



UNIVERZA V MARIBORU



FAKULTETA ZA ELEKTROTEHNIKO,  
RAČUNALNIŠTVO IN INFORMATIKO  
2000 Maribor, Smetanova ul. 17. 17

Diplomsko delo univerzitetnega študijskega programa

# ROBOTIMING

## PRIPRAVA PRIKAZA

Študent:	Marko Keber
Študijski program:	Elektrotehnika
Smer:	Avtomatika
Mentor:	doc. dr. Aleš Hace
Somentor:	doc. dr. Martin Terbuc

Maribor, september 2008





UNIVERZA V MARIBORU



FAKULTETA ZA ELEKTROTEHNIKO,  
RAČUNALNIŠTVO IN INFORMATIKO  
2000 Maribor, Smetanova ul. 17

Številka: E-2444  
Datum: 04. 04. 2007

### SKLEP O DIPLOMSKEM DELU

1. **Marko Keber**, študent univerzitetnega študijskega programa Elektrotehnika, smer Avtomatika, izpolnjuje pogoje, zato se mu dovoljuje izdelati diplomsko delo.
2. **Tema diplomskega dela** je s področja Inštituta za avtomatiko pri predmetu **PROGRAMSKO INŽENIRSTVO**.

**MENTOR:** doc. dr. Aleš Hace  
**KOMENTOR:** doc. dr. Martin Terbuc

3. **Naslov diplomskega dela:**  
**ROBOTIMING PRIPRAVA PRIKAZA**
4. **Vsebina diplomskega dela:**  
Pri robotskem tekmovanju FERI je pomemben sprotni prikaz rezultatov. Pripravite okolje, ki bo omogočalo prikaz rezultatov v času tekmovanja. Izvedena naj bo povezava z nadzorno aplikacijo za meritev. Izdelajte uporabniški vmesnik tako, da bo možna lokalna uporaba in uporaba preko spleta.
5. Diplomsko delo je potrebno izdelati skladno z "Navodili za izdelavo diplomskega dela" in ga oddati v treh izvodih do 04. 04. 2008 v referatu za študentske zadeve.

PREDSTOJNIK INŠTITUTA

izred. prof. dr. Boris Tovornik



DEKAN

red. prof. dr. Igor Tičar

doc. dr. Aleš Hace



## ZAHVALA

Somentorju doc. dr. Martinu Terbucu za kvalitetno in vsestransko pomoč pri nastajanju projekta Robotiming ter višjemu predavatelju Janezu Pogorelcu za pomoč in sodelovanje pri projektu, ki smo ga lahko preizkusili na tekmovanju v vožnji z mobilnimi roboti. Posebna zahvala gre doc. dr. Alešu Hacetu za dobro opravljeno mentorstvo ter pomoč pri nastajanju diplomskega dela.



# ROBOTIMING PRIPRAVA PRIKAZA

**Ključne besede:** robotiming, robotsko tekmovanje RoboT, štopanje, programska oprema, spletne aplikacije

**UDK:** 004.4:007.52(043.2)

**Povzetek:** Diplomsko delo opisuje izdelavo programske opreme za brezkontaktno merjenje časa in njegov prikaz na vsakoletnem tekmovanju z mobilnimi roboti RoboT. Poleg splošnega razvoja tekmovanja RoboT so opisane tehnologije, ki so bile uporabljene za projekt Robotiming, izdelava prikaza na velikem zaslonu, izdelava programske opreme za spremljanje robotskega tekmovanja preko interneta ter delovanje in uporaba obeh sistemov. Prav tako je natančno opisan postopek prilagajanja aplikacije za uporabo na vsakoletnem robotskem tekmovanju zaradi vsakoletnih sprememb kot so na primer letnice, sponzorji, uvodni nagovori, sprememba naslovov strežnikov.





# DEVELOPMENT OF ROBOTIMING DISPLAY

**Key words:** Robotiming, stop watching, software, web application, robot contest  
**RoboT**

**UDK:** 004.4:007.52(043.2)

**Abstract:** This diploma work features development of software used for contactless stop watching and its projections on the yearly competition with mobile robots called RoboT. Beside historical development of RoboT, technologies which were used with Robotiming are being explained along with development of projection on the big screen, software for watching RoboT over internet and explanation how both systems works. Moreover, the procedures to adjust the software for yearly RoboT competition are explained, for example: change of graphics, texts and system settings.



## Uporabljene kratice

WYSIWYG	What You See Is What You Get	Opisuje sistem pri katerem je vsebina že med urejanjem zelo podobna končnemu produktu
BPMN	Business Process Modeling Notation	grafična notacija za modeliranje poslovnih procesov
CSS	Cascade Style Sheets	kaskadni listi, html dokumenti, ki vsebujejo več nivojev stilov
ARM	Advanced RISC Machine	Napredne RISC naprave
HTML	HyperText Markup Language	Označevalni jezik za oblikovanje večpredstavnostnih dokumentov, ki omogoča povezave znotraj dokumenta ali med dokumenti
XML	Extensible Markup Language	Razširljivi označevalni jezik
MP3	MPEG-1 Audio Layer-3	tehnologija za stiskanje in shranjevanje zvočnih posnetkov
PNG	Portable Network Graphics	grafični format za stiskanje slik
PDF	Portable Document Format	Format za prenašanje dokumentov
PSD	Photoshop Document	Tip zapisa v katerega zapisuje Adobe Photoshop
FTP	File Transfer Protocol	Protokol za prenos datotek
RGB	Red Green Blue	osnovne barve za prikaz na zaslonu
CMYK	Cyan Magenta Yellow Black	Višnjeva škrlatna rumena črna



# Kazalo

1 UVOD.....	1
2 SPLOŠNO O ROBOTSKEM TEKMOVANJU .....	3
2.1 Robotsko tekmovanje skozi čas .....	5
2.2 Zakaj je bila uvedba elektronskega merjenja časa nujno potrebna .....	6
3 SPLOŠNO O ROBOTIMING-U.....	7
3.1 Splošno o uporabi flasha za namene grafičnih vmesnikov v vgrajenih sistemih....	7
3.2 Prednosti vgradnih Flashev .....	8
3.3 Bogate internet aplikacije [5] .....	13
3.4 Metode in tehnike .....	15
4 POSTOPEK IZDELAVE PRIKAZA ROBOTIMING .....	19
4.1 Izdelava prikaza na samem tekmovanju.....	19
4.2 Prikaz merjenja časa in rezultatov v Flashu .....	35
4.3 Spreminjanje grafičnega vmesnika v Photoshop-u .....	46
5 SKLEP.....	51
DODATEK A - SPLOŠNO O PLATFORMI FLASH.....	53
DODATEK B - SPLOŠNO O MACROMEDIA DREAMWEAVER.....	69
DODATEK C - SPLOŠNO O ADOBE PHOTOSHOP CS2 .....	83
DODATEK D - BPMN.....	89



## Slike

Slika 3.1: Celotna družina rešitev za nadzor in upravljanje z »inteligentno« hišo, katere vmesniki temeljijo na flash tehnologiji.....	9
Slika 3.2: Nadzorni video sistem vgrajen v stavbo, ki deluje na flash platformi. ....	10
Slika 3.3: Adobe Flash za vgradnjo in njegova programska arhitektura.....	10
Slika 3.4: Uporabniški vmesnik narejen s tehnologijo Flash za namene vremenske postaje, video nazdora oziroma aplikacije inteligentna hiša.....	12
Slika 4.5: Prikaz na tekmovanju – uvodna stran.....	20
Slika 4.6: Končni izdelek oblikovanja glave za namen prikaza na tekmovanju in spletu .....	21
Slika 4.7: Pozdravno sporočilo z urnikom.....	22
Slika 4.8: Oglaševalski prostor .....	22
Slika 4.9: Prikaz merjenja časa .....	23
Slika 4.10: Prikaz rezultatov v študentski kategoriji .....	24
Slika 4.11: Prikaz štartne liste.....	25
Slika 4.12: Tabela rezultatov .....	26
Slika 4.13: Uvodna stran je zgrajena iz menija, glave in vsebinskega dela. ....	27
Slika 4.14: Izdelava podmenija.....	29
Slika 4.15: Izdelava menijev za spletno stran v Adobe Flashu in končna postavitvev letih v Adobe Dreamweaverju .....	30
Slika 4.16: Izdelava gumba, pri katerem se ob prehodu miške preko statične slike sproži video posnetek .....	31
Slika 6.17: Nastavitev odmikov od roba in slike za ozadje spletne strani.....	32





Slika 6.18: Vnos dodatnega parametra, ki naredi ozadje Flash animacije transparentno - brezbarvno .....	32
Slika 4.19: Spletna stran in prikaz štartne liste v dijaški kategoriji.....	33
Slika 4.20: Prikaz rezultatov in pretočnega videa v živo s tekmovanja .....	34
Slika 4.21: Prikaz vseh trenutnih/končnih rezultatov vseh tekmovalcev .....	34
Slika 4.22: Diagram poteka prikaza merjenja časa .....	36
Slika 4.23: Prikaz merjenja časa v času testnih voženj .....	37
Slika 4.24: Osnovne nastavitve Flash animacije .....	37
Slika 4.25: Nastavitve dinamičnega tekstovnega polja, ki prikazuje minute vmesnega časa .....	38
Slika 4.26: Shematski prikaz pridobivanja podatkov, potrebnih za animacijo ter prikaz rezultatov in uvrstitev .....	41
Slika 4.27: Uporaba BPMN za grafični opis procesov in medsebojno izmenjevanje podatkov pri prikazu in merjenju časa na tekmovanju .....	45
Slika B.1: Dreamweaver omogoča simultano delo v vizualnem WYSIWYG urejevalniku .....	66
Slika B.2: Interaktivni zemljevid spletne strani, ki je služila za prikaz rezultatov v živo na spletu .....	67
Slika B.3: Urejanje CSS stila deluje s panelom za urejanje oznak, s katerim urejamo stile .....	74
Slika B.4: V razširjenem načinu dela s tabelami so začasno prikazani robovi s povečano odebelitvijo in razmikom med celicami, kar omogoča veliko lažje urejanje .....	75
Slika B.5: Spry zavihki.....	76
Slika C.1: Adobe Photoshop - orodje namenjeno oblikovanju .....	84



---

# POGLAVJE 1

---

## 1 UVOD

Že pred našim štetjem si je človek prizadeval, da bi s pomočjo nekega naravnega pojava izmeril pretečeni čas in s tem bil sposoben tudi numerično opisati trajanje posameznih dogodkov. V času življenja ljudi v preprostih agrarnih skupnostih, se je čas meril predvsem na podlagi opazovanja dneva in noči ter spreminjanja letnih časov. Čas je bil tako povsem podrejen pojavom na nebu. Tako ni čudno, da so prve znanstvene definicije, s katerimi si je človek pomagal pri štetju časa, temeljile ravno na opazovanju astronomskih pojavov. Leta 742 pr. n. št. se je pojavila prva pristna omemba sončne ure, ki je tako razbila dan na ure. Velik napredek v merjenju časa je predstavljal tudi sončev koledar s prestopnimi leti, ki sta ga leta 46 pr. n. št. razvila Julij Cezar in Sosigen. Koledar je bil relativno natančen, vendar je z napačnimi upoštevanji astronomskih pojavov na vsakih 134 let pridobil za en dan. Kakorkoli pa je bila za tisti čas taka natančnost zadostna. Leta 1582 je pomanjkljivi julijanski koledar nadomestil gregorijanski, ki ga uporabljamo še danes. V Milanu se leta 1335 prvič pojavi mehanska ura, leta 1656 Christiaan Huygens izdela prvo natančno uro na nihalo in sledi izum kvarčne ure leta 1928. Da ne bi prihajalo do časovnih razhajanj med različnimi državami in deželami, je že leta 1884 Greenwich v Angliji (0 stopinj geografske dolžine) postal osnova za čas po svetu. Leta 1967 se končno definira 1 sekunda z nihanjem valovanja cezijevega atoma  $^{133}\text{Cs}$ , kar je bila tudi osnova za vse nadaljnje atomske ure in z zaostajanjem manj kot ene sekunde v petih milijonih letih, veljajo za najbolj točne ure na svetu.

Potrebe po samem merjenju časa in njegovi točnosti so se skozi zgodovino nenehno povečevale, tako v znanosti kot tudi v povsem vsakdanjem človeškem življenju. Med drugim tudi na področju športa, kot je atletika, plavanje, smučanje, kolesarjenje ali avtomobilizem, kjer so športniki skozi čas postali zelo izenačeni ter so zato pomembne postale tudi desetinke, stotinke in celo tisočinke sekunde. Merjenje časa je postalo zelo pomembno tudi v industriji, predvsem pri upravljanju in nadzoru kompleksnih in hitrih proizvodnih linij. Včasih so za proženje merjenja časa uporabljali mehanske prožilce, dandanes pa lahko prečkanje določene linije (npr. start, stop) določenega predmeta zaznamo brezkontaktno s pomočjo optičnih senzorjev in mu na tak način izmerimo čas.

---

## POGLAVJE 2

---

### 2 SPLOŠNO O ROBOTSKEM TEKMOVANJU

Kot že samo ime pove, gre za tekmovanje z mobilnimi roboti lastne konstrukcije, ki ima v osnovi študentsko in dijaško kategorijo. Namen tekmovanja je spodbujati mlade k ustvarjalnosti, inovativnosti in pridobivanju novih znanj na področjih robotike, elektronike in avtomatike. Prav tako pa gre za spodbujanje veščin programiranja in gradnje gibajočih se elektromehanskih naprav (gradniki so servomotorji iz R/C (radijsko vodenih) modelov, senzorji, mikrokrmilniki, vse pa povezuje programska oprema, razvita na običajnem osebem računalniku) [1]. Seveda sovpada s tekmovanjem tudi druženje in tekmovalni duh, ki je ves čas prisoten med tekmovalci oziroma ustvarjalci nadobudnih robotov. Le-ti šele na tekmovanju pokažejo kako dobro so jih njihovi lastniki in ustvarjalci skozi celo leto pripravljali na finalno preizkušnjo.

Finalno tekmovanje RoboT 200x se organizira letno na Univerzi v Mariboru, Fakulteti za elektrotehniko, računalništvo in informatiko, Smetanova ulica 17 in poteka v disciplini: vožnja po labirintu. "Labirint znanja" je površina 2 krat 2,5 m, pregrajena s stenami, tako da je možnih več poti od starta do cilja in najkrajša je "prava pot", ki jo mora mini mobilni robot najti in prevoziti v treh poskusih v čim krajšem času. Tekmovanje se hkrati organizira za študente (obeh univerz, gostujočih tujih študentov po programu Socrates/Erasmus in višjih strokovnih šol Slovenije) ter za dijake srednjih šol Slovenije (izjemoma se lahko prijavijo tudi učenci višjih razredov osnovnih šol). Študentje (A) in dijaki (B) nastopajo vsak v svoji kategoriji.

Pravico nastopa v finalu imajo vsi, ki so uspešno nastopili na izbirnem tekmovanju (regijska izbirna tekmovanja oziroma slovensko izbirno tekmovanje). Vsaka ekipa (največ trije študentje ali dijaki) ima lahko svojega mentorja, predstavi pa se z (enim) izdelkom - mini mobilnim robotom lastne konstrukcije.

Cilj tekmovanja je gradnja in programiranje avtonomnega mobilnega robota ter prikaz navigacijskih in orientacijskih sposobnosti ob vožnji po labirintu. Vozilo (mini mobilni robot) mora brez posredovanja tekmovalca prevoziti pot od startne pozicije (START) do ciljne pozicije (CILJ) v čim krajšem času. Dober rezultat je mogoče doseči ali z zelo hitro (brutalno) vožnjo po vseh možnih poteh ali pa z inteligentnim načinom vožnje, kjer se robot ob vsakokratnem poskusu uči (pomni) ovire.

Avtonomni mobilni robot ima lastni pogon, napajanje in vnaprej programirano krmilno logiko bodisi vgrajeno bodisi implementirano v stacionarnem računalniku z brezžično zvezo z vozilom; za razpoznavanje ovir v okolici ima lahko vgrajene infrardeče (IR) ultrazvočne in/ali druge senzorje. Najpomembnejša omejitev glede izvedbe mobilnih robotkov je avtonomnost, torej ko vozilo starta, ni več dovoljena nobena interakcija med lastnikom in robotkom. Vozilo se sicer lahko dotakne stene, ampak je ne sme premikati v smislu ustvarjanja prednosti. Robot, ki bi poskušal intenzivno premikati stene, bi bil v tistem poskusu vožnje diskvalificiran. Posebne omejitve v zvezi z velikostjo in maso robota niso potrebne oziroma so pogojene z velikostjo koridorjev poligona. Nastop s komercialnimi sestavljenimi roboti ni možen. Pri mini mobilnem robotu mora biti vsaj ena komponenta (senzorski sklop, pogonski sklop, ohišje z baterijo, krmilnik) lastni izdelek.

Labirint, po katerem tekmujejo roboti, je fizično omejen na področje 250 x 200 cm (površina iz oplemenitene bele iverne plošče), zunanje in notranje stene so iz ivernih belih plošč višine 10 cm in debeline 18 oz. 12 mm. Najmanjša razdalja med stenami koridorja je 20 cm, povprečna 25 cm in maksimalna 30 cm. Labirint je klasični dvo-dimenzionalni z eno ali več poti od Starta do Cilja, ki sta locirana ob zunanjih stenah. Sledenje bodisi leve bodisi desne stene ne sme vplivati na rezultat.

Tekmovanje poteka tako, da ima vsak robot na voljo največ po tri vožnje oziroma poizkuse, od katerih se upošteva vožnja robota z najboljšim oziroma najkrajšim časom. Robot, ki ima najkrajši čas, je zmagovalec. V primeru, da robot v treh minutah ne pride do cilja, se upošteva največja razdalja, ki jo je dosegel robot od starta. V praksi se le-ta odčita

s pomočjo kvadrantov, ki so glede na razdalje od starta ustrezno točkovno obteženi. Čas se meri ročno in elektronsko. Pri elektronskem merjenju se uporabljajo senzorji sprožitve z ločljivostjo 0,01 s.

## **2.1      *Robotsko tekmovanje skozi čas***

Začetki študentskega tekmovanja v vožnji z mobilnimi roboti RoboT segajo v leto 1999. Takrat se je relativno malo študentov prvič pomerilo, najprej v predtekmovanju v sledenju določeni poti in dobra dva meseca kasneje tudi v finalnem tekmovanju - vožnji po labirintu. Na tekmovanje se je prijavilo enajst tekmovalcev, ki so tekmovali v dveh ekipah, in sicer z mobilnimi robotki lastne gradnje in komercialnimi mobilnimi robotki Boe-BOT. Tekmovanje je bilo medijsko zelo zanimivo, saj je šlo za eno izmed prvih robotskih tekmovanj v Sloveniji sploh. Zanimanje za tekmovanje je skokovito naraslo, kar se je leta 2001 pokazalo tudi v dvakrat večji udeležbi tekmovalcev med študenti. Tega leta se je zaradi velikega zanimanja med dijaki dodala tudi dijaška kategorija, ki je bila že prvič številčno nekoliko močnejše zastopana kot študentska kategorija. Tudi to leto so roboti tekmovali v predtekmovanju po začrtani liniji, v finalni vožnji pa so se pomerili v velikem labirintu, katerega velikosti so ostale nespremenjene do danes. Zanimivo je tudi, da je bil zaostanek drugouvrščenega tekmovalca za takrat vodečim robotom več kot štirideset sekund in tudi ostale razlike med tekmovalci so bile redko manjše od desetih sekund. Leta 2002 je Fakulteta za elektrotehniko in računalništvo v Mariboru ponovno gostila najboljše študente in dijake, ki so se pomerili tokrat samo v finalni vožnji po labirintu. Razlike med tekmovalci so se zmanjšale tudi na manj kot 5 sekund, kar je bil dober pokazatelj, da so se roboti iz leta v leto izpopolnjevali in tako dvigovali kvaliteto samega tekmovanja. Tudi obiskanost se je v obeh kategorijah konstantno povečevala in tako kazala na vedno večje zanimanje za mnoge nenavadno tekmovanje, ki je pridobivalo na popularnosti tudi med samimi gledalci. V takem trendu se je leta 2003 izvedlo že četrto študentsko in tretje dijaško tekmovanje, ki je postreglo z zelo izenačenim končnim rezultatom, saj je takrat razlika med drugo uvrščenim in zmagovalcem prvič znašala manj kot sekundo in sicer [2] natanko 0.8 sekunde. Takrat so se prvič pojavila ugibanja in dvomi v samo natančnost

merjenja časa. Le-ta se je do leta 2003 na tekmovanjih RoboT meril ročno, in sicer je bil čas vožnje robota povprečje izmerjenega časa treh merilcev, ki so uporabljali navadno štoparico. Poleg tega pa je bilo merjenje časa izvedeno dokaj nepregledno za gledalce. Merilci so takrat končne čase voženj robotov sporočali vodji tekmovanja, ki jih je beležil in šele po končanem tekmovanju razglasil uvrstitve in čase voženj. Postalo je jasno, da bodo razlike na prihodnjih tekmovanjih lahko tudi manjše od 0.2 sekundi, uporaba tedanje metode ročnega merjenja časa pa neprimerna, nenatančna ter neatraktivna za gledalce.

## ***2.2 Zakaj je bila uvedba elektronskega merjenja časa nujno potrebna***

Da bi šla tehnična izvedba robotskega tekmovanja v korak s časom in predvsem razvojem tekmovalnih robotov, se je porodila ideja o konceptu merjenja časa in njegovem sprotnem prikazu ter prenosu tekmovanja v živo preko spleta tako živega prenosa videa kot tudi trenutnih časov in razvrstitev.



---

## POGLAVJE 3

---

### 3 SPLOŠNO O ROBOTIMING-U

Rdeča nit pri snovanju sistema, ki smo ga poimenovali RoboTiming, je bila predvsem izboljšati kvaliteto tekmovanja z mobilnimi roboti RoboT. Obenem pa smo želeli tekmovanje narediti bolj zanimivo, tehnološko dovršeno ter ga približati širši populaciji z uporabo najnaprednejših programskih orodij in tehnologij. Zastavili smo si cilj, da naredimo časomerilni sistem, ki bo v prvi vrsti meril končni in vmesni čas posameznih voženj robotov. Poleg tega pa bo sposoben kompletnega prikaza vseh informacij, ki so pomembne tako za tekmovalce kot tudi gledalce. Z uporabo spletnih tehnologij pa so bile te informacije vidne ne samo na kraju tekmovanja, ampak tudi širom Slovenije oziroma sveta.

#### *3.1 Splošno o uporabi flasha za namene grafičnih vmesnikov v vgrajenih sistemih [3]*

Adobe Flash je zanimiv za vgrajene sisteme predvsem zaradi dejstva, da vgradni Flash ponuja enake funkcije, animacije, kot jih ponuja namizni/internetni ekvivalent. Prav tako Flash prikazuje vsebino, ki teče na vgrajenih sistemih na enak način kot vsebino prikazuje Flash na namizju. Razvoj Flash animacij lahko poteka na katerikoli platformi, ki podpira

Adobe Flash razvojna orodja. Dejstvo je da je Adobe Flash tudi v praksi postal standard za razvoj naprednih in bogatih uporabniških vmesnikov. Postal je prisoten vsepovsod na spletu in najbolj priljubljena tehnologija med oblikovalci in programerji, s katero se da ustvariti prepričljive uporabniške izkušnje. Posledično pa sedaj Adobe Flash postaja tudi nujna programska oprema vgrajenih sistemov, ki se povezujejo na splet. Vendar pa lahko Flash vgrajenim sistemom ponudi veliko več, kot le kompatibilnost s spletnimi brskalniki. Adobe Flash lahko služi kot platforma na kateri lahko zgradimo profesionalne, uporabniške vmesnike, ki dajo našim vgrajenim sistemom dodano vrednost. Enaka orodja in izrisovalni pogoni, ki so uporabljeni za izdelavo in prikaz Flash vsebine v namizni aplikaciji na spletu lahko prav tako uporabimo tudi za razvoj vgrajenih aplikacij. Največje razlike med namiznim in vgradnim Flash razvojem ter namestitvijo se kažejo predvsem kot različni velikosti zaslona in zmogljivosti procesorja. Vgrajeni sistemi imajo običajno manjše zaslone, manj procesorske moči in omejeno količino pomnilniških sredstev v primerjavi z namiznimi računalniki.

### **3.2 *Prednosti vgradnih Flashev***

Velika prednost Flasha je, da lahko predstavlja osnovo za uporabniški vmesnik, ki je konsistenten med vsemi končnimi produkti nekega podjetja, ne glede na procesor, operacijski sistem ali druge različice. To je izvedljivo, ker Flash omogoča, da je uporabniški vmesnik implementiran s pomočjo vsebine in ne skozi programsko opremo vgradnega sistema. Prav tako je Adobe Flash predvajalnik kot tak neodvisen od procesorja kot tudi operacijskega sistema, saj gre za ti. interpretiran programski jezik. Poenoten izgled in zunanji vtis programske opreme je zelo pomemben adut pri uveljavljanju določene znamke podjetja. Po drugi strani pa pomaga pri zmanjševanju potrebe po nujenju tehnične podpore, saj se morajo uporabniki naučiti le enega uporabniškega vmesnika. Prav tako je poenoten uporabniški vmesnik možno zadržati kljub neizogibnim strojnimi in programskim spremembam, ki bodo doletele proizvode nekega podjetja v prihodnosti.

Potrebno je izpostaviti še eno prednost, ki jo dosežemo s pomočjo uporabniškega vmesnika narejenega v Flashu. In sicer se nam pri izbiri takega pristopa na stežaj odpirajo možnosti

uporabe že narejenega in dostopnega grafičnega materiala ter s tem omejitve kreativnosti predstavlja le naša domišljija. Na tak način lahko oblikovalci v produkt, ki ga razvijajo, vgradijo res vse pomembne in za končnega uporabnika privlačne elemente. Ves razvoj tega vmesnika poteka s pomočjo standardnega razvojnega paketa Flash. To predstavlja relativno enostaven in cenovno ugoden razvoj, vzdrževanje, nadgradnjo uporabniškega vmesnika in prenos na ostale sorodne produkte in produkte prihodnjih generacij. Vzemimo za primer proizvodnjo linijo izdelkov, ki vsebuje fiksni model, majhen prenosni model in kompakten ročni model (sl. 3.1). Vse tri vrste izdelkov iz te linije uporabljajo različne procesorje in operacijske sisteme. Implementacija identičnih ali vsaj podobnih uporabniških vmesnikov na vseh teh napravah bi bilo pogumno opravilo. Z uporabo Flash tehnologije pa je relativno enostavno razviti en, enostaven in poenoten vmesnik.



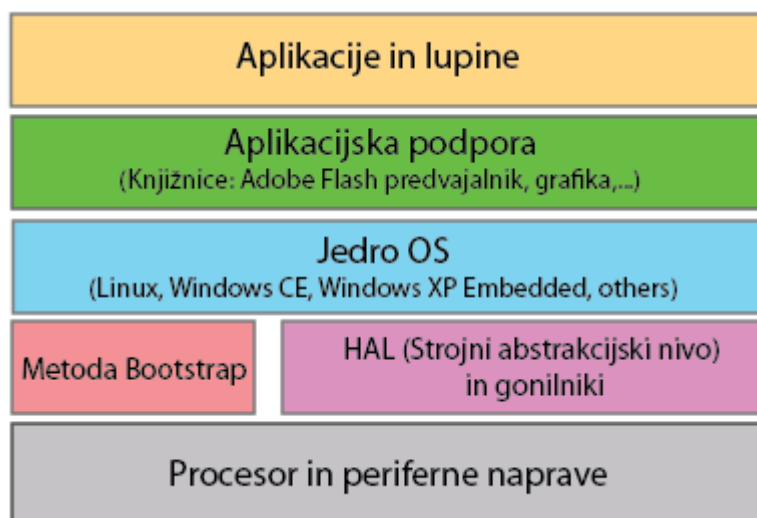
Slika 3.1: Celotna družina rešitev za nadzor in upravljanje z »inteligentno« hišo, katere vmesniki temeljijo na flash tehnologiji

Dobri grafični vmesniki so še posebej pomembni, npr. na kiosk sistemih, kjer ljudje nakupujejo in nadzornih sistemih za varovanje objektov pritrjenih na steno (sl 3.2). Od takih naprav se pričakuje poenoten vmesnik po vseh trgovinah ali domovih, ne glede na izkušnje uporabnika, saj imajo lahko nekatere lokacije novejšo ali drugačno različico produkta. V takih primerih konsistentnost zelo veliko prispeva k zadovoljstvu kupca. Za podjetje je predvsem pomembno, da njihov uporabniški vmesnik, ki je narejen za neko napravo, predstavlja veliko konkurenčno prednost, kar pomeni, da izboljša dohodek, profitabilnost in direktno ali indirektno izboljša denarni pretok. Rešitve, ki so osnovane na Flash tehnologiji, so lahko dosti bolj konkurenčne ter iz vidika stroškov in razvojnih sredstev ugodna alternativa.



Slika 3.2: Nadzorni video sistem vgrajen v stavbo, ki deluje na flash platformi.

Kljub vsemu pa je potrebno poudariti, da čeprav je Adobe Flash neodvisen glede na procesor in operacijski sistem, ima vseeno določene zahteve na področju strojne opreme in velikosti pomnilniških virov. Katerikoli procesor z arhitekturo x86 in z vsaj 366MHz ali ARM ali XScale procesor pri frekvenci 400 MHz zagotavlja sprejemljivo zmogljivost. Sam Flash predvajalnik zaseda nekje 1-1,5 Mb prostora.



Slika 3.3: Adobe Flash za vgradnjo in njegova programska arhitektura

Adobe Flash predvajalnik je mogoče prenesti na x86, ARM, XScale, PPC, MIPS in SH4 procesorje in na Windows XP Embedded, Windows CE, Linux, QNX in BeOS operacijske sisteme. Glede na to, da Flash deluje na aplikacijskem nivoju (sl. 3.3) in ne komunicira direktno s strojno opremo, je migracija med temi sistemi za razvijalce z ustreznimi izkušnjami relativno enostavna.

Sama optimizacija ali razširitev Adobe Flash predvajalnika zahteva širše znanje in globlje razvojne izkušnje kot pri le enostavni migraciji. Glede na zahteve je potrebno različno znanje, od strojnega programiranja, grafičnega izrisovanja, pisanje dodatne podporne programske opreme in znanje jedra operacijskega sistema. Obenem pa je potrebno podrobno znanje in razumevanje izvorne kode Flash predvajalnika. Ko je izvršena migracija, je potrebno spremenjeno verzijo Flash predvajalnika testirati na določeni strojni opremi in dobiti verifikacijo s strani Adobe, preden lahko gre končni produkt v prodajo.

Adobe prav tako ponuja razvojno programsko okolje z izvorno kodo, orodje za testiranje in dokumentacijo. Vse to je namenjeno za proizvajalce, ki bodo serijsko vgrajevali Flash v svoje produkte.

#### **Primer uporabe Adobe Flash tehnologije v industriji [4]**

Podjetje Convergent Living Inc. se ukvarja z izdelavo izdelkov, ki so namenjeni centralnemu upravljanju domačih naprav (sl. 3.4). Njihova naloga je bila razviti zadovoljiv, dinamičen uporabniški vmesnik za njihov produkt iz družine Companion, ki je namenjen kontroli nad domačimi napravami. Glavne zahteve, ki so jih imeli pri razvoju tega produkta so bile, da mora aplikacija delovati samostojno (in ne v spletnem brskalniku) ter da mora biti zasnovana na odprti, dostopni tehnologiji. Potrebno je bilo zagotoviti enostavno uporabo trgovcem, ki prodajajo Companion po domovih in podjetjih ter tudi prodajalcem, ki so vgrajevali Companion v svoje izdelke, npr. proizvajalci osvetljav ali domačega kina.



Slika 3.4: Uporabniški vmesnik, narejen s tehnologijo Flash za namene vremenske postaje, video nadzora oziroma aplikacije inteligentna hiša.

Podjetje Convergent Living je s svojo družino Companion zanetilo pravo malo revolucijo na področju oblikovanja uporabniških vmesnikov. Njihovi produkti iz te družine so vgrajeni v najbolj naprednih domovih in pisarnah, imajo male, na dotik občutljive zaslone, ki imajo moč združevati nadzor nad vsemi varnostnimi napravami, domačim kinom, zvokom, razsvetljavo in okoliške sisteme. Nadzorni center ni samo prijeten kos strojne opreme, ki visi na steni, ampak lahko prikazuje tudi privlačno grafiko, kot so npr. digitalne fotografije. Prav tako je možno nastaviti svetlobno pot luči, npr. iz garaže pa vse do spalnice.

Glede na to, da podjetje Convergent Living v veliki večini prodaja uporabniško izkušnjo nadzora nad domačim sistemom na povsem nov način, je bilo zelo pomembno, da je ta izkušnja po eni strani enostavna za uporabo in po drugi strani prepričljiva. Dejstvo, da je bila ravno ta interakcija med njihovim produktom in uporabnikom jedro celega proizvoda, je pomenilo, da je bila izbira razvojne platforme zelo pomembna.

Zato se je podjetje Convergent Living odločilo njihov uporabniški vmesnik produkta Companion razviti v Adobe Flash-u. Companion vsebuje Flash predvajalnik, namenjen za vgradnjo. Convergent Living ga je enostavno integriral z njihovo strojno opremo z uporabo programskega razvojnega okolja za predvajalnik Adobe Flash, namenjen za ustvarjanje

fleksibilne in dinamične platforme, na kateri so oblikovali učinkovite uporabniške izkušnje.

Hitrost razvoja je bila največja prednost, ki jo je prinesla uporaba Macromedia Flash. Glede na ocene podjetja Convergent Living so uspeli prihraniti čas enega leta, enega človeka. Poleg tega pa so glede na to, da je uporabniški vmesnik ločen glede na kodo, z lahkoto ločili oblikovanje in razvoj. Prav tako je ločen razvoj omogočal razvijalcem hitro izdelavo predstavitvenih prototipov za prodajne namene. Zahvaljujoč uporabi vektorske grafike pa se je isti uporabniški vmesnik lahko implementiran tako na 3,5 palčnem kot 10 palčnem zaslonu.

#### **Prednost za partnerje**

Sam proces preoblikovanja uporabniškega vmesnika za aplikacije partnerjev je bil zelo poenostavljen, obenem pa se je skrajšal čas izobraževanja na temo uporabe produkta

### **3.3 *Bogate internetne aplikacije [5]***

Tradicionalne aplikacije so bile zgrajene na osnovi odjemalec strežnik arhitekture z enostavnim odjemalcem v ospredju. V tem sistemu je vsa obdelava podatkov narejena na strežniku in naloga odjemalca je samo, da prikaže statično (v tem primeru HTML) vsebino. Največja slabost tega sistema je ta, da mora vsa interakcija z aplikacijo potekati preko strežnika. Tak sistem zahteva, da se podatki pošljejo na strežnik, strežnik da na njih odziv in stran se osveži glede na dobljene podatke s strežnika. Z uporabo tehnologije, orientirane na odjemalčevi strani, ki lahko izvrši ukaze na odjemalčevem računalniku, lahko bogate internet aplikacije zaobidejo to počasno in sinhronizacijsko zanko za mnoge uporabniške interakcije. Vse bogate internet aplikacije združuje ista lastnost, in sicer da vsebujejo napredno plast kode, pogosto imenovano odjemalčev pogon (angl. client engine) med uporabnikom in strežnikom. Ta odjemalčev pogon je običajno prenesen s spleta pred začetkom aplikacije. Odjemalčev pogon deluje kot dodatek k spletnemu brskalniku in običajno prevzame odgovornost za izris uporabniškega vmesnika in komunikacije s strežnikom. Omejitev pri izdelavi bogatih internetnih aplikacij pogosto predstavlja sistem,

ki teče na odjemalčevi strani. Na splošno pa je odjemalčev pogon izdelan in programiran z namenom, da izvaja aplikativne funkcije za katere njihov oblikovalec meni, da bodo pomagale k izboljšanju uporabniškega vmesnika ali pomagale pri odzivnosti pri določenih uporabniških interakcijah, v primerjavi s standardnim brskalniško orientiranim načinom implementacije. Tudi bogate internet aplikacije izvajajo dodatno asinhrono komunikacijo s strežniki.

Prednost bogatih internetnih aplikacij je predvsem v tem, da omogočajo dosti več svobode pri izdelavi uporabniškega vmesnika kot strani s standardnimi HTML spletnimi aplikacijami. Ta svoboda se izraža skozi različne funkcionalnosti, ki jih je možno implementirati v aplikacijo na odjemalčevi strani, npr.: primi in spusti, vertikalni in horizontalni drsnik za spremembo podatkov, izračuni izvršeni izključno na odjemalčevi strani, ki ne rabijo biti poslani nazaj na strežnik. Vse to prinaša veliko hitrejšo odzivnost uporabniškega vmesnika v primerjavi s standardnim spletnim komuniciranjem, s strežnikom za vsak izračun. Obenem to pomeni manjšo komunikacijo s spletnim strežnikom in njihovo razbremenitev ter možnost komuniciranja z istim spletnim strežnikom z več odjemalci hkrati. Ena izmed prednosti je tudi asinhrona komunikacija med odjemalcem in strežnikom. **To v praksi pomeni, da se podatki ne prenašajo samo takrat, ko uporabnik izvrši eno izmed akcij na strani uporabniškega vmesnika, npr. klik gumba ali hiperpovezave.** Ta možnost omogoča programerjem bogatih internetnih aplikacij prenos podatkov med odjemalcem in strežnikom, brez da bi uporabnik na njih čakal. Verjetno najbolj uporabljena aplikacija je inteligentno prednalaganje, kjer aplikacija predvidi prihodnje potrebe po določenih podatkih in jih prenese na odjemalca, še preden jih uporabnik zahteva ter s tem pospeši sledeče odzive aplikacije. Google Maps uporablja tehniko, da k uporabniku avtomatsko prenaša določene segmente okoliških zemljevidov, glede na pozicijo kje se uporabnik nahaja.

Čeprav je razvoj aplikacij, ki tečejo pod spletnim brskalnikom bolj omejujoč, težaven in kompleksen proces kot je npr. namizna aplikacija, pa se trud ponavadi izplača zaradi veliko prednosti, ki jih taka "online" aplikacija prinaša. Prva je ta, da instalacija programa ni potrebna in da se vsi popravki vršijo avtomatsko vedno znova, ko naložimo spletno stran, na kateri se nahaja ta aplikacija. Poleg tega lahko s to aplikacijo delajo uporabniki na dislociranih enotah po celem svetu, ne glede na to, kateri operacijski sistem uporabljajo.



Obenem pa so spletne aplikacije tudi bolj varne, saj se preko njih ne prenašajo virusi, kar ne moremo reči za izvršne datoteke, ki jih zaganjamo na namizju. S povečevanjem uporabe spleta je vedno več uporabnikov bolj naklonjenih spletnim aplikacijam, čeprav so včasih počasnejše od namiznih.

### **3.4      *Metode in tehnike***

#### **JavaScript**

JavaScript je prva glavna odjemalniško orientirana tehnologija, dostopna z možnostjo poganjati kodo, ki je bila naložena na večino spletnih brskalnikov. Čeprav je bila njena uporaba na začetku relativno omejena, pa je v kombinaciji s plastmi in ostalimi razvoji v DHTML-ju postopoma začela omogočati razvoj bogate internetne aplikacije. Ajax je nov skovan termin, ki se nanaša na kombinacijo tehnik in je bil najbolj opazno uporabljen s strani Googla v projektih kot so Gmail ali Google Maps. Vendar pa je izdelava večjih aplikacij v tem razvojnem okolju zelo težavna, saj je za dobro delovanje in kompatibilnost z različnimi spletnimi brskalniki potrebno poznati veliko različnih tehnologij in njihovih interakcij. Da bi se ta proces poenostavil, so se razvila določena Ajax razvojna okolja. Ena največjih težav vsestranskega JavaScript pristopa v bogatih internetnih aplikacijah je ta, da je JavaScript še vedno vezan na domnevno podporo multimedijskim tipom s strani spletnih brskalnikov. Video bo prikazan na različne načine v različnih spletnih brskalnikih z JavaScript pristopom, prav tako je nepredvidljiva podpora predvajanju zvoka, komunikaciji v realnem času, video pogovorom in prosojnostim. Zato vsesplošni JavaScript pristopi združujejo svojo bogatost predvsem na področju osveževanja teksta in slik.

#### **Java apleti**

Java apleti tečejo na standardnih HTML straneh in se običajno poženejo avtomatsko, ob zagonu spletne strani z relativno naprednim spletnim brskalnikom. Java apleti imajo dostop do zaslona (znotraj področja na HTML strani), zvočnikov, tipkovnice in miške kateregakoli računalnika, na katerem se je izvršila njihova koda. Java apleti predstavljajo napredno okolje v katerem je možno izdelati aplikacije, ki tečejo v realnem času.

### **Java aplikacije**

Bogate spletne aplikacije, ki temeljijo na Javi, je možno poganjati tako iz spletnega brskalnika kot tudi kot samostojne aplikacije s pomočjo Java Web Start. Aplikacije, ki temeljijo na Javi lahko popolnoma izkoristijo Java platformo in njeno podporo tako 2D, 3D grafiki kot tudi zmožnostim dela brez povezave, vendar za ceno zakasnjene zagona. Obstajajo številna okolja za izdelavo bogatih internetnih aplikacij v javi, vključno z XML okolji kot je XUL oziroma XUI ter Swix

### **Adobe Flash in AIR**

Adobe Flash je ena izmed tehnologij, ki omogoča izdelavo bogatih internetnih strani. Ta tehnologija je kompatibilna z večino operacijskih sistemov in zelo uporabna in močna na področju izdelave uporabniških vmesnikov. Adobe Flex omogoča izdelavo Flash uporabniških vmesnikov s pomočjo prevajanja MXML, ki predstavlja uporabniški opisni jezik, kateri temelji na XML-ju. Adobe trenutno dela na produktu, ki bo predstavljal zelo močno platformo, namenjeno izdelavi bogatih spletnih aplikacij imenovano AIR. Le-ta bo združeval tako Flash kot tudi PDF.

### **Windows Presentation Foundation**

S platformo .NET 3.0 Framework, je Microsoft predstavil Windows Presentation Foundation (WPF), ki omogoča izdelavo aplikacije z določenimi podobnostmi bogatim spletnim aplikacijam v kombinaciji z XAML in programskimi jeziki, kot so C# in Visual Basic. Microsoft Silverlight (kodno poimenovan Windows Presentation Foundation/Everywhere ali WPF/E) je okolje, namenjeno bogatim aplikacijam, temelječih na spletnih brskalnikih. Omogoča podskupino animacijskih, grafičnih - vektorskih in video zmožnosti Windows Presentation Foundation. Okolje je na voljo za Microsoft Windows in Mac OS X. V razvojni fazi pa je tudi podpora za Linux s kodnim imenom Moonlight.

**ActiveX Controle**

Vgrajevanje ActiveX kontrol v HTML je zelo uporaben način za izdelavo bogatih internetnih aplikacij, vendar pa delujejo zagotovo pravilno samo v Internet Explorerju. Ker lahko ActiveX kontrole prestopijo omejitve angl. sandbox modela, predstavljajo grožnjo za okužbo z virusi in ostalo zlobno kodo. V tem trenutku je Adobe Flash predvajalnik za Internet Explorer implementiran kot ActiveX kontrola za Microsoftova okolja. ActiveX je dobra izbira za izdelavo aplikacij samo, če se Internet Explorer uporablja kot primarni internetni brskalnik (s tem si zožimo področje uporabnikov).



---

## POGLAVJE 4

---

# 4 POSTOPEK IZDELAVE PRIKAZA ROBOTIMING

### *4.1 Izdelava prikaza na samem tekmovanju*

Vizualni prikaz na robotskem tekmovanju je predstavljal poseben izziv, saj je bilo potrebno združiti že določeno predhodno znanje s področja oblikovanja, uporabniških vmesnikov, Flasha in izdelave spletnih strani ter ga nadgraditi z znanjem o najnovejših tehnologijah. Zato in zaradi že predhodnega poznavanja tehnologij, sem se odločil uporabiti orodja družine Adobe Flash, Dreamweaver ter Photoshop. Najprej je bilo potrebno ugotoviti katere podatke želimo prikazovati na velikem zaslonu ter da bodo vidni celotnemu občinstvu. Po eni strani je potrebno prikazati veliko informacij, po drugi strani pa mora vmesnik delovati preprosto in čisto. Še več, informacije na velikem ekranu morajo povezati tekmovalca z njegovim robotom ter vzpostaviti stik med publiko in tekmovalcem samim. Prav tako je potrebno najboljše tekmovalce nagraditi in jih vizualno posebej izpostaviti, saj so le-ti zaradi nadpovprečnih rezultatov bolj zanimivi za občinstvo. Namen spremljanja tekmovanja na velikem zaslonu je poleg vsega tudi skozi informiranost narediti tekmovanje bolj napeto in zanimivo ter atraktivno za povsem laične ljudi na področju mobilnih robotov ter naključne obiskovalce. Da bi to dosegli, smo vnesli elemente prikaza merjenja časa, ki jih uporabljajo pri televizijskih prenosih smučarskih tekmovanj in so javnosti širše poznani. Sam prikaz je moral biti daljinsko oz. ročno voden, saj zaradi same narave tekmovanja ni mogoče točno časovno določiti vseh dogodkov.

Zaradi možnih človeških napak pri izvedbi na samem tekmovanju pa morajo poleg avtomatizma obstajati tudi ročne komande ,npr. preskoči tekmovalca, ročno vnosi kvadrant v primeru da tekmovalčev robot ne pride do cilja v določenem času in druge. Odločil sem se, da bom sam način prikaza razdelil na tri tipe, in sicer uvodni oziroma pozdravni prikaz, prikaz tekmovanja in prikaz končnih rezultatov. Posamezni prikaz je sestavljen iz štirih okvirjev (angl. framov). Za tak način prikazovanja sem se odločil zaradi zmanjšanja prenosa podatkov pri sekundnem osveževanju, ki je potrebno za prikazovanje rezultatov v realnem času. Okvirji nam omogočajo večjo kontrolo nad osveževanjem določenega dela prikaza in lahko na tak način osvežujemo točno določen okvir. Tak način smo uporabili pri prikazu rezultatov, štartni listi ter prikazu končnih razvrstitev. Dejansko smo bili primorani v rešitev z okvirji, saj je sicer pri čeprav par sekundnem osveževanju celoten prikaz deloval zelo utripajoče in neprofesionalno. Okvirje in celotno predlogo strani sem izdelal s pomočjo orodja za izdelavo spletnih strani Dreamweaver (dodatek B).

### Uvodni prikaz

Najprej se srečamo s sliko uvodnega prikaza (sl. 4.5), ki je v osnovi razdeljena na prostor za glavo, prostor za pozdravno sporočilo z urnikom, oglaševalski prostor in prostor za merjenje časa.



Slika 4.5: Prikaz na tekmovanju – uvodna stran

## Glava

Je najpomembnejši in najbolj udarni del prikaza in lahko gledalca pritegne ali odvrne od pregleda ostale vsebine. Pri oblikovanju glave sem upošteval, da je večinska populacija moška, da je povprečna starost tekmovalcev in gledalcev okrog 19 let in da prihajajo iz tehničnega profila. Obenem pa nisem smel zanemariti osnovnega motiva tekmovanja – to so roboti in vsa pripadajoča elektronika. Upošteval je vse te parametre sem se odločil, da bom za osnovno barvo uporabil modro-belo kombinacijo. Le-ta izžareva tehnično dovršenost in prijetnost na oko, še posebej za moški spol. Za samo ozadje v glavi sem uporabil motiv robota, in sicer sem slikal detajl vezja robota, ki sva ga skupaj s sotekmovalcem sestavila. Ravno ta robot je bil nato tisto leto, ko smo predstavili in preizkusili projekt Robotiming v praksi, obenem tudi zmagovalni robot na odprtem državnem študentskem in dijaškem tekmovanju z mobilnimi roboti RoboT 2004 v študentski kategoriji. Pri tipu pisave sem želel poudariti predvsem tehnično dovršenost in ustvarjalnost ter napredek, ki ga tekmovanje RoboT želi vzpodbuditi med mladimi. Uporabil sem pisavo imenovano MicrogrammaDEEMedExt, ki sem jo tudi nagnil in s tem uporabil retro efekt tehničnega sloga. V ozadje pa sem z efektom »zameglitev« (angl. Blur) skoraj neopazno zakril še nekaj elementov s področja računalništva in merjenja časa. Prav tako pa sem že v sami glavi zaradi boljše prepoznavnosti jasno nakazal ime tima oz. merilnega sistema. Tako je bila v osnovi glava (sl 4.6), ki sem jo kasneje uporabil tudi za osnovo pri oblikovanju spletne strani, končana.



Slika 4.6: Končni izdelek oblikovanja glave za namen prikaza na tekmovanju in spletu

## Prostor za pozdravno sporočilo z urnikom

V osnovi sem se odločil da bodo vsi prostori obrobjeni s kvadratom, ki ima zaobljene robove. Da bi vse skupaj delovalo še malo bolj profesionalno in moderno, sem v Photoshopu s pomočjo obrobe in 3D efekta (angl. Bevel and Emboss) ustvaril 3D okvirje

oziroma tako imenovane oblačke. Na pozdravno stran sem vklopil osnovno glavo organizatorja, pozdravno sporočilo in urnik tekmovanja (sl 4.7).



Slika 4.7: Pozdravno sporočilo z urnikom

### Oglaševalski prostor

Njegov namen je bil izpostaviti sponzorje, ki so ali finančno ali moralno podprli tekmovanje. Da bi ta prostor deloval čim bolj pregledno, sem nekako smiselno in pregledno razvrstil vse logotipe, ki sem jih poprej ustrezno obdelal v Photoshopu in odstranil morebitno barvno ozadje. Obenem pa sem vse te logotipe prekril z belim prosojnim kvadratom, ki rahlo ublaži preveč sijoče belo ozadje in daje sami strani enoten izgled (sl 4.8).



Slika 4.8: Oglaševalski prostor



**Prostor za prikaz merjenja časa:**

Le-ta je bil na zgornji desni polovici in je preprosto meril čas. Na uvodni strani ima funkcijo merjenja testnih voženj mobilnih robotov, ki so sestavni del uvodnega tekmovanja z namenom, da tekmovalci prvič preizkusijo svojega robota v tekmovalnem labirintu. Z njegovo končno postavitvijo se tekmovalci seznanijo šele na dan tekmovanja. Prikaz merjenja časa je sestavljen iz 3D okvirja in Flash komponente, ki deluje v povezavi z merilnim sistemom (sl. 4.9).



Slika 4.9: Prikaz merjenja časa

**Prostor za prikaz tekmovanja**

Predstavlja najbolj pomemben del, ki je poleg prostora za glavo na vrhu prikaza merjenja časa, sestavljen še iz prostora za prikaz rezultatov in uvrstitev v določeni kategoriji ter prostora za prikaz dinamične štartne liste.

**Prostor za prikaz rezultatov in končna razvrstitev**

Največje okno je rezervirano za prikaz končne razvrstitve tekmovalcev v trenutno aktivni kategoriji (študentska ali dijaška). Vsak tekmovalec je glede na rezultat razvrščen v tabelo od najboljšega (vodečega) navzdol. Naenkrat je lahko prikazanih okoli 15 tekmovalcev. Pri tej številki smo predvsem upoštevali kompromis med čim večjim številom prikazanih in čitljivostjo. Predpostavili smo, da mora biti najmanjši podatek na končni projekciji tako velik, da ga bo mogoče brez težav prebrati z razdalje dvajsetih metrov. Ravno zato smo za izpis uporabili pisavo imenovano Verdana, ki ne vsebuje angl. serifov in je kot taka bolj čitljiva. Sam prikaz je vsebinsko sestavljen iz naslova oziroma kategorije, ki je trenutno

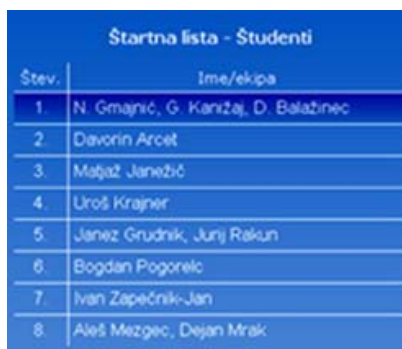
aktivna ter posameznih tabel z naslednjimi podatki: uvrstitev, ime oz. ekipa, vmesni čas, ki ga je tekmovalec dosegel ter končni čas. Kot uvrstitev posameznega tekmovalca se vedno upošteva najboljši končni čas. V kolikor le-ta ni dosegel končnega časa oz. ni prišel do cilja pa se pod končni čas izpiše ustrezen kvadrant. Prav tako se ob končani vožnji tekmovalca in uvrstitvi na listo petnajstih najboljših tekmovalcev, ustrezno poudari vrstica oz. tekmovalec, ki se je pravkar uvrstil med rezultate. Obenem pa je prvo uvrščeni vedno napisan s še posebej odebeljeno pisavo, s čemer tudi naključni obiskovalci veliko hitreje opazijo najbolj uspešnega tekmovalca (sl. 4.10). Oblikovno predlogo za tabele sem ustvaril z orodjem Dreamweaver. Le-to uporablja skriptni jezik php, ki stran prikazuje.

Mesto	Ime/ekipa	Vmesni čas	Končni čas/kvadrant
<b>1.</b>	<b>Uroš Derstvenšek, Andrej Koštomaj</b>	<b>00:01.32</b>	<b>00:05.46</b>
2.	Jernej Škvarč	00:00.96	+00:01.20
3.	Mažar Dolinar, Marko Keber	00:03.06	+00:02.38

Slika 4.10: Prikaz rezultatov v študentski kategoriji

### **Prostor za prikaz dinamične štartne liste**

Glede na to, da sem bil tudi sam tekmovalec vem, da se zadnji dan na tekmovanju velikokrat pojavijo težave, ki se jih odpravlja še zadnje trenutke pred nastopom. Včasih, ko je bila štartna lista samo na papirju komentatorja, je bilo praktično zelo težko oceniti koliko časa še imaš, da se pripraviš. Zato smo se odločili, da prikažemo podatek startne liste, ki je informativne narave predvsem za tekmovalce delno pa tudi za gledalce, da lahko spremljajo in ne zamudijo vožnje svojega favorita med roboti. Prostor za prikaz startne liste je opremljen z ustreznim naslovom in je sestavljen iz tabele z dvema stolpcema. V enem so zaporedne številke, ki predstavljajo koliko tekmovalcev je na vrsti še pred določenim tekmovalcem. V drugem stolpcu pa je ime tekmovalca oziroma ekipe. Tekmovalec, ki bo naslednji na vrsti, ima številko ena in njegova vrstica je posebej označena s temno modro barvo (sl. 4.11).



The image shows a screenshot of a digital interface titled "Startna lista - Studenti". It contains a table with two columns: "Stev." (Number) and "Ime/ekipa" (Name/team). The table lists eight entries, with the first entry highlighted in a darker blue.

Stev.	Ime/ekipa
1	N. Gmajnić, G. Kanizaj, D. Balazinec
2	Davorin Arcet
3	Matjaž Janežič
4	Uroš Krajner
5	Janez Grudnik, Jurij Rakun
6	Bogdan Pogorelec
7	Ivan Zapečnik-Jan
8	Aleš Mezgec, Dejan Mrak

Slika 4.11: Prikaz štartne liste

### **Prikaz končnih rezultatov**

Le-ti se prikazujejo ob koncu voženj posamezne kategorije. Po tekmovanju pa se izmenično prikazujejo rezultati dijaške in študentske kategorije kot glavni dve kategoriji tekmovanja. Prikazujejo vmesne oziroma po drugi vožnji tudi končne rezultate in pripadajoče razvrstitve. Sestavljen je iz glave in prostora za prikaz tabele rezultatov.

### **Prostor za prikaz tabele rezultatov**

Tabela rezultatov je razširjenja čez cel ekran, saj smo želeli v eni vrstici prikazati vse pomembne dosežke in podrobnosti o tekmovalcu. Le-ti so razporejeni od najboljšega navzdol, upošteva najboljši čas obeh voženj. Vodeči tekmovalec ima izpisan čas, medtem ko imajo vsi tekmovalci pod njim namesto časa izpisan zaostanek za njim. Poleg tega so prikazani vsi časi posamezne vožnje in podatki o šoli/univerzi tekmovalca ter letniku, ki ga obiskuje (sl. 4.12).

Mesto	Ime/ekipa	Najboljši dosežek	1.votnja	2.votnja	3.votnja	Sota	Letnik
1	Rok Benedičič-ROBO-GT	00:20.80	00:22.45	00:20.80		TSC Kranj	3.
2	Primož Bagček-M3	+00:00.56	00:22.22	00:21.35		TSC Kranj	2.
3	Aleš Kotnik-Tornado	+00:03.24	00:24.04	D8		TSC Kranj	2.
4	Jernej Valenti, Miša Valenti-Cicibot 2	+00:04.82	00:27.66	00:25.62		FERI Mb	3. I, UNI E
5	Cvetko Pirš, Nino Ferle-I Robot	+00:09.44	00:30.24	E2		SERS Maribor	4. I
6	Simon Tržan	+00:09.71	00:30.51	00:33.37		Šolski center Velenje	4.
7	Damijan Čuš-Čuša	+00:10.03	00:31.41	00:30.83		Šolski center Ptuj	2. ER
8	Tadej Ferle-FATC	+00:10.68	00:31.63	00:31.48		SERS Maribor	1.
9	Rok Furman-Muzikant	+00:11.62	00:32.92	00:32.42		Šolski center Ptuj	2. PTI
10	Aleksaj Cegnar, Denis Kozar, Sašo Tomazič-ROBO n.1	+00:12.29	00:33.09	00:33.30		SERS Maribor	1., 1., 1.
11	Rok Pučko-Stanič	+00:13.32	00:35.88	00:34.12		Šolski center Ptuj	2. EE
12	Jernej Mioč Krmelj-Speedy	+00:14.43	00:35.23	E6		Šolski center Velenje	4.
13	N Ferle, S Zemljč, S Markovič-ROBO n.2	+00:18.83	00:39.63	C2		SERS Maribor	1., 1., 1.
14	Mina Grim-Same	+00:19.60	00:40.30	F2		PTEKŠ Celje	4. EE
15	Janez Rudolf-Hibi	+00:19.60	00:42.42	00:40.30		Šolski center Ptuj	1. EE
16	Aleksander Svencšek-Merc-Mad Max	+00:19.73	00:42.74	00:40.53		Šolski center Ptuj	2. ER
17	Aleš Bergauer	+00:22.79	00:43.59	A5		FERI Mb	2. I, UNI E
18	Miha Poljanec-Rail	+00:22.87	00:43.67	00:45.41		Šolski center Ptuj	1. EE

Slika 4.12: Tabela rezultatov

### Izdelava prikaza na spletu

Tekmovanje z mobilnimi roboti v študentski in dijaški kategoriji pritegne veliko število obiskovalcev in radovednežev, saj je samo po sebi zelo atraktivno in zanimivo za širšo množico. Še več pa je takih, ki bi radi spodbujali svojega favorita in bili informirani o dogajanju na tekmovanju, a zaradi takšnih in drugačnih ovir na tekmovanje ne morejo priti. Da bi rešili ta problem in tudi, da bi se tekmovanje samo po sebi lahko bolje promoviralo, smo izdelali spletno stran, ki je omogočala spremljanje tekmovanja v živo. Le-ta je v osnovi sestavljena iz uvodne strani, spremljanja tekmovanja v živo, rezultatov in informativne strani o Robotiming ekipi (sl. 4.13). Stran je bila zgrajena s pomočjo orodja Adobe Flash MX 2004 in Adobe Dreamweaverja MX 2004.



Slika 4.13: Uvodna stran je zgrajena je iz menija, glave in vsebinskega dela.

Meni komponenta je narejena v celoti v Flashu. Sestavljena je iz grafike, ki je bila predhodno oblikovana v Photoshopu in ustreza oblikovalskim smernicam, ki so bile prvotno zasnovane že za sam prikaz na tekmovanju ter območij, ki vsebujejo tekst s hiperpovezavo na vsebinsko stran oziroma so občutljiva na kazalec miške in se ob tej akciji pokaže še dodatna možnost izbire. Tak primer je bil gumb startna lista, kjer je lahko obiskovalec nadalje izbiral med študentsko ali dijaško kategorijo. Tak način izbire zmanjša število osnovnih gumbov in naredi stran bolj pregledno. Izdelava gumbov oz. menijev v Flashu je razdeljena v dve fazi, in sicer priprava grafike in njena umestitev ter pisanje kode in animiranje. V Photoshopu pripravljene grafične elemente sem shranil v grafični format png, ki za razliko od jpg-ja podpira transparencio oziroma brezbarvno ozadje. Te elemente potem s pomočjo funkcije »import to library« prenesemo v Flash knjižnico, katere vsebina je prikazana v desnem oknu razvojnega okolja Flash. Če je ne vidimo, jo lahko vklopimo s pomočjo izbire menija "Windows" (angl.) in izberemo možnost "Library" (angl.). Ko imamo enkrat element v knjižnici, ga lahko enostavno prestavimo na delovni prostor oz. angl. "Stage". V našem primeru je bilo potrebno najprej prenesti osnovni okvir. Le-tega je

bilo potrebno razdeliti na šest enakih razdelkov. Da bi se ti razdelki vizualno lažje ločili, sem jih ločil s pokončno črtico. Narisal sem jo s pomočjo »Line Tool" (angl.). Da bi imel večjo kontrolo nad elementom in da bi se le-ta tudi pojavil v knjižnici elementov, je potrebno označeno grafiko klikniti z desnim gumbom na miški in izbrati "Convert to symbol" (angl.) ter ustrezno izbrati tip simbola oz gradnika. Izbiramo lahko med grafičnim elementom, gumbom in animacijo oz. angl. "Movie Clip" (angl.). Ker ločevalna črtica predstavlja grafični element, ga ustrezno izberemo ter jo tudi poimenujemo. Pod tem imenom jo bomo kasneje lahko hitro našli v knjižnici elementov in ponovno uporabili ali urejali. Nato sem grafični element črtice prestavil iz knjižnice in ga petkrat skopiral ter ustrezno umestil, tako da sem dobil šest razdelkov. Da bi določil ustrezno belo barvo, črtici več ne rabim urejati vsakega elementa posebej, ampak to storim samo z dvoklikom na element črtice v knjižnici in s pomočjo določanja lastnosti grafičnemu elementu, ki se nahaja v oknu na dnu razvojnega okolja Flash pod angleškim imenom "Properties", spremenim barvo črte na belo. S tem se vseh pet črtic, ki sem jih predhodno ustvaril, avtomatsko ustrezno spremeni. Nato je bilo potrebno vpisati imena osnovnih gumbov menija. To sem storil tako, da sem iz orodne vrstice na levi strani razvojnega okolja izbral ikono "Text Tool" (angl.) ter se pomaknil v prvi razdelek menija in vpisal ustrezen tekst. Prvi gumb ima funkcijo, da uporabnika povrne na prvo stran in sem ga zato poimenoval po robotskem tekmovanju RoboT + letnica tekmovanja. Da bi tekst, ki sem ga napisal, imel lastnost gumba, sem to storil z ukazom "Convert to symbol" in izbiro »Button« (angl.) ter ga ustrezno poimenoval. Pri urejanju posameznega gumba imamo štiri stanja (angl. "Up", "Over", "Down" in "Hit«) V našem primeru za gumb "Robot 2008" so stanja "Up", "Over", "Down" enaka. Pri stanju Hit pa sem s pomočjo orodja "Rectangle Tool" (angl.) narisal kvadrateg, ki označuje aktivno področje gumba. Nato je bilo potrebno določiti še akcijo ob pritisku na sam gumb. To sem ustvaril s pomočjo ukaza »on (Release)« kjer sem vpisal url naslov, kjer se nahaja osnovna stran tekmovanja RoboT. Da bi se znebili vsakoletnega popravljanja menija oz. hiperpovezave ob zamenjavi internetnega naslova strežnika, je bilo potrebno Flash izdelati tako, da je ob zagonu prebral IP naslov strežnika iz tekstovne datoteke oz. spremenljivke. Tako je bilo potrebno ustvariti povsem nov "Layer" (angl.), ki je imel samo en okvir, v katerega sem vpisal kodo s pomočjo katere je Flash ob zagonu iz datoteke server.txt prebral vse spremenljivke. V našem primeru je bila to spremenljivka naslov\_serverja. Le-to pa je možno enostavno urejati preko vsakega tekstovnega

urejevalnika. Tako je bila končna hiperpovezava, ki sem jo vpisal za gumb RobotT 2008 sledeča:

```
getURL("http://"+_level0.load_lv.naslov_serverja+"/tekma/www/index.htm","_top");
```

Ker se štartna lista, ki je predstavlja drugi gumb v menijski vrstici deli na študentsko in dijaško, je bilo potrebno narediti podgumba študenti in dijaki, ki pa se pokažeta šele, ko se z miško premaknemo čez gumb startna lista (sl. 4.14).



Slika 4.14: Izdelava podmenija

Da bi to dosegli, je bilo najprej potrebno ustvariti gumba študenti in dijaki. Pri obeh sem ob stanju "Over" uporabil moder zaobljen pravokotnik ter določil ustrezno veliko aktivno polje pod stanjem "Hit". Vsakemu gumbu sem s pomočjo kode vpisal tudi ustrezno hiperpovezavo, ki se naj vzpostavi ob pritisku na gumb. Nato sem ustvaril posebno animacijo "Movie Clip" (angl.), ki je imela dva okvirja. Prvega sem pustil praznega, na drugega pa sem ustrezno umestil gumba študenti in dijaki.

Da bi vse skupaj delovalo kot sem si zamislil, je bilo potrebno le še povezati gumb startna lista z animacijo, v kateri sta se nahajala oba gumba. To sem storil tako, da sem s pomočjo kode vpisal v gumb Startna lista, da se ob prehodu miške čez gumb Startna lista animacija z gumboma premakne na okvir 2 in s tem postaneta gumba tudi dejansko vidna. To sem dosegel z ukazom:

```
on (rollOver) {
    _root.startna_list.gotoAndStop(2);
}
```

Da bi se ob odmiku z gumba startna lista podmeni oz. izbira kategorije startne liste skrila, sem to dosegel s sledečim ukazom:

```
on(rollOut) {
```

```

_root.startna_list.gotoAndStop("1");

}

```

Da pa gumb Startna lista ne bi ostal neaktiven, sem mu vseeno dodelil privzeto hiperpovezavo, ki je kazala na startno listo dijakov, kjer je bilo več tekmovalcev:

```

on (release) {

getURL("http://" + _level0.load_lv.naslov_serverja + "/tekma/www/startna_dij.php",
_top");

}

```

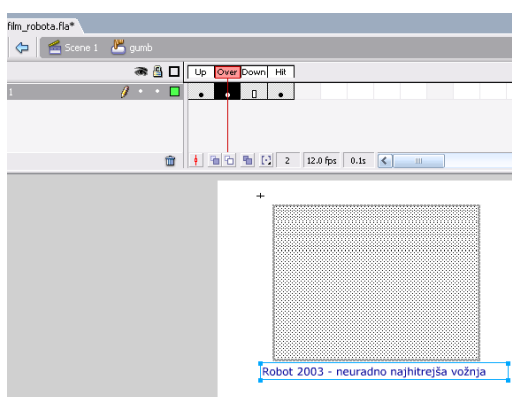
Takšen način podmenija sem uporabil tudi pri gumbu Rezultati. Ostali gumbi pa so bili narejeni po podobnem postopku kot prvi gumb, ki sem ga opisal. Tako je bil osnovni meni za spletno stran namenjeno prikazu tekmovanja RoboT na internetu pripravljen (sl. 4.15).



Slika 4.15: Izdelava menijev za spletno stran v Adobe Flashu in končna postavitev le-teh v Adobe Dreamweaverju

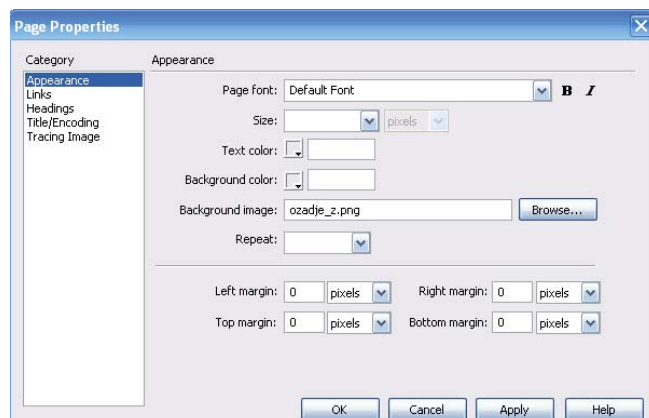


Za glavo sem uporabil že predhodno izdelano glavo za prikaz na tekmovanju. Vsebinski del pa sem sestavil iz dveh delov. Levi del vsebuje tekst z vsemi informacijami v zvezi s tekmovanjem s hiperpovezavami, desni pa je sestavljen iz slike, ki se ob dotiku z miškinim kazalcem spremeni v predvajanje majhnega filmčka oziroma odlomka iz preteklega tekmovanja. Namen tega je, da obiskovalec v kratkem času dobi približno sliko o robotskem tekmovanju. Filmček je stisnjen s pomočjo Flash kodeka imenovanega On2 VP6. Tak film enostavno uvozimo v Flash preko ukaza "File>Import>Video" ter ga iz knjižnice nato vključimo v Flash razvojno okolje. Naredil sem nov gumb, v katerega sem vključil predvajanje filma ob njegovem stanju Over (angl.). V osnovnem stanju gumba - Up (angl.) pa sem vklopil navadno sliko (sl. 4.16).



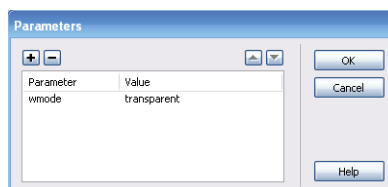
Slika 4.16: Izdelava gumba pri katerem se ob prehodu miške preko statične slike sproži video posnetek

S pomočjo Dreamweaverja in Flash animacij sem nato ustvaril prvo in osnovno vstopno stran, imenovano index.html, ki predstavlja vstopno stran za spletne obiskovalce. Najprej sem moral določiti ozadje strani. Le-to je sestavljeno iz prehoda iz temne v svetlo modro. Da bi to dosegel, sem v s pomočjo Photoshopa iz osnovnega dizajna izrezal eno piko širok in 1024 pik dolg trak, ki sem ga shranil v png format. Le-tega sem potem vstavil v ozadje strani, kjer se vzorec ponavlja iz leve proti desni (sl. 4.17) S ponavljanjem določenega vzorca se prihrani čas, ki bi sicer bil potreben, da bi se preneslo ozadje v enem kosu in kar bi močno povečalo datoteko za ozadje ter podaljšalo čas nalaganja strani. Tudi odmike od robov levo, desno, spodaj, zgoraj sem nastavil na 0, tako da je bila Flash animacija, ki je vsebovala meni, povsem na vrhu spletne strani.



Slika 4.17: Nastavitev odmikov od roba in slike za ozadje spletne strani

Da bi lahko vstavil vse tri Flash animacije, ki so glavni gradniki spletne strani, sem ustvaril tabelo široko 800 in visoko 600 pik. Le-ta je imela tri vrstice, kamor sem potem vstavil glavni meni, glavo spletne strani in telo oziroma vsebinski del. Ker imajo Flash animacije zaobljene robove je bila želja, da je ozadje same animacije brezbarvno. To je možno doseči pri lastnostih posameznega Flasha, ki se nahajajo v spodnjem delu razvojnega okolja Dreamweaver, kjer kliknemo na »Parameters« in vstavimo pod ime parametra »wmode« in kot njegovo vrednost »transparent« (sl. 4.18).



Slika 4.18: Vnos dodatnega parametra, ki naredi ozadje Flash animacije transparentno - brezbarvno

Za velikost spletne strani 800 x 600 pik sem se odločil zaradi dejstva, da so nekateri uporabniki še vedno uporabljali takšno velikost oziroma ločljivost na svojem zaslonu in je veljala za standardno pri izdelavi spletnih strani.

Stran, ki je dostopna ponavadi že nekaj dni pred tekmovanjem je startna lista (sl. 4.19). Le-ta se prikazuje s pomočjo skriptnega jezika php. Podatki pa so pridobljeni iz podatkovne baze, kamor se predhodno vnesejo informacije o tekmovalcih, njihovih mentorjih in robotih. Za vsakega tekmovalca je objavljeno kateri po vrsti bo startal v svoji kategoriji, njegovo ime/ime ekipe, šola s katere prihaja, ime mentorja ter letnik šolanja. Na eni strani

so izpisani vsi tekmovalci. Sam izpis in njegov izgled sta podobna kot pri prikazu na samem tekmovanju. To je bil tudi namen, saj je potrebno neke začete oblikovne smernice nadaljevati oziroma ostati konsistenten. Hkrati pa podoben dizajn za gledalce oz. tekmovalce pomeni enostavni prehod pregledovanja podatkov iz spleta na samo tekmovanje in obratno.



Mesto	Ime/ekipa	Šola	Mentor	Letnik
1.	Nejc Lokar-Mercedes	TŠC Nova Gorica		3.
2.	Klemen Matko	Šolski center Velenje		2.
3.	G. Lešnik, A. Pintarič, U. Stojko-ROBO n.4	SERŠ Maribor		1., 1., 1.
4.	Mitja Grajfoner, Dejan Čižič-ROBO n.5	SERŠ Maribor		1., 1.
5.	A. Plohl, I. Fric, J. Mušič-ROBO n.6	SERŠ Maribor		1., 1., 1.
6.	Branimir Tadić-NIGHT2	Željeznička tehniška škola Zagreb		2.
7.	Sebastjan Bodlaj, Luka Pivk-Floki Type-S	SŠER Ljubljana		3. ER
8.	Silvo Brunčič-Kankole	Šolski center Ptuj		1. ER
9.	Rok Jerenko-Butkač	Šolski center Ptuj		3. EE

Slika 4.19: Spletna stran in prikaz štartne liste v dijaški kategoriji

Na dan tekmovanja lahko spletni obiskovalci pod povezavo »Tekma« spremljajo robotsko tekmovanje v živo (sl. 4.20). Stran je nastavljena tako, da se rezultati iz podatkovne baze osvežujejo na vsakih petnajst sekund. Prikazanih je deset najboljših v kategoriji, razvrščenih od najboljšega navzdol. Pri vsakem tekmovalcu se poleg imena tekmovalca/roboata izpiše tudi vmesni čas ter končni čas oziroma kvadrant. Trenutni rezultati se prikazujejo za dijaško in študentsko ekipo ter se spreminjajo tako, kot se to spreminja na samem tekmovanju.

Da bi lahko osveževanje izvršili posebej, je bilo potrebno spletno stran ločiti na dva okvirja. Eden je bil namenjen konstantnemu osveževanju, drugi pa se ni osveževal. To je bilo nujno potrebno, saj smo želeli, da se stran uporabniku sama osvežuje, obenem pa na isti strani prikaže video, ki pa se mora naložiti samo enkrat. V drugem okvirju se tako nahaja pretočni video s tekmovanja, za katerega je spletni obiskovalec moral namestiti

Windows Media Player oz. mplayer, v kolikor ga njegov spletni brskalnik še ni imel nameščenega.



**RoboT 2006**  
powered by RobotIMING™

**Rezultati - Dijaški**  
Prikazanih je 10 trenutno najboljših v posamezni kategoriji. Stran se osvežuje na 20 sekund.

Mesto	Ime/ekipa	Vmesni čas	Končni čas/Avadranit
1.	Rok Benedičič-ROBO-GT	00:09:46	00:20:80
2.	Primož Bajželj-M2	00:09:21	+00:00:55
3.	Aleš Kotnik-Tomago	00:09:13	+00:03:24
4.	Jernej Valenti, Miša Valenti-Cicibot 2	00:09:16	+00:04:82
5.	Cvetko Priš, Nino Ferle-I Robot	00:12:13	+00:09:44
6.	Simon Tržan	00:12:28	+00:09:71
7.	Damjan Čuš-Čuča	00:12:05	+00:10:03
8.	Tadej Ferš-FATC	00:12:37	+00:10:88
9.	Roki Furman-Muzikant	00:12:67	+00:11:62
10.	Aleksaj Cagnar, Denis Kozar, Sašo Tomazič-ROGO n.1	00:13:41	+00:12:29

**RoboT v živo**

Uporabniki Mozilla z desnim sponkjem kliknite na to povezavo in skrajšano datoteko odprete v Windows Media Playerju ali v programu za preigranje te datoteke.

Slika 4.20: Prikaz rezultatov in pretočnega video v živo s tekmovanja

Prikaz vseh trenutnih in ob koncu tekmovanja tudi končnih rezultatov je dostopen pod povezavo Rezultati (sl. 4.21). Tekmovalci so razvrščeni od najboljšega navzdol. Poleg imena tekmovalca in robota pa je zraven izpisan še njegov najboljši dosežen čas, časi iz posameznih voženj, šola in letnik.



RoboT 2006 | Startna lista | Tekma | Rezultati | Legošum | RobotIMING™

**Rezultati:**  
» Dijaška kategorija «

Mesto	Ime/ekipa	Najboljši dosežek	1 vožnja	2 vožnja	3 vožnja	Šola	Letnik
1	Rok Benedičič-ROBO-GT	00:20:80	00:22:45	00:20:80		TŠC Kranj	3.
2	Primož Bajželj-M2	+00:00:55	00:22:22	00:21:35		TŠC Kranj	2.
3	Aleš Kotnik-Tomago	+00:03:24	00:24:04	D8		TŠC Kranj	2.
4	Jernej Valenti, Miša Valenti-Cicibot 2	+00:04:82	00:27:66	00:25:62		FERI Mb	3. L. UNI E
5.	Cvetko Priš, Nino Ferle-I Robot	+00:09:44	00:30:24	E2		SERŠ Manbor	4., 1.
6.	Simon Tržan	+00:09:71	00:30:51	00:33:37		Šolski center Velenje	4.
7.	Damjan Čuš-Čuča	+00:10:03	00:31:41	00:30:83		Šolski center Ptuj	2. ER
8.	Tadej Ferš-FATC	+00:10:88	00:31:63	00:31:46		SERŠ Manbor	1.
9.	Roki Furman-Muzikant	+00:11:62	00:32:92	00:32:42		Šolski center Ptuj	2. PTI

Slika 4.21: Prikaz vseh trenutnih/končnih rezultatov vseh tekmovalcev

## 4.2 *Prikaz merjenja časa in rezultatov v Flashu*

Gre za eno izmed zelo pomembnih komponent sistema Robotiming, ki skrbi za ažurnost posredovanja informacij in dogajanja na robotskem tekmovanju. Njegova naloga je v realnem času na slikovit način širši publiki prikazati merilni proces, primerjalne rezultate in uvrstitve. Osnovne zahteve modula za prikaz so bile:

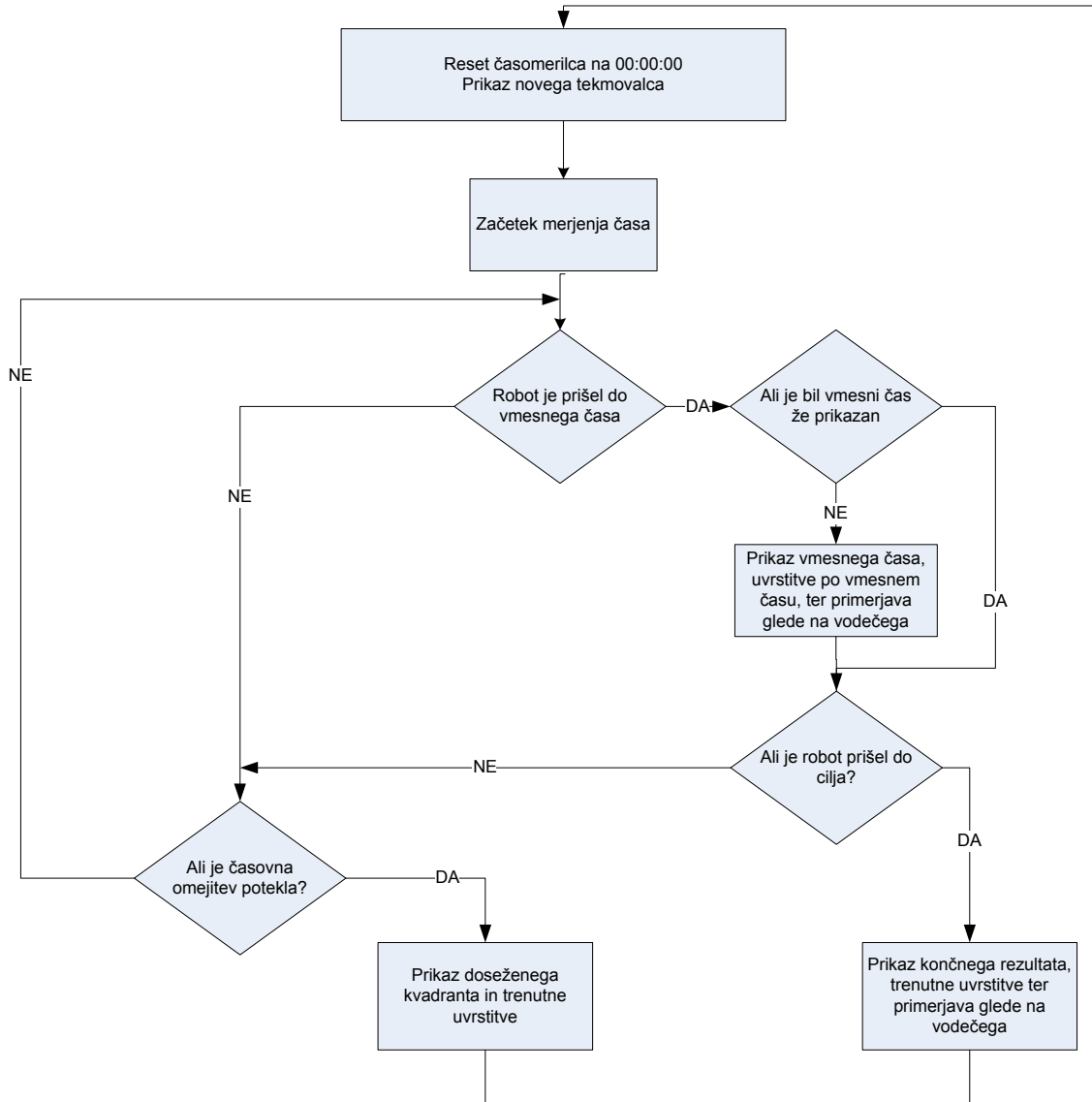
Vsebine, ki jih mora prikazati:

- animacija merjenja časa do stotinke natančno,
- ime ekipe, ki trenutno vozi oziroma je na vrsti,
- izmerjen vmesni čas ter razlika glede na najboljšega - do stotinke natančno,
- uvrstitev glede na vmesni čas v primerjavi z vsemi ostalimi tekmovalci,
- končni čas ter razlika glede na najboljšega - do stotinke natančno,
- v kolikor tekmovalec ne pride do cilja, se mora prikazati dosežen kvadrant.

Kontrole za upravljanje:

- začetek testnih voženj,
- začetek tekme,
- začetek merjenja časa,
- prikaz vmesnega časa za 5 sekund,
- konec merjenja časa,
- prikaz trenutnih/končnih rezultatov študentov,
- prikaz trenutnih/končnih rezultatov dijakov.

Podrobna razlaga kontrol je opisana v diplomski nalogi z naslovom Robotiming meritev časa avtorja Igorja Rakuša.



Slika 4.22: Diagram poteka prikaza merjenja časa

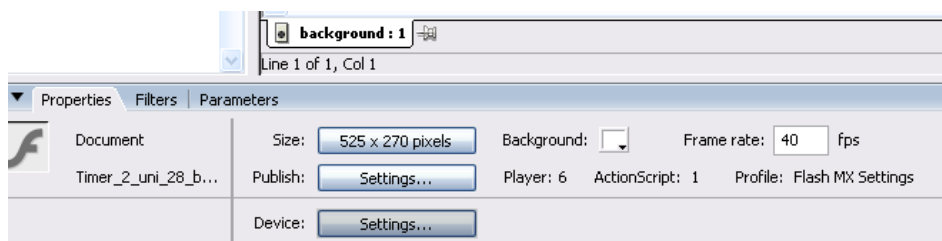
### Izdelava prikaza merjenja časa v Flashu

Ko so bile postavljene okvirne zahteve in znan scenarij, po katerem se bo merjenje časa odvijalo (sl. 4.22), je bilo potrebno zasnovati grafično podobo prikaza merilca časa, ki je morala biti skladna z osnovno modro-belo kombinacijo.



Slika 4.23: Prikaz merjenja časa v času testnih voženj

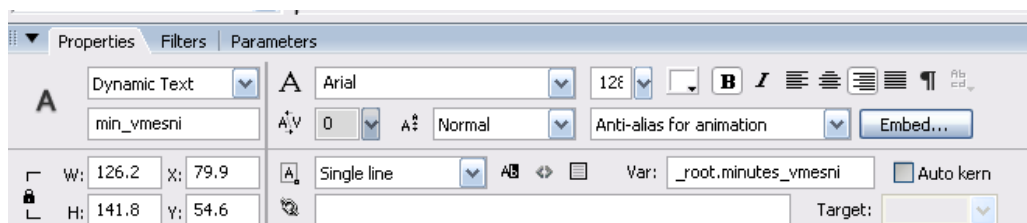
Nato sem izdelal čisto preprosti prikaz časa na stotinko sekunde natančno, ki sem ga preko proženja osnovnih ukazov lahko pognal, trenutno zaustavil, ustavil in resetiral (sl. 4.23). Namen tega preprostega merilca časa je bil preizkusiti tekoče prikazovanje, ki je potem služilo kot osnova za nadaljnje delo. Gladkost animacije štetja se v Flashu nastavlja s parametrom, ki določa število slik, ki se prikazujejo na sekundo. S povečevanjem tega parametra pa se po drugi strani povečuje zahtevnost oz. potreba animacije po procesorskem času. Zato sem poiskal kompromis glede na vizualni učinek in obremenitev procesorja ter nastavlil Flash animacijo, da je delovala s štiridesetimi slikami na sekundo. Prav tako sem ustrezno določil še velikost Flash animacije (sl 4.24), ki je sovpadala z dimenzijami predvidenimi v celostnem prikazu.



Slika 4.24: Osnovne nastavitve Flash animacije

Sledilo je vstavljanje grafičnih elementov v Flash ter ustvarjanje polj za prikaz vsebine. Flash loči med dvema tipoma teksta - dinamičnim in statičnim. Pri statičnem tekstu že ime pove, da gre za tekst okvirček, kjer vpišemo tekst, ki je fiksni in se ne spreminja. Če pa želimo v nekem okvirčku vsebino dinamično spreminjati pa moramo uporabiti dinamični tekst okvirček. Vsakemu okvirčku določimo tudi ime objekta, preko katerega določamo prikaz tega dinamičnega tekstovnega polja in spremenljivke, preko katere v kodi določimo

vsebino dinamičnega tekstovnega polja. Vsakemu tekstovnemu polju določimo tudi osnovne parametre, kot so velikost in tip pisave, poravnava, glajenje robov pisave. Tekstu lahko določimo tudi hiperpovezavo ter vgradimo različne nabore znakov (sl. 4.25).



Slika 4.25: Nastavitve dinamičnega tekstovnega polja, ki prikazuje minute vmesnega časa

Potrebno je bilo narediti mnogo dinamičnih tekstovnih polj, ki so se ob določenem dogodku prikazala z ustreznimi podatki; od dinamičnih polj, ki so osveževala stotine, sekunde, minute in ure, do polj za prikaz imen tekmovalcev, uvrstitev, vmesnih časov, skupne uvrstitve in ostalih. Da bi tem dinamičnim tekstovnim poljem lahko zagotovili najnovejše in sveže podatke pa je bilo potrebno uporabiti eno izmed ustreznih metod komunikacije med Flashom in ostalimi programskimi jeziki oziroma podatkovnimi bazami. Uporabil sem metodo LoadVar, ki sem ga opisal na naslednjih straneh.

### Pošiljanje in nalaganje spremenljivk

Swf datoteka, ki jo ustvari Flash, predstavlja okno za zajem in prikaz informacij, podobno kot HTML spletna stran. Kljub temu pa imajo swf datoteke dobro lastnost, da lahko ostanejo naložene v spletnem brskalniku in se kontinuirano osvežujejo z novimi informacijami, brez da bi bilo za to potrebno ponovno naložiti celo spletno stran. Z uporabo funkcij in metod ActionScript-a, je možno pošiljati in sprejemati informacije iz strežniško orientiranih skript ter sprejemati informacije iz tekstovnih in XML datotek.

Strežniško orientirane skripte lahko zahtevajo določene informacije iz podatkovne baze in jih nato prenesejo v swf datoteko. Strežniško orientirane skripte so lahko napisane v različnih skriptnih jezikih; nekatere najbolj pogoste so PHP, ASP, CFML, PERL. S shranjevanjem informacij v podatkovno bazo in pridobivanjem le-teh nazaj, je možno ustvariti dinamično in specifično vsebino za neko swf datoteko.



Obstajajo različne ActioScript funkcije in metode, ki dopuščajo prenašanje informacij v in izven swf datoteke. Vsaka funkcija in metoda uporablja protokol za prenos informacij in zahteva, da so podatki urejeni na različne načine:

- Funkcije in MovieClip metode, ki uporabljajo http ali https protokol za pošiljanje podatkov v URL-kodiranem formatu so: `getURL( )`, `loadVariables( )`, `loadVariablesNum( )`, `loadMovie( )` in `loadMovieNum( )`.
- Metode `LoadVars`, ki uporabljajo http ali https protokol za pošiljanje in nalaganje podatkov v URL-kodiranem formatu so: `load( )`, `send( )` in `sendAndLoad( )`,
- Metode, ki uporabljajo https ali https protokol za pošiljanje in nalaganje podatkov v obliki XML so: `XML.send( )`, `XML.load( )` in `XML.sendAndLoad( )`.
- Metode, ki ustvarijo in uporabljajo TCP/IP (angl. socket) povezave za pošiljanje in prejemanje podatkov kot XML so: `XMLSocket.connect( )` in `XMLSocket.send( )`.

Pomembno je, da je datoteka iz katere beremo, shranjena v UTF-8 (priporočljivo), UTF-16BE ali UTF-16LE formatu.

Funkcije in metode, kot so `loadVariables`, `getURL`, `loadMovie` in `loadMovieNum`, lahko komunicirajo s strežniško orientiranimi skripti s pomočjo http in https protokola. Te funkcije in metode pošiljajo vse spremenljivke iz časovnice v Flashu, v kateri se ta funkcija oziroma metoda nahaja. Če se uporabljajo kot metode znotraj nekega MovieClip objekta, `loadVariables()`, `getURL()` in `loadMovie()` pošiljajo vse spremenljivke, ki se nahajajo znotraj nekega MovieClip-a. Vsaka funkcija ali metoda obravnava svoje odzive na različne načine:

- `getURL()` funkcija vrača kakršnokoli informacijo v okno spletnega iskalnika in ne v Flash predvajalnik,
- metoda `loadVariables()` naloži spremenljivke v točno določen del časovnice oziroma nivo v Flash predvajalniku,
- `loadMovie()` metoda naloži swf datoteko v točno določen nivo ali Flash animacijo v Flash predvajalniku,

Pri uporabi `loadVariables()`, `getURL()` ali `loadMovie` lahko uporabimo še različne dodatne parametre:

- URL je pot do datoteke, kjer se nahajajo spremenljivke,
- LOKACIJA je nivo ali tarča v swf datoteki, ki sprejme spremenljivke (`getURL()` funkcija ne sprejme tega parametra),
- SPREMENLJIVKE določajo http metodo, ki je lahko ali GET (doda spremenljivke na konec url naslova) ali POST (pošlje spremenljivke v različnih http glavah), s pomočjo katerih lahko pošljamo spremenljivke. V primeru, da tega parametra ne uporabimo, bo Flash predvajalnik uporabil prednastavljeno spremenljivko GET, vendar ne bo poslana nobena spremenljivka,

Če bi na primer želeli v Flashu prikazovati trenutne rezultate, ki so shranjeni v neki datoteki na strežniku, bi uporabili `loadVariables()`, da bi jih naložili v nek določen "movie clip" npr. `rezultati_mc`. Primer klica funkcije za zajem teh podatkov oziroma spremenljivk je sledeč:

```
this.createEmptyMovieClip("rezultati_mc", 10); // ustvarimo nov movie clip in ga postavimo na globino 10
```

```
loadVariables("rezultati.php", rezultati_mc, "GET") // prenesemo spremenljivke iz datoteke rezultati.php v movie clip rezultati_mc
```

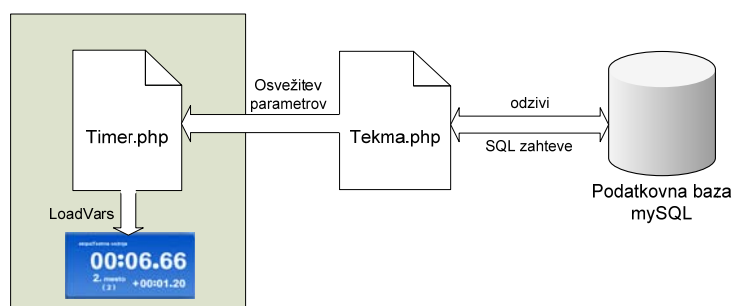
Datoteka, ki je določena v URL parametru (v našem primeru je to `rezultati.php`) funkcije `loadVariables()`, mora vsebovati vrednosti v posebni obliki, da jih lahko Flash prebere. Datoteka lahko vsebuje neomejeno število spremenljivk; spremenljivke in njene vrednosti pa morajo biti ločene z „ampersandi“ (&), in besede znotraj ene vrednosti morajo biti ločene z znakom plus (+). Priporočljivo je, da je ta datoteka shranjena v UTF-8 formatu.

Primer:

```
mesto1=Joze+Novak&mesto2=Miran+Gradisek&mesto3=Goran+Komar
```

Še večjo fleksibilnost kot `loadVariables()` pri prenosu podatkov v swf datoteko s strežnika lahko dosežemo z `LoadVars` razredom. Le-ta je bil predstavljen s Flash predvajalnikom verzije 6.0 in ponuja čistejši, bolj objektno orientiran vmesnik za pogosto opravilo izmenjave podatkov s spletnim strežnikom. Prednosti razreda `LoadVars` so naslednje:

- Ne rabimo več kreirati posebnih movie clipov za shranjevanje podatkov, ki se prenesejo vanje preko klient/strežnik komunikacije.
- Vmesnik razreda je podoben vmesniku XML objekta, ki ponuja nekaj združljivosti z ActionScriptom. Uporablja metode `load()`, `send()` in `sendAndLoad()` za vzpostavitev komunikacije s strežnikom. Glavna razlika med `LoadVars` in XML razredom je v tem, da so `LoadVars` podatki vsebovani v `LoadVars` objektu, za razliko od XML Document Object Model (DOM) drevesno strukturiranega shranjevanja v XML objektu.
- Možno je dobiti dodatne informacije o komunikaciji s pomočjo uporabe metode `getBytesLoaded` in `getBytesTotal`,
- Možno je spremljati tudi napredovanje nalaganja podatkov, čeprav do njih ne moremo dostopati vse dokler niso popolnoma naloženi,
- Na voljo imamo sporočilo o napakah nalaganja,
- Dodamo lahko poljubne glave s http zahtevami,



Slika 4.26: Shematski prikaz pridobivanja podatkov potrebnih za animacijo ter prikaz rezultatov in uvrstitev

Glede na potrebe in zahteve za prikaz sem se odločil uporabiti razred `LoadVars`. Preko njega sem v zanki nalagal podatke iz datoteke `timer.php`, ki jih je predhodno ustvarila php

skripta, imenovana `tekma.php`, na podlagi podatkov pridobljenih iz podatkovne baze in realno časovnih meritev, opravljenih s pomočjo senzorskega sistema (sl. 4.26). Težava se je pojavila, ker je animacija Flash imela veliko časovno odvisno porabo pomnilnika in je tako po pol ure delovanja porabila za svoje delovanje 512 MB pomnilnika. Kot se je izkazalo, je bilo kljub dejstvu, da je bila količina podatkov relativno majhna in da so bili shranjeni lokalno, branje iz datoteke v zanki prehitro v primerjavi z nalaganjem podatkov. Flash animacija je ob začetku branja parametrov iz datoteke rezervirala določen prostor v pomnilniku za nove parametre, vendar je še pred koncem branja prvih parametrov pričela s ponovnim branjem in rezervacijo prostora v pomnilniku za nove. Tako se je količina navidezno rezerviranega pomnilnika večala iz minute v minuto. Sistem pa se je sčasoma upočasnil in na koncu "zmrznil" zaradi pomanjkanja pomnilnika. Da bi se temu lahko izognil, sem ponovno branje datoteke omejil glede na uspešno predhodno naložitev podatkov.

To sem dosegel s pomočjo povratnega klica preko ActionScript metode `onLoad`. Ta možnost je velika prednost, ki jo ima razred `LoadVars` pred arhaičnim `onClipEvent (data)` pristopom, ki ga zahteva `loadVariables`. Nato je bilo potrebno napisati funkcije, ki urejajo prikaz ob posameznih dogodkih. Tako je na primer za prikaz testnih voženj potrebno skriti vse razen napisa testne vožnje in prikaz resetiranega števca. Celoten potek merjenja časa pri testnem načinu poteka nekoliko drugače, saj pri vmesnem času ne kažemo primerjave z ostalimi vmesnimi časi, ampak se preprosto "štoparica" samo ustavi za nekaj sekund in nato nadaljuje s prikazom merjenja časa. Ko je robot zaključil testno vožnjo, se izpiše samo končni čas, ki se ga potem preko nadzornega programa resetira in že je lahko na vrsti naslednji tekmovalac.

Ko pride na vrsto tekmovanje, je potrebno preko kontrolnega programa to pognati, parameter `test` v datoteki `timer.php` se spremeni iz 1 na 0, kar Flash animaciji pove, da se je tekmovanje začelo in da je potrebno prikazati ime ekipe in resetirano štoparico. Flash animacija nato čaka, da se bo parameter `start_timer` iz 0 spremenil na 1. Ko se to zgodi, se sproži štoparica. Le-ta čaka na spremembo parametra `pause_timer` iz 0 na 1. Takrat se prikaže vmesni čas, uvrstitev glede na vmesni čas (v trenutni vožnji in skupno) ter primerjava glede na najboljši vmesni čas. Ko robot pride na cilj, se parameter `stop_timer`

postavi iz 0 na 1 ter s tem sproži prikaz končnega rezultata, uvrstitve ter časovne primerjave glede na pretekle najboljše končne čase.

Kadarkoli med vožnjo se lahko merjenje časa oziroma štoparica ustavi, saj se v primeru da poteče časovna omejitev robota, ki je v labirintu, ali da se je le-ta kje zataknil kot uvrstitev šteje najbolj oddaljen kvadrant od starta, ki je ustrezno točkovan in ga je robot v tisti vožnji dosegel. Takrat se parameter `kva_event` spremeni iz 0 na 1 in tako naznani konec ter Flashu da ukaz, da mora prekiniti s štopanjem in prikazati končni rezultat ter dosežen kvadrant prebranega iz parametra `kvadrant`.

Glede na to, da se naslovi strežnikov vsako leto menjajo in da je potrebno za vsako tako menjavo popraviti v `actionsriptu` hiperpovezavo do datoteke `timer.php` ter nato ponovno ustvariti Flash animacijo, sem vanjo vgradil funkcijo, ki ob prvem zagonu Flasha iz datoteke `server.txt` pridobi naslov strežnika. Datoteka `server.txt` se mora nahajati vedno v istem imeniku kot Flash animacija. Tako lahko sedaj vsako leto popravimo samo datoteko `server.txt` oziroma parameter `naslov_serverja` in že je Flash animacija pripravljena za uporabo oziroma merjenje časa na tekmovanju.

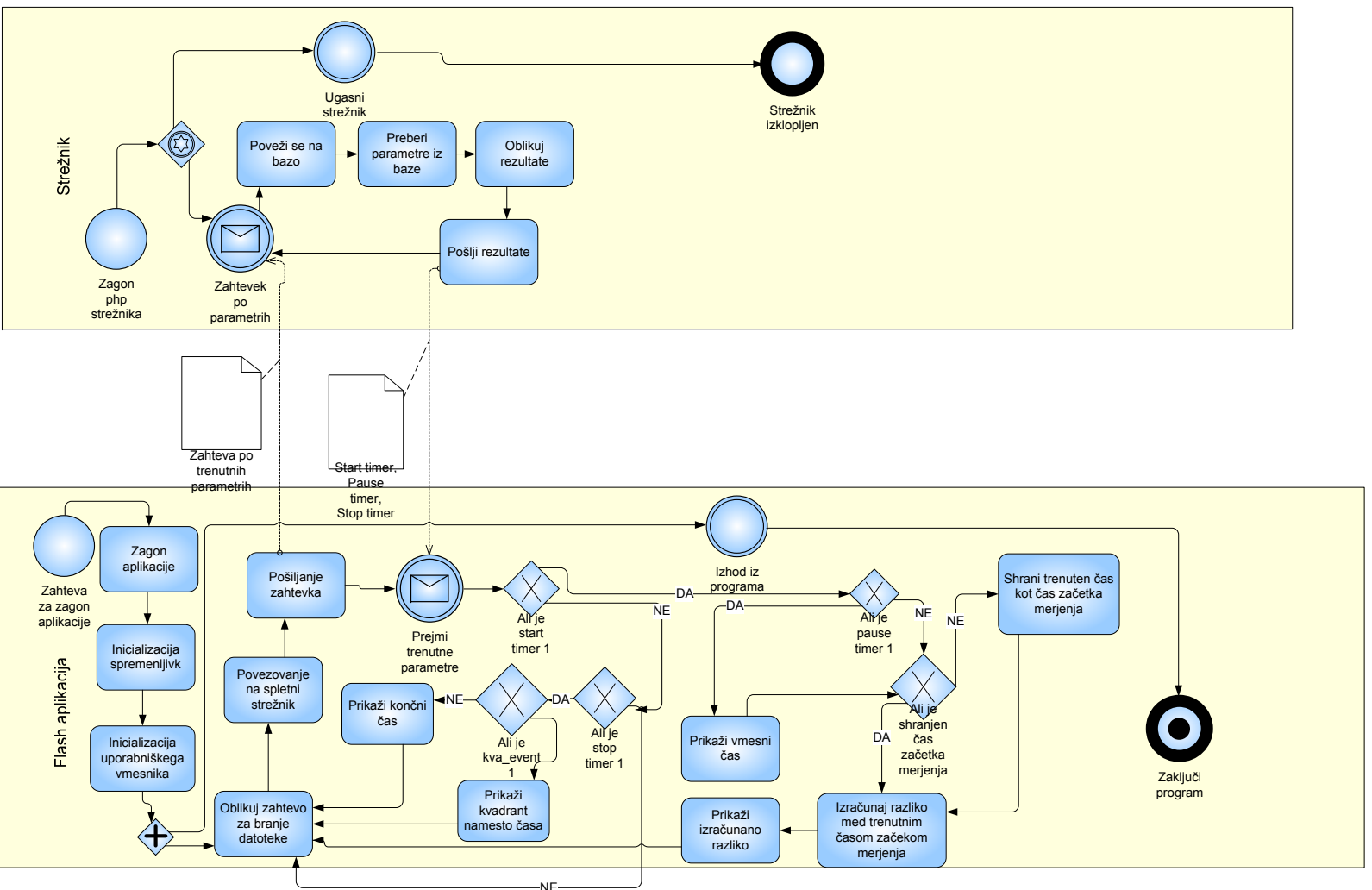
Preklop iz tekmovanja na prikaz rezultatov oziroma prikaz rezultatov druge kategorije je rešen z odprtjem nove spletne strani v istem oknu, kjer se s pomočjo `php`-ja zajamejo podatki iz podatkovne baze in ustrezno, glede na grafično predlogo, prikažejo v `html` obliki. Da bi se lahko vrnil nazaj na tekmovanje ali prikaz rezultatov druge kategorije (npr. iz dijaške v študentsko), je bilo potrebno v spletno stran prikaza tekmovanja integrirati manjšo Flash animacijo. Naloga le-te je bilo branje iz datoteke `server.txt`, iz katere smo pridobili naslov strežnika, kjer se nahaja datoteka `timer.php`. Iz `timer.php` beremo spremenljivke in glede na spremembo posameznega parametra prikažemo ustrezno spletno stran, kjer se prav tako v ozadju nahaja ista Flash animacija, ki smo jo imenovali preklopnik. Prikažemo lahko tekmovanje ali rezultate v dijaški ali študentski kategoriji. Flash animacija, ki smo jo posebej vgradili na spletno stran v osnovi torej omogoča preklapljanje med prikazi startne liste, tekmovanja in rezultatov posameznih kategorij in tako sledi konceptu centraliziranega upravljanja merjenja časa in prikaza s pomočjo glavnega kontrolnega programa.

Na preteklih tekmovanjih sem opazil, da je veliko ljudi gledalo v sam labirint in da je bila njihova pozornost usmerjena predvsem v vožnjo robotov. Z uvedbo prikazovanja

rezultatov s pomočjo projekcije pa se je pojavila še ena zanimiva informativna točka za gledalce z dodatnimi informacijami. Da bi gledalce in tekmovalce opozorili na prikaz rezultatov neposredno med izvedbo štopanja, sem v Flash prikaz vgradil še zvočne efekte, ki se sprožijo ob startu, vmesnem času in na cilju ob prikazu končnega rezultata in uvrstitve. Ker so zvočni efekti ves čas enaki, tekmovalci in gledalci postanejo pozorni na njih. Gledalci tako že po pisku ločijo ali gre za start, vmesni čas ali končni čas. Najbolj pomembno pa je, da se pritegne njihova pozornost in so na tak način v celoti informirani o dogajanju v labirintu in najnovejših informacijah, projiciranih na velikem platnu.

### **Dokumentiranje prikaza in merjenje časa na tekmovanju s pomočjo BPMN [6]**

Modeliranje poslovnih procesov BPMN (angl. Business Process Modeling Notation) se uporablja za opisovanje, dokumentiranje in nazorni prikaz procesov, interakcije med njimi in podatkov, ki si jih izmenjujejo. BPMN uporablja privatne (interne) in abstraktne (zunanje) procese ter medsebojno sodelujoče procese (interakcija med različnimi poslovnimi entitetami). BPMN je grafična notacija za modeliranje poslovnih procesov in delovnih tokov ter kot taka zelo nazorno prikazuje interakcijo med različnimi stezami oziroma udeleženci v procesu. Z uporabo BPMN se zmanjšuje prepad med načrtovanim in implementacijo procesov in delovnih tokov, saj njena specifikacija poleg grafičnih simbolov definira tudi preslikavo diagrama procesa v izvršljiv jezik, ki temelji na XML. Glede na vse te prednosti in dejstvo, da nam BPMN lahko služi tudi kot del poslovnika kakovosti (ISO 9001:2000), delovna navodila za zaposlene in obenem tudi kot osnova za informacijski sistem sem se odločil, da tudi sam uporabim to grafično notacijo in z njo opišem prikaz in merjenje časa na tekmovanju (sl. 4.27). Nenazadnje, dandanes vsi največji ECM (Enterprise Content Management) sistemi vsebujejo BPM oziroma grafično okolje, v katerem s pomočjo gradnikov BPMN notacije, ki je nastala kot razvijajoči se standard za različne BPM jezike, programerji, informatiki in poslovni uporabniki načrtujejo in implementirajo tudi najbolj zahtevne poslovne procese oziroma informacijsko logiko. Kot primer lahko navedemo ECM sisteme kot so SAP, Documentum, Filenet (IBM), ki jih uporabljajo največja svetovna podjetja v Sloveniji. Njihovi uporabniki so na primer: Revoz, Krka, Lek, Cimos, Sava, Gorenje, Telekom Slovenije itd...



Slika 4.27: Uporaba BPMN za grafični opis procesov in medsebojno izmenjevanje podatkov pri prikazu in merjenju časa na tekmovanju

### **4.3      *Spreminjanje grafičnega vmesnika v Photoshop-u***

Kljub veliki stopnji avtomatiziranosti Robotiming sistema je potrebno vsako leto izvesti določene spremembe. Vzroki za to ležijo predvsem v malenkostnih spremembah samega tekmovanja, eventualne spremembe spletnega strežnika na katerem robotiming teče, spremembe pri sponzorjih, drugačni teksti na spletnih straneh in uvodnem pozdravu na tekmovanju. Poleg tega pa je v glavi sistema robotiming tudi letnica, ki jo je potrebno vsako leto znova osvežiti.

#### **Predpriprave za prikaz na tekmovanju**

Priprava glave

Najprej si moramo v sistem naložiti dodatno pisavo v kateri je bila ta glava izdelana – MicrogrammaDEEMedExt. Nato v grafičnem orodju Photoshop odpremo datoteko z imenom robotiming\_napis\_prikaz\_na\_tekmovanju\_template.psd ter popravimo letnico. Leto shranimo pod imenom frame\_zgornji.jpg.

Priprava uvodnega teksta in urnika

V photoshopu odpremo datoteko z imenom frame\_levi.psd, ustrezno popravimo vsebino in sliko shranimo kot frame\_levi\_zac.jpg.

#### **Priprava okna s spoznorji**

Glede na to, da so vsako leto določene sponzorske spremembe, je potrebno v photoshopu odpreti datoteko frame\_desni.psd ter ustrezno dodati ali izbrisati sponzorske logotipe. Zaradi enotnega izgleda je priporočljivo, da ima logotip transparentno ozadje ali vsaj belo. Končno podobo je potrebno shraniti po imenom frame\_desni\_zac.jpg.

Zaradi dinamičnega branja naslova strežnika, kjer se sistem robotiming nahaja in glede na katerega se potem ravnaajo Flash animacije, kot so prikaz štoparice in rezultatov ter preklopi, je nujno ustrezno popraviti datoteko server.txt. V njej se nahaja parameter



naslov\_serverja, ki hrani informacijo o naslovu strežnika, ki gostuje robotiming. Server.txt se mora nahajati v imeniku tekma, tako kot vse datoteke, ki služijo za prikaz na tekmovanju (npr. [www.naslov\\_serverja/tekma/](http://www.naslov_serverja/tekma/)).

Do osnovne strani, ki je sestavljena iz štirih okvirjev, dostopamo preko index2.htm. Znotraj nje se potem v vsakem okvirju naloži posamezna stran. Na vrhu imamo okvir, kjer se nahaja glava in v katerem se naloži datoteka frame\_zgornji.htm. Dva okvirja, ki prikazujeta startno listo ali sponzorje in trenutno listo rezultatov ali uvodni tekst, vsebujeta html kodo in vsebino, ki se dinamično ustvari s pomočjo podatkovne baze in php skript. Ti dve strani frame\_levi.php in frame\_desni.php se osvežujeta na dve sekundi. Poleg tega imamo tudi okvir s stranjo, imenovano frame\_sredinski.htm, kjer je prikazana Flash štoparica. Prednost le-te je v tem, da ne potrebuje nenehnega osveževanja strani, da bi pridobila sveže podatke iz podatkovne baze. Tak način pridobivanja bi bil sicer tudi zelo počasen in neprimeren za nek prikaz merjenja časa v realnem času. Namesto tega Flash animacija v zanki bere datoteko timer.php, v kateri se nahajajo parametri, ki jo kontrolirajo. Seveda je hitrost branja odvisna od hitrosti povezave oz. dostopa do strežnika. V našem primeru se je datoteka timer.php, do katere je dostopala Flash štoparica, nahajala v isti omrežni soseščini, kar je omogočalo branje parametrov nekje 2-3 krat na sekundo. Taka hitrost branja oziroma osveževanja Flash štoparice se je na samem tekmovanju izkazala za zadovoljivo.

Flash animacija preklopnik je že integrirana v Flash štoparico in omogoča preklapljanje iz osnovne strani index2.htm oziroma osnovnega prikaza tekmovanja na prikaz startne liste študentov/dijakov oziroma trenutnih/končnih rezultatov dijakov/študentov.

Vse te strani (rezultati\_dij.htm, rezultati\_stu.htm, start\_dij.htm, start\_stud.htm) so sestavljene iz dveh okvirjev. Prvi vsebuje glavo z grafiko, ki označuje vsebino prikaza, drugi pa php skripto, ki prikaže ustrezne, glede na podatke iz baze in po predhodno izdelani grafični predlogi razvrščene, podatke.

### **Predpriprave za prikaz na svetovnem spletu**

Vse datoteke, ki spadajo k prikazu na spletu, se morajo nahajati v mapi [www.ime\\_serverja/tekma/www/](http://www.ime_serverja/tekma/www/). Prav tako moramo v to mapo skopirati tudi server.txt in

mu ustrezno nastaviti parameter `naslov_serverja`. Osnovna stran je sestavljena iz Flash menija, glave in vsebine.

Hiperpovezave v osnovnem meniju na vrhu strani se dinamično določajo glede na parameter iz datoteke `server.txt` in jih tako ni potrebno urejati.

### **Priprava glavnega menija**

V datotekah `meni fla` in `meni_1 fla` popravimo tekst Robot 200x na ustrezno letnico in nato s pomočjo Flasha ustvarimo datoteki `meni.swf` in `meni_1.swf`.

### **Priprava glave**

Pod meniji se nahaja glava, ki je narejena v Flashu in vsebuje sliko glave široko 800 pik. Toliko tudi znaša širina celotne spletne strani. Flash dinamično naloži sliko iz lokalne mape, zato moramo zamenjati samo datoteko `robotiming_xzyw_web.png` z najnovejšo. To pa naredimo s pomočjo orodja Photoshop, s katerim odpremo dokument `robotiming_napis_prikaz_na_web_template.psd`, popravimo letnico tekmovanja in shranimo pod imenom `robotiming_xyzw_web.png`.

### **Priprava vsebine osnovne strani**

Osnovna stran vsebuje določene informacije v zvezi s tekočim in preteklimi tekmovanji. Vsebino lahko uredimo s pomočjo orodja Adobe Flash. Odpremo datoteko `GUMBI8_SPOD_1.FLA` ter popravimo vsebino. Prav tako je potrebno preveriti hiperpovezave, ki se nahajajo v tekstu. Na desni strani dokumenta `GUMBU8_SPOD_1.FLA` pa se nahaja komponenta, katera skrbi za predvajanje videa iz flv datoteke. Pomembno je, da preverimo pot do videa, za katerega želimo da se predvaja, ko uporabnik na spletni strani preleti z miško čez sliko robota. To storimo z dvoklikom na `movie Clip` (angl.) z imenom »robot« v knjižnici elementov. Prikaže se nam urejanje `movie clipa robot`. Kliknemo na siv pravokotnik, ki se nahaja na sceni. Na dnu v orodni vrstici z nastavitvami izberemo jeziček »parametri« in izberemo `Lounch Component Inspector`, kjer preverimo ustreznost poti do filma, ki želimo, da se bo predvajal. Na koncu

kliknemo na File, Export settings in ustrezno popravimo pot, kamor želimo, da se ustvari končna datoteka gumbi8\_spod.swf.

### **Priprava strani za predvajanje videa v živo**

Osnovna stran se imenuje tekma.htm in je razdeljena na štiri okvirje. V zgornjih dveh se nahajata glava in osnovni meniji (dokumenta tekma\_sz\_le.htm in tekma\_zg\_de.htm). V spodnjem levem okvirju pa se nahajajo trenutni rezultati, ki se dinamično osvežujejo na vsakih dvajset sekund in se prikazujejo s pomočjo php skripte (dokument tekma\_spo\_le.php). V spodnjem desnem okvirju pa se nahaja dokument tekma\_spo\_de.htm, ki vsebuje video komponento, katera se povezuje na pretočni video v živo s tekmovanja. Da bi ta video povezava ustrezno delovala, je potrebno v dokumentu tekma\_spo\_de.htm popraviti pot do pretočnega videa ter v lokalno mapi shranjene asx datoteke, ki služijo predvsem uporabnikom spletnega brskalnika Mozilla ter vsem ostalim, ki jim avtomatska povezava s predvajalnikom Media Player ne deluje, da si jo shranijo in jo ročno odprejo v Media Playerju ter se tako vseeno lahko povežejo na pretočni video. V Internet Explorerju se povezava na pretočni video vzpostavi avtomatsko.

Uspelo nam je izvesti sistem brezkontaktnega merjenja časa ter njegov prikaz na samem mestu dogodka in svetovnem spletu. Sistem smo poimenovali Robotiming, kateri je bil prvič uporabljen na tekmovanju Robot 2004, s tem pa smo tudi izpolnili svoj zadani cilj in naredili tekmovanje še bolj razburljivo in dostopno širšemu krogu gledalcev.



---

## POGLAVJE 5

---

### 5 SKLEP

Razvoj novih tehnologij in orodij ter njihova vsesplošna uveljavitev sta omogočila hitro realizacijo projekta Robotiming vse od zasnove pa do realizacije. Projekt je potekal po fazah, ki so obsegale analizo obstoječega stanja, načrtovanje, implementacijo in testiranje. Flash kot tehnologija je v zadnjih letih doživela izjemen uspeh na področju prijaznih grafičnih vmesnikov in s svojo razširjenostjo ter neodvisnostjo predstavlja od operacijskega sistema vodilno animacijsko orodje za spletne aplikacije. O tem priča tudi dejstvo, da ima Flash predvajanik 98 % razširjenost med osebnimi računalniki, povezanimi na spletu. Da bi se tehnologija Flash še bolj razširila, je podjetje Adobe razvil platformo Flex ter programsko okolje AIR, ki omogoča poganjanje Flash/Flex aplikacij kot namizne aplikacije in s tem omogočil funkcije, kot so npr. povleci in spusti datoteke med flash in namiznimi aplikacijami. Razvoj Flex in AIR-a odpira ogromne možnosti uporabe in razširitve Flasha, tako med splošnimi uporabniki spleta kot tudi profesionalnimi, poslovnimi uporabniki. Potrebno se je namreč zavedati, da se dandanes tudi podjetja vedno bolj zavedajo pomena samega načina komuniciranja med podjetjem in končnimi uporabniki v smislu sodobnih uporabniških vmesnikov. V te namene se tudi poslovne aplikacije skušajo vedno bolj približati končnim uporabnikom in ne obratno. Da je tehnologija res dozorela, pričajo tudi zadnje informacije iz podjetja Adobe, ki je objavil namero o začetku projekta Open Screen, katerega vizija je odstraniti ovire, ki pestijo

razvijalce pri razvoju aplikacij za različne platforme – torej za običajne operacijske sisteme, mobilne naprave in multimedijske centre. Poleg tega pa se je Adobe odločil, da bo objavil SWF in FLV specifikacije ter odstranil licenčne omejitve SWF in FLV/F4V formatom. Da bi se Flash še bolj razširil tudi med uporabniškimi vmesniki na vgrajenih sistemih pa namerava Adobe izdati še dodatne API-je za naprave, objaviti specifikacije za Flash Cast (flash platforma za mobilne naprave) ter AMF in odstraniti licenčnine za Flash Player in AIR naprave. Ravno uporaba animacijskega orodja Flash pa omogoča konsistenten prikaz in spremljanje mobilnih robotov v živo tako na računalnikih, ki tečejo pod Windows kot tudi odprto kodnih operacijskih sistemih, MacOS in drugih.

Cilj mojega dela in diplomske naloge je bil čim bolj nazorno in uporabniško usmerjeno, s pomočjo najnovejših tehnologij prikazati robotsko tekmovanje na samem kraju dogodka kot tudi na spletni strani. S pomočjo Flasha sem tako izdelal dinamično aplikacijo, ki je služila kot prikaz na tekmovanju kot tudi spletno stran, na kateri je bilo moč tekmovanje spremljati v živo preko interneta. Posebnost projekta je bila tudi ta, da je moral biti realiziran in preizkušen do datuma in ure natančno (do pričetka tekmovanja), kar je predstavljalo še dodaten izziv. Uporaba tehnologije Flash za prikaz na samem tekmovanju se je izkazala kot robustna in neodvisna od operacijskega sistema. Obenem pa nam je omogočala, da je bil projekt uspešno zaključen relativno hitro in kar je najpomembneje, v roku. Prikaz merjenja na velikem zaslonu sem še dodatno popestril z zvočnimi efekti ter merjenjem vmesnega časa, kar je naredilo vzdušje na tekmovanju še bolj napeto in tekmovalno. Merjenje časa je zelo spominjalo na spremljanje avtomobilske dirke ali pa še najbolj na smučarsko tekmovanje. Sam koncept prikaza in upravljanja bi lahko uporabili tudi za namenske aplikacije za vodenje procesov (SCADA) v industriji, kjer se vmesniki z zasloni, občutljivimi na dotik in prijaznimi vmesniki za upravljanje, vedno bolj uveljavljajo. Še posebno pa so uporabniški vmesniki pomembni v komercialnem sektorju, kjer oblika in dizajn zagotavljata že polovico uspeha nekega produkta na tržišču.

---

## DODATEK A

---

# 1 SPLOŠNO O PLATFORMI FLASH [7]

## *1.1 Kdaj in kako se je razvijal – zgodovina*

Začetki Flash-a segajo iz idej, ki jih je imel Jonathan Gay v srednji šoli, fakulteti in kasneje, ko je delal za podjetje Silicon Beach Software in njegove naslednike. To se je začelo v osemdesetih letih. Januarja leta 1993 so Charlie Jackson, Jonathan Gay in Michelle Welsh ustanovili majhno podjetje imenovano FutureWave, ki se je ukvarjalo z razvojem programske opreme. Njihov prvi produkt, ki so ga razvili, se je imenoval SmartSketch. Namen risarskega programa je bil narediti risanje z računalnikom tako enostavno kot risanje na navaden papir. Čeprav je bil SmartSketch zelo inovativen program, pa ni dosegel velikega uspeha na tržišču. Z razvojem interneta je FutureWave začel spoznavati velik potencial v orodju za izdelavo vektorskih animacij na spletu, ki bi z lahkoto lahko konkuriral Macromediji in njeni tehnologiji Shockwave, ki je bila takrat po velikosti relativno okorna za takratna internetne povezave. Leta 1995 je podjetje FutureWave preoblikovalo svoje orodje SmartSketch, mu dodalo možnost animiranja vsake slike animacije posebej in ga izdala kot FutureSplash Animator na PC in Macintosh platformi. Ekipi se je medtem že pridružil programer Robert Tatsumi, umetnik Adam Grofcsik in specialist za stike z javnostjo Ralph Mittman. Programski paket je bil že takrat ponujen podjetju Adobe in uporabljen s strani Microsofta v začetkih dela z internetom. Decembra 1996 je Macromedia sprejela vektorsko orientirano animacijsko orodje in ga

kasneje izdala kot Flash 1.0. Leta 1997 je Macromedia izdala Flash 2 s podporo stereo zvoku, napredni integraciji bitnih slik, dodani so bili gumbi, možnost uporabe knjižnice in sposobnost avtomatskega animiranja - prehoda med dvema barvama. Flash 3 je izšel leta 1998 in s sabo prinesel velike izboljšave na področju animacij, predvajanja in postopka izdelave izvršnih datotek ter prvič predstavil možnosti enostavnega skriptnega jezika, ki je služil za interaktivnost. Prav tako je tisto leto Macromedia prodala svoj stotisoči programski paket. S prihodom Flash 4 leta 1999, ki je že imel podporo predvajanju zvokov v mp3 obliki ter avtomatskemu generiranju animacij glede na začetno in končno stanje, je Macromediji uspel velik podvig. Njihov Flash predvajalnik si je namreč naložilo že okoli 100 milijonov ljudi po celem svetu. Zanj je bila najbolj zaslužna Microsoftova odločitev, da bo Flash predvajalnik integriral v svoj Internet Explorer 5. Prvotno sicer Flash predvajalnik kot dodatek ni bil predhodno vgrajen v popularnih spletnih brskalnikih, tako da so ga morali uporabniki posebej pretočiti z Macromediine spletne strani. Leta 2000 pa je Flash predvajalnik postal del brskalnika AOL, Netscape in Internet Explorerja. Dve leti kasneje je bil vključen tudi v operacijski sistem Windows XP, in tako dosegel 92 % vseh internetnih uporabnikov. Leta 2000 je Macromedia izdala Flash 5, ki je predstavljal velik korak naprej glede funkcionalnosti, predvsem s predstavitvijo skriptnega jezika ActionScript, ki je na stežaj odprl nove možnosti upravljanja objektov s pomočjo skripta. Flash 6, ki je izšel leta 2002, je s seboj prinesel podporo za video, programske komponente, skupne knjižnice in povezljivost s podatkovnimi bazami. Septembra leta 2003 je izšel Flash MX 2004, ki se je lahko pohvalil s pohitritvami in optimizacijami, ki so bile tudi do osemkratne glede na predhodno verzijo, naprednim prevajalnikom, novim Flash predvajalnikom 7, možnostjo ustvarjanja grafov, tabel in dodatnih tekst efektov. Prav tako je bil takrat prvič predstavljen ActionScript 2.0, ki je narekoval razvijalcem vsaj formalno objektno orientiran pristop k programiranju v Flashu. Leta 2004 je bil predstavljen tudi Flash predvajalnik za mobilne telefone Flash Lite 1.1, kar je pomenilo začetek prodora Flasha tudi na področje »pametnih« telefonov. Leta 2005 je Macromedia izdala Flash 8, ki je takoj postal velika uspešnica, saj je prinašal ogromno novosti, izboljšav in pohitritev tako na področju efektov, animacij kot video kompresije. Dodani so bili filter efekti, začasno shranjevanje bitnih slik, novi video kodek imenovan On2 VP6, novi izboljššan pogon za izris poimenovan FlashType, emulator za mobilne naprave in mnoge izboljšave v ActionScript 2 specifikacijah. Decembra leta 2005 je bila Macromedija



z vsemi svojimi produkti (tudi Flashem) prevzeta s strani podjetja Adobe Systems. Nato je leta 2006 izšel Flash Player 9 za operacijski sistem Windows in Mac OS ter v januarju leta 2007 tudi pod Linuxi. Prvič v zgodovini Flasha bo prihodnja verzija predvajalnika prej širše razširjena, kot pa bo izdana ista različica razvojnega okolja. Zadnje informacije iz podjetja Adobe pa kažejo, bo Adobe odstranil licenčne omejitve SWF in FLV/F4V formatom ter objavil specifikacije, tako za SWF kot tudi za FLV. V prihodnosti pa nameravajo izdati API-je za naprave ter odstraniti licenčnine za Flash Player naprave.

Adobe [8] je maja letos tako objavil namero za začetek projekta Open Screen, katerega vizija je, da bi odstranili ovire, ki pestijo razvijalce pri razvoju aplikacij za različne platforme – torej za običajne operacijske sisteme, mobilne naprave in multimedijske centre – pri čemer nameravajo seveda koristiti prednosti obeh tehnoloških paradnih konjev Adobe-a – Flash Player in v prihodnosti tudi AIR.

Projekt, ki ga bom opisal, je bil razvit s takrat najbolj svežim razvojnim okoljem Flash verzije 8.0 podjetja Adobe Systems.

Flash animacije je možno predvajati na več različnih načinov. Najbolj običajna metoda je predvajanje v spletnem brskalniku kjer je Flash lahko vključen v html stran ali pa kot samostojna spletna stran, ki je sestavljena iz več manjših Flash animacij oziroma SWF datotek. Flash predvajalnik je prav tako na voljo kot samostojna aplikacija (znana kot projektor), ki se uporablja za predvajanje animacij brez da potrebujemo spletni brskalnik ali kakršen koli vtičnik. Tak način predvajanja se pogosto uporablja, kjer Flash animacije poganjamo z zgoščenke, diskov ali kakšnega drugega prenosnega medija.

## **Tipi datotek v Flashu 8**

Flash datoteke (.fla) so zasnovane tako, da zagotavljajo učinkovito razvojno okolje za projekte vseh velikosti. Znotraj tega okolja je vsebina organizirana na scene in spreminjanje vrstnega reda scen je enostavno ves čas izdelave animacije. Sloji omogočajo enostavno pripravo grafične vsebine znotraj vsake scene in skozi efekta "Guide" ter "Mask

layers" pomagajo pri risanju ali celo omogočajo posebne vizualne učinke. Časovnica prikazuje okvirje oziroma "keyframe", gibanje (angl. motion), oblikovne transformacije (shape tween), označbe in komentarje. V knjižnico (ki je lahko v skupni rabi tako med razvojem animacij kot tudi poganjanjem le-teh) lahko shranimo vse simbole, ki nastopajo v projektu od grafike, pisav, animiranih elementov, zvokov ali videa ter komponente.

### **Flash animacije**

Razvojno okolje iz Flash dokumenta ustvari končno Flash animacijo oziroma datoteko s končnico .swf. Ta tip datoteke je optimizirana verzija Flash dokumenta, ki vsebuje samo elemente, ki so dejansko uporabljeni pri Flash animaciji. Večina informacij, ki so se nahajale v originalnem Flash dokumentu (.fla) so izpuščene z namenom, da bi končna animacija v swf obliki bila čim manjša. Ko se Flash animacija izvozi, sicer vsi originalni elementi ostanejo, vendar pa se sloji sploščijo (flatten) in poganjajo samo na eni časovnici v zaporedju, ki je bil zasnovan v Flash dokumentu. Vsebina knjižnice se naloži in shrani na prvi okvir, kjer je uporabljena. Zaradi optimizacije se večkrat uporabljena vsebina iz knjižnice shrani samo enkrat in se skozi celotno animacijo, kjer se ta pojavlja, uporabljajo samo bližnjice, ki se nanašajo na to vsebino. Bitne slike in zvoki so stisnjeni („komprimirani“) glede na kvaliteto z različnimi nastavitvami.

Flash je v nasprotju z ostalimi programi hibridna aplikacija. Je interaktivno multimedijsko razvojno okolje ter sofisticirano animacijsko orodje, primerno za izdelavo animacij širokega spektra - od enostavnih spletnih okraskov, do predvajanja visoko kvalitetnih risank. Obenem pa Flash podpira tudi relativno enostaven, a močen skriptni jezik, imenovan ActionScript. Le-ta se je razvil iz omejenih »potegni in spusti« metod v visoko izkušen, objektno orientiran programski jezik, ki je zelo podoben JavaScriptu. Flash ActionScript lahko deluje v povezavi z XML (eXtension Markup Language), HTML in mnogimi ostalimi programi ter deli spleta. Flash vsebino je mogoče dinamično pridobiti s pomočjo mnogih strežniško orientiranih tehnologij, vključno z Flash Remoting MX in Flash Communication Server MX. Tudi Flash predvajalnik vsebuje integrirano podporo dinamičnemu nalaganju slik, MP3 datotek, videa in ostale vsebine. Flash lahko brez težave deluje s programom ColdFusion ali ostalimi aplikacijskimi strežniki, ki poganjajo PHP,

Microsoft .NET in XML »socket servers« ter s tem končnim uporabnikom omogoča dinamično in interaktivno izkušnjo.

Srce Flash razvojnega okolja je vektorska animacijska komponenta, ki najbolj izstopa in je praktično brez konkurence. Čeprav je Flash zmožen delati z bitnimi slikami, je njegov prvinski tip grafike vektorsko orientiran. Za razliko od ostalih animacijskih in multimedijskih programov se Flash pri generiranju končnih Flash animacij zanaša na majhen in optimiziran ter za internet prenos zelo primeren vektorski format. Namesto da bi shranjeval megabajte informacij o poziciji pik, Flash shrani podatke o vsakem okvirju v kompaktni vektorski obliki. Za razliko od Flasha pa se animacijski programi, ki bazirajo na bitni sliki (recimo Applov Quicktime) na vsak način trudijo prikazati vsako bitno sliko v čim bolj optimalni kakovosti glede na majhen prenos podatkov. Kot je potrebno, Flash hitro pretvori vektorske opise z majhno obremenitvijo, tako za podatkovni prenos kot tudi za računalnik končnega uporabnika. To pa je zelo velika prednost pri prenašanju animacij in ostale grafične vsebine preko spleta. Flash predvajalnik 6 in višji vsebujejo video pogon imenovan Sorenson Spark kodek, s čimer se vtičnik Flash predvajalnika uvršča med najmanjše na svetu. Flash predvajalnik 8 vsebuje dodatni video kodek, imenovan On2 VP6 codec, ki ima zavidljivo kompresijo in kvaliteto slike. Možno je tudi uvoziti video datoteke direktno v Flash 8 dokument (.fla) ali ločeno ustvariti Flash Video datoteke (.flv), ki se naložijo v Flash animacijo. Da bi uporabnik gledal video v Flashu tako ne potrebuje instaliran Apple QuickTime, RealSystems RealOne ali Microsoft Windows Media predvajalnik ali katerikoli drugi. Od Flash predvajalnika verzije 6.0, imajo Flash animacije (.swf) zmožnost, da naložijo MP3 datoteke med tem ko se le-te predvajajo. Prav tako je mogoče v Flash (.fla) uvoziti tudi druge tipe avdio datotek. Zvoki se lahko pripnejo na sličice ali gumbe, za zvok v ozadju ali zvočne efekte. Zvočna datoteka je lahko enakomerno porazdeljena po časovnici tako, da se lahko zvok že predvaja, medtem ko celotna Flash datoteka sploh še ni naložena. Če je srce Flasha vektorsko-risarsko orientirano okolje, potem je telo Flash-a zagotovo multimedijsko razvojno okolje. Flash dokumenti (.fla) lahko vsebujejo različne multimedijske dodatke, kot so zvok, video, animacije, slike. Kar je še več, Flash je močno orodje za izdelavo interaktivne vsebina, ker omogoča dodajanje komand v ActionScript-u in tako omogoča dinamično kontrolo nad predvajanjem Flash animacije (.swf). Večina multimedijskih razvojnih okolij imajo komponento za nizanje vsebine kot animacijo in tudi Flash tukaj ni izjema. V Flash-u je

animacijsko zaporedje jedro aplikacije. Okno s časovnico upravlja s prikazom celotne vsebine, tako statične kot dinamične. V časovnem oknu sta dva področja, ki omogočata organizacijo vsebine v vizualnem prostoru in na premočasovni osi. Sloji in mape slojev omogočajo spremljanje vsebine, ki je bila vstavljena v Flash dokument. Posebej se lahko nastavlja vidnost vsakega sloja in tako omogoča enostavno izoliranje posameznih elementov med delom. Sloji so vidni od začetka do konca znotraj vsakega okvirja na časovnici - elementi na zgornjih slojih prekrivajo elemente na spodnjih slojih. Na en sloj se lahko položi neomejeno število elementov, vendar se s tem zmanjša kontrola nad vrstnim redom prekrivanja. Znotraj posameznega sloja so vedno na najnižji nivo prekrivanja postavljene nepovezane vektorske črte in liki, medtem ko so na višjem nivoju bitne slike, tekst in povezani elementi. Struktura, ki povzroči iluzijo gibanja v Flash animaciji, je zaporedje slik oziroma okvirjev. Vsaka slika predstavlja časovno stoječ trenutek. S pomočjo upravljanja in premikanja glavnega lika skozi slike, lahko upravljamo hitrost, trajanje in zaporedje animiranega zaporedja. S tem, ko spreminjamo vsebino v slojih na vsakem naslednjem okvirju oz. sliki, lahko ročno ustvarimo animacijo slika-za-slika. Eden izmed razlogov, zakaj je Flash postalo tako zelo močno animacijsko orodje, je sposobnost avtomatskega interpoliranja in izdelave vmesnih animacij. S tem, ko določimo vsebino na začetnem in končnem okvirju ter izberemo funkcijo »motion tween« ali »shape tween«, lahko hitro ustvarimo ali spremenimo animacije, ki vsebujejo transformacije likov ali premikanje le-teh glede na vidno polje. Znotraj enega Flash dokumenta je možno kreirati več ločenih scen, kjer je vsaka scena nadaljevanje iste glavne časovnice. Scene je možno poimenovati in jim kadarkoli zamenjati vrstni red. Same scene se predvajajo od prve do zadnje brez prekinitev, razen če ni to posebej določeno z ukazom »actions«. Zadnje verzije Flasha so prinesle veliko povečanje možnosti, kako integrirati Flash vmesnike s strežniško orientiranimi tehnologijami in dinamičnega nalaganja vsebine s pomočjo XML, ColdFusion in novih strežniških tehnologij, kot so PHP, ASP, .NET, JSP, Flash Remoting in Flash Communication Server. Ta napredek izhaja iz izboljšave in relativne zrelosti programskega jezika ActionScript, ki se je izkazal tudi kot zelo praktičen programski jezik. Flash se je razvil v alternativni uporabniški vmesnik za velike podatkovne baze, kar pomeni, da lahko služi kot spletna trgovina, MP3 predvajalnik ali večigralska igra ter spletna klepetalnica, kar je velik dosežek za »animacijski program«.

## 1.2 *Katere so prednosti in slabosti*

Kaj zmore Flash 8 in kdaj ga uporabiti?

- Flash kreira zelo majhne datoteke, medtem ko na drugi strani producira visoko kvalitetne animacije z optimalno reprodukcijo zvoka. Celo podjetja, ki proizvajajo svetovno znane risanke, kot je npr. Disney, uporabljajo Flash pri določenih opravilih. Zaradi malih velikosti datotek, so Flash animacije (voščilnice, obvestila) primerne tudi za pošiljanje po elektronski pošti.
- Praktično je mogoče integrirati v Flash katerikoli tip datoteke. Vektorske slike (kot so EPS, FreeHand, Illustrator in PDF datoteke), bitne slike (GIF, PCT, TIF, PNG, in JPG), zvočne datoteke (kot so WAV, AIF ali MP3) in video (AVI in MOV) je možno vse uvoziti v animacije. Ni potrebe po vtičnikih ali dodatni programski opremi tretjega proizvajalca (čeprav tudi obstaja), da bi lahko uvozili te datoteke.
- Flash omogoča natančne izrise z vgrajenimi pisavami. Oblikovanje je pri uporabi HTML-ja za opisovanje izgleda običajno nekonsistentno in je prav tako odvisno od spletnega brskalnika. Z uporabo Flash animacij pa smo lahko sigurni, da bo oblika teksta in animacij enaka končnemu prikazu v Flash predvajalniku. Flash 8 je dodal podporo za čist izgled teksta tudi pri majhnih velikostih. Tako je sedaj kontrola nad tem, kako se pisava izriše pri majhnih velikostih pike, dosti večja. V predhodnih verzijah so se namreč pojavljale težave, da so bile majhne pisave oz. tekst nekoliko zamegljene in popačene.
- Tekst, animacije, slike, zvok ter video so lahko prikazane v Flash animaciji iz oddaljenega podatkovnega vira. V animacijo je možno vkomponirati dinamično vsebino pod pogojem, da je vir podatkov (kot je podatkovna baza ali XML datoteka) dosegljiva preko spletnega ali aplikacijskega strežnika. Flash predvajalnik 8 lahko komunicira direktno preko spletnih servisov, kar omogoča razvoj B2B (business to business) aplikacij, ki izkoriščajo prednost javnih ali privatnih virov informacij, kot so vremenska napoved ali informacije o delnicah.

- S pomočjo ustrezne programske opreme na strežniški strani je možno ustvariti večuporabniško interaktivnost. Od Flash predvajalnika 5 naprej je možno uporabiti XML »socket« za prenos podatkov med socket strežnikom in enim ali več povezanih Flash animacij. XML delovanje je hitrejše z uporabo Flash predvajalnika 6 ali novejšega. Tudi z izdajo Flash Communication Serverja MX imajo razvijalci na voljo konsistentni API (Application Programmable Interface) za razvoj večuporabniških aplikacij. Remote Shared Object so eden izmed mehanizmov, ki jih izrablja Flash Communication Server MX. Le-ta uporablja efektiven in optimiziran binarni protokol RTPM (Real Time Messaging Protocol) za prenos in osveževanje dinamične vsebine.
- Več manjših Flash animacij je možno naložiti v eno večjo. Možno je ustvariti nadrejeno Flash animacijo in nato vanjo naložiti več Flash animacij, vsako za individualno področje vmesnika ali predstavitev. Z uporabo te metode je možno v neki delovni in razvojni ekipi razdeliti opravila, kjer lahko več oblikovalcev in razvijalcev dela vzporedno. Tak potek dela omogoča kreiranje po velikosti optimizirane spletne strani in aplikacije, v katerih je veliko multimedijske vsebine, ki lahko deluje tako, da se pretoči vsebina po potrebi, medtem ko uporabnik operira s Flash animacijo.
- Flash 8 omogoča dinamično nalaganje slik in MP3 datotek z uporabo Flash predvajalnika 6 ali novejšega. Te verzije Flash predvajalnika lahko nalagajo standardne JPEG slike in progresivno pretakajo in igrajo MP3 datoteke. Flash predvajalnik 7 lahko celo progresivno pretaka Flash Video (.flv) datoteke preko standardnega spletnega protokola http. Flash predvajalnik 8 pa lahko progresivno nalaga tudi JPEG, PNG in GIF grafične formate. Progresivno pretakanje je možno pri vseh tipih datotek, ki jih je možno predvajati, še preden je celotna vsebina prenešana s pomočjo Flash predvajalnika. Pri progresivnem pretakanju spletni brskalnik začasno shranjuje vsebino. S pomočjo Flash Communication Server-ja pa je možno doseči pravo pretakanje avdio in video vsebin, kjer se nič ne shranjuje v Flash predvajalnik ali spletni brskalnik.
- Zelo uporabna novost v Flashu 8 za malo bolj napredne uporabnike predstavlja možnost kreiranja komponent. Le-te omogočajo razvijalcem, da ustvarijo ponovno

uporabljive predloge za Flash animacije. Komponente, ki so priložene razvojnemu paketu Flash 8 močno zmanjšajo razvojno dobo vmesnikov, ki so sestavljeni iz pogosto uporabljenih standardnih elementov na področju uporabniških vmesnikov. To so npr. vnosna tekstovna polja in izbirni gumbi. Komponente so tudi zelo enostavno prilagodljive v oknu z lastnostmi in veliko nastavitvev je mogoče spremeniti brez posega v ActionScript.

- Flash predvajalnik je narejen za različne operacijske sisteme in naprave, vključno z Windowsi, Macintosh, Solaris, Linux, OS/2, SGI IRIX, Pocket PC in celo z nekaterimi mobilnimi telefoni. 96 % spletnim obiskovalcem ni potrebno dodatno nalagati Flash predvajalnika, kar predstavlja velik potencial in ogromno tržno prednost.
- Flash animacije je možno poganjati tudi izven spletnega brskalnika povsem neodvisno in kot navadne izvršljive programe, ki so poznani kot Projector-ji. Le-ti so Flash animacije, ki se izvajajo s pomočjo posebnega vgrajenega predvajalnika in jih je možno zapeči na DVD, CD ali posneti na katerikoli drugi prenosni medij. Projector-ji so idealni za predstavitve ali kataloge.
- Podobno kot pri HTML straneh je vsebina iz Flash animacije lahko natisnjena direktno. PrinterJob API v Flash predvajalniku 7 ali višji ActionScript jezik omogoča natančen nadzor nad izgledom natisljive strani. Za razliko od starejših Flash predvajalnikov, lahko sedaj na tiskalnik preko Flasha pošljemo tudi več strani hkrati in tako naenkrat ustvarimo končno vsebino za tiskanje.

#### Slabosti Flasha:

- Flash animacije se predvajajo v spletnem brskalniku z uporabo vtičnika. Kljub dejstvu, da je Flash prisoten vsepovsod pa se še vedno najdejo uporabniki, ki morajo ta vtičnik posebej pretočiti z interneta. Prav tako je potrebno paziti, da ima veliko uporabnikov na internetu še staro verzijo Flash predvajalnika, ker je vmes niso posodobili. To v praksi pomeni, da je pri izdelavi Flash animacije potrebno biti kompatibilen s Flash predvajalnikom, ki je generacijo starejša od trenutno najnovejše. To pa po drugi strani onemogoča uporabo vseh novih lastnosti

("bonbončkov"), ki jih najnovejše verzije Flasha prinašajo s seboj. Drugo je, če je aplikacija individualna, torej ni namenjena za širšo množico ali pa je sestavni del nekega embedded sistema. Mnoge univerze in javne institucije ne dovolijo delavcem ali študentom inštalacije dodatnih aplikacij, kar zajema tudi vtičnike in ActivX kontrole.

- Spletni brskalniki ne bodo avtomatsko preusmerjeni na alternativno vsebino v primeru, da Flash predvajalnik ni instaliran. Razvijalec mora sam poskrbeti in ustvariti mehanizme za zaznavo verzije Flash predvajalnika obiskovalcev svoje spletne strani in narediti ustrezne preusmeritve. Flash 8 ima sedaj sicer že opcijo, ki zazna potrebo po osvežitvi in jo je mogoče nastaviti v izvoznih parametrih.
- 3D formate datotek ni mogoče direktno uvoziti ali prikazati v Flash animacijah. Da bi dosegli 3D efekte, je potrebne ustvariti animacije tipa slika-po-sliki. Običajno se neka animacija kreira v 3D razvojnem okolju in se v Flash izvozi kot zaporedje jpg sličic, ki jih Flash predvaja in tako nastane končna 3D animacija. Vsekakor pa ni mogoče manipulirati z nekim 3D objektom. Vendar pa ima Macromedia Director že vgrajeno opcijo za uvoz, ustvarjanje in manipuliranje s 3D objekti.
- Večina iskalnikov (ali spletnih pajkov) imajo probleme z indeksiranjem vsebine Flash animacije. Če je celotna internet stran bazirana na Flash tehnologiji, je potrebno ustvariti neko alternativno HTML vsebino, ki je prijazna za spletne iskalnike. Če se to ne naredi, potem je velika verjetnost, da stran ne bo indeksirana na nobenem spletnem brskalniku. Flash 8 sedaj že omogoča dodajanje meta podatkov v Flash animacijo (.swf). Ti podatki so v obliki, ki jo spletni iskalniki prepoznajo in tako bolje uvrstijo stran pri iskalnih zadetkih, kot bi jo sicer, če teh podatkov sploh ne bi bilo.
- Flash strani niso bile nikoli mišljene kot popolna zamenjava za tekstovno orientirane HTML strani. Za strani, ki vsebujejo veliko tekstovnih informacij z osnovnimi grafičnimi elementi uporaba Flash tehnologije ni priporočljiva. Označevanje in tiskanje vsebine iz Flasha ni vedno tako intuitivno kakor iz standardne HTML strani.



- V veliko primerih je razvoj in izdelava HTML strani hitrejša, lažja in predvsem cenejša kot pa enaka stran s Flash animacijo.
- Velik problem, predvsem pri starejših pa tudi v Flashu 8 leži v tem, da pogon, ki skrbi za izrisovanje animacije na ekranu uporabnika, ne podpira strojnega pospeševanja s strani grafične kartice - GPU-ja (Graphics Processing Unit), ampak uporablja namesto tega osnovni procesor - CPU (Central Processing Unit). Takšen pristop pa povzroča veliko obremenitev tudi za dandanašnje moderne procesorje, kjer se Flash animacije odvijajo z veliko hitrostjo, v njih nastopa veliko objektov z različnimi bitnimi slikami ter se vmes, poleg samega izrisovanja, izvajajo še zahtevne računske operacije. Takšne Flash animacije so zelo nezaželjene, predvsem pa lahko zelo pokvarijo obiskovalčev vtis o neki spletni strani, ki poleg tega, da deluje zelo počasi, obenem 100 % obremeni njegov CPU in s tem delovanje njegovega sistema.

### ***1.3 Uporaba Flash animacije***

Verjetno obstaja več načinov uporabe Flasha kot pa pridevnikov, ki bi jih opisali, vendarle pa bom naštel nekaj primerov uporabe:

- Obrazci, ki temeljijo za zbiranje uporabnikovih informacij in dinamičnem nalaganju vsebine po meri.
- Večuporabniška interakcija na forumu ali spletni strani za podporo z vključitvijo žive avdio/video vsebine vseh udeleženi strani.
- Video predstavitve z uporabo vgrajenih Flash 8 zmožnosti uvažanja in dinamičnega nalaganja vsebine.
- Animirana identifikacijska mesta in nalaganje zaslonov z vgrajeno podporo za zaznavo potrebnega dodatnega prenosa.
- Praktično spletno orodje, kot je računalnik za obresti ali orodje za iskanje.

- Robustna klepetalnica s sobami, ki temelji na XML-ju in server-socket tehnologiji.
- Avdio vmesnik, ki po potrebi nalaga pesmi na zahtevo, s pomočjo uporabe Flash 8 podpore za MP3 predvajanje.
- Interaktivni poskusi, ki vpletajo več uporabnikov s pomočjo avdio/video, snemanje in predvajanje interakcije.
- Nakupovanje in rešitve na področju e-trgov, temelječe popolnoma na Flash in strežniško orientiranih tehnologijah.
- Projektorji oziroma samostojno izvršljive Flash animacije za ustvarjanje diapozitivnih predstavitev v stilu PowerPointa-a, ki so lahko ali na CD-ROM-u ali kateremkoli drugem alternativnem pomnilniškem mediju.
- Predvajanje oglasnih sporočil, risanih animacij, kratkih posnetkov uporabnikov. Predvsem je v zadnjem času opazna porast strani, ki ponujajo prostor in možnost pretakanja video posnetkov v realnem času. Prvi takšen primer je bila stran [www.youtube.com](http://www.youtube.com), kjer lahko kdorkoli objavi video posnetke in jih nemudoma ponudi na ogled celotnemu spletu. Stran [youtube.com](http://youtube.com) zagotavlja vse potrebno, da si lahko na tisoče uporabnikov hkrati ogleduje video vsebine, ki se v živo pretakajo z njihovih strežnikov. Ravno Flash 8 video podpora in na strani uporabnika vsesplošno razširjen Flash 8 predvajalnik, pa sta bila ključna pri hitrem razširjanju te popularne storitve.

Optimizirane animacije spletne animacije, ki jih je mogoče predvajati tudi na prenosnih telefonih in dlačnikih, itd.

Aprila 2007 je izšla najnovejša verzija razvojnega okolja z imenom Adobe Flash CS3 Profesional. Le-ta prinaša nove uporabne funkcionalnosti, kar ga naredi še bolj primerne za ustvarjanje visoko kvalitetnih interaktivnih vsebin. Napredna grafična podoba in animacijska orodja prinašajo večjo prilagodljivost za izražanje še tako zahtevnih in kreativnih idej. Nove funkcije programskega jezika ActionScript 3.0 le še povečajo produktivnost razvojnega okolja, ki se ponaša z izboljšano kompatibilnostjo z ostalimi

grafičnimi orodji podjetja Adobe. Naj naštejemo le nekaj izmed novih izboljšav, ki jih prinaša orodje Adobe Flash CS3 Professional:

### **Možnost direktnega uvoza iz Photoshopa ali Illustratorja**

Flash poleg Photoshopa in Illustratorja predstavlja le eno izmed programskih orodij družine CS3, ki so med seboj dobro povezana. Oblikovalci in razvijalci imajo sedaj proste roke pri izbiri aplikacije s katero najraje ustvarjajo in lahko kljub temu brezkompromisno izkoristijo vse napredne grafične in razvojne funkcije, ki jih ponuja okolje Flash.

Za hitro uvažanje vsebine skrbi nov čarovnik za lepljenje, ki omogoča ohranitev funkcionalnosti in strukture pri enostavnem kopiranju in lepljenju iz Photoshopa ali Illustratorja v Flash. Da bi bila integracija med temi tremi programi še boljša, je možno v Flash CS3 dokument direktno uvoziti Photoshop (psd) in Illustratorjeve (ai) datoteke.

### **Napredna integracija z Adobe Photoshop CS3**

Pri uvozu datoteke tipa PSD v Flash se pojavi okno, ki prikazuje celotno hierarhijo plasti Photoshopove datoteke. Nato imamo možnost izbire posameznih plasti, ki jih želimo uvoziti in njihovih individualnih nastavitev. Tekst iz Photoshopa lahko na primer ohranimo kot tekst, ki ga lahko urejamo še nadalje v Flashu, pretvorimo Photoshop plasti ali skupine plasti v »movie clips« in celo upravljamo z nastavitvami za končno ustvarjanje Flash animacije. Flash lahko nastavi tudi velikost delovne površine projekta tako, da se bo ujemala z velikostjo Photoshop datoteke in obenem omogoča točno postavitev vsebine plasti.

ActionScript omogoča oblikovalcem boljše definiranje premikov posameznih elementov v animacijah in večji nadzor nad njimi. Pisanje kompleksne kode posameznih animacij pa zahteva veliko več časa kot enostavno grafično animiranje. Zato je sedaj v Flash CS3 možno posamezne transformacijske animacije neposredno pretvoriti v ActionScript 3.0 Tako je sedaj možno vizualno animiranje in zatem enostavna transformacija gibov v ponovno uporabljivo kodo, ki jo lahko naknadno urejamo. Preprosto označimo animacijsko transformacijo na časovnici (angl. timeline) in izberemo Edit > Timeline > Copy As

ActionScript 3.0. Ko določimo ime instance, lahko prilepimo programsko kodo v poljubni okvir. Ta funkcija je uporabna tudi za tiste, ki se še nikoli niso srečali z ActionScriptom in jim tako omogoča enostavno spoznavanje z njim. Ta transformacijska funkcija podpira spremembo velikosti, rotacijo, spremembo izometričnega pogleda, barve, filtre in spremembe lastnosti okvirjev.

### **ActionScript 3.0 razvoj z novim razhroščevalnikom**

Flash 8.0 prinaša povsem nov nivo zmogljivosti v razvoju z ActionScript 3.0. Širok programski jezik, ki je primeren tako za začetnike, obenem pa ponuja profesionalno in močno programsko razvojno okolje.

Testiranje vsebine je možno s pomočjo povsem prenovljenega ActionScript razhroščevalnika, ki omogoča pregledovanje korak po korak skozi različne nastavitve in lastnosti v programski kodi v času njenega izvajanja in omogoča visoko stopnjo prilagodljivosti in povratnih informacij. Novi razhroščevalnik je enak tudi za Adobe Flex Builder 3, kar še poveča konsistentnost med tema dvema aplikacijama.

Namesto starega nepreglednega prikaza v enem samem oknu, je novi razhroščevalnik »debugger« sestavljen iz večih prikazov, ki povečajo preglednost in pomikanje skozi programsko kodo. Ko vstopimo v »Debug Mode«, Flash prikaže delovno okolje in zažene SWF datoteko v samostojnem Flash predvajalniku.

### **Adobe Device Central**

Integriran z orodjem Flash CS3 ter prav tako tudi v ostalih orodjih družine Creative Suite 3, je Adobe Device Central, ki poenostavlja izdelavo mobilnih vsebin s pripadajočimi preoblekami mobilnih telefonov in redno osveženimi profili naprav. Sedaj lahko enostavno ustvarimo vsebino, ustvarimo predogled na točno določenem telefonu in testiramo mobilno vsebino, kot so npr. interaktivne aplikacije Flash Lite in bogati uporabniški vmesniki.

Adobe Device Central prikazuje realistične preobleke mobilnih naprav in predogled vsebine prikazane na teh napravah. Mogoča je realna interakcija in simulacija posameznih naprav ter prav tako kontrola nad možnostmi simulacije, ki omogočajo ustvarjanje

različnih scenarijev. Adobe Device Central prinaša s seboj knjižnico naprav med katerimi lahko izbiramo vsako s profilom, ki vsebuje informacije o napravi in vsebini, ki jo podpira. Prav tako lahko iščemo med ponujenimi napravami, jih primerjamo med seboj in ustvarimo sete naprav, ki jih največ uporabljamo.

Adobe Device Central je torej orodje, ki omogoča prikaz Flash vsebin v sliki mobilnega telefona. Poleg možnosti hitrega dostopa do vseh pomembnih tehničnih specifikacij posamezne naprave Adobe Device Center, prav tako na njej simulira sposobnosti prikaza vsebin in tako omogoča natančen predogled ustvarjenih vsebin. Pravzaprav je Adobe Device Central dovolj zmogljiv, da zagotavlja realne časovne povratne informacije ob določenih vsakdanjih scenarijih, kot je situacija ob nizkem stanju baterije ali dohodni klic. Nenazadnje je omogočeno simuliranje tudi različne zasedenosti pomnilnika naprave, osvetlitve zaslona in svetlih zunanjih pogojev.

### **Nove optimizirane komponente z enostavno menjavo preoblek, napisane v ActionScriptu 3.0**

S Flash CS3 lahko ustvarimo interaktivno vsebino s pomočjo optimiziranih komponent, ki so sestavni del uporabniškega vmesnika. Z enostavnim izborom že vgrajenih komponent, napisanih v ActionScriptu 3.0, lahko le-te enostavno povlečemo iz knjižnice in jim enostavno spremenimo videz, ki se ujema z zeleno obliko našega vmesnika. Komponente so še posebej optimizirane, da delujejo hitreje in so prijazne do razvijalcev, ki jih želijo razširiti.



---

## DODATEK B

---

# 1 SPLOŠNO O MACROMEDIA DREAMWEAVER [9]

## *1.1 Adobe Dreamweaver MX 2004*

Za izdelavo spletne strani, oziroma prikaza na samem tekmovanju sem uporabil profesionalno orodje za izdelavo tako statičnih kot dinamičnih spletnih strani, imenovano Dreamweaver MX 2004. Nekoč orodje podjetja Macromedia je sedaj privzelo podjetje Adobe Systems in ga priključilo že zelo pestri paleti orodij za izdelavo, oblikovanje in urejanje večpredstavnostnih (multimedijskih) vsebin. Med mnogimi prednostmi Dreamweaver-ja najbolj izstopa dejstvo, da je bil le-ta prvo orodje za izdelavo spletnih strani, ki je bilo sposobno delati z različnimi web servisi. Ta funkcija omogoča razvijalcem, ki uporabljajo ASP, Colfusion ali JavaServer Pages enake pogoje za delo. Poleg direktne izdelave HTML strani je Dreamweaver primeren tudi za kodiranje širokega spektra spletnih formatov, kot so npr. JavaScript, XML in ActionScript. Dreamweaver omogoča hitro konstruiranje in gradnjo spletnih strani ter poenostavlja njihovo vzdrževanje.

### **Dreamweaver v dinamičnem svetu**

Dreamweaver je aplikacija, ki ima korenine v resničnem svetu. Kot sem že omenil, omogoča izdelavo aplikacij za različne web servise oz. modele. Ker pa se resnični svet venomer spreminja, je Dreamweaverjeva arhitektura zasnovana tako, da na stežaj odpira vrata razvoju vtičnikov za podpiranje praktično kateregakoli web servisa. Še več, Dreamweaver pozna težave nekompatibilnih ukazov in naslovov spletnih brskalnikov, zato generira kodo, ki je kompatibilna med brskalniki. Dreamweaver vključuje HTML validacijo glede na specifični spletni brskalnik, s pomočjo katere je možno videti kako deluje obstoječa ali nova koda v določenem spletnem brskalniku. Da bi bilo delo z orodjem Dreamweaver še bolj prijazno, le-to ob odpiranju spletnih strani avtomatsko ustvari poročilo o strani in morebitnih napakah na njej.

### **Možnosti povezovanja**

Povezovanje je v Dreamweaverju več kot le ključna beseda. Je namreč osnovni koncept. Dreamweaver omogoča povezovanje z večino možnih podatkovnih virov ter s pomočjo široko uporabljenih aplikacijskih strežnikov: ASP, ASP.NET, ColdFusion, PHP in JSP. Dejanski tip povezave je precej fleksibilen, saj se lahko razvijalci odločijo za povezavo, ki jo je lažje implementirati in je sicer manj robustna ali povezavo, ki zahteva več izkušenj na strežniški strani, vendar ponuja velike možnosti skalabilnosti. Dreamweaver ponuja veliko izbiro programskih jezikov za številne aplikacijske strežnike in zbirko že vnaprej pripravljenih CSS standardnih predlog.

Dreamweaver lahko dostopa do standardnih kazalcev na zapise v bazi, podskupin v bazi kot tudi do bolj kompleksnih podatkovnih virov, kot so seje ali programske spremenljivke in shranjene procedure. S pomočjo implementacije piškotkov in strežniško orientirane kode lahko aplikacije, izdelane v razvojnem okolju Dreamweaver, beležijo obiskovalce in jim, če je potrebno, obisk strani tudi onemogočijo.

V aplikaciji Dreamweaver je prav tako možno najti podporo za napredne tehnologije, kot so spletni servisi, JavaBeans in ColdFusion komponente. Dreamweaver omogoča pregled nad elementi vseh tehnologij, ki omogočajo programerjem hitro razumevanje programske sintakse, metod in funkcij.

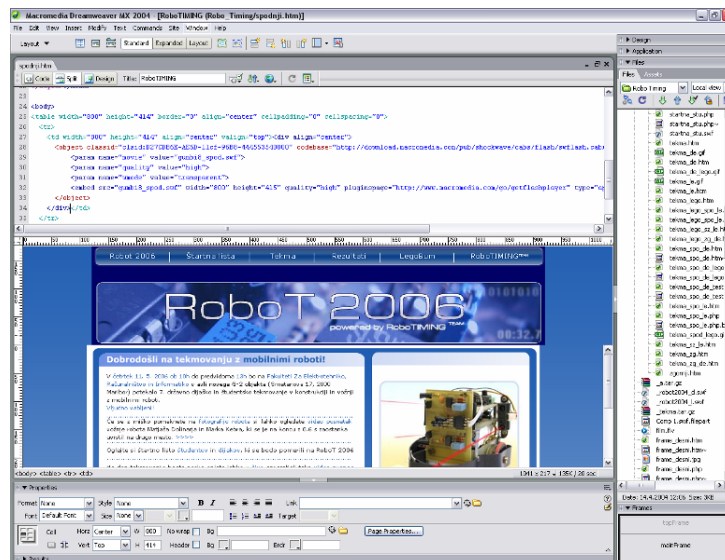


## Predstavitev podatkov „True data representation“

Zares inovativna prednost Dreamweaverja je, da zna integrirati zahtevane podatke s spletno stranjo že v fazi načrtovanja. Dreamweaverjev "Live Data view" pošlje "page-in" proces strežniški aplikaciji, ki okarakterizira zapise iz podatkovnega vira na trenutni strani. Vsi elementi na strani imajo v »Live Data view« možnost urejanja; možno je celo dinamično urejati podatke in takoj videti spremembe, ki so nastale. "Live Data view" skrajša cikel dela s tem, da razvijalcu točno pokaže kaj bo uporabnik videl. Prav tako je možno stran videti pod različnimi pogoji s pomočjo različnih nastavitvev v "Live Data view".

## Integrirani urejevalniki vizualne in tekstovne vsebine

Na samem začetku razvoja svetovnega spleta je večina razvijalcev spletnih strani ročno programirala strani z uporabo enostavnih tekstovnih urejevalnikov, kot je npr. Notepad ali SimpleText. Druga generacija orodij za izdelavo spletnih strani je prinesla s seboj možnost vizualnega oblikovanja ali oblikovanje spletnih strani po principu "What you see is what you get" oz. na kratko "WYSIWYG" (sl. B.1).



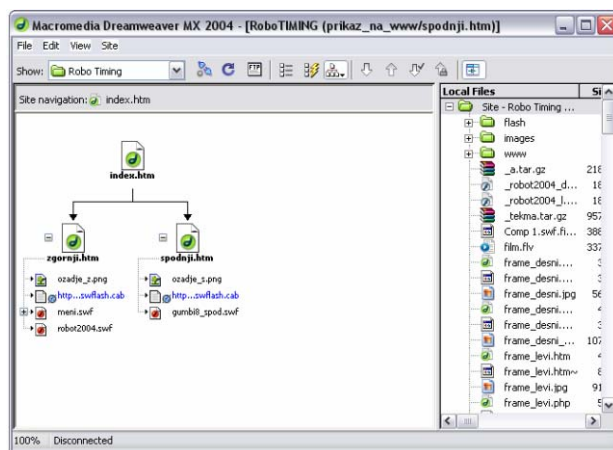
Slika B.1: Dreamweaver omogoča simultano delo v vizualnem WYSIWYG urejevalniku in urejevalniku kode

Ta orodja so sicer zelo dobro rešila problem oblikovanja in zunanjšega izgleda strani, vendar pa niso ponujala neke celovite rešitve na strani programske kode. Profesionalni razvijalci spletnih strani so še vedno bili prisiljeni ročno programirati svoje spletne strani, kljub obstoju relativno naprednih WYSIWYG urejevalnikov. Orodje Dreamweaver vsebuje napreden vizualni urejevalnik, ki ponuja enak pregled nad dokumentom kot spletni brskalnik. Možno je delati tako grafično v oblikovalskem kot programersko v kodirnem pogledu. Obstaja tudi možnost, da se na ekranu prikazujeta oba pogleda hkrati. Slika 5.3 prikazuje skupno delovanje Dreamweaverjevega vizualnega urejevalnika in kodirnega urejevalnika. Vsaka sprememba, ki je narejena v oblikovalskem pogledu, se odraža v kodirnem pogledu in obratno. Možna je uporaba drugih tekstovnih urejevalnikov.

Dreamweaverjeva koda je pregledna in čitljiva ter tudi sintaktično obarvana, kar pa je možno tudi izklopiti.

### Orodja za vzdrževanje spletne strani

Pri Dreamweaverju je razvidno, da so njegovi ustvarjalci razumeli, da je razvoj spletne strani samo ena izmed nalog razvijalca spletnih rešitev. Vzdrževanje je lahko, dolgoročno gledano, časovno precej potratno opravilo, katero pa Dreamweaver s pomočjo lastnih orodij močno olajša in pospeši (sl. B.2).



Slika B.2: Interaktivni zemljevid spletne strani, ki je služila za prikaz rezultatov v živo na spletu, narejen v Dreamweaverju.

Eno izmed teh orodij je npr. knjižnica ponavljajočih se elementov ali pa možnost zaklepanja določenih datotek in s tem omogočanje timskega dela in popravkov. S pomočjo zemljevida strani, ki omogoča takojšen vpogled nad strukturo spletne strani in dostop do katerekoli datoteke za morebitne popravke (slika 5.9). Povezave se avtomatsko osvežijo, če se je npr. datoteka premaknila iz ene mape v drugo ali pa jih je možno tudi ročno popraviti. Poleg uporabe knjižnice s ponavljajočimi se elementi, lahko tudi ustvarimo predloge za nadzor nad izgledom celotne spletne strani in z urejanjem ene predloge posodobimo vse podstrani v našem projektu.

### **Uporabniški vmesnik orodja Dreamweaver**

Njihovi razvijalci ob ustvarjanju spletne strani ponavljajo predvsem dva koraka. Vnašajo elemente, kot so tekst, slike ali plasti in jih potem prilagajajo. Da bi bil proces ustvarjanja spletnih razvijalcev tem bolj tekoč in učinkovit, Dreamweaver vsebuje dobro kombinacijo pomožnih oken in nadzornih plošč, preko katerih je možno hitro in enostavno urejati posamezne parametre elementov.

### **Napredne možnosti urejanja postavitve strani**

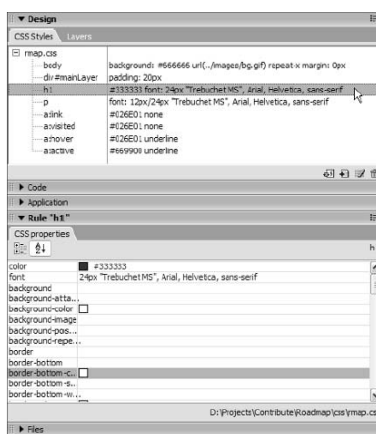
Dreamweaver je za razliko od ostalih vizualnih HTML urejevalnikov močno podoben namizno-založniškem programu. Dandanašnje zmogljivosti spletnih brskalnikov dovoljujejo pozicioniranje teksta in slik na poljubne lokacije na spletni strani. Ta koncept je znan kot »absolutno pozicioniranje". Da bi popolnoma izkoristili moč tega koncepta, je Dreamweaver vključil opcijo ravnal in mrež. Določimo lahko enotno mero (palec, pika ali centimetri) kot tudi razmik in prikaz pomožne mreže. Prav tako je mogoče vklopiti opcijo lepljenja objektov na mrežo in s tem privarčujemo na času, ki bi ga porabili za poravnavo elementov.

## Uporaba najnovejših standardov kodiranja

Večina spletnih strani je narejenih v HTML-ju (HyperText Markup Language). Ta programski jezik je sestavljen iz serije označb, ki določajo tekstovno datoteko in je standardiziran s strani organizacije, imenovane World Wide Web Consortium ali W3C ([www.w3.org](http://www.w3.org)). Vsaka nova verzija HTML-ja združuje napredni set ukazov in funkcij. Dandanes večina spletnih brskalnikov prepozna trenutno aktualno verzijo HTML-ja 4.0 Dreamweaver generira jasno in enostavno, brskalniku prijazno in s standardom HTML 4.01 kompatibilno kodo, kadarkoli vstavimo ali spremenimo nek element v vizualnem urejevalniku.

## CSS podpora

Cascading Style Sheets (CSS) so predloge, ki določajo izgled spletnih strani pri uporabniku. Z njimi določamo tipografijo in velikosti črk ter postavitev na zaslonu. Ker je CSS vedno bolj popularen med spletnimi brskalniki, se je tudi Dreamweaver odločil, da ga podpre. Poleg naprednega izrisovanja v oblikovalskem pogledu za napredne CSS efekte, kot so ozadja, pozicioniranje, je Dreamweaver korenito olajšal izdelavo in urejanje CSS stila (sl. B.3).



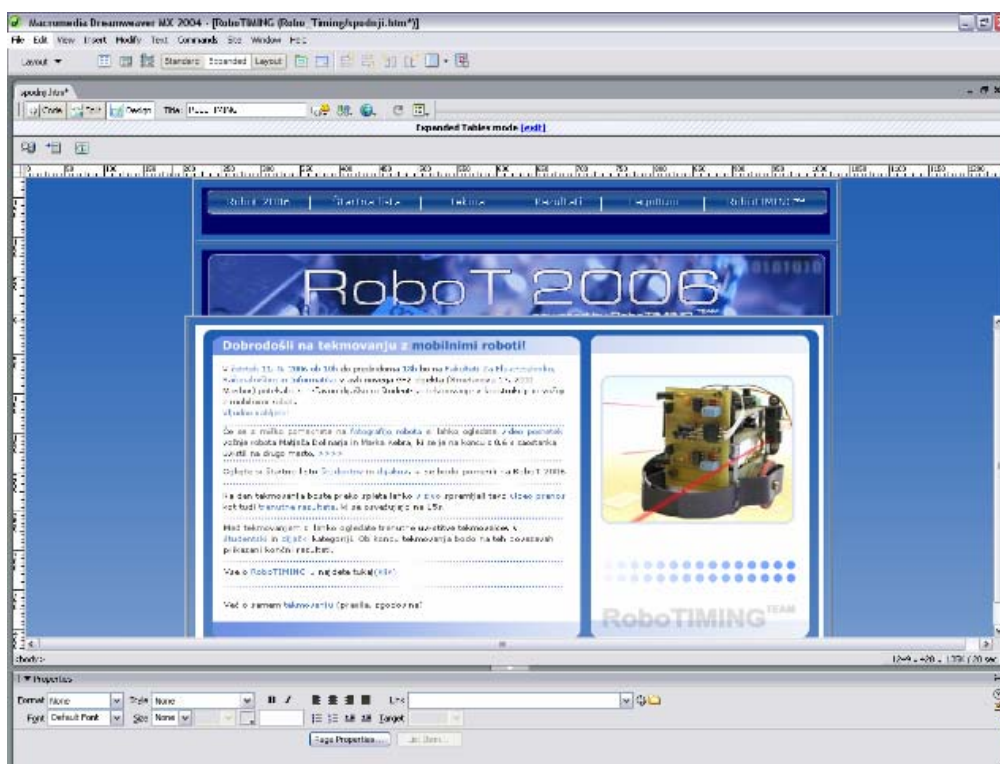
Slika B.3: Urejanje CSS stila deluje s panelom za urejanje oznak s katerim urejamo stile.

CSS stili so lahko aplicirani ali ustvarjeni kar iz panela za urejanje lastnosti. Dreamweaver nima težav tudi z bolj zapletenimi CSS stili in njihovimi lastnostmi, tako da se lahko

oblikovalci spletnih strani bolj osredotočijo na sam izgled strani kot pa na mučenje z raznoraznimi triki v kodi.

## Delo s tabelami

Tabele so ključna komponenta današnjih spletnih strani in Dreamweaver omogoča popolno kontrolo nad njihovo funkcionalnostjo. Dreamweaver je spremenil neprijetno opravilo ustvarjanja tabele z mukotrpnim ročnim kodiranjem, v enostavno »klikni in potegni« rutino. S pomočjo dela v razširjenem načinu (sl. B.4) Dreamweaver omogoča enostaven dostop do elementov.



Slika B.4: V razširjenem načinu dela s tabelami so začasno prikazani robovi s povečano odebelitvijo in razmikom med celicami, kar omogoča veliko lažje urejanje.

Tabele same so zelo fleksibilne in tudi menjava tipa pisave je enostavna in možna za vse celice, vrstice in stolpce. Prav tako pa standardni ukazi omogočajo avtomatsko urejanje in sortiranje tabele.

### **Klikni in potegni princip nastavitve okvirjev na strani**

Okvirji omogočajo, da se lahko hkrati na enem ekranu v različnih okvirjih prikazujejo različne spletne strani. Dreamweaver je to dokaj zahtevno opravilo nastavljanja poenostavil s »klikni in potegni« metodami vlečenja robov okvirjev. Ko nastavimo strukturo okvirjev, jo nato lahko poljubno spreminjamo in Dreamweaver avtomatsko kreira vso potrebno kodo za povezavo vseh HTML strani v posameznih okvirjih, ne glede na to, koliko spletnih strani je uporabljeno.

### **Večpredstavitvene zmožnosti**

Dreamweaver omogoča enostavno vnašanje poljubnega števila večpredstavitvenih razširitev, vtičnikov, apletov ali kontrol. Nastavitve in spreminjanje teh gradnikov so enostavno dostopne preko okna za lastnosti le-teh. Še posebej pa sta podprta gradnika Flash in Shockwave animacije, ki sta bila oba narejena s strani Macromedie. Ob vstavljanju teh dveh objektov, Dreamweaver avtomatsko tudi vključi potrebno HTML kodo, ki zagotavlja široko sprejemljivost s strani spletnih brskalnikov in omogoča urejanje vseh lastnosti. Dreamweaver polno podpira širok spekter multimedijskih vsebin s pomočjo namenskih objektov, ki omogočajo kompleksne slike, avdio in vključevanje ter prikazovanje predstavitev, prikazanih direktno na spletni strani. Dreamweaver tudi izkorišča prednosti novega Flash formata, imenovanega »prevedena« (angl. compiled) Flash animacija, z vključevanjem prvega v seriji Flash elementov. Ta element je pregledovalnik slik. Kot ostali Flash elementi, je pregledovalnik slik Flash animacija, ki omogoča nastavitve atributov znotraj Dreamweaverja, kar omogoča široko paleto oblikovalskih možnosti in ponastavitvev.

### **XML [10]**

XML je okrajšava za angleški izraz „Extensible Markup Language“, razširljiv označevalni jezik, in je jezik, ki ga pogosto srečate, če brskate po Internetu. XML je preprost računalniški jezik, podoben HTML-ju, ki nam omogoča format za opisovanje strukturiranih podatkov ali arhitektura za prenos podatkov in njihovo izmenjavo med več

omrežji. XML spreminja mnogo aspektov računalništva, še posebej na področju komuniciranja aplikacij in strežnikov. Da se ga tudi razširijo, saj ima namreč to možnost, da si lahko sami izmislimo imena etiket (angleško TAG). Zelo je uporaben za komunikacije, saj ima zelo preprosto in pregledno zgradbo. Dreamweaver je sposoben izvažanja in uvažanja XML etiket, ne glede na definicijo etikete. Prav tako je možno ustvariti, spremeniti ali ovrednotiti XML datoteke.

### **3D plasti**

Dreamweaver se zaradi lastnosti, imenovane "Dynamic HTML" oziroma dinamični HTML, ponaša tudi kot prvo 3D orodje za izdelavo spletnih strani. Pred dinamičnim HTML-jem so strani bile grajene na dveh ravninah - sliko in tekst je bilo možno postaviti samo drug ob drugega. Dreamweaver podpira absolutno postavitev s CSS-om, čemu pravimo tudi plasti, kar pomeni, da je objekte možno postaviti pred ali za nek drug objekt. Plasti lahko vsebujejo tekst, grafiko, povezave in ukaze. Imamo celo možnost ugnezdjenja ene plasti v drugo.

Skozi vedno večjo prevlado CSS-a na spletu, vse več spletnih oblikovalcev vključuje lastnosti absolutnega pozicioniranja direktno v <div> etiketo. Ta proces ohranja lastnosti plasti (pravila CSS-a) ločene od vsebine in omogoča veliko enostavnejše vzdrževanje. Dreamweaver ponuja enostavno urejanje in uveljavljanje <div> etiket .

Drug način kreiranja plasti v Dreamweaverju je mogoč s pomočjo gumba Nariši plast, ki se nahaja na orodni vrstici, imenovani Vstavi. Dreamweaver vstavi potreben CSS, poravnani s plastjo (<div>) etiketo. Ko so ustvarjene, lahko umestimo plasti kjerkoli na strani s klikanjem in vlečenjem. Tako kot tudi ostale objekte v Dreamweaverju, lahko tudi plasti urejamo v "Property inspector«.

### **Dinamično osveževanje stilov**

Dreamweaver popolnoma podpira CSS specifikacije, ki so bilo potrjene s strani World Wide Web združenja. Dreamweaver je sposoben delati s CSS-om kot poteka urejanje stilov v urejevalniku teksta. Na primer, lahko naredimo vse etikete <h1> modre, položne in

samo malih črk. Če se spremeni shema barv na spletni strani, je mogoče vse etikete <h1> narediti rdeče in to dosežemo s pomočjo samo enega ukaza. Dreamweaver omogoča nadzor nad stili ozadja, blokov, okvirjev, seznamov in pozicioniranja.

Dreamweaver omogoča spreminjanje stilov s povezavo (online) ali brez povezave (offline). S povezovanjem CSS spremembe ob neki uporabniški akciji, kot je npr. premikanje kazalca miške, se lahko tekst poudari in obratno, območja na ekranu se lahko zasvetijo ali pa se animira neka slika. Prav tako pa je lahko vse narejeno brez neprestanega povezovanja s strežnikom ali velikega prenosa datotek.

### **JavaScript ukazi**

Dreamweaver združuje moč JavaScripta z enostavnim »označi-in-klikni« vmesnikom. JavaScript ukaz je definiran kot kombinacija dogodka in akcije oziroma, ko uporabnik spletne strani naredi nekaj, kar povzroči nek drug dogodek. Ti ukazi so zelo uporabni, saj ne zahtevajo nobenega programiranja. Bazirajo na JavaScriptu, kar je zelo pomembno, saj je le-ta široko podprt med spletnimi brskalniki. Dreamweaver je poenostavil opravilo prepoznavanja kateri JavaScript ukaz deluje z določenim spletnim brskalnikom. Enostavno označimo element na spletni strani, ki bi ga radi uporabili za upravljanje z neko akcijo in odpremo orodno vrstico z ukazi. Kot je to prikazano na sliki 5.12, Dreamweaver omogoča izbiro JavaScript ukaza, ki deluje z vsemi ali točno določenim spletnim brskalnikom. Nato iz seznama izberemo akcije, ki so na voljo, kot so »pojdi na spletno stran«, »zaigraj zvok«, »odpri sporočilo v novem oknu« ali »začni animacijo«. Možno je določiti več akcij ter tudi to, kdaj se bodo zgodile.

### **Orodja za vzdrževanje**

Dreamweaver združuje več orodij, ki spletnemu programerju prihranijo veliko časa. Standardna praksa je namreč, da je še nekaj časa za tem, ko je stran že objavljena na spletu, še vedno potrebno le-to obnavljati in popravljati določene malenkosti. Poleg objavljanja preko FTP-ja, ki je že standardna funkcija Dreamweaverja, se lahko le-ta pohvali z novostjo, imenovano Knjižnice. Eno izmed očitnih dejstev je, da kakor hitro imamo



element, ki se ponavlja na naši strani, se pogosto zgodi, da ga moramo urediti ali spremeniti na vsaki strani posebej. S pomočjo knjižnice v Dreamweaverju se znebimo tega časovno potratnega opravila. V knjižnico lahko dodamo praktično katerikoli element na strani: odstavek, tekst, slika, hiperpovezava, tabela, obrazec, ActivX kontrolo in tako naprej. Potem ko smo dodali element v knjižnico, ga lahko tekom izdelave spletne strani in tudi kasneje poljubno krat uporabimo. Vsaka spletna stran lahko ima svojo knjižnico in prav tako je možno prenesti elemente iz ene knjižnice v drugo.

### **Predloge**

Dreamweaver omogoča uporabo predlog za spletne strani, s čimer se poenoti izgled spletne strani ter čas, ki je potreben za ponavljajoče delo - izdelave nove strani. Dreamweaverjeva predloga lahko hrani osnovne podatke o strukturi strani, sliko v ozadju, navigacijsko vrstico na levi strani ali na primer mere tabele za hranjenje glavnega teksta. Dreamweaverjeve tabele so več kot le šablone za izdelavo strani. Predloge so sestavljene iz več zaklenjenih in odklenjenih področij za urejanje. Da bi posodobili celotno spletno stran, ki sloni na eni predlogi, je potrebno le spremeniti področje, ki je zaklenjeno. Prav tako je pogojno mogoče skriti ali prikazati področja z opcijskimi področji. Dreamweaverjev podpira tudi ugnezdene predloge, tako da se spremembe lahko prenesejo skozi več zaklenjenih in odklenjenih področij.

### **Preverjanje hiperpovezav**

Spletne strani se vseskozi spreminjajo in danes je postal pravi izziv kontrolirati in preverjati veljavnost vseh hiperpovezav na strani. Dreamweaver že ima vgrajeno preverjanje hiperpovezav, ki hitro prikaže, katere povezave so prekinjene, katere datoteke imajo hiperpovezave do zunanjih strani in katere datoteke so osamljene oziroma se na njih ne navezuje nobena hiperpovezava.

### **FTP objava**

Končni korak pri izdelavi spletne strani je njena objava na internetu. Kot je marsikomu, ki je delal spletne strani znano, je ta končni korak tisti, ki se večkrat ponavlja, ko se stran osvežuje in naknadno ureja. Dreamweaver ima vgrajen FTP klient in celoten sistem, ki poenostavlja delo in objavljane spletne strani. Še bolj pomembno pa je, da Dreamweaver omogoča sinhronizacijo med spletno stranjo, ki je na lokalnem disku in spletno stranjo, ki je objavljena na spletu. Ni nujno, da se čisto vse datoteke spletne strani, ki se nahajajo na lokalnem disku, vedno posodobijo ob sinhronizaciji z oddaljeno spletno stranjo na strežniku. Dreamweaver vsebuje funkcijo, imenovano »cloaking«, ki omogoča označevanje map, katere se ne osvežujejo med sinhronizacijo. Dreamweaverjeva orodna vrstica z datotekami omogoča označevanje več datotek in njihovo prenašanje na oddaljen strežnik ter po potrebi tudi nazaj.

Najnovejša verzija orodja Dreamweaver je izšla aprila leta 2007 in se imenuje Adobe Dreamweaver CS3. Nekaj novih funkcij v orodju Dreamweaver CS3:

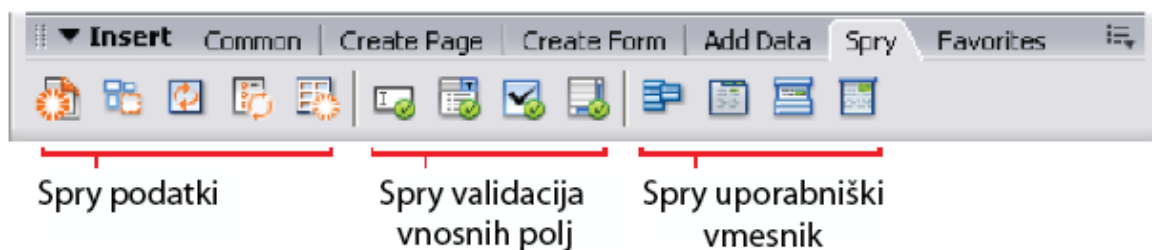
### **Spry framework za Ajax**

Ajax, kratica stoji za Asinhroni JavaScript in XML, je razvojna tehnika za ustvarjanje interaktivnih spletnih aplikacij. Ajax daje spletnim stranem občutek večje odzivnosti, saj se celotna stran ne rabi osvežiti vsakokrat, ko poteče interakcija med stranjo in uporabnikom. Rezultat je povečanje interaktivnosti, hitrosti in uporabnosti spletnih strani.

Z Dreamweaverjem CS3 lahko vizualno oblikujemo, razvijamo in namestimo dinamične uporabniške vmesnike s pomočjo Spry framework za Ajax. Spry framework za Ajax predstavlja JavaScript knjižnico za oblikovalce spletnih strani, ki jim omogoča izgradnjo strani, katere nudijo obiskovalcem bogato uporabniško izkušnjo. Za razliko od ostalih Ajax frameworkov je Spry dosegljiv tako oblikovalcem kot tudi razvijalcem, saj je 99% Sprya pravzaprav koda, napisana v HTML-ju.

## Spry zavihki

S pomočjo Dreamweaverja CS3 lahko spletnim stranem dodamo najpogosteje uporabljene komponente uporabniškega vmesnika s pomočjo Spry zavihkov. Spry zavihki so že narejene komponente za ustvarjanje najbolj pogostih elementov uporabniškega vmesnika, ki jih lahko preoblikujemo s pomočjo CSS in nato vključimo v spletno stran. Vključujejo XML sezname in tabele, uporabniške vmesnike z zavihki ter obrazce s preverjanjem posameznih polj. Do sedaj je bilo te gradnike težko zgraditi, njihova izgradnja pa je zahtevala poglobljene programerske sposobnosti. Spry zavihki (sl. B.5) so zelo fleksibilni in so primerni tako za manj zahtevne uporabnike kot tudi tiste, ki jih želijo razširiti in prilagoditi.



Slika B.5: Spry zavihki

So enostaven in eleganten način izboljšave izgleda in uporabniške izkušnje spletne strani. S pomočjo Spry efektov lahko dodamo efekte praktično vsakemu elementu na HTML strani. Efekti za delovanje ne zahtevajo kakršnekoli strežniške logike ali skript. Ko obiskovalec spletne strani, napisane v HTML-ju, sproži efekt, se osveži samo objekt, na katerega se nanaša efekt in osveževanje celotne strani ni potrebno.

S pomočjo Spry efektov lahko dosežemo, da se določeni elementi povečajo, skrčijo, počasi izginejo, poudarijo in še več. Na primer, efekt lahko poudari informacije, ustvari animiran prehod ali vizualno zamenja element na strani za določen čas.

Spry efekti omogočajo enostavno ustvarjanje prehodov in animacij, brez da bi bilo potrebno napisati eno samo vrstico kode.

**Adobe CSS Svetovalec**

Dandanes se tehnologije menjavajo z veliko hitrostjo, kar pomeni potrebo po nenehnem spremljanju težav, ki jih imajo spletni brskalniki ter eventualnih možnih obličih. Ravno zato je Dreamweaver pričel ponujati povezave na novo spletno stran, poimenovano Adobe CSS Svetovalec, kjer ponuja rešitve in namige za težave, ki jih je odkril med procesom preverjanja kompatibilnosti na posameznih spletnih brskalnikih. Adobe CSS Svetovalec vsebuje informacije o zadnjih sporih, ponuja uporabne namige in predloge ter je namenjen in vzdrževan s strani skupnosti. Uporabniki lahko dodajo svoje CSS spore, jih komentirajo in ocenjujejo. CSS Svetovalec omogoča enostavno komentiranje s predlogi in izboljšavami k obstoječim vsebinam ali dodajanje novih sporov za dobrobit celotne razvojne skupnosti. Adobe CSS Svetovalec oziroma spletna stran, na kateri se nahaja ta skupnost, vabi k izmenjavi znanja in izkušenj o CSS, s skupnostjo, usmerjeno v razvijanje spletnih strani.

---

## DODATEK C

---

### **1 SPLOŠNO O ADOBE PHOTOSHOP CS2 [11]**

Razvoj metod manipulacije digitalnih slik je popolnoma spremenilo industrijo fotografske produkcije. Veliko revolucijo je prineslo retuširanje slik. Predvsem pa se je zmanjšal čas in potrebno znanje, ki je potrebno za izvršitev določenih opravil, povezanih z obdelavo slik. Z novimi orodji so lahko tudi amaterji v kratkem času dosegli to, za kar so se prej profesionalni fotografi trudili ure in ure. Prav prihod digitalne manipulacije slik je ogromno prispeval k napredku sveta fotografije s tem, ko je omogočil manipulacije, ki so bile prej zelo težke ali praktično nemogoče. Nenazadnje so digitalne manipulacije omogočile praktično neuničljivost slik ter enostavno reverzibilnost sprememb, ki so se vršile na slikah. Ravno Photoshop pa je bil odgovoren za ogromno inovacij, ki so se zgodile na tem področju.

## 1.1 *Kaj je Photoshop in kaj omogoča?*



Slika C.1: Adobe Photoshop - orodje namenjeno oblikovanju

Adobe Photoshop ali preprosto Photoshop je profesionalni program za obdelavo fotografij (sl. C.1) in druge grafike, razvit in izdan s strani podjetja Adobe Systems. Trenutno je vodilno orodje na trgu za obdelavo bitnih slik in njihovo manipulacijo. Poleg Adobe Acrobat je Photoshop tudi najbolj poznano programsko orodje, ki ga je izdelalo podjetje Adobe Systems. Postal je industrijski standard v večini opravil, povezanih z uporabo vizualnih elementov. Photoshop je na voljo za okolje Microsoft Windows, Mac OS X in Mac OS. Prav tako lahko s pomočjo programa, kot je npr. CrossOver poganjamo Photoshop do verzije 9 tudi na operacijskem sistemu Linux.

Čeprav je bil Photoshop prvotno načrtovan za urejanje slik za namizno založništvo, se dandanes vedno bolj uporablja za ustvarjanje slik namenjenih za spletne strani. V zadnjih različicah je k osnovnemu programu priložena tudi aplikacija Adobe ImageReady, ki ponuja zelo specializirano paleto orodij. Photoshop ima močne vezi z ostalimi programskimi paketi podjetja Adobe, ki služijo za urejanje multimedijskih vsebin, animacij in razvoja aplikacij. Photoshop zna delati z številnimi različnimi modeli barv, kot so: RGB, Lab in CMYK, sivine, rastrske slike in »duotone« [12].

V zadnjih trinajstih letih, kolikor je Adobe Photoshop prisoten na tržišču, se je ta, sprva relativno preprost program, razvil v zelo kompleksno in profesionalno orodje za digitalno manipulacijo slik. Photoshop 9.0 oziroma CS2 vsebuje zares ogromno funkcij, od katerih pa naj naštejemo samo nekaj.

### **Seti plasti: (Layer Sets)**

Omogočajo organizacijo ali kategorizacija pogostih ali kakorkoli povezanih plasti skozi orodno vrstico »plasti«. Da bi ustvarili nov set plasti moramo najprej razširiti meni »layer« in označiti New -> Layer set. Odpre se okno »novi set plasti«, ki nas poziva k vnosu imena za nov set. Ko kliknemo OK, se pojavi ikona z obliko mape, kamor lahko potem preprosto vnašamo že obstoječe ali nove plasti in s tem močno povečamo preglednost nad poplavo plasti, obenem pa je možno nad setom plasti uveljavljati stil, kar lahko v praksi prihrani ogromno časa.

### **Orodje za izdelavo rezin (Slice tool)**

Orodje se nahaja v glavni pokončni orodni vrstici in je označeno z majhnim nožkom. Orodje za izdelavo rezin je namenjeno izdelavi pravokotnih, na klik občutljivih območij na neki sliki. Photoshop omogoča dodajanje akcije, ko se na območju pojavi miškin kazalec ali ko je le-to kliknjeno in je zelo uporaben pri ustvarjanju dinamičnih slik ali za ustvarjanje menijev. Območju na klik lahko zelo preprosto dodamo hiperpovezavo ali katero drugo informacijo. V rezine razdeljene slike na straneh se hitreje naložijo, kot bi se stran sicer, če bi bila slika v enem kosu.

### **Spletna foto galerija**

Funkcijo, ki prihrani veliko časa, je mogoče najti pod File->Automate-> Web Photo Gallery in služi hitremu ustvarjanju visoko kvalitetne galerije slik v HTML obliki. Preprosto vnesemo izvorno in ciljno mapo s slikami ter pustimo Photoshopu, da opravi ostalo. Le-ta ustvari toliko spletnih strani z malimi sličicami, kot je le to potrebno, da bi zajel vse slike v izvorni mapi. Prav tako je možno vnesti dodatne parametre, kot so naslov

strani in »ALT" tekst. Prav tako omogoča, da se ob kliku vsake male sličice pojavi na strani slika v polni velikosti.

### **Prilepi v**

Do njega dostopamo preko menija Edit. Ta funkcija vstavi sliko v neko drugo sliko in obreže notranjo sliko, ki je tako videti kot da je znotraj zunanje slike. Na primer, če bi narisali okrogli kvadrat 100 pik širok in 20 pik visok in nato narisali še en zaokrožen kvadrat 40 pik širok in 20 pik visok direktno na vrhu njega, bi lahko potem s pomočjo orodja »Magic Wand tool«, označili območje znotraj manjšega kvadrata in prilepili fotografijo v označeno območje. Tako bo vidno samo območje malega kvadrata. V povezavi s tekstom ta funkcija omogoča odlično osnovo za izdelavo naslova spletne strani.

### **Stili**

Orodna vrstica s stili je repozitorij prednastavljenih efektov. Če npr. narišemo lik in izberemo enega izmed stilov, se bo skupek vseh prednastavljenih lastnosti in efektov, ki jih predstavlja stil, avtomatsko preneslo nanj. Prav tako imamo na izbiro veliko število stilov, namenjenih spletnemu oblikovanju vključno z gumbi in »roll-overs«, ki nam lahko prihranijo ogromno časa pri izdelavi menijev in obrazcev.

### **Skripte**

Uporaba funkcije skript v Photoshopu je orodje, ki nam lahko prihrani veliko časa. In sicer je namen skript avtomatiziranje ponavljajočih se opravil, kot je npr. izvažanje vseh plasti našega dokumenta v PDF obliko z namenom, da ustvarimo predstavitev. Photoshop prav tako vsebuje svoj razhroščevalnik skript »script debugger« in omogoča pisanje skript v programskem jeziku Visual Basic in JavaScript. Photoshop ima že v osnovi vgrajenih veliko skript do katerih lahko dostopamo preko File -> Scripts ->Browse. Orodje za pisanje skript je podrobneje opisano v številnih PDF priročnikih v Photoshop CS mapi.



### **Orodna vrstica s plastmi**

Brez nje si je Photoshop kar težko predstavljati. Privzeto je locirana na spodnjem delu delovne površine - skrajno desno in vsebuje seznam vseh plasti ter nekatera orodja za urejanje plasti. Le-te lahko skrijemo ali prikažemo ter spremenimo njihovo prosojnost. Te lastnosti so zelo uporabne pri izdelavi animacij in slikovnih zamenjav. Prav tako pa se največji pomen orodne vrstice s plastmi kaže skozi možnostjo dodajanja efektov posameznim plastem.

### **Efekti**

So zelo podobni stilom v tem, da lahko preko njih direktno spremenimo vizualni izgled posamezne plasti. Največja razlika med efekti in stili pa je v tem, da so efekti vezani na vsebino plasti. To pomeni, da ko spremenimo plast, se z njo spremeni tudi efekt. Na voljo imamo tudi izbor že prednastavljenih efektov od senčenja, obrob do gradientov ali polnila z vzorci.

### **Shrani za splet**


Ta funkcija je uporabna za optimizacijo slik, ki so namenjene za objavo na internetu. V kombinaciji z nastavitvami na desni strani je mogoče doseči kompromis med kvaliteto posamezne slike in podatkovno velikostjo. To nam lahko prihrani dragocene sekunde pri nalaganju te slike na spletni strani. Zmanjševanje tako imenovanega filtra "dither", ki služi za ustvarjanje iluzije barvne globine pri omejeni barvni paleti ter povečevanje filtra "web snap" s katerim se lahko popolnoma približamo za splet varni paleti barv, zmanjšamo velikost datoteke. Le-to dosežemo tudi z zmanjševanjem količine barv ali povečevanjem parametra "lossy", ki ob svoji maksimalni vrednosti odstrani največjo možno količino detajlov iz posamezne slike.

Na tem mestu naj omenim še nekaj najbolj naprednih funkcij, ki jih s seboj prinaša Adobe Photoshop CS3.

## **Pametni Filtri**

Pametni filtri so preprosto Photoshop filtri, ki jih nanašamo brez da bi pri tem uničili original ter ostajajo živi in jih je možno urejati. Tak pristop nam omogoča fleksibilnost pri eksperimentiranju z različnimi filtri in nastavitvami, brez da bi se bilo potrebno vračati nazaj in začeti z novo originalno sliko. Možno je urejati masko filtra tako, da natančno prilagodimo učinke filtra na različnih področjih slike. Še več, filtri se nanašajo znotraj plasti objektov, kar pomeni, da je mogoče ustvarjeno grafiko enostavno povečevati ali zmanjševati, preoblikovati in pri tem spreminjati nastavitve filtrov na posameznih delih slike, brez da bi pri tem uničili oziroma spremenili original.

## **Orodje za hitro označevanje [13]**

Photoshop je že od nekdanj slovel po svojih močnih orodjih za označevanje, ki omogočajo urejanje točno določenih delov slike. Photoshop CS3 precej skrajša čas, ki je potreben za označevanje točno določenih delov slike s pomočjo orodja »Quick Selection« . Sedaj namesto počasnega in zahtevnega označevanja zunanjih robov slike, ki ji želimo urejati, enostavno pobarvamo ozadje slike. Photoshop CS3 nas opazuje med barvanjem in analizira sliko, da bi označevanje dokončal avtomatsko. Povečevanje ali krčenje označenega je zelo preprosto in kar je še več, orodje Quick Selection lahko uporabimo v povezavi z vsemi ostalimi orodji za označevanje v Photoshopu.

## **Redefiniranje robov**

Končni pristop k vsakem označevanju slike predstavlja novo orodje za definiranje robov »Refine Edge«, ki je enostavno in fleksibilno ter omogoča popoln nadzor nad robovi slike preden jo pričnemo urejati. Orodje omogoča natančno glajene, mehčanje, krčenje ali razširitev trenutne označitve slike s pomočjo drsnih kontrol.

---

## DODATEK D

---

### **BPMN (Business Process Modeling Notation)**

#### **Splošno o BPMN**

Je grafična notacija za modeliranje poslovnih procesov in delovnih tokov. Nastala je pod okriljem organizacije BPMI (Business Process Management Initiative), v letu 2005/2006 pa je prešla v okrilje skupine OMG (Object Management Group), ki je predhodno botrovala tudi nastanku jezika UML. Razlog za to združitev lahko po eni strani vidimo kot poskus poenotenja notacij za modeliranje procesov, po drugi strani pa kot poskus za večjo prepoznavnost notacije »pod velikim bratom«. Trenutno so v okviru skupine OMG v fazi sprejemanja poleg zadnje različice notacije BPMN tudi komplementarni standardi BMM in SVBR (Semantics Of Business Vocabulary and Business Rules) in BPDM (Business Process Definition Metamodel). Za razliko od BPMN, zadnja dva zmetka standardov nudita le poenoten slovar pojmov, ki nastopajo v poslovnih procesih, ne pa grafične notacije. Osnovni namen BPM notacije je, da jo razumejo vsi poslovni uporabniki, poslovni analitiki, ki načrtujejo poslovni proces, tehnično podkovani razvijalci, odgovorni za implementacijo izvajanja procesov ter poslovni uporabniki, ki procese upravljajo in nadzorujejo.

## Uporaba notacije

BPMN se uporablja za opis internih poslovnih procesov, prav tako tudi za opis medorganizacijskih poslovnih procesov. Opisi procesov v notaciji BPMN nam lahko služijo kot:

- del poslovnika kakovosti (ISO 9001:2000),
- delovna navodila za zaposlene,
- osnova za informacijski sistem.

Specifikacija notacije poleg grafičnih simbolov definira tudi preslikavo diagrama procesa v izvršljiv jezik – XPDŁ ali BPEŁ, ki temeljita na XML. Tako lahko formalna definicija procesa služi kot vhod v procesni strežnik. Nenazadnje se s tem zmanjšuje prepad med načrtovanim in implementacijo poslovnih procesov.



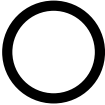









## Sestava notacije

BPMN sestavlja množica enostavnih grafičnih elementov oz. elementov za opredelitev toka dogodkov:

- dogodek,
- aktivnost,
- prehod,
- povezovalni elementi,
- steze,
- informacijski objekti (dokumenti).

## Elementi za opredelitev toka dogodkov – Dogodek

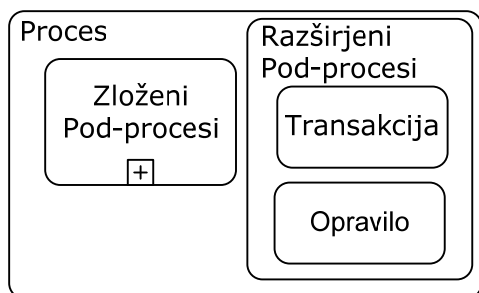
Dogodek je nekaj, kar se zgodi med potekom poslovnega procesa. Dogodki so krogi s praznim osrednjim delom, kjer je prostor za dodatne oznake. Razlikujemo različne sprožilce in posledice. Primer: »Elektronsko sporočilo prejeto«, »ura je 15:00«, »trgovina prazna«, »Kritična napaka«,....

Potek dogodka	Začetni	Vmesni	Končni	Opis
Splošni				Začetni dogodek označuje začetek določenega procesa. Vmesnik dogodek se zgodi med začetnim in končnim dogodkom. Čeprav ima vpliv na potek procesa pa le-tega ne začinja ali ga (direktno) končuje. Končni dogodek označuje konec procesa.
Sporočilo				Sporočilo prispe s strani udeleženca in sproži dogodek. To sproži proces, ki začne, nadaljuje ali se konča, če je čakal na sporočilo ali pa spremeni potek dogodkov, če pride do izjeme. Tip končnega sporočila ponazarja, da je bilo sporočilo ob zaključku procesa poslano določenemu udeležencu.
Časovnik				Lahko določimo točen čas, ali cikel, ki bo sprožil začetek procesa ali ga nadaljeval. Vmesniki časovnik se uporablja za modeliranje časovnih zakasnitev.
Napaka				Ta zaključek ponazarja, da se mora zgoditi točno določena napaka. To napako bo zaznal vmesniki dogodek znotraj konteksta dogodkov.
Preklic				Ta tip dogodka je uporabljen znotraj transakcijskega pod-procesa in mora biti pripet na mejo/rob pod-procesa. Sproži se v primeru ko je dosežen dogodek preklic znotraj transakcijskega pod-procesa.

Kompenzacija				Uporablja se za izvajanje kompenzacije – tako za njeno pripravljanje kot izvajanje. Kompenzacija se kliče v primeru da je dogodek del normalnega poteka. Reagira na določen kompenzacijski klic, če je pripet na mejo/rob aktivnosti. Kompenzacijski dogodek je zelo uporaben za modeliranje razveljavitvenih akcij znotraj transakcije.
Pravilo				Ta tip dogodka se sproži, ko so izpolnjeni pogoji za določeno pravilo. Pravila so lahko zelo uporabna za prekinitev procesa v zanki, kot npr. »Število ponovitev = N«. Vmesno pravilo se uporablja izključno za delo z napakami.
Povezava				Povezava je mehanizem za povezovanje konca (rezultata) procesa z začetkom (proženjem) drugega procesa. Po navadi sta to dva pod-procesa znotraj istega nadrejenega procesa. Povezava se lahko uporablja npr. ko je delovna površina (stran) premajhna – pojdi na drugo stran.
Večkratni				Ta tip dogodka označuje, da obstaja več načinov proženja procesa. Samo eden izmed njih bo potreben za {začetek, nadaljevanje, konec} procesa.
Konec				Ta tip zaključka ponazarja, da se morajo vse aktivnosti znotraj procesa nemudoma zaključiti. Vključno z vsemi primerki »Multi-Instances«. Proces je zaključen brez kompenzacije ali upravljanja z dogodki.

## Aktivnost

Je splošen termin za delo, ki se izvaja. Aktivnost je lahko atomarna (opravilo) ali sestavljena (iz procesov in pod-procesov). Primeri: »Pošlji pismo«, »Napiši poročilo«, »Izračunaj število interesentov«...









Opravilo se uporablja za predstavitev aktivnosti na najnižjem abstraktnem nivoju.

Opravilo/posebni atributi pod-podprocesov		
Ponavljajoče se		Opravilo ali pod-proces se ponavlja.
Ad Hoc		Opravila znotraj pod-procesa ne morejo biti povezana z zaporedjem poteka v času načrtovanja.
Več instanc		Ustvarjenih bo več instanc opravila ali pod-opravila
Kompenzacija		Simbol predstavlja kompenzacijsko opravilo ali pod-opravilo.

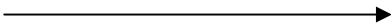
## Prehod

Prehod se uporablja za ločevanje ali združevanje potekov več procesov. Opredeljuje osnovne odločitve kot tudi razcepe in združevanja poti. Oznake znotraj simbola opredeljujejo tip nadzora nad delovanjem. Primeri: »Pogoj pravilen? –ja/ne«, »Izberi barvo? – rdeča/zelena/modra«, ...

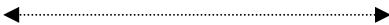
Tipi prehodov		
XOR (PODATKOVNI)	 	Odločitve, ki temeljijo izključno na podatkih ali združevanju. Oba simbola imata enak pomen. Glej tudi pogojni potek.
XOR (DOGODKOVNI)		
OR		Dogodek, ki temelji na podatkih, odločitvah ali kombinaciji obeh.
KOM- PLEKSNO		Kompleksni pogoji (kombinacija osnovnih pogojev)
AND		Paralelno razdruževanje in združevanje (sinhronizacija).

### Grafično povezovalni elementi:

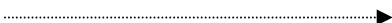
Poznamo tri načine povezovanj objektov (dogodkov, aktivnosti, prehodov), medsebojno ali z drugimi informacijami z uporabo toka zaporedja, toka sporočil ali asociacij.

Tok zaporedja 

Uporablja se za predstavitev zaporedja aktivnosti, ki se v procesu izvaja.

Tok sporočil 

Uporablja se za prikaz toka sporočil med dvema akterjema (za predstavitev akterjev uporabljamo različne steze) v procesu.

Asociacija 

Uporablja se za predstavitev asociacije podatkov, teksta in ostalih izdelkov z elementi za opredelitev toka dogodkov. Asociacija se uporablja za prikaz vhodov in izhodov iz aktivnosti.



**Steze:**

Predstavljajo mehanizem za organizacijo aktivnosti v vizualno ločene kategorije. S pomočjo tega lahko prikažemo različne funkcionalne zmožnosti in odgovornosti. Primeri: »Fakulteta«, »Prodajni oddelek«, »Nakupovalni center«, »ERP sistem«,...

<b>Steza</b>	Podsteza
	Podsteza

Steza mora vsebovati nič ali en poslovni proces.

Steza lahko vsebuje nič ali več podstev.



Steza predstavlja udeleženca v procesu. Vsebuje poslovni proces in je uporabljena v B2B primerih.

Dve stezi sta lahko povezani medsebojno samo preko toka sporočil.

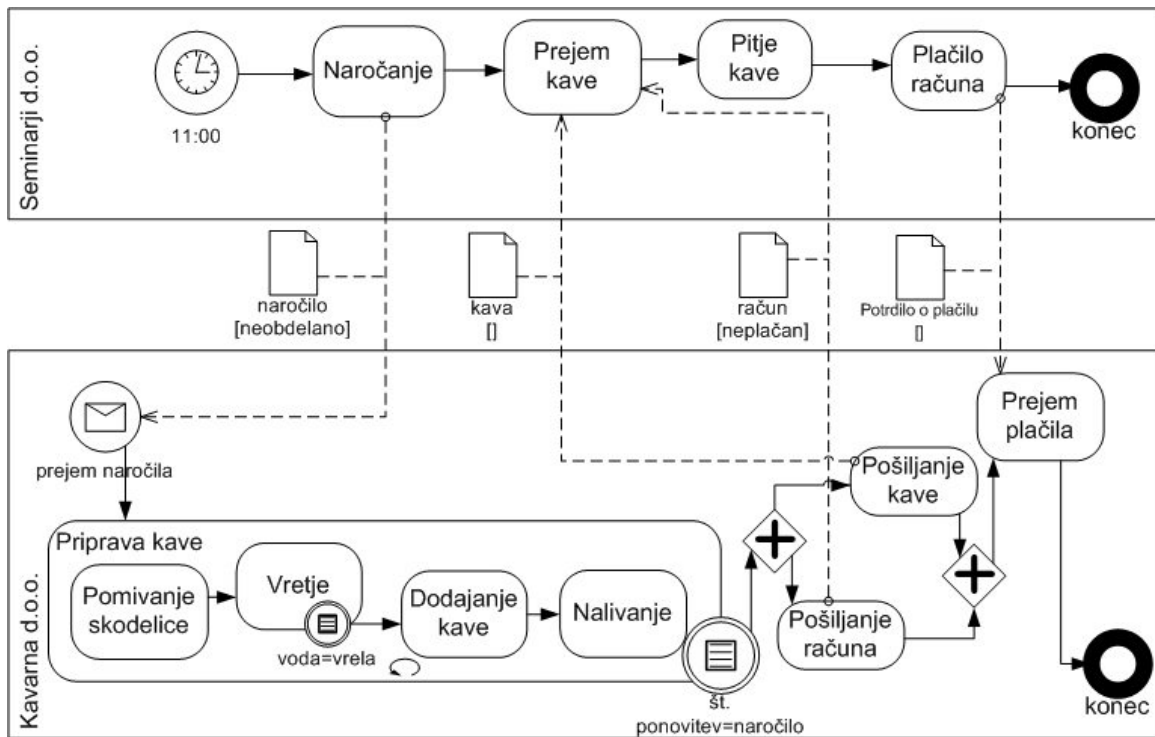
Podsteza je podrazdelek znotraj steze. Njen namen je organizirati in kategorizirati aktivnosti.

**Informacijski objekti (dokumenti)**

Njihov namen je zagotavljati dodatne informacije o procesu. Če je potrebno, lahko načrtovalec doda tudi svoje nove objekte. Primeri podatkovnih objektov: »Pismo«, »Elektronsko sporočilo«, »XML dokument«, »Potrdilo«, ...

Standardizirani podatkovni objekti		
Podatkovni objekt	 [stanje]	Podatkovni objekti zagotavljajo informacije o tem katere aktivnosti je potrebno sprožiti in/ali kaj je njihov produkt. Jemljemo jih kot objekte, ker nimajo nikakršnega direktnega vpliva na zaporedje poteka ali toka sporočil v samem procesu. Stanje podatkovnega objekta mora biti nastavljeno.
Skupina		Grupiranje se lahko uporablja za dokumentacijo in namene analize. Skupine se prav tako lahko uporabi za prepoznavanje aktivnosti distribuiranih transakcij, ki so prikazane na stezah. Grupiranje aktivnosti ne vpliva na zaporedje poteka ali tok sporočil.
Opomba	[Opis]	Tekstovne opombe so mehanizem za načrtovalce, da zagotovijo bralcu BPMN diagrama dodatne informacije.

Primer - Proces kuhanja kave:



## Literatura

- [1] Pravilnik za izvedbo študentskega in dijaškega tekmovanja z mobilnimi roboti RoboT 200x, [na spletu] [http://www.ro.feri.uni-mb.si/tekma/PravilnikRo\\_02.htm](http://www.ro.feri.uni-mb.si/tekma/PravilnikRo_02.htm), [dostop: 30.08.2008]
- [2] Vec novosti o tekmovanjih, [na spletu], [http://www.ro.feri.uni-mb.si/tekma/vec\\_novosti.htm](http://www.ro.feri.uni-mb.si/tekma/vec_novosti.htm), [dostop: 30.08.2008]
- [3] Using Macromedia Flash as an embedded device UI, [na spletu], <http://www.windowsfordevices.com/articles/AT2364930496.html>, [dostop: 30.08.2008]
- [4] Adobe - Showcase : Adobe Success Story : CorAccess, [na spletu], [http://www.adobe.com/cfusion/showcase/index.cfm?event=casestudydetail&loc=en\\_us&casestudyid=3164](http://www.adobe.com/cfusion/showcase/index.cfm?event=casestudydetail&loc=en_us&casestudyid=3164), [dostop: 30.08.2008]
- [5] Rich Internet application - Wikipedia, the free encyclopedia, [na spletu], [http://en.wikipedia.org/wiki/Rich\\_Internet\\_application](http://en.wikipedia.org/wiki/Rich_Internet_application), [dostop: 30.08.2008]
- [6] ITPoster.net, [na spletu], <http://www.itposter.net/itPosters/bpmn/bpmn.htm>, [dostop: 30.08.2008]
- [7] Robert Reinhardt and Snow Dowd, John Wiley & Sons 2006, *Macromedia Flash 8 Bible*
- [8] Slashdot | Adobe Opens the FLV and SWF Formats, [na spletu], <http://tech.slashdot.org/article.pl?sid=08/05/01/0628254&from=rss>, [dostop: 30.08.2008]
- [9] Joseph W. Lowery, Wiley Publishing, *Dreamweaver MX 2004 Bible*
- [10] XML - Wikipedija, prosta enciklopedija, [na spletu], <http://sl.wikipedia.org/wiki/XML>, [dostop: 30.08.2008]
- [11] Adobe Photoshop - Wikipedia, the free encyclopedia, [na spletu], [http://en.wikipedia.org/wiki/Adobe\\_Photoshop](http://en.wikipedia.org/wiki/Adobe_Photoshop), [dostop: 30.08.2008]

- [12] 20 Things About Photoshop You Have to Know, [na spletu], <http://www.devaricles.com/c/a/Photoshop/20-Things-About-Photoshop-You-Have-to-Know/>, [dostop: 30.08.2008]
- [13] Deke McClelland, Laurie Ulrich Fuller and Laurie Ulrich Fuller, John Wiley & Sons © 2005, *Photoshop CS2 Bible*

## Življenjepis:

Ime in priimek:	Marko Keber
Rojen:	29. september 1981
Osnovna šola:	1988 – 1996 OŠ Savo Kladnik, Sevnica
Srednja Šola:	1996 – 2000 Gimnazija Celje Center
Fakulteta:	2000 – 2008 Feri Maribor Program: elektrotehnika Smer: avtomatika
Zaposlitev:	2008 - Mikrospin d.o.o.





UNIVERZA V MARIBORU

**IZJAVA O ISTOVETNOSTI TISKANE IN ELEKTRONSKE VERZIJE  
DIPLOMSKE NALOGE**

Ime in priimek študenta: MARKO KEBER  
Številka indeksa: 93483439  
Študijski program: ELEKTROTEHNIKA - AVTOMATIKA  
Naslov diplomskega dela: ROBOTIMING  
Mentor: DOC. DR. ALES HACE  
Somentor: DOC. DR. MARTIN TERBUC

Podpisani MARKO KEBER se strinjam, da se moja diplomska naloga objavi na portalu Digitalne knjižnice Univerze v Mariboru.

Izjavljam, da sem diplomsko nalogo izdelal sam ob pomoči mentorja in nisem kršil avtorskih pravic in intelektualne lastnine drugih. Tiskana verzija diplomske naloge je istovetna elektronski verziji, ki sem jo oddal v Digitalno knjižnico Univerze v Mariboru.

Podpis študenta: