

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matej Pintar

**IMPLEMENTACIJA TELEFONSKEGA IMENIKA V SISTEM  
MS SHAREPOINT SERVER 2007**

DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Mentor: prof. dr. Saša Divjak

Ljubljana, 2008



## **Zahvala**

Zahvaljujem se prof. dr. Saši Divjaku za mentorstvo, pomoč in informiranje pri izdelavi diplomskega dela. Za lektoriranje diplomskega dela se zahvaljujem ga. Miri Prevc. Na tem mestu bi se rad zahvalil še domačim in vsem ostalim, ki so mi v času študija in pisanja diplomske naloge kakorkoli pomagali in mi stali ob strani.

Hvala vsem!



# Kazalo

Povzetek .....	1
Abstract.....	3
1 Uvod.....	5
2 Uporabljene tehnologije .....	7
2.1 Microsoft Office SharePoint Server 2007 .....	7
2.1.1 Sestavni deli MOSS 2007.....	8
2.1.1.1 Ponudnik skupnih storitev MOSS 2007 .....	9
2.1.1.2 Funkcije MOSS 2007 .....	10
2.1.2 Arhitektura MOSS 2007.....	11
2.1.2.1 Logična arhitektura .....	11
2.1.2.2 Fizična arhitektura.....	12
2.1.3 ASP .NET stran v MOSS 2007 .....	13
2.1.3.1 Glavna stran.....	14
2.1.3.2 Vsebovana stran .....	15
2.1.3.3 Obravnava zahteve za .aspx stran v MOSS 2007.....	17
2.1.3.4 Stran s spletnimi gradniki.....	18
2.1.3.5 Spletni gradniki .....	20
2.1.4 Potek dela .....	20
2.1.5 Časovno opravilo.....	21
2.2 XML.....	22
2.2.1 XSLT .....	23
3 Implementacija telefonskega imenika v MOSS 2007 .....	27
3.1 Spletni gradnik Stiki .....	31
3.2 Tok dogodkov končnih uporabnikov:.....	36
4 Posodabljanje stikov v telefonskem imeniku .....	39
4.1 Rešitev s potekom dela .....	39
4.1.1 Potek dela ImenikUpdaterWF .....	39
4.2 Rešitev s časovnim opravilom .....	41
4.2.1 Časovno opravilo ImenikUpdaterJob .....	41
4.3 Primerjava obeh rešitev .....	43

5	Zaključek .....	45
	Uporabljeni viri .....	47
	Izjava o avtorstvu .....	49

## Kazalo slik

Slika 1: Umestitev MOSS 2007 v skladu tehnologij.....	7
Slika 2: Komponente MOSS 2007 .....	8
Slika 3: Logična arhitektura MOSS 2007.....	11
Slika 4: Fizična arhitektura MOSS 2007 .....	12
Slika 5: Povezava glavne strani z vsebovano stranjo .....	13
Slika 6: Primer napotka za glavno stran .....	14
Slika 7: Primer enostavne glavne strani .....	15
Slika 8: Primer napotka vsebovane strani, ki vsebuje povezavo na glavno stran .....	15
Slika 9: Primer enostavne vsebovane strani .....	16
Slika 10: Glavna stran in vsebovana stran med izvajanjem .....	16
Slika 11: Obravnavanje poizvedbe strani v MOSS 2007 .....	18
Slika 12: Stran s spletnimi gradniki.....	19
Slika 13: Vsebovana stran z enim vsebovalnikom in dvema območjema za spletne gradnike. ....	19
Slika 14: Razporeditev območij spletnih gradnikov v MOSS 2007.....	20
Slika 15: Splošna oblika oznake elementa .....	23
Slika 16: Primer dokumenta XML .....	23
Slika 17: Primer dokumenta XSLT .....	24
Slika 18: Formatiran izhod dokumenta XML povezanega z dokumentom XSLT.....	25
Slika 19: Pogled na seznam Stiki .....	27
Slika 20: Ustvarjanje novega elementa (stika) v seznamu Stiki.....	28
Slika 21: Prazna vnosna maska za ustvarjanje novega elementa (stika) v seznamu Stiki .....	29
Slika 22: Osnovna stran telefonskega imenika na portalu »Središče EG«.....	30
Slika 23: Spletni gradnik Stiki.....	32
Slika 24: Ogrodje spletnega gradnika DataFormWebPart .....	33
Slika 25: Oblika poizvedbe za iskanje po telefonskem imeniku .....	34
Slika 26: XSLT predloga, po navodilih katere se prikažejo rezultati.....	36
Slika 27: Zgradba poteka dela ImenikUpdaterWF (povezave med stanji).....	39
Slika 28: Stanja poteka dela ImenikUpdaterWF .....	40

## Seznam uporabljenih kratic in simbolov

### **MOSS 2007**

Microsoft Office SharePoint Server 2007.

### **WSS** (angl. Windows SharePoint Services)

Storitve Windows SharePoint-a.

### **ASP .NET**

Microsoftova tehnologija za razvijanje dinamičnih spletnih strani.

### **IIS** (angl. Internet Information Services)

Microsoftov spletni strežnik. IIS je paleta internetnih storitev za strežnike z operacijskim sistemom Microsoft Windows.

### **SSP** (angl. Shared Services Provider)

Ponudnik skupnih storitev v MOSS 2007.

### **BDC** (angl. Business Data Catalogue)

Storitev, ki jo ponuja MOSS 2007, in omogoča povezovanje MOSS 2007 s podatkovnimi bazami ali spletnimi storitvami.

### **KPI** (angl. Key Performance Indicator)

Gljučni dejavniki uspeha so kazalniki, s katerimi organizacija opredeli in meri doseganje strateških ciljev.

### **HTML** (angl. Hyper Text Markup Language)

Označevalni jezik za oblikovanje spletnih strani.

### **XML** (angl. Extensible Markup Language)

Razširljiv označevalni jezik.

### **XSL** (angl. Extensible Stylesheet Language)

Družina jezikov, namenjenih pretvorbi dokumentov XML.

### **XSLT** (angl. XSL Transformations)

Jezik iz družine jezikov XSL za transformacijo dokumentov XML v druge dokumente.

### **DFWP** (angl. DataFormWebPart)

Spletni gradnik za izbiranje, oblikovanje in prikazovanje podatkov.

### **CAML** (angl. Collaborative Application Markup Language)

Označevalni jezik, ki temelji na jeziku XML in se uporablja v tehnologijah WSS in MOSS 2007 za definiranje strani, seznamov, vsebovanih polj, obrazcev in pogledov.

## **Povzetek**

Diplomska naloga opisuje vzpostavitev aplikacije telefonskega imenika v intranetnem portalu, ki je bil razvit na podlagi tehnologije Microsoft Office SharePoint Server 2007.

Rešitev omogoča končnim uporabnikom portala, da preko iskalnika telefonskega imenika na hiter in enostaven način pridejo do informacij, ki jih ta ponuja. Uredniku telefonskega imenika in administratorju pa ponuja še funkcionalnosti dodajanja novih in brisanja obstoječih stikov ter funkcionalnosti pregledovanja in urejanja podrobnosti posameznega stika. Za uporabnike nevidna funkcionalnost, pa je funkcionalnost posodabljanja stikov v telefonskem imeniku.

Vsebina diplomske naloge predstavi tehnologije in principe delovanja, ki so pomembni za razumevanje implementacije. Opisani so prijemi, ki so uporabljeni pri razvoju telefonskega imenika. Predstavljeno je tudi delovanje telefonskega imenika in glavni tokovi dogodkov končnega uporabnika pri uporabi telefonskega imenika. Na koncu pa sta podana in med seboj primerjana dva načina posodabljanja stikov v telefonskem imeniku. V prvem načinu je posodabljanje realizirano s pomočjo poteka dela. V drugem načinu pa je za posodabljanje uporabljeno časovno opravilo.

### **Ključne besede:**

Telefonski imenik, MOSS 2007, spletni gradnik, XSLT, časovno opravilo, potek dela



## **Abstract**

The bachelor's thesis presents a phone directory implementation into intranet portal based on Microsoft Office SharePoint Server 2007.

The Phone directory enables portal end-users to quickly and easily access its information through its search bar. Phone directory editor and administrator have also some additional features. They can add or delete items in phone directory and they can view or edit details of each item in phone directory. There is also one, for all users except for administrator, invisible feature that updates each item details. It assures, that the information that phone directory provides is accurate.

The thesis introduces the technologies and principles important for understanding the implementation. It describes how the phone directory application is constructed. Furthermore, it presents how the phone directory works and what are the most frequently used complete sequences of events, described from the point of view of the end-user (Use cases). At the end, two possible ways of updating the phone directory are presented and compared. The first one uses Workflow for serving its purpose and the second one uses Timer Job.

### **Key words:**

Phone directory, MOSS 2007, web part, XSLT, Timer Job, Workflow



# 1 Uvod

Podjetje Elektro Gorenjska se ukvarja z distribucijo in proizvodnjo električne energije. V podjetju je zaposlenih več kot 300 ljudi, od katerih večina svoje delo opravlja na računalniku.

V preteklosti so zaposleni skupne dokumente, predloge, preglede, razne aplikacije, druge datoteke in oblike informacij shranjevali na datotečnih strežnikih. Kaj hitro se je zgodilo, da se je določeni dokument na strežniku pojavil večkrat. Zaposleni niso imeli mesta, ki bi jim ponudilo informacije, ki so zanje v tistem trenutku aktualne. Zato smo se v podjetju Elektro Gorenjska odločili za postavitev intranetnega portala na platformi Microsoft Office SharePoint Server 2007. Poimenovali smo ga »Središče EG«. Končni cilj postavitve intranetnega portala v podjetju Elektro Gorenjska je, da bi uporabniki hitro in na enostaven način prišli do pomembnih informacij, ki jih potrebujejo za delo. S tem bi se povečala njihova učinkovitost in s tem uspešnost podjetja.

Ena glavnih zahtev, ki sem jo moral najprej realizirati, pa je tudi implementacija internega telefonskega imenika v portal »Središče EG«. Velik del komunikacije zaposlenih med seboj in zaposlenih z zunanjimi sodelavci poteka prav preko telefona. Zato je zelo pomembno, da telefonski imenik deluje hitro in še več, da posreduje ažurne informacije.

Pred objavo telefonskega imenika, katerega implementacija je opisana v diplomski nalogi, sta se hkrati uporabljali dve aplikaciji za telefonski imenik. Informacije, ki jih je ponujala ena aplikacija se niso popolnoma ujemale z informacijami druge, saj sta aplikaciji črpali podatke iz različnih virov. Vsaka od obeh aplikacij pa je imela tudi svojega urednika. Zaradi navedenih razlik se je v podjetju že dalj časa pojavljala potreba po enotni rešitvi.

Telefonski imenik v okviru portala »Središče EG« se je že začel uporabljati, obe prejšnji različici pa smo umaknili iz uporabe. Da bi bila nova aplikacija telefonskega imenika med uporabniki dobro sprejeta, je bilo potrebno v novo aplikacijo vključiti čim več dobrih lastnosti prejšnjih dveh aplikacij, hkrati pa skušati odpraviti čim več slabih.

Drugo poglavje diplomske naloge opisuje uporabljene tehnologije in podrobneje nekatere segmente, ki sem jih pri realizaciji svoje rešitve uporabil. Poglavje vsebuje opis platforme Office SharePoint Server 2007, pregled zgradbe strani ASP .NET, opis poteka dela in časovnega opravila ter predstavitev jezika XML v povezavi z jezikom XSLT.

Tretje poglavje opisuje implementacijo telefonskega imenika. V tem poglavju sem predstavil delovanje telefonskega imenika in prijeme, ki sem jih uporabil v svoji rešitvi. Opisani so glavni tokovi dogodkov za končnega uporabnika in pa dodatne funkcionalnosti, ki jih imata urednik telefonskega imenika in administrator portala »Središče EG«.

V četrtem poglavju sta opisani dve rešitvi za posodabljanje stikov v telefonskem imeniku. V prvi rešitvi je uporabljen potek dela s stanji, v drugi pa časovno opravilo. Poglavje se konča s primerjavo med obema rešitvama.

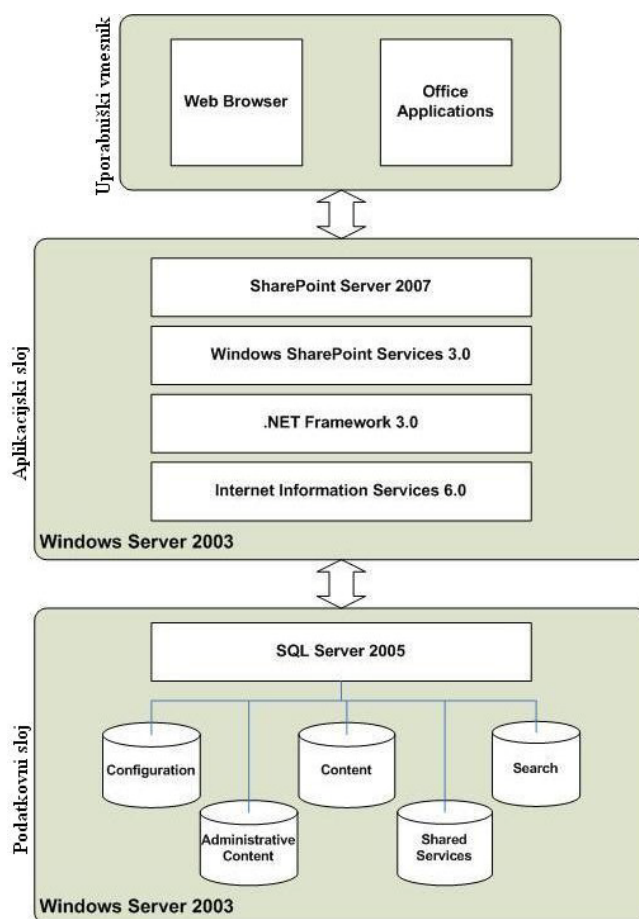


## 2 Uporabljene tehnologije

Pravilna izbira tehnologije igra pomembno vlogo pri doseganju zastavljenih ciljev nekega projekta. V podjetju Elektro Gorenjska d.d. smo se odločili za tehnologijo Microsoft Office SharePoint Server 2007, saj nam ponuja vse, kar potrebujemo za uvedbo enotne in centralne informacijske rešitve.

### 2.1 Microsoft Office SharePoint Server 2007

Microsoft Office SharePoint Server 2007 (MOSS 2007) [1] je robustna in stabilna platforma, ki uporabnikom omogoča povezovanje, komunikacijo in dostop do ključnih informacij oziroma dokumentov. Uporablja se za gostovanje spletnih strani, preko katerih se dostopa do skupnih delovnih prostorov in dokumentov. Prav tako pa je produkt specializiran tudi za gostovanje različnih spletnih aplikacij, kot so na primer baze znanja in forumi. Namen SharePoint-a ni nadomestiti celotni datotečni strežnik, ampak samo tisti del, ki je skupen večjemu številu uporabnikov.



Slika 1: Umestitev MOSS 2007 v skladu tehnologij

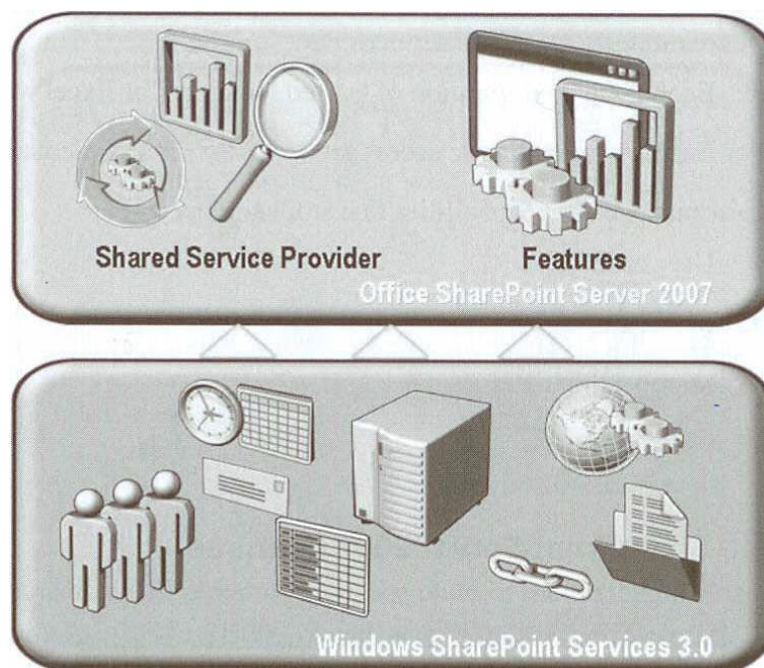
Tako MOSS 2007 kot tudi storitve Windows SharePoint Services (WSS) 3.0 se močno opirata na veliko produktov in tehnologij, ki jih ponuja Microsoft, vključno s strežnikom SQL (angl. SQL Server), ogrodjem .NET (.NET Framework) in strežnikom Windows Server 2003. MOSS 2007 je zgrajen na podlagi ogrodja .NET 3.0 (angl. .NET Framework 3.0) in Windows Workflow Foundation. Lahko deluje v povezavi z ASP .NET 1.1 in 2.0 spletnimi gradniki (angl. WebPart). SharePoint Server teče na storitvah Internet Information Services (IIS) 6.0 ali novejših, deluje pa tako v povezavi z Microsoft SQL Server 2000, kot tudi v povezavi z SQL Server 2005.

### 2.1.1 Sestavni deli MOSS 2007

Rešitve MOSS 2007 vsebujejo mnogo funkcionalnosti (angl. feature) ponujenih s strani storitev Windows SharePoint Services 3.0 nad katerimi je zgrajen. Na primer: sezname, knjižnice dokumentov, povezave, kontakti, dogodki, objave, koledarji in pametne poslovne rešitve (angl. Business Intelligence Solutions). Možnost, da lahko dostopamo do teh informacij preko spletnega brskalnika in preko mobilnega telefona, nam zagotavlja uporabnost in prilagodljivost teh rešitev.

MOSS 2007 je zgrajen nad temeljni storitev WSS tako, da ponuja dve glavni komponenti, ki pomagata organizacijam zgraditi močne in specifične poslovne rešitve [2]:

- Ponudnik skupnih storitev MOSS 2007 (angl. MOSS 2007 Shared Service Provider),
- Funkcije MOSS 2007 (angl. MOSS 2007 Features).



Slika 2: Komponente MOSS 2007

### 2.1.1.1 Ponudnik skupnih storitev MOSS 2007

SSP (angl. Shared Services Provider) [2] izvaja naloge, ki so lahko skupne različnim mestom in zbirkam mest, celo pripadnikom iste farme strežnikov. SSP je samostojna komponenta, neodvisna od neke konkretne rešitve ali spletne strani, čeprav lahko funkcionalnost nastavimo različno za vsako spletno aplikacijo posebej.

SSP vsebuje:

- možnosti za iskanje, ki:
  - preiskujejo vsebine, vključno s stranmi SharePoint, zunanje vire podatkov preko povezave BDC (angl. Business Data Catalogue), datotečne sisteme in druga informacijska skladišča;
  - zgradijo in vzdržujejo indekse na te vsebine;
  - izvajajo poizvedbe na te indekse in vračajo rezultate iskanja;
- storitve Microsoft Office Excel, ki:
  - omogočajo nastavljanje zaupnih lokacij za Excel-ove delovne zvezke;
  - omogočajo programski dostop do računskih storitev Excel-a (angl. Microsoft Excel Calculation Services);
- možnosti shranjevanja in upravljanja lastnosti, ki se nanašajo na končne uporabnike:
  - uporabniški profili,
  - uporabniške strani,
  - podatki o članstvu,
  - podatki o prijateljih.

### 2.1.1.2 Funkcije MOSS 2007

Za gradnjo specifičnih rešitev MOSS 2007 organizacijam ponuja množico funkcij, ki se osredotočajo na praktična področja funkcionalnosti. Funkcijo lahko aktiviramo in s tem omogočimo funkcionalnost, ki jo ponuja, ali pa jo deaktiviramo in s tem onemogočimo, da bi bila funkcionalnost uporabljena. Funkcije imajo lahko različne obsege delovanja, odvisno od tega kako so bile nastavljene. Njihov obseg lahko obsega celotno farmo strežnikov, lahko pa tudi samo eno spletno aplikacijo, samo eno zbirko strani, ali pa samo en nivo strani.

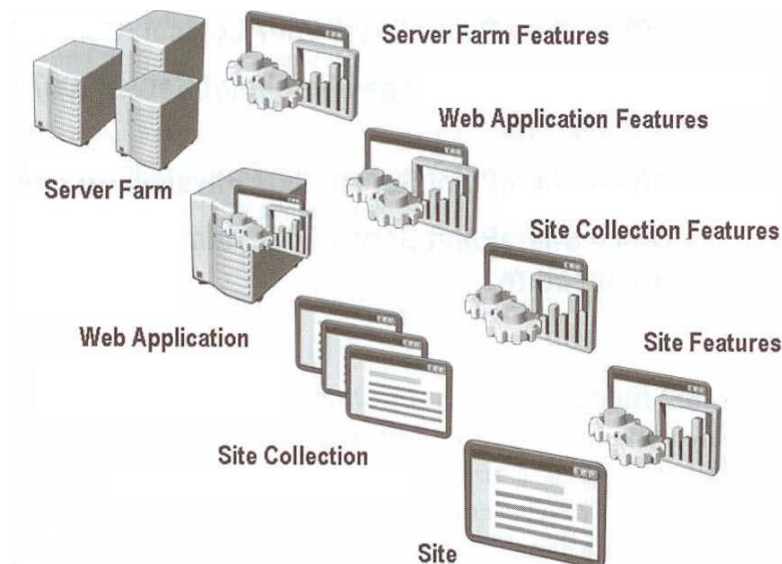
MOSS 2007 vsebuje naslednje funkcije, zgrajene nad platformo WSS, da zagotavlja funkcionalnost različnim rešitvam:

- funkcionalnost BDC in spletni gradniki, ki:
  - uporabljajo pooblastila za varno dostopanje do virov podatkov;
  - prejemajo podatke iz teh virov podatkov;
  - prikažejo prejete podatke uporabnikom.
- funkcionalnost objavljanja vsebine in poteki dela (angl. Workflow), ki:
  - omogočajo spremljanje vsebin in vodijo nadzor nad procesi lastništva, urejanja in objavljanja;
  - zagotavljajo, da je upravljanje z informacijami v skladu s politiko in pravili;
  - omogočajo scenarije za objavljanje vsebine v več jezikih.
- funkcionalnost poslovne inteligence in spletni gradniki, ki:
  - prikazujejo grafe in tabelarične podatke;
  - prikazujejo trende med podatki (KPI – angl. Key Performance Indicator).

## 2.1.2 Arhitektura MOSS 2007

Za implementacijo rešitve MOSS 2007 je pomembno razumevanje logične in fizične arhitekture sistema in kako se logične komponente preslikajo na fizične.

### 2.1.2.1 Logična arhitektura



Slika 3: Logična arhitektura MOSS 2007

Rešitve MOSS 2007 so zgrajene na hierarhiji logičnih komponent, od katerih vsaka zagotavlja specifično funkcionalnost.

Logične enote MOSS 2007 so:

- Farma strežnikov (angl. Server farm)

Ta najvišje ležeča enota je predstavljena z večjim številom spletnih in aplikacijskih strežnikov, ki so med seboj logično povezani in si delijo isto konfiguracijsko bazo.

- Spletna aplikacija (angl. Web application)

Ta enota zagotavlja funkcionalnosti spletnega strežnika. Spletna aplikacija MOSS 2007 se ujema s spletno stranjo Internet Information Services (IIS).

- Zbirka mest (angl. Site collection)

Ta enota definira nastavitve in zvezo za grupiranje več mest in podmest skupaj. Zbirko mest lahko razumemo podobno kot navidezni direktorij na najvišjem nivoju v IIS, čeprav v IIS ne bo preslikave na navidezni direktorij.

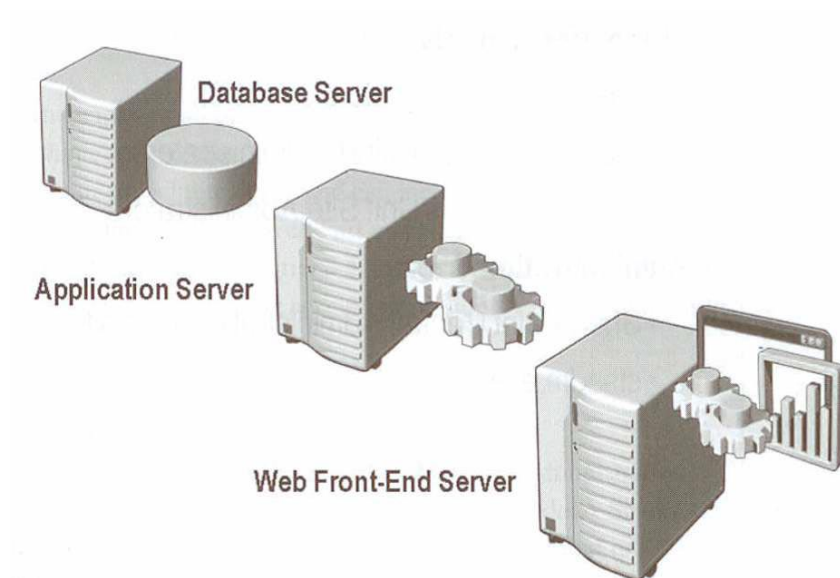
- Mesto (angl. Site)

Ta enota zagotavlja vhodno točko za specifično rešitev ali za množico funkcionalnosti. Mesto lahko primerjamo s podmapo navideznega direktorija na najvišjem nivoju v IIS, čeprav povezava v IIS ne bo obstajala.

- Funkcija (angl. Feature)

Ta enota zagotavlja specifične funkcionalnosti in podatke. Funkcija lahko vsebuje podatke, metapodatke in funkcionalnosti.

### 2.1.2.2 Fizična arhitektura



Slika 4: Fizična arhitektura MOSS 2007

Fizična arhitektura MOSS 2007 je sestavljena iz naslednjih enot:

- strežnik baz podatkov,
- aplikacijski strežnik,
- spletni strežnik.

### *Strežnik baz podatkov*

Ta komponenta shranjuje in upravlja s konfiguracijskimi podatki, podatki mest, metapodatki ter indeksira bazo podatkov. Vsi strežniki, ki so v isti farmi strežnikov, morajo uporabljati isti strežnik baz podatkov, ker le ta shranjuje in upravlja s konfiguracijsko bazo podatkov, ki hrani nastavitve za celotno farmo strežnikov.

### *Aplikacijski strežnik*

Ta komponenta izvaja aplikacijske storitve. Lahko imamo na primer, en aplikacijski strežnik v farmi strežnikov, ki bo upravljal iskanje in indeksiranje, in drugega, ki bo upravljal uvažanje uporabniških profilov in njihovo sinhronizacijo.

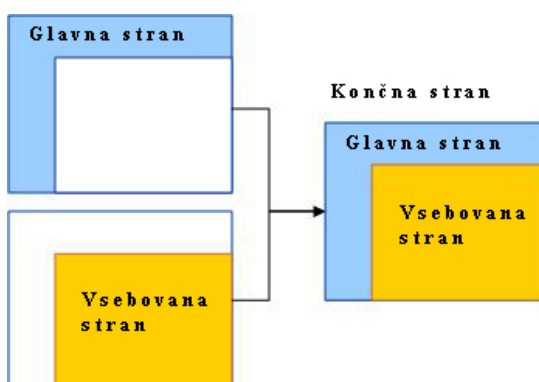
### *Spletni strežnik (angl. Web front-end server)*

Ta komponenta upravlja z zahtevami za MOSS 2007 rešitve. Sestavljena je iz večjega števila navideznih direktorijev, ki nudijo aplikacijske funkcije, kot so aplikacijsko upravljanje s stranmi, predlogami, temami in registriranimi komponentami, kot so spletni gradniki.

## **2.1.3 ASP .NET stran v MOSS 2007**

Spletna stran, ki je prikazana v brskalniku končnega uporabnika je kombinacija dveh strani: glavne strani (angl. Master Page) in vsebovane strani (angl. Content Page). Glavna in vsebovana stran skupaj tvorita instanco strani, to je spletno stran, ki jo končni uporabnik vidi v brskalniku.

Ko se generira HTML stran za prikaz v brskalniku, se glavna stran avtomatsko združi z vsebovano stranjo, tako da je izhod ena sama HTML stran.



Slika 5: Povezava glavne strani z vsebovano stranjo

### 2.1.3.1 Glavna stran

Glavna stran (angl. Master Page) [3, 5] je posebna stran, ki omogoča, da na enem mestu določimo skupne dele spletnih strani. Z uporabo glavne strani je neko spletno mesto (mesto tvori več spletnih strani) prikazano skladnejše.

Glavna stran vsebuje območja, ki jih imenujemo vsebovalniki strani (angl. ContentPlaceholder). Predstavljajo območja, katerim se bo vsebina spremenila za vsako stran posebej (npr. vsebovalnik strani `PlaceholderSiteName` bo na vsaki strani vseboval drugačen naslov strani). Območje zunaj vsebovalnikov strani pa je enako za vse strani, ki uporabljajo isto glavno stran, in se ne spreminja.

Na glavni strani se načrtuje pozicija elementov, kot so na primer kontrolniki za krmarjenje, logotipi podjetij, objave o avtorskih pravicah, vrstica za odjavo uporabnika iz sistema, vnosno polje za iskanje, vrstico, ki prikazuje, kje se trenutno nahajamo v drevesni strukturi strani (angl. breadcrumb), itd.

Vsako mesto ima privzeto eno glavno stran in je shranjeno v galeriji glavne strani in postavitve strani. V galeriji glavne strani je mogoče shraniti poljubno število glavnih strani, kot privzeto glavno stran za mesto pa je mogoče izbrati samo eno glavno stran. Glavne strani je mogoče tudi gnezditi.

Glavne strani so navadno bolj kompleksne in zato razvite s strani razvijalcev, medtem ko so vsebovane strani ustvarjene s strani končnih uporabnikov. Glavna stran je ASP .NET stran shranjena v datoteki s končnico `.master`. Glavno stran prepoznamo po značilnem napotku `Master`.

```
<%@ Master Language="C#" %>
```

*Slika 6: Primer napotka za glavno stran*

```
<%@ Master Language="C#" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
  1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html>
<head id="Head1" runat="server" >
  <title>Master page title</title>
</head>
<body>
  <form id="form1" runat="server">
    <table>
      <tr>
        <td>
          <asp:contentplaceholder id="Main" runat="server" />
        </td>
        <td>
          <asp:contentplaceholder id="Footer" runat="server" />
        </td>
      </tr>
    </table>
  </form>
</body>
</html>
```

Slika 7: Primer enostavne glavne strani

### 2.1.3.2 Vsebovana stran

Vsebovana stran (angl. Content Page) je vsaka spletna stran, ki uporablja oz. je povezana z glavno stranjo. Vsebovane strani so ASP .NET strani, ki so shranjene v datotekah z običajno .aspx končnico. Vsebovani spletni strani določimo glavno stran s pomočjo napotkov. V napotku Page lastnosti MasterPageFile dodelimo pot do želene predloge strani.

```
<%@ Page Language="C#" MasterPageFile="~/MasterPages/Master1.master"
  Title="Content Page" %>
```

Slika 8: Primer napotka vsebovane strani, ki vsebuje povezavo na glavno stran

Vsebovana stran ni veljavna samostoječa spletna stran. Povezana mora biti z glavno stranjo, da tako pridobi podatke o izgledu in postavitvi strani.

Vsebovana stran vsebuje definicije vsebovalnikov strani, ki se pojavijo v glavni strani. Vse, česar v vsebovani strani ni v enem od vsebovalnikov strani se odraža kot napaka (razen blokov, napisanih v skriptem jeziku). Sicer pa v vsebovalniku strani lahko realiziramo vse, kar lahko realiziramo v običajni ASP .NET strani.

```

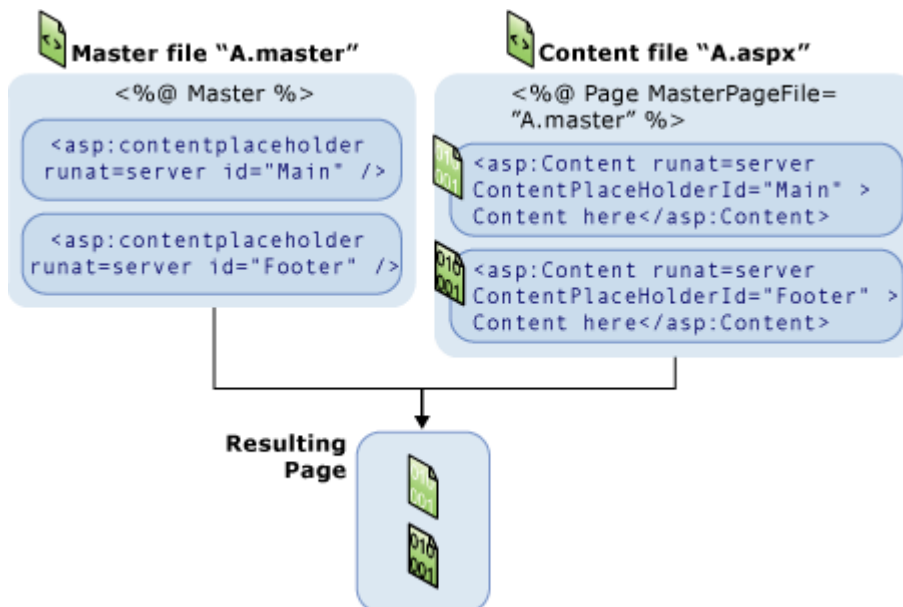
<% @ Page Language="C#" MasterPageFile="~/Master.master" Title="Content
Page 1" %>
<asp:Content ID="Content1" ContentPlaceHolderID="Main" Runat="Server">
    Glavna vsebina strani
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="Footer" Runat="Server" >
    Vsebina v glavi strani.
</asp:content>

```

Slika 9: Primer enostavne vsebovane strani

V zgornjem primeru lahko vidimo, da vsebovana stran ne vsebuje elementov, ki jih mora vsebovati vsak HTML dokument (kar v končni fazi vsaka .aspx stran tudi je). Elemente, kot so npr. <html>, <head runat="server">, <body>, itd. mora vsebovati glavna stran.



Slika 10: Glavna stran in vsebovana stran med izvajanjem

Končni uporabnik vidi združeni glavno stran in vsebovano stran kot eno samo stran. URL naslov te končne strani pa je tak, kot ga ima vsebovana stran.

### 2.1.3.3 Obravnava zahteve za .aspx stran v MOSS 2007

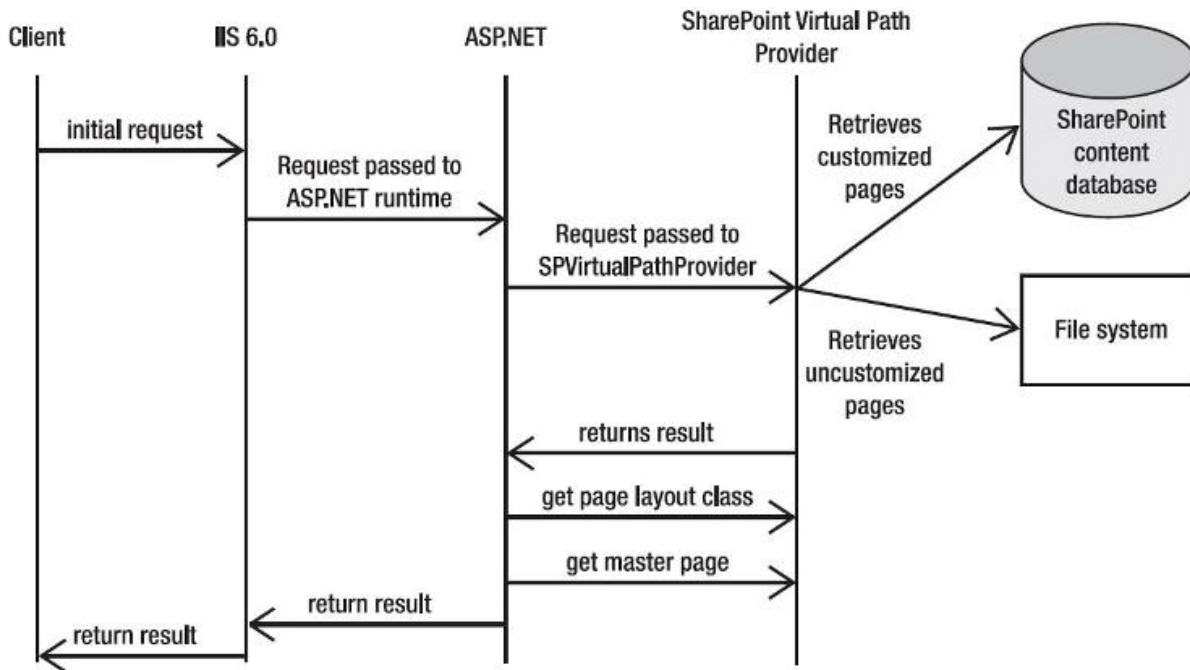
V MOSS 2007 so vhodne zahteve [4] vedno najprej obravnavane s strani izvajalnega okolja ASP .NET (angl. ASP .NET Runtime). Infrastruktura MOSS 2007 se zanaša na ASP .NET 2.0 tako, da definira HTTP module in HTTP upravljalce (angl. handlers), ki so zadolženi za obravnavanje zahtev, povezanih s SharePoint stranmi.

V datoteki web.config, ki pripada SharePoint spletni aplikaciji, lahko opazimo, da oznaka `<httpHandlers>` vsebuje upravljalca `http`, ki se imenuje `Microsoft.SharePoint.ApplicationRuntime.SPHttpHandler`. Ta upravljalca zagotavlja, da bo upravljalca `SPHttpHandler` navsezadnje obravnaval zahteve za vse tipe datotek (npr. `.aspx`, `.txt`, `.doc`, `.docx`).

Ko ASP .NET konča s procesiranjem zahteve, pokliče SharePoint ponudnika navidezne poti (angl. `SharePoint Virtual Path Provider`, poznan tudi kot `WSS File Provider` ali `SPVirtualPathProvider`). SharePoint ponudnik navidezne poti se uporablja za razčlenjevanje `.aspx` strani. SharePoint ponudnik navidezne poti lahko pridobi `.aspx` strani iz podatkovne baze SQL strežnika ali pa iz datotečnega sistema.

MOSS 2007 podpira dva tipa strani: neprirejene (angl. `uncustomized`) in prirejene (angl. `customized`). Neprirejene strani so predloge strani, ki so shranjene v datotečnem sistemu spletnega strežnika. Prirejene strani pa so tiste predloge strani, ki so bile kakorkoli spremenjene in so shranjene v SharePoint podatkovno bazo vsebine. Ena izmed nalog, za katero je zadolžen SharePoint ponudnik navidezne poti je, da se, ko obravnava zahtevo po strani, odloči, ali se stran nahaja v datotečnem sistemu ali v SharePoint podatkovni bazi vsebine.

Ko je SharePoint ponudnik navidezne poti enkrat obravnaval (ampak ne prevedel) stran, le to vrne nazaj izvajalnemu okolju ASP .NET, ki stran prevede. Nato ponovno vpraša ponudnika virtualne poti po razredu postavitve strani, ki je tudi že preveden. Izvajalno okolje ASP .NET pridobi še glavno stran, ki je povezana s stranjo, ki jo trenutno obravnava. Glavno stran prevede, nato pa pošlje odgovor na poizvedbo.

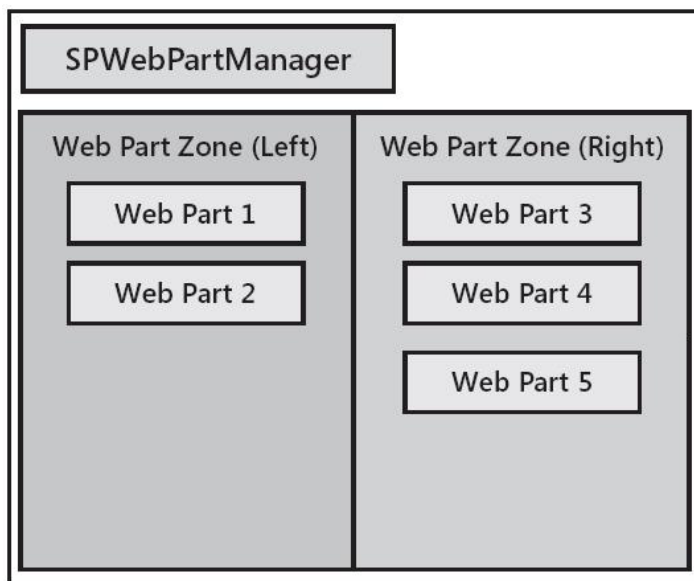


Slika 11: Obravnavanje poizvedbe strani v MOSS 2007

#### 2.1.3.4 Stran s spletnimi gradniki

Stran s spletnimi gradniki (angl. Web Part page) [3] je običajna ASP .NET stran (datoteka ima končnico `.aspx`), ki pa mora vsebovati natanko eno instanco kontrolnika `WebPartManager` in pa eno ali več območij spletnih gradnikov (kontrolnikov `WebPartZone`). Območja spletnih gradnikov imenujemo tudi vsebniki spletnih gradnikov. V večini primerov nam ni potrebno skrbeti za kontrolnik `WebPartManager`, ker je edina potrebna instanca tega kontrolnika že definirana v standardni glavni strani (npr. `default.master`). Kadar torej ustvarjamo novo vsebovano stran in je ta povezana z glavno stranjo, ki že vključuje kontrolnik `WebPartManager`, nam le tega ni potrebno še enkrat ustvariti.

Za območje spletnih gradnikov velja, da ga lahko ustvarimo samo na strani s spletnimi gradniki. Vsako območje spletnih gradnikov pa lahko vsebuje nič, enega ali več spletnih gradnikov.



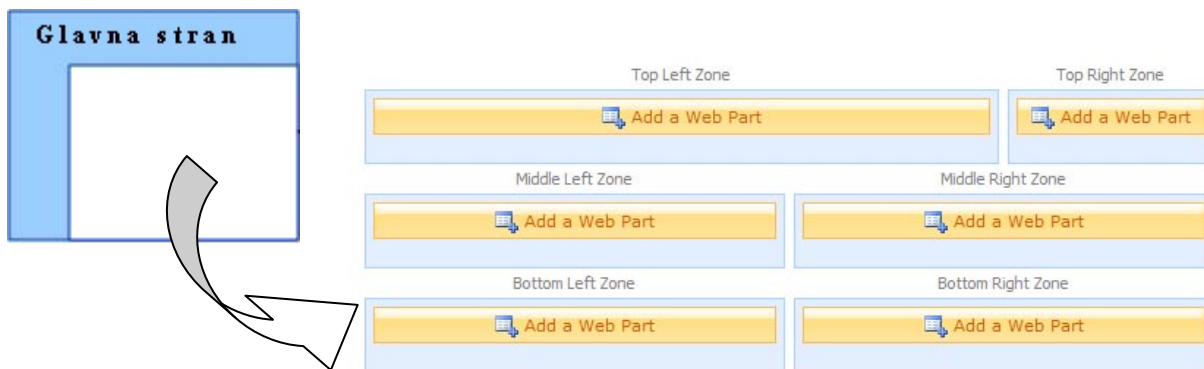
Slika 12: Stran s spletnimi gradniki

Spodnji primer predstavlja situacijo, ki je narisana na zgornji sliki. Koda opisuje vsebovano stran, ki vsebuje en vsebovalnik strani, v katerem sta definirani dve območji spletnih gradnikov. Prvo območje zaseda levi del strani, drugo pa desni del. Tako razporeditev najenostavneje dosežemo z uporabo gradnika tabela. Opazimo lahko, da nikjer ne ustvarimo gradnika WebPartManager. V primeru je predpostavljeno, da je instanca tega kontrolnika kreirana že v glavni strani default.master, s katero je vsebovana stran povezana.

```
<%@ Assembly Name="[Fully-qualified name for Microsoft.SharePoint.dll]" %>
<%@ Page language="C#" MasterPageFile="~masterurl/default.master"
    Inherits="Microsoft.SharePoint.WebPartPages.WebPartPage" %>
<%@ Register Tagprefix="WebPartPages"
    Namespace="Microsoft.SharePoint.WebPartPages"
    Assembly="[Fully-qualified name for Microsoft.SharePoint.dll]" %>

<asp:Content ID="Content1" ContentPlaceHolderId="PlaceholderMain"
runat="server">
    <h3>My Custom Web Part Page</h3>
    <table border="5" cellpadding="5" cellspacing="0">
        <tr>
            <td valign="top">
                <WebPartPages:WebPartZone runat="server" ID="Left"
                    Title="Left Zone" />
            </td>
            <td valign="top">
                <WebPartPages:WebPartZone runat="server" ID="Right"
                    Title="Right Zone" />
            </td>
        </tr>
    </table>
</asp:Content>
```

Slika 13: Vsebovana stran z enim vsebovalnikom in dvema območjema za spletne gradnike



Slika 14: Razporeditev območij spletnih gradnikov v MOSS 2007

Če kreiramo novo stran s spletnimi gradniki iz predloge v MOSS 2007, se nam območja spletnih gradnikov kreirajo že sama, razporejena pa so, kot prikazuje zgornja slika.

### 2.1.3.5 Spletni gradniki

Spletni gradnik (angl. WebPart) je modularna enota informacij, ki oblikuje temeljne gradnike strani s spletnimi gradniki. Spletne gradnike lahko končni uporabniki dodajajo v območja spletnih gradnikov na strani s spletnimi gradniki med samim delovanjem. Neposredno v brskalniku lahko posamezen spletni gradnik tudi prilagodijo ter tako zase ustvarijo edinstveno stran.

Funkcionalnosti v povezavi s spletnimi gradniki so dodajanje, brisanje, premikanje iz enega območja spletnih gradnikov v drugega, skrivanje, minimiziranje, spreminjanje vsebine prikaza, spreminjanje izgleda in spreminjanje obnašanja. Razvijalec se odloči, katere funkcionalnosti bodo končnim uporabnikom na voljo in katere ne.

### 2.1.4 Potek dela

Potek dela (angl. Workflow) [2] je naravna pot za organiziranje in spremljanje poteka opravil ali korakov napredovanja neke aktivnosti. Lahko vključuje ljudi in njihovo interakcijo s sistemom. Potek dela si lahko predstavljamo kot izvajalno predstavitev poteka dela.

Poznamo dve vrsti poteka dela, ki jih ponuja storitev Windows SharePoint Services.

- Zaporeden potek dela (angl. Sequential Workflow) je modeliran kot diagram poteka in je primeren za sistemsko orientirane poteke dela. Zaporeden potek dela predstavlja zaporedje korakov, ki se izvajajo v določenem vrstnem redu, dokler ni dokončan zadnji korak. Ko je dokončan zadnji korak, je zaporeden potek dela zaključen. Na zaporedne poteke dela lahko vplivajo tudi zunanji dogodki in lahko vključujejo

vzporedne logične poti, zato se lahko natančno zaporedje, v katerem so aktivnosti izvršene, spreminja.

- Potek dela s stanji (angl. State Machine Workflow) je modeliran kot diagram stanj in je najbolj primeren za poteke dela, ki vključujejo ljudi. Potek dela s stanji je predstavljen z množico stanj, dogodkov in prehodov med stanji. Eno od stanj je označeno kot začetno stanje. Prehod v drugo stanje se naredi glede na dogodek, ki se zgodi. Potek dela s stanji ima lahko tudi končno stanje. Če se zgodi, da potek dela zavzame to stanje, pravimo, da se je potek dela s stanji zaključil. Poteki dela s stanji so učinkoviti v zajemanju procesov, ki se lahko spremenijo medtem ko so v teku. Npr.: bančni uslužbenec lahko stranki odobri posojilo kljub slabim izkušnjam v preteklosti na podlagi posebnih okoliščin, ki prvotno niso bile modelirane v potek dela.

### 2.1.5 Časovno opravilo

Veliko različnih tipov aplikacij potrebuje usluge neke vrste procesov, ki se izvajajo po zanje določenem urniku. Imenujemo jih časovna opravila (angl. TimerJob) [6]. Ti procesi so uporabljeni za računsko zahtevne operacije, obvestila, preverjanje pravilnosti podatkov in še za mnogo drugih opravil.

Primer za tak proces v WSS je indeksiranje vsebine v določenih krajših intervalih, da iskalnik lahko najde iskano vsebino, če sploh obstaja. Drug primer bi bil npr. tedensko pošiljanje e-mail sporočil uporabnikom, ki so se prijavili na obveščanje o spremembah na nekem seznamu.

Časovna opravila v MOSS 2007 so vodena s strani storitve SharePoint Timer. To je storitev operacijskega sistema Windows, ki se nastavi že ob namestitvi MOSS 2007.

Šele storitev Windows SharePoint Services 3.0 je omogočila, da lahko na enostaven način ustvarimo lastno časovno opravilo in ga dodamo med ostala časovna opravila, ki so že del sistema.

Časovno opravilo definiramo z uporabo enega razreda, ki je podedovan iz razreda `Microsoft.SharePoint.Administration.SPJobDefinition`. Znotraj tega razreda ponovno definiramo (angl. override) metodo `Execute()`, ki vsebuje kodo opravila, ki se izvede ob izvajanju časovnega opravila.

V naslednjem koraku je potrebno opravilo registrirati. To storimo tako, da definiramo nov razred, ki je podedovan iz razreda `Microsoft.SharePoint.SPFeatureReceiver` in vsebuje dva upravljalca dogodkov (angl. event handler), in sicer: `FeatureActivated` in `FeatureDeactivating`.

## 2.2 XML

XML (angl. Extensible Markup Language) [7] je označevalni jezik. Jezik je podoben jeziku HTML. Obstaja pa velika razlika. XML ni namenjen prikazovanju podatkov, ampak njihovemu prenašanju in shranjevanju, s poudarkom na tem, kaj ti podatki sploh so in kaj predstavljajo. XML ne naredi ničesar. Samo strukturira, shranjuje in prenaša informacijo na čim bolj samoopisen način.

Dokumentu XML pravimo, da je dobro oblikovan (ustreza sintaksi), če zadostuje naslednjim pogojem:

- Dokument XML lahko vsebuje samo en korenski element. Vsa vsebina, ki je del dokumenta samega, je vsebovana v korenskem elementu.
- Elemente XML lahko gnezdimo.
- Vsak elementi XML vsebuje začetno in končno oznako.
- Imena začetnih in končnih oznak morajo biti enaka.
- Elementi XML se ne smejo prekrivati med seboj.
- Znotraj besedila v dokumentu XML se ne smejo pojavljati naslednji znaki: < > &. To so znaki, ki imajo določen pomen za razčlenjevalnik XML.

XML dokument je veljaven, če je dobro oblikovan in če se lahko prilagodi določenemu naboru omejitev, ki so definirane v shemi XML.

Shema XML je formalna specifikacija, ravno tako napisana v jeziku XML, ki določa strukturo dokumenta XML, imena elementov in obogatenih podatkovnih tipov, kateri elementi se lahko prikažejo v kombinaciji in kateri atributi so na voljo za posamezen element.

XML dokument je sestavljen iz dveh glavnih delov: uvoda in korenskega elementa. Dokument XML lahko vsebuje tudi komentarje. Uvod vsebuje deklaracijo XML. Ta navaja, da je to dokument XML, vsebuje navodila za obdelavo, pove kako obravnavati dokument in deklaracije shem XML za preverjanje veljavnosti dokumenta.

Korenski del elementa je glavni del dokumenta XML. Informacije v korenskem elementu so shranjene v dveh vrstah zgradb XML: elementih in atributih. Elementi so glavni gradniki dokumenta XML in se uporabljajo za predstavitev strukture dokumenta XML. Začetna oznaka elementa pa lahko vsebuje tudi attribute, ki se uporabljajo za opis določenih lastnosti tega elementa.

```
<začetna_oznaka atribut="vrednost">podatki o elementu<končna_oznaka>
```

Slika 15: Splošna oblika oznake elementa

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Ta dokument XML vsebuje podatke o zaposleni osebi. -->
<Employees>
  <Employee>
    <FirstName>
      Janez</ FirstName >
    <LastName>
      Novak</ LastName >
    </Employee>
  <Employee>
    <FirstName>
      Jure</ FirstName >
    <LastName>
      Gortnar</ LastName >
    </Employee>
  <Employee>
    <FirstName>
      Luka</ FirstName >
    <LastName>
      Dovžan</ LastName >
    </Employee>
</Employees>
```

Slika 16: Primer dokumenta XML

### 2.2.1 XSLT

XSLT (angl. Extensible Stylesheet Language Transformations) [8, 9] je podskupina še enega jezika, zasnovanega na jeziku XML: XSL (angl. Extensible Stylesheet Language). XSL se uporablja za oblikovanje dokumentov XML, XSLT pa vsebuje predloge in ukaze za izbiranje in upravljanje strukture podatkov.

Za razumevanje delovanja jezika XSLT si moramo ogledati, kako brskalniki, npr. Microsoft Internet Explorer, obravnavajo XML podatke. Ko se v brskalniku odpre podatkovna datoteka XML, brskalnik uporabi vgrajeni razčlenjevalnik, da XML dokument razčleni v drevesno strukturo. Če v podatkovni datoteki XML najde sklic na dokument XSLT, odpre slogovno datoteko in jo uporabi na vnosnem drevesu. Brskalnik preleti izvorno datoteko in v datoteki XSLT poskuša najti vzorce, ki ustrezajo določenim elementom v XML podatkovni datoteki. Ko brskalnik najde par, na elementih in izhodnih podatkih uporabi ustrezne predloge po pravilih v predlogi. Predloga lahko vsebuje ukaze za upravljanje s podatki (izbiranje, filtriranje, sortiranje), HTML oznake za oblikovanje ali oboje. Rezultat je izhodno drevo z vsemi spremembami, ki jih naredijo predloge in HTML oznake, dodane zaradi oblikovanja

datoteke. Nato brskalnik razčleni izhodno drevo in ga prikaže, kot določajo HTML navodila. Na najnižji ravni se XSLT torej uporablja za preoblikovanje ene drevesne strukture v drugo.

```
1. <xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl" version="1.0">
2.   <xsl:template match="/">
3.     <HTML>
4.       <HEAD>
5.         <TITLE>Zaposleni</TITLE>
6.       </HEAD>
7.       <BODY>
8.         <h4>Razvrščeni seznam imen zaposlenih</h4>
9.         <xsl:for-each select="//Employees">
10.          <xsl:sort select="LastName" order="ascending"/>
11.          <xsl:value-of select="LastName"/>,
12.          <xsl:value-of select="FirstName"/>
13.          <br/>
14.        </xsl:for-each>
15.      </BODY>
16.    </HTML>
17.  </xsl:template>
18.</xsl:stylesheet>
```

*Slika 17: Primer dokumenta XSLT*

Vrstica 1. V primeru s slike 17 je standardna glava XSLT datoteke in je sestavljena iz začetne oznake elementa, ki označuje slogovno datoteko. Odsek `xmlns:xsl` določa predpono za imenski prostor.

2. vrstica predstavlja pravilo XSL predloge, ki se uporabi, ko naletimo na določen del izvirnega dokumenta. V tem primeru `"/` ustreza korenskemu elementu.

XSLT navodilo v vrstici 9 pomeni, da naj se razčlenjevalnik premakne (označeno z `»/«`) relativno glede na trenutni položaj (v tem primeru s korenskega elementa), na podrejene elemente oznake elementa `»Employees«` in začne ponavljati skozi podrejene elemente te oznake. Navodilo v vrstici 10 določa sortiranje, v vrsticah 11 in 12 pa izbiranje.

Ostale oznake podajajo dodatne informacije o oblikovanju (HTML oznake) ali pa so zaključne oznake za vse odprte oznake.

Če dokumentu XML s slike 16 dodamo povezavo na dokument XSLT s slike 17, bi brskalnik osnovni dokument XML pretvoril v naslednjo obliko:

**Razvrščeni seznam imen zaposlenih**

Dovžan,Luka

Gortnar,Jure

Novak,Janez

*Slika 18: Formatiran izhod dokumenta XML povezanega z dokumentom XSLT*



### 3 Implementacija telefonskega imenika v MOSS 2007

Za osnovo pri implementaciji telefonskega imenika v portal »Središče EG« je bil uporabljen gradnik seznam, ki ga ponuja MOSS 2007. Vsi podatki telefonskega imenika so shranjeni v seznamu Stiki.

#### Seznam »Stiki«

Seznam v MOSS 2007 je struktura, ki je sestavljena iz stolpcev in vrstic. Vsak stolpec ima določen tip podatkov, ki jih lahko vsebuje. Tipi so ena vrstica teksta, več vrstic teksta, število, valuta, datum in čas, izbira med določenimi vrednostmi (npr. za izbiro sta na voljo samo vrednosti veljaven oz. neveljaven zapis), ja/ne, oseba ali skupina, hiperpovezava, vrednost izračunana iz ostalih stolpcev, HTML koda, slika in še nekateri drugi. Vsakemu stolpcu posebej se lahko nastavi, ali je vnos vrednosti obvezen ali ne. En zapis v seznamu predstavlja ena vrstica.

Seznam Stiki je zgrajen iz stolpcev z imeni: Ime, Priimek, Polno ime, Podjetje, ORG (organizacijska enota), MPO, GSM, Faks, EVI (evidenčna številka), E-poštni naslov in še nekateri drugi. Poleg naštetih stolpcev pa seznam Stiki vsebuje še nekaj stolpcev, ki vsebujejo metapodatke za zapise. Taki stolpci so: last\_synchronization, Date\_created, Date\_modified, Created\_by, Modified\_By. Vrednosti za te stolpce se vnašajo oz. spreminjajo ob dogodkih, kot so ustvarjanje novega stika, spreminjanje stika, posodabljanje stikov.

Priimek	Ime	Polno ime	Podjetje	EVI	Interneta	MPO	GSM	ORG	Faks	Telefon
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...

Slika 19: Pogled na seznam Stiki

## Ustvarjanje novega stika v seznamu Stiki

Ko urednik telefonskega imenika želi dodati nov stik, se mu ponudita dve možnosti.

- Iz spustnega seznama lahko izbere osebo, za katero želi ustvariti stik. V spustnem seznamu so našteje osebe, ki so trenutno zaposlene v podjetju Elektro Gorenjska in še nimajo zapisa v seznamu Stiki (poizvedba na zunanje vire podatkov, ki je v tem primeru podatkovna baza DB2). Če urednik telefonskega imenika izbere to možnost, se mu odpre vnosna maska za vnos podatkov o osebi, katero je izbral, pri čemer so nekatera polja že izpolnjena (Ime, Priimek, Polno ime, Podjetje, EVI, E-poštni naslov, ORG).
- Lahko pa izbere možnost, da ustvari stik, pri čemer izhaja iz enake maske kot v prvem primeru, le da so vsa vnosna polja prazna.

Središče EG > Imenik > Stiki > Nov element  
Stiki : Nov element

Ustvari stik za izbrano osebo

Ustvari stik za izbrano osebo

Ustvari stik (prazna maska)

Ustvari stik (prazna maska)

Slika 20: Ustvarjanje novega elementa (stika) v seznamu Stiki

Prva možnost se uporabi, ko urednik telefonskega imenika želi v seznam Stiki dodati stik osebe, ki se je na novo zaposlila v podjetju. Podatki (Ime, Priimek, Polno ime, Podjetje, EVI, E-poštni naslov, ORG) teh oseb so shranjeni v podatkovni bazi DB2, kamor jih vnaša kadrovska služba.

Druga možnost pa se uporabi, če je oseba npr. študent ali pa se stik navezuje na osebo, ki je zunanji sodelavec. Za osebe pod to skupino kadrovska služba ne vnaša podatkov v tabele podatkovne baze DB2.

Stike oseb, opisane v prvi možnosti bi lahko dodali tudi na način, opisan pod drugo možnostjo, a bi bila s tem možnost napake pri vnosu večja, saj ne bi izkoristili že enkrat vnesenih in pravih podatkov.

Središče EG > Imenik > Stiki > Nov element  
**Stiki : Nov element**

V redu    Prekliči

Priloži datoteko | Črkovanje... \* označuje zahtevano polje.

Polno ime *	<input type="text"/>
Ime	<input type="text"/>
Priimek	<input type="text"/>
E-poštni naslov	<input type="text"/>
Podjetje	<input type="text"/>
ORG	<input type="text"/>
Interna	<input type="text"/>
MPO	<input type="text"/>
GSM	<input type="text"/>
Faks	<input type="text"/>
EVI	<input type="text"/>
GSM tip	<input type="text"/>
GSM1	<input type="text"/>
MPO1	<input type="text"/>
GSM tip1	<input type="text"/>
Spletna stran	Vnesite spletni naslov: (Če želite izvesti preskus, kliknite sem) <input type="text" value="http://"/> Vnesite opis: <input type="text"/>
Opombe	<div style="border: 1px solid gray; padding: 5px;"> <div style="font-size: small; margin-bottom: 5px;"> <span>A</span> <span>A1</span> <span>K</span> <span>L</span> <span>P</span>   <span>☰</span> <span>☷</span> <span>☶</span> <span>☵</span> <span>☴</span> <span>☳</span> <span>☲</span> <span>☱</span> <span>☰</span> <span>☷</span> <span>☶</span> <span>☵</span> <span>☴</span> <span>☳</span> <span>☲</span> <span>☱</span> </div> <div style="height: 100px;"></div> </div>
GSM2	<input type="text"/>
MPO2	<input type="text"/>
GSM tip2	<input type="text"/>
LOK	<input type="text"/>
Soba	<input type="text"/>
OBJ	<input type="text"/>
TIP	<input type="text"/>
SIS	<input type="text"/>
VRS	<input type="text"/>
OZN CR	<input type="text"/>
OZN ST	<input type="text"/>
NAD	<input type="text"/>
Uporabnik	<input type="text"/>
Telefon	<input type="text"/>

V redu    Prekliči

Slika 21: Prazna vnosna maska za ustvarjanje novega elementa (stika) v seznamu Stiki

Po končanem vnosu podatkov se s pritiskom na gumb »V redu« stik vnese v seznam Stiki.

## Urejanje, brisanje in pregled podrobnosti stika

Urednik telefonskega imenika lahko obstoječ stik ureja, izbriše ali pregleduje vizitko z vsemi podrobnostmi stika.

Obrazec za urejanje je enak obrazcu za ustvarjanje stika, z razliko, da so vnosna polja izpolnjena z vrednostmi, ki jih ima trenutno nek stik. Po končanem urejanju se podatke potrdi s pritiskom na gumb »V redu«.

Tudi vizitka z vsemi podrobnostmi stika je podobna obrazcu za ustvarjanje stika, s to razliko, da urejanje vnosnih polj ni mogoče.

## Osnovna stran telefonskega imenika - Default.aspx

Funkcionalnosti telefonskega imenika, opisane do tega mesta, so na voljo samo uredniku telefonskega imenika in administratorjem Središča EG. To je zagotovljeno s sistemom pravic, ki je tudi del MOSS 2007.

Navadni uporabniki si od vseh strani, razvitih znotraj telefonskega imenika, lahko ogledajo samo prvo stran, default.aspx.

The screenshot shows a web browser window displaying the 'Središče EG' portal. The main content area is titled 'Imenik' and contains a search bar and a grid of contact entries. The grid has columns for 'Podjetje', 'EVI', 'NAZIV', 'Interna', 'MPO', 'GSM', 'Faks', and 'OE'. The 'Zadnje spremembe' sidebar on the right lists recent changes with columns for 'EVI', 'Polno ime', and 'Interna'. The page header includes the 'elektro gorenjska' logo and navigation links like 'Središče EG', 'Elektro Gorenjska', 'Za zaposlene', 'Na trgu', 'Standardi kakovosti', and 'Dejanja mesta'.

Slika 22: Osnovna stran telefonskega imenika na portalu »Središče EG«

Izhodiščna stran telefonskega imenika je zgrajena kot spletna stran s spletnimi gradniki (angl. WebPart) in vsebuje dva spletna gradnika, kot je razvidno tudi iz zgornje slike.

Prvi je poimenovan Stiki in služi kot iskalnik po seznamu Stiki.

Drugi spletni gradnik pa je poimenovan Zadnje spremembe in prikazuje stike, ki so bili na novo ustvarjeni ali pa samo spremenjeni preko funkcije urejanje. To so tisti stiki, ki so glede na metapodatka date\_created in date\_modified mlajši od 10 dni glede na trenutni datum.

### 3.1 Spletni gradnik Stiki

```

<WebPart xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.microsoft.com/WebPart/v2">
  <Title>Stiki</Title>
  <FrameType>Default</FrameType>
  <Description>Uporabi za enostavne obrazce</Description>
  <IsIncluded>true</IsIncluded>
  <PartOrder>1</PartOrder>
  <FrameState>Normal</FrameState>
  <Height>600px</Height>
  <Width>785px</Width>
  <AllowRemove>true</AllowRemove>
  <AllowZoneChange>true</AllowZoneChange>
  <AllowMinimize>true</AllowMinimize>
  <AllowConnect>true</AllowConnect>
  <AllowEdit>true</AllowEdit>
  <AllowHide>true</AllowHide>
  <IsVisible>true</IsVisible>
  <DetailLink>Lists/Stiki</DetailLink>
  <HelpLink />
  <HelpMode>Modeless</HelpMode>
  <Dir>Default</Dir>
  <PartImageSmall />
  <MissingAssembly>Spletnega gradnika ni mogoče uvoziti.</MissingAssembly>
  <PartImageLarge>/_layouts/images/mssfw.gif</PartImageLarge>
  <IsIncludedFilter />
  <ExportControlledProperties>true</ExportControlledProperties>
  <ConnectionID>00000000-0000-0000-0000-000000000000</ConnectionID>
  <ID>g_c813f5d0_3313_43cd_855f_8dbadc7547c0</ID>
  <Content xmlns="http://schemas.microsoft.com/WebPart/v2/SimpleForm">
    <![CDATA[
      <div style="height:400px;">
        <script type="text/javascript">
          var timeout = -1;

          function searchChange(textbox) { //funkcija, ki se kliče ob spremembi vrednosti v vnosnem polju
            var text = textbox.value;
            if (timeout != -1)
              clearTimeout(timeout);
            timeout = setTimeout("commitSearch(' + text + '", 400);
          }

          function commitSearch(text) { //funkcija, ki sproži ISKANJE
            var frame = document.getElementById("contactFrame");
            if (text == "")
              frame.src = "Results.aspx?ReturnAll=Yes";
            else
              frame.src = "Results.aspx?Search=" + text;
          }

          function setFocus(){ //funkcija nastavi kurzor v vnosno polje
            var param = queryString("src");
            if(param !=false){
              document.getElementById("contactSearch").focus();
              document.getElementById("contactSearch").value = param;
              commitSearch(param);
            }
            else{
              document.getElementById("contactSearch").focus();
            }
          }

          function queryString(parameter) { //vrne vrednost parametra src iz url naslova ...?src=vrednost

```

```

var loc = location.search.substring(1, location.search.length);
var param_value = false;

var params = loc.split("&");
for (i=0; i<params.length;i++) {
    param_name = params[i].substring(0,params[i].indexOf('='));
    if (param_name == parameter) {
        param_value = params[i].substring(params[i].indexOf('=')+1)
    }
}
if (param_value) {
    return param_value;
}
else {
    return false; // prameter ni bil poslan
}
}

function filter(text){ //funkcija, ki se kliče ob pritisku na gumbe od A do Ž, sproži se ISKANJE
var frame = document.getElementById("contactFrame");
frame.src = "ResultsFiltered.aspx?Search=" + text;
}
}
</script>

<div class="ms-listdescription">
    Uporabi spodnje tekstovno polje za iskanje stikov po imenu, priimku, EVI,
    interni, GSM, MPO ali faks številki.
</div>
<br/>Išči:
<input type="text" style="border:1px gray solid; width: 450px;" id="contactSearch" name="contactSearch"
    onkeyup="searchChange(this);"/>

<br/><br/>
<input name="a" type="button" value="A" onclick="filter('a')" style="width: 20px; text-align: center;"/>
<input name="b" type="button" value="B" onclick="filter('b')" style="width: 20px; text-align: center;"/>
<input name="c" type="button" value="C" onclick="filter('c')" style="width: 20px; text-align: center;"/>
<input name="č" type="button" value="Č" onclick="filter('č')" style="width: 20px; text-align: center;"/>
<input name="d" type="button" value="D" onclick="filter('d')" style="width: 20px; text-align: center;"/>
<input name="e" type="button" value="E" onclick="filter('e')" style="width: 20px; text-align: center;"/>
<input name="f" type="button" value="F" onclick="filter('f')" style="width: 20px; text-align: center;"/>
<input name="g" type="button" value="G" onclick="filter('g')" style="width: 20px; text-align: center;"/>
<input name="h" type="button" value="H" onclick="filter('h')" style="width: 20px; text-align: center;"/>
<input name="i" type="button" value="I" onclick="filter('i')" style="width: 20px; text-align: center;"/>
<input name="j" type="button" value="J" onclick="filter('j')" style="width: 20px; text-align: center;"/>
<input name="k" type="button" value="K" onclick="filter('k')" style="width: 20px; text-align: center;"/>
<input name="l" type="button" value="L" onclick="filter('l')" style="width: 20px; text-align: center;"/>
<input name="m" type="button" value="M" onclick="filter('m')" style="width: 20px; text-align: center;"/>
<input name="n" type="button" value="N" onclick="filter('n')" style="width: 20px; text-align: center;"/>
<input name="o" type="button" value="O" onclick="filter('o')" style="width: 20px; text-align: center;"/>
<input name="p" type="button" value="P" onclick="filter('p')" style="width: 20px; text-align: center;"/>
<input name="r" type="button" value="R" onclick="filter('r')" style="width: 20px; text-align: center;"/>
<input name="s" type="button" value="S" onclick="filter('s')" style="width: 20px; text-align: center;"/>
<input name="š" type="button" value="Š" onclick="filter('š')" style="width: 20px; text-align: center;"/>
<input name="t" type="button" value="T" onclick="filter('t')" style="width: 20px; text-align: center;"/>
<input name="u" type="button" value="U" onclick="filter('u')" style="width: 20px; text-align: center;"/>
<input name="v" type="button" value="V" onclick="filter('v')" style="width: 20px; text-align: center;"/>
<input name="z" type="button" value="Z" onclick="filter('z')" style="width: 20px; text-align: center;"/>
<input name="ž" type="button" value="Ž" onclick="filter('ž')" style="width: 20px; text-align: center;"/>
<br/>
<br/>
<table style="float:left; width:768px; border:1px gray solid; margin:0 0 0 0; padding:2px; background-
    color:#CCCCCC">
<tr>
<td class="ms-vb" style="width:82px">Podjetje</td>
<td class="ms-vb" style="width:61px;">EVI</td>
<td class="ms-vb" style="width:221px;">NAZIV</td>
<td class="ms-vb" style="width:80px;">Interna</td>
<td class="ms-vb" style="width:80px;">MPO</td>
<td class="ms-vb" style="width:90px;">GSM</td>
<td class="ms-vb" style="width:65px;">Faks</td>
<td class="ms-vb">OE</td>
</tr>
</table>
<br/>
<br/>
<iframe id="contactFrame" style="width:100%; height: 482px; margin-bottom:-30px; padding-bottom:30px;"
    src="Results.aspx?ReturnAll=Yes" scrolling="auto" frameborder="no">
</iframe>
</div>
]]>
</Content>
</WebPart>

```

Slika 23: Spletni gradnik Stiki

Zgornji spletni gradnik je podan v deklarativnem načinu, v jeziku XML. Koda XML opisuje lastnosti spletnega gradnika. Izjema je element <Content>, ki ne vsebuje kode napisane v jeziku XML, ampak vsebuje posebno območje, ki se začne z zaporedjem znakov <![CDATA[ in se konča z zaporedjem znakov ]/>. Omenjeni zaporedji znakov sta za razčlenjevalnik XML (angl. XML parser) navodilo, da naj območja, ki ga omejujeta ne razčleni.

Koda v omenjenem območju je napisana v jeziku HTML in javascript. Služi pa prikazovanju elementov, ki jih vidi končni uporabnik: opisni tekst, vnosno polje za tekst, po katerem se vrši iskanje, gumbi z oznakami od A do Ž, glava tabele za prikaz rezultatov in pa element iframe, v katerem se izpišejo rezultati iskanja.

Na koncu območja CDATA se pojavi element iframe. Iframe je element jezika HTML, ki omogoča, da znotraj ene strani prikažemo neko drugo stran. V tem primeru se v elementu iframe strani default.aspx prikaže stran Results.aspx. Na strani Results.aspx se izvaja poizvedba po seznamu Stiki. Ko je zahteva po tej strani enkrat obdelana, se prikažejo rezultati iskanja v tabelarični obliki znotraj elementa iframe, ki pa se nahaja znotraj spletnega gradnika stiki.

Results.aspx vsebuje spletni gradnik DataFormWebPart (DFWP). DFWP je prišel skupaj z WSS 3.0 in uporablja kontrole ASP .NET 2.0 za dostop do podatkov, za prikaz le teh pa uporabi XSLT.

```
<WebPartPages:DataFormWebPart>
  <DataSources>
  <ParameterBindings>
  <DataFields>
  <Xsl>
    <xsl:stylesheet>
      ...
      ...
    </xsl:stylesheet>
  </Xsl>
</WebPartPages:DataFormWebPart>
```

Slika 24: Ogradje spletnega gradnika DataFormWebPart

Znotraj elementa <DataSources> podamo tip podatkov WSS, po katerih bomo vršili iskanje (v tem primeru je to seznam), poizvedbo za izbiro podatkov in parametre. Znotraj elementa <ParameterBindings> se parametre poveže z mestom, od kjer bodo črpali vrednosti, ki se bodo uporabile v poizvedbi. V elementu <DataFields> so našteje spremenljivke, ki so dosegljive znotraj Results.aspx. Ena spremenljivka je definirana s parom oblike @imeSpremenljivke, stolpecSeznamaStiki;. Element <Xsl> pa vsebuje predlogo, po kateri se bodo podatki pretvorili in prikazali na zaslonu.

Za podajanje poizvedbe za iskanje po telefonskem imeniku se uporabi jezik CAML (angl. Collaborative Application Markup Language). CAML je označevalni jezik, ki temelji na jeziku XML, in se uporablja v tehnologijah WSS in MOSS 2007. Z njim lahko določimo

polja, ki jih poizvedba vrne, filtriranje le teh in njihovo medsebojno razporeditev (naraščanje, padanje).

```
<View>
  <Query>
    <Where>
      <Or>
        ...
      <Or>
        <Or>
          <Or>
            <BeginsWith>
              <FieldRef Name='Priimek' />
              <Value Type='Text'>
                { nizZnakovPoKateremIscemo }
              </Value>
            </BeginsWith>
            <BeginsWith>
              <FieldRef Name='Ime' />
              <Value Type='Text'>
                { nizZnakovPoKateremIscemo }
              </Value>
            </BeginsWith>
          </Or>
        <BeginsWith>
          <FieldRef Name='PolnoIme' />
          <Value Type='Text'>
            { nizZnakovPoKateremIscemo }
          </Value>
        </BeginsWith>
      </Or>
    <BeginsWith>
      <FieldRef Name='EvidencnaSt' />
      <Value Type='Text'>
        { nizZnakovPoKateremIscemo }
      </Value>
    </BeginsWith>
  </Or>
  ...
</Where>
<OrderBy>
  <FieldRef Name='Priimek' />
  <FieldRef Name='Ime' />
</OrderBy>
</Query>
</View>
```

Slika 25: Oblika poizvedbe za iskanje po telefonskem imeniku

Zgornja poizvedba se uporabi za iskanje po seznamu Stiki. V rezultat iskanja se uvrsti vsak tisti stik, kateremu se vsaj eden od osmih parametrov (Ime, Priimek, PolnoIme, EvidencnaSt, Interna, GSM, MPO, Fax) začne z nizom znakov, po katerem iščemo.

Rezultati iskanja se nato prikažejo po navodilih XSLT predloge, ki se nahaja znotraj elementa XSL spletnega gradnika DataFormWebPart. Vsak zadetek, ki ga vrne poizvedba, se predstavi kot tabela dimenzije 1 x 8 (ena vrstica, 8 stolpcev). Tabele vseh zadetkov pa so med seboj razporejene ena pod drugo tako, da se spodnja in zgornja robova sosednjih tabel med seboj stikata. V primeru, da zadetkov ni, se izpiše tekst »Noben zapis ne ustreza iskani besedi.«.

```

<Xsl>
  <xsl:stylesheet xmlns:x="http://www.w3.org/2001/XMLSchema"
    xmlns:d="http://schemas.microsoft.com/sharepoint/dsp" version="1.0" exclude-result-
      prefixes="xsl msxsl ddwrt"
    xmlns:ddwrt="http://schemas.microsoft.com/WebParts/v2/DataView/runtime"
    xmlns:asp="http://schemas.microsoft.com/ASPNET/20"
    xmlns:__designer="http://schemas.microsoft.com/WebParts/v2/DataView/designer"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-microsoft-com:xslt"
    xmlns:SharePoint="Microsoft.SharePoint.WebControls" xmlns:ddwrt2="urn:frontpage:internal">
    <xsl:output method="html" indent="no"/>
    <xsl:decimal-format NaN="" />
    <xsl:param name="dvt_apos">'</xsl:param>
    <xsl:param name="ListID">{51CF2983-0B60-45A0-85FE-AF84CDB2199}</xsl:param>
    <xsl:param name="SearchQS" >r</xsl:param>
    <xsl:param name="_1"></xsl:param>
    <xsl:param name="_2"></xsl:param>
    <xsl:param name="_3"></xsl:param>
    <xsl:param name="_4"></xsl:param>
    <xsl:param name="_5"></xsl:param>
    <xsl:param name="_6"></xsl:param>
    <xsl:param name="_7"></xsl:param>
    <xsl:param name="_8"></xsl:param>
    <xsl:param name="_9"></xsl:param>
    <xsl:variable name="dvt_1_automode">0</xsl:variable>
    <xsl:template match="/" xmlns:x="http://www.w3.org/2001/XMLSchema"
      xmlns:d="http://schemas.microsoft.com/sharepoint/dsp"
      xmlns:asp="http://schemas.microsoft.com/ASPNET/20"
      xmlns:__designer="http://schemas.microsoft.com/WebParts/v2/DataView/designer"
      xmlns:SharePoint="Microsoft.SharePoint.WebControls">
      <xsl:call-template name="dvt_1"/>
    </xsl:template>
    <xsl:template name="dvt_1">
      <xsl:variable name="dvt_StyleName">Table</xsl:variable>
      <xsl:variable name="Rows" select="/dsQueryResponse/Rows/Row"/>
      <xsl:variable name="dvt_RowCount" select="count($Rows)"/>
      <xsl:variable name="dvt_IsEmpty" select="$dvt_RowCount = 0"/>
      <xsl:choose>
        <xsl:when test="$dvt_IsEmpty">
          <xsl:call-template name="dvt_1.empty" />
        </xsl:when>
        <xsl:otherwise>
          <div>
            <xsl:call-template name="dvt_1.body">
              <xsl:with-param name="Rows" select="$Rows" />
            </xsl:call-template>
          </div>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:template>
    <xsl:template name="dvt_1.body">
      <xsl:param name="Rows" />
      <xsl:for-each select="$Rows">
        <xsl:call-template name="dvt_1.rowview"/>
      </xsl:for-each>
    </xsl:template>
    <xsl:template name="dvt_1.rowview">
      <table style="float:left; width:768px; border:1px gray solid; background-color:#F5F5F5">
        <tr>
          <td class="ms-vb" style="width:80px;"><xsl:value-of select="@Podjetje"/></td>
          <td class="ms-vb" style="width:60px;"><xsl:value-of select="@EvidencnaSt"/></td>
          <td class="ms-vb" style="width:225px; color:black; font-weight:bold;"><xsl:value-of
            select="@PolnoIme"/></td>
          <td class="ms-vb" style="width:80px; color:#BF3018; font-weight:bold;"><xsl:value-of
            select="@Interna" /></td>
          <td class="ms-vb" style="width:80px;"><xsl:value-of select="@MPO" /></td>
          <td class="ms-vb" style="width:90px;"><xsl:value-of select="@GSM" /></td>
          <td class="ms-vb" style="width:63px;"><xsl:value-of select="@Fax" /></td>
          <td class="ms-vb"><xsl:value-of select="@OE" /></td>
        </tr>
      </table>
    </xsl:template>
    <xsl:template name="dvt_1.empty">

```

```

<xsl:variable name="dvt_ViewEmptyText">Noben zapis ne ustreza iskani besedi.
</xsl:variable>
<table border="0" width="100%">
  <tr>
    <td class="ms-vb">
      <xsl:value-of select="$dvt_ViewEmptyText"/>
    </td>
  </tr>
</table>
</xsl:template>
</xsl:stylesheet>
</Xsl>

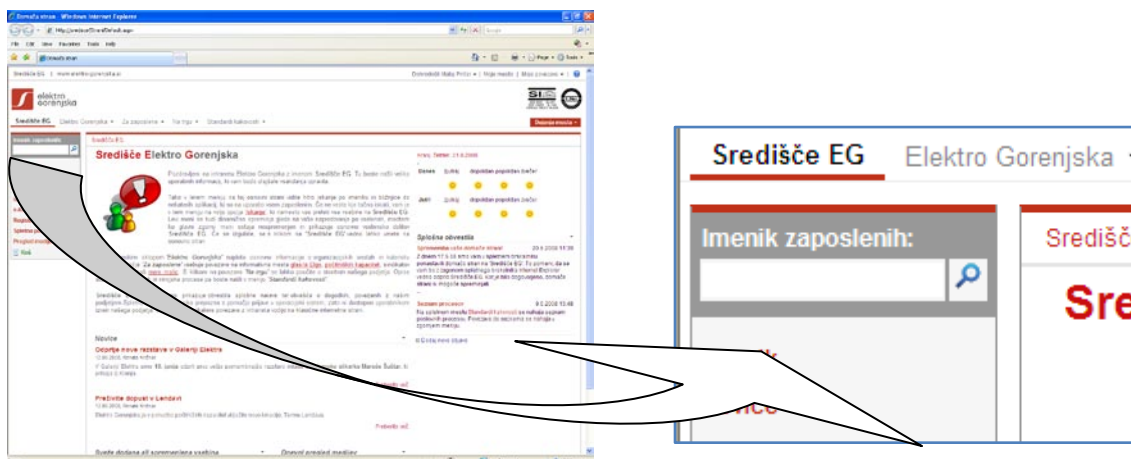
```

Slika 26: XSLT predloga, po navodilih katere se prikažejo rezultati

### 3.2 Tok dogodkov končnih uporabnikov:

Končni uporabniki lahko telefonski imenik uporabijo na tri načine:

1. Uporabnik iskanje začne na začetni strani portala »Središče EG«. Začetna stran portala »Središče EG« vsebuje vnosno polje za iskanje po telefonskem imeniku. V to polje uporabnik vnese niz znakov, po katerem želi izvršiti iskanje. S pritiskom tipke enter ali klikom miške na gumb poleg vnosnega polja se odpre aplikacija telefonskega imenika, v kateri se prikažejo rezultati iskanja. Uporabniki imajo začetno stran portala »Središče EG« nastavljeno kot domačo stran v brskalniku Internet Explorer. To pomeni, da je dostop do informacij telefonskega imenika hiter in enostaven. Prav zato si uporabniki najpogosteje poslužujejo tega načina za iskanje po telefonskem imeniku.



2. Uporabnik odpre spletno aplikacijo telefonskega imenika (klik na povezavo Imenik na začetni strani portala »Središče EG«). Nato v vnosno polje vnese niz znakov, s katerimi želi sprožiti iskanje. Poizvedba se izvede avtomatsko 400 milisekund po zadnji spremembi vrednosti v vnosnem polju, zato ni potrebno iskanja prožiti s klikom na gumb oz. s pritiskom tipke enter. Pod vnosnim poljem se prikažejo rezultati iskanja.

3. Uporabniki imajo pod vnosnim poljem za iskanje na začetni strani telefonskega imenika tudi gumbe označene od A do Ž. S pritiskom na enega od gumbov se sproži poizvedba, ki vrne tiste stike, katerim se vrednost v polju Polno ime začne na črko pritisnjene gumba.



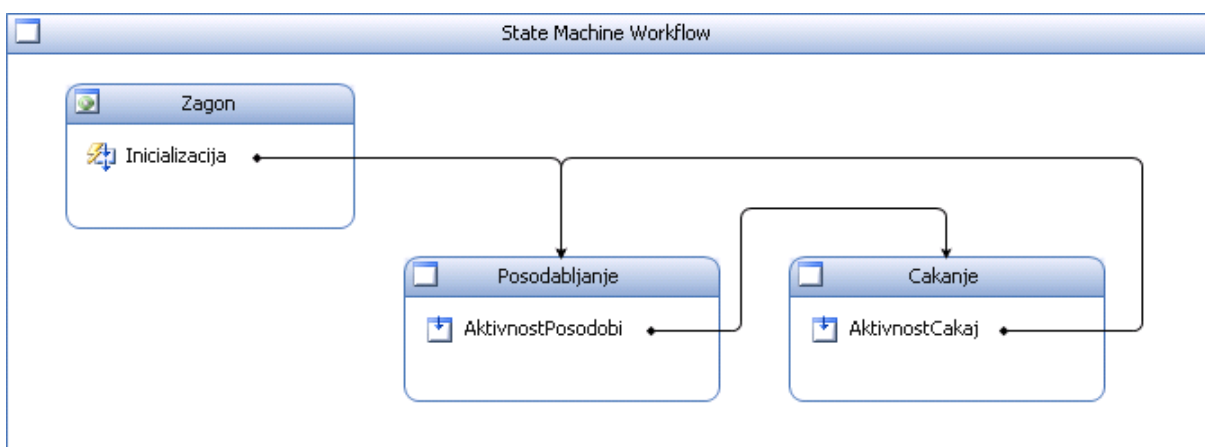
## 4 Posodabljanje stikov v telefonskem imeniku

V podjetju Elektro Gorenjska, prav tako kot v drugih podjetjih, prihaja do prerazporejanja zaposlenih z enega delovnega mesta na drugo, napredovanja, končanja delovnega razmerja, itd. Podatki o vseh naštetih situacijah se beležijo v podatkovni bazi, kar pomeni, da jih lahko uporabimo za posodabljanje stikov v telefonskem imeniku in s tem vedno zagotavljamo pravilnost informacij, ki jih le ta ponuja.

### 4.1 Rešitev s potekom dela

#### 4.1.1 Potek dela ImenikUpdaterWF

V svoji rešitvi sem uporabil vrsto poteka dela s stanji. Potek dela vključuje tri stanja: Zagon, Posodabljanje in Cakanje.



Slika 27: Zgradba poteka dela ImenikUpdaterWF (povezave med stanji)

#### Stanje Zagon

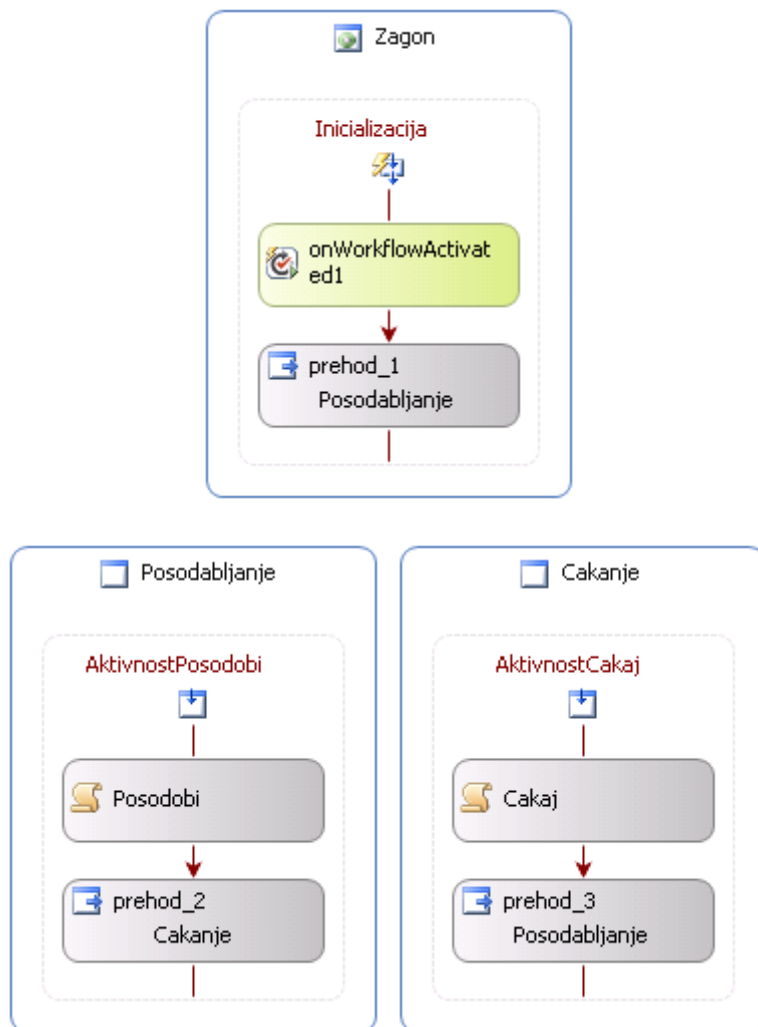
Stanje Zagon je označeno za začetno stanje. Ko se ustvari nov primerek poteka dela ImenikUpdaterWF, se izvajanje poteka dela začne v tem stanju. Brez pogojev se takoj zgodi prehod v stanje Posodabljanje.

#### Stanje Posodabljanje

V stanju Posodabljanje se izvede koda za posodabljanje telefonskega imenika. Ko je posodabljanje končano, potek dela spremeni stanje v stanje Cakanje.

## Stanje Cakanje

V stanju Cakanje se izračuna časovna razlika med trenutkom, ko je potek dela stopil v to stanje, in trenutkom, ko želimo, da se naslednjič izvede posodobitev. Za to časovno razliko izvajalna nit, ki izvaja potek dela `ImenikUpdaterWF`, zaspi. Ko časovna razlika preteče, se izvajalna nit prebudi in potek dela zamenja stanje nazaj v Posodabljanje.



Slika 28: Stanja poteka dela `ImenikUpdaterWF`

Opisani potek dela je modeliran ciklično in se bo izvajal v neskončnost, saj nima ustavitvenega pogoja oz. dogodka, ki bi sprožil prehod v neko četrto stanje, ki bi bilo končno.

## 4.2 Rešitev s časovnim opravilom

### 4.2.1 Časovno opravilo ImenikUpdaterJob

Časovno opravilo definiramo z razredom `ImenikUpdaterJob`, ki je podedovan iz razreda `Microsoft.SharePoint.Administration.SPJobDefinition`. Znotraj razreda pa ponovno definiramo metodo `Execute()`, ki vsebuje kodo opravila, ki se izvede ob izvajanju časovnega opravila.

#### Razred `ImenikUpdaterJob`

```
namespace SharePointJobs {
    public class ImenikUpdaterJob : SPJobDefinition
    {
        public ImenikUpdaterJob ()
            : base(){}

        public ImenikUpdaterJob (string jobName, SPService service,
            SPServer server, SPJobLockType targetType)
            : base (jobName, service, server, targetType){}

        public ImenikUpdaterJob (string jobName, SPWebApplication webApplication)
            : base (jobName, webApplication, null, SPJobLockType.ContentDatabase) {
            this.Title = "ImenikUpdaterJob";
        }

        public override void Execute (Guid contentDbId)
        {
            String strDashListRoot;
            SPSite site;
            SPWeb web;
            SPList list;
            SPListItemCollection listItems;

            strDashListRoot = "http://sredisce/imenik/Lists/Stiki/";
            site = new SPSite(strDashListRoot);
            web = site.OpenWeb();
            list = web.Lists["Stiki"];

            String evi;
            listItems = list.Items;
            int itemCount = listItems.Count;

            //zanka preko vseh elementov v seznamu Stiki
            for (int k = 0; k < itemCount; k++)
            {
                SPListItem listItem = listItems[k];

                //
                // sledi programska koda, ki po potrebi posodobi ali izbriše
                // k-ti element v seznamu Stiki
                //
            }
        }
    }
}
```

## Razred ImenikUpdaterJobInstaller

Za namestitev časovnega opravila ImenikUpdaterJob v sistem MOSS 2007 poskrbi razred ImenikUpdaterJobInstaller, ki je podedovan iz razreda Microsoft.SharePoint.SPFeatureReceiver in vsebuje dva upravljalca dogodkov, in sicer: FeatureActivated in FeatureDeactivating.

Ko časovno opravilo aktiviramo, se izvede upravljalca dogodkov FeatureActivated. V primeru, da pod imenom ImenikUpdaterJob že teče primerek časovnega opravila, se le ta najprej izbriše. Nato pa se ustvari nov primerek časovnega opravila ImenikUpdaterJob. Časovnemu opravilu se določi še urnik, po katerem se bo opravilo izvajalo.

Ob deaktiviranju časovnega opravila ImenikUpdaterJob pa se izvede upravljalca dogodkov FeatureDeactivating, ki pobriše časovno opravilo z imenom ImenikUpdaterJob iz množice aktiviranih opravil.

```
namespace SharePointJobs
{
    class ImenikUpdaterJobInstaller : SPFeatureReceiver
    {
        const string JOB_NAME = "ImenikUpdaterJob";

        public override void FeatureInstalled (SPFeatureReceiverProperties properties) {}

        public override void FeatureUninstalling (SPFeatureReceiverProperties properties) {}

        public override void FeatureActivated (SPFeatureReceiverProperties properties) {
            // kazalec na objekt, trenutne starni
            SPSite site = properties.Feature.Parent as SPSite;

            // ali je opravilo že registrirano (če je, ga izbrišemo)
            foreach (SPJobDefinition job in site.WebApplication.JobDefinitions) {
                if (job.Name == JOB_NAME)
                    job.Delete();
            }

            // namestimo novo opravilo
            ImenikUpdaterJob myImenikUpdaterJob = new ImenikUpdaterJob(JOB_NAME,
                site.WebApplication);

            // nastavimo urnik izvajanja za novo opravilo
            SPMMinuteSchedule schedule = new SPMMinuteSchedule();
            schedule.BeginSecond = 0;
            schedule.EndSecond = 59;
            schedule.Interval = 1440; // 1400 minut = 24 ur
            myImenikUpdaterJob.Schedule = schedule;

            myImenikUpdaterJob.Update();
        }

        public override void FeatureDeactivating (SPFeatureReceiverProperties properties) {
            SPSite site = properties.Feature.Parent as SPSite;

            // izbrišemo opravilo
            foreach (SPJobDefinition job in site.WebApplication.JobDefinitions) {
                if (job.Name == JOB_NAME)
                    job.Delete();
            }
        }
    }
}
```

### 4.3 Primerjava obeh rešitev

Z uporabo ene ali druge rešitve je končni rezultat enak. Seznam Stiki vsebuje podatke, ki so posodobljeni in odražajo trenutno stanje v podjetju. A vendar je pot do tega rezultata pri rešitvi s časovnim opraviлом veliko bolj elegantna.

Osnovni način uporabe poteka dela je, da se ob ustvarjanju novega ali pa ob spremembi že obstoječega elementa seznama ali knjižnice ustvari nov primerek poteka dela nad tem elementom. Torej ima lahko vsak element seznama ali knjižnice nič, enega ali celo več primerkov istega poteka dela naenkrat. Če bi problem posodabljanja rešili na tak način, bi imeli toliko primerkov poteka dela, kolikor imamo elementov (stikov) v seznamu Stiki. En primerek poteka dela pa bi v tem primeru skrbel za posodabljanje samo tistega elementa, kateremu pripada. Take rešitve vsekakor ne želimo.

Veliko boljša je rešitev, kjer obstaja samo en primerek poteka dela. V tem primeru bi z enim obhodom čez vse elemente seznama Stiki naenkrat posodobili vse elemente (stike).

V rešitvi z uporabo poteka dela `ImenikUpdaterWF` je uporabljena slednja možnost. Ročno je nad nekim elementom (samo enim) ustvarjen nov primerek poteka dela `ImenikUpdaterWF`. Problemi se pojavijo, če se potek dela izvaja ravno nad elementom, ki ga želimo izbrisati. Ob brisanju tega elementa izgubimo tudi primerek poteka dela, ki mu pripada, in s tem funkcionalnost, ki nam jo ponuja in ki jo potrebujemo. Lahko bi ustvarili element (stik v seznamu Stiki), ki bi obstajal samo zato, da bi na njem tekkel potek dela. S tem bi odpravili opisan problem, saj nikoli ne bi imeli razloga, da bi ta stik pobrisali (navidezen zaposleni pač ne more biti odpuščen). Spet pa se pojavi nova negativna stran rešitve, v seznamu Stiki bi imeli element, ki predstavlja osebo, ki ne obstaja.

Zaradi vseh zgoraj naštetih problemov raje uporabimo rešitev s časovnim opraviлом. Prav časovna opravila so tista, ki opravljajo naloge, ki tečejo neodvisno od akcij uporabnikov. Ko enkrat časovno opravilo `ImenikUpdaterJob` vnesemo v MOSS 2007, kontrolo nad upravljanjem s tem opraviлом prevzame storitev `SharePoint Timer`, ki je ena od storitev operacijskega sistema Windows. S tem imamo zagotovljeno nemoteno izvajanje funkcionalnosti, ki smo jo podali časovnemu opraviłu.



## 5 Zaključek

Rezultat mojega dela na področju, ki ga opisuje diplomska naloga je spletni portal »Središče EG« v podjetju Elektro Gorenjska, ki je že v uporabi. V okviru portala pa že od prvega dne, odkar je ta začel z delovanjem, uspešno deluje tudi novi interni telefonski imenik. S tem dnem, smo uporabnikom odvzeli tudi pravico do uporabe obeh starih različic telefonskega imenika, ki sta se uporabljali do takrat.

Prehod na novo aplikacijo telefonskega imenika uporabnikom ni predstavljal večjih težav. Odzivi uporabnikov na telefonski imenik in na celoten portal nasploh so pozitivni.

Nova aplikacija telefonskega imenika se od prejšnjih dveh različic iz zornega kota končnih uporabnikov razlikuje predvsem v tem, da deluje hitreje in da vsak trenutek ponuja točne informacije.

Pri delu sem naletel na nekaj težav, ki sem jih uspešno rešil. Tako sem npr. problem posodabljanja stikov najprej rešil s potekom dela. Kasneje pa sem ugotovil, da obstajajo časovna opravila, ki so namenjena prav takim nalogam, in jih uporabil v končni rešitvi.

Nadaljnje delo na področju, ki ga opisuje diplomska naloga, bo prilagajanje obstoječih funkcionalnosti in dodajanje določenih novih po zahtevah končnih uporabnikov.



## Uporabljeni viri

### Literatura:

- [1] Bill English, *Microsoft Office SharePoint Server 2007, Administrator's Companion*, 2007
- [2] Microsoft Official Course 5061A, *Implementing Microsoft Office SharePoint Server 2007*, 2007
- [3] 7 Development Projects for Microsoft Office SharePoint Server 2007 and Windows SharePoint Services Version 3.0
- [4] Margriet Bruggeman, Nikander Bruggeman, *Pro SharePoint 2007 Development Techniques*, 2007

### Internetni viri:

- [5] ASP.NET Master Pages Overview, <http://msdn.microsoft.com/en-us/library/wtxbf3hh.aspx>, (14.08.2008)
- [6] Creating Custom Timer Jobs in Windows SharePoint Services 3.0, <http://msdn.microsoft.com/en-us/library/cc406686.aspx>, (15.08.2008)
- [7] XML , <http://office.microsoft.com/sl-si/infopath/HP010967281060.aspx>, (04.09.2008)
- [8] Pretvarjanje datotek XML z jezikom XSLT, <http://office.microsoft.com/sl-si/access/HA010345761060.aspx>, (05.09.2008)
- [9] XSLT Transformations, <http://www.w3.org/TR/xslt>, (05.09.2008)



## **Izjava o avtorstvu**

Izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja prof. dr. Saše Divjaka. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Matej Pintar