно комуницирају и обезбеђују кохерентан систем еУправе на целој територији АП Војводине.

Иако не постоји препрека хетерогености система, у оквиру једне институције пожељно је да се, због лакшег управљања и одржавања и смањења трошкова, софтверска и хардверска архитектура униформише. На различитим нивоима, могуће је, и врло вероватно, постојање хетерогених система, у зависности од локалних фактора (распожива средства за увођење информационих система, могућност/немогућност ангажовања стручних лица за одређене врсте софтверских система и сл). Све док је услов поштовања наведених стандарда испуњен, интероперабилност оваквих система није доведена у питање.

Најважнији стандард, који мора бити усвојен као темељни, а који се налази у основи већине модерних стандарда за резмену структурираних електронских докумената, за све врсте комуникације између интероперабилних система, је XML. XML је у основи и овде набројаних стандарда: XHTML, SOAP, WSDL, UDDI, ebXML, OpenDocument, UBL.

Поштовањем ових препорука гарантује се међусобна интероперабилност разнородних система, без обзира којом технологијом (.NET, J2EE, PHP или нешто друго) су дати системи реализовани. На овај начин неће бити занемарен нити умањен значај напора које су неке локалне средине уложиле у развој својих система за еУправу, него ће се дефинисати стандардни оквир за размену који ће омогућити да се сви сервиси временом стандардизују и усагласе. Улога покрајинских органа управе у том погледу је да не намећући специфичне имплементације, предводе напоре за стандардизацију, и помажу локалним срединама у постизању овог циља.

1.2. Преглед постојећег стања у Војводини

Тренутно стање софтверске инфраструктуре са становишта успостављања сервиса еУправе неопходно је посматрати на више нивоа, стање у републичким органима управе, покрајинским органима управе, јединицама локалне самоуправе, али је за пројекат еВојводина најзначајнији преглед стања у органима покрајинске управе (Извршно веће АПВ, Скупштина АПВ), као и у јединицама локалне самоуправе на територији АПВ

На републичком нивоу, у последњих неколико година приметан је труд да се одређене информације правовремено учине доступне корисницима (грађанима и пословним субјектима) путем веб страница Владе и министарстава. Приметно је да је већина ових сајтова настајала индивидуално, без заједничке визуелне и програмске основе. На сајту Владе републике Србије (<http://www.srbija.sr.gov.yu/>) на пример, не постоје линкови на веб презентације појединих министарстава. Већина активних (динамичких) презентација је реализована коришћењем скриптог језика PHP, неке су нпр. Канцеларије за придруживање Европској унији (<http://www.seio.sr.gov.yu/>), која делује као тело Владе, реализоване коришћењем ASP технологије, а сајт Пореске

управе Републике Србије (<http://www.poreskauprava.sr.gov.yu>) реализован је Java технологијом. На свим презентацијама могуће је преузети кључне документе (законе, уредбе, правилнике, обрасце, најчешће у pdf формату), док интерактивних садржаја нема.

Захваљујући редовним пописима инвентара, могуће је доћи до одређених података о техничкој опремљености органа покрајинске управе. У органима покрајинске управе тренутно постоји око 600 рачунара, и око 350 штампача. Карактеристично је да су у највећем броју случајева рачунари, ако су уопште међусобно повезани, повезани у локалне мреже само у оквиру секретаријата, док јединствена мрежна инфраструктура није присутна. Од укупног броја рачунара број серверских конфигурација, као и њихова намена нису јасно специфицирани. Приступ Интернету није јединствено уређен, већ су различити секретаријати своје потребе задовољили повезивањем на различите начине (највише *dial-up* приступом), коришћењем услуга различитих провајдера. Контрола права приступа и сигурност приступа Интернету није дефинисана. Већина секретаријата има неки вид презентација на Интернету путем којих је могуће добити одређене информације из делокруга рада секретаријата.

Од оперативних система на десктоп рачунарима користе се само различите верзије оперативног система Windows, али ни ту ситуација није унифицирана. Од канцеларијских апликација користи се Microsoft Office, различите верзије на различитим местима.

Јединствена архива докумената у електронском облику не постоји. Пословне апликације за различите намене сигурно су имплементирани (служба рачуноводства, писарница и сл), али дате апликације нису имплементирани у неким напредним технологијама. Систем за управљање документима није присутан. У раду² могу се наћи детаљније информације о стању инфраструктуре у Извршном већу АПВ. Додатне информације о стању мрежне инфраструктуре могуће је наћи у документу³.

За поуздано утврђивање тренутног стања у овој области, предлаже се формирање упитника који би омогућио прикупљање детаљних информација о постојећим хардверским и софтверским решењима на свим нивоима државне управе на територији АП Војводина. Само овакво опсежно проведено истраживање може дати праву слику о тренутном стању информатичке инфраструктуре. Као полазну тачку, с обзиром на то да су концентрисане на једном месту, ово истраживање треба провести у телима покрајинске управе, јер се ту релативно брзо могу очекивати повратне информације.

На основу тренутно доступних података јасно је да ће за потребе увођења сервиса еВојводине бити неопходно обезбедити како техничку инфраструктуру, тако и дефинисати софтверску инфраструктуру која је неопходна како би сервиси уопште заживели и гарантовали ажурност докумената који би се на овај начин учинили доступним грађанима и институцијама. Ово подразумева успостављање мрежне инфраструктуре која је неопходан основ за рад дистрибуираних апликација које један овакав сервис подразумева. Осим тога неопходно је успоставити систем управљања документима, дефинисати потребе за базама података за похрану докумената,

² Арсовски Саша, *Информационе технологије у покрајинској управи – стање и потребе*, Међународна конференција о стратегији развоја покрајинске управе, Нови Сад, март 2006.

³ "Приказ затеченог стања", *Пројекат рачунарске мреже ИВ и Скупштине АПВ*, Нови Сад, 2006.

обезбедити контролисан приступ Интернету, и контролисано и сигурно објављивање података на Интернету и сл.

На нивоу локалних самоуправа, стање информатичке инфраструктуре је различито од места до места јер није постојала јединствена стратегија развоја. Неке општине (Инђија, Суботица, Нови Сад, Кикинда...) у већој или мањој мери имају имплементирани одређене сервисе за грађане (виртуелни матичар, увид у бирачке спискове, комуникација са комуналним службама и градском управом и сл – интерактивни ниво еУправе), док већина општина има барем Интернет презентацију која се доста редовно ажурира и бар на тај начин нуди одређене информације грађанима. Неке од општина су на овај начин учиниле корак напред у односу на покрајинску управу, уводећи web технологије, које су базиране на различитим решењима и различитим технологијама, али прате трендове развоја услуга које нуде јавне службе својим корисницима.

1.3. Предлог поступка анализе софтверских захтева

Методологија спецификације даје препоруке за скуп активности које треба да се изврше у циљу дефинисања захтева система, те његове анализе, пројектовања, имплементације и тестирања. Дефинисање захтева система и његова анализа започињу анализом организационе структуре система, те података и процеса у систему.

Како тренутно не постоји ниједан сегмент система који би требао да чини скуп сервиса еВојводине, при анализи захтева, те хардверске и софтверске инфраструктуре неопходно је кренути од искустава земаља које су у већој или мањој мери имплементирале сервисе који су предвиђени за еВојводину.

Први корак при дефинисању софтверске архитектуре је анализа неопходних софтверских алата/апликација која ће омогућити ефикасну припрему докумената који ће бити нуђени путем сервиса еВојводине, као и обраду докумената који буду примљени путем одговарајућих сервиса. У овом кораку неопходно је анализирати десктоп апликације које се користе (или требају бити коришћене) од стране запослених у јавним службама. Разматрање десктоп апликација је дато у одељку 2.1.

Други корак предстаља анализа неопходних серверских система, као и софтвера на серверским системима који су неопходни за обезбеђивање јавних сервиса који ће бити дефинисани у оквиру информационог система еВојводине. Овде је неопходно разматрати проблем системског софтвера на серверским конфигурацијама, затим проблем апликационих сервера који имплементирају највећи део логике оваквих сервиса, као и проблем развојних окружења и технологије коју би било препоручљиво користити за развој сервиса еВојводине.

Као трећи корак неопходно је извршити анализу примењивих софтверских решења која би омогућила трајно чување података који се путем интранета/Интернета акумулирају у информационом систему еВојводине. У овом кораку неопходно је извршити анализу расположивих система за управљање релационим базама података.

На основу анализа изнетих у пројекту и искустава стечених у овој области у другим земљама, надлежно тело/служба ИВ Војводине треба донети одређене препоруке за избор софтвера за потребе еВојводине.

У прилогу пројекта дати су могући примери имплементације сервиса еУправе у два случаја: комуникација локалних органа самоуправе и органа покрајинске управе (пример 1), и комуникација између крајњег корисника услуге еУправе (Универзитет) и

Полазне основе за израду пројекта

надлежног покрајинског секретаријата (пример 2). На овим примерима дат је оквирни приказ функционисања одређених сервиса еУправе, као и краћи опис сценарија комуникације.

2. Елементи софтверске архитектуре – решења у свету

Анализом доступне литературе која се односи на софтверске архитектуре неопходне за имплементацију пројекта еВојводина, издвојени су јасно дефинисани сегменти које је неопходно детаљно размотрити:

- Десктоп апликације
- Оперативни системи за серверске конфигурације
- Системи за управљање базама података
- Апликациони сервери и CMS системи

Ови сегменти су детаљно разрађени у тексту овог поглавља. Као посебан одељак дата је студија Европске уније у утицају и могућностима коришћења *open source* (или *freeware*) софтвера у јавним службама државне администрације. Разматрања у овом поглављу су основа препорука датих у трећем поглављу.

2.1. Десктоп оперативни системи и апликације

У јавном сектору у већини земаља, у овом сегменту апсолутну доминацију имају Microsoft-ове апликације, укључујући ту разне верзије оперативног система Windows, као и пакет канцеларијских апликација Microsoft Office. Доминација Microsoft-ових апликација значи уједно и да су формати докумената које користе Microsoft апликације тренутно *de facto* стандардни формати докумената за размену, како унутар самих јавних служби, тако и са широм јавношћу. Најраширенији од свих формата докумената је *doc* формат докумената које користи Microsoft-ова апликација за обраду текста Microsoft Word, али су у великој мери заступљени и сви други формати који су подразумевани за остале апликације Microsoft Office пакета. У оваквој ситуацији, свако *open source* или било које друго решење које се јавља као претендент да буде прихваћен као стандардна апликација за канцеларијску употребу у оквиру јавних служби, мора задовољити један почетни услов: мора бити у стању да коректно приказује, едитује и снима документа у формату који је дефинисан Microsoft Office апликацијама. Иако данас постоје канцеларијски пакети који су у стању да манипулишу MS Office документима, пуна компатибилност и даље није остварена, јер су формати које користе Microsoft-ове апликације у одређеним деловима заштићени. На овај начин, иако постоје бесплатне (или јефтине) алтернативе MS Office пакету, оне неће бити увек у могућности у потпуности коректно процесирати документа креирана MS Office апликацијама (Microsoft је објавио формат *doc* датотека до верзије Word 97, касније верзије нису објављене – иако се сматра да већих измена у структури докумената није било јер све касније верзије омогућавају снимање докумената у овом формату без приметног губитка функционалности. Ипак, примећена је разлика у величини снимљених датотека, што имплицира да постоје одређене измене).

Из ових разлога, пожељно је да се на дуже стазе, у област десктоп апликација уведе отворени стандарди, нарочито у погледу отвореног стандарда формата докумената за обраду текста. Успостављање оваквог стандарда се тренутно разматра [OASIS].

Тренутни формати текстуалних докумената који се често појављују као формати за размену путем Интернета су:

- *doc* формат - подразумевани формат Microsoft Word програма. Формат је власништво Microsoft корпорације. 2003. године Microsoft је објавио MS

XML Reference Schemas – које специфицирају формат докумената уколико се текстуални документ сними као XML документ. Овај формат је у потпуности специфициран и бесплатан, али на жалост није подразумевани формат са којим ради MS Word. Microsoft је 22. новембра 2005. године поднео Европској асоцијацији произвођача рачунарске опреме (European Computer Manufacturer's Association) [ECMA] званични захтев за ревизију и стандардизацију XML Reference Schema-е за верзију докумената са којима ће радити MS Office 12, а иста спецификација ће бити упућена и на стандардизацију као ISO стандард. Microsoft OpenXML формат је још увек у фази разматрања у оквиру ECMA.

- *rtf (rich text format)* – Такође Microsoft-ов формат осмишљен као јединствени формат за размену текстуалних докумената који могу садржавати и графику - није широко распрострањен.
- *pdf (Adobe Portable Document Format)* – формат за репрезентацију докумената. Омогућава преносивост докумената без обзира на апликацију којом је документ креиран, оперативни систем на којем је креиран, или хардверску платформу на којој се апликација која је креирала документ извршавала. Омогућава потпуно укључивање свих графичких елемената у сам документ, тако да се гарантује истоветан изглед документа без обзира на којој клијентској платформи се отвори. Овај формат докумената је временом прихваћен као стандардни формат за размену докумената у Интернет окружењу. Постоји одређен број бесплатних алата за креирање и приказ pdf докумената, док Adobe нуди бесплатан Adobe Reader за преглед докумената и комерцијални пакет Adobe Acrobat, који омогућава и креирање и манипулацију pdf документима. Овај формат проистекао је из PostScript језика, као његово поједностављење. Карактерише га релативно мала величина докумената, уз доста висок квалитет записа графичких садржаја. Важно је напоменути да је pdf формат постао *de facto* стандард за објављивање докумената путем Интернета. Већина докумената се нуди управо у овом облику као алтернатива изворном формату документа.
- *Adobe XDP (XML Data Package)* – формат који је недавно објавио Adobe. XDP представља XML формат који "пакује" pdf документе у XML, заједно са XML формама и *template*-има.
- *OpenOffice.org/StarOffice* формат докумената је XML базиран формат који је у потпуности документован и дистрибуција овог формата је потпуно бесплатна. OpenOffice је изникла из канцеларијског пакета StarOffice компаније Sun, а послови на стандардизацији формата докумената ових пакета је започет уз помоћ OASIS-а. Циљ овакве стандардизације је управо формирање отворене, XML базирани спецификације формата докумената за канцеларијске софтверске пакете [OpenDoc]. Microsoft је од 15. марта 2006 године постао члан комитета за евалуацију ODF стандарда, али је најавио да неће учествовати у самом процесу ратификације. OpenDocument Format (ODF) је 03. маја 2006. званично прихваћен од стране ISO стандардизационог тела и сада се ODF води као **ISO/IEC 26300**. Технички део стандарда је прихваћен једногласно, и сада се обављају формалности за објављивање формата као званичног ISO стандарда. Према анализи аналитичке куће Gartner која је јавно доступна на [Gartner_ISO-ODF], њихова процена је да

постоји 70 % вероватноће да ISO сада неће одобрити MS OpenXML као ISO стандард, како би се избегло формирање вишеструких XML стандарда за исту намену. Њихова процена је да ће до 2010 године 50% служби државне управе захтевати овај формат и око 20% комерцијалних организација. Gartner анализа такође упозорава да трошкови потпуног преласка на овај формат могу бити и доста високи. Великим корисницима Microsoft апликација се сугерише да изврше притисак да овај формат буде прихваћен и у MS Office апликацијама. Европска комисија засад има уздржан став и иако подржава ODF, IDABC још није издао званичну препоруку за коришћење овог формата као преферираног. Претпоставка је да ће ЕУ снажно подржати овај формат, али ће оставити могућност да се у случају усвајања као ISO стандарда, препорука односи и на MS Open XML стандард.

- *TeX/LaTeX* – су у суштини системи за припрему докумената - нарочито популарни у академским круговима, за јавне службе нису од већег интереса. Најчешће се документи креирани на овај начин дистрибуирају у виду *PostScript* докумената.
- *PostScript* - је језик за опис страница. Документи у овом формату се добијају најчешће штампањем у *PostScript* датотеку докумената креираних помоћу различитих апликација. Језик је првенствено намењен за штампарске потребе, није нарочито примерен размени докумената, мада се у одређеним академским круговима појављује доста често као паралелан формат *pdf* формату.

Десктоп апликације које би се користиле у јавним установама морале би имати могућност коришћења, односно прихватања барем докумената у *pdf* и *doc* формату, као и *rtf* текстуалних докумената. Иако *rtf* формат није широко распрострањен, те нуди само рудиментарне могућности формирања докумената, могуће га је креирати и једноставним програмима за обраду текста (нпр. WordPad у Windows окружењима), већина напредних програма за обраду текста подржава читање докумената у овом формату, па га је пожељно подржати. Уколико се као улазни формати прихватају формати MS Office пакета тј. MS Office је присутан на радним станицама, *rtf* формат је по дефиницији подржан. Како је формат докумената које користи OpenOffice отворени стандард (OpenDocument), за очекивати је да његово прихватање расте.

У погледу објављивања садржаја које тела јавних служби нуде путем система еУправе, неопходно је повести рачуна о **принципу једнаке доступности**. Ово у суштини значи да сви обрасци, документи и формулари чије преузимање у електронској форми је омогућено преко система еУправе, морају бити доступни у облику који је читљив **бесплатно** доступним апликацијама. За овакве документе може бити примењен неки од отворених или међународних стандарда (OpenDocument 1.0 – сада ISO/IEC 26300, ISO 19005-1:2005 PDF 1.4, W3C HTML). Могућа је и примена заштићених формата, али у том се случају на web страницама на којима се налази документ мора обезбедити и бесплатан приступ апликацији која је у могућности читати, снимати и штампати дати документ. Оваква бесплатна апликација мора бити доступна за најзаступљеније платформе (Windows, Linux, MacOS). У противном, употреба заштићених формата би била дискриминирајућа за кориснике који нису у могућности набавити легалан софтвер за рад са датим форматом докумената, те је неприхватљива за контекст еУправе. Неопходно је посветити и одређену пажњу

приликом израде web садржаја како би се избегло да креирани садржаји буду коректно читљиви само у одређеним *web browser*-има.

Open source алтернативе Microsoft Office канцеларијском пакету

Иако постоје различити канцеларијски пакети који су *open source/freeware* апликације, у последње три године, StarOffice (*open source*, али не и *freeware*) и OpenOffice се издвајају као канцеларијски пакети који су по функционалности најближи Microsoft Office пакету. Осим ових вреди поменути и KOffice и GNOMEOffice, али ови пакети су развијени за Linux оперативни систем. Остали пакети нису у фази високе разрађености, па не могу представљати озбиљнију конкуренцију. StarOffice и OpenOffice су већим делом идентични. OpenOffice је у потпуности *open source* пројекат, док StarOffice садржи још и одређени лиценци део, укључујући и апликације за рад са базама података. Компанија Sun продаје StarOffice као комерцијални производ.

StarOffice/OpenOffice је потенцијално интересантан за јавне службе јер је овај производ у фази високе развијености, поузданости и квалитета. Карактеришу га следеће особине:

- комплетан пакет канцеларијских апликација еквивалентан пакету MS Office,
- XML је у основи формата свих докумената са којима ради овај пакет,
- јако добра компатибилност са документима креираним одговарајућим Microsoft апликацијама (једини озбиљни проблеми се јављају при коришћењу VB макроа у MS документима), уз могућ делимични поремећај распореда елемената на страницама (у новој верзији OpenOffice 2.0 овај проблем је знатно боље решен).

OpenOffice конверзију најчешће обавља у потпуности без губитка информација, а у већини случајева ће бити у могућности конвертовати MS Office документе без додатних проблема. Опрезна процена, рађена у студији [OSS_in_eGov] је да ће до проблема при конверзији, али које је могуће превазићи, доћи у 10-20 % случајева. Проблеми компатибилности који се могу појавити уколико се део неке организације одлучи за ексклузивно коришћење OpenOffice пакета, док остали задрже само MS Office, били би у том случају ипак решиви. Ипак, у неким случајевима може се рачунати на одређене проблеме, барем у иницијалном периоду. Тешко је замислити, да ће, док се не успостави широко прихваћен отворени стандард за овакве документе, било који *open source* канцеларијски пакет бити у могућности без икаквих недостатака користити документе Microsoft Office пакета.

Успостављање MS Office-а као неформалног, али доминантног стандарда за канцеларијске апликације може се, са једне стране, посматрати и као позитивна чињеница, јер таква доминантна позиција омогућава да се са већином корисника, без икаквих проблема размењују документи. Са друге стране, компетитивност других канцеларијских пакета је доведена у питање, због могућих проблема са некомпатибилношћу. Ово може довести до тзв. *lock-in* ситуације, која осигурава и будућу доминантну улогу једног добављача (у овом случају Microsoft-а).

Од осталих десктоп апликација, од интереса су *web browser* и *email* клијентске апликације. Већина ових софтвера се тренутно дистрибуирају као бесплатни, па избор неког од решења нема директног утицаја на трошкове увођења информационог система. *Web browser*-и су, порастом броја web оријентисаних апликација, прерасли у клијентске апликације преко којих велики број корисника приступа различитим

апликацијама. Иако и овде постоји неоспорна доминација Microsoft Internet Explorer-а (IE), у овом подручју постоји и доста најразличитијих *freeware* и *open source* решења, која нарочито у последње време добијају на популарности. Своју доминантну позицију Internet Explorer првенствено дугује чињеници да је био један од првих *browser*-а који се дистрибуирао бесплатно са оперативним системом. Пораст популарности “алтернативних” *browser*-а произилази из чињенице да је у међувремену постојећа верзија (IE 6) постала доста застарела, врло је подложна нападима, а и није у потпуности компатибилна са HTML стандардима W3C конзорцијума. Тренутно најраширенији алтернативни *browser* је Mozilla Firefox - *open source* пројекат. Како је већина данас коришћених *browser*-а бесплатна, финансијски фактор не утиче на избор. Већина корисника који су се одлучили за *open source / freeware browser*-е, за њих се одлучује због њихове поузданости, мање рањивости на нападе, и у последње време веће брзине рада у односу на IE. Осим њега мањи број корисника користи и *browser* Opera, на MacOS платформама је најзаступљенији Safari, док је на Linux платформама уз Mozilla Firefox, врло заступљен и *browser* Mozilla у различитим верзијама, као и *browser*-и који се нуде као уграђени у одговарајуће десктоп окружење. На пољу *email* клијената такође је велики проценат корисника везан уз Microsoft Outlook/Outlook Express, мада и овде постоји широк спектар конкурентних производа. И у овом случају велики део ових апликација се дистрибуира бесплатно, па примарни услов за избор одређеног *email* клијента није цена, него стабилност и сигурност апликације. Због своје велике интегрисаности у сам оперативни систем Internet Explorer и Outlook Express су се показали и изузетно рањивим на различите нападе, више него остали производи. Статистика коришћења *web browser*-а у 2005/2006 години по месецима дата је у табели 2.1.

2006	IE7	IE6	IE5	Mozilla Firefox	Mozilla	Netscape Navigator	Opera
Мај	1.1%	56.7%	4.5%	25.7%	2.3%	0.3%	1.5%
Април	0.7%	57.3%	5.0%	25.2%	2.5%	0.4%	1.5%
Март	0.6%	58.8%	5.3%	24.5%	2.4%	0.5%	1.5%
Фебруар	0.5%	59.5%	5.7%	25.1%	2.9%	0.4%	1.5%
Јануар	0.2%	60.3%	5.5%	25.0%	3.1%	0.5%	1.6%
2005	IE6	IE5	Mozilla Firefox	Mozilla	Netscape Navigator 7	Opera 8	Opera 7
Децембар	61.5%	6.5%	24.0%	2.7%	0.4%	1.3%	0.2%
Новембар	62.7%	6.2%	23.6%	2.8%	0.4%	1.3%	0.2%
Октобар	67.5%	6.0%	19.6%	2.6%	0.4%	1.2%	0.2%
Септембар	69.8%	5.7%	18.0%	2.5%	0.4%	1.0%	0.2%
Август	68.4%	6.3%	18.9%	2.4%	0.4%	0.8%	0.3%
Јул	67.9%	5.9%	19.8%	2.6%	0.5%	0.8%	0.4%
Јун	65.0%	6.8%	20.7%	2.9%	0.6%	0.7%	0.5%
Мај	64.8%	6.8%	21.0%	3.1%	0.7%	0.7%	0.6%
Април	63.5%	7.9%	20.9%	3.1%	0.9%	0.4%	1.0%
Март	63.6%	8.9%	18.9%	3.3%	1.0%	0.3%	1.6%
Фебруар	63.9%	9.5%	17.9%	3.3%	1.0%		1.7%
Јануар	64.8%	9.7%	16.6%	3.4%	1.1%		1.9%

Табела 2.1. Извор: http://www.w3schools.com/browsers/browsers_stats.asp

(статистика се даје уз ограду да је анализа рађена на сајту w3с, што не мора у потпуности да одсликава опште стање, али даје генерални тренд)

2.2. Серверски оперативни системи

Врло битан део софтверске инфраструктуре представљају оперативни системи (ОС) који се користе за сервере. Ови оперативни системи због специфичне намене морају бити пре свега робусни, поуздани и отпорни на малициозне софтверске нападе. Као додатне пожељне особине било би неопходно обратити пажњу и на скалабилност, као и на захтевану хардверску платформу. Иако тренутно у свету постоји велики број оперативних система, само неколико је широко прихваћено и у општој је употреби. У табели 2.2. дат је преглед основних података оперативних система који су погодни за серверске платформе. Маркирани су системи који су разматрани као решење прихватљиво за потребе информационог система ИВ Војводине тј. за потребе система еВојводина. Табеле коришћене у овом поглављу (редуковане за оперативне системе који нису погодни за серверске системе) преузете су са [wikipedia-OS]. FreeBSD, OpenBSD, NetBSD (BSD – Berkeley Software Distribution – Unix оперативни систем развијен на калифорнијском Berkeley универзитету) су оперативни системи настали на темељу 386BSD оперативног система, тј. првог преноса BSD Unix-а на Intel 386 платформу. FreeBSD је директни наследник ове верзије оперативног система, NetBSD је верзија настала са циљем да подржи већи број платформи, а OpenBSD је верзија у којој се екстремно велика пажња поклања сигурносним питањима. Због сличности између ових верзија врло често се неке од њих посматрају заједнички нпр.

Free/OpenBSD. Више информација о верзијама BSD Unix-а могуће је наћи на [BSDRoundup].

Табела 2.3. приказује основне техничке карактеристике наведених оперативних система. У табели су дати подаци о:

- **Подржаним типовима процесора**, што је врло важна карактеристика, јер избор одређеног оперативног система који је развијен за рад на само одређеним породицама процесора, може надаље форсирати велика улагања у набавку специфичног хардвера. Осим тога може узроковати проблеме при накнадним надоградњама.
- **Подржаним типовима *filesystem*-а**. Иако сам тип датотечног система, тј. организација похрањивања података на дискове, није од претеране важности за крајњег корисника, пожељно би било да изабрани оперативни систем подржава неке стандардизоване *filesystem*-е као што су ISO 9660 (формат записа на CD-y), UDF формат (ISO/IEC 13346, користи се као формат на магнето-оптичким, DVD, DVD-R и DVD-RW медијима, CD-RW медијима), FAT (није стандард, али је један од основних формата који користи Windows, па би барем могућност читања медија форматираних на овај начин била пожељна).
- **Типу кернела**. (језгре) оперативног система. Кернели се деле по начину организације (и функционалностима које су уграђене у сам кернел) на микрокERNELе, монолитне KERNELе (најчешће са модулима који реализују додатне системске функције које нису уграђене у сам кернел) и хибридне врсте кернела.
- **Постојање подразумеваног графичког радног окружења**. Иако је за кориснике постојање графичког интерфејса велико олакшање, код серверских ОС, које по правилу одржавају администратори, и при томе се реконфигурација софтвера на серверима не дешава свакодневно, графички интерфејс није од пресудне важности. Уколико постоји то је свакако погодност.
- **Начину управљања софтверским пакетима**, ова особина заправо показује да ли ОС долази са неким алатима помоћу којих је могуће на једноставан, сигуран и поуздан начин вршити инсталацију и деинсталацију софтвера.
- **Начину управљања софтверским надоградњама**, ова особина првенствено говори о начинима на које је могуће извршити надоградњу пре свега системског софтвера тј. делова самог оперативног система.
- **Примарним API-јима** које је могуће користити за развој апликација за наведени оперативни систем.

У табели 2.4. дат је приказ одређених сигурносних система уграђених у наведене оперативне системе. Сигурносни системи који су уграђени у сам оперативни систем су врло значајни за поуздан рад система, јер могу спречити неовлашћени приступ ресурсима рачунара и разне врсте напада чији би крајњи циљ био компромитовање целокупног информационог система и/или ускраћивање сервиса које сервери пружају корисницима (DoS напади). Аспекти сигурносних система који су дати у табели су:

- **контрола приступа ресурсима** – начин на који ОС контролише приступ ресурсима рачунарског система. Unix базирани и POSIX компатибилни системи имају једноставан и врло ефикасан систем доделе права приступа за сваку поједину датотеку у систему. Права приступа се дефинишу посебно за три класе корисника (власника датотеке тј. корисника који ју је креирао - *user* класа корисника, групу којој корисник припада - *group* класа корисника и остале кориснике система – *others*

класа корисника). Свакој од ових класа могуће је доделити или одузети право читања, мењања (писања) и извршавања одређене датотеке. У табели је могућност ОС-а да користи овакав принцип контроле приступа означена са *unix*. Овакав начин контроле представља симплифицирану форму листи права приступа. POSIX такође омогућава и коришћење тзв. листи права приступа (ACL – Access Control List). Овај принцип је примењен само на одређеним датотечним системима и у одређеним верзијама *unix-like* оперативних система. DOS базирани системи (што укључује и Windows 95, Windows 98, и Windows Me) уопште не поседују контролу права приступа, могуће је само поставити *read-only* атрибут, али се ово дозвољава било ком кориснику. Ови ОС се и не користе као серверски па се аспекти њихове сигурности овде не разматрају. MacOS X, Microsoft Windows NT и његови деривати (укључујућу ту и Windows 2000, Windows 2003 Server и Windows XP), као и VMS и OpenVMS користе првенствено листе за контролу права приступа (ACL) за доделу сложенијих права приступа (MacOS X може користити и *unix-like* контролу права приступа). Неки оперативни системи имају и одређене варијације ова два примарна начина контроле права приступа.

- **механизми изолације подсистема** – тзв. *sandbox* је сигурносни механизам који омогућава сигурно извршавање програма. Врло често се користи за покретање процеса који извршавају нетестиран код, или код добијен од треће неверификоване стране, или за програме које покрећу корисници који немају високе сигурносне привилегије. *Sandbox* типично креира стриктно контролисано окружење у коме се процес одвија и у оквиру којег се контролише приступ ресурсима система, нпр. приступ мрежним уређајима, права за претраживање *host* система, читање са улазних уређаја и сл. Овај механизам представља врсту виртуелизације којом се за кориснички процес креира виртуелна машина чија комуникација са хардверским уређајима је строго контролисана. Постоји више механизма којима се ово постиже, неки од њих су наведени у табели:
 - *chroot* – на *unix* базираним ОС ова операција омогућава да се промени корени директоријум који користи тренутни процес. Процес коме је на овај начин промењен корени директоријум не може приступати датотекама које се налазе ван датог директоријума. На овај начин креира се *sandbox* за дати процес. *chroot* представља поједностављену верзију *jail* механизма.
 - *jail* – већина имплементација овог механизма ради на концепту партиција, у којима се група програма са различитим кореним директоријумима изолије и којој се даје заједнички приступ ресурсима (унапред ограниченим), као што су време коришћења процесора, простор на дисковима. Свакој партицији се могу поставити ограничења и на улазно – излазни подсистем и приступ мрежним ресурсима. На *unix* базираним системима *jail* представља екстензију стандардног *chroot* механизма.
 - *sysrtrace* – ограничава приступ рачунарским ресурсима од стране апликација на тај начин што успоставља политику права приступа системским рутинама. На овај начин могу се смањити ефекти сигурносних проблема који се јављају услед преплављивања бафера и сл. Користи се на различитим верзијама *unix* базираних система. *sysrtrace* је нарочито користан када се покрећу непроверене или апликације које се дистрибуирају само у бинарној форми.
 - *zones* - је још једна варијација *jail* механизма уведена у ОС Solaris

- *capabilities* – принцип по коме се кориснички програми развијају тако да директно деле одређена права (*capabilities* или *keys*) у складу са правима приступа ресурсима.
- остали системи приказани у табели по правилу представљају одређене варијације ових техника

Циљ свих ових техника јесте да омогуће покретање процеса у контролисаном окружењу чиме се смањују потенцијалне штете које малициозни процеси могу проузроковати, првенствено се смањује могућност компромитовања серверског система због "изгладњивања" критичних серверских процеса. Малициозни процес би могао покушати одузети неопходне ресурсе (процесорско време, меморијски простор, диск простор, мрежни пропусни опсег и сл), што би у случају да не постоје овакви механизми изолације могло довести до пада перформанси или отказа серверског система. Јасно је да серверски оперативни систем који нема бар неки од ових механизма заштите не може пружити довољну поузданост.

- **постојање интегрисаног firewall-a** – *firewall* представља хардверску и/или најчешће софтверску компоненту чија је улога да спречи или омогући одређену мрежну комуникацију у складу са унапред дефинисаном сигурносном политиком. Понекад се овакви системи називају и **Border Protection Device (BPD)**, или филтри пакета. Основни задатак *firewall-a* је да контролише саобраћај између зона мреже са различитим нивоима сигурности. Типична употреба је за контролу саобраћаја између Интернет зоне и зоне локалне рачунарске мреже у одређеној институцији. Постојање интегрисаног *firewall-a* је пожељна особина за сваки ОС, поготово за ОС намењен за рад на серверима.
- **енкрипција система датотека (filesystem encryption)** – омогућава да се део датотечног система заштити енкрипцијом и тиме повећа сигурност похрањених података. Ова особина је пожељна за чување сигурносно врло осетљивих података.
- **заштита од извршења нежељеног кода** – ова заштита има за циљ да спречи неовлашћено извршавање програмског кода до које може доћи код нпр. напада преплављивањем бафера. Може се обезбедити хардверски или софтверском емулацијом. Хардверска заштита темељи се на коришћењу посебног бита за контролу права извршавања кода (код x86 фамилије процесора то је NX – *no eXecute* бит у сваком запису *paging* табеле којим се контролише да ли се дати код може извршити у датој меморијској зони). Овај концепт од раније постоји у другим фамилијама процесора. Коришћење ове технике се показало као доста успешно управо у спречавању напада техником преплављивања бафера.
- **број познатих нерешених сигурносних пропуста** – овај податак треба стално проверавати, јер се нови пропусти проналазе скоро свакодневно. Врло користан извор информација о новооткривеним сигурносним пропустима јесте United States Computer Emergency Readiness Team [US-CERT]. Овај центар за експертизу сигурности Интернета се бави управо проналажењем, те анализом опасности које представљају одређени пропусти како у оперативним системима, тако и у апликацијама. Основан је 2003. године са циљем заштите интернет инфраструктуре САД-а, као и за координацију активности које представљају одговор на *cyber* нападе. Центар обезбеђује седмичне извештаје о сигурносним пропустима који могу бити потенцијално опасни по сигурност система, података и мрежне инфраструктуре. Центар користи тзв. метрику у распону од 0 до 180 којим се описује степен опасности који одређени пропуст представља.

Назив	Произвођач	Први пут објављен	Претходна верзија	Датум последња стабилне верзије	Тип лиценце	Тип циљног система
AIX	IBM	1986	System V R3	5.3 (Август 2004)	У власништву произвођача (proprietary)	Server, Network Appliance, Workstation
FreeBSD	The FreeBSD Project	Децембар 1993	386BSD	6.0 (3.11.2005)	BSD	Server, Workstation, Network Appliance
HP-UX	Hewlett-Packard	1983	Unix	11.23 "11i v2" (Децембар 2005)	У власништву произвођача (proprietary)	Server, Workstation
Inferno	Bell Labs	1997	Plan 9	Fourth Edition / 20.07.2005	MIT/GPL/LGPL/LPL	Network Appliance, Server, Embedded
GNU/Linux	Различити аутори	Септембар 17, 1991	Minix	Kernel 2.6.16 (20.03.2006)	Најчешће GNU GPL/Copyleft	Различите дистрибуције за различите намене
Mac OS X	Apple Computer	Март 2001.	NeXTSTEP, Mac OS	10.4.5 "Tiger" (14.02.2006)	Proprietary, делом APSL, GPL, и остале врсте лиценци.	Workstation, Home Desktop, Server (посебна верзија)
NetBSD	The NetBSD Project	Мај 1993	386BSD	3.0 (23.12.2005)	BSD	Network Appliance, Server, Workstation, Embedded
NetWare	Novell	1985	S-Net	6.5 SP4 (Септембар 2005)	У власништву произвођача (proprietary)	Server
OpenBSD	The OpenBSD Project	Октобар 1995	NetBSD 1.0	3.8 (Новембар 1, 2005)	BSD	Server, Network Appliance, Workstation, Embedded
OpenVMS	DEC (now HP)	Фебруар 1978	RSX-11M	8.2-1 (Септембар 2005)	У власништву произвођача (proprietary)	Server
OS/2	IBM / Microsoft	Децембар 1987	MS-DOS	4.52 (Децембар 2001)	У власништву произвођача (proprietary)	Home Desktop, Server
Plan 9	Bell Labs	1993	Unix	Fourth Edition (Daily snapshots)	LPL	Workstation, Server, Embedded, HPC
Solaris	Sun	Јул 1992	SunOS	10 1/06 (Јануар, 2006)	CDDL	Server, Workstation
Windows Server 2003	Microsoft	Април 2003	Windows 2000	5.2 SP1 (30.03.2005)	У власништву произвођача (proprietary)	Server

Табела 2.2. Општи подаци о оперативним системима – извор http://en.wikipedia.org/w/index.php?title=OS_comparison&redirect=no

Назив	Подржане архитектуре процесора	Подржани file системи	Тип кернела	Интегрисано графичко окружење GUI	Управљање софтверским пакетима	Управљање update-има	Примарни API
AIX	POWER, PowerPC	JFS, JFS2, ISO 9660, UDF, NFS, SMBFS, GPFS	Микрокернел	Не	installp, RPM	Service Update Management Assistant (SUMA)	SysV, POSIX
FreeBSD	x86, AMD64, PC98, SPARC, others	UFS2, ext2, FAT, ISO 9660, UDF, NFS, and others	Монолитни са модулима	Не	ports tree, packages	by source (CVSup), network binary update (freebsdupdate)	BSD, POSIX
HP-UX	PA-RISC, IA-64	VxFS, HFS, ISO 9660, UDF, NFS, SMBFS	Монолитни са модулима	Не	Swinstall	?	SysV, POSIX
Linux	x86, Alpha, AMD64, PPC, SPARC, others	ext2, ext3, ReiserFS, FAT, ISO 9660, UDF, NFS, and others	Монолитни са модулима	Не	Зависе од дистрибуције		POSIX
Inferno	x86, Alpha, MIPS, PPC, SPARC, others	Styx/9P2000, kfs, FAT, ISO 9660	Монолитни са модулима	Да	-	?	proprietary
Mac OS X	PPC, x86	HFS+ (default), HFS, UFS, AFP, ISO 9660, FAT, UDF, NFS, SMBFS, NTFS (read only), FTP, WebDAV	Хибридни	Да	OS X Installer	Software Update	Carbon, Cocoa, BSD/POSIX, X11 (од верзије 10.3)
NetBSD	x86, 68k, Alpha, AMD64, PPC, SPARC, many others	UFS, UFS2, ext2, FAT, ISO 9660, NFS, LFS, and others	Монолитни са модулима	Не	Pkgsrc	by source (CVS, CVSup, rsync) or binary (using sysinst)	BSD, POSIX
NetWare	x86	NSS, NWFS, FAT, NFS, AFP, UDF, ISO 9660	Хибридни	Не	NWCONFIG.NLM, RPM	binary updates, Red Carpet	proprietary
OpenBSD	x86, 68k, Alpha, AMD64, SPARC, VAX, others	UFS, ext2, FAT, ISO 9660, NFS, some others	Монолитни са модулима	Не	ports tree, packages	by source	BSD, POSIX
OpenVMS	VAX, Alpha, IA-64	Files-11, ISO 9660, NFS	Монолитни са модулима	Не	PCSI, VMSINSTAL	-	Unix-like
OS/2	x86	HPFS, JFS, FAT, ISO 9660, UDF, NFS	Монолитни	Да	Feature Install and others	-	proprietary
Plan 9	x86, Alpha, MIPS, PPC, SPARC, others	fossil/venti, 9P2000, kfs, ext2, FAT, ISO 9660	Монолитни	Да	None	replica	Unix-like (као и опциони слој за POSIX компатибилност)
Solaris	SPARC, SPARC64, AMD64, x86	UFS, ZFS, ext2, FAT, ISO 9660, UDF, NFS, some others	Монолитни са модулима	Не	SysV packages (pkgadd)	Sun Update Connection	SysV, POSIX
Windows Server 2003	x86, AMD64, IA-64	NTFS, FAT, ISO 9660, UDF	Хибридни	Да	MSI, custom installers	Windows Update	Win32, Win64, .NET

Табела 2.3. – Основне техничке карактеристике оперативних система – извор http://en.wikipedia.org/w/index.php?title=OS_comparison&redirect=no

Напомена: већина оперативних система који немају посебно GUI окружење, дистрибуирају се са GUI окружењима базираним на X Windows.

Назив	Контрола приступа ресурсима	Механизми изолације подсистема	Интегрисани <i>firewall</i>	Енкрипција датотечног система	Заштита од извршења нежељеног кода (Data execution prevention)		Број познатих нерешених сигурносних пропуста	
					хардверска	емулација	број	најстарији
AIX	Unix, ACL	chroot	none	Не	?		2	2002-10-11
FreeBSD	Unix, ACL, MAC	chroot, jail, MAC Partitions	IPFW2, IPFilter, PF	Да	?		0	-
HP-UX	Unix, ACL	chroot	IPFilter	Не	?		0	-
Inferno	Unix	Namespaces	?	?	Не	Не	нема доступних података	
Linux	Unix, ACL, MAC	chroot, capabilities, seccomp, SELinux	Netfilter/Varied by distribution	Да	Да	Не	Зависи од дистрибуције	
Mac OS X	Unix, ACL	chroot	ipfw	Да	?		0	нема доступних података
NetBSD	Unix, Veriexec	chroot, systrace	IPFilter, PF	Да	Да	Не	нема доступних података	
NetWare	ACL	none	IPFLT.NLM	Да	Да	Не	1	2003-10-30
OpenBSD	Unix	chroot, systrace	PF	Да	Да	Да	0	-
OpenVMS	ACL	logical name tables	?	?	?		0	-
OS/2	ACL	нема	нема	Не	?		нема доступних података	
Plan 9	Unix (?)	Namespaces	ipmix	Да	Не	Не	нема доступних података	
Solaris	Unix, Role Based Access Control (RBAC)	chroot, Zones	IPFilter	?	Да	Да	2	2005-04-13
Windows Server 2003	ACL	нема	Windows Firewall	Да	Да	Не	6	2003-06-11

Табела 2.4. – Сигурносни системи које одређени оперативни системи подржавају – извор http://en.wikipedia.org/w/index.php?title=OS_comparison&redirect=no

(Unix) – права приступа датотекама по *unix* принципу

(ACL) - access control list – листе за контролу приступа

(MAC) - mandatory access control - обавезна контрола приступа

2.3. Системи за управљање базама података – RDBMS системи

Database management system (DBMS) је рачунарски програм, или најчешће пакет програма, дизајнираних за управљање базама података. Ови програми омогућавају чување, модификовање и екстраховање података из базе података. За развој информационих система су се историјски користили различити модели података, но данас је у комерцијалној употреби преовлађујући тзв. релациони модел. Системи за управљање базама података који се данас користе најчешће су базирани на овом моделу и називају се системи за управљање релационим базама података (RDBMS системи). У последње време у употреби су и објектне базе података и XML базе податка, али су и даље преовлађујући системи управо релациони. У даљем тексту биће анализиране особине управо система за управљање релационим базама података. Табеле преузете у овом поглављу могу се наћи на [wikipedia-RDBMS].

У табели 2.5. дате су основне информације о неким RDBMS системима.

Назив	Производ одржава	Датум објаве прве верзије	Последња стабилна верзија	Тип лиценце
Adaptive Server Enterprise	Sybase	1987	15.0	У власништву произвођача
ADS	ADS	2003	3.4	У власништву произвођача
DB2	IBM	1982	8.2	У власништву произвођача
Firebird	Firebird Foundation	25.07.2000	1.5.3	Initial Developer's Public License
Informix	IBM	1985	10.0	У власништву произвођача
HSQldb	HSQl Development Group	2001	1.8.0	BSD
H2	H2 Software	2005	0.9	Freeware
Ingres	Ingres Corp.	1974	Ingres 2006 II 9.0.4	GPL и у власништву произвођача
InterBase	Borland	1985	7.5.1	Proprietary
MaxDB (SAP DB)	MySQL AB, SAP AG	?	7.6	GPL или у власништву произвођача
Microsoft SQL Server	Microsoft	1989	9.00.1399 (2005)	У власништву произвођача
MySQL	MySQL AB	Novembar 1996	5.0	GPL или у власништву произвођача
Oracle	Oracle Corporation	1977	10g Release 2	У власништву произвођача
PostgreSQL	PostgreSQL Global Development Group	Jun 1989	8.1.3	BSD
Pyrrho DBMS	University of Paisley	Novembar 2005	0.5	У власништву произвођача
SmallSQL	SmallSQL	16.04.2005	0.12	LGPL
SQLite	D. Richard Hipp	17.08.2000	3.3.4	Public domain
Teradata	Teradata	1984	V2R6.1	У власништву произвођача

Табела 2.5. Опште информације о системима за управљање базама података

У табели 2.6. дата је листа оперативних система на којима се одређене базе података могу покренути. Ова особина је врло важна за прелиминарни избор погодног система за управљање базама података. Базе података које је могуће инсталирати и покренути на само одређеним оперативним системима сужавају могућност избора при накнадним надоградњама софтвера. Најпогоднији су они системи који могу радити на више различитих оперативних система, јер то омогућава накнадну промену или покретање додатних сервера, а да је при томе миграција самих података једноставна, јер се не мења сам тип RDBMS система.

Назив	Windows	Mac OS X	Linux	BSD	UNIX	z/OS ¹
Adaptive Server Enterprise	Да	Да	Да	Да	Да	Не
ADS	Да	Не	Да	Да	Да	Да
DB2	Да	Не	Да	Не	Да	Да
Firebird	Да	Да	Да	Да	Да	Можда
HSQLDB	Да	Да	Да	Да	Да	Да
H2	Да	Да	Да	Да	Да	Можда
Informix	Да	Да	Да	Да	Да	Не
Ingres	Да	Да	Да	Да	Да	Можда
InterBase	Да	Не	Да	Не	Да (Solaris)	Не
Adabas	Да	Не	Да	Не	Да	Да
MaxDB (SAP DB)	Да	Не	Да	Не	Да	Можда
Microsoft SQL Server	Да	Не	Не	Не	Не	Не
MySQL	Да	Да	Да	Да	Да	Можда
Oracle	Да	Да	Да	Не	Да	Да
PostgreSQL	Да	Да	Да	Да	Да	Можда
Pyrrho DBMS	Да (.NET)	Не	Да (MoNe)	Не	Не	Не
SmallSQL	Да	Да	Да	Да	Да	Да
SQLite	Да	Да	Да	Да	Да	Можда
Teradata	Да	Не	Да	Не	Да	Не

Табела 2.6. – Оперативни системи на којима се одређене базе података могу покренути

напомена (1): Open source базе података које су UNIX компатибилне, се врло често могу покренути и под z/OS системом у оквиру уграђеног UNIX System Services (USS) подсистема. Већина база које су Linux-компатибилне могу се покренути и на z/OS на истом серверу, уколико се користи Linux за zSeries.

Основне особине RDBMS система

У табели 2.7. дате су карактеристике наведених RDBMS система у погледу основних особина дефинисаних за релационе базе података. Такође истакнута је и способност одређене базе података да користи Unicode сет карактера. Подршка за Unicode сет карактера је критична, нарочито у окружењима где се очекује вишејезичност коришћених текстуалних података. Коришћење Unicode карактер сета омогућава обраду свих писама на јединствен начин без потребе мењања кодне странице са којом сервер RDBMS система ради. Остале особине приказане у табели су:

- **ACID** – акроним за **A**tomicity, **C**onsistency, **I**solation, and **D**urability. Ове особине се сматрају кључним за трансакциону обраду и гарантовање интегритета података.
 - **атомичност (atomicity)** је особина базе података која гарантује да ће се или све акције у оквиру трансакције извршити или ниједна. Ова особина у суштини омогућава да се трансакција посматра као једна недељива (атомска) операција над подацима.

- **конзистентност (consistency)** се односи на стање базе на почетку и на крају трансакције. Ово једноставно значи да трансакција, ма шта обављала не сме резултовати стањем базе које нарушава правила интегритета података. Нпр. уколико се у одређеној колони дозвољавају само не-нула вредности трансакција не сме резултовати постављањем нула вредности у дату колону.
- **изолација (isolation)** се односи на способност апликације да изолује трансакцију од свих других процеса. Овај захтев се поставља како би се спречило да било који процес види прелазне резултате који су резултат одређене фазе извршења трансакције, а не представљају финални резултат.
- **трајност (durability)** је особина која гарантује да ће након што је клијент примио потврду да је трансакција успешно завршена, подаци остати трајно промењени, тј. да након тога трансакција не може бити поништена. Између осталог резултати трансакције би након потврде о извршењу морали преживети и пад система. Због овога се на већини система води тзв. *transaction log*.
- **Референцијални интегритет** – референцијалним интегритетом се гарантује конзистентност података између табела које су повезане одређеном релацијом. Референцијални интегритет се најчешће имплементира комбинацијом примарних кључева, кључева кандидата и страних кључева. Како би се референцијални интегритет одржао, било које поље табеле, које је декларисано да садржи страни кључ, не сме садржати ниједну вредност која се не налази као примарни (или кандидат) кључ у родитељској табели. Уколико RDBMS не гарантује референцијални интегритет, не може се заиста сматрати релационом базом. Код таквих система провера валидности података се преноси на ниво корисничке апликације, што усложњава развој апликације, а сам ниво поузданости је смањен, јер је могуће превидети одређене начине измене података који би довели до невалидног стања података у везаним табелама.
- **Трансакциони режим** – ова особина говори да ли посматрани RDBMS систем подржава концепт трансакција. Трансакција представља скуп акција над базом које се обављају као недељива целина. Ову особину треба посматрати у конјункцији са ACID.

Назив	ACID	Референцијални интегритет	Трансакциони режим	Подршка за Unicode
Adaptive Server Enterprise	Да	Да	Да	Да
ADS	Да	Да	Да	Да
DB2	Да	Да	Да	Да
Firebird	Да	Да	Да	Да
HSQLDB	Да	Да	Да	Да
H2	Да	Да	Да	Да
Informix	Да	Да	Да	Да
Ingres	Да	Да	Да	Да
InterBase	Да	Да	Да	Да
MaxDB (SAP DB)	Да	Да	Да	Да
Microsoft SQL Server	Да	Да	Да	Да
MySQL	Условно ¹	Условно ¹	Условно ¹	Да / UTF-8 (3-byte) & UCS-2
Oracle	Да	Да	Да	Да
PostgreSQL	Да	Да	Да	Да / UTF-8 (4-byte)
Pyrrho DBMS	Да	Да	Да	Да
SQLite	Да	Не ²	Основни ²	Да
Teradata	Да	Да	Да	Да

Табела 2.7. – Основне особине RDBMS система

напомена (1): Како би се обезбедио трансакциони и референцијални интегритет, мора се користити InnoDB тип табеле. Подразумевани тип табела MyISAM не подржава ове особине. Чак и InnoDB тип табеле дозвољава похрану података чија вредност превазилази опсег декларисаног типа у табели, што се такође у неким случајевима сматра нарушавањем интегритета.

напомена (2): FOREIGN KEY ограничења се парсирају, али се не спроводе. Да би се ово исправило могуће је користити окидаче. Угнежене трансакције нису подржане.

Додатни типови табела и погледи (*views*)

У табели 2.8. дате су особине RDBMS система у погледу могућности креирања привремених табела, као и подршке за концепт погледа (*view*).

- **Привремене табеле** – користе се врло често за смештање међурезултата трансакција или приликом увођења података из других база пре него се резултати трајно сачувају у стандардним табелама, као и током извршавања одређених процедура (*stored procedure*) у бази. Подршка за креирање привремених табела је из ових разлога врло значајна карактеристика DBMS система.
- **Материјализовани погледи** - Код база података *view* представља виртуелну табелу која садржи колоне дефинисане корисничким упитом. Кад год се врши упит над овом табелом или промена њеног стања RDBMS систем конвертује ове упите или модификације у одговарајуће упите којима се манипулише оригиналним табелама које се посматрају кроз дати *view*. **Материјализовани поглед** је другачији приступ, код којег се табела која је дефинисана корисничким упитом формира као конкретна табела, која се релативно ретко освежава из оригиналних табела. Ово омогућава подизање перформанси система, али носи опасност да се поједини подаци у оригиналним табелама понекад нађу у неажурном стању. Други бенефит од постојања ове табеле као конкретне табеле у бази је у томе што је над њом могуће обавити све операције као и над било којом другом стандардном табелом, укључујући и изградњу индекса. Овај концепт материјализованог

погледа је имплементиран прво у Oracle-у. Иако ова особина може да подигне перформансе система, није од критичне важности за функционисање RDBMS.

Назив	Привремене табеле	Материјализовани погледи (Materialized view)
Adaptive Server Enterprise	Да ¹	Не
ADS	Да	Да
DB2	Да	Да
Firebird	Не	Не
HSQLDB	Да	Не
H2	Да	Не
Informix	Да	Да
Ingres	Да	Ingres r4
InterBase	Да	Не
MaxDB (SAP DB)	Да	Не
Microsoft SQL Server	Да	Слично ²
MySQL	Да	Не
Oracle	Да	Да
PostgreSQL	Да	Слично ³
Pyrho DBMS	Не	Не
SQLite	Да	Не
Teradata	Да	Да

Табела 2.8. – Подршка за додатне типове табела и погледе

напомена (1): Сервер обезбеђује тзв. *tempdb*, која се може користити за јавне и приватне (током сесије) привремене табеле

напомена (2): MS SQL сервер обезбеђује индексиране погледе (*indexed views*)

напомена (3): *Materialized views* могу се емулирати коришћењем PL/pgSQL, PL/Perl, PL/Python, или других процедуралних језика.

Индексирање

У табели 2.9. дате су информације о типовима индексирања које одређени RDBMS системи подржавају (пored стандардних индекса B-/B+ стабла – подаци у базама података се обично организују у некој форми Б-стабла).

- **R-/R+ стабла** - су структуре сличне Б-стаблу, али се користе за просторне методе приступа, тј. за индексирање мултидимензионалних података. Нпр. ови индекси су погодни за индексирање просторних података по *x* и *y* (или више) координатама или при раду са мултимедијалним подацима. Корисност оваквих индекса је што омогућавају нпр. проналажење информације о објектима који се налазе у оквиру одређене удаљености од задате (*x,y*) локације.
- **Hash - hash функција** или **hash алгоритам** представља функцију (алгоритам) којим се анализирају улазни подаци и на основу њих генеришу излазни подаци фиксне дужине. Ови излазни подаци се називају *hash* вредност. Два различита улаза имају малу вероватноћу генерисања истоветног излаза. Индексирање по овом принципу омогућава да се подаци пронађу задавањем *hash* вредности као кључа претраживања.
- **Изрази** – индексирање израза омогућава да се индексирање обави на основу резултата задатог израза.
- **Парцијални** – индекси који се дефинишу само над одређеним делом табеле, у зависности од неког услова.

- **Реверзни** – индекси који се граде на основу реверзне вредности кључа. Овакви индекси могу знатно да убрзају претраживање табела чији индекси садрже неки унапред задати префикс.

Назив	R-/R+ стабло	Hash	Izrazi	Parcijalni	Reverzni	Bitmap	GiST
Adaptive Server Enterprise	Не	Не	Не	Не	Да	Не	Не
ADS	Да	Да	Да	Не	Да	Да	Не
DB2	Не	?	Не	Не	Да	Да	Не
Firebird	Не	Не	Не	Не	Не	Не	Не
HSQLDB	Не	Не	Не	Не	Не	Не	Не
H2	Не	Не	Не	Не	Не	Не	Не
Informix	Да	Да	Да	Да	Да	Да	Не
Ingres	Да	Да	Ingres r4	Не	Не	Ingres r4	Не
InterBase	?	?	Не	Не	Не	Не	Не
MaxDB (SAP DB)	?	?	Не	Не	Не	Не	Не
Microsoft SQL Server	?	Hen/Cluster & fill factor	Не	Не	Не	Не	Не
MySQL	Само за табеле типа MyISAM	Само за табеле типа HEAP	Не	Не	Не	Не	Не
Oracle	Само у EE верзији	Cluster Tables	Да	Не	Да	Да	Не
PostgreSQL	Да	Да	Да	Да	Да ¹	Да	Да
Pyrrho DBMS	Не	Не	Не	Не	Не	Не	Не
SQLite	Не	Не	Не	Не	Да	Не	Не
TeraData	Не	Да	Да	Да	Не	Да	Не

Табела 2.9. – Подржани типови индексирања

напомена (1): функционално индексирање код PostgreSQL може се користити за креирање реверзног индекса

- **Bitmap** – корисни су за процесирање сложених упита.
- **GiST** - **GiST** (*Generalized Search Tree*), је структура података и API који омогућава изградњу произвољног стабла претраживања над скоро свим типовима података. Данас је овако генерализован концепт подржан само у PostgreSQL.

Остали објекти

У табели 2.10. дат је преглед додатних DB објеката које одређени RDBMS системи подржавају.

- **Домен** – описује способност базе да подржи ограничења на вредности података у одређеним колонама, не само по питању типа или максималног опсега, него и провером да ли је вредност унутар унапред задатог скупа дозвољених вредности (домена).
- **Курсор** – курсор у суштини представља *handle* за скуп резултата неког SQL упита. Помоћу курсора могуће је кретати се кроз скуп резултата унапред или уназад за по један или више редова.
- **Окидачи (тригери)** – представљају процедуру која се обавља аутоматски приликом обављања неке друге акције (нпр. уметања новог реда у табелу, или брисања реда из табеле). Тригер може бити подешен тако да се обавља

непосредно пре или непосредно после акције на коју реагује. Помоћу тригера могуће је аутоматизовати ажурирање података у одређеним везаним табелама.

- **Функције и процедуре** – оба појма показују да ли дати RDBMS систем подржава писање интерних рутина коришћењем SQL-а и/или неког процедуралног језика карактеристичног за дати систем (нпр. PL/SQL).
- **Екстерне рутине** – рутине за ажурирање базе података, али које су писане у неком независном програмском језику (нпр. C).

Назив	Домен	Курсори	Окидачи <i>Trigger</i>	Функције	Процедуре	Екстерне рутине
Adaptive Server Enterprise	Да	Да	Да	Да	Да	Да
ADS	Да	Да	Да	Да	Да	Да
DB2	Не	Да	Да	Да	Да	Да
Firebird	Да	Да	Да	Да	Да	Да
HSQLDB	?	Не	Да	Да	Да	Да
H2	?	Не	Да	Да	Да	Да
Informix	?	Да	Да	Да	Да	Да
Ingres	Да	Да	Да	Да	Да	Да
InterBase	Да	Да	Да	Да	Да	Да
MaxDB (SAP DB)	Да	Да	Да	Да	Да	?
Microsoft SQL Server	Да (од вер. 2000)	Да	Да	Да	Да	Да
MySQL	Не	Да	Да	Не	Да	Да
Oracle	Да	Да	Да	Да	Да	Да
PostgreSQL	Да	Да	Да	Да	Да	Да
Pyrrho DBMS	Да	Да	Да	Да	Да	Да
SQLite	Не	Не	Да	Не	Не	Да
TeraData	Не	Да	Да	Да	Да	Да

Табела 2.10. – DB објекти које одређени RDBMS системи подржавају

Партиционисање

Партиционисање представља принцип поделе логички јединствене базе података у физички независне делове. Ово се обично ради како би се постигла боља управљивост, боље перформансе и повећање доступности података. Партиционисање се ради у складу са неким одабраним критеријумом. Уобичајени критеријуми за партиционисање су:

- **Range партиционисање** (по опсегу кључа за партиционисање) – код оваквог партиционисања одлука о партицији у којој се налазе подаци се доноси на основу вредности одговарајућег поља које је кључ за партиционисање.
- **Hash партиционисање** – вредност која се добије из *hash* функције представља услов за проверу припадности одређеној партицији.
- **Композитно партиционисање** – представља одређену комбинацију претходних шема.
- **Партиција помоћу листи** – свакој партицији се додељује листа вредности кључа за партиционисање. Поређењем вредности кључа за партиционисање и листе вредности за одређену партицију доноси се одлука да ли одређени подаци припадају или не датој партицији.

Табела 2.11. садржи информације о томе које шеме партиционисања су подржане од стране одређених RDBMS система.

Назив	Range	Hash	Composite (Range+Hash)	List
Adaptive Server Enterprise	Не	Да	Не	Не
ADS	Да	Да	Да	Да
DB2	Да	Да	Да	Да
Firebird	Не	Не	Не	Не
Informix	Да	Да	?	?
Ingres	Да	Да	Да	Да
InterBase	Не	Не	Не	Не
MaxDB	Не	Не	Не	Не
Microsoft SQL Server	Да	Не	Не	Не
MySQL	Не	Не	Не	Не
Oracle	Да	Да	Да	Да
PostgreSQL	Да ¹	Да ¹	Да ¹	Не
Pyrrho DBMS	Не	Не	Не	Не
SQLite	Не	Не	Не	Не
TeraData	Да	Да	Да	Да

Табела 2.11. Подржане шеме за партиционисање базе података.

напомена (1): PostgreSQL 8.1 обезбеђује подршку за партиционисање кроз ограничења (*check constraints*). Range, List и Hash методе се могу емулирати помоћу PL/pgSQL или неког другог процедуралног језика.

Подршка за рад са XML документима

Прихватањем XML формата као основног стандарда за рад са електронским документима, појавило се питање подршке коју системи за управљање базама података нуде за рад са овим форматом. Одговор на ово питање се тражи на два различита начина: уградњом одређеног вида подршке за процесирање XML докумената у стандардне релационе базе података, и развојем тзв. природних XML база података. Базе података које поседују уграђену подршку за XML, поседују екстензије које омогућавају трансформацију података из XML формата у њихове интерне структуре података. Овакве базе користе одговарајуће шеме помоћу којих се дефинише мапирање. Природне XML базе података користе генеричке структуре и могу садржавати потпуно произвољне XML документе у изворном облику.

Системи за управљање базама података са уграђеном подршком за XML

Базе података са уграђеном подршком за XML имају следеће особине:

- обезбеђују механизам за мапирање XML модела података у традиционалне моделе
- ова конверзија не мора гарантовати очување редоследа којим су се подаци јављали у XML документу
- подржавају упите који садрже услове сачињене од различитих погледа на XML податке и резултат враћају у XML облику

Различити произвођачи наведене особине постижу коришћењем различитих приступа. У табели 2.12 дат је делимични списак база података које имају уграђену подршку за XML.

Назив	Произвођач	Тип лиценце	Тип базе података
Access 2002	Microsoft	Комерцијална	Релациона
Cache	InterSystems Corp.	Комерцијална	Пост-Релациона
DB2	IBM	Комерцијална	Релациона
eXtremeDB	McObject	Комерцијална	Навигациона
FileMaker	FileMaker	Комерцијална	FileMaker
FoxPro	Microsoft	Комерцијална	Релациона
Informix	IBM	Комерцијална	Релациона
Matisse	Matisse Software	Комерцијална	Објектна
Objectivity/DB	Objectivity	Комерцијална	Објектна
OpenInsight	Revelation Software	Комерцијална	Multi-valued
Oracle	Oracle	Комерцијална	Релациона
PostgreSQL	PostgreSQL Global Development Group	Open Source	Релациона
SQL Server 2000	Microsoft	Комерцијална	Релациона
Sybase Adaptive Server Enterprise 12.5	Sybase	Комерцијална	Релациона
UniData	IBM	Комерцијална	Угнеждена Релациона
UniVerse	IBM	Комерцијална	Угнеждена Релациона
View500	eB2Bcom	Комерцијална	Proprietary (LDAP)

Табела 2.12. Базе података са уграђеном подршком за рад са XML-ом.

– <http://www.rpbouret.com/xml/XMLDatabaseProds.htm#native>

У табели 2.12 су дате верзије RDBMS система у којима је први пут примењена нека врста подршке за рад са XML подацима. Новије верзије су или задржале или прошириле ову подршку. Од осталих база које се помињу у табелама 2.5 – 2.11 Adabas омогућава враћање резултата упита у XML формату, као и коришћење XQuery упита, Firebird не нуди никакве посебне опције, осим што је путем екстерних процедура могуће креирати механизам за конверзију XML докумената у интерне структуре података, Interbase омогућава експортовање података у XML формату помоћу посебног модула, MySQL такође омогућава да се добије XML форматирани излаз, а од верзије 5.1.5 уведене су и две посебне функције које подржавају основне XPath функционалности, а помоћу којих је могуће вршити селектовање делова XML-а који се начешће чувају у TEXT пољима. MaxDB не нуди посебну подршку за рад са XML-ом. Pyrho DBMS је SQL 2003 компатибилан и подржава како XML тип података тако и функције за рад са тим подацима. У наставку је дат преглед кључних механизма који омогућавају рад са XML документима у базама приказаним у табели 2.12. Подаци су преузети са [XMLEnabled_DB]. Подаци који су додатно ажурирани у пројектном тексту су дати као напомена.

Access 2002

Подаци задњи пут ажурирани: фебруар, 2002

Access трансформише податке из/у XML документе користећи механизам мапирања на основу табела. Индивидуални подаци морају бити у елементима (атрибути се игноришу), а називи табела/колона морају се подударати са називима елемената.

Cache

Подаци задњи пут ажурирани: март, 2004

Cache је "пост-релациона" база података која податке чува у мултидимензионалним низовима.

Cache подржава рад са XML-ом на три начина:

- обезбеђује повезивање XML података (XML data binding), који користи објектно оријентисани поглед на податке. Омогућава креирање DTD-ова и XML Schema за класе које се чувају у бази као и за креирање класа из XML Schema. На овај начин могуће је накнадно вршити трансформацију података из XML-a у објекте који се чувају у бази.
- методе било које класе која се чува у бази, могу се објавити као методе Web сервиса
- Cache Web апликациони сервер (Cache Server Pages) може податке експортирати у XML формату.

DB2

Подаци задњи пут ажурирани: октобар, 2005

Нове верзије DB2 (октобар, 2005) подржавају рад са XML у основном DB2 производу, у Net Search Extender-у, као и у Web services framework-у (DB2 WOrF). Подршка за XML у DB2 бази састоји се од омогућавања чувања XML у изворном облику, подршке за XQuery, функције за публикавање SQL/XML (XMLELEMENT, XMLATTRIBUTE, XMLFOREST, XMLAGG, etc.), у различитим SQL/XML функцијама (XMLVALIDATE, XMLCAST, etc.), као и механизму декомпозиције. Колоне чији је тип XML намењене су за чување XML докумената у изворном облику. XML садржаји се индексирају тако што се идентификују чворови који ће бити индексирани (коришћењем подскупа Xpath језика), и одређивањем да ли ће идентификовани чворови бити индексирани као стрингови, нумеричке вредности, датуми и сл.

eXtremeDB

Подаци задњи пут ажурирани: јун, 2005

eXtremeDB је објектно оријентисана база. Подржава рад са XML-ом помоћу повезивања XML података (XML data binding). На основу шеме базе података eXtremeDB може генерисати нове методе за креирање нових објеката из XML докумената, серијализацију постојећих објеката у XML документе, као и ажурирање постојећих објеката на основу XML докумената. Додатно, може генерисати и XML Schema-е за генерисане XML документе.

FileMaker

Подаци задњи пут ажурирани: фебруар, 2002

FileMaker омогућава корисницима да врате резултат упита као XML. резултати се могу вратити на два различита начина:

- називи поља у бази се користе као називи елемената у резултујућем XML документу.
- генерички приступ при којем се прво враћа скуп метаподатака, а затим подаци у елементима <ROW>, <COL>, и <DATA>.

FoxPro

Подаци задњи пут ажурирани: април, 2002

Visual FoxPro поседује три метода за трансфер података из XML докумената у табеле:

CURSORTOXML, XMLTOCURSOR, и XMLUPDATEGRAM. CURSORTOXML и XMLTOCURSOR трансформишу податке из FoxPro курсора или табеле у XML документ (и обратно) користећи табеле за мапирање. Подаци у колонама табела могу бити приказани било као атрибути или као угнеждени елементи у XML документу. Уколико постоји XML Schema, FoxPro одређује мапирање на основу ње, а уколико не постоји врши се двопролазни поступак, у првом пролазу се одређује структура документа и конструише мапирање, док се у другом пролазу врши само пресликавање података. XMLUPDATEGRAM генерише XML документ измена (XML updategram) на основу табела и курсора у тренутној сесији, која је имала баферовање промена. Овај XML документ садржи податке који детаљно описују стање табела и курсора пре и после ажурирања.

Informix

Подаци задњи пут ажурирани: новембар, 2000

Informix подржава XML помоћу Object Translator-a и Web DataBlade компоненте. Подршка за XML је обезбеђена генерисањем метода које обезбеђују трансформацију података из XML докумената у објекте и обрнуто. Web DataBlade је апликација која омогућава креирање XML докумената на основу предлогака (template) који садрже угнеждане SQL изразе и друге скриптне језике.

Matisse

Подаци задњи пут ажурирани: август, 2003

Matisse је објектна база података. Рад са XML је подржан путем повезивања XML података (XML data binding), помоћу којег се постиже мапирање XML докумената на објекте. Ово мапирање се ради помоћу алата који генерише класе на основу DTD-a.

Objectivity/DB

Подаци задњи пут ажурирани: фебруар, 2002

Objectivity XML Interface Tool подржава експортовање објеката из Objectivity/DB (Release 6) у XML формат и увозење XML форматираних података у Objectivity DB.

OpenInsight

Подаци задњи пут ажурирани: јануар, 2003

OpenInsight је развојно окружење које омогућава чување података у сопственом типу табела, као и у Oracle, Lotus Notes, ODBC и OLE/DB доступним базама података. Подршка за XML је обезбеђена за сопствени тип табела. Пресликавање података се врши на основу табела за мапирање.

Oracle

Подаци задњи пут ажурирани: новембар, 2005

Oracle обезбеђује подршку за рад са XML документима помоћу више концепата: XML тип података, SQL/XML, XQuery, XSLT, DOM, XML индекси, XML DB Repository, и XML Schemas.

XMLType је посебан тип података који имплементира тип података дефинисан помоћу SQL/XML. Подаци у колонама типа XMLType могу бити чувани на два начина: помоћу објектно-релационог похрањивања података и као CLOB. Апликације које раде са подацима у овим колонама су инваријантне без обзира који тип чувања података је у питању.

Колонама типа XMLType може се задати и додатно ограничење да морају садржати податке валидне у односу на одређену XML Schema-у. Ово не само да ограничава документе који се могу сместити у одређену колону, него и омогућава механизму за претраживање да оптимизује претрагу на основу података у датим шемама.

Индексирање колона типа XMLType може се вршити на 4 начина:

1. Када се користи објектно-релациони модел смештања података, колоне на које се мапирају XML елементи и атрибути могу бити индексирани конвенционалним Б стаблом.
2. Индекси типа XMLIndex могу се конструисати над било којом XML вредношћу, без обзира на начин њиховог смештања.
3. Индекси на бази функција такође се могу користити над било којим XML вредностима. Ови индекси користе интерне функције као што је `extractValue` како би идентификовале вредности у XML документу које ће бити индексирани и евалуирају се на основу XPath израза.
4. могу се користити и Oracle Text индекси

XML вредности могу се претраживати коришћењем XQuery језика, који се позива из SQL израза помоћу SQL/XML XMLQUERY функције.

Релациони подаци могу се користити за конструисање XML вредности путем SQL/XML функција (XMLELEMENT, XMLATTRIBUTE, XMLFOREST, XMLAGG, итд.). Такође, могуће је над њима извршити и XQuery, тако што се прво конструише XML поглед ових података, а затим се XQuery упит извршава над овим конструктом. Овакви погледи се могу конструисати помоћу SQL/XML функција, помоћу XML Schema-е, или помоћу интерне `ora:view` функције.

Oracle садржи две имплементације XQuery: имплементацију која је део средњег слоја, или имплементацију која је интерни део система за управљање базом података. Прва имплементација се користи за упите над индивидуалним XML вредностима као што су XML поруке, док је интерна имплементација (у RDBMS систему) оптимизована за упите над вредностима у бази података. Ова имплементација све XPath и XQuery изразе покушава да представи као SQL изразе. Ово је могуће само када се XML вредност чува у објектно-релационом моделу, или ако се израз може разрешити на основу индекса. Уколико ниједан од ових услова није испуњен, израз се обрађује у изворном облику.

Oracle такође садржи и XML DB Repository, који обезбеђује поглед на објекте у бази који је врло сличан функционисању система датотека. (Иако је развијен специјално за приступ XML вредностима, XML DB Repository може се користити за приступ било објекту у бази). Објектима се додељује путања и кореспондирајући URL и могуће им је

приступити преко HTTP, WebDAV, и FTP, као и преко JDBC, PL/SQL, OCI, и ODP.NET.

PostgreSQL

Подаци задњи пут ажурирани: октобар, 2005

PostgreSQL директно не подржава XML. Доступан је додатак који подржава SQL/XML функције (XMLELEMENT, XMLFOREST, etc.)

SQL Server 2000

Подаци задњи пут ажурирани: новембар, 2000

Microsoft SQL Server 2000 подржава рад са XML-ом на три начина:

- помоћу FOR XML клаузуле у SELECT изразима,
- XPath упитима који користе XML-Data Reduced schema-е,
- OpenXML у интерним процедурама.

Напомена: Од верзије MS SQL Server 2005 подршка за XML је доживела знатне измене:

- уведен је посебан тип података xml, посебно намењен за рад са XML подацима,
- уведена је подршка за XQuery. Сада је могуће поставити XQuery упит над XML подацима смештеним у колонама чији је тип **xml**,
- функционалност OPENROWSET команде је побољшана како би се омогућило читавање целих XML докумената,
- побољшана функционалност FOR XML клаузуле и OPENXML функције (уведених у SQL Server 2000),
- омогућена је употреба **xml** типа и у погледима, T-SQL изразима, и као параметара снимљених процедура

Нека од ограничења **xml** типа:

- XML тип се не може конвертовати у тип text или ntext,
- Само типови података који представљају једну од верзија стринг типа се могу конвертовати у XML,
- колоне XML типа не смеју се користити у GROUP BY изразима.
- дистрибуирани партиционисани погледи, као и материјализовани погледи не смеју садржати колоне типа XML,
- колоне XML типа не могу бити део примарног или страног кључа,
- колоне XML типа се не могу декларисати као **unique**.
- COLLATE клаузула не може се користити над колонама XML типа.
- колоне XML типа не могу учествовати у креирању правила (*rules*).
- једине уграђене скаларне функције које се могу применити на колоне XML типа су ISNULL и COALESCE.
- табела може садржати највише 32 колоне типа XML.
- табеле које садрже колоне XML типа не могу имати примарни кључ који сачињава више од 15 колона,
- табеле са колонама XML типа не могу садржати колоне типа **timestamp** као део свог примарног кључа,
- XML документи који се чувају на овакав начин у бази могу садржавати хијерархију дубине максимално 128

Sybase ASE 12.5

Подаци задњи пут ажурирани: јули, 2002

Sybase подржава рад са XML подацима на два начина:

1. ResultSetXml класа може се користити за пренос података у/из базе и XML документа. ResultSetXml објекат може се креирати на основу XML документа или SELECT израза. Осим тога, апликације могу модификовати податке у ResultSetXml објекту, серијализовати објекат у XML, или креирати SQL скрипт за креирање табеле и снимање података у базу. XML документ који се користи за ResultSetXml је посебног формата, тако да садржи и скуп ColumnMetaData елемената, након којих следе Row и Column елементи.
2. Sybase такође име и уграђене XML могућности. XML документи се могу чувати у препарсираном, индексираном облику у BLOB колонама.

Напомена: верзија 15.0 увела је унапређења по питању подршке за рад са XML-ом.

Сада је подржано чување, претраживање и процесирање XML података у изворном облику. Sybase има специфичан дуалан приступ SQL/XML који омогућава:

- генерисање XML резултата и једноставне SQL-у-XML трансформације,
- XML/SQL мапирање како би се XML "исекао" у податке организоване по релационом моделу
- Storage/Retrieval – омогућава чување и преузимање XML података у оригиналној форми (без потребе за трансформацијом), али и чување и претраживање у релационом моделу (редови / колоне),
- индексирање – брзи, аутоматизовани XML индекси за боље перформансе упита,
- међусобну измењивост SQL/XML упита, подршку за XPath и SQLX,
- способност "гранулисања резултата" путем упита и резултујућих докумената, фрагмената докумената, елемената итд,
- интеграција/дељење података – могућност приказивања података директно путем web сервиса и интеграција тих података у/из еџтерних web сервиса,

UniData

Подаци задњи пут ажурирани: април, 2004

UniData је угнеждена релациона база података, тј. релациона база података која не следи прву нормалну форму (дозвољава да једна колона садржи више од једне вредности по једном реду – ефективно се ово остварује помоћним табелама).

Постоје три опције за мапирање података у XML документ:

- атрибутски оријентисано мапирање – подаци из колоне са једноструком вредношћу се мапирају на атрибуте елемента који представља један ред података из базе у XML документу, колоне са вишеструким вредностима се мапирају на атрибуте елемената који представљају помоћну табелу,
- мапирање оријентисано на елементе – подаци из колоне се чувају у елементима XML документа. У оба ова случаја називи атрибута/елемената представљају називе табела, колоне, асоцијација.
- последња опција је тзв. "мешано" мапирање, при којем корисник специфицира да ли се неки подаци мапирају на атрибуте или на елементе. У овом случају корисник може и да дефинише називе атрибута/елемената.

Апликације наглашавају да желе резултат у виду XML документа тако што се наводи

TOXML клаузула у LIST команди у UniQuery или SELECT изразу у UniData SQL. Могућа је и употреба XPath израза.

UniVerse

Подаци задњи пут ажурирани: април, 2004

Ова база има у суштини исти скуп особина како и претхидна, али подржава и XML базиран језик за објектно-релационо мапирање.

View500

Подаци задњи пут ажурирани: јануар, 2006

View500 је LDAP директоријум базиран на посебном (*proprietary*) механизму за чување података. Подршка за рад са XML-ом обезбеђена је на основу XED (XML Enabled Directory) спецификације, која је тренутно (Jan. 2006) IETF предлог стандарда. Основне особине за подршку XML:

- XED дефинише XML верзију LDAP протокола (XLDAP).
- XED дефинише XML кодирање за ASN.1 вредности, Robust XML Encoding Rules (RXER).
- XED омогућава корисницима да дефинишу типове података за атрибуте директоријума помоћу DTD-a, XML Schema, итд.

Природне XML базе података

Под природном (native) XML базом података се сматра она која:

- дефинише логички модел за XML документе, насупрот подацима који се смештају у те документе, и која чува и претражује податке по тако дефинисаном моделу. Минимално, модел мора подржавати елементе, атрибуте, PCDATA, и уређење докумената. Примери оваквих модела су XPath модел података, XML Infoset, као и модел DOM и модел имплициран догађајима у SAX 1.0.
- користи XML формат као основну јединицу чувања података, баш као што је то релација у релационом моделу
- Не захтева (не имплицира) ниједан посебан модел физичке организације похрањених података. Нпр, може бити изграђена над релационом, хијерархијском или објектно оријентисаном базом података
- Не користи неке од заштићених формата записа нпр. одређене врсте индексних, компримованих датотека

Природне XML базе могу се грубо сврстати у две категорије:

- Базе које чувају целе XML документе – Похрањују читав документ у текстуалној или бинарној форми и обезбеђују одговарајуће функције за приступ документу. Једноставан приступ је похрањивање докумената у BLOB поље у релационим базама података, или снимање документа у одговарајуће директоријуме, а затим изградња XML компатибилних индекса над документом. Софистициранија опција је похрањивање XML докумената у посебан тип који нуди подршку за индексирање, трансакције и сл.
- базе које чувају чворове XML докумената – код оваквих база се у постојећим или посебним типовима података чувају појединачни чворови XML докумената. Нпр, овакав принцип похрањивања података може се постићи мапирањем DOM

елемената у табеле под називима Elements, Elements, Attributes, Entities, или се обезбедити чување DOM у парсираном облику у посебно изграђеном репозиторију.

Овакве “природне” XML базе су тренутно у фокусу разних покушаја унапређења, али је још увек њихова употреба мања него употреба релационих база.

Табела 2.13. даје преглед "природних" XML база података.

Назив	Произвођач	Тип лиценце	Тип базе података
4Suite, 4Suite Server	FourThought	Open Source	Објектно оријентисана
Berkeley DB XML	Sleepycat Software	Open Source	Кључ-вредност
Birdstep RDM XML	Birdstep	Комерцијална	Објектно оријентисана
Centor Interaction Server	Centor Software Corp.	Комерцијална	Посебан тип (proprietary)
DBDOM	K. Ari Krupnikov	Open Source	Релациона
dbXML	dbXML Group	Open Source	Заштићен
DOMSafeXML	Ellipsis	Комерцијална	базирана на систему датотека(?)
eXist	Wolfgang Meier	Open Source	Релациона
eXtc	M/Gateway Developments Ltd.	Комерцијална	Пост-Релациона
Extraway	3D Informatica	Комерцијална	Files plus indexes
GoXML DB	XML Global	Комерцијална	Посебан тип (proprietary) (Model-based)
Infonyte DB	Infonyte	Комерцијална	Посебан тип (proprietary) (Model-based)
Ipedo	Ipedo	Комерцијална	Заштићен
Lore	Stanford University	Развојна	Делимично структурирана
MarkLogic Server	Mark Logic Corp.	Комерцијална	Посебан тип (proprietary) (?)
myXMLDB	Mladen Adamovic	Open Source	MySQL
Natix	data ex machina	Комерцијална	базирана на систему датотека(?)
NaX Base	Naxoft	Комерцијална	Посебан тип (proprietary)
Neocore XMS	Xpiori	Комерцијална	Посебан тип (proprietary)
ozone	ozone-db.org	Open Source	Објектно оријентисана
Sedna XML DBMS	ISP RAS MODIS	Бесплатна	Посебан тип (proprietary)
Sekaiju / Yggdrasill	Media Fusion	Комерцијална	Посебан тип (proprietary)
SQL/XML-IMDB	QuiLogic	Комерцијална	Proprietary (природна XML и Релациона)
Sonic XML Server	Sonic Software	Комерцијална	Објектно оријентисана (ObjectStore). Релациони и други модели организације податка путем Data Junction
Tamino	Software AG	Комерцијална	Заштићен. Релациона преко ODBC.
TeraText DBS	TeraText Solutions	Комерцијална	Посебан тип (proprietary)
TEXTML Server	IXIASOFT, Inc.	Комерцијална	Посебан тип (proprietary) (Text-based)

TigerLogic XDMS	Raining Data	Комерцијална	Pick
Timber	University of Michigan	Open Source (само за некомерцијалну употребу)	Shore, Berkeley DB
TOTAL XML	Cincom	Комерцијална	Објектно-Релациона?
Virtuoso	OpenLink Software	Комерцијална	Посебан тип (proprietary). Релациона путем ODBC
XDBM	Matthew Parry, Paul Sokolovsky	Open Source	Посебан тип (proprietary) (Model-based)
XDB	ZVON.org	Open Source	Релациона
XediX TeraSolution	AM2 Systems	Комерцијална	Посебан тип (proprietary)
X-Hive/DB	X-Hive Corporation	Комерцијална	Посебан тип (proprietary). Релациона путем JDBC
Xindice	Apache Software Foundation	Open Source	Посебан тип (proprietary) (Model-based)
XML Transactional DOM	Ontonet	Комерцијална	Објектно оријентисана
XpSQL	Makoto Yui	Open Source	Релациона (PostgreSQL)
XQuantum XML Database Server	Cognetic Systems	Комерцијална	Посебан тип (proprietary)
XStreamDB Native XML Database	Bluestream Database Software Corp.	Комерцијална	Посебан тип (proprietary) (Model-based)
Xyleme Zone Server	Xyleme SA	Комерцијална	Посебан тип (proprietary)

Табела 2.13. Природне XML базе података.
– извор <http://www.rpbouret.com/xml/XMLDatabaseProds.htm#native>

2.4. Апликациони сервери и CMS системи

Апликациони сервери су стандардна компонента било ког модерног информационог система, јер се на њима извршавају све операције у средњем слоју (тростројне архитектуре). Избор апликационог сервера зависи од великог броја фактора, као што су очекивано оптерећење сервера (број корисника које сервер треба да опслужи у одређеном временском интервалу), интензивност апликација у погледу захтева за приступ слоју података, захтевани интерфејси за размену података са другим апликацијама и сл. Као врло важно питање поставља се избор технологије за коју (и најчешће у којој) је одређени апликациони сервер развијен. Данас на овом пољу у суштини доминирају само две технологије: J2EE (базирана на Java програмском језику) и Microsoft .NET технологија. Врло често се одређене апликације из домена еУправе реализују коришћењем PHP скриптног језика. Иако има многе погодности (врло широку базу програмера, огроман број доступних модула са најразличитијим функцијама, велики број развијених апликација), овај језик је стриктно интерпретерски, подршка за објектно оријентисано програмирање не следи кључне концепте, језик је изразито слабо типизиран, не постоји подршка за мултитаскинг. Због ових слабости, решења рађена на овај начин нису препоручљива за велике дистрибуиране системе. Овакав приступ може бити прихватљив у мањим окружењима и локалним срединама из више разлога (скромна новчана средства за развој апликација, непостојање стручног кадра за рад у другим окружењима и сл.). Неке од локалних управа већ имају развијене системе базиране на PHP/MySQL платформи. Овакве апликације не треба у старту одбацити и занемарити, већ примарни циљ треба бити њихово усаглашавање и интероперабилност са апликацијама писаним у другим језицима. За апликације намењене за еВојводину, кључно питање је да ли изабрана технологија подржава *Web* сервисе, обраду XML докумената, и да ли омогућавају да се кориснички интерфејс имплементира путем *web* оријентисаних технологија (што омогућава корисницима да као клијентске апликације користе само произвољни *browser*-и). Обе наведене технологије имају потпуна решења за све наведене захтеве.

CMS системи су апликације, које се најчешће извршавају на апликационим серверима, а намењене су за креирање, обраду, публикување и архивирање дигиталног садржаја.

■ Апликациони сервери

У тростројном моделу апликација, апликациони сервери су програми који се извршавају у средњем слоју. Типични задаци апликационих сервера јесу:

- извршавање пословне логике,
- припрема и форматирање излаза према корисницима (најчешће је овај излаз у форми динамички генерисаних *web* страница, којима клијенти приступају преко својих *web browser*а),
- процесирање захтева и обрада података који долазе из форми које попуњавају корисници,
- управљање сигурносним системом апликације (ауторизација и аутентификација, контрола права приступа и слично),
- приступ слоју базе података,

- развојем web сервиса, апликациони сервери су добили и компоненту која омогућава директну комуникацију две или више апликација преко стандардизованог интерфејса.

Апликациони сервери су се временом развили у платформу за обављање електронског пословања (*e-bussiness*), а играју централну улогу у омогућавању развоја дистрибуираних апликација у *web* окружењу. Апликациони сервери се граде на основу стандардизованих протокола и API-ја за развој апликација. Битна особина која је омогућила брзу експанзију и прихватање апликационих сервера као стандардног *middleware-a* је и њихова особина да је могуће њихово повезивање, преко стандардизованог интерфејса, са постојећим базама података, чиме је транзиција са концепта клијент-сервер била олакшана.

Најчешће се термин апликациони сервери повезује са серверима базираним на Java технологији (тачније на J2EE технологији). Са развојем .NET платформе Microsoft Internet Information Services (IIS), скуп сервиса уграђених у NT базиране Windows платформе, је такође добио и улогу апликационог сервера. Важно је напоменути да Microsoft IIS није сервер него скуп сервиса који се покрећу под серверским верзијама оперативног система Windows. Две технологије (J2EE и .NET) су данас и преовлађујуће у сегменту апликативних сервера. Обе технологије обезбеђују пуну подршку и добро развијене API-је за све стандарде везане за Web сервисе (XML, WSDL, UDDI, SOAP), као и за развој портала.

Као и у другим сегментима софтверске индустрије и на пољу апликационих сервера постоје лиценцна решења у власништву компанија које су развиле дате сервере и сервери базирани на *open source* решењима. На пољу J2EE апликационих сервера ова понуда је врло велика, од сервера погодних за развојна окружења и мале и средње системе, до сервера погодних за велике дистрибуиране системе, са најразличитијим ценовним групама, од *freeware* софтвера до сервера који коштају по неколико десетина хиљада долара по серверској инсталацији. На страни .NET платформе понуда је мања. Поред Microsoft IIS-а, постоји и *open source* решење Base 4 Application Server. .NET платформа је у суштини везана за коришћење оперативног система Windows (иако постоји и пројекат развоја .NET платформе за Linux), што са J2EE технологијом није случај. J2EE апликационе сервере могуће је покренути на различитим ОС.

Основне особине неких често коришћених апликационих сервера базираних на J2EE технологији дате су у табели 2.15. Слична табела за .NET технологију није дата, јер осим Microsoft IIS-а, не постоје озбиљнији сервери који користе .NET платформу, јер сама спецификација платформе већим делом није отворена.

Као што је већ напоменуто, обе технологије, и J2EE и .NET, пружају пуну подршку за кључне концепте свих модерних web апликација, као што су руковање XML документима, подршка за Web сервисе и портале.

Web сервис (по дефиницији датој од стране W3C конзорцијума) представља софтверски систем дизајниран са циљем да подржи машинску интероперабилност између апликација преко мреже (и Интернета). Оваква апликација поседује интерфејс описан помоћу документа у WSDL (Web Services Description Language) формату. Овакав документ је машински разумљив и омогућава аутоматизовано генерисање клијентске апликације која ће користити Web сервис. Интеракцију са Web сервисом, друге апликације обављају разменом порука чији формат је специфициран такође на основу WSDL спецификације. Ове поруке се најчешће смештају у SOAP поруку -

омотач. Најчешћи начин преноса је помоћу HTTP протокола. Поруче су по правилу у XML формату, уз могуће постојање и делова поруке у другим форматима.

Web сервиси постали су кључни аспект развоја дистрибуираних апликација, које се покрећу на различитим платформама, при чему клијентске апликације и сам Web сервис могу бити реализовани коришћењем различитих технологија. Java у оквиру своје J2EE платформе поседује све неопходне компоненте које омогућавају лаку имплементацију Web сервиса. Истоветна ситуација је и са .NET окружењем, које такође садржи врло добро разрађен *framework* за рад са Web сервисима, што је и природно јер је .NET концепт од самог почетка био изграђен око концепта Web сервиса.

Web портали су Интернет локације које омогућавају персонализован приступ информацијама. Дизајнирају се тако да користе дистрибуиране апликације, путем којих се обезбеђују сервиси са различитим садржајима, најчешће из више различитих извора. Пословни портали, као специјализација овог приступа, најчешће се дизајнирају тако да омогућавају дељење докумената и сарадњу у оквиру радних група.

▪ .NET и J2EE платформе – основни концепти

.NET платформа састоји се од следећих кључних компоненти:

- **C#**, "нови" објектно оријентисани програмски језик помоћу којег се програмирају класе и компоненте будућих апликација. C# интегрише неке елементе језика C, C++ и Java, и додаје неке нове особине као што су подршка за метатагове, везане за развој компоненти. Поред C#, у .NET окружењу могуће је користити и друге програмске језике (VisualBasic .NET, C++).
- **"common language runtime"** – окружење за извршавање програма преведених на тзв. заједнички интерни језик Internal Language (IL) (пуни назив у новијим верзијама је Microsoft Intermediate Language - MSIL), које извршава бајт код у интерном језику. Постоји могућност да се код писан у било ком језику користи у оквиру .NET платформе, уколико је развијен IL компајлер за дати језик.
- Скуп основних компоненти (**base components**), који је доступан из извршног окружења. Ове компоненте обезбеђују различите функције, као што су подршка за рад са мрежним подсистемом, контејнери и сл.
- **ASP+**, нова верзија ASP-а која подржава могућност компилације ASP кода у интерни језик (IL). Ова особина отвара могућност да се као скриптни језик у ASP страницама користи произвољни програмски језик који има развијен .NET IL компајлер.
- **Win Forms** и **Web Forms**, нове улазно-излазне компоненте, доступне из Visual Studio (Microsoft-ово интегрисано развојно окружење).
- **ADO+**, нова генерација ADO технологије за приступ базама података, која садржи и компоненте које користе XML и SOAP за размену података.

J2EE платформа је програмска платформа базирана на Java програмском језику намењена развоју и извршавању дистрибуираних вишеслојних апликација, базираних најчешће на модуларним софтверским компонентама, које се извршавају под окриљем J2EE компатибилних апликационих сервера. J2EE платформа је дефинисана спецификацијом *Java Enterprise Edition*, и сматра се стандардом, јер произвођачи софтвера морају испоштовати одређене захтеве како би могли прогласити производ

Java EE компатибилним. Ипак овај стандард не постоји као ISO или ECMA дефинисан формални стандард. Основни концепти J2EE платформе су:

- Употреба Java програмског језика.
- **EJB** – Enterprise Java Beans API – дефинише скуп API-ја које контејнер за дистрибуиране објекте треба да подржи како би обезбедио перзистенцију, могућност удаљеног позивања процедура (Remote Procedure Call – коришћењем RMI или RMI-IIOP), контролу конкурентности и приступа дистрибуираним објектима.
- **Servlet API** је скуп API-ја за web контејнер (који извршава део апликације доступан преко стандардног web-а). Сервлети су одговорни за процесирање корисничких захтева и испоруку одговора (JSP стране се такође компајлирају у сервлете).
- **Java Naming and Directory Interface (JNDI) API** – омогућава клијенту да пронађе и приступи подацима и објектима путем њиховог имена. Као и сви Java API-ји који комуницирају са сервисима *host* система, и овај API је независан од начина њихове имплементације на *host* систему.
- **Java Database Connectivity (JDBC) API** – дефинише генерализовани скуп операција чијим коришћењем клијентска апликација може приступити бази података. Обезбеђује методе за прикупљање и измену података у бази. Усмерен је на рад са релационим базама података. Данас постоје JDBC компоненте (*driver*-и) за повезивање на све доступне (комерцијалне и/или *open source* релационе базе података), при чему су неки од њих такође доступни као *open source* пројекти.
- **Java Transaction API (JTA)** омогућава да се дистрибуиране трансакције обављају над више XA ресурса. (XA стандард је X/Open спецификација за обраду дистрибуираних трансакција - DTP).
- **Java API for XML Processing** – скуп API-ја за процесирање докумената у XML формату.
- **Java Message Service (JMS) API** - дефинише слање порука између два или више клијента код апликација које користе Java Message Oriented Middleware (MOM).

Поређење .NET и J2EE технологија

Анализа дата у овом поглављу је објављена као чланак аутора *Jima Farley-a* на Интернет сајту O'Reilly Network компаније O'Reilly Media (чланак се може наћи на адреси: [Farley]). Овај чланак представља једну од првих анализа сличности/разлика између .NET и J2EE технологија, као и утицаја .NET-а. Чланак је објављен 2000/01 године, и доста често је репродукован. Додатна запажања су дата као напомене.

.NET платформа се састоји од више обједињених технологија које Microsoft представља као алтернативу Java или CORBA технологијама. У табели 2.14. дат је сажет преглед упоредних карактеристика .NET и J2EE платформи.

Microsoft .NET	J2EE	Кључне разлике
Програмски језици C#, VisualBasic .NET	Програмски језик Java	И C# и Java су језици настали на основама C/C++ језика. Најбитније особине (управљање меморијом помоћу тзв. garbage collection, и хијерархијски именски простор за компоненте) су присутни у оба језика. C# је у концепту компоненти сличан JavaBeans-има (<i>properties/attributes, events</i>), али има и неке додатне концепте (као што су тагови за метаподатке). Синтакса језика се, наравно, разликује. Java се извршава на било којој платформи која има инсталирану Java виртуелну машину. .NET ће се и у догледној будућности моћи покретати само на Windows платформама. C# је имплицитно укључен у окружење за извршавање програма преведених у IL, и извршава се као <i>just-in-time</i> (JIT) компајлирани бајт код, а може се и у потпуности компајлирати до машинског кода. Java код се извршава као бајткод на Java Virtual Machine (JVM) који се или интерпретира или може бити <i>just-in-time</i> компајлиран код (Sun Java HotSpot ради у овом режиму). Постоје компајлери који омогућавају и да се Java код компајлира до машинског кода - губи се независност од хардверске платформе.
.NET заједничке компоненте (познате и као .NET Framework SDK)	Java core API	.NET компоненте високог нивоа укључују и подршку за дистрибуирану обраду података коришћењем XML и SOAP технологија (погледати ADO+).
Active Server Pages+ (ASP+)	Java ServerPages (JSP)	ASP+ омогућава коришћење Visual Basic, C#, и евентуално других програмских језика као скриптог језика. ASP+ се компајлира до извршног кода посредством окружења за извршавање програма у MSIL. Ово је промена у односу на класични ASP који је при сваком приступу био интерпретиран. JSP као скриптни језик користи Javi, компајлира се у Java бајткод, било при првом захтеву за JSP страном (<i>on demand</i>) или се прекомпајлира пре активирања апликације.
IL Common Language Runtime	Java Virtual Machine and CORBA IDL and ORB	.NET окружење за извршавање кода у MSIL језику омогућава програмима писаним на више програмских језика да користе заједнички скуп компоненти, на Windows платформама. Ово окружење је основа .NET платформе. Java Virtual Machine омогућава извршавање Java бајткода на било којој платформи са одговарајућом виртуелном машином. CORBA је још једна технологија која омогућава програмима писаним на различитим програмским језицима да користе заједнички сет објеката, на било којој платформи са ORB подршком. Подршка за CORBA-у постоји у Javi, али није чврсто интегрисана.
Win Forms и Web Forms	Java Swing	Сличне web компоненте (које би биле базиране на JSP) не постоје у стандардној Java платформи, мада постоје лицензни Java пакети који су доступни (најчешће у оквиру интегрисаних развојних окружења). Подршка за Java Swing постоји у многим IDE окружењима. Win Forms и Web Forms RAD развој је подржан кроз MS Visual Studio IDE – подршка од стране других IDE окружења још не постоји.
ADO+ and SOAP-based Web Services	JDBC, EJB, JMS and Java XML Libraries (XML4J, JAXP)	ADO+ је изграђен на претпоставци размене података у XML формату (између удаљених <i>Data</i> објеката и слојева вишеслојне апликације) на бази HTTP (тј. коришћење SOAP-a). .NET web сервиси у основи подразумевају SOAP модел размене порука. EJB, JDBC, и сл. остављају избор протокола за размену података корисницима, при чему размена функционише над HTTP, RMI/JRMP или IIOP.

Табела 2.14. Основне карактеристике .NET и J2EE платформе

Могућности: .NET и J2EE нуде у суштини исти или врло сличан скуп могућности, али се то остварује различитим технологијама.

Преносивост: Језгро .NET технологије ради само на Windows платформама, али барем теоретски подржава могућност развоја софтвера на много програмских језика (када се дефинише под/надскуп жељеног језика и када се развије компајлер који би дати језик преводио у MSIL). Такође употреба SOAP технологије у .NET окружењу омогућава компонентама развијаним на другим програмским платформама да размењују поруке са .NET компонентама. Док се одређени делови .NET технологије, као што су SOAP и припадајући протоколи за проналажење сервиса, доступни као јавне спецификације, спецификација основних компоненти .NET платформе (MSIL окружење за извршавање – *runtime environment*, интерни делови ASP+, WebForms и WinForms компоненте) није објављена, и Microsoft јесте и биће једини произвођач који ће испоручивати комплетно развојно и извршно окружење за .NET платформу. Иако је већ извршен одређени притисак од стране програмерске заједнице да се и ове спецификације објаве, то би било у супротности са стандардном праксом Microsoft-a, па није веома изгледно.

Са друге стране, J2EE је могуће покренути на било којој платформи са одговарајућом Java виртуелном машином и одговарајућим скупом сервиса за дату платформу (EJB контејнер, JMS сервиси и сл). Све спецификације које дефинишу целокупну архитектуру J2EE платформе су јавно доступне и јавно се евалуирају. Велики број произвођача нуди своје производе, развојна окружења и компоненте које су компатибилне са спецификацијом. J2EE је, ипак, за разлику од .NET-a, платформа која подржава само један програмски језик – Java. Позиви који долазе из, или иду ка, компонентама развијеним на другим програмским језицима су могући путем CORBA технологије, али овај део није кључни део J2EE платформе.

Напомена: *Анализа је рађена 2000. године, од тада је у стандардну Java спецификацију уграђена потпуна подршка за Web сервисе, XML и SOAP – тако да се сада размена порука са апликацијама писаним у другим програмским језицима првенствено обавља на овакав начин, чинећи комуникацију потпуно XML оријентисаном и транспарентном.*

Ова последња разматрања наглашавају основне разлике између .NET и J2EE технологија, и упућују на крајњи циљ Microsoft-ове промоције .NET-a: отварање за програмере који користе друге програмске језике, као и отварање приступа својим компонентама за друге компоненте, које нису развијене у .NET окружењу, путем XML и SOAP.

Омогућавајући интеракцију између компоненти развијаних у различитим програмским језицима, .NET привлачи програмере који користе Perl, Eiffel, Cobol, и друге језике, да развијају програме у Microsoft *sandbox*-у. Програмери који користе ове језике могу бити заинтересовани за овакво решење, јер су до сада били потпуно занемарени у борби за тржиште коју воде Microsoft/Sun/OpenSource заједница. Коришћењем XML-a и SOAP-a, уграђује одређену дозу отворености у своју платформу, недостатак чега је до сад био чест повод критика Microsoft-a.

Какав је утицај .NET приступа?

За програмере на Microsoft платформама:

.NET је за све који су и раније били посвећени развоју софтвера на Microsoft платформама корак напред. ASP+ је бољи од “обичног” ASP-a, ADO+ такође је бољи,

али је и различит у односу на ADO и DCOM, C# је напреднији у односу на C/C++. Иницијална верзија .NET-а је припремана за објављивање током 2001. године и од тада је *de facto* постала стандардно развојно окружење на Microsoft платформама.

Циљеви неких делова .NET платформе су доста амбициозни и питање колики је њихов реални утицај. Нпр. окружење за извршавање IL кода мора бити широко подржано да би донело веће користи широком кругу програмера. Сваки програмски језик који жели да користи заједничке компоненте .NET технологије доступне из окружења за извршавање IL, мора дефинисати надскуп или подскуп језика који се једнозначно мапира на/из .NET IL извршног окружења, и мора дефинисати конструкције које обезбеђују метаподатке које компоненте у .NET окружењу захтевају. Осим тога, морају се развити и компајлери (*други_језик-у-IL*, и *IL-у-други_језик*) како би се омогућило компајлирање свих програмских концепата (структура, објеката, компоненти) у бајткод компоненти IL, као и да генерише специфичне интерфејсе (за дати програмски језик) ка постојећим IL компонентама.

За овакав приступ постоји и историјски преседан. Различити приступи за инкорпорирање других програмских језика у Java окружење (JVM) су развијени током времена, као што су JPython, PERCobol, Tcl/Java, Eiffel-to-JVM систем. Осим, евентуално, JPython, ови алати никад нису били широко прихваћени, чак ни у програмерским заједницама које су посвећене датим језицима, иако су обећавали могућност писања кода за Java извршно окружење (додуше не за целокупно J2EE окружење) коришћењем преферираног програмског језика уместо Java. Највероватнији разлог слабе прихваћености оваквих приступа је неспремност програмера да се ухвате у коштац са проблемима које неминовно доноси још један међукорак у превођењу од изворног језика до циљног окружења – уколико је већ Java окружење циљно окружење за развој апликације, програмери ће се вероватно одлучити да науче Java као подразумевани језик за дато окружење. Врло вероватно ће слично бити искуство и са инкорпорирањем других програмских језика у .NET окружење – програмери ће се вероватно одлучити да науче C# и пишу програме и компоненте на тај начин, и да се обуче се за рад у .NET развојном окружењу.

Што се тиче перформанси, неопходно је нагласити да је .NET приступ базиран на комуникацији темељеној на SOAP приступу. SOAP у суштини представља размену XML докумената преко HTTP протокола. Сам HTTP протокол није дефинисан као протокол високих перформанси, а употреба XML-а имплицира и постојање нивоа апликације намењеног за парсирање докумената, које захтева доста процесорског времена. Ова комбинација може довести до смањења брзине обављања трансакција у односу на алтернативне методе комуникације. С друге стране, XML је веома богат, робустан језик, а употреба HTTP-а обезбеђује велику преносивост апликација и омогућава избегавање највећег броја проблема са *firewall*-има. Ипак, уколико је сама брзина обављања трансакција кључна за рад апликације треба имати у виду и друге опције за комуникацију међу апликацијама.

За програмере Java и *open source* заједнице:

.NET представља промену у стратегији Microsoft-ове пропагације својих производа. Иако се компанија и до сад доста добро борила са конкурентским производима, при томе доводећи у питање уобичајене замерке о стварној цени коришћења одређених решења, са .NET технологијом ушли су у директну конкуренцију са Java платформом и *open source* иницијативама, примењујући одређени

(лимитирани) вид отворености софтвера и покушајима да се директно одговори на потребе програмера – два поља на којима до сад, нису имали баш много успеха. Microsoft-ово окружење за извршавање IL, има барем једну озбиљну предност, иако је њена примењивост упитна, а то је тежња да се елиминише баријера програмског језика при покушају да се користи ова платформа. Java елиминише платформску баријеру, али како би се користио J2EE, неопходно је програме писати у Javi. .NET жели да омогући коришћење било ког језика (у старту је могуће коришћење бар VisualBasic-а и C#) за изградњу .NET апликација. Као што је већ раније речено, да ли ће овај приступ бити широко прихваћен је велико питање. Тек, овакав приступ указује на једну слабост J2EE технологије – везаност за само један програмски језик тј. Java. Колико је ово питање заиста од важности за програмере је врло упитно, али би га вредело размотрити. Уколико би овакав приступ био од великог интереса, онда би било неопходно консолидовати напоре за изградњу бајт код компајлера за друге језике.

***Напомена:** Да је могућност развоја програма на другим програмским језицима за Java окружење заиста била врло тражена особина, вероватно би у последњих неколико година (нарочито након појаве .NET платформе) овакав приступ био широко прихваћен и већ би се развили озбиљнији пројекти за бајт код компајлере за неке језике. С друге стране, коришћење само једног програмског језика – Java - у свим аспектима платформе, врло вероватно омогућава лакше одржавање и редовно унапређење компајлера, исправљање грешака и увођење нових компоненти и могућности у платформу, избегавајући при том могуће неусаглашености верзија за различите програмске језике.*

Концентришући се на J2EE, постоји неколико слабости којима је било неопходно посветити пажњу како би се платформа ојачала у оним пољима на које је циљао и .NET. Као прво, подршка за процесирање XML-а мора бити у потпуности интегрисана у Java окружење, тј. подршка за размену порука у XML формату, као и за SOAP. У J2EE је у последњих пар година укључено неколико пакета које нуде потпуна решења за обраду XML докумената, размену порука коришћењем SOAP-а и друге неопходне технологије које су изразито усмерене на коришћење XML технологија.

Произвођач Производ	Верзија	Датум објављи вања	верзија JDK	EJB	JSP	JMS	Лиценци ран за J2EE верзију	J2EE сертифи кат	Платформе на којима је могуће покретање сервера
Apache Tomcat	v5.x	Dec-01	1.3 1.4.1		2.0		✓	1.3	Било која на којој се минимално извршава JDK 1.1.8 JVM
Apple WebObjects	v5.x	Jan-02	1.4.2	1.1	2.0				Mac OS X Server, NT 4.0, Win2K, Solaris 2.6 and 2.7, HP- UX 11
ATG Dynamo	v6.1	Sept-03	1.3.1	2.0	1.2	1.0.2	✓	1.3	NT, Solaris 8,9, AIX, HP-UX 11i, MS Windows 2000 Server, Red Hat Enterprise Linux 2.1
BEA WebLogic	Express 8.1 Basic	Feb-03	1.4.1		1.2	1.0.2	✓	1.3	Win 2K, Solaris 8,9 Redhat Linux 2.1
	Express 8.1 Premium								
	Workgroup 8.1								
	Server 8.1 Advantage			2.0	1.0.2			Red Hat Linux AS 2.1 64 bit	
	Server 8.1 Premium								
	Server 8.1 Technology Preview for 64 bit								Windows2K/NT/XP, Solaris, AIX, HP-UX, True64, OpenVMS, Linux (Miracle, RedHat, SuSE), SCO UnixWare
Borland Enterprise Server	AppServer Edition 6.0	Jan-04	1.4.2	2.1	1.2	1.0.2	✓	1.3	Windows, Linux, Solaris, HP-UX, AIX
Caucho Resin	v3.0	Jun-02	1.1.x 1.2.x 1.3.x		2.0			1.4	NT, Win2K, Solaris, Linux (било која на којој се извршава JDK)

Табела 2.15. Преглед особина J2EE сервера
(табела представља скраћену верзију табеле доступне на <http://www.theserverside.com/reviews/matrix.jsp>)

Произвођач Производ	Верзија	Датум објављивања	верзија JDK	EJB	JSP	JMS	Лиценциран за J2EE верзију	J2EE серти фикат	Платформе на којима је могуће покретање сервера
IBM WebSphere	Express 5.0	Oct-02	IBM 1.3.1		1.2				NT, Win2K, Solaris,AIX, OS/400, HP-UX, Red Hat Linux, SuSE Linux, Turbo Linux, Linux/390, HetWare, OS/390
	Enterprise 5.0	Mar-03							
	WebSphere v5	Nov-02		2.0		1.0.2 +XA		1.3	
	WebSphere Hetwork Deployment v5								
	WebSphere Application Server v5 for z/OS	Apr-03						z/OS only	
Ironflare Orion	v2.0	Mar-03	1.4.1	2.0	1.2	1.0.2		1.3	NT, Solaris, Linux
JBoss JBoss	3.2	Sept-04	1.3.X, 1.4.X, 5.X	2.0, 3.0	1.3	1.1		1.4	NT, Win2k, OS/390, Solaris, OS/400, Unix, Netware, Linux, HP-UX, Tru64, (било која на којој се извршава JDK 1.3, 1.4, 5.X)
Macromedia JRun Server	Developer License v4	May-02	1.3.1	2.0	1.2	1.0.2		1.3+	NT, Win2K, WinXP, Solaris, Suse Linux, Red Hat Linux, HP-UX, Compaq Tru64, AIX
	Full License v4								
ObjectWeb JOnAS	v3.3.6	Oct-03	1.3	2.0	1.2	1.1		1.3	NT, Linux, Solaris, AIX, HP-UX, Win2K, Hetware
	v4.1	n/a						1.4	

Табела 2.15. Преглед особина J2EE сервера (наставак)

Произвођач Производ	Верзија	Датум објављивања	верзија JDK	EJB	JSP	JMS	Лиценциран за J2EE верзију	J2EE сертификат	Платформе на којима је могуће покретање сервера
Oracle Application Server 10g (includes Toplink)	Java Edition v9.0.4	Dec-03	1.4.1 1.4.2 (Linux only)	2.0	1.2	1.0.2b		1.3	Solaris, HP-UX, Redhat Linux, United Linux
	Standard Edition v9.0.4	Sep-03							Platforms to follow: Windows, AIX, HP Tru64, IBM S390 based Linux
	Enterprise Edition v9.0.4								Напомена: Верзија 9.0.3 је доступна за све набројане платформе
	OC4J Standalone v10.0.3 Developer Preview Release with J2EE 1.4 Support		1.4.1 1.4.2	2.1	2.0	Pure Java - runs on Unix, Linux, Windows			
SAP AG Web Application Server	v6.3	Apr-02	1.3.1	2.0	1.2	1.0.2		1.3	XP/NT/2000, Solaris, HP-UX, AIX, Linux
Sun Microsystems Sun Java System Application Server	Platform Edition 8.0	Mar-04	1.4	2.1	2.0	1.1		1.4	Solaris 8, 9 Sun Java Desktop System Red Hat Enterprise Linux 2.1, 3.0 Windows 2000, XP, 2003
	Standard Edition 7.0	Oct-02		2.0	1.2	1.0.2		1.3	Solaris 8, 9; WinXP, Win2K; Linux Red Hat 7.2, Sun Linux 5.0, SPARC 8,9
	Enterprise Edition 7.0	Sept-03							

Табела 2.15. Преглед особина J2EE сервера (наставак)

Произвођач Производ	Верзија	Датум објављивања	верзија JDK	EJB	JSP	JMS	Лиценциран за J2EE верзију	J2EE сертификат	Платформе на којима је могуће покретање сервера
Sybase EAServer	Developer v4.1.2	Sep-02	1.2.2 1.3.1	2.0	1.2	1.0.2		1.3	Windows, Solaris, HP-UX, AIX, Linux
	Small Bus. v4.1.2								
	Advanced v4.1.2								
	Enterprise v4.1.2								
	Enterprise v4.2								
	5.0 Beta	May-04	1.4	2.1	2.0	1.1		1.4	Windows
Together Teamlösungen EDV- Dienstleistungen GmbH. in cooperation with ObjectWeb Enhydra Server	5.2	Mar - 05	1.3		1.1				Windows NT, Windows 2000, Windows XP, Linux, Solaris
	v6.2		1.4						
	Enterprise v6.2		1.4	2.1	2.0	1.1			
	Enterprise v3.2.0 Dual CPU								

Табела 2.15. Преглед особина J2EE сервера (наставка)

■ CMS

CMS системи (*Content Management System*) су софтверски системи који омогућавају креирање, обраду, публиковање и архивирање дигиталног садржаја. Такви системи углавном омогућавају рад више лица на дигиталном садржају (*collaborative creation*). Обично су везани за *web* садржај и сами системи су обично реализовани као *web* апликације.

CMS системи нису *workflow management* системи, иако неки CMS системи у себе укључују и *workflow management* подсистеме. CMS системи заправо омогућују публиковање дигиталних садржаја кроз стандардизован кориснички интерфејс и на конзистентан начин, а све то за више корисника.

CMS системи су почели да се развијају у компанијама које се и иначе баве публиковањем разних садржаја у електронском облику. Временом се развио велики број CMS система, који се могу поделити у следеће категорије:

- WCMS (*Web Content Management System*), који омогућују аутоматизовано публиковање *web* садржаја,
- T-CMS (*Transactional Content Management System*) који аутоматизују *e-commerce* трансакције,
- I-CMS (*Integrated Content Management System*) који управљају документима у предузећима,
- P-CMS (*Publications Content Management System*) који управљају животним циклусом процеса издавања књига, часописа, упутстава и сл.
- L-CMS (*Learning Content Management System*) који управљају животним циклусом *e-learning* система,
- E-CMS (*Enterprise Content Management System*) који обједињују више технологија у циљу управљања животним циклусом великог броја различитих садржаја унутар великих предузећа. Ове технологије обухватају: управљање документима (*Document Management*), управљање пословним процесима (*Business Process Management*), управљање токовима података (*Workflow Management*) и, наравно, управљање садржајем *web* презентација (*Web Content Management*).

WCMS

Оно што се данас сматра за CMS је само подкуп CMS-а, а то је WCMS (*Web Content Management System*). WCMS у основном облику омогућује корисницима да публикују своје *web* презентације кроз WCMS тако што се садржај визуализује разним шаблонима (*templates*). У даљем тексту ће се WCMS звати CMS.

Типични аспекти коришћења CMS система су:

- унос докумената од стране корисника,
- обезбеђивање права приступа различитом садржају према различитим критеријумима,
- обезбеђивање потврде садржаја документа пре објављивања или у току обраде документа,
- архивирање садржаја и подршка контроли верзија докумената,
- употреба шаблона за приказ садржаја (чиме се обезбеђује конзистентан изглед) и
- подршка пријављивању/одјављивању за дистрибуиране кориснике.

Постоје две основне предности коришћења CMS система над ручним уносом садржаја:

- корисници могу лако да ажурирају садржај, без техничке помоћи и

- приступ подацима преко CMS система је знатно лакши зато што ови системи подржавају различите формате докумената, било код уноса (где је омогућена обједињена претрага по различитим типовима документа), било код излаза (чиме је омогућено да се један документ изведе у различитим облицима).

Осим наведених, предности су и у:

- смањењу трошкова одржавања, зато што није потребан посредник у виду техничког особља да би се садржај публикувао. Ово доводи и до бржег публикувања садржаја;
- лакшем праћењу историјата измена кроз логовање и контролу верзија и
- конзистентном приказу садржаја кроз употребу шаблона.

У табели 2.16. дата је упоредна табела послова који би се обављали без и са употребом CMS система.

Посао	Са CMS системом	Без CMS система
Креирање нових страница	Нова страница се креира на основу унапред дефинисаног шаблона. Сви линкови се аутоматски ажурирају и могуће је пратити историјат рада са страницом кроз систем логова.	Нова страница се креира као копија постојеће. Мапа сајта и линкови се морају ручно ажурирати.
Конзистентност садржаја	Шаблони су раздвојени од садржаја, чиме се омогућује конзистентан приказ кроз сајт.	Садржај и приказ су спојени и свака измена на изгледу већег броја страница се веома тешко изводи.
Workflow процеси	Уграђени подсистеми за контролу токова података (<i>workflow management</i>) омогућују контролу процеса публикувања садржаја, уз константно логовање свих акција. Након коначног одобрења, садржај се аутоматски публикује.	Контрола токова података се типично обавља путем <i>email</i> -а. Поруке се шаљу различитим особама унутар организације, и по пријему свих потребних одобрења, садржај се ручно публикује.
Времена издавања садржаја	Садржај се публикује моментално (по приспећу коначног одобрења).	Садржај се публикује када web администратор може да то уради.
Усаглашеност садржаја	Усаглашеност садржаја је наметнута самим системом контроле токова, логовањем и контролом верзија.	Усаглашеност садржаја зависи од чланова тима који публикује. Све измене се морају ручно архивирати, а сваки приступ подацима ручно логовати.

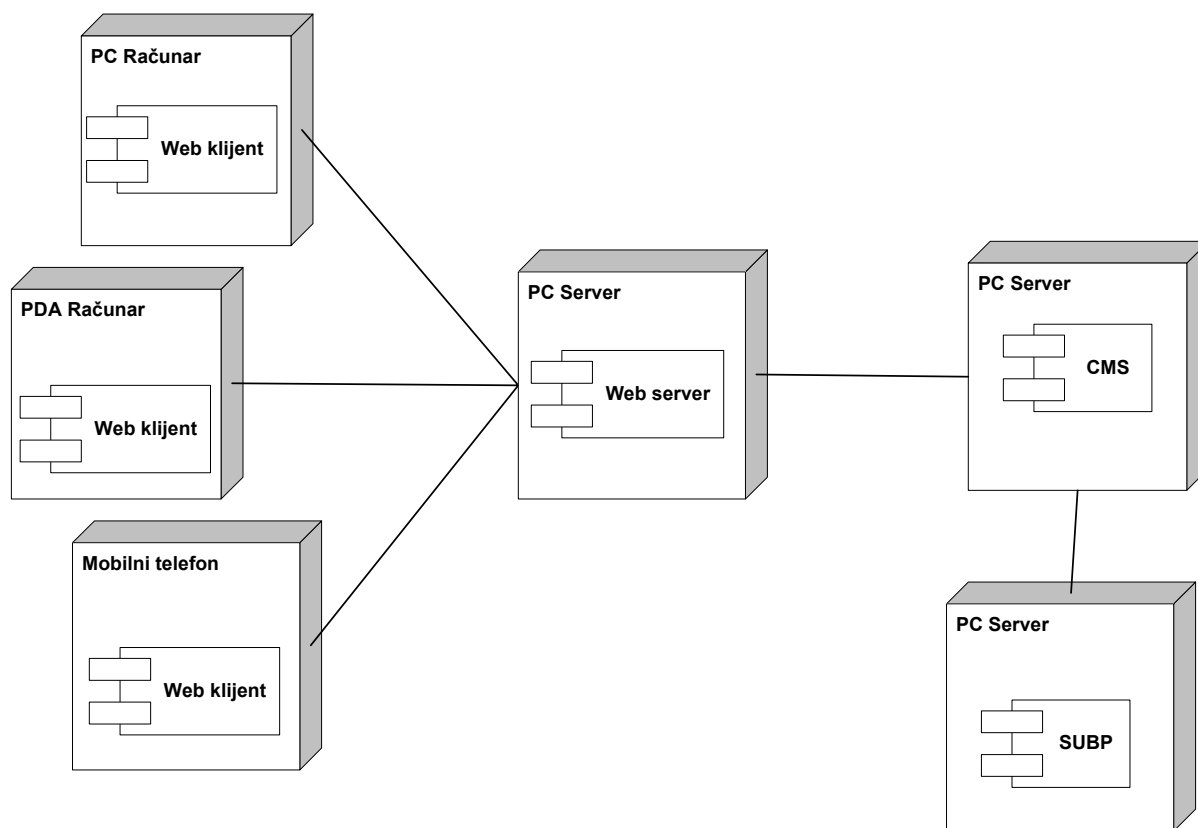
Табела 2.16. Поређење основних акција са и без CMS система

Реализација CMS система је обично у виду вишеслојне апликације, која се састоји из следећих елемената:

- CMS сервер,
- клијенти,
- web сервер и

- сервер базе података.

Шематски приказ је дат на слици 2.1.



Слика 2.1. Елементи CMS

Основа сваког CMS система је CMS сервер. Он омогућује креирање, управљање и испоруку садржаја. Уобичајено је да се садржај испоручује у виду HTML страница које се публикују на *web* серверу. Такође је уобичајено да се подаци чувају у неком систему за управљање базама података. Осим испоруке података, и сам процес креирања и управљања подацима је обично реализован као *web* апликација. То се ради зато што *web* апликације искључују потребу за инсталацијом специјализованог софтвера, тако да сви корисници могу да учествују у обради садржаја само уз употребу *web browser-a*.

Употреба CMS система у пројектима еУправе

Све већи број Интернет корисника има за последицу све већу потребу за *web* порталима свих нивоа власти [devOS-CMS]. Употреба CMS система омогућује конзистентан унос и презентацију података.

e-Government иницијатива покрива више циљева:

- комуникација са грађанима: осим прегледа информација (за шта би био довољан и статички *web* сајт), потребно је обезбедити и дисеминацију података, као и могућност претраге како *web* садржаја, тако и докумената на које се референцирају *web* странице (PDF документа, Word документа итд.). Осим овога, форуми у оквиру оваквих сајтова повећавају интересовање и учешће грађана.
- комуникација са компанијама: компаније које обезбеђују производе и услуге владиним агенцијама могу да приступају релевантним подацима преко *web-a*.

- комуникација унутар владиних агенција: правилно управљање подацима може да омогући креирање базе знања које може да подигне квалитет документације.
- комуникација између владиних агенција: дељење података између различитих CMS система може да побољша сарадњу између владиних агенција.

Типичан развој *e-government* [ejisdс] система илуструје и развој управљања садржајем који води ка CMS системима. На почетку, у *broadcast* моду, држава или њене агенције су присутни са статичким садржајем, и једносмерном комуникацијом. У моду *интеркације*, владини сајтови пружају информације и сервисе грађанима, тако што грађани могу да претражују и добављају ресурсе преко CMS система интегрисаних у портале. У *транзакционом* режиму грађани могу да обављају различите облике трансакција са државом, што захтева озбиљније сигурносне механизме. Коначно, у *интеграционом* моду, различите агенције сарађују у циљу елиминисања дуплих послова.

CMS системи у *eGovernment* пројектима

У Великој Британији се користи APLAWS+ [aplaws] CMS систем који се користи од стране локалних власти за постављање садржаја на web. Почео је 2001. године као APLAWS Pathfinder Project, да би 2004. године постао основа за Local Authority Websites National Project (<http://www.aplaws.org.uk/project/laws.php>). Систем је у сагласности са Government's Interoperability Framework (eGIF) стандардом и подржава Dublin Core стандард за размену метаподатака.

У Аустралији је развијен AGIMO (Australian Government Information Management Office) [agimo], на основу MySource Matrix *open source* CMS система. Портали који су на располагању становништву су дати на страници: <http://www.agimo.gov.au/resources/achievements>

На Филипинима се користи DO-CMS E-Government Version, развијен од стране EACOMM Corporation.

У Немачкој се користи "Government Site Builder" [gsb], развијен од стране фирми "Materna" и CoreMedia.

На Новом Зеланду се користи *open source* CMS систем под именом Plone [plone], који се базира на Zope [zope] апликационом серверу.

Подела CMS система

Постојећи CMS софтвер се дели на *open source* и комерцијална решења. Такође, CMS системи се деле и према програмској платформи у којој су направљени. Ту су присутна углавном решења базирана на Javi, PHP-у и ASP.NET платформи.

Осим шаблонизованог уноса и приказивања података, који представљају основне могућности CMS система, они обично поседују и следеће елементе:

- управљање токовима података (*Workflow Management System*),
- кеширање садржаја, које убрзава приказ садржаја
- управљање улогама и корисницима, које омогућује различит приступ подацима на основу поделе по улогама и корисницима,
- различите нивое сигурности, који омогућавају гранулацију приступа подацима на основу сигурносних поставки.

На основу наведених података, на [wikipedia-comparison-CMS] се може наћи табела која нуди упоредну анализу постојећих система (табела 2.18). Особине које су анализирале у овом поређењу дате су у табели 2.17.

Критеријум	Објашњење
WYSIWYG	<i>What you see is what you get.</i> Да ли CMS омогућава да се садржај едитује у облику у којем ће бити и приказан?
Workflow	Да ли CMS подржава WMS (<i>Workflow Management System</i> : content sign-off, approval, commenting, roll forward / roll back)?
Caching	Да ли презентациони сегмент подржава кеширање садржаја у циљу убрзања приказа?
Member Support	Да ли CMS поседује механизам за управљање корисницима и њиховим правима приступа систему)?
Multiple Security Roles	Да ли CMS подржава различите нивое права приступа (администратори, уредници, аутори, итд.)?
Multi-Site	Да ли CMS подржава вишеструке web сајтове кроз један административни модул?
Friendly Links	Да ли CMS подржава тзв. "пријатељске URL-ове", одн. да ли су URL-ови на сајту у облику погоднијем за претраживаче (попут /mypage/mytopic/ у односу на: /mycms?pageid=22)

Табела 2.17. Критеријуми за поређење CMS система

CMS	WYSIWYG	Workflow	Caching	Member Support	Multiple Security Roles	Multi-Site	Friendly Links
Accura Site CMS	Да	Да	Да	Да	Да	Да	Не
ADXSTUDIO CMS 2005	Да	Да	Да	Да	Да	Да	Да
Apache Lenva	Да	Да	Да	Да		Да	Да
bitweaver	Да	Да	Да	Да	Не	Не	
boastMachiHe	Да	Да	Да	Делимично	Не	Не	Не
Cola	Да	Да	Да	Да	Да	Да	Не
Bricolage		Да	Да				Да
CMScout	Не	Да	Да	Делимично	Да	Да	Да
Colony	Да	Да	Да	Да	Да	Да	Да
Corinis	Да	Да	Да	Да	Да	Да	Да
Digimaker CMS HetServer	Да	Да	Да	Да	Да	Да	Да
DotHetNuke	Да	Да	Не	Да	Да	Да	Да
Drupal	Да	Да	Да	Да	Да	Да	Да
Easy Website Builder	Да	Да	Да	Да	Да	Да	Да
eDIY Software	Не	Да	Не	Да	Да	Да	Да
ElementCMS	Да	Да	Да	Да	Да	Да	Да
e107	Да	Да	Не	Да			
evoArticles	Да	Да		Да			
eZ publish	Да	Да	Да	Да	Да	Да	Да
FatWire	Да	Да	Да	Да	Да	Да	Да
Freestyler CMS	Да	Да	Да	Да			
Geeklog	Да	Да	Да	Не	Да	Да	Да
Joomla!	Да	Да	Да	Да	Да	Да	Не
Magnolia	Да	Да	Не	Да	Да	Да	Да
Mambo	Да	Да	Да	Да	Да	Да	Не
Nucleus CMS	<i>Plugin</i>	Не	<i>Plugin</i>	Да	Да	Да	Опционо
Numotion CMS	Да	Да	Да	Да	Да	Да	Да
ocPortal	Да	Да	Да	Да	Да	Да	Да
OpenCMS	Да	Да	Да				
OpeHedit	Да	<i>Sitewide</i>	Да	Да	Да	Да	Не
PhpWCMS	Да	Да	Да	Да	Да	Да	Да
PostNuke	Да	Ограничено	Да	Да	Да	Да	Опционо
PloHe	Да	Да	Да	Да	Да	Да	Да
Scoop	Не		Да				
Serendipity	Да	Да	<i>Plugin</i>	Да	Да	Ограничено	Да
Silva	Да	Да	Да	Да	Да	Да	Да
Simplicis CMS	Да	Не	Да	Да	Да	Да	Да
SyntaxCMS	Да	Не	Да	Да	Да	Да	Да
Textpattern	Не	Да	<i>Plugin</i>		Да		Да
Typo3	Да	Да	Да	Да	Да	Да	<i>Plugin</i>
WordPress	Да	Да	<i>Plugin</i>	Не	Да	Да	Да
Xarava	Да	Да	Да				
XOOPS	Да		Да				Не
ubiCMS	Не	Делимично	Делимично	Да	Да	Не	Да
Web Wiz Site Hews	Да	Не	Не	Не	Не	Не	Не
XSM	Да	Делимично	Да	Да	Да	Да	Да

Табела 2.18. Поређење постојећих CMS система
извор: http://en.wikipedia.org/wiki/Comparison_of_content_management_systems

Много детаљнији преглед могућности дат је у [gilbane-report]. Сва решења се пореде по следећим критеријумима:

- основни подаци (лиценце, инсталација, подршка, итд.),
- креирање садржаја (аквизиција, агрегација и ауторизација),
- управљање садржајем (workflow, едитовање, одобрење публикације, фазе, смештај садржаја),
- испорука садржаја,
- побољшања у животном циклусу CMS-а (безбедносна питања, пословна логика, могућност интеграције, управљање мета подацима, итд.).

Према наведеним критеријумима, направљена је база знања о постојећим CMS системима, која омогућује поређење по два одабрана CMS система. База знања је доступна на следећој адреси: <http://www.cmsreview.com>

База знања подржава CMSML и RDF структуру која омогућује претраживачима да претражују овај списак и извлаче податке из њега.

Анализа CMS система

Анализирана су три CMS система: Apache Lenya, Magnolia и OpenCMS.

Apache Lenya

Основни подаци - Apache Lenya је *open source* пројекат Apache фондације који пружа све могућности CMS система: управљање садржајем, контролу верзија, *scheduling*, претрагу, WYSIWYG едиторе HTML страница и управљање токовима података (workflow management).

Lenya се испоручује и у виду бинарне архиве и у виду изворног кода. Инсталација бинарне архиве се своди на копирање *war* архиве у *servlet* контејнер.

Креирање садржаја - Систем омогућује интеграцију WYSIWYG едитора и испоручује се са два едитора: Кири и ВХЕ. Систем такође садржи и едитор форми. Основа је XHTML и CSS. Систем омогућује употребу специјализованих JSP тагова.

Управљање садржајем - Структура сајта је приказана у облику стабла и подржане су све основне операције над садржајем. Свака страница је карактерисана следећим атрибутима: подаци, мета подаци, *assets*, статус у току података, верзија, контрола приступа и *scheduling*.

Уграђени *workflow* систем има следеће карактеристике: токови базирани на XML-у, корисници се могу обавештавати *email*-ом о документима који захтевају одобрења, сви кораци у току података се бележе у логове.

Испорука садржаја - Садржај се испоручује у *friendly links* облику. Уграђени *navigation framework* аутоматски креира меније, стабла приступа и табове. Креирање мапе сајта је једноставно. Постоји контрола средстава (*asset control*) која омогућује праћење слика и докумената који припадају страници.

Додатне особине - Интернационализација је подржана. Свака страница може бити заштићена SSL-ом. Деловима сајта се могу дефинисати посебна права приступа на нивоу група и корисника. Корисници могу да се аутентификују и преко LDAP сервера.

Magnolia

Основни подаци - Магнолија је *open source* производ Magnolia Organisation LLC. Постоји два нивоа лиценцирања: бесплатна *community edition* и *Enterprise edition*. Осим овога, постоји и *Document Management System* који се посебно лиценцира (не

постоји бесплатна верзија) и који омогућује рад са документима, индексирање и претрагу. Осим DMS, постоји и додатни модул за *Magnolia for Business Processes* који представља процесно-оријентисан подсистем за праћење пословних процеса и активности и који се посебно наплаћује (не постоји бесплатна верзија).

Магнолија се испоручује и у виду бинарне архиве и у виду изворног кода. Инсталација бинарне архиве се своди на копирање war архиве у JSP контејнер.

Креирање садржаја - Систем поседује WYSIWYG едитор. Систем такође садржи и генератор форми. Систем омогућује употребу специјализованих JSP тагова. Постоји мод за преглед и едитовање страница.

Управљање садржајем - Структура сајта је приказана у облику стабла и подржане су све основне операције над садржајем. Аутоматско складиштење мета тагова о страницама је подржано.

Испорука садржаја - Садржај се испоручује у friendly links облику. Претрага сајта је аутоматизована. Web интерфејс је побољшан и не заснива се на стандардном скупу контроли web читача, већ су развијене посебне контроле.

Додатне особине - Интернационализација је подржана. Систем подржава кеширање страница, чиме се постиже знатно убрзање у приступу. Архитектура је модуларна, али се најбитнији модули наплаћују. Подржава скалабилност, кластеризацију и балансирање оптерећења.

Open CMS

Основни подаци - OpenCMS је Open Source пројекат Alkacon Software GmbH компаније. Систем омогућује основне CMS радње, а садржај се смешта у базу података (*Database driven*). Тренутно је подржан MySQL. Подржава и елементарну контролу тока података кроз механизам публикувања страница. Испоручује се и у виду бинарне архиве и у виду изворног кода. Инсталација бинарне архиве се своди на копирање war архиве у *servlet* контејнер.

Креирање садржаја - Систем поседује WYSIWYG едитор. Систем омогућује употребу специјализованих JSP тагова. Постоји мод за преглед и едитовање страница. Постоји више погледа на сајт: странице сајта, административни и workflow.

Управљање садржајем - Структура сајта је приказана у облику стабла и подржане су све основне операције над садржајем. Систем подржава вишеструке сајтове.

Испорука садржаја - Садржај се испоручује у friendly links облику. Систем подржава кеширање страница, чиме се постиже знатно убрзање у приступу.

Додатне особине - Интернационализација је подржана. Систем омогућује персонализацију web садржаја. Систем подржава SSL.

Табела 2.19. приказује преглед наведених особина по анализираним CMS системима:

Систем	Креирање садржаја	Управљање садржајем	Испорука садржаја	Додатне особине
Apache Lenya	WYSIWYG едитор	- структура сајта у облику стабла - подржане су све основне операције над садржајем	- friendly links облик - уграђени navigation framework - аутоматски креира меније, стабла приступа и табове	- интернационализација - свака страница може бити заштићена SSL-ом. - деловима сајта се могу дефинисати посебна права приступа на нивоу група и корисника - корисници могу да се аутентификују и преко LDAP сервера
Magnolia	- WYSIWYG едитор - генератор форми - специјализовани JSP тагови	- структура сајта у облику стабла - подржане су све основне операције над садржајем - аутоматско складиштење мета тагова	- friendly links облик - претрага сајта је аутоматизована	- интернационализација - систем подржава кеширање страница - архитектура је модуларна, али се најбитнији модули наплаћују
OpenCMS	- WYSIWYG едитор - специјализовани JSP тагови	- структура сајта у облику стабла - подржане су све основне операције над садржајем - систем подржава вишеструке сајтове.	- friendly links облик - систем подржава кеширање страница	- интернационализација - систем омогућује персонализацију web садржаја - систем подржава SSL

Табела 2.19. Преглед особина анализираних CMS система

2.5. Смернице и искуства у ЕУ

Open source софтвер у *eGovernment*-у

Постепени прелазак на еУправу неизоставно ће донети потребу великог инвестирања у ИТ технологије током следећих неколико година. Стога се при разматрању решења која ће бити прихваћена у овом сектору јавних служби, мора извршити процена свих расположивих алтернатива, укључујући ту и *open source* решења, чија примена може имати одлучујући утицај на иницијалну цену увођења јавних сервиса еУправе.

Архитектура софтверских система који треба да обезбеде подршку за увођење јавних сервиса еУправе, је у општем случају најчешће трослојна архитектура. Како се у овој архитектури јасно разликују три слоја:

- клијентски слој,
- средњи слој (*middleware* - слој апликационих сервера, пословне логике) и
- слој базе података.

Могуће је извршити анализу расположивих *open-source* решења у сваком појединачном слоју. Могуће је анализу извршити и по месту примене одређеног софтвера, па је при анализирању економске вредности употребе *open source* софтвера у јавним службама могуће анализирати употребу оваквих решења у три кључна сегмента:

- у сегменту десктоп апликација (канцеларијски софтверски пакети, клијентски софтвер за различите намене),
- инфраструктурни софтвер,
- примена *open source* решења при изради апликација по наруџби

Ове две поделе се у већем делу преклапају – сегмент десктоп апликација одговара клијентском слоју, инфраструктурни софтвер највећим делом припада другом слоју, мада једним делом (системи за управљање базама података) припада и трећем слоју. Апликације по наруџби могу бити реализоване као део било ког слоја трослојне архитектуре.

Десктоп апликације представљају у суштини канцеларијске софтверске пакете које особље у јавним службама користи за обављање редовних послова. Под овим се подразумевају апликације за обраду текста, софтвер за разне врсте израчунавања, припрему презентација, као и *web browser*-и, *email* клијентске апликације за размену email порука, разне врсте организатора и сл.

Инфраструктурни софтвер представља апликације које крајњи корисници не користе директно, него ове апликације посредно омогућавају другим апликацијама успешно обављање одређених активности. Под овим сегментом можемо посматрати разне *web* сервере, *mail* сервере, сервери база података, оперативни системи. Овај софтвер представља темељну инфраструктуру која чини могућем обављање послова јавне администрације у електронском облику. Најбитнији део инфраструктуре која омогућава еУправу је постојање добре телекомуникационе инфраструктуре и поуздан и константан приступ Интернету, јер се основна веза између грађана, предузећа и јавне администрације обавља на овај начин тј. коришћењем разних Интернет сервиса.

Апликације које се развијају за одређену, специјалну намену, такође могу бити развијане коришћењем *open source* платформи.

FLOSS пројекат европске уније – анализе коришћења *open source* софтвера у јавним институцијама и компанијама

Европска унија је у оквиру пројекта Free/Libre Open Source Software [FLOSS] наручила анализу тренутног стања коришћења *open source* софтвера (OSS). Током ове анализе урађен је већи број анкета у различитим јавним службама и корпорацијама, на основу којих је израђен коначни извештај пројекта у коме је дат пресек прикупљених података као и потенцијални правци кретања на одређеним сегментима софтверског тржишта. Завршни извештај пројекта поднет је у лето 2002. године. Овај извештај је био један од докумената на основу којих је Европска унија усвојила своје смернице за прихватање *open source* базираних софтверских решења за потребе јавних сервиса. Ова анализа се концентрисала на неколико земаља ЕУ-а. Део закључака овог пројекта је дат у наставку. Пуни извештај FLOSS пројекта могуће је наћи на [FLOSS Reports].

Степен прихватања OS софтвера

Једна од веома корисних анализа је био степен прихваћености и планираног прихватања *open source* софтвера по областима ИТ индустрије (резултати дати у табели 2.18). Анализа је рађена за сваку земљу уз посебан осврт на мале и велике институције/компаније.

	Велика Британија		Шведска		Немачка		Укупно
	мале институције	велике институције	мале институције	велике институције	мале институције	велике институције	
OSS као серверски оперативни систем	8,1%	3,7%	9,8%	11,0%	30,7%	30,6%	15,7%
	6,4%		10,1%		30,7%		
OSS за базе података	13,3%	4,6%	7,5%	8,2%	14,1%	20,8%	11,1%
	9,9%		7,6%		15,7%		
OSS на десктоп рачунарима	7,6%	2,0%	3,4%	3,2%	13,7%	6,5%	6,9%
	5,4%		3,3%		12,0%		
OSS за развој web презентација	7,9%	4,3%	7,5%	8,7%	15,8%	17,3%	10,2%
	6,5%		7,8%		16,2%		

Резултати истраживања
на узорку n=395

Табела 2.18. Прихваћеност OSS решења у различитим областима намене (анализа из 2001. год.)

Видљиво је да је *open source* софтвер најзаступљенији у сегменту серверских оперативних система, где у просеку 15.7% свих анализираних институција користи OSS као што су Linux или Free/Open BSD као серверску платформу у редовним делатностима, или то планирају у скорој будућности. При томе видљиве су велике разлике у заступљености у појединим земљама.

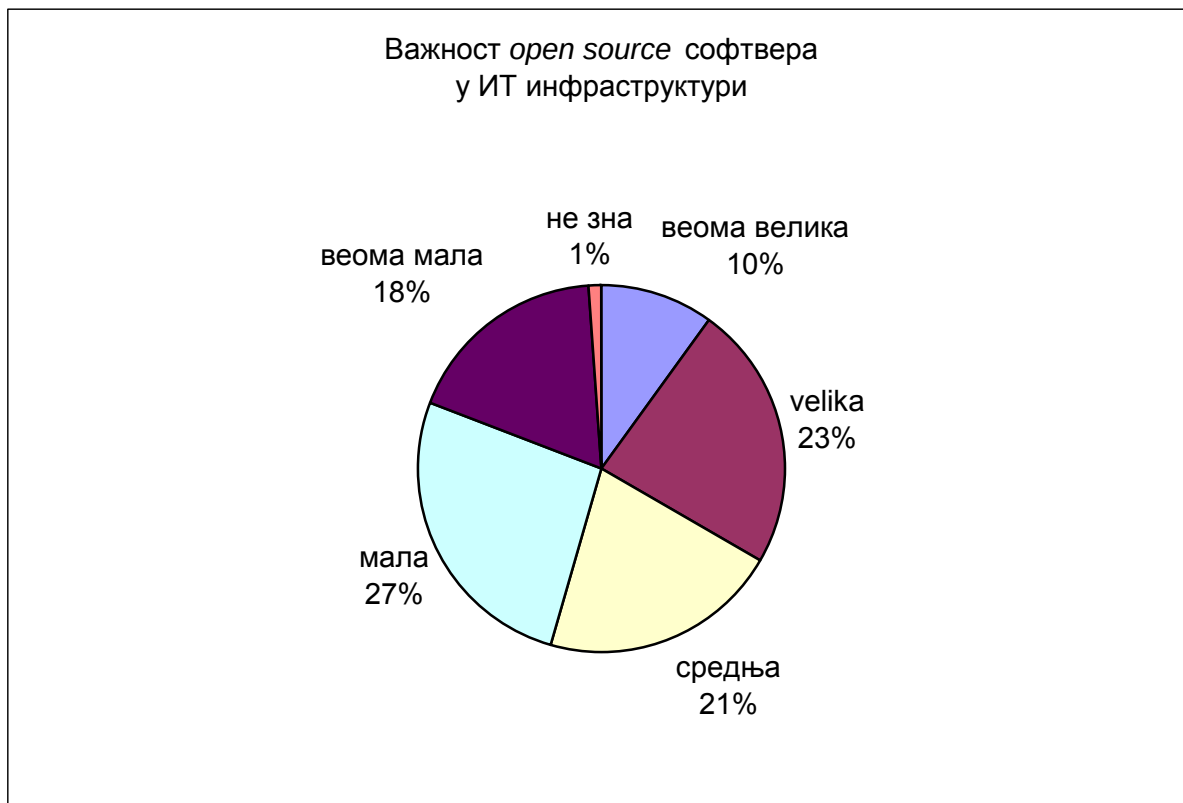
Друга по популарности област у којој се OSS решења редовно користе је област система за управљање базама података. У овом сегменту постоје *open source* решења која су широко прихваћена: MySQL, PostgreSQL, Interbase и SAP-DB. У просеку 11.1% институција користи ове базе података за обављање редовних делатности, при чему је степен коришћења поновно највиши у Немачкој.

Следећа по популарности је област креирања и одржавања web садржаја. У просеку 10,1% институција користи OSS производе за ове намене. У овом сегменту постоји велики број апликација нпр. web сервери (Apache), проху сервери (Squid), окружења за развој web апликација, системи за одржавање садржаја.

Коначно, open source се може наћи и на десктоп рачунарима, али је степен продора open source апликација у ову област ипак знатно мањи.

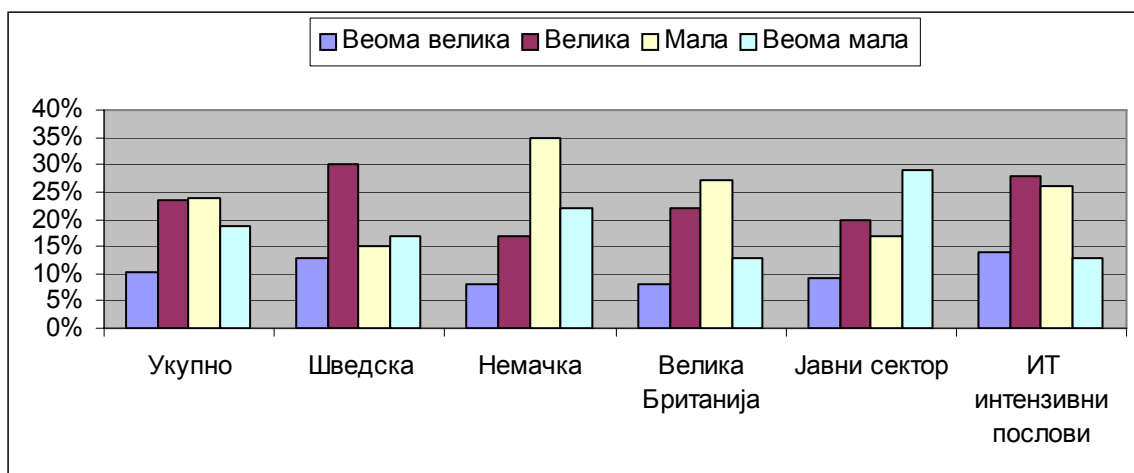
Важност *open source* софтвера у ИТ инфраструктури

Друга занимљива анализа односила се на процену степена важности OSS-а у области ИТ инфраструктуре институција (резултати приказани на дијаграму 2.1).



Дијаграм 2.1. Оцена важности OS решења за укупну ИТ инфраструктуру институција

Са овог дијаграма видљиво је да већина институција сматра да је коришћење OSS-а у овој области средње до јако важно за успешно функционисање њихове ИТ инфраструктуре. Осим ове уопштене анализе, када је исто питање разврстано по земљама у којима се налазе анкетиране институције, и области деловања (јавни сектор, компаније са интензивним ИТ секторима) добијени су резултати приказани на дијаграму 2.2.



Дијаграм 2.2. Оцена важности OS софтвера за ИТ инфраструктуру институција, преглед по земљама, у јавном сектору, као ИТ интензивним делатностима

И овде су видљиве одређене разлике по земљама: OSS се у Немачкој сматра најмање битним за функционисање ИТ инфраструктуре, док је тај ниво у Шведској и Британији већи. Током анализе нису уочене веће разлике у ставовима мањих и већих компанија по овом питању. Компаније са интензивним ИТ секторима сматрају OSS много важнијим за своју ИТ инфраструктуру него институције у јавном сектору.

Анализа бенефита при коришћењу open source софтвера

Једна од анализа у оквиру FLOSS пројекта бавила се проценом бенефита који институције/компаније очекују од употребе OSS-а. Иако би за анализу користи било најбоље имати могућност процене директног финансијског утицаја, ово није реалистичан циљ, због мноштва краткотрајних и дуготрајних фактора које треба узети у обзир, и компаније које су учествовале у анкети у општем случају нису могле извести такву процену. Због тога је сачињена листа потенцијалних критеријума користи који су утицали на одлуку о прихватању OSS-а.

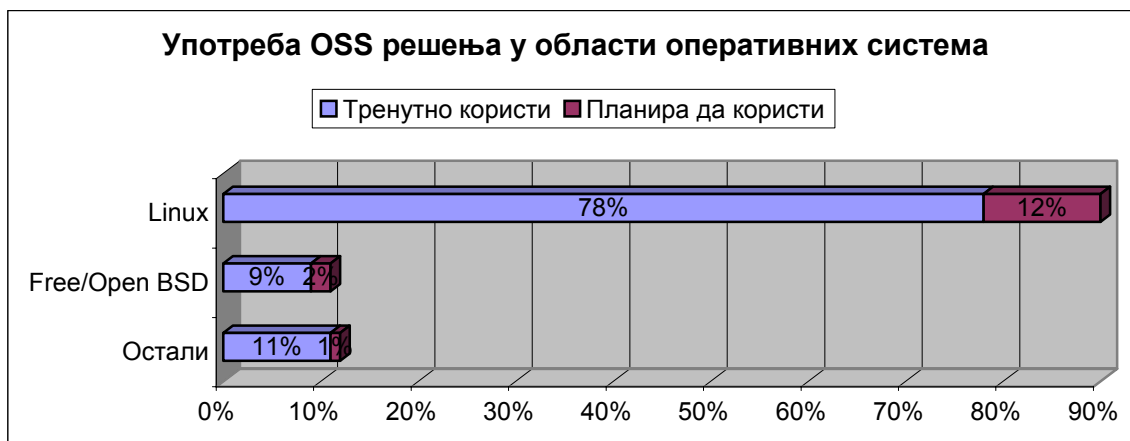
Важност ових критеријума је евалуирана у оквиру претходно споменутих области примене:

- *open source* софтвер као оперативни систем на серверима (Linux or Free/Open BSD),
- *open source* софтвер за базе података (MySQL, PostgreSQL, Interbase, SAP-DB),
- *open source* софтвер на десктоп (клијентским) рачунарима (Linux, KDE, Gnome, Mozilla, StarOffice/OpenOffice)
- *open source* софтвер као алат за креирање, одржавање и hosto-овање web сајтова (Apache, PHP, Perl, Python, Squid и сл).

OSS као серверски оперативни систем

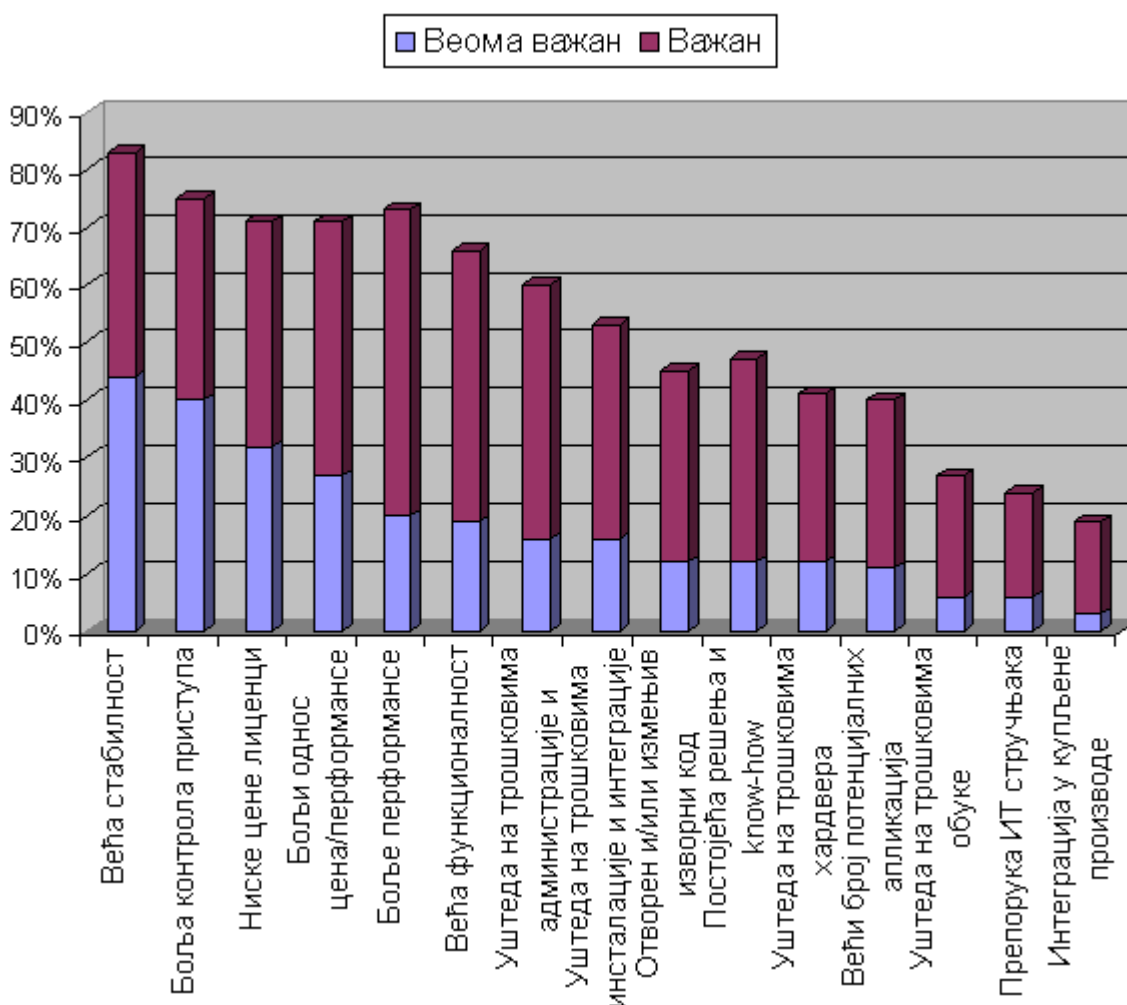
56% анкетираних корисника користи OSS као серверски оперативни систем. Од 395 институција/компанија, 220 већ користи или планира увођење OSS оперативног система.

Учешће различитих оперативних система дато је на дијаграму 2.3.



Дијаграм 2.3. Употреба *open source* софтвера у области оперативних система

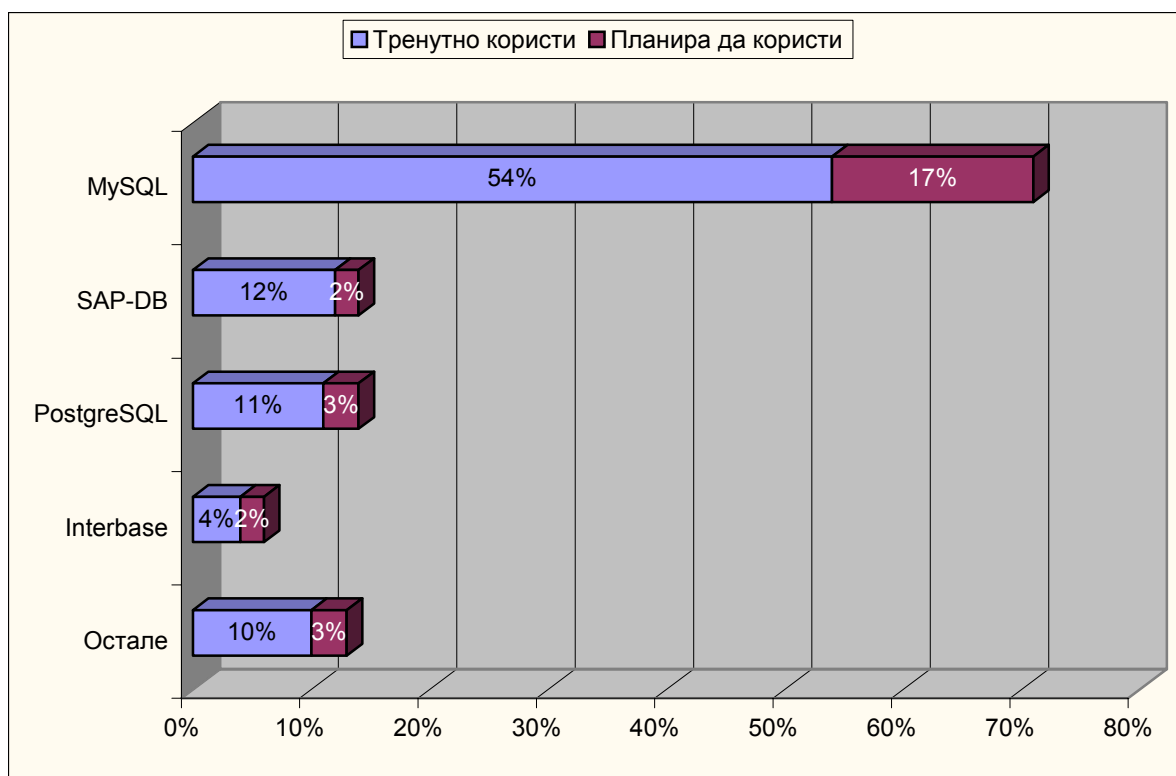
Види се да је од свих OSS серверских оперативних система апсолутно најзаступљенији Linux (у различитим дистрибуцијама). Фактори који су утицали на одлуку да се изабере OSS оперативни систем рангирани по релевантности су дати дијаграму 2.4.



Дијаграм 2.4. Релевантност одређених фактора при избору OSS решења у области оперативних система

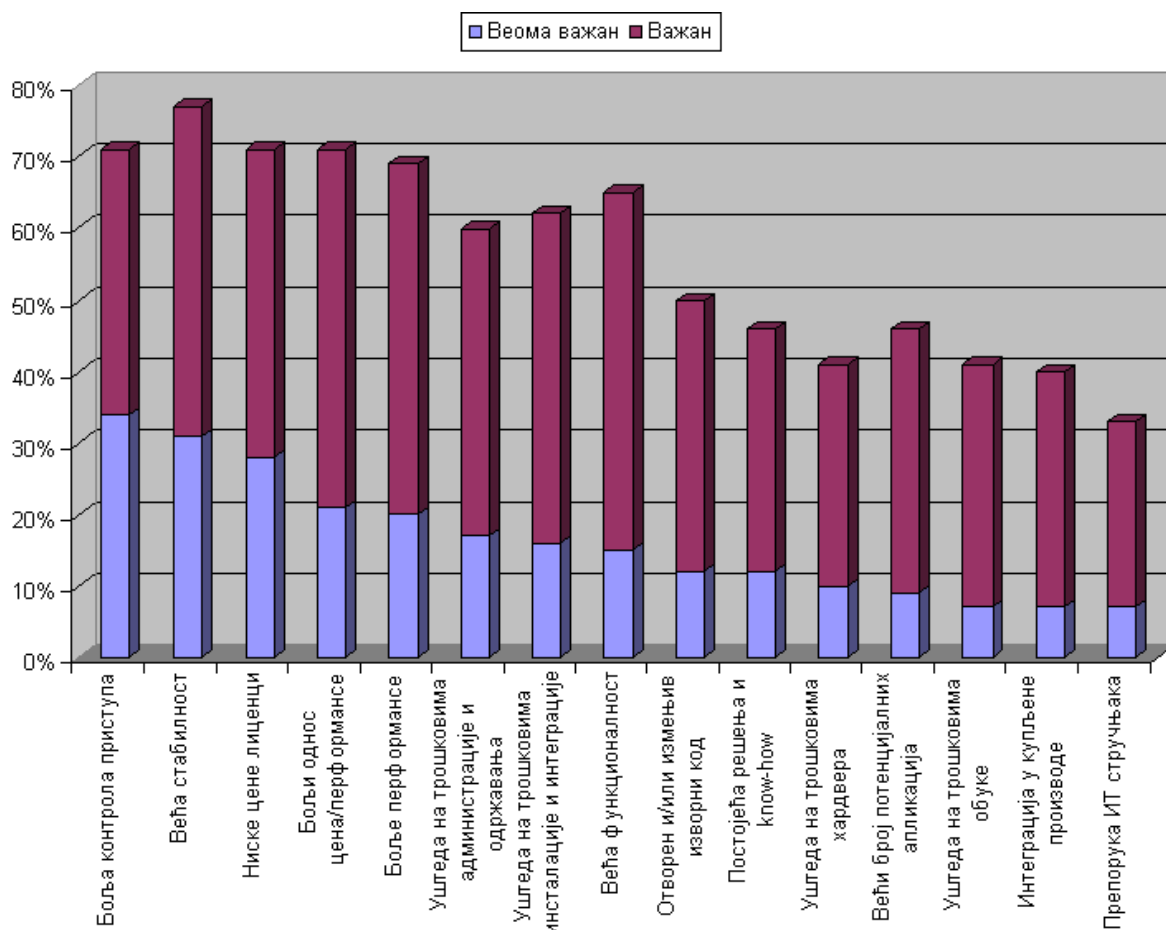
OSS у системима за управљање базама података

Базе података су незаобилазан сегмент ИТ инфраструктуре било које компаније или јавне институције. Иако неке апликације имају инкорпорирану одређену врсту базе података, у већини случајева институције и компаније инсталирају самосталне системе за управљање базама података на којима похрањују највећи део својих података. Ово је нарочито случај у дистрибуираним системима где више различитих корисника може приступати истим базама података. У овој анализи пажња је посвећена управо оваквим самосталним системима за управљање базама података. На дијаграму 2.5 дато је учешће одређених OSS базираних система за управљање базама података у укупном броју оваквих система.



Дијаграм 2.5. Удео одређених RDBMS система

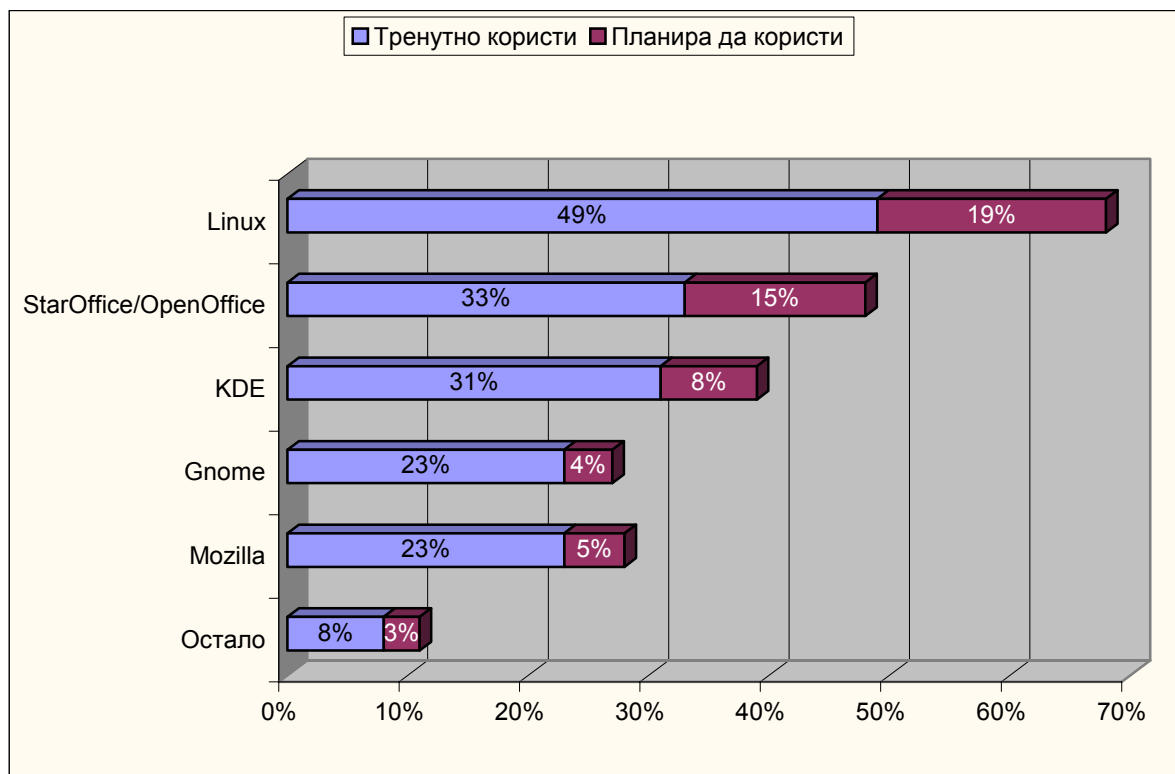
Са дијаграма је видљиво да је у већини институција најзаступљенија OSS базирана база података MySQL. Важно је напоменути да ова база података и поред широке распрострањености има и врло озбиљне недостатке (нпр. недостатак референцијалног интегритета, непостојање концепта *stored процедура* итд). Занимљиво је да је SAP-DB који је кратко пре прављења ове анкете постао *open source* пројекат, успео за кратко време да се пробије на друго место. Као и у случају примене OSS базираног софтвера за серверске оперативне системе, и овде је урађена анализа критеријума који су утицали на одлуку да се изабере одређена база података. На дијаграму 2.6. дат је приказ ових резултата.



Дијаграм 2.6. Утицај одређених фактора при избору OSS решења у области база података

Десктоп апликације и софтвер на радним станицама

У FLOSS пројекту урађена је анализа употребе OSS базираног софтвера на радним станицама. Резултати ове анализе приказани су на дијаграму 2.7.

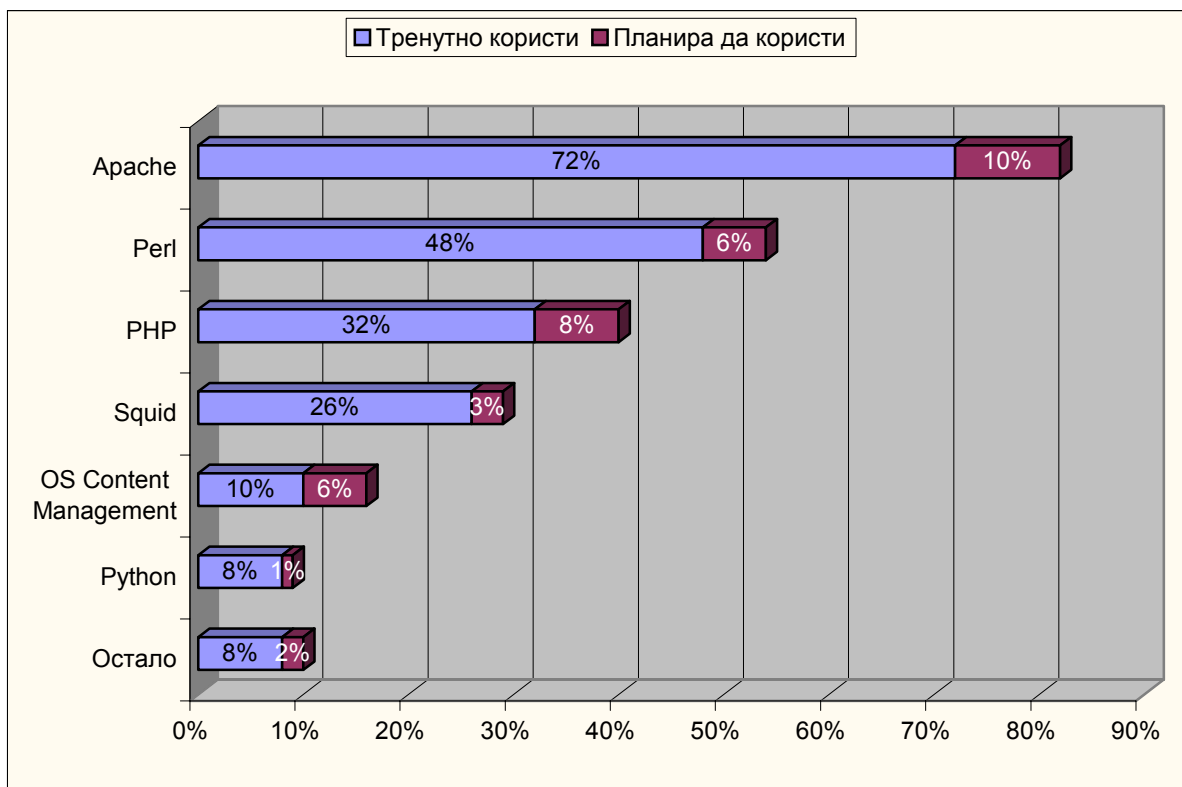


Дијаграм 2.7. Заступљеност OSS на десктоп конфигурацијама

Примена OSS базираног софтвера за креирање, одржавање и host-овање web садржаја

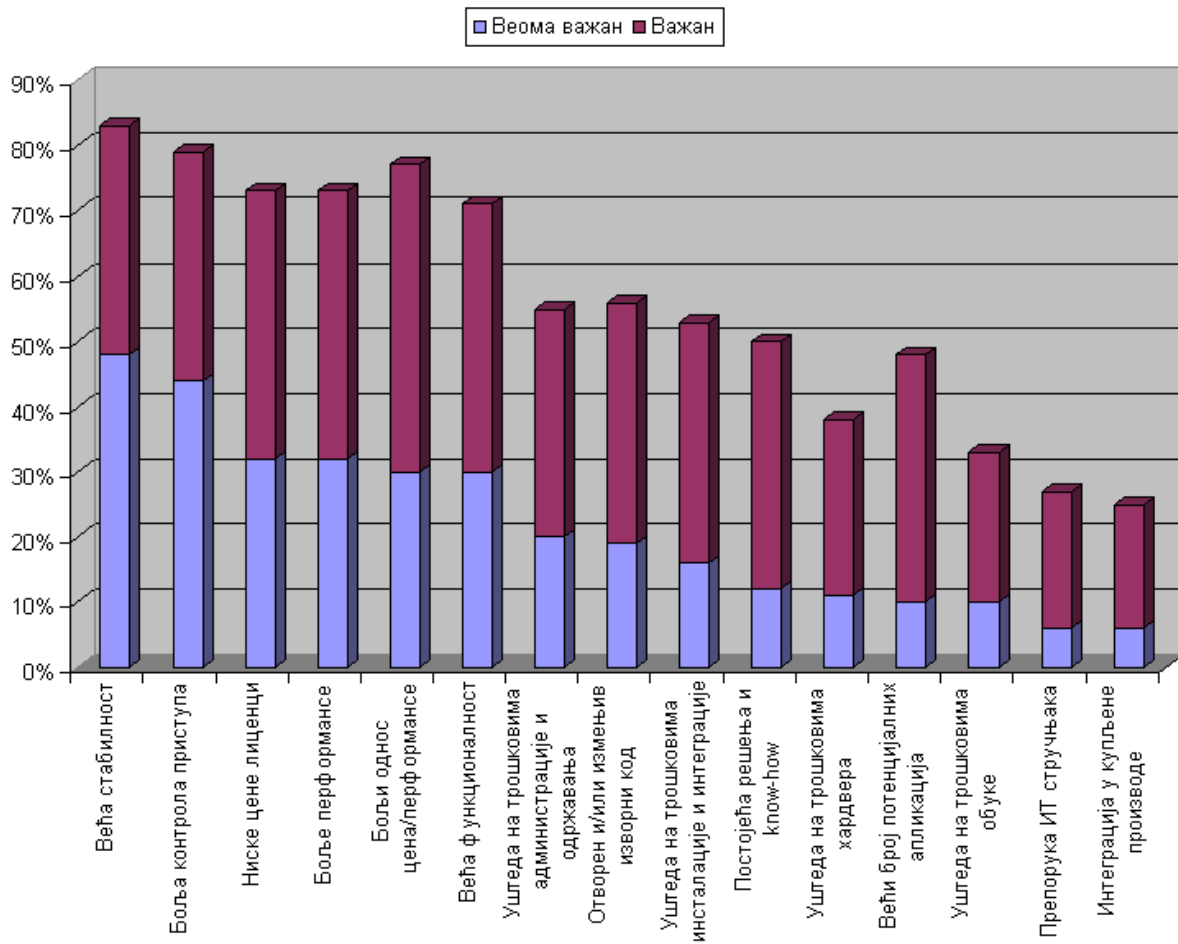
Осим у области серверских оперативних система, примена OSS базираног софтвера популарна је и у овој области. Web сервер Apache, који је *open source* пројекат је тренутно доминантан web сервер на Интернету, при чему је Apache на чак 57% свих web сервера. Осим тога у овој области су и врло популарни алати за израду динамичких web садржаја, као што су PHP, Python.

Удео појединих апликација у укупном броју OSS базираних решења у овој области дат је на дијаграму 2.8.



Дијаграм 2.8. Заступљеност OSS решења у области алата за web садржаје

При овој анализи врло је занимљиво погледати критеријуме за избор OSS базираних решења у овој области. На дијаграму 2.9. је дат приказ критеријума по њиховој релевантности, где се види да је и у овом случају највећи утицај при избору open source решења имало уверење да ће на овај начин корисници постићи већу стабилност и већу сигурност.



Дијаграм 2.9. Утицај одређених фактора на избор OSS решења за област web презентација

3. Препоруке за избор софтверских компоненти за еВојводину

На основу претходних анализа и искустава других земаља у овој области могуће је донети одређене препоруке у погледу избора софтверске архитектуре за потребе увођења система еВојводине. Због комплексности система који би обухватио све сервисе које је могуће понудити у оквиру еУправе, немогуће је дати једнократну и тачну процену о потребном броју web сервера, сервера база података, апликационих сервера. Овакву процену могуће је дати тек анализирајући поједине сервисе, утврђујући очекивану оптерећеност сервиса, потенцијалну базу корисника, комплексност послова који се обављају, количину докумената коју дати сервис треба да понуди/обради. На основу оваквих анализа могуће је донети процену иницијалних потреба, као и предвидети евентуална проширења инфраструктуре у зависности од раста тражње за услугама датог сервиса. У иницијалним фазама, док је број корисника релативно мали, економски је исплативо одржавати више сервиса на истој хардверској инфраструктури, што смањује иницијалне трошкове увођења сервиса.

3.1. Оперативни системи и десктоп апликације на радним станицама

С обзиром на апсолутну доминацију Microsoft Windows оперативних система у домену десктоп рачунара, као и на чињеницу да је у нашој земљи већина корисника, дуги низ година, употребу рачунара везивала управо за оперативни систем Windows, за десктоп рачунаре се препоручује употреба неког од система из фамилије Microsoft Windows. Овакав избор је оправдан и чињеницом да у већини случајева није неопходна дуга и детаљна обука корисника за коришћење овог OS-а. Верзија која би тренутно задовољила потребе запослених у администрацији је Windows XP. На овој верзији могуће је покренути и најновије верзије апликативних софтвера. Као алтернатива овом оперативном систему на десктоп рачунарима би се могле инсталирати одређене дистрибуције Linux-а. У овом случају би се избегли трошкови плаћања лиценце за оперативни систем, али би било немогуће инсталирати неке од најчешће коришћених програмских пакета, продуктивност рада код већине службеника би била знатно смањена у одређеном временском интервалу, а било би, за скоро све службенике, неопходно обезбедити и детаљну обуку.

Друго питање које је везано за десктоп рачунаре запослених у јавним службама јесте избор пакета канцеларијских програма. И у овом случају, због велике заступљености и етаблирања као *de facto* стандарда, као препоручени избор се намеће пакет Microsoft Office (погледати поглавље 2.1.) Овај пакет садржи већину неопходних алата за обављање свакодневних канцеларијских послова, као и за припрему докумената за интерну употребу. Документе у MS Office форматима могуће је публиковати и на Интернет-у путем јавних сервиса. Као што је то случај и са оперативним системом Windows, већина корисника је у одређеној мери већ оспособљена за коришћење програма из овог пакета. Ову оспособљеност, би ипак, у погледу продуктивности и коришћења напредних функција, вероватно требало тестирати и по потреби кориснике додатно обучити, како би се постигли максимални ефекти.

За све десктоп рачунаре на којима се креирају документи који могу бити објављени путем јавних сервиса, неопходно је обезбедити неки софтвер који омогућава

конверзију докумената у pdf формат (Adobe Acrobat, или неко од *open source/freeware* решења). Ово је неопходно, јер је овај формат постао стандард за објављивање докумената на Интернету, а сем тога јавни сервиси државне администрације би требали да омогуће објављивање докумената у форматима за које постоје бесплатни читачи, како се не би форсирала неопходност набавке тачно одређених софтверских пакета (у овом случају MS Office-а) и од стране грађана, који се налазе у улози корисника јавних сервиса. Овакав издатак би за одређене групе грађана био оптерећење (овакав став је наглашен у [FLOSS, Danish-Software-Strategy]), а подстицање на коришћење нелегалног софтвера (који би у том случају могло да представља инсистирање на документима у MS Office формату) је за јавне службе недопустиво.

За одређени број радних станица би било пожељно размислити и о могућности коришћења OpenOffice пакета, као начин за обезбеђење отворености система јавних сервиса и за оне кориснике које своје документе креирају у пакетима чији формат докумената се базира на OpenDocument [OpenDoc]формату.

Као неизбежан софтвер на данашњим десктоп рачунарима се, свакако појављују, Интернет читачи (*browser*-и). MS Internet Explorer је *browser* који долази преинсталиран уз сваку верзију Windows-а. Ипак, првенствено из безбедносних разлога, а донекле и због брзине приказа Интернет садржаја, као валидну алтернативу треба разматрати неко од бесплатних решења (понајвише Mozilla Firefox, или Opera). Ови *browser*-и су много отпорнији на неке нападе који типично искориштавају пропусте лоциране у Internet Explorer-у. Ови сигурносни пропусти, наравно, могу бити на време санирани редовним ажурирањем помоћу Windows Update сервиса. Поред *browser*-а, неизоставан софтвер на свакој радној станици у данашњим условима је и одређена врста *email* клијентске апликације. Уз сваку верзију оперативног система Windows испоручује се поред InternetExplorera као стандардног *browser*-а и OutlookExpress као подразумевана *email* клијентска паликација. Уколико се на радним станицама користи и канцеларијски пакет MS Office, могуће је као *email* клијента користити и MS Outlook који нуди и додатне функционалности.

На основу наведених особина као препоруке за софтвер на радним станицама могу се усвојити следеће (верзије приказане су у тренутку писања пројекта биле последње стабилне верзије):

Оперативни систем:

1. Microsoft Windows XP – најприхватљивија опција због широке базе већ обучених корисника, као и због посебног уговора који је републичка администрација потписала са корпорацијом Microsoft

Канцеларијски програмски пакет:

1. Microsoft Office 2003
2. OpenOffice 2.0.3.

Web browser

1. Mozilla Firefox 1.5.0.4 – првенствено због изразито добре компатибилности са важећим W3C HTML стандардима, брзине приказа страница, као и мање осетљивости на нападе
2. Internet Explorer 6 (тренутно последња стабилна верзија, верзија 7 очекује се ускоро)

Email клијентска апликација

1. Microsoft Outlook – уколико се користи пакет MS Office – првенствено због додатних функционалности

2. Mozilla Thunderbird / MS Outlook Express – нуде у суштини сличне особине, други је досад био нешто више подложен нападима

3.2. Серверски оперативни систем

Док је код десктоп оперативних система одређену препоруку лако донети, код оперативних система за серверске конфигурације, проблем је нешто комплекснији. У овом случају од кључне важности су сигурност система, поузданост рада система, могућност покретања одређених серверских апликација, а од немале важности су и цене појединих оперативних система за ову намену, које су у правилу знатно веће од десктоп оперативних система. Код ових система развијеност графичког интерфејса није од кључне важности, јер би, бар у начелу, овакве системе требали одржавати обучени професионалци.

У погледу сигурности система Unix и системи базирани на сличним концептима (разне врсте Linux дистрибуција) имају више подржаних сигурносних механизма (погледати поглавље 2.2) него што је то случај са Windows оперативним системом. Осим тога, како серверска верзија Windows система дели највећи део платформе са десктоп верзијама, подложна је скоро свим нападима којима су Windows-и и иначе највише изложени од свих осталих ОС, због саме чињенице њихове популарности и распрострањености. Због сигурносних питања, као и због цене набавке, у овом случају препоручљиво је да се за оперативни систем на серверским конфигурацијама користи нека од дистрибуција Linux-а. Осим директног ефекта који овај избор доноси на сигурности и стабилности самог оперативног система (Linux системи имуни су на све вирусе развијене тако да циљају Windows платформе, а број вируса који циљају Linux платформе је скоро занемарљив), постиже се и додатни ефекат јер већина корисника није претерано упозната са овим оперативним системом, те се најчешће избегавају и ситуације промене подешавања сервера од стране некомпетентних особа.

Као битна карактеристика мора се узети и понуђена техничка подршка, као и цена и доступност тренинг курсева за администраторе. Подршку за Microsoft-ове платформе обезбеђује Microsoft преко свог локалног огранка Microsoft Srbija i Crna Gora (још увек званични назив у моменту писања пројектне документације) и преко мреже својих партнера. Обука се такође врши код ауторизованих партнера (<http://www.microsoft.com/scg/partneri/directory/skolskicentri.mspx>).

За Linux платформе стање је нешто другачије. Различите дистрибуције Linux-а заступају различите фирме у Србији. Нпр. за дистрибуцију RedHat Linux ауторизовани партнер је Сага д.о.о. Београд. Дистрибуцију Novell SUSE Linux у Србији заступа и дистрибуира MP-Soft (<http://www.mp-soft.co.yu/>) који нуди и тренинг курсеве. qSTC центар Србија (http://www.qstc.co.yu/contact/serbia_montenegro/address.asp) организује SUSE Linux академију за различите нивое корисника. компанија IBM је у сарадњи са Факултетом организационих наука у Београду основала Linux центар који нуди едукацију из области примене Linux оперативних система. База Linux корисника у Србији је првенствено концентрисана у ИТ компанијама, Интернет провајдерима, те у академским институцијама. Заступљеност разних дистрибуција Linux-а у овим круговима је значајна, а узимајући у обзир заинтересованост великих компанија, могуће је очекивати и даљи развој тржишта Linux дистрибуција и подршке за њих. Осим комерцијалних дистрибуција постоје и потпуно бесплатне дистрибуције (FedoraCore коју спонзорише RedHat, OpenSUSE као бесплатна верзија SUSE Linux-а).

Потребно је нагласити да је без обзира на дистрибуцију основа Linux оперативног система заједничка, те је и основна обука иста за све дистрибуције које деле исту верзију кернела.

При разматрању прихватљивости Linux решења треба узети у обзир да је пожељно користити најновију стабилну верзију кернела, која нуди подршку за највећи број нових хардверских компоненти, као и да се изабере она дистрибуција за коју постоје одређени вид подршке и обуке. Трошкови обуке Linux администратора или повременог ангажовања Linux администратора су у правилу већи од трошкова обуке администратора Windows система, али се показује да је потреба за одржавањем и администрирањем Linux сервера далеко мања, просечан *up-time* између две интервенције је нешто дужи код Linux сервера. Иако у нашој земљи нема много анализа остварених уштеда или додатних трошкова при преласку на Linux оперативне системе, једно од искустава тренутно доступно је искуство у делу ЕПС-а – РБ Колубара, које је набавком рачунара са Linux конфигурацијама остварио иницијалну уштеду (по интерној рачуници) око 150.000 долара само на рачун лиценци. Детаљније информације са анализом структуре трошкова и трошковима тренинг курсева су доступне у чланку [ekonomistLinux].

Осим Windows и Linux решења могуће је разматрати и коришћење Unix оперативних система који скоро све велике компаније (IBM, Sun, HP) нуде и преко својих партнера у Србији. Осим бесплатно доступних BSD верзија и Sun Solaris-a, код већине ових система кључну улогу игра цена лиценци.

Најважнији критеријуми за процену погодности оперативног система за употребу на серверским системима, за потенцијално занимљиве системе, изведени на основу података у табелама дати су у табели 3.1.

На основу података у табели 3.1. уз узимање у обзир података о изложености одређеног оперативног система нападима, због озбиљнијих сигурносних пропуста (38% - 50% сигурносних пропуста на Windows 2003 Server-у спада у категорију озбиљних, наспрам 10% код Linux-a) [SecurityReport, US_CERT]. препоруке за избор оперативног система на серверским конфигурацијама су следеће:

1. Linux – одговарајућа дистрибуција за коју постоји подршка и обука, или њена бесплатна верзија (у том случају подршка није обезбеђена)
2. Windows 2003 Server – уз неопходну пажњу и правовремену реакцију на откривене пропусте
3. Solaris – По техничким и сигурносним критеријумима овај систем заслужује и бољу позицију, али му се као мана може узети ипак мала заступљеност и потреба за специфичном обуком

Назив	Постојање подршке	Оквирна цена	Сигурносни системи				
			Контрола приступа ресурсима (бр. врста)	Изолација подсистема	Интегр. firewall	Заштита од извршења нежељеног кода (хардверска 1, софтверска 1, обе 2)	Енкрипц. система датотека
FreeBSD	Само путем корисничких група	Бесплатан	Постоји (3)	Постоји	Постоји		Постоји
OpenBSD	Само путем корисничких група	Бесплатан	Постоји (1)	Постоји	Постоји	2	Постоји
Linux	Постоји	Бесплатан (одређене дистрибуције се плаћају уз обезбеђену техничку подршку нпр. RedHat 349\$ - 2400\$)	Постоји (3)	Постоји	Постоји	1	Постоји
Solaris	Постоји	Бесплатан (разне опције техничке подршке се плаћају)	Постоји (2)	Постоји	Постоји	2	Постоји
Windows 2003 Server	Постоји	999\$ (за 5 клијената)	Постоји (1)	Не постоји	Постоји	1	Постоји

Табела 3.1 Кључни критеријуми за процену погодности оперативног система

3.3. Базе података

Код система за управљање базама података, препорука за избор одговарајућег система је доста сложена одлука. Ови системи би, по правилу, требали да обезбеде основне потребе за похраном, претраживањем и ажурирањем података за дужи временски период, као и да омогуће проширење апликација за нове и надлазеће услуге. Осим тога морали би да имају поуздан систем *back-up*-а података, као и поуздане системе за контролу приступа. Скоро сви данас доступни RDBMS су са овог становишта у врло високом степену развијености. Карактеристично за комерцијално доступне RDBMS системе је да их одликује врло висока цена (цене се обично формирају по броју клијената које сервер може да опслужи, броју процесора на којима се покреће и сл).

При избору RDBMS система било би пожељно водити рачуна о неколико особина:

- Пожељно је да исти RDBMS систем постоји у верзијама за различите оперативне системе. Ово омогућава да се при евентуалној одлуци о промени ОС на серверској конфигурацији избегне неопходност пребацивања података у други тип RDBMS система. Иако данас постоји велики број алата који омогућава извоз и увоз података из једног у други систем, овакав поступак, скоро по правилу, никад није једнозначан и неопходни су додатни кораци како би сви подаци и све релације у новом окружењу били валидни.
- За било коју озбиљну примену апсолутно је неопходно да RDBMS систем у потпуности поштује све основне концепте релационог модела, да подржава трансакције, а у нашем окружењу и да обавезно подржава рад са Unicode карактерима.
- Пожељно је да RDBMS систем подржава и неке напредније концепте, као што су сложенији индекси (што омогућава брже извршавање неких сложенијих претрага), DB објекти (као нпр. тригери, функције – *stored procedure*).
- За велике, дистрибуиране системе у којима се очекује рад са врло великим количинама података, којима се при томе често приступа, пожељна је особина могућност партиционисања базе.
- Пожељно је да RDBMS систем има уграђен одређени вид подршке за рад са подацима у XML формату.

Из ове анализе је видљиво да би нпр. избор MS SQL сервера, који има врло добре особине (осим могућности партиционисања) форсирао избор оперативног система Windows, јер ова база не може функционисати нити на једном другом оперативном систему. Избор MySQL сервера чији је основни број верзије мањи од 5, није погодан за сложене апликације, јер у основној верзији овај систем не подржава чак ни неке од круцијалних концепата релационог модела (као што су референцијални интегритет података). Неопходно је нагласити да се ово запажање не односи на најновију верзију MySQL сервера која, уз услов да се користи InnoDB механизам за похрану података (*storage engine*), подржава све битне концепте релационог модела. Што се тиче сложенијих индексних структура из табеле 2.9. је видљиво да само мањи број RDBMS система у већој мери подржава овакве индексе (PostgreSQL, Informix, Oracle, ADS, делимично MSSQL итд). Додатне DB објекте, у већој или мањој (SQLite и

MySQL) мери подржавају скоро сви разматрани RDBMS системи. Подршка за шеме партиционирања базе је међутим знатно мања (табела 2.11).

На основу података из табела могуће је донети одређене закључке:

- *Open source* RDBMS PostgreSQL (при томе и *freeware*) издваја се по својим карактеристикама, јер у скоро свим аспектима има подржане све концепте који су анализирани.
- Од комерцијалних система у складу са својом репутацијом Oracle нуди највише, а по броју имплементираних концепата следе га Informix, MSSQL Server и DB2.
- Од *open source* и *freeware* решења осим већ поменутог PostgreSQL, већину концепата (осим могућности партиционисања и напреднијих индекса) солидно подржавају и MAX DB (SAP DB) и Firebird.
- Највећи недостатак MSSQL Servera је управо ограничење на Windows оперативни систем.

На основу ових запажања могуће је дати одређене смернице. Као најпожељније решење, и решење које ће на дуже стазе задовољавати потребе система, намеће се PostgreSQL. Осим врло добрих особина, овај производ је настао као *open source*, и доступан је бесплатно. Као други валидан избор јавља се MAX DB, који може бити нарочито интересантан, јер поседује могућност рада у тзв. Oracle компатибилном режиму. Следећи избор би могао бити Firebird. Један недостатак овог система је тај што је при конфигурисању конекција на ову базу неопходно знати и тачан положај фајла који представља тражену базу у фајл систему, што осим што је непрактично, може и непотребно експонирати детаље хијерархије фајл система. MySQL, иако врло често коришћен RDBMS систем, није претерано погодан за употребу у сложенијим и захтевнијим системима, уколико се не користи најновија верзија 5.x и уколико се користи подразумевани механизам похране података MyISAM. Уколико се користи најновија верзија MySQL сервера, уз употребу InnoDB механизма за похрану података и ова база задовољава већину кључних захтева савремених RDBMS система. Иако у начелу бесплатан, верзије MySQL 5.0 Pro Certified Server и MySQL 5.0 Cluster се нуде уз одређене лицендне накнаде (4 различита пакета), које обезбеђују различити степен подршке и едукацију. Ове верзије су намењене првенствено великим системима и гарантују велику стабилност и брз одзив сервиса за подршку.

Код комерцијалних RDBMS система највећи недостатак је највероватније висока цена. Иако се при набавци ових система цена уговара од случаја до случаја, по правилу је врло висока. Oracle је и даље лидер на пољу комерцијалних RDBMS система. MS SQL Server се такође појављује као прихватљива опција, али треба водити рачуна да овакав избор има за последицу и форсиран избор оперативног система на датој серверској конфигурацији.

На основу свега наведеног може се креирати табела критеријума за оцену погодности одређене базе. Сви ови критеријуми већ су наведени у табелама 2.4 - 2.11. Критеријуми су дати у табели 3.2.

Свака од водећих компанија које производе комерцијалне RDBMS системе (Oracle, Microsoft, IBM) има своје овлаштене представнике у Србији, тако да постоји обезбеђена техничка подршка, као и образовање корисника. OpenSource решења, иако у раширеној употреби, немају организовану мрежу дистрибутера и подршке.

На основу података у табели 3.2 могуће је донети следеће препоруке:

а) Од бесплатно доступних (*open source*) база података по својим карактеристикама се издвајају:

1. PostgreSQL – једини озбиљан недостатак је непостојање уграђене подршке за обраду XML података, али она постоји као додатни модул
2. MAXDB (раније SAP DB) – недостатак непостојање подршке за XML, као и непостојање подршке за партиционисање, што може бити проблем з а очекиване велике количине података
3. MySQL ver 5.x, кључни недостатак – мора се водити рачуна који тип система за чување података се користи, непостојање подршке за партиционисање, што може бити проблем за очекиване велике количине података

б) Од комерцијално доступних база података издвајају се следеће:

1. Oracle
2. MS SQL Server 2005
3. DB 2 (или Informix)

Напомена: Уколико су примарно опредељење комерцијалне базе података, ове базе нуде врло сличан скуп особина, тако да кључну улогу у избору може имати договорена цена. Иако постоје оквирне цене, врло често су при закључивању уговора, уговара посебна цена у зависности од случаја до случаја.

Назив	Подршка за рад на више платформи	Подржани концепти релационог модела	Подршка за напредније коцепте (домен, тригери, stored procedure функције и сл.)	Подршка за партиционисање	Уграђена подршка за рад са XML-ом
Adaptive Server Enterprise	Постоји	Да	Постоји	Постоји	Постоји
ADS	Постоји	Да	Постоји	Постоји	-
DB2	Постоји	Да	Постоји (све сем домен огрницења)	Постоји	Постоји
Firebird	Постоји	Да	Постоји	Не постоји	Не постоји
HSQldb	Постоји	Да	Постоји	Постоји	Не постоји
H2	Постоји	Да	Постоји	Постоји	Не постоји
Informix	Постоји	Да	Постоји	Постоји	Постоји
Ingres	Постоји	Да	Постоји	Постоји	?
InterBase	Постоји	Да	Постоји	Не постоји	Делимично, само експорт
MaxDB (SAP DB)	Постоји	Да	Постоји	Не постоји	Не постоји
Microsoft SQL Server	Не постоји	Да	Постоји	Постоји	Постоји
MySQL	Постоји	Условно	Постоји	Не постоји	Делимично
Oracle	Постоји	Да	Постоји	Постоји	Постоји
PostgreSQL	Постоји	Да	Постоји	Постоји	Ограничено (екстерни модул)
Pyrrho DBMS	Постоји	Да	Постоји	Постоји	Не постоји
SQLite	Постоји	Делимично	Постоји	Постоји	Не постоји
Teradata	Постоји	Да	Постоји	Постоји	Постоји

Табела 3.2 Кључни критеријуми за процену погодности система за управљање базама података

3.4. Апликациони сервери и CMS системи

На пољу апликационих сервера кључни избор се заправо своди на избор .NET или J2EE технологије. Ове две технологије захваљујући Web сервисима могу међусобно комуницирати, па и сам избор не мора бити у потпуности искључив, могуће је део јавних сервиса развити на .NET платформи, а део на J2EE платформи. Да ли је такво решење о паралелном раду на две платформе и економски прихватљиво је питање за додатну анализу.

При одабиру платформи треба водити рачуна о следећим проблемима:

- Оријентацијом само на .NET платформу имплицира се и избор Windows оперативних система, као неопходне подлоге за рад .NET окружења. Барем део серверских система у овом случају мора бити на Windows платформи. У суштини не постоји могућност избора апликационог сервера, тренутно је једина проверена платформа Microsoft IIS. Овакво решење везује све апликације само на Microsoft решења, јер је .NET, барем још увек, немогуће мигрирати на било коју другу платформу.
- Оријентација на J2EE платформу имплицира усмеравање само на Java програмски језик. Ово подразумева да сви програмери, на садашњим и будућим пројектима, морају познавати овај програмски језик. С друге стране постоји велика могућност избора како комерцијалних тако и *freeware* апликационих сервера различитих перформанси. Осим самих апликационих сервера постоји и могућност избора између различитих комерцијалних и *freeware* развојних окружења (што није случај са .NET платформом).

По питању избора платформе не може се саставити једноставна табела критеријума за поређење (осим оних већ датих у табели 2.14).

Највероватнији правилан пут би био обезбеђење подршке за обе платформе. Такође врло чест начин интегрисања ових платформи јесте коришћење J2EE апликационих сервера, док се самосталне клијентске апликације реализују било као Java или као .NET апликације. Обе платформе имају врло широку базу програмера у нашој земљи, и подржава их велики број ИТ компанија, а постоји и већи број компанија и образовних установа које нуде школовање/курсеve за рад са овим платформама.

Препорука која се може дати за избор развојне платформе је да су обе технологије подједнако прихватљиве. Одређена предност би се могла дати Java технологији првенствено из разлога могућности развоја и имплементације софтвера на различитим хардверско-софтверским платформама, што се може очекивати у великим системима као што је целокупан систем еУправе на територији Војводине. Ово наравно не искључује коришћење .NET технологије тамо где су већ инсталирани Windows серверски системи.

Концентришући се на J2EE платформу, јавља се и питање избора апликационог сервера. Постоји велики број комерцијално доступних решења које имају J2EE сертификацију, али долазе у прилично високом ценовном рангу. Већина ових сервера су способни да издрже огромна оптерећења, у стању су да распоређују оптерећење на више паралелних сервера и сл. Као најзначајнији и најзаступљенији сервери у овом сегменту јављају се IBM WebSphere, BEA WebLogic, ATG Dynamo, Macromedia JRun. С друге стране постоји одређени број J2EE сертифицираних *open source* серверских пројеката који се нуде као бесплатно решење. Нека од ових решења карактеришу се

такође врло добрим особинама у погледу обраде великих броја захтева, такође поседују могућност повезивања више сервера за паралелни рад. Као једно од најприхваћенијих решења и апсолутно најзаступљенији *open source* апликациони сервер фигурише JBoss. JBoss се намеће као најприхватљивије решење и за потребе јавних сервиса еВојводине, нудећи врло добре перформансе, велику стабилност. Како је овај сервер и J2EE сертифициван, апликације које су покренуте под овим сервером, могуће је без већих (или чак никаквих) модификација преселити у случају потребе и на други J2EE сертифициван сервер, без обзира да ли је у питању комерцијални, или *freeware* сервер.

Препоруке за избор апликационог сервера у овом случају су

а) Бесплатна *open source* решења

1. JBoss – најраширенији и врло добро документован бесплатан *open source* сервер, J2EE сертифициван и врло добрих перформанси
2. Jakarta Tomcat – референтна развојна платформа

б) Комерцијално доступни сервери – карактеришу их добре перформансе при врло високим оптерећењима, али и врло висок ценовни ранг. Овде се издвајају:

1. IBM WebSphere
2. BEA WebLogic

Препорука је да се при избору апликационог сервера првенствено иде на прво решење (JBoss)

Са становишта CMS система, такође постоји могућност избора између комерцијалних решења и *open source* решења. Критеријуми на основу којих су вредновани посматрани системи дати су у табели 2.19. Од комерцијалних решења, као проверен систем се показао CMS сервер Magnolia, који је у основној варијанти бесплатан за коришћење, али све напредније опције подразумевају лиценцирање. Са друге стране, велики број *open source* решења је у употреби, чак и у оквиру решења за *еУправе*, чиме би се могла оправдати одлука да се за CMS сервер употреби неко од решења из ове групе. У том случају, Apache Lenya се издваја због могућности које пружа, а у које спадају између осталог и подршка за интернационализацију, подсистем за криптографску заштиту приступа садржају и квалитетан подсистем за аутентификацију и ауторизацију корисника.

На серверским платформама на којима би се користили и Windows оперативни системи као и .NET окружење, улогу апликационог сервера обавља Microsoft IIS, а као систем за управљање садржајима може се користити Microsoft Content Management Server.

За успешан развој еУправе на територији АП Војводине, неопходно је да покрајинска управа, након спроведене анкете којим се утврђује тренутно, почетно стање свих ресурса битних за развој, одржавање и унапређење оваквог система, успостави одређени механизам којим би се омогућио континуирани мониторинг стања у органима управе и локалне самоуправе. Овакав мониторинг би омогућио правовремене акције чији би циљ био праћење актуелних трендова у подручју еУправе (усвајање нових стандарда и сл), и смањење стопе технолошког застаревања система.

Литература:

Документа са наведених сајтова су преузета у периоду фебруар-април 2006. године

[OASIS] <http://www.oasis-open.org/home/index.php>

[IDABC] <http://europa.eu.int/idabc/>

[IETF] <http://www.ietf.org/>

[W3C] <http://www.w3c.org>

[ECMA] <http://www.ecma-international.org/>

[OpenDoc] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office

[Gartner_ISO-ODF] -

http://www.gartner.com/resources/140100/140101/iso_approval_of_oasis_opendo_140101.pdf

[OSS_in_eGov] Open-source software in e-government, Analysis and recommendations drawn up by a working group under the Danish Board of Technology, October 2002, <http://europa.eu.int/idabc/en/document/4617/5676>

[wikipedia-OS] http://en.wikipedia.org/wiki/Comparison_of_operating_systems

[BSDRoundup] <http://www.unixreview.com/documents/s=7750/uni1040336129456/>

[wikipedia-RDBMS] http://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems

[XMLEnabled] <http://www.rpbouret.com/xml/ProdsXMLEnabled.htm>

[wikipedia-eGov] <http://en.wikipedia.org/wiki/EGovernment>

[farley] http://www.oreillynet.com/pub/a/oreilly/java/News/farley_0800.html

[FLOSS] <http://flossproject.org/>

[FLOSSReport] <http://flossproject.org/report/index.htm>

[Danish-Software-Strategy] <http://europa.eu.int/idabc/en/document/4618/5676> ili
<http://ec.europa.eu/idabc/servlets/Doc?id=21859>

[wikipedia-comparison-CMS] http://en.wikipedia.org/wiki/Comparison_of_content_management_systems

[gilbane-report] <http://www.gilbane.com>

[aplaws] <http://www.aplaws.org.uk/home/index.php>

[agimo] <http://www.agimo.gov.au/>

[gsb] <http://www.government-site-builder.de/>

[plone] <http://plone.org>

[zope] <http://www.zope.org>

[devOS-CMS] Abhijeet Chavan, Developing An Open Source Content Management Strategy for E-Government, 42nd Annual Conference of the Urban and Regional Information Systems Association, URISA 2004, November 9th, 2004, Reno, Nevada

[ejisdc] The Electronic Journal on Information Systems in Developing Countries, (2003) 14, 4, 1-20,
<http://www.ejisdc.org>

[ekonomistLinux] <http://www.ekonomist.co.yu/magazin/ebit/26/pr4/pr4.htm>