

**UNIVERZA V LJUBLJANI
FAKULTETA ZA DRUŽBENE VEDE**

DUŠAN MARKOVIČ

Mentor: doc. dr. ANDREJ MRVAR

Somentor: doc. dr. ANTON KRAMBERGER

**IZDELAVA KONCEPTA IN IZVORNE KODE
SPLETNE TRGOVINE ZA MALA PODJETJA**

DIPLOMSKO DELO

LJUBLJANA, 2003

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Dušan Markovič, z vpisno številko 2101823, rojen 22.4.1959 v Ljubljani, EMŠO 2204959500373, sem avtor diplomskega dela z naslovom:

Izdelava koncepta in izvirne kode spletne trgovine za mala podjetja

S svojim podpisom zagotavljam, da:

- je predloženo diplomsko delo izključno rezultat mojega lastnega raziskovalnega dela;
- sem poskrbel, da so dela in mnenja drugih avtorjev oz. avtoric, ki jih uporabljam v predloženem delu, navedena oz. citirana v skladu s fakultetnimi navodili;
- sem poskrbel, da so vsa dela in mnenja drugih avtorjev oz. avtoric navedena v seznamu virov, ki je sestavni element predloženega dela in je zapisan v skladu s fakultetnimi navodili;
- sem pridobil vsa dovoljenja za uporabo avtorskih del, ki so v celoti prenesena v predloženo delo in sem to tudi jasno zapisal v predloženem delu;
- se zavedam, da je plagiatorstvo - predstavljanje tujih del, bodisi v obliki citata bodisi v obliki skoraj dobesednega parafraziranja bodisi v grafični obliki, s katerim so tuje misli oz. ideje predstavljene kot moje lastne - kaznivo po zakonu (Zakon o avtorstvu in sorodnih pravicah, Uradni list RS št. 21/95), prekršek pa podleže tudi ukrepom Fakultete za družbene vede v skladu z njenimi pravili;
- se zavedam posledic, ki jih dokazano plagiatorstvo lahko predstavlja za predloženo delo in za moj status na Fakulteti za družbene vede.

V Ljubljani, dne: 18.9.2003

Podpis avtorja:
Dušan Markovič

KAZALO:

UVOD	4
1. Internetni nakupi v Sloveniji, EU in ZDA	6
1.1. Uporabniki in raba interneta	6
1.1.1. Časovna distanca.....	7
1.1.2. Gospodinjstva	9
1.1.3. Podjetja	13
1.1.3.1. Podjetja s spletnimi stranmi oz. število spletnih strani na 1000 prebivalcev.....	16
1.1.3.2. Podjetja s prodajo preko spletnih strani	18
1.2. Nakupi on-line	18
1.3. Digitalni razkorak	19
1.4. Značilnosti nakupov.....	20
1.5. Želje uporabnikov - varnost	21
1.6. Zaključek	24
1.7. Sklep	27
2. Industrijska organizacija in e-poslovanje	30
2.1. Uvod.....	30
2.2. Industrijska organizacija in klasično podjetje.....	30
2.3. Klasično podjetje in razvoj vertikalnih struktur	37
2.4. Virtualno podjetje kot ekstremna oblika spletne trgovine - sistemska analiza in primerjava poslovnih funkcij virtualnega in klasičnega podjetja ..	40
2.5. e-Commerce	43
2.5.1. Definicije e-Commerce	43
2.5.2. e-trgovina in e-trgovanje	45
2.5.3. Integracija informacijskih sistemov.....	46
2.5.4. Sklep	50
3. Značilnosti interneta kot okolja za delovanje spletnih aplikacij.....	52
3.1. Uvod.....	52
3.2. TCP/IP, HTTP in koncept strežnik-odjemalec	52
3.3. Skripti na uporabniški in strežniški strani internetne povezave	57
3.3.1. Posebnosti skriptov na strežniški strani internetne povezave	58
3.3.2. Raznolikost jezikov na strežniški strani internetne povezave	60
3.3.2. Baze podatkov in SQL	61
3.3.3. Izbira skriptnega jezika na strežniški strani internetne povezave ...	62
3.3.4. Hramba vrednosti spremenljivk med spletnimi dokumenti	63
3.3.5. Zaključek.....	64
4. Analiza programa	66
4.1. Katalog, zgradba in delovanje procedur.....	68
4.2. Košarica: zgradba in medsebojno delovanje procedur.....	72
5. PRILOGA:	81
5.1. scirpti.htm	81
5.2. trg1.asp.....	89
5.3. trg2.asp.....	93
VIRI:	94

UVOD

Internet je kot splošen komunikacijski medij še posebej pomemben za politiko in javno upravo. Omogoča namreč in tudi zahteva nove in drugačne interakcije med državljanom in državnim aparatom, kar postavlja pred javno upravo nove izzive. V ta namen je potrebno konceptualizirati in omogočiti nove načine sodelovanja. Osnovna računalniška pismenost za vse to enostavno ni dovolj. Ta predpostavlja namreč le znanje uporabe računalnika kot več ali manj učinkovitega pisalnega stroja in v zvezi s tem obvladovanje urejevalnikov teksta oz. tabelarnih programov. Tako kot osnovna računalniška pismenost zahteva znanje uporabe urejevalnikov teksta, tako poznavanje interneta kot medija zahteva v osnovi poznavanje osnovnih programskih oziroma jezikovnih orodij. Samó poznavanje osnovnih programskih jezikov namreč omogoča konceptualizacijo potreb in ustrezno sodelovanje z programerji, ki lahko spremenijo zahteve v programe brez tiste "To se pa ne da!" ugotovitve. Seveda je komunikacija tako učinkovitejša, hitrejša in tudi cenejša. Toliko o pomenu naloge.

Osnovni namen te naloge je torej v praksi na primeru preizkusiti poznavanje osnov jezika HTML in skriptnih programskih jezikov na uporabniški in strežniški strani internetne povezave, kar bi moralo pripeljati do delujočega programa. Ker gre za preizkušanje osnovnega znanja programskih jezikov, je do neke mere vseeno, kaj naj bi program počel, pod pogojem, da se uporabijo vsa tista sredstva, ki se tudi sicer uporabljajo na spletnih straneh političnih subjektov in državne uprave. Tu mislim predvsem na uporabo strežniških skriptov, skriptov na uporabniški strani (client side script), piškotkov (cookies) in baze podatkov. Naloga je zato razdeljena na naslednje vsebinske sklope:

- a. Pregled raziskave Raba interneta v Sloveniji za leto 2002 in maj 2003 in primerjava s trendi v ZDA in EU s poudarkom na spletnih nakupih in splošnih značilnostih nakupovanja preko interneta. Tu ugotavljam, ali uporaba interneta kot orodja za nakup narašča in ali se vzporedno integrira internetna prodaja v poslovno okolje malih podjetij. Na podlagi ugotovitev, pričakovanj in nakupovalnih navad slovenskih in tujih

uporabnikov interneta, ki na internetu tudi kupujejo, bomo zastavili v četrtem vsebinskem sklopu koncept spletne trgovine.

- b. Internet in siceršnja informacijska struktura podjetja z vidika industrijske organizacije.
 - c. Značilnosti internetnega programskega okolja. Posebnosti le tega s poudarkom na arhitekturi odjemalec-strežnik. Značilnosti programskega jezika HTML in protokola HTTP, ter omejitve le teh (doseg oziroma veljavnost spremenljivk na straneh HTML). Primerjava skriptnih jezikov Perla, PHP, Jscripta, Java Scripta in VBScripta z ASP-jem, osnove SQL. Omejitve samega poizkusa glede na razpoložljivo strežniško okolje. Na osnovi tega vsebinskega sklopa bomo ugotovili možnosti, s katerimi razpolagamo pri uresničevanju pričakovanj internetnih potrošnikov.
 - d. Analiza in zgraditev osnovne strukture programa na osnovi zgornjih ugotovitev in poteka programskega toka (Program flow), medsebojno delovanje funkcij in osnovnih spletnih strani.
 - e. Program oziroma sama zgradba funkcij kot priloga.
- Ti vsebinski sklopi se pokrivajo tudi z osnovnimi poglavji diplome.

1. Internetni nakupi v Sloveniji, EU in ZDA

1.1. Uporabniki in raba interneta

Ne glede na to, da nas zanimajo predvsem podatki o nakupovalnih navadah slovenskih internetnih uporabnikov, se nekaterim splošnim kazalcem uporabe interneta ne moremo izogniti. Primerjali jih bomo z primernimi kazalci v EU in ZDA in pri tem iskali trende.

Raba interneta se lahko opazuje z dveh strani in sicer z uporabniške strani in s strani ponudnikov vsebin. Ker gre v primeru te naloge za mala podjetja, je potrebno ugotavljati vsaj rabo interneta v podjetjih. Ta pa ima spet številne sestavine: od odnosov s strankami, vpetosti interneta in intraneta v samo informacijsko strukturo podjetja (naročanje, prodaja, deljene baze ipd.), do same prodaje in spletne predstavitve kot osnovnega minimuma. Za potrebe te naloge se bom omejil predvsem na spletne predstavitve.

Med splošnimi kazalci bom pozornost usmeril predvsem na naslednje:

- Število osebnih računalnikov na prebivalstvo oz. na gospodinjstvo,
- Dostop do interneta od doma oz. z delovnega mesta.

Med za nalogo specifičnimi kazalci pa bomo opazovali naslednje kazalce o prodaji preko interneta:

- Število uporabnikov interneta, ki so v tekočem letu opravili vsaj en nakup,
- Odstotek prometa, ki se opravi preko interneta v razmerju s celotnim prometom v državi,
- Glavni vzroki za nenakupovanje preko interneta.

V nalogo bomo vključili tudi specifične kazalce, ki sicer zaradi omejenosti dostopa do primerljivih raziskav v tujini niso na splošno na razpolago, v Sloveniji pa so dostopni. Ne glede na nezmožnost primerjave pa imajo ti podatki močno pojasnjevalno vrednost, in so zanimivi sami zase.

Primerjali bomo izsledke raziskav RIS za Slovenijo, Eurobarometra za EU in Poročilo Ministrstva za Trgovino ZDA, urada za popis, sektorja službe za statistiko¹. Slednjega bomo uporabili za prikaz odstotka prometa, ki se opravi preko interneta (Vehovar Lobe, 2003: 4). Druge kazalce za ZDA bomo pridobili iz raziskav Organizacije TNS², ki ima v 53 uradih širom sveta 9000 zaposlenih in opazuje po istih kazalcih približno 110 držav. V svojih raziskavah uporablja organizacija po državah reprezentančen vzorec prebivalstva, ki ga anketira preko anketarjev ali pa s telefonsko anketo, podobno kot pri nas raziskovalci RIS.

Potrebno je opozoriti na metodološke razlike pri pridobivanju podatkov. Medtem ko raziskava Raba interneta v Sloveniji temelji na telefonski anketi prebivalstva Slovenije in pri tem upošteva reprezentativni vzorec več kot 2000 anketirancev, temelji recimo ameriška raziskava Ministrstva za Trgovino na reprezentativnem vzorcu 10.000 podjetij, ki je selekcioniran iz MRTS³ vzorca, pri tem je uporabljena stratificirana, naključno - vzorčna metoda vzorčenja. Kljub temu pa menim, da so podatki med seboj primerljivi, saj so primerljivi izidi raziskav, kljub temu, da so načini pridobivanja podatkov različni. Ne primerjamo torej med seboj neprimerljivih količin in podatkov. Vendar pa sami raziskovalci RIS-a ugotavljajo sistematično zaostajanje RIS-ove ankete za anketo Eurobarometra, ki znaša v posameznih segmentih tudi do 5%.

1.1.1. Časovna distanca

Za sprejemanje odločitev glede koncepta spletne trgovine sami podatki o računalnikih in številu le-teh niso primarnega značaja. Bolj je pomembna analiza trendov oz. smeri razvoja spletnega poslovanja in dela s strankami glede na računalniško infrastrukturo, ki je trenutno na uporabniški strani. Da bi lažje ugotavljali predpostavljeni bodoči obseg poslovanja in trgovanja preko internetnih povezav, ki bo večje, če bo več računalnikov na število prebivalcev države, je koristno vpeljati pojem časovne distance.

Pri tej se kot spremenljivka pojavi čas v odvisnosti od nekega kazalca. Tako opazujemo časovni razmik med dvema časovnima točkama, pri katerih doseže kazalec isto vrednost. Če to poenostavimo na primeru dvodimenzionalnega grafa,

¹ United States Department of Commerce, US Census Bureau, Service Sector Statistics.

² Taylor Nelson Sofres.

³ Monthly Retail Trade Survey - Mesečna raziskava trgovine na drobno.

potem predstavlja os y vrednosti opazovane spremenljivke, os x pa predstavlja čas. V grafikon vnašamo vsaj dve ali več primerjalnih serij. Pri analizi časovne distance penetracije računalnikov v Sloveniji se bom naslonil na analizo, ki jo je v letu 2002 opravil dr. Pavle Sicherl za MID (Ministrstvo za informacijsko družbo).

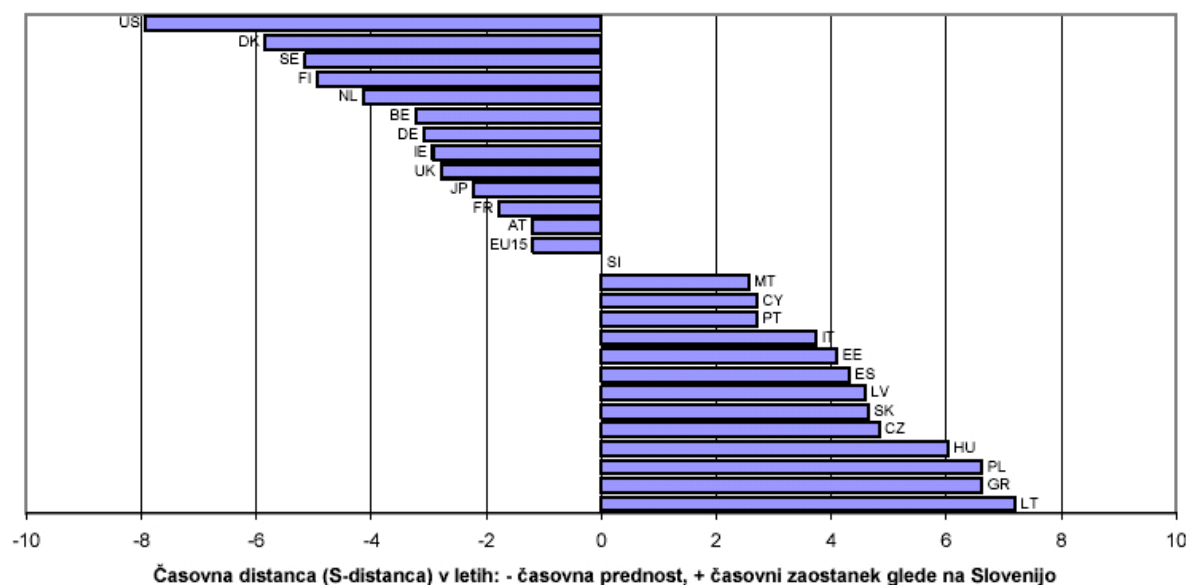
S časovno distanco povezan pojem je S-distanca, ki jo Sicherl opredeli takole:

"S-distanca predstavlja specialno kategorijo pojma časovna distanca, ki je povezana z nivojem analiziranega indikatorja. Predlagana statistična mera S-distanca meri v časovnih enotah razdaljo (bližino) med dvema točkama v času, ko dve primerjalni seriji dosežeta neko specificirano raven indikatorja X."

(Sicherl, 2001: 1)

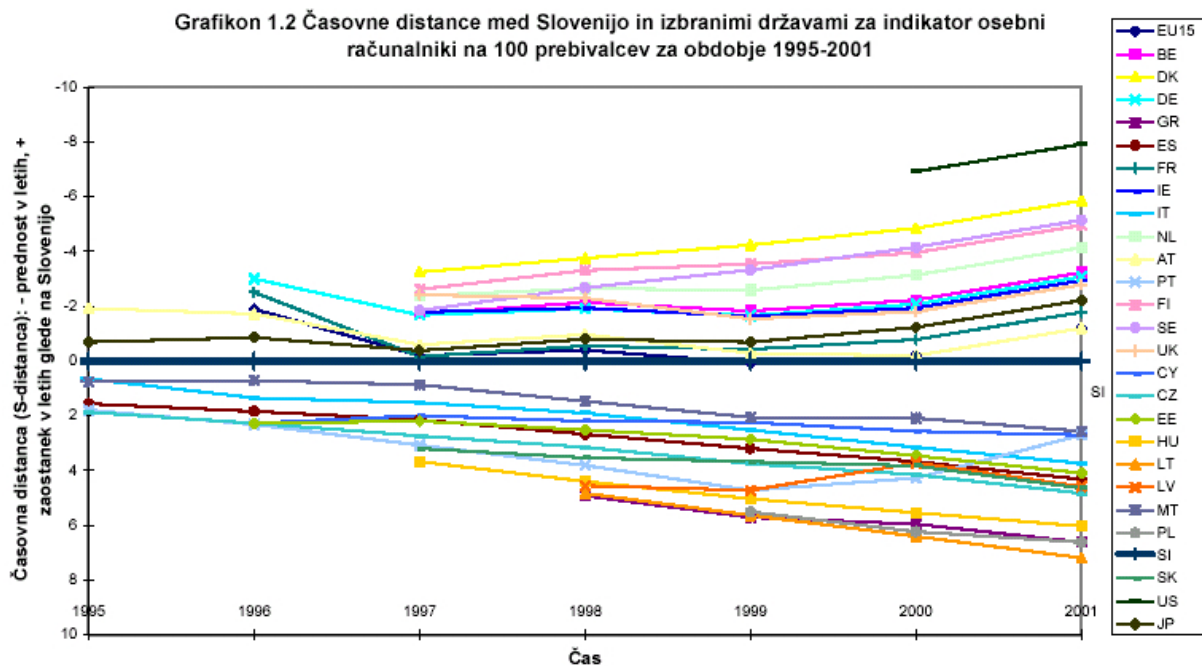
V našem primeru (Sicherl, 2002: 5) se kot indikator opazuje število računalnikov na 100 prebivalcev.

Grafikon 1.1 Časovna distanca med Slovenijo in izbranimi državami za osebne računalnike na 100 prebivalcev 2001



Grafikon 1.1: S-distanca med Slovenijo in izbranimi državami za leto 2001

Iz grafikona 1.1, na katerem je za osnovo vzeta vrednost za Slovenijo, vidimo, da je ta v letu 2001 zaostajala za povprečjem držav članic EU za približno 1 leto oziroma nekaj več, za ZDA (US) pa približno 8 let. Opazovan indikator je v tem primeru razširjenost računalnikov na 100 prebivalcev. Zanimivo pa je, da je iz grafikona moč razbrati, da je časovna distanca v EU zelo velika in da Slovenija prehiteva številne države predvsem na jugu EU. V tem grafikonu (grafikon 1.1) je možno opazovati trenutno stanje, trendi, ki so pripeljali do tega stanja oziroma do te časovne distance, pa so razvidni iz grafikona 1.2 (Sicherl, 2002: 7).



Grafikon 1.2: Časovna distanca med Slovenijo, EU in ZDA za obdobje 1995-2001

Vidimo lahko, da se časovna distanca na splošno povečuje⁴ (Slovenija je os x). Razlike v letu 2001 so med posameznimi opazovanimi državami na splošno večje kot v letu 1995. Zanimivo je tudi, da se je do leta 1997 Slovenija približevala povprečju EU, kasneje pa se je od njega odmikala. Na grafu se zato odmika EU, ker so vrednosti na osi y za Slovenijo vedno 0.

1.1.2. Gospodinjstva

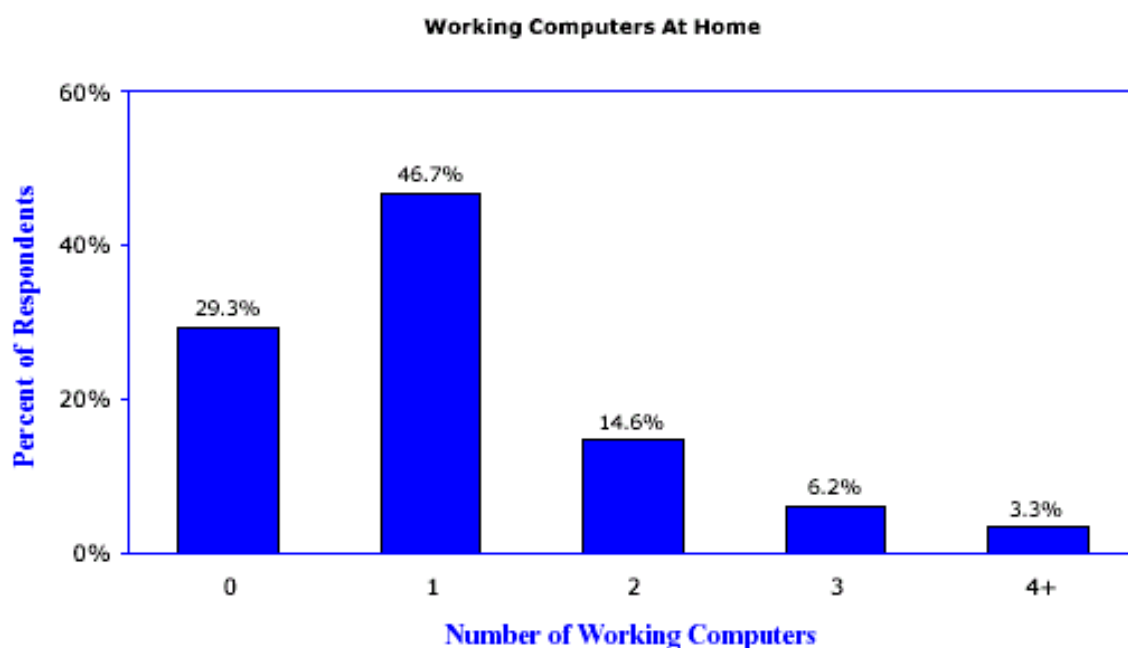
Glede na to, da je namen te naloge izdelati spletno trgovino za mala podjetja in da naj bi preko te trgovine potekala prodaja predvsem končnemu uporabniku, kar pomeni, da gre za B2C⁵ model trgovine, lahko "upravičeno" predpostavljamo, da morebitni kupec ne bo kupoval preko interneta s pomočjo službenega računalnika. Kupoval naj bi od doma, zato nas v začetku zanimajo predvsem gospodinjstva. Ugotoviti želimo najprej, koliko potencialnih kupcev lahko preko internetne trgovine dosežemo. Teoretično vse, ki imajo doma računalnik. Vendar pa se izkaže, da imajo nekateri sicer računalnik doma, pa ga za dostop do interneta ne uporabljajo.

⁴ Slovenija je na grafu os x oziroma ničelna točka s katere se meri oddaljenost.

⁵ Business too Consumer oziroma svobodno prevedeno podjetje - uporabnik.

Verjetno pa lahko na le-te računamo kot na bodoče "surfarje" in potencialne kupce in ob tem ugotavljamo, kaj jih ovira, da že do sedaj niso začeli brskati po internetnem omrežju. Ugotovitve:

Po raziskavi Poročilo o digitalni prihodnosti - v nadaljevanju UCLA report (Lebo, 2003: 26) - je živel 67,8% prebivalcev ZDA v letu 2002 v gospodinjstvu, ki poseduje računalnik (2002 Scarborough report, 2002: slide 23) oziroma je v tem letu imelo doma delujoč računalnik 70,7% Američanov. Raziskava je potekala na vzorcu 2000 ameriških gospodinjstev od aprila do junija 2002.



Grafikon 1.3: Osebni računalniki po gospodinjstvih v ZDA

Po Anketi RIS2002 Informacijske tehnologije v gospodinjstvih (Vehovar in Remec, 2003: 6), ki je bila izvedena približno v istem času kot UCLA report⁶, ima osebni računalnik 58,3% gospodinjstev. Anketi sta med seboj primerljivi, le da je slovenski vzorec veliko bolj reprezentativen. V primerjavi z ZDA predstavlja to v odstotkih približno 12,4% manj.

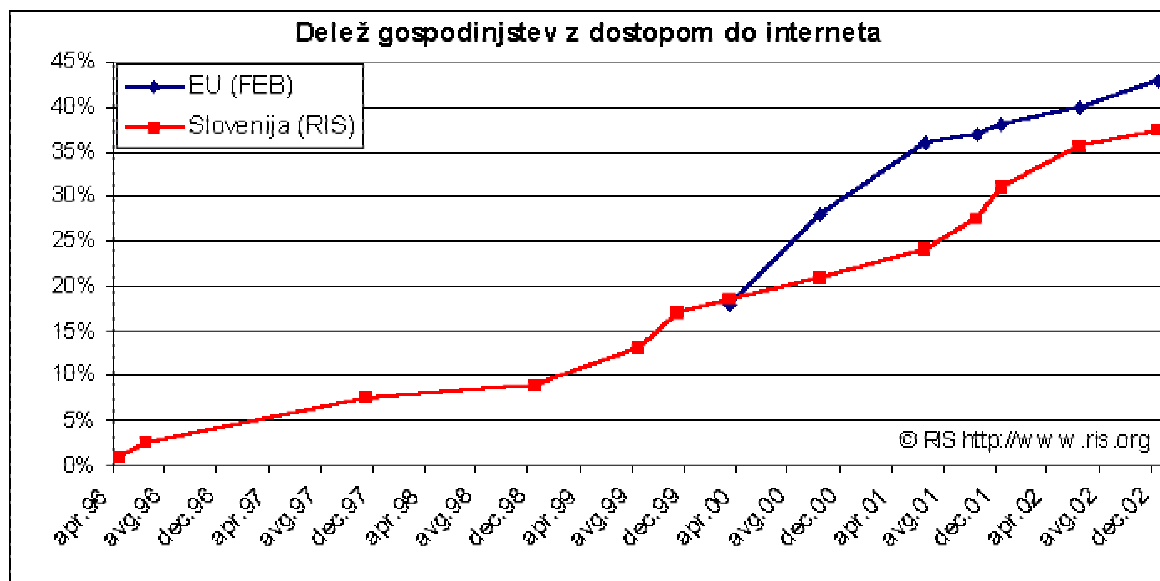
V istem obdobju naj bi imelo dostop do računalnika doma 57% Evropejcev⁷.

Ob tem pa do interneta od doma dostopa v tem času približno 37% Slovencev, v aprilu 2003 se je ta delež povečal na 38%. V primerjavi z EU znaša zaostanek 4% in je relativno stabilen (Vehovar in Kragelj, 2003: 2). Raziskava je bila narejena

⁶ junija 2002.

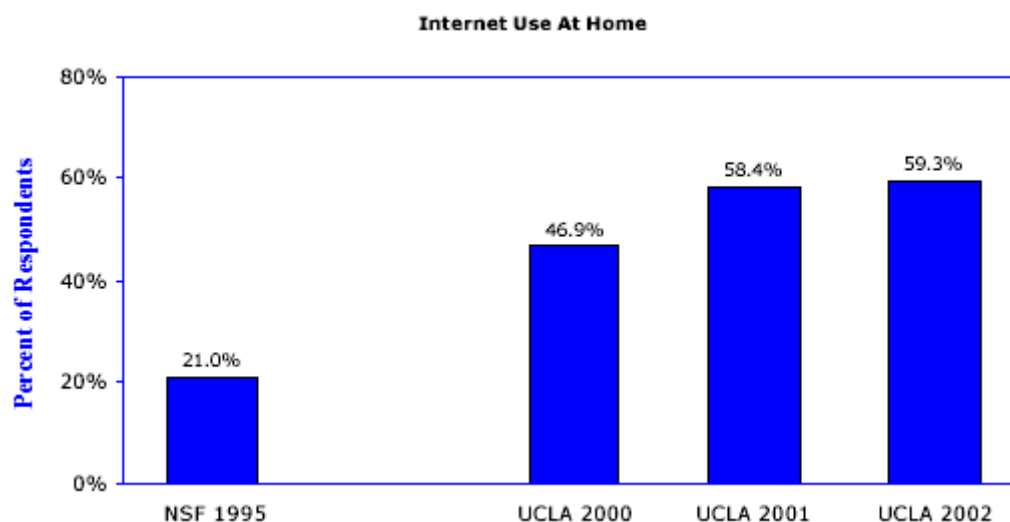
⁷ Podatek ni popolnoma zanesljiv.

aprila 2003 in kaže, da do interneta ne dostopajo vsa gospodinjstva z računalnikom.



Grafikon 1.4: Delež gospodinjstev z dostopom do interneta

Grafikon 1.4 kaže razmerje v decembru 2002, kjer gre za podoben razmik, pri čemer gre za relativno stabilno razmerje. Uporabljam ga za primerjavo z deležem gospodinjstev z dostopom do interneta v ZDA v juniju 2002 iz UCLA report-a, ko je bil delež posameznikov, ki so dostopali do interneta od doma naslednji:



Grafikon 1.5: Uporabniki interneta v ZDA z dostopom od doma

(Lebo, 2003: 24)

Opozoriti je treba, da zgornji grafikon (grafikon1.5) ne kaže na uporabnike interneta, ki do le-tega dostopajo od doma. Niso namreč vsi, ki imajo dostop do

interneta od doma, tudi uporabniki interneta⁸. Zaostanek EU za ZDA znaša več kot dve leti, Slovenija pa je trenutno nekje na ravni ZDA iz leta 1999. V maju 2002 pa je znašal časovni razkorak skoraj 7 let, če primerjamo ti dve raziskavi. To pomeni, da se razkorak Med Slovenijo in ZDA zmanjšuje. Predpostavljamo lahko, da dosega ZDA začetne točke zasičenosti in da se letna rast zmanjšuje. Pri nas pa smo še v obdobju, ko so rasti v primerjavi z ZDA velike. To potrjuje tudi sama Tabela 1.1., saj vidimo, da se je delež ameriških gospodinjstev z dostopom do interneta med letoma 2000 in 2001 povečal za skoraj 50%, v naslednjem letu pa le za slab odstotek. Ne glede na to pa so razlike tako velike, da lahko malce podvomimo v rezultate, ki pa le nakazujejo neke trende. Če bi se popolnoma zanesli na UCLA raziskavo, bi lahko sklepali, da bo v letu 2003 v EU rast priključevanja na internet 50%, v kar pa osebno malce dvomim.

Zanimiva je tudi primerjava uporabnikov interneta v celotni populaciji ne glede na to, ali je gospodinjstvo, v katerem živijo, priklopljeno na internet. Tako iz zadnje raziskave Sibus vidimo, da tako kot obstajajo lastniki računalnikov, ki imajo računalnik doma, pa ga ne uporabljajo za dostop do interneta, tako obstajajo tudi člani gospodinjstev, ki imajo možnost dostopa do interneta od doma, pa te možnosti ne izkoristijo in do interneta sploh ne dostopajo ali pa ga uporabljajo z drugih lokacij. Po podatkih raziskave projekta SIBIS⁹ (Sibus Pocket Book 2002/03, 2003: 16) je takšnih ali drugačnih uporabnikov interneta v januarju 2003 v Sloveniji 45%, v EU 54% in v ZDA maja 2002 77%. Primerjave med Slovenijo na eni strani in EU in ZDA na drugi strani so sicer izvedene z več kot polletnim zamikom, so pa potekale na enakih vprašalnikih, kar omogoča primerjave.

Iz zgoraj navedenih odstotkov je mogoče ugotoviti, da je penetracija uporabe interneta manjša kot penetracija internetnih priključkov po gospodinjstvih, kar pomeni lahko le to, da ne dostopajo do interneta vsi člani gospodinjstev. Problem sem povzel v podpoglavju z naslovom Digitalni razkorak. Predpostavljamo pa lahko, da bodo prej ali slej tisti, ki do interneta dostopajo iz drugih lokacij, do tega dostopali tudi od doma, torej da bodo tisti, ki bi lahko dostopali do interneta od doma, to možnost tudi izkoristili.

⁸ Acces to internet at home je fraza, s katero Lebe pojasnjuje tabelo, kar kaže na napako v naslovu tabele.

⁹ Opravljena je bila v aprilu - maju 2002 za EU in ZDA ter v januarju 2003 za Slovenijo in pridružene članice EU.

1.1.3. Podjetja

Na vsakem trgu se srečujeta vsaj dve strani in sicer stran povpraševanja in stran ponudbe oz. kupci in prodajalci. Zgoraj smo opazovali položaj Slovenije v primerjavi z EU in ZDA glede na internetno informacijsko infrastrukturo. Pri tem sem usmeril pozornost predvsem na stran trenutnih in bodočih potencialnih kupcev. V nadaljevanju pa bom obravnaval ponudbeno stran internetnega trga. Pri tem je potrebno upoštevati, da se je v teoriji zaradi lažjega statističnega opazovanja in zaradi nekaterih specifičnosti uveljavila delitev internetnega trga na dve kategoriji in sicer na prodajo podjetje - podjetju in prodajo podjetje - končni uporabnik. Obe prodaji se lahko odvijata preko interneta. Pri tem ima vsaka svoje značilnosti in posebnosti. Naj navedem samo nekatere, ki se mi zdijo najpomembnejše:

1. B2B¹⁰ večinoma prodaja znanim kupcem, pri B2C¹¹ pa poteka večina prodaje neznanim kupcem.
2. Evidenca B2B glede prodaje se povezuje s centralno bazo podatkov in s šifranti in je kot celota vključena v sistem prodaje. Vključenost kupca v računalniško evidentiranje dogodkov in s tem povezano vodenje sprememb v bazi podatkov sta torej pri prodaji B2B veliko bolj razdelana kot v prodaji B2C.
3. Internetna prodaja B2B je večinoma bolj redna se odvija v več ali manj podobnih časovnih intervalih. Prodaja B2C pa je bolj naključna.
4. Prodaja B2B zahteva veliko večjo stopnjo varnosti naročanja, v povezavi s tem pa zahteva ta vrsta prodaje dodatne programske module, ki skrbijo za varnost in tajnost prenesenih podatkov. Pri prodaji B2C pa se varnostni moduli vključijo običajno le, v kolikor se sama prodaja zaključi tudi s plačilom preko kreditne kartice in preko interneta. Ti moduli se običajno naročajo kot celoten program, ki se doda siceršnjemu prodajnemu katalogu.

Ker naloga obravnava spletno trgovino za mala podjetja, bomo izločili trg B2B in nadaljevali s primerjavo tržišča B2C med Slovenijo, ZDA in EU. Pri tem me najbolj zanima vključenost interneta v samo informacijsko strukturo podjetja.

¹⁰ B2B Business to Business oziroma podjetje - podjetju.

¹¹ B2C Business to Consumer, kar pomeni prodajo podjetja potrošniku.

Predpostavljam namreč, da ni potrebno tako kot pri gospodinjstvih ugotavljati, ali so slovenska podjetja opremljena z računalniki in kolikšen je pri tem razkorak med Slovenijo in ZDA ali EU. Raziskava RIS2002 podjetja elektronsko poslovanje (Vehovar in drugi, 2003: 2) namreč ugotavlja, da je imelo internetno povezavo kar 93% slovenskih podjetij, se pravi, da je imelo vsaj toliko podjetij tudi vsaj en računalnik. Razlika med Slovenijo in katerokoli drugo razvitejšo državo je lahko torej največ 7%. Teh 7% pa ne predstavlja velike možnosti širjenja na tej osnovni ravni. Kazalci za primerjanje informatizacije Slovenije v primerjavi z razvitejšimi so preveč kompleksni za raven te naloge, saj večinoma ni mogoče ugotavljati zaostajanja le z nekaj kazalci temveč s različnimi sestavljenimi oz. večdimenzionalnimi kazalci. Tako bi morali npr. meriti programsko opremljenost, usposobljenost osebja in vključenost hardversko - softverskih tehnologij s kompleksnimi indikatorji, sestavljenimi iz večih kazalcev. Pri vsem tem pa je primerljivih podatkov zelo malo in so zame nedosegljivi. Kljub temu pa lahko do neke mere primerjamo stopnjo penetracije interneta v Sloveniji, EU in ZDA z nekaj osnovnimi kazalci. Za spletno trgovino je pomembna predvsem penetracija komponente www¹². Pri tem se naslanjam na statistiko o internetnih strežnikih, ki jo vodi RIPE Network Coordination Centre (RIPE Host Count, June 2003, 1). RIPE v okviru svoje siceršnje dejavnosti, to je vodenje baze podatkov o internetnih domenah in gostiteljih le-teh, skenira domene in v okviru le-teh poišče strežnike. Je skratka eden od štirih (ARIN, APNIC, LACNIC) regionalnih internetnih registrov, ki alocirajo bloke IP naslovov lokalnim internetnim registrom (pri nas ARNES). Spodnja tabela prikazuje del pregleda za maj 2003.

Osnovno internetno strukturo, ki je na razpolago podjetjem, lahko opazujemo s pomočjo treh kazalcev in sicer z številom prebivalcev na strežnik, s številom strežnikov na domeno, ter s številom domen na 10.000 prebivalcev. Pri tem se je potrebno zavedati, da število domen, dodeljenih poddomeni .si oziroma dodeljenih v okviru nacionalnih poddomen, ni popolnoma verodostojen podatek. Podjetja uporabljajo namreč še .com, .net, .org in druge osnovne domene, te pa je po nacionalnih ključih zelo težko spremljati. Register teh gTLD¹³ namreč ni nacionalno voden. Število dodeljenih domen za posamezno državo v .com in .net prostoru bi bilo mogoče ugotavljati le z anketami po posameznih državah. Znan pa

¹² E-mail nas kot trgovce manj zanima.

¹³ gTLD - global top level domain oz. slov. globalne vrhnje domene (.com, .net, .org, .edu itd.).

je podatek, da se približno 85% .com domen registrira v ZDA (Waraker, 23.1.2003,1). Januarja 2001 je bilo 42% gTLD domen registrirano v ZDA (Zook, 2001;1). Ostalih 27% gTLD domen so registrirali v drugih državah, preostalih 30% domen pa predstavljajo vse ccTLD¹⁴ domene skupaj. Predpostavka, da se gTLD domene dodeljujejo v razmerju z nacionalnimi domenami, se izkaže za napačno. Kljub temu lahko iz tabele 1.2. (RIPE 2003) ugotovljamo neka razmerja, dinamiko le - teh pa ne.

vir: RIPE; maj 2003	TLD	Št. prebivalcev na strežnik	Št. domen	Število domen na 10000 prebivalcev	Št. prebivalcev CIA - 2002	Št. strežnikov na domeno
Finska	fi	4	37073	71,4	5.190.000,00	34,9
Nizozemska	nl	5	710738	440,2	16.146.000,00	7,9
Danska	dk	7	71854	133,5	5.383.000,00	15,6
Švedska	se	10	85795	96,6	8.878.000,00	13,4
Avstrija	at	16	187573	229,1	8.187.000,00	4,4
Velika Britanija	uk	19	1858943	310,4	59.894.000,00	3,7
Irska	ie	26	23458	59,8	3.922.000,00	8,5
Nemčija	de	31	4002939	479,7	83.452.000,00	3,6
Francija	fr	37	109400	18,2	59.959.000,00	19,4
Belgija	be	51	93880	91,8	10.228.000,00	3,9
Španija	es	57	38146	9,5	40.110.000,00	20,1
Portugalska	pt	61	7283	7,2	10.101.000,00	25,6
Grčija	gr	61	54018	50,6	10.665.000,00	5,7
Italija	it	73	208794	36,2	57.742.000,00	3
Luxemburg	lu	168	3742	82,6	453.000,00	1,8
EU povprečje		41,73		141,13		11,43
Slovenija	si	50	11953	61,8	1.935.000,00	4,7

Tabela 1.1: Primerjava Slovenija EI v letu 2003 po številu domen in strežnikov

Kar zadeva strežnike, vidimo, da en strežnik streže približno 8 prebivalcev več v Sloveniji kot v EU, da je v EU registrirano na 10.000 prebivalcev skoraj enkrat več domen (128%) oziroma je količnik 2.28, ter da pride v Sloveniji 2,43-krat manj strežnikov na domeno kot v EU. Že bežen pregled na nove spletne strani v Sloveniji pa pokaže zanimivo razmerje. Domene .si predstavljajo le 7,5% vseh novih spletnih objav, kar pomeni, da domene in registracija le-teh le niso tisti osnovni kazalec informacijske infrastrukture, temveč, da je strežniško omrežje, preko katerega slovenska podjetja dostopajo do interneta, najpomembnejši kazalec. Tu pa za Evropo zaostajamo za približno 20%, kar verjetno ni zelo veliko.

¹⁴ ccTLD - country code Top Level Domain, slov.: vrhnje domene po državnih kodah (.si, .cro, ipd.).

Navedenih podatkov ne smemo jemati preveč natančno, saj obstajajo podobne raziskave, ki kažejo samo v Sloveniji tudi 3-krat več strežnikov. Internet Software Consortium je tako za januar 2003 zabeležil pri nas 30.740 strežnikov od tega aktivnih 30.002. (Distribution by Top-Level Domain Name by Name Jan 2003, 2003: 1), vendar upam, da daje metodologija RIPE podobne rezultate, ki kažejo na neka razmerja med državami, več pa ne.

Števila domen ne smemo enačiti s številom spletnih predstavitev. Slednjih je namreč veliko več, saj lahko pride na eno domeno tudi 10.000 spletnih predstavitev. Podjetje Netcraft, ki do neke mere spremlja spletne predstavitve po strežnikih beleži 13.680 spletnih predstavitev s .si ccTLD na strežnikih, ki se odzovejo na http zahtevo. Če upoštevamo, da je 25% vseh novih spletnih predstavitev poddomena domene .si, potem lahko sklepamo, da gostimo v Sloveniji približno 54.720 spletnih predstavitev oziroma na 35 prebivalcev eno predstavitev. To pa v EU okvirih ni slabo. Glede na to, da je spletna predstavitev sestavljena iz spletnih strani, potem je zanimivo uporabiti ugotovitve iz članka The Web: More Than 2 Billion Pages Strong (Pastore, 2000: 1), ki povzema Cyveillance študijo "Sizing the Internet", ki meni, da je bilo v letu 2000 v povprečju 23 internih povezav na dokumente na povprečno spletno stran. To bi pomenilo, da je v Sloveniji na spletu objavljeno 4.195.200 dokumentov oziroma več kot 2 na prebivalca Slovenije. Tovrstni izračuni pa niso objektivni, ker so lahko povezave na iste spletne strani navedene v več različnih spletnih predstavitvah.

1.1.3.1. Podjetja s spletnimi stranmi oz. število spletnih strani na 1000 prebivalcev

Vsaj toliko kot sama internetna infrastruktura nam pove tudi število spletnih strani na 1000 prebivalcev, odstotek podjetij z spletnimi predstavitevami ter odstotek podjetij z spletno trgovino B2C.

Poročilo OECD Communications Outlook 2003 (OECD Communications Outlook 2003, 2003:123) ugotavlja, da je bilo julija 2002 v ZDA postavljenih 65 spletnih strani¹⁵ na 1000 prebivalcev ZDA, v EU približno 38, za Slovenijo pa ta podatek ni

¹⁵ Gre za spletne predstavitve in ne posamične html dokumente.

dosegljiv. Grobo lahko ocenimo 28 spletnih predstavitev pod pogojem, da znaša delež novih spletnih predstavitev pod .si domeno 25% (pregled novih spletnih predstavitev na www.najdi.si za obdobje petek 11.7.2003 do četrtek 17.7.2003). Zaostanek je še večji kot pri strežnikih. Če pa upoštevamo slovenske spletne strani na direktoriju www.najdi.si, znaša vsota le teh 50.167, kar pomeni 25 na 1000 prebivalcev. Obe metodi pripeljeta do podobnih izidov. Upoštevati pa je potrebno, da je pregled narejen na dan 17.7.2003, se pravi več kot leto kasneje kot primerljiva primerjava OECD. Kljub temu pa lahko vidimo, da je zaostanek Slovenije za EU in še posebej za ZDA zelo velik, veliko večji kot pri merjenju zaostanka v razmerju med prebivalstvom, informacijskimi tehnologijami in internetom.

Oglejmo si še razmerje med Slovenijo, EU in ZDA z ozirom na število spletnih predstavitev po podjetjih. Iz raziskave RIS2002 - Podjetja Internet in informacijske tehnologije (Vehovar in drugi, April 2003: 4) je razvidno, da je imelo konec leta 2002 spletno predstavitev 38% slovenskih podjetij, od tega 71% velikih podjetij. V EU za primerjavo lahko uporabimo raziskavo E-Business in EU in 2002 (Selhoffer, 2002: 19), po kateri je imelo v juniju/juliju 2002 spletno stran 54% vseh EU podjetij, v katerih pa je zaposlenih 70% delavcev vseh podjetij. Podatkov za ZDA v tej obliki za leto 2002 ni mogoče dobiti. V začetku leta 2001 pa je po podatkih WOW! Facts 2002 imelo spletno stran približno 54% podjetij, ki niso v lasti ameriških manjšin (WOW! Facts 2002, 2002: 746). Glede na te podatke je mogoče sklepati na eno leto in pol zamika med EU in ZDA, Slovenija pa za EU zaostaja skoraj za 16 odstotkov, kar pomeni, da bi morali slovenski podjetniki povečati število spletnih predstavitev skoraj za polovico, če bi hoteli dohiteti EU. V zvezi s tem so zanimivi podatki iz članka Small Businesses Still Lag In E-Commerce Ventures (Lisser, 2003:1), v katerem avtorica navaja podatke raziskave Yankee Group, ki ugotavlja, da ima tako ali drugačno internetno prisotnost 54% malih in 62% srednje velikih podjetij v ZDA. Malce drugačne podatke pa predstavlja publikacija Targets for e-business policies for SMEs (Targets for e-business policies for SMEs, 2003: 12), po kateri naj bi bilo v ZDA kar 77% podjetij s spletno stranjo in v EU (samo 13 članic) 62%, kar kaže na še večji razkorak med razvitimi in Slovenijo.

1.1.3.2. Podjetja s prodajo preko spletnih strani

Če primerjamo za začetek položaj Slovenije in ZDA, lahko ugotovimo, da je že leta 2001 (WOW! Facts 2002, 2002: 746) uporabljalo trgovino on-line 10% ameriških podjetij. Zanimivo je, da je delež med ameriški manjšinami še večji. Leto kasneje pa je 6% slovenskih podjetij prejelo naročila preko spletnih aplikacij (Vehovar Pfajfar in drugi, 2003: 4). Opozoriti velja, da spletna aplikacija še ni online trgovina, temveč je mogoče razumeti, da se v tem pojmu skrivajo tudi B2B naročila. V istem času je opravljalo prodajo preko interneta 6% podjetij v EU (Sibis Pocket Book 2002/03, 2003: 56), vendar pa gre v tem primeru le za prodajo končnim potrošnikom in v to prodajo ni vključena prodaja B2B. Je pa potrebno dodati, da je bila ta raziskava delana na vzorcu 7 držav, v katerem ni bilo Grčije in Portugalske kot manj razvitih dežel, ter Danske, Norveške in Irske kot bolj razvitih.

1.2. Nakupi on-line

Nakupi preko internetnih trgovin so kot področje, ki je še posebej povezano s to nalogo, relativno nerazviti. ZDA kot računalniško najrazvitejša država realizira preko internetnih trgovin okoli 2% skupne prodaje (Vehovar in Lobe in drugi, 2003: 4), vendar pa se letni obseg tovrstne prodaje povečuje s stopnjo 25%, kar je resnično zelo veliko. Strokovnjaki ocenjujejo, da bo delež dosegel 10%. Vseh proizvodov se namreč preko interneta ne da prodajati. V Sloveniji predstavljajo nakupi on-line približno 0,2% (Vehovar in Lobe in drugi, 2003: 5) skupne prodaje in v tem smislu za ZDA zelo zaostajamo. V teh odstotkih je zaobsežena prodaja, v kateri se tudi plačilo realizira preko internetnih povezav, in tudi prodaja, pri kateri se preko internetnih povezav dobijo le podatki in naročilo, sama prodaja pa se realizira na druge načine.

Odstotek prebivalstva, ki kupuje preko internetnih povezav, pa kaže malce drugačno sliko. V ZDA kar 44% prebivalstva kupuje tudi on-line, v EU je online kupcev več kot polovico manj oziroma približno 19%. V Sloveniji pa se giblje odstotek prebivalstva, ki včasih kupi kako stvar on-line, okoli 7% (SIBIS Pocket Book 2002/03, 2003:44). Glede na to, da pri deležu internetne prodaje z ozirom na celotno prodajo za ZDA skoraj 10-krat zaostajamo, pa je tu zaostanek veliko

manjši. Delno si lahko to pojasnimo z manjšo ponudbo slovenskih spletnih trgovin, delno pa tudi s tem, da v ZDA kupujejo tudi državljani drugih držav. Prav internet namreč omogoča povprečnemu Slovencu, da doseže ponudnike izdelkov in storitev, do katerih še pred leti ni imel dostopa in o njih sploh ni imel informacij. Na tem področju torej ZDA izkoriščajo svoj vodilni tehnološki, informacijski in tudi kulturni položaj v svetu.

1.3. Digitalni razkorak

Samo podatki o razporeditvi internetnih priključkov po gospodinjstvih oziroma o odstotku prodaje preko interneta v primerjavi s celotno prodajo v državi imajo sicer veliko pojasnjevalno vrednost, vendar pa lahko izgubimo številne pomembne informacije, vkolikor se usmerimo predvsem na te kazalce. V primerjavi s stanjem v EU oziroma v ZDA lahko sicer ugotavljamo morebitna gibanja internetne prodaje v prihodnosti, to pa je tudi vse, kar lahko ugotovimo. Če želimo ugotoviti kaj več o nakupovalnih navadah internetnih potrošnikov, potem moramo v analizo vključiti tudi druge kazalce. **Raba interneta in s tem povezano nakupovanje ni enakomerno porazdeljeno po vseh slojih družbe.** Kot ugotavljata Vehovar in Vukčević (Vehovar in Vukčević, 2001: 2), na uporabo interneta vplivajo predvsem sociodemografske značilnosti opazovane družbe.

Pojem digitalni razkorak, ki ga uvedeta kot pojasnjevalno orodje, se običajno računa na osnovi enostavnih razlik v penetraciji uporabe interneta po različnih sociodemografskih skupinah.

Glede na to, da je slovenska družba kot vsaka družba sociodemografsko strukturirana bodisi po starosti, spolu, regiji bivanja, dohodku in predvsem po izobrazbi, so to tudi glavne kontrolne spremenljivke, ki jih Vehovar in Vukčević primerjata med drugim s spremenljivkama Dostopa do interneta od doma in Lasten PC doma. Glede na to, da primerjalnih študij, ki bi vsebovale tudi digitalni razkorak v tujini, ni na voljo v prosti obliki za isto obdobje, bomo v nadaljevanju iz izvrstne študije zgoraj omenjenih slovenskih raziskovalcev povzeli le za nalogo najpomembnejše ugotovitve (Vehovar, Vukičević, 2001:5):

- "Izkaže se, da je izobrazba običajno najbolj izrazita spremenljivka, predvsem v prvih letih posvajanja določene tehnologije (PC, internet), kasneje pa se kot pomembni pojavljajo starost, dohodek, zaposlitveni

status in tip krajevne skupnosti, ki v nekaterih primerih celo izpodrinejo izobrazbo kot najpomembnejši dejavnik digitalnega razkoraka.

- Če populacijo razdelimo glede na kontrolno spremenljivko (starost, dohodek ali izobrazba) v štiri skupine, potem znaša časovna razlika med sosednjima skupinama tipično štiri do pet let in preko 15 let med obema skrajnima skupinama.
- Tovrstno oblikovanje skupin slovenske populacije pokaže, da najdemo več kot tri četrtine uporabnikov interneta le v petini populacije, to je segmentih, ki so opredeljeni z izobrazbo, starostjo, tipom krajevne skupnosti in dohodkom." (Vehovar in Vukčević, 2001: 5-6)

Kaj ti podatki pomenijo za podjetnika, ki želi tržiti preko internetne spletne trgovine? Potencialne stranke lahko pričakuje v petini populacije, predvsem med mladimi visoko oz. višje izobraženimi, ki stanujejo v urbanih naseljih, ali pa med šolajočo se mladino. V 10 do 15 letih pa se bo verjetno struktura potencialnih strank izenačila. Pri tem se bodo v prvih petih letih v čedalje večji meri pojavljali mladi srednješolsko izobraženi kupci in kasneje tudi starejše srednješolsko izobražene generacije.

1.4. Značilnosti nakupov

Med izdelki in storitvami, ki se najbolj prodajajo, izstopajo po TNS Interactive - Global eCommerce report June-2002 (TNS Interactive - Global eCommerce report June-2002, 2003: 1) knjige, glasba in CD plošče, oblačila, elektronika in električni aparati, storitve s področja turizma¹⁶, PC strojna oprema, programska oprema ipd. Opazen je trend zviševanja nakupovalnih zneskov. Ne gre za višje cene, temveč za pojavljanje blagovnih skupin na internetnem tržišču, ki so dražje kot celota.

Kakšne so prednosti on-line nakupov kot jih vidijo potrošniki? Predvsem so to dostop do trgovin skoraj povsod po svetu in s tem povezana skoraj neomejena izbira, neodvisnost od obratovalnega časa, nižje cene artiklov kot v klasičnih trgovinah, dostava na dom, možnost takojšnjega prenosa posameznih artiklov s

¹⁶ Rezervacije terminov in lokacij ter letalske karte ipd.

spletne strani¹⁷ (Nakupovanje na spletu, 2002,1). Spletne strani naj imajo po željah potrošnikov bogato izbiro in na prvem mestu lahko in hitro navigacijo, podrobne opise blaga v prodaji, možnost vračanja izdelkov, blago naj bo dobavljeno v najkrajšem možnem roku oziroma v roku, ki je dogovorjen in vnaprej določen. Potrošniki pričakujejo še zanesljivost in kredibilnost same trgovine.

1.5. Želje uporabnikov - varnost

Pri tem je potrebno termin kredibilnosti dodatno razdelati, kar bomo najlažje storili, če povzamemo iz Confidentiality concern and on-line shopping (glej Vehovar in Lobe, 2001: 6) najpomembnejše ovire pri nakupovanju. Strahovi, ki povprečnega potencialnega kupca odvrčajo od nakupa preko interneta, so, kot ugotavlja Vehovar:

- nezaupanje v resničnost podatkov o izdelkih oz. podjetju, ki te izdelke predstavlja (Credibility),
- nezaupanje v zasebnost naročanja oziroma strah pred zlorabo podatkov in skrb za varnost osebnih podatkov, ki jih dobi z naročilom trgovec pred potencialnimi drugimi podjetji (Privacy).
- nezaupanje v zaupnost podatkov(Confidentiality).

Dodal bi še nezaupanje v samo finančno transakcijo. Pri tem gre lahko za strah pred zlorabo kreditne kartice, strah pred tem, da naročenega blaga ne bo oziroma da ne bo ustrezno kvalitetno, ter odpor zaradi zapletenega postopka reklamiranja ali strah, da reklamacija ne bo realizirana.

Gre torej za sklop zahtev in pojmov, ki se prepletajo med seboj, ki jih je težko meriti, vse pa povezuje med seboj pojem varnosti. Posamičnemu trgovcu je odpravljanje teh strahov v marsičem tudi onemogočeno oziroma odpravljanje teh zadržkov ni v njegovi moči. Dejstvo je, da lahko potencialni trgovec celoten postopek prodaje in svojih evidenc v zvezi s tem maksimalno zaščiti, to tudi objavi, pa mu potencialni kupec tega ne bo verjel, saj nima možnosti preveriti resničnosti trditev. Povprečen kupec bi moral na primer poskusiti vdreti v sistem, da bi videl, ali so njegovi podatki res tako zaščiteni, kot je obljubljeno. In če bi ta povprečen kupec ugotovil, da so vse varnostne mere ustrezno izvedene, še vedno ne bi mogel vedeti, če se za to varnostno popolno fasado ne skrivajo napačni

¹⁷ Računalniški programi, fotografije, internetni časopisi ipd.

nameni. Podjetje s popolnim varnostnim sistemom s Secure socket layer's povezavami in z enkripcijo z 128 bitnim ključem bi tako lahko po polovični ceni prodajalo npr. tiskalnice v vrednosti 5.000 \$, pobralo denar s kreditne kartice, lahko celo preko specialne linije, in nato nikoli ne poslalo naročenega blaga, temveč sprožilo stečaj ali pa enostavno izpraznilo svoj bančni račun in poniknilo. Gre torej za to, da lahko varnost zagotovi le ustrezna institucija za nadzor internetnih prodajalcev, ki bi lahko preverila programsko in tehnološko zaščitenost prodajnega sistema in ob tem še kredibilnost podjetja. Podatki o tem, katera podjetja so uspešno prestala preverjanja, bi morali biti dostopni na javnem mestu. Sama podjetja pa bi bila še kako zainteresirana za to, da bi lahko imela na spletni trgovini nek znak, ki bi dokazoval, da spadajo v sistem nadzora. V Sloveniji je jasno vsakemu samostojnemu podjetniku, da mora njegova obratovalnica ustrezati minimalnim tehničnim in drugim pogojem za obratovanje. Za spletno obratovalnico to še ne velja.

Za zaključek lahko torej ugotovimo, da je nadaljnji razvoj trgovanja online odvisen od naslednjih dejavnikov:

- Standardizacija varnostne programske opreme in s tem povezana povezljivost varnostno zaščitnih komponent s samimi oblikovalskimi in programskimi deli spletnih trgovin.
- Standardizacija na področju države. Ta bi s svojimi ukrepi zagotovila neko stopnjo varnosti predvsem skozi *ukrepe za hitro in učinkovito reševanje sporov* oziroma pritožb. V okviru tega bi lahko ustanovila nek **organ**, (e-ombudsman za spore in nek sistem podeljevanja značk odličnosti), ki bi pospeševal reševanje sporov, skrbel za *registracijo spletnih trgovin*, in bi na nek način ugotovil, ali ima podjetje ustrezne programske, finančne in druge vire za učinkovito poslovanje. S tem bi ustvarjali pri kupcih zaupanje v e-trgovino. Primer je znak slovenske kakovosti, ki bi ga v določeni obliki lahko vpeljali kot znak, ki bi ga lahko imela trgovina. Ta bi stranki zagotavljal, da je podjetje, katerega izdelke si ogleduje, resno in zaupanja vredno, da bo v primeru reklamacij ravnalo hitro bodisi z vračilom denarja ali pa z dobavo novega izdelka, da podjetje ni v stečaju in da bo za svoj denar stranka dobila tisto, kar naroča in podobno.

- Razvoj internetnega omrežja, hitrost omrežja, ki omogoča multimedijско predstavitev ipd.

Glede na to, da je problem pereč, se tudi pri nas sprožajo pobude za standardizacijo ocenjevanja on-line trgovin. Eno od teh navajam v celoti in jo povzemam iz projekta Izvensodno reševanje potrošniških sporov na področju e-poslovanja končno poročilo (Kutin in drugi, 2001: 92). Gre za 15 kriterijev, po katerih naj bi se ocenjevale spletne trgovine:

1. Izložbeno okno / prva stran
2. Obseg ponudbe
3. Iskanje izdelkov
4. Opis izdelkov
 - fotografija in opis
 - dodatne informacije
5. Nakup
 - enostaven nakupni proces
 - arhiv nakupov
 - samodejno pismo zahvale
6. Naročanje / plačevanje
 - izbira med različnimi kreditnimi karticami
 - po povzetju
7. Načini dostave
 - neposreden prenos s spletne strani
 - hitro (v istem dnevu)
 - v 24 urah
 - običajno
8. Sledenje naročilom
9. Pospeševanje prodaje
 - darilni boni
 - igre z zanimivimi nagradami
 - vprašalniki
 - darila
10. Programi zvestobe
 - nakupne točke

- kuponi
- pošiljanje novic
- preizkus izdelkov
- mnenja obiskovalcev spletne strani
- predlogi

11. Dodatne storitve

- e-voščilnice
- podatki o vsebini / novosti

12. Politika zasebnosti

- zaščita pred tretjo osebo
- politika, povezana s piškotki (cookies)

13. Politika vračanja izdelkov

- garancije

14. Pomoč kupcem

- telefonske številke
- možnost e-pomenkovanj
- najpogostejše zastavljena vprašanja (FAQ)

15. Personalizacija

1.6. Zaključek

Kot vidimo je razkorak med Slovenijo in razvitejšimi državami relativno majhen. Moramo pa upoštevati, da so informacijska infrastruktura, pogosta raba interneta in spletni nakupi sicer med seboj soodvisni, niso pa v linearnem razmerju, kar bi pomenilo, da se s spremembo ene spremenljivke avtomatično spremeni za vnaprej določeno vrednost tudi druga spremenljivka.

Položaj bi lažje opredelili kot splet spremenljivk, ki so vse med seboj soodvisne, vendar pa je vpliv posamezne spremenljivke na vse ostale odvisen od medsebojnih razmerij med velikostjo ostalih spremenljivk.

Zaradi omejenosti diplomske naloge pač ni mogoče vključiti vseh spremenljivk. Na tem mestu bi še posebej rad opozoril na določene dejavnike, ki spletnega udejstvovanja ne pospešujejo samo linearno, se pravi v nekem fiksnem razmerju, temveč bi si upal trditi, da ga pospešujejo celo geometrijsko. Če poenostavimo: določen dejavnik A ob neki ravni vpliva na dejavnik B tako, da se

ta podvoji. Če se raven dejavnika A podvoji, se ne bo podvojila raven dejavnika B, temveč se bo mogoče celo potrojila. Potrojitev dejavnika B pa bo nato mogoče **vzvratno** vplivala na potrojitev dejavnika A.

Eden od takih dejavnikov, ki ga je zelo težko meriti, je ponudba spletnih vsebin. To je lepo razvidno iz samega razvoja interneta. Ko je bil ta omejen v svoji zgodnji različici na vojaške in akademske kroge, so ga ti relativno uspešno prevzeli. Sama komunikacija je bila bolj novičarsko , BBS¹⁸ oziroma e-mail omejena. BBS so bili strežniško programski sistemi, ki so ponujali prenos datotek¹⁹ in gostovanje novičarskih skupin. Bili so več ali manj konzolno oblikovani, včasih z enostavnimi grafičnimi vmesniki, ki pa jih je bilo potrebno programirati. Vsak BBS je bil program zase, tako kot je program zase npr. računovodski program. Zato je bilo potrebno tudi veliko časa za spremembe, naj so bile te vizualne ali vsebinske. Če povzamemo, imeli so telefonske vode in nov TCP/IP protokol, ki je povezal **hardversko različno** zasnovana omrežja v eno omrežje. Mreže različnih proizvajalcev so lahko komunicirale. S pojavom označevalnega jezika HTML (nadgradnja TCP/IP), ki je omogočil postavitve grafično oblikovanih strani in web strežnikov praktično brez programiranja z izredno lahkoto, se je situacija popolnoma spremenila. Vključilo se je gospodarstvo, najprej spet računalniška industrija z akademsko izobraženimi vodstvenimi delavci in razvojniki, nato pa vsi ostali. Opazujemo lahko drastično povečanje vsebin za vse na novo vključene, skratka povečanje ponudbe, ta je pripeljala do rasti povpraševanja, tako kot ponudba vedno pripelje do povpraševanja, le-to pa nato vzvratno vpliva na ponudbo. Ta medsebojno pogojeni **vzvratni** vpliv med ponudbo in povpraševanjem je težko meriti. Dejstvo pa je, da so se s pojavom vsebin, ki niso bile namenjene le vojakom oz. akademikom in še to predvsem matematikom, fizikom in programerjem, temveč tudi npr. otrokom²⁰, ženskam, manj izobraženim²¹, vključile tudi te skupine in tudi lačno zahtevale nove dodatne vsebine.

Tukaj lahko lepo vidimo, kako tehnologija (HTML) vpliva najprej na dejavnik A (ponudba spletnih vsebin), ta vpliva na dejavnik B (povpraševanje po spletnih vsebinah), ta pa nato vzvratno vpliva na dejavnik A. Ko oba dosežeta določeno

¹⁸ Bulletin board system - sistem razglasne table ali oglasna deska.

¹⁹ Igrice, različne informacije in programe.

²⁰ Npr. igrice.

²¹ Npr. potovanja.

raven, se pojavi potreba po rasti začetnega dejavnika tehnologije. V HTML se vključijo nova orodja, skriptni jeziki in java. Vse skupaj pa zahteva čisto hardverski razvoj omrežja, prepustnost, strežniško regulativne sisteme, nove možnosti za prenos podatkov, kot so ISDN in ADSL. Spomnimo se, da se je vse skupaj začelo z mrežo preko telefonskih vodov in TCP/IP protokola, BBS-ov in e-mail sistemov.

Za nas je ponudba spletnih vsebin še posebej pomembna. Običajno se ta meri s kazalci, ki merijo odstotek podjetij s spletnimi stranmi, s številom podjetij s prodajnimi spletnimi stranmi, z razvojem e-uprave, z digitalno pismenostjo ali z odstotkom informatikov v neki državi in podobno. Vendar pa ti kazalci ne zadostujejo popolnoma in včasih tudi deloma zavajajo. Prihaja namreč do pojava, ki je v neki meri podoben ekonomiji obsega v ekonomskih znanostih. Za kaj gre? Predpostavimo na primer dve državi, v katerih se ljudje sporazumevajo v za državo lastnem jeziku. Četudi imamo v obeh primerjanih državah, od katerih ima npr. ena 250 milijonov prebivalcev, druga pa 100 krat manj, npr. 2,5 milijona prebivalcev, isto število programerjev na prebivalca, bo prva država proizvedla 100-krat več vsebin, pomembnih za svoje državljane. Ponudba vsebin bo torej enormna, s tem bo tudi veliko večja zanimivost vsebin, vsak bo namreč lahko našel nekaj zase, pri tem pa bo število spletnih strani na prebivalca v obeh državah enako. Glede na to, da običajno ljudje ne postavljajo popolnoma enakih spletnih strani, lahko opazimo, da se pri večjem obsegu ponudba specificira oz. prilagaja specifičnim skupinam uporabnikov z specifičnimi potrebami. Lahko rečemo, da se prav skozi ta proces specificiranja univerzalizira in tako postaja univerzalna. Za ilustracijo si predstavljajmo zanimivost interneta za npr. Američane in Angleže, ki bi jim za eksperiment odvzeli vse angleške spletne strani in jim v angleščini omogočili le ogled polovice v angleščino prevedenih finskih spletnih strani. Pri istih kazalcih bi lahko opazovali strmo upadanje in ugašanje priključkov na internet. OECD Communication Outlook 2003 (OECD Communication Outlook 2003, 2003: 127) tako ugotavlja, da je kar 58% vsebin na varnih strežnikih v angleškem jeziku, več kot $\frac{3}{4}$ vsebin pa samo v 3 jezikih: angleščini, nemščini in japonščini. Pa vendar so prav severnoevropski mali narodi dokaz, da ta ekonomija obsega ne deluje vedno oziroma da se jo da preseči s ciljno ponudbo države, kot je npr. e-državna uprava ali objava rezultatov raziskav kot RIS, ki daje

brezplačno na razpolago podatke, ki se npr. v ZDA drago plačujejo. Država se pojavlja kot producent oz. založnik in investitor družbenorelevantnih internetnih vsebin ali pa vključuje internetne vsebine kot strukturni del svojega mehanizma v svoje normalno delovanje (oddaja obrazcev, vlog ipd.). To pomeni, da na nek način odpira internet kot enega od kanalov svojega komunikacijskega mehanizma med seboj in državljani oziroma v komunikaciji med različnimi deli svoje sestave.

Jasno je, da se v tej "ekonomiji obsega" kot odločujoči faktor pojavljata jezik kot omejevalni dejavnik na eni in država kot spodbujevalni dejavnik na drugi strani. Ta "ekonomija obsega" zagotovo ne deluje Slovincem v prid, kljub temu pa verjetno pod predpostavko, da Slovenci ne znamo slovensko, v osnovnem in srednjem šolstvu namenjamo največ ur prav učenju slovenščine. Omenimo še zanimivo ugotovitev, da je protokol TCP/IP omogočil povezovanje med mrežami različnih proizvajalcev in je v tem smislu pomenil neko vrsto **standardizacije**. Ta proces pa lahko opazujemo skozi celoten razvoj računalništva in programerske industrije do Java COM in .NET arhitekture. Standardizacija je za Slovence kot majhen narod še posebej pomembna predvsem v smislu prilagoditve. Tako kot postaja ameriška hardverska in programska oprema standard sveta, tako postaja angleščina standarden jezik komunikacije v svetu oziroma svetovni jezikovni standard. Izključno znanje slovenščine postaja tako omejevalni dejavnik posameznikovega uspeha. Standardizacija kot univerzalen pojav razvoja tehnologije vpliva tudi na jezik kot sredstvo komunikacije.

1.7. Sklep

Če podrobno pregledamo ugotovitve iz poglavja o željah potrošnikov in kriterije, po katerih naj bi ocenjevali spletne trgovine, vidimo, da jih lahko za naše potrebe razvrstimo nekako v tri skupine:

1. Kriteriji, katerih upoštevanje je odvisno predvsem od zahtev naročnika oz. imetnika spletne trgovine.
2. Kriteriji, ki se uresničujejo predvsem skozi kodo HTML. Ker je ta v osnovi namenjena oblikovanju, so tudi ti kriteriji nekako vsebinsko vezani na oblikovanje, tu mislim predvsem na kriterij prve strani, obveščanje o postopkih reklamiranja, pogosto zastavljena vprašanja (FAQ), ipd.

3. V to skupino sem uvrstil za nalogo najpomembnejše kriterije, katerih uresničevanje je odvisno predvsem od programske kode. Zaradi pomembnosti jih bom posebej naštel:
- a. Obseg ponudbe
 - b. Iskanje izdelkov
 - c. Opis izdelkov (fotografija in opis, dodatne informacije)
 - d. Nakup (enostaven nakupni proces, arhiv nakupov, samodejno pismo zahvale)
 - e. Naročanje / plačevanje (izbira med različnimi kreditnimi karticami, ali po povzetju)
 - f. Načini dostave (neposreden prenos s spletne strani)
 - g. Sledenje naročilom
 - h. Pospeševanje prodaje (vprašalniki)
 - i. Programi zvestobe(nakupne točke , kuponi, pošiljanje novic, mnenje obiskovalcev spletne strani, predlogi)
 - j. Dodatne storitve (e-voščilnice)
 - k. Politika zasebnosti (zaščita pred tretjo osebo, politika, povezana s piškotki (cookies))
 - l. Pomoč kupcem (možnost e-pomenkovanj)
 - m. Personalizacija

Programsko bomo zadostili naslednjim kriterijem:

Po obsegu ponudbe, iskanju izdelkov in opisu izdelkov - fotografija in opis, dodatne informacije z vključitvijo ponudbe za razširljivo bazo podatkov.

Enostaven nakupni proces bomo vzpostavili z nakupno košarico. Arhiva nakupov pa ne bomo programsko izpeljevali iz razlogov, ki jih bomo podrobneje obravnavali v 2. poglavju. Pismo zahvale naj bi bilo poslano po potrditvi naročila.

Pri naročanju in plačevanju sem se zaradi nezaupanja, ki večinoma vlada²², odločil za plačilo po povzetju. Izbire med različnimi kreditnimi karticami oziroma plačevanja z njimi ne nameravam posebej izdelovati, saj je za to potrebno precej več kode kot za samo spletno trgovino. Poleg tega zahteva tovrstno programiranje posebna znanja programskih specialistov in tem je najbolje prepustiti oblikovanje delovanja plačilnega modula. Samo plačilo po povzetju pa v začetnih zagonskih

²² Slovenija je pri tem nekakšna izjema. Verjetno zaradi manjše razvitosti e-trgovine, uporabniki še nimajo slabih izkušenj.

fazah vzpostavitve spletne trgovine pomembno prispeva k zmanjševanju stroškov in hkrati deloma odpravlja strahove, ki so prisotni pri potencialnih kupcih. Pri tem pa zagotavlja visoko stopnjo varnosti. Prav tako ne bomo programsko omogočali neposredne dostave preko spletnih strani, saj gre pri njej večinoma za podjetja, ki svojo košarico naredijo sama po svoji meri. Sledenje naročilom se običajno omogoča s povezavo na spletno stran podjetja za dostavo. Ker ne vemo, katero podjetje bo to v konkretnem primeru, in tudi ne vemo, kakšen ukaz in parametre uporablja podjetje za dostavo pri izpolnjevanju svojega obrazca (GET ali POST), bomo to reševali od primera do primera.

Pospeševanje prodaje z vprašalniki, programi zvestobe, nakupnimi točkami , kuponi, pošiljanjem je odvisno predvsem od poslovne politike podjetja in naročnika. Novice in novosti je po novem zakonu o varstvu potrošnikov iz februarja 2003 brez vnaprejšnjega pisnega soglasja prejemnika prepovedano pošiljati končnim potrošnikom. Knjigo vtisov, mnenj s predlogi obiskovalcev spletne strani pa nameravam ponuditi kot poseben programski modul, ki ne bo povezan s samo aplikacijo spletne trgovine. Podobno je z e-voščilnicami. Politike zasebnosti iz razlogov, ki jih bomo posebej obdelali v naslednjem poglavju, ne bomo posebej izdelali. Politika, povezana s piškotki (cookies), pa je odvisna predvsem od samega naročnika in morebitnega izvajalca reklamnih akcij preko interneta. Gre za tako imenovane piškotke tretje strani (third party cookies). Piškotke neke domene lahko namreč pregleduje, čeprav izjemoma, tudi tretja domena. Zaradi varnosti bodo naši piškotki samo začasni piškotki, shranjeni v brskalnikovem delovnem spominu, brez zapisovanja na disk. Po odhodu uporabnika z naše spletne strani se tak piškotek zbriše iz spomina in nikoli ni zapisan na disk. Tako nima določenega datuma poteka. Ker bodo piškotki samo začasni, ne bomo mogli uvesti personifikacije spletne strani. Možnosti e-pomenkovanj ne nameravam omogočiti. Zahteva namreč stalno prisotnost nekoga na spletni strani, česar pa si mala podjetja, za katere spletno trgovino izdelujem, ne morejo privoščiti.

2. Industrijska organizacija in e-poslovanje

2.1. Uvod

Pred izdelavo programske kode je potrebno sprejeti še nekaj pomembnih odločitev. Med njimi je na prvem mestu predvsem odločitev o umestitvi in stopnji integracije spletne trgovine v siceršnji informacijski sistem podjetja. Da bi lahko opredelili obseg, vlogo in naloge informacijskega sistema v podjetjih, je potrebno vsaj v grobem preučiti podjetje kot industrijsko organizacijo, še posebej s stališča položaja informacijskega sistema v njem. V nadaljevanju pa bom raziskal, v kolikšni meri je lahko informacijski sistem povezan z internetom, še posebej glede na e-poslovanje, in v okviru tega z e-trgovino (e-commerce) kot posebnim podsistemom e-poslovanja. Ob tem pa bom obravnaval še virtualno podjetje, kot posebno obliko klasičnega podjetja.

2.2. Industrijska organizacija in klasično podjetje

Področje raziskovanja industrijske organizacije, kot jo definira Cabral (Cabral, 2000: 3), zadeva predvsem delovanja trgov in industrij, še posebej pa proučuje, kako podjetja tekmujejo drugo z drugim. Pri tem je potrebno razmejiti področje raziskovanja industrijske organizacije od mikroekonomike. Slednja naj bi se po Cabralovem mnenju (Cabral, 2000: 3) ukvarjala predvsem z proučevanjem tržišč. Pri industrijski organizaciji pa je poudarek na podjetju in interakcijah med podjetji in tržišči ter na strategijah podjetij, ki jih ta oblikujejo med temi interakcijami. Opozoriti je še potrebno na sam termin industrijska, ki ga ni za razumeti v "slovenskem" smislu industrijske proizvodnje, temveč v smislu proizvodnje za trg²³. Termin zato vsebuje tako klasično jeklarsko industrijo kakor tudi storitvena turistična ali marketinška podjetja. Polje preučevanja teorije industrijske organizacije je mogoče najlažje razumeti skozi razliko med le-to in ekonomijo ponudbe in povpraševanja. Slednja razume podjetje kot črno skrinjico, katerega notranjega delovanja ne opazuje. Delovanje proučuje le z vidika maksimalizacije profita. Raziskovalci industrijske organizacije pa razgrajujejo

²³ Ang. industries = prizadeven, izraz manufacture = ročno delati pa je v ang. jeziku osnova za tisti, kar v Sloveniji pojmuje kot industrija.

prav notranji mehanizem črne skrinjice oz. organizacije v interakciji z zunanjim okoljem.

Na tržiščih, na katerih nastopajo, se podjetja borijo za tržno moč, ki jo Cabral (Cabral, 2000: 6) definira kot zmožnost nastavitve cen nad stroški, še posebej nad marginalnimi stroški²⁴.

Tržna moč je obratno sorazmerna z elastičnostjo povpraševanja, ki jo isti avtor definira kot količino povpraševanja, deljeno z odstotkom variacije cene za izdelek.

Podjetja nastopajo na svojih osnovnih trgih kot ponudniki izdelkov, na drugih trgih pa kot kupci izdelkov drugih podjetij, njihov relativni položaj pa glede na konkurenčna podjetja opredeljujeta tako njihova tržna moč kot elastičnost povpraševanja njihovih kupcev.

Večina ekonomistov preučuje industrijsko organizacijo z modelom (ogrođjem) znanim kot STRUKTURA - VEDENJE - UČINEK. Pri tem struktura trga in same organizacije vplivata na vedenje, to pa povzroča različne učinke. Ob tem, da tvegamo poenostavljanje, bi lahko dejali, da je v tem smislu podjetje proces transformiranja inputa v output. Michael Porter ta model input - output oblikuje nekoliko drugače (Porter v Cabral, 2000: 12) in sicer kot model petih sil, ki vplivajo na industrijsko tekmovanje. Te sile so: dobavitelji, kupci, nadomestni proizvodi, morebitni (novi) vstopajoči na trg in obstoječi tekmeci na trgu. Mimogrede je že iz same te strukture mogoče ugotoviti, da podjetja, ki si delijo določen trg, lahko ob medsebojno oligopolno dogovorjenih cenah za končne potrošnike zmanjšujejo raven tekmovanja med seboj, dosegajo ekstra dobičke in si s tem izboljšujejo položaj za pridobivanje dobičkov na drugih trgih.

Omeniti je potrebno še vertikalno in horizontalno razsežnost podjetij. Ti se oblikujeta v konkurenčnem boju. Pri tem horizontalno razsežnost opredeljuje količina sorodnih izdelkov, ki jih podjetje proizvaja, oz. delež trga, ki ga podjetje zaseda. Vertikalna razsežnost pa določa, koliko proizvodnih stopenj do končnega izdelka se odvija v samem podjetju. Ob tem vertikalna razsežnost, ki v bistvu pomeni način organiziranja proizvodnje, oblikuje tudi strukturo odločanja in vodenja podjetja. Vertikalna razsežnost s tem, da podjetje samo proizvaja polizdelke, zmanjšuje lasten trg dobaviteljev in povečuje elastičnost

²⁴ Stroškom za proizvodnjo dodatnega izdelka.

povpraševanja. Horizontalna razsežnost pa deluje predvsem na trgu, na katerem podjetje svoje izdelke prodaja.

Čeprav je ožji namen te naloge pojasniti, kako izdelati koncept in kodo za spletno trgovino, je splošna obravnava industrijske organizacije pomembna predvsem zaradi proučevanja same organiziranosti podjetja. Zgornje dejavnike smo našteali zato predvsem kot model splošnih sil, ki delujejo na organiziranost nekega podjetja.

Kakšna je vloga informacijskega sistema v stičišču zgoraj navedenih socioekonomsko-organizacijskih dejavnikov, ki se sekajo v samem podjetju? V kolikor razumemo informacijo kot podatek z neko uporabno vrednostjo, potem je informacijski sistem podsistem podjetja za proizvodnjo in zbiranje podatkov o raznih vidikih poslovanja in spreminjanje le-teh v informacije, se pravi umestitve podatkov v nek pomenski okvir. Informacijski sistem je torej v tem smislu neodvisen od računalniške opremljenosti podjetja. Zaključni račun je tako zbirka podatkov z nekimi pomeni, katerih vrednost je popolnoma neodvisna od tega, na kakšen način je npr. računovodja prišel do rezultata, pod pogojem seveda, da je rezultat pravilno izračunan (da so upoštevani veljavni računovodski standardi). Dejstvo pa je, da s pomočjo informacijskih tehnologij proizvedemo veliko več podatkov, jim veliko lažje priredimo ustrezen pomenski okvir in ob tem vse to storimo veliko hitreje. Informatizacija z računalništvom torej povečuje učinkovitost info podsistema vsake organizacije, pa tudi poslovnega podjetja. Informacijski sistem definirajo strokovnjaki na različne načine. Damij (Damij, 1992: 31) tako definira informacijski sistem kot: množico ljudi, strojev, idej, aktivnosti, podatkov in postopkov, ki skupaj omogočajo pridobivanje koristnih informacij. Gre torej za celoto človeških in materialnih faktorjev, ki skupaj proizvajajo informacije. Tako opredeljen informacijski sistem izvaja naslednje aktivnosti (Damij, 1992: 31):

1. Sprejemanje podatkov,
2. Obdelava podatkov,
3. Izpis informacij.

Videti je, da je glavna naloga sistema proizvodnja informacij, ker se z njo nekako zaključi krog, ki ga začenja zajemanje podatkov. Podatku kot osnovnemu elementu informacijskega sistema posveča po mojem mnenju malce premalo

pozornosti. Definira ga kot (Damij,1992: 20) "neko golo dejstvo". In to je vse, kar pove o podatkih. Več pozornosti namenja informaciji in v zvezi z le-to ugotavlja naslednje (Damij,1992: 20): "Podatki postanejo informacije takrat, ko jih transformiramo v komunikacijski pomen, znanje, ideje ali določene zaključke". Gre torej za semantično definicijo informacije, ki je tako definirana kot podatek s pomenom (dodatno: uporabnost, namen). Informacije naj bi imele po njegovem mnenju naslednje attribute: natančnost, oblika, pogostnost, širina, izvor, časovni interval, primernost, popolnost in pravočasnost. Za informacijski sistem je seveda pomembna tudi njegova meja. Definirana je v smislu ločevanja od zunanjega okolja. V zvezi s tem ugotavlja (Damij,1992: 6):

"Meja sistema izolira sistem od njegovega okolja. Sistem obstaja znotraj določene meje in katera koli stvar, ki je zunaj te meje, spada k okolju. Torej meja sistema določa, kaj sodi k sistemu in kaj ne. Iz tega sledi, da je meja sistema določena s strukturo sistema."

Meja je tako nekako strukturni del sistema. V sistem vstopajo podatki preko standardnih vhodov in iz njega izstopajo preko izhodov. V kolikor opazujemo sistem samo s stališča vhodov in izhodov, potem zreduciramo interakcije sistema z okoljem na vhode in izhode. To je seveda v praksi edino koristno in učinkovito, vendar nastopijo težave, kadar gre za vzpostavljanje in načrtovanje informacijskega sistema. Izgubimo namreč njegovo povezanost z okoljem. Meja sistema zato ni samo območje, kjer se informacijski sistem ločuje od svojega okolja, temveč tudi področje, kjer se z okoljem stika, torej področje posrednega vpliva okolja na informacijski sistem. Meja torej ločuje in hkrati povezuje sistem z okoljem. Šele okolje in informacijski sistem tvorita namreč skupaj širši, "globalen" sistem. Tako stališče nam omogoča odgovoriti na vprašanje, kaj naj informacijski sistem sploh obdeluje oziroma kaj naj vanj "spuščamo". Osnova vsakega informacijskega sistema je namreč njegova selektivnost. To pa določa ravno odvisnost sistema od svojega okolja.

Kovačič in Vintar se tej težavi deloma izogneta z ugotovitvijo, da predstavlja informatika v smislu informacijskega sistema "infrastrukturno področje, nujno potrebno za uspešno delovanje organizacije v smislu zagotavljanja njene konkurenčne prednosti" (Kovačič,Vintar:1994,23). Zahtevata tudi, da je informatika sestavni del strateškega podjetniškega načrtovanja, ki opredeljuje

poslanstvo, cilje in strategijo organizacije. Zanju je torej informacijski sistem sestavni del proizvodnega procesa oziroma njegov informacijski podsistem. Pri njiju torej meja informacijskega sistema istočasno povezuje in ločuje okolje ter sam sistem. Zato lahko v nadaljevanju ugotovita, da se začne načrtovanje in razvijanje informatike v organizaciji z opredelitvijo poslovnih funkcij in procesov. Na tej podlagi se nato določijo strateške funkcije in procesi in šele nato pride na vrsto oblikovanje informacijske infrastrukture (Kovačič, Vintar:1994,23). Če poenostavimo: siceršnji proizvodni proces določa informacijski sistem. Kakšna je v zvezi s tem vloga podatka kot osnovnega elementa informacijskega sistema? Kovačič in Vintar ga definirata tako: "Podatek je opis nekega pojava ali dejstva in je lahko predstavljen v številčni, tekstovni ali grafični obliki." Semantične razlike med podatki in informacijami ne obravnavata in je zato semantika podatkov zanju podatek o podatku. Pojma podatek in informacija se v tem smislu dokaj prekrivata. Tak pristop se deloma ujema s stališči klasične informatike kot znanosti. Ta običajno obravnava informacijo kot enoznačno vrednost, ki lahko vsebuje le eno od dveh možnosti. Ključna pri tem je razsežnost ali velikost informacije, ki je zaradi dvojnosti izbire lahko samo potenca števila 2.

Kot vidimo, Kovačič in Vintar pravilno ugotavljata, da je podatek neka vrednost in da je vnos podatkov odvisen od širšega proizvodnega sistema. Pri informacijskem spremljanju proizvodnih procesov in funkcij si pomagata s pojmom modela oziroma modeliranja informacijskega sistema. Model je zanju "vedno preslikava neke predstave ali nekega pogleda na stvarnost, ne pa tudi same stvarnosti"(Kovačič, Vintar:1994,69). S pomočjo modela poslovnih funkcij in procesov ustvarimo informacijski model in ga nato praktično izvedemo v fizično obliko (baze podatkov, aplikacijski sloj ipd.). Ta predstavitev je sicer zelo poenostavljena shema, ki zaradi omejenosti prostora komajda še ustreza kompleksnosti, ki sta jo oblikovala oba raziskovalca. Kot vidimo, naj bi torej bilo modeliranje s pomočjo modelov poslovnih funkcij in procesov in s tem povezanih modelov informacijskih tokov tisto "orodje" selekcioniranja neskončne množice podatkov, ki naj vstopajo v sistem. Informacijski sistem naj bi torej sprejemal in obdeloval ter kasneje predstavljal tiste podatke, ki jih model pokaže kot pomembne. Informacijski model naj bi oblikovala skupina strokovnjakov podjetja

z različnih področij, sam postopek pa naj bi se vodil na podlagi različnih CASE študij. Pa vendar oba raziskovalca ugotavljata (Kovačič, Vintar:1994,165):

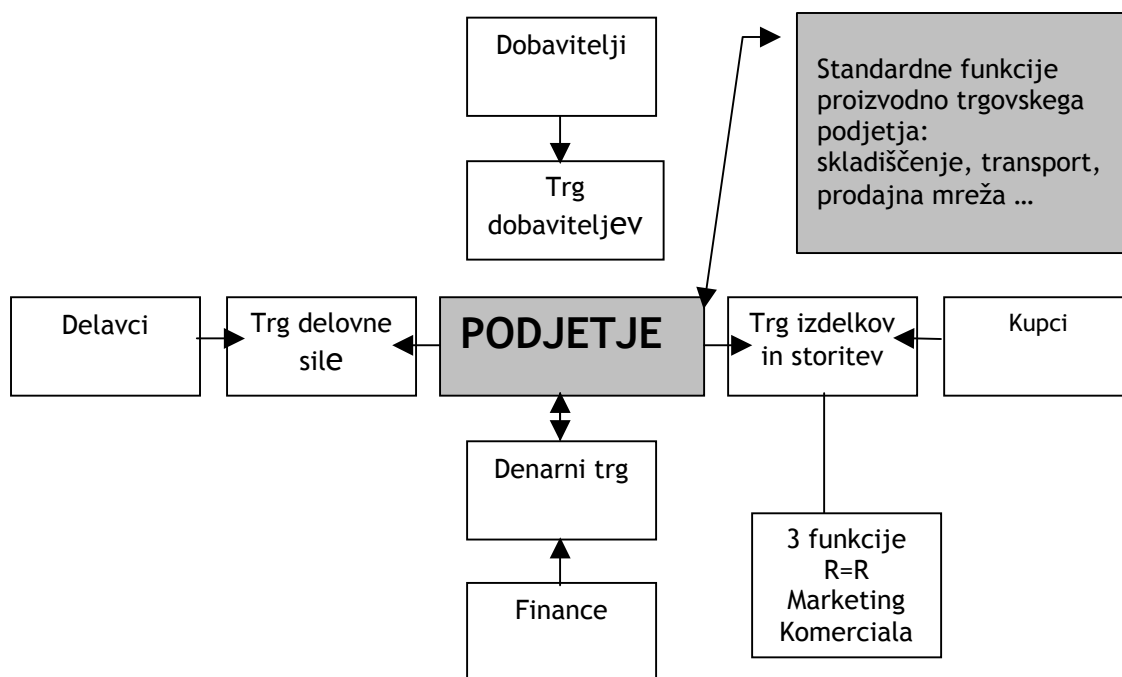
"Često ugotavljamo, da uporabnik nekaj zahteva, drugo želi, potrebuje pa nekaj tretjega."

Prav ta, sicer visoko usposobljen uporabnik, pa naj bi sodeloval v projektni skupini, ki naj bi vodila oblikovanje informacijskega sistema. Ti strokovnjaki naj bi bili zmožni razlikovati med svojimi zahtevami in željami na eni in realnimi potrebami na drugi strani. Zdi se mi, da je to precej nerealno pričakovanje, česar se zavedata tudi Kovačič in Vintar, in ugotavljata, da je potrebno postopke modeliranja standardizirati (vprašalniki ipd.). Vendar pa v bistvu samo prenašata problem na višjo raven, ker ne odgovorita na vprašanje KAKO standardizirati standarde. V praksi zadeva deluje verjetno zaradi izkušenosti in znanja sistemskih analitikov. Kljub temu pa vprašanje standardiziranja standardov ostaja in se rešuje več ali manj intuitivno. Dokler opredeljujemo podatke samo kot takšne ali drugačne vrednosti, iz slepe ulice nezmožnosti standardizacije ni izhoda. Rešitev vidim v dodatni specifikaciji pojma podatek, ki ga moramo pojmovati kot del merskega sistema siceršnjega proizvodnega procesa.

Podatki so torej vrednosti kazalcev meril, s katerimi merimo in opredeljujemo delovanje in stanja samega poslovno proizvodnega procesa.

Šele tako pojmovanje podatkov, nam omogoča oblikovanje standardov, s pomočjo katerih lahko informatiki od strokovno visoko usposobljenih uporabnikov pridobijo mere, s katerimi ti tudi sicer merijo delovanja in stanja poslovno proizvodnega procesa. Na podlagi teh mer in iz njih izvedenih meril in kazalcev teh meril se lahko učinkovito spreminja kazalce v podatke. Pri tem predstavljajo sami kazalci podatke, mere pa predstavljajo pomen oziroma kontekst teh kazalcev.

Ker se vsaka proizvodnja in s tem povezano poslovanje odvija v določeni obliki organiziranosti, me v nadaljevanju zanima odvisnost med organizacijsko strukturo podjetja in z njo povezano informacijsko strukturo. Zgoraj navedeni Poterjev model petih sil bi lahko v zvezi s tem oblikovali bolj kot funkcionalno strukturo in model:



Shema 2.1: Razvojno-poslovne (mikro) funkcije klasičnega podjetja

Trg dobaviteljev sem razdelil na trg delovne sile in trg ostalih dobaviteljev, obdržal sem trg izdelkov, ter posebej vključil denarni trg, ker ta zahteva vodenje svojih evidenc. Svojih neposrednih in potencialnih konkurentov podjetje v svoje dnevno obratovanje neposredno ne vključuje, razen mogoče v primeru spremljanja stanja na tržišču, pri analizi relativnega položaja matičnega podjetja glede na konkurenco (analiza tržnih prednosti). Funkcije klasičnega podjetja so torej neposredno odvisne od trgov, na katerih se to pojavlja bodisi kot kupec ali prodajalec. Informacijski sistem podjetja je logično povezan z beleženjem inputov, ki jih podjetje prejema s posamičnih trgov, ter z beleženjem outputov, ki jih podjetje proizvede in pošlje na trg dobrin in storitev. Seveda so v info sistem, v kolikor gre za rutinsko proizvodnjo, vključeni tudi inputi in outputi samega proizvodnega procesa (informatizacija proizvodnje).

Vsak omenjen trg ima svoje značilnosti. Z ozirom na trg delovne sile informacijski sistem beleži in vodi evidence o plačilih, opravljenih delovnih urah, izračunih davčnih obveznosti, evidence v zvezi z delovnimi pogodbami in odpovedmi le-teh ter seveda šifrante, ki so potrebni za vodenje teh evidenc. V grobem bi lahko navedli nekaj najpomembnejših: šifrant izobrazbene strukture, šifrant kadrov oz. kadrovske evidence, šifranti trajnih delovnih mest, davčni

šifranti ipd. V zvezi s finančnimi trgi so evidence povezane z lastništvu, opremo, amortizacijo, davčnimi obveznostmi, posojili, investicijami, dolgovi, delnicami ipd., ter seveda ustreznimi šifranti. Kot kupec, ki se pojavlja na trgu dobaviteljev, podjetje kupuje surovine in polizdelke oziroma nabavlja materialni in storitveni input. Pri tem proizvaja podjetje s svojim informacijskim sistemom nabavno stran skladiščnega informacijskega sistema, kot prodajalec na trgu storitev in izdelkov pa proizvaja prodajno stran informacijskega skladiščnega sistema (dobavnice, prejemnice, naročilnice in šifranti izdelkov in storitev). Seveda je ves materialni in storitveni input povezan s plačevanjem in izdajanjem računov ter mezd ter s spremljanjem plačil. Informacijski sistem je lahko vključen tudi v ta segment poslovanja preko računovodskih evidenc in v zvezi s tem povezanih šifrantov. Pomemben del informacijskega sistema je seveda tudi beleženje dogodkov, povezanih s samim proizvodnim procesom.

Če poenostavimo, tako lahko razdelimo informacijski sistem podjetja na naslednje področja:

1. Spremljanje glavnega predmeta poslovanja
2. Knjigovodstvo in računovodstvo
3. Prodaja in nabava
4. Skladiščenje
5. Kadrovske evidence
6. Statistične obdelave (standardna poročila, analitska poročila)
7. Podpora odločanju (poročila, podrejena doseganju glavnih ciljev)
8. Oblikovanje, design, DTP.

2.3. Klasično podjetje in razvoj vertikalnih struktur

"Ugotovili smo že, da ob težnji za maksimiranjem dobička podjetje razvija svojo vertikalno strukturiranost. Sam ta proces, zlasti zaradi rasti podjetja, pa pripelje do tega, da se težišče delovanja podjetja pri neki velikosti deloma prenese z maksimiranja dobička na ohranjanje svoje strukture. Dobiček torej ni več edino gibalno delovanje podjetja. Transformacija podjetij od majhnih do velikih je popisana v zgodovini gospodarstva sleherne države. Vzdlž tega historičnega procesa se je razvijala tudi sama ekonomska teorija podjetij in ostale alternativne (socioekonomske in sociološke) teorije gospodarskih organizacij. Ko

je predvsem zaradi spajanja in prevzemov, ne pa toliko zaradi rasti trgov, prišlo do nastanka večjih organizacij in do koncentracije lastništva in upravljanja, se je pričela utrjevati dolgoročna nacionalna tržna struktura, v kateri največja podjetja težijo k ohranjanju za vsako ceno. Pod vtisom tega dogajanja so seveda neoklasično teorijo podjetja, ki bolj "učbeniško" govori le o profitu kot cilju (majhnega in srednjevelikega) podjetja, dopolnile postkeynesianske teorije podjetja, ki so uvedle še druge cilje podjetja mimo dobička. Tako na primer ugotavljajo, da na določeni stopnji razvoja in zlasti pri določeni velikosti podjetja (številčna - zaposlitvena in finančna) rast podjetja prične nadomeščati dobiček kot glavni cilj podjetniške uprave. Kasneje je tudi t.i. transakcijska paradigma podjetja, ki raziskuje, čemu je za podjetja koristno, da se ukvarjajo z vzpostavljanjem partnerskih namesto tekmovalnih odnosov, to ugotovitev le še učvrstila. (več o tem v Lah 2001; Kramberger, Ilič in Kohont 2003).

Sami lastniki podjetij si seveda vedno prizadevajo maksimirati dobiček, vendar pa se z razcepom na poslovodsko in lastniško funkcijo podjetništva pojavijo novi upravljalci podjetij - menagerji, ki podjetja v resnici operativno vodijo in imajo drugačne motive kot lastniki. Cabral (glej Cabral, 2000: 35) se tako sprašuje, ali podjetja res maksimirajo dobičke (naslov podpoglavja). Ugotavlja, da je uresničevanje interesov delničarjev, še posebej, če ima podjetje veliko število malih delničarjev, odvisno predvsem od delovanja in interesov nadzornih svetov, ki imenujejo upravne odbore podjetij, interesi nadzornih svetov pa večkrat konvergirajo z interesi poslovodnih struktur. V nadaljevanju nato ugotavlja - tudi na konkretnih primerih - razlike in učinkovitost pri maksimiranju dobičkov v odvisnosti od različnih plačilnih aranžmajev, ter na koncu resignirano ugotovi, da je pač najučinkovitejše sredstvo pritiska, ki obstaja za poslovodne strukture, morebiten prevzem s strani drugega konkurenčnega podjetja. Edino morebitni prevzemnik ima znanje in informacije, ki mu zagotavljajo, da oceni rezerve v poslovanju, ki jih trenutna vodstvena garnitura ni izkoristila. Jasno je, da bo garnitura ob morebitnem prevzemu izgubila položaje, saj ni izkoristila vseh potencialov podjetja. Če bi jih, bi bila cena podjetja tako visoka, da se ga ne bi splačalo kupiti. Glavni motiv poslovođenja je torej voditi podjetje tako dobro, da se morebitni prevzem ne bo splačal in ne - maksimiranje dobička. *Klasično*

podjetje tako gradi strukture odločanja, katerih motiv je preživetje na daljši časovni rok. Toliko o dopolnjenih motivih delovanja klasičnega podjetja.

Z orisom razvoja podjetij smo želeli predvsem pridobiti vsebinsko osnovo za opredelitev e-poslovanja in ugotavljanje prioritet v izgradnji informacijskih podsistemov. Informacijski sistem klasičnega podjetja torej predvsem spremlja glavni predmeta poslovanja: finančni uspeh (knjigovodstvo in računovodstvo), prodajo in nabavo, skladiščenje, kadrovske evidence, statistične obdelave, podporo odločanju, oblikovanje, design in DTP. Vsako od teh komponent lahko informacijsko spremljamo ločeno ali pa bolj povezano, na primer tako, da informacijski sistem spremlja omenjene vidike tudi s pomočjo WAN²⁵ ali LAN²⁶ omrežnih povezav na eni strani ali pa preko omrežij vodenih preko TCP/IP protokola na drugi strani. Za trenutek odmislimo intranete kot posebno obliko WAN/LAN omrežij, ki so TCP/IP povezana. Zdi se, da lahko govorimo o e-poslovanju takrat, kadar je v zaključenem informacijskem sistemu podjetja omogočen kontroliran input oziroma output v razmerju do drugih poslovnih subjektov preko TCP/IP protokola. Za e-poslovanje, primerno sedanji stopnji razvoja informacijsko-komunikacijske tehnologije, sta torej potrebna dva pogoja:

1. TCP/IP prenos podatkov in informacijskih pomenskih okvirov
2. med različnimi poslovnimi subjekti oziroma informacijskimi sistemi različnih subjektov²⁷.

Povzetek: pri e-poslovanju gre torej za komunikacijo oziroma prenos podatkov in informacij med različnimi zaključenimi informacijskimi sistemi različnih poslovnih subjektov preko TCP/IP protokola.

Intranet je v tem smislu izjema, ki kot vsaka izjema potrjuje pravilo. Pri intranetu gre za TCP/IP protokol, ne gre pa za različne poslovne subjekte temveč za en informacijski sistem, ki je povezan preko TCP/IP protokola.

²⁵ Wide area networks - omrežje širokega območja.

²⁶ Local area networks - lokalna omrežja.

²⁷ Tudi kupec, ki kupuje preko spletne trgovine, ima lasten informacijski sistem, čeprav je to samo Windows okolje in IE brskalnik; ima namreč svoj začasno dodeljen internetni naslov.

2.4. Virtualno podjetje kot ekstremna oblika spletne trgovine - sistemska analiza in primerjava poslovnih funkcij virtualnega in klasičnega podjetja

Ker nameravamo skozi to nalogo oblikovati koncept in razložiti kodo za informacijsko pokritje spletne trgovine, se ne moremo izogniti virtualnim oz. on-line podjetjem kot posebni obliki spletne trgovine in ob tem tudi specifičnim oblikam klasičnega podjetja. Prej smo nakazali funkcije in sestavne dele klasičnega podjetja ter jih navezali na e-poslovanje. Zdi se, da se z uvajanjem virtualnih podjetij nekatere od funkcij klasičnega podjetja krepijo, druge pa umikajo, s tem da se zaradi tega najverjetneje tudi rast tovrstnih podjetij vede drugače kot pri klasičnih podjetjih.

Ob pojavu tovrstnih podjetij se je zdelo, da se iz njih oblikuje predvsem nova oblika prodajne mreže, ki bo povezovala proizvajalce in potrošnike. Nekateri so jih pozdravljali kot marketing gverilo ali najnaprednejšo in ob tem najcenejšo obliko trgovine, ki posluje skoraj brez skladišč in s tem povezanih stroškov skladiščenja in transporta, brez stroškov trgovske mreže za prodajalce in stroškov administracije uprave in marketinga. Storitve, ki jo ponujajo, pa je pravzaprav le posredništvo med proizvajalcem in potrošnikom. Ker za to nalogo niso toliko pomembni ekonomski temelji tovrstnih podjetij, se v le-te ne nameravam podrobneje spuščati. Ugotovim naj le, da je njihov obstoj odvisen od siceršnje marketinške in prodajne mreže drugih, "klasičnih" trgovskih in marketinških podjetij. Ta ustvarjajo prepoznavnost in tržni položaj izdelkov, blagovne znamke, ki jih nato prodajajo tudi virtualna podjetja. Tak, od drugih ustvarjen tržni položaj nekega izdelka nato virtualna podjetja izkoristijo in z zniževanjem cene, natančneje tržne marže, kar lahko najprej dosežejo zaradi manjših stroškov pri oglaševanju kupcem (internet infrastrukturo v veliki meri plačuje država, cena prometa ne vključuje v celoti amortizacije za ICT infrastrukturo), pridobivajo določen tržni delež. Ceno lahko znižujejo še zaradi tega, ker ne vlagajo v marketing izdelkov in ker izdelkov ne skladiščijo in nimajo prodajne mreže, ki predstavlja strošek. Vendar pa brez drugih tudi njih ne bi bilo, oziroma bi morali opravljati vse poslovne funkcije klasičnega podjetja. Kljub temu "parazitstvu" pa prispevajo k produktivnosti trgovske mreže in zniževanju njenih stroškov. V

tem smislu virtualna podjetja lahko pospešujejo prodajo ter med proizvajalci in kupci odigrajo vlogo koristnega mediatorja oziroma brokerja.

Če opazujemo razvitost poslovnih funkcij na prehodu iz klasičnega na on-line podjetje, lahko iz sheme 2.1 (Funkcije klasičnega podjetja) izvedemo primerjavo z virtualnim podjetjem in ugotovimo predvsem redukcijo funkcij klasičnega podjetja na nekatere funkcije, povezane s trgom dobrin in storitev. Vse ostale funkcije so pri virtualnem podjetju zreducirane na minimum in se opravljajo le v meri, ki še zagotavlja obstoj podjetja. Ta redukcija je seveda nujna, če hoče podjetje doseči znižanje cene, na kateri gradijo svojo konkurenčno možnost in tržno nišo.

Druga, zdi se zelo pomembna lastnost teh podjetij, je njihova motivacija za tržno učinkovitost, ki je usmerjena predvsem v doseganje dobička, odločanje za nadaljnje poslovanje pa se vselej zreducira na cost-benefit ekonomijo. Pri klasičnem podjetju postaja preživetje podjetja, zlasti če to raste, čedalje pomembnejši motiv pri odločanju in zato tudi gradi dodatno strukturo odločanja oziroma se vertikalno strukturira. To vertikalno strukturiranje zagotavlja klasičnemu podjetju obstoj v obdobjih negativnega poslovnega uspeha. Pri virtualnih podjetjih pa običajno poslovodska funkcija ni ločena od lastniške (niso vertikalno strukturirana) in lastniki podjetja v njem tudi sami delajo. Meje med mezo in dobičkom se brišejo tako, da se na razporejanje plač gleda kot na delitev dobička. Njihovo preživetje je odvisno le od tega, ali je dobiček ali pa ga ni. Še več, negativna obdobja v poslovanju podjetij ogrožajo fizično eksistenco lastnikov in ti so zato še kako zainteresirani za zaprtje podjetja, če dobička ni. V tem smislu so (vsaj zaenkrat) virtualna podjetja zgrajena predvsem na merilih tržne učinkovitosti, manj pa na merilih dolgoročnega preživetja in dodatnih meril uspešnosti; predvsem ne kopičijo finančnih rezerv, s katerimi bi se zavarovala pred tveganji nihajočih poslovnih ciklov v tržni ekonomiji.

Zaradi vseh teh razlogov si upam potrditi hipotezo, da poteka diferenciacija med on-line in klasičnimi podjetji predvsem po vertikalni zgradbi strukture odločanja, ki skrbi za preživetje podjetja v "hudih časih".

Informacijski sistemi spletnih podjetij - največkrat trgovin - so podrejeni opisanim ciljem: v primerjavi s klasičnim podjetjem so na nekaterih točkah okrepljeni, na drugih pa okrnjeni. Za snovalce spletnih trgovin pa pomeni

snovanje informacijskega sistema za tako podjetje tudi precejšnjo težavo. Spletni katalog izdelkov se sicer poveže z nakupovalno košarico, sama povezljivost s siceršnjim preostalim možnim informacijskim sistemom podjetja pa se običajno predvsem zaradi ozke dobičkonosne usmerjenosti oblikuje bolj na ad-hoc nivoju, od primera do primera različno.

Spletno trgovino oziroma kodo zanjo bomo poskušali zasnovati tako, da bo obstajala možnost vključljivosti v informacijski sistem podjetja. Ne nameravam pa izdelati takšnega informacijskega sistema, ki bi temeljil izključno na spletni komponenti. Zaradi specifičnosti in raznovrstnosti tovrstnih podjetij to presega okvire te naloge. Če bi hotel izdelati spletu povsem ustrezen informacijski sistem, bi moral oblikovati takorekoč program za izdelovanje programov, namenjenih spletnim aplikacijam.

Poleg teh pa obstoji še veliko bolj tehten razlog za omejitev, ki ga bomo natančneje obdelali prav v naslednjem razdelku. Naj kot uvod vanj predpostavim, da poteka vzporedno z razvojem e-trgovanja tudi standardizacija povezljivosti med različnimi programi različnih proizvajalcev. Gre za standardizacijo pri izmenjavi podatkov in informacij, ki jih proizvajajo in distribuirajo različni informacijski sistemi, med različnimi komponentami, kakor tudi med informacijskimi sistemi različnih poslovnih subjektov kot takimi. Tej standardizaciji izmenjave podatkov pa se "mi kot njihovi uporabniki" lahko samo prilagodimo, ker nanjo praktično nimamo vpliva. Dejstvo je, da vodijo uvajanje standardov pri izmenjavi podatkov med informacijskimi sistemi različnih proizvajalcev in različnih poslovnih podjetij prav velika podjetja, globalne transnacionalne korporacije, ki imajo za to početje dovolj tržne moči oziroma denarnih sredstev. Te standarde nato mala podjetja prevzamejo od večjih poslovnih partnerjev kot del njihovega poslovnega sodelovanja in jim prilagodijo svojo lastno programsko opremo. To prilagajanje programske opreme je za manjša podjetja veliko cenejše kot pa morebitno ustvarjanje nove. Zaradi zgoraj navedenih ugotovitev bi pomenila izdelava izvirnega programa za informacijski sistem virtualnega podjetja na nek način tudi ustvarjanje standardov za povezljivost dokumentov. Te standarde pa bi upoštevalo le podjetje, za katero so bili oblikovani; malo je možnosti, da bi jih uporabljala večja podjetja. Standard, ki ni vsaj deloma splošno priznan, pa ni standard.

2.5. e-Commerce

2.5.1. Definicije e-Commerce

Tehnični vidik e-poslovanja smo že uvedli, pogledajmo še njegovo vsebinsko plat. Statistiki in raziskovalci ekonomskih in socioloških fenomenov, povezanih z internetom, so do sedaj in še vedno proizvajajo veliko število definicij e-commerce-a. Dobesedni prevod bi pomenil enostavno e-trgovino ali e-prodajo. Že tu se zaplete. Nekateri namreč menijo, da tako pojmovanje oži raziskovalno polje, ki ga zamejuje termin e-commerce, ter obravnavajo e-commerce bolj na splošno, v smislu e-poslovanja. Drugi pa ugotavljajo, da je ravno razmejevanje e-poslovanja od e-prodaje koristno in da lahko proizvaja koristne spoznavne učinke.

Internet kot nov komunikacijski medij vpliva na številna področja človekovega delovanja in zato lahko pojem e-commerce opazujemo z različnih vidikov. Opredelitve deloma sledijo temu, kateri del poslovnih funkcij klasičnega podjetja se poudarja. O'Daniel (O'Daniel, 1999: 1) tako ugotavlja, da pomeni e-Commerce s tehnološkega vidika informacijsko infrastrukturo, ki podpira razpečevanje informacij, uslug, storitev ali plačil skozi računalniška omrežja, pri tem pa prečka organizacijske meje²⁸. Z ekonomskega stališča predstavlja po njegovem mnenju e-commerce orodje za povezovanje kupcev in prodajalcev. S stališča ved o vodenju predstavlja e-commerce proces aplikacije ICT tehnologij, usmerjen k avtomatizaciji (rutinskih) poslovnih transakcij in postopkov dela. Za tržnike pa predstavlja e-commerce nov kanal za dostavo informacij ciljnemu občinstvu preko oglaševanja, promocijske aktivnosti in verjetno on-line publikacij. Tehnologi posledično pri opazovanju fenomena e-Commerce opazujejo število transakcij, ki se odvijajo med dvema subjektoma, ekonomisti beležijo število plačil, naročil in podobnih ekonomskih dejavnosti, managerji merijo povečanje produktivnosti ali učinkovitosti, ki je povezano z on-line dogajanjem, tržniki pa merijo druge dogodke, klike ipd. Vsaka definicija proizvede svoje specifične rezultate.

Barua povezuje informacijski vidik e-commerce-a z ekonomskim dogajanjem in predlaga (Barua in drugi, 1999: 3) štirinivojski (štiriplastni) model proučevanja tega pojava. Ti štirje nivoji oz. plasti so pravzaprav podnivoji dveh osnovnih ravni dogajanja e-commerce. Od tega predstavljata prva dva nivoja informacijsko

²⁸ Verjetno subjektov udeleženih v e-commerce izmenjavi.

strukturo, zadnja dva pa ekonomsko dogajanje. V zvezi s prvo plastjo proučuje in meri samo internetno infrastrukturo oziroma internet kot omrežje. Naslednjo plast predstavlja po njegovem mnenju softverska komponenta, se pravi, da je to nivo programov oz. aplikacij. Sem spadajo še učenje, usluge integriranja sistemov in svetovanja. Ekonomska plast je prav tako razdeljena na dva podnivoja in sicer nivo ekonomskih transakcij in nivo posredovanj.

Vehovar (Vehovar, 1999: 5) za razliko od Barue poskuša natančneje opredeliti pojem e-commerce skozi razlikovanje podobnih kategorij. Tako razmejuje pojem e-commerce od trgovine na internetu in ugotavlja, da prihaja pri takem pojmovanju e-commerce do preučevanja obsegov dogajanja na trgih, povezanih z internetom kot takim (npr. promet, ustvarjen skozi prodajo oblikovanja spletnih strani ipd.). Prav tako upravičeno razmejuje e-commerce od e-mail naročil ter opozarja na nevarnost, ki jo predstavlja zoževanje pojma e-commerce na EDI izmenjavo (Electronic Data Interchange), na on-line finančne transakcije in on-line nakupe. Resne probleme lahko predstavlja tudi lokacija on-line nakupa²⁹.

Raziskovalec Avstralskega statističnega urada McGeachie pa ima do definicij e-poslovanja in e-commerce odklonilno stališče. Tako definicijam obeh omenjenih konceptov nasprotuje najprej s praktičnega stališča in ugotavlja, (McGeachie, 1999: 1) da sta termina e-commerce in e-poslovanje, četudi pogosto uporabljana in pogosto upoštevana kot ključna koncepta za merjenje, smatrana za "ne preveč koristna". V nadaljevanju ugotavlja, da če zožimo npr. termin e-commerce na čisto transakcijsko analizo, zaobsega ta pojem številne dejavnosti, ki na poslovno življenje različno vplivajo. Če hočemo zagotoviti učinkovito analizo vplivov teh novih elektronskih orodij, potem moramo razčleniti (subdivide) e-commerce oz. e-poslovanje na smiselne komponente, kar šele ustvarja temelje za tako analizo.

Zdi se, da izhajajo težave pri opredeljevanju pojma e-commerce in e-poslovanja iz kompleksnosti samega pojava. Preko njega se prepletajo aktivnosti z različnih področij, od hardverskih in softverskih pa do ekonomskih, komercialnih, marketinških in celo socioloških, če naštejemo samo nekaj glavnih. Kljub tej definicijski raznovrstnosti pa se pri praktičnem delu statistiki lažje poenotijo okoli glavnih kriterijev za opazovanje pojava in merijo prav komponente, ki smo jih

²⁹ Če kupi npr. Finec izdelek v ZDA. Je to finski ali ameriški e-commerce dogodek?

zgoraj povzeli po Vehovarju, s tem, da obravnavajo e-commerce kot celovit pojav, in te kazalce jemljejo bolj kot indikativne komponente pojava.

Za **potrebe te naloge** lahko operacionaliziramo oz. zožimo pojem e-commerce na e-trgovanje, če se pri tem zavedamo, da je e-trgovina podsistem e-poslovanja. Če gre torej pri e-poslovanju za komunikacijo oziroma prenos podatkov in informacij med različnimi zaključenimi informacijskimi sistemi različnih poslovnih subjektov preko TCP/IP protokola, potem gre pri e-commerce (v nadaljevanju e-trgovanje) za *komunikacijo oziroma prenos podatkov in informacij med različnimi zaključenimi informacijskimi sistemi različnih poslovnih subjektov preko TCP/IP protokola v zvezi z komercialnimi dejavnostmi podjetij*. e-trgovanje torej opredeljujejo trije dejavniki:

1. prenos informacij med različnimi informacijskimi sistemi različnih podjetij,
2. TCP/IP protokol,
3. komercialne dejavnosti.

S tem smo nekako zaključili krog primerjav med klasičnim in virtualnim (prodajnim) podjetjem in se, z zožitvijo virtualnega podjetja na tehnično podlago ter komercialno dejavnost, vrnili na izhodišče, ko smo opazovali informacijski sistem neke industrijske organizacije v odvisnosti od njenih funkcij z ozirom na trge, na katerih deluje. Virtualno podjetje, ki se ukvarja z e-trgovanjem, je posebna oblika tržnega podjetja, ki se zaradi posebnih okoliščin, v katerih deluje, razvija predvsem v horizontalni, skoraj nič pa v vertikalni smeri.

2.5.2. e-trgovina in e-trgovanje

Ker vse industrijske organizacije oz. podjetja proizvajajo nek proizvod, morajo z njim na tak ali drugačen način skozi abstraktni mehanizem, ki mu pravimo tržišče. Tržišče preko sil ponudbe in povpraševanja oceni relativno vrednost izdelka in ga preko kupcev tudi plača. Na različnih trgih se opravljajo različne (dobaviteljske, komercialne in razvojno-marketingške) dejavnosti. Za potrebe te naloge bomo zožili področje opazovanja na B2C trge in pri tem upoštevali, da na nekaterih trgih B2C lahko nastopajo kot končni potrošniki izključno podjetja. e-trgovina je v tem smislu torej tisti segment e-trgovanja, na katerem se odvija

neposredna prodaja proizvedenih izdelkov preko TCP/IP povezave med različnimi informacijskimi sistemi prodajalcev in kupcev.

Za nalogo je pomemben predvsem informacijski sistem prodajalca, ki je v to dogajanje vpleten. Pred podjetjem, ki se odloča za odprtje spletne trgovine, se odpirajo z ozirom na stopnjo spletne vključenosti siceršnjega informacijskega sistema podjetja različne možnosti, od popolne ločenosti informacijskega sistema od spletne trgovine do popolne integracije obeh, v kateri je spletna trgovina le del informacijskega sistema podjetja. Med obema skrajnostma obstajajo še številne vmesne oblike. Obe skrajnosti sta na nek način idealni model, skozi katerega lahko ocenjujemo raznovrstnost praktično uveljavljenih oblik, skozi katere se v večji ali manjši meri oba modela udejanjata.

2.5.3. Integracija informacijskih sistemov

Na tržišču se pojavljajo številni proizvajalci programske opreme za podjetja. Programi posameznih programskoproizvodnih podjetij pa zapisujejo podatke v več ali manj lastnem formatu, ki ga lahko včasih razume le druga aplikacija istega podjetja. Razlike so posledica konkurenčne faze tekmovanja na globalnem trgu proizvajalcev programske opreme. Aplikacije različnih softverskih podjetij istih podatkov zato večinoma ne razumejo in so aplikacijsko neizmenljivi, ker je format, v katerega so vkalupljeni podatki, popolnoma "subjektivno" oblikovan.

Za primer uporabimo dve podjetji, ki poslovno sodelujeta. V tem odnosu nastopa prvo, recimo mu podjetje A, kot dobavitelj, drugo, podjetje B pa kot kupec. Obe podjetji uporabljata na primer "Accessovo" bazo podatkov (Microsoftov patent) in zaradi tega bi morali biti podatki o prodanem oziroma kupljenem izdelku z lahkoto izmenljivi³⁰. Pa vendar se izvorna programa obeh partnerskih podjetij ne bosta mogla sporazumeti zaradi nekompatibilnosti šifrantov, po katerih se oblikujejo posamične tabele v bazi. V prvem se bo tako ime izdelka vodilo kot npr. "izdlme", v drugem pa drugače, npr. kot "imelzd". In ko bo skladiščni program kupca hotel vpisati ime izdelka, bo med podatki iskal "imelzd" in bo spregledal "izdlme" kot ime izdelka pri dobavitelju - če ta povezava ni vnaprej predvidena.

³⁰ Načeloma so z isto lahkoto izmenljivi tudi podatki med bazami različnih proizvajalcev.

Ker so programske hiše med seboj neodvisne in se med seboj srečujejo na tržišču, jim je celo v interesu, da so podatki med seboj nekompatibilni. V primerih vertikalnih integracij podjetij namreč prodajo več programske opreme. Po drugi strani pa bi bila kompatibilnost programske opreme tržna prednost za potencialne kupce in prodajalce, saj bi programske aplikacije, ki lahko prepoznajo tuje podatke, lahko dosegale višjo ceno in imele več zainteresiranih kupcev.

Pri ocenjevanju dejanske stopnje integriranosti e-trgovine v preostali informacijski sistem podjetja lahko opazujemo na dva načina:

- Horizontalna povezanost informacijskih sistemov med različnimi komponentami informacijskega sistema nekega podjetja. Pri tem me zanima predvsem, ali aplikacije, preko katerih vodimo glavni predmet poslovanja, knjigovodstvo in računovodstvo, prodajo in nabavo, skladiščenje, kadrovske evidence, statistične obdelave, podporo odločanju, oblikovanje, design in DTP³¹, omogočajo prenašanje podatkov med seboj. Na konkretnem primeru bi to pomenilo, ali se npr. dogodki, knjiženi skozi prodajo, odrazijo v evidencah aplikacije, ki vodi skladiščno poslovanje. Podrobneje: modul spletne komponente informacijskega sistema bi razknjižil zalogo v skladišču, pripravil račun z dobavnico in ju "knjižil", pripravil etiketo z naslovom naročnika, ter v kasnejši fazi realizacije transakcije spremljal plačila in ob morebitnem neplačilu izpisal opomin. Plačilo bi ob tem potekalo on-line.
- Vertikalna povezanost informacijskih sistemov med različnimi informacijskimi sistemi različnih proizvajalcev v verigi, ki proizvaja končen izdelek. Tako bi npr. spletna trgovina ob prejemu naročila za izdelek, ki ga je v skladišču zmanjkalo, avtomatsko izpolnila naročilnico in jo poslala dobavitelju (že) naročenega izdelka. Po tem, ko bi dobavitelj dostavil blago, bi vertikalno integriran informacijski sistem popravil zalogo v skladišču in kasneje poskrbel za on-line plačilo računa, ob tem pa še knjižil vse vmesne dogodke.

Računalniška izmenjava podatkov (RIP) postaja z razvojem informatike nuja, tako da se zdi, da šele RIP postavlja temelje pravi informatizaciji družbe. Ker gre

³¹ DTP desktop publishing, izraz bi lahko prevedli v slovenščino kot namizno založništvo.

za splošen interes, se mora vsaj v začetni fazi vključiti država. Povezljivost oziroma izmenljivost podatkov programske opreme lahko zagotovijo:

1. Državni in meddržavni organi, ki so zadolženi za standardizacijo,
2. Velike programske hiše, ki same ne proizvajajo konkretnih poslovnih aplikacij, oziroma mednarodne organizacije, ki določajo specifikacije za programsko opremo in orodja(npr.: W3C).

Različne meddržavne organizacije so že oblikovale standarde za format poslovnih sporočil s slovarji, ki določajo opis in vsebino dokumentov. Eden od takih standardiziranih opisov dokumentacije in podatkov v podjetjih je že omenjeni EDI oz. UN/EDIFACT. Podatki, ki jih dostavlja dokument EDI, so semantično primerljivi in jih lahko uporablja katerakoli aplikacija na svetu ne glede na programski jezik, v katerem je napisana, pod predpostavko, da razume format EDI. Uvajanje dokumentov EDI pa je zaenkrat kljub najboljšim namenom še zelo drag proces, saj je struktura EDI dokumentov zelo zapletena in zahteva seznanjenost programerjev s formatom (krivulja učenja je zelo položna). Prednost in hkrati tudi največja slabost uvajanja EDI je možnost različnega pristopa podjetij k temu kvazi-standardu. Če na primer eno podjetje neodvisno od svojih poslovnih partnerjev uvaja EDI v svoje poslovanje, lahko pride namreč do situacije, v kateri ima podjetje vpeljan sistem EDI, njegovi poslovni partnerji pa se zaradi stroškov, povezanih z uvajanjem, za standardizacijo EDI (še) ne odločijo. Posamično podjetje torej lahko doseže izmenljivost podatkov, pa jih nima s kom izmenjevati. Če gre za večje podjetje, potem lahko včasih vsili svojim partnerjem, da npr. izpolnjujejo dobavnice EDI ipd., vendar pa informacijska sistema v bistvu še nista usklajena (dvojno delo za podrejena podjetja!). Podjetje z delujočim sistemom EDI bo imelo sicer zelo učinkovit informacijski sistem in zaradi tega v administraciji zelo malo zaposlenih, gledano s perspektive njegovega okolja, v katerem deluje, pa bo predstavljalo le osamljen otok učinkovitosti. Podjetja, s katerimi sodeluje, recimo da od njih nabavlja surovine in polizdelke, bodo imela še vedno neučinkovit informacijski sistem in posledično tudi dražje polizdelke, kar se bo seveda odražalo na končni ceni končnega izdelka, ki ga na primer proizvaja podjetje s prilagojenim informacijskim podsistemom

EDI. Vprašanje je, če povečanje produktivnosti zaradi omejene uporabe EDI upravičuje strošek, povezan z investicijo v vpeljavo tega sistema.

Računalniško izmenjavo podatkov omogoča tudi jezik XML. XML je tako kot HTML izveden iz SGML-ja (Standardizirani splošni označevalni jezik), vendar za razliko od označb HTML, ki označujejo obliko dokumentov, vzpostavlja datotečni format za predstavitev podatkov oziroma shemo za opis podatkovnih struktur in mehanizem za razširitev jezika HTML s semantičnimi informacijami. Če z označbami HTML določamo obliko zapisa, pa z označbami XML določamo vsebino oziroma pomen zapisa. Če se torej dva uporabnika podatkov, to sta lahko poslovna partnerja, dve programski aplikaciji ipd., dogovorita o pomenu nekega podatka, ga lahko med seboj izmenjujeta. Uporabnika sta na nek način ustvarila standard, ki pa velja samo med udeleženci takšnega dogovora. XML omogoča ustvarjanje takih, recimo temu "omejenih" standardov.

Prilagajanje siceršnjih aplikacij za uvoz in izvoz XML posredovanih podatkov za programerje ne bi smelo predstavljati večjega problema. Medtem, ko je uvajanje sistema EDI zelo zahtevno dogajanje, je standardiziranje preko označevalnega jezika XML izredno lahko. Omogoča tudi vertikalno povezovanje proizvajalcev od surovine do izdelka in s tem povečanje produktivnosti po celotni vertikalni strukturi. Če uvede podjetje v svoje poslovanje EDI standarde, se praktično prilagodi poslovanju sveta, čeprav posluje s konkretnimi partnerji, s katerimi pa na ta način ne more poslovati. Uvajanje delnih standardov pa omogoča prav sodelovanje med že obstoječimi poslovnimi partnerji, katerih podatkov neudeleženci v sistemu ne razumejo, vendar pa to načelno informacijsko razumevanje niti (še) ni potrebno.

Skratka, uvajanje standardizacije podatkov preko XML omogoča relativno poceni vertikalno in tudi horizontalno integracijo informacijskih sistemov. Sama standardizacija pa se z lahkoto, na način okužbe, širi na nove člane skupnosti, ki si izmenjuje podatke. XML je tudi ena od bistvenih sestavin arhitekture .NET, kot jo je zasnoval Microsoft, ki je pripravil tudi knjižnice za lahko ustvarjanje in obdelavo dokumentov XML. Uvoz dokumentov XML se lahko omogoči z moduli k siceršnji programski opremini. Izredno pomembno je po mojem mnenju dejstvo, da si XML in EDI ne stojita nasproti kot dva izključujoča si pola, temveč predstavljata bolj dva bregova iste reke. Ravno XML namreč omogoča izmenjavo strukturiranih

podatkov EDI v omejenem obsegu. EDI se pa po drugi strani vzpostavi na nekem omejenem delu informacijskega sistema, na drugem pa ne. Na drugih delih informacijskega sistema poslovnega partnerja običajno niti ni potreben.

Ne glede na to, da bo razvoj v prihodnjih letih verjetno pripeljal do vertikalne in horizontalne integracije preko XML prav do EDI, pa v sedanjih razmerah popolna integracija (spletnih in ostalih komponent informacijskega sistema) ni nujno tudi idealna rešitev.

Naj navedem še nekaj dejavnikov, zaradi katerih je ločenost obeh sistemov (siceršnjega informacijskega sistema in komponente za spletne trgovine) lahko koristna za podjetje:

- varnost in zaščita pred vdori
- fizična lokacija spletne trgovine in stroški
- lasten spletni strežnik in stroški
- najeti vodi in stroški
- vzdrževanje sistema in stroški
- programska oprema za integracijo in stroški
- cena za produktivnost, ki jo prinese integracija
- ali je vertikalna povezljivost sploh mogoča?
- za normalno obratovanje sistema je potrebno znanje

2.5.4. Sklep

Za malo (virtualno) podjetje je zaradi zgoraj navedenih dejavnikov v začetni fazi najsmotrnejša ločenost informacijskega sistema od sistema spletne trgovine. Vendar pa mora biti zasnova same trgovine tako zastavljena, da omogoča lahek prehod na integracijo s siceršnjim informacijskim sistemom podjetja. Ta prehod je potrebno opazovati z dveh vidikov in sicer z vidika inputa podatkov (šifranti, podatki o izdelkih in cenah) v spletno trgovino in z vidika outputa, ki ga ta proizvede (naročilo in podatki o naročniku). Inputa podatkov v aplikacijo spletne trgovine ne moremo vnaprej zagotoviti, če ne poznamo siceršnjega formata podatkov v informacijskem sistemu bodočega naročnika. Vnaprej namreč ne moremo vzpostaviti standardov z obstoječo programsko opremo bodočega naročnika, če naročnika in njegove programske opreme sploh ne poznamo. Za vzpostavitev standarda sta namreč potrebna vsaj dva, ali pa zahtevamo od

programerjev bodočih naročnikov, da prilagodijo obstoječo opremo našim standardom. Isto velja za output, ki ga proizvede naročilo. Menim, da je veliko lažje prilagajati spletno trgovino siceršnjemu informacijskemu sistemu dejanskega uporabnika kot obratno - zgraditi idealno spletno trgovino za vse potencialne uporabnike.

Na osnovi dosedanjega premisleka bom zasnoval aplikacijo spletne trgovine tako, da bo prilagoditev na nove dobavitelje in kupce oziroma na njihove informacijske rešitve možna, ne bom pa je vsiljeval s popolnoma izdelanim sistemom. Na tak način bomo omogočili sprotno prilagajanje specifičnim potrebam naročnika, še posebej, ker samo prilagajanje v bistvu ni težko izvedljivo.

3. Značilnosti interneta kot okolja za delovanje spletnih aplikacij

3.1. Uvod

V tem delu naloge nameravam na kratko opozoriti na tiste značilnosti internetnega okolja, ki bistveno opredeljujejo izbire programerja aplikacije glede:

1. Programskega jezika, v katerem bo napisana koda,
2. Strežniško odjemalske (brskalnik kupca) komunikacije.
3. Dela aplikacije, ki bo tekel na strežniški strani, kar tudi do neke mere določa izbiro programskega jezika strežniškega dela aplikacije.
4. Strežniškega operacijskega sistema.

Pri tem se ne nameravam spuščati v podrobnosti, temveč bom obdelal področje le v tolikšni meri, kot je potrebno za to, da opozorim na vzroke, ki opredeljujejo izbire.

3.2. TCP/IP, HTTP in koncept strežnik-odjemalec

Internet se od običajnega omrežja ne razlikuje po številu strežnikov, ki sodelujejo med seboj. Če bi ugotovili, da je Internet največje obstoječe omrežje in da je to tisto, kar ga določa kot internet, potem bi bili v zmoti. To kar ga ločuje od navadnega omrežja, je predvsem njegova programska in strojna različnost, heterogenost, le-to pa omogoča sklad TCP/IP. Zato lahko trdimo, da je prav sklad TCP/IP tisto, kar internet v največji meri opredeljuje. V omrežju, kot je internet, je veliko različnih računalnikov in tudi veliko različnih povezav med njimi. Da bi si lahko med seboj izmenjavali podatke, je nujno, da vsi govorijo isti jezik. Tega pa definira skupek pravil, ki mu pravimo protokol (Mesojedec, 1996: 41). Povezava med dvema računalnikoma na internetu pa ne temelji samo na enem protokolu, temveč je protokolno razslojena. Govorimo običajno o štirih slojih in sicer:

1. Povezovalni sloj (vodi - fizični ali drugačni)
2. Omrežni
3. Prenosni sloj oz. kontrolni sloj
4. Aplikacijski sloj.

Povezovalni sloj je sloj fizičnih povezav med računalniki, te so lahko optične, ADSL, telefonske, brezžične ipd. Povezovalni sloj je zadolžen za prenos informacij med prejemnikom in oddajnikom. Odvija se po protokolih oz. gonilnikih, ki so vezani na strojno opremo. Ta sloj v sebi lastnem, strojnem formatu prenaša informacije zgornjega - omrežnega sloja IP (internet protokol) sloja. Pri slednjem (IP) se začne neodvisnost interneta od strojne opreme, ker se prejemniki ne identificirajo več po strojni, temveč po univerzalni shemi, neodvisni od strojne opreme. Po tej shemi se računalniku na internetu dodeli **naslov**, v obliki 32-bitnega števila, razdeljenega po bajtih na štiri zloge, razdeljene s piko (dotted quad notation). Na tem sloju se vzpostavlja univerzalnost komunikacije internet protokola (IP). **Osnovno načelo delovanja tega sloja je načelo prenosa brez trajno vzpostavljene povezave (Connectionless oriented protocol).** Prenosni sloj oziroma TCP (transmission control protocol) je, kot ime pove, zadolžen za kontrolo prenosa podatkov. Sam internetni protokol oziroma omrežni sloj je, kot ugotavlja Kirch, **po zasnovi nezanesljiv** (kar je logično glede na to, da je **Connectionless oriented protocol**). Zanesljivost mu daje ravno TCP protokol. Aplikacijski sloj je tisti sloj, s katerim se običajno srečuje uporabnik preko svoje aplikacije. To je sloj HTTP, FTP, SMTP in drugih internetnih protokolov aplikacijskega sloja. Preko tega sloja namreč potekajo komunikacije med aplikacijami. Za nalogo je pomemben ta četrti,³² protokolni *aplikacijski sloj in načelo prenosa podatkov brez trajno vzpostavljene povezave*, zato bomo s TCP/IP povezovanjem zaključili.

Aplikacijski sloj je tisti sloj, s katerim ima programer spletne trgovine pravzaprav največ opraviti, in ta sloj je tisti, ki določa najpomembnejša pravila, komunikacije, ki jih mora upoštevati. Na tem sloju se šele pojavi delitev med računalniki v internetnem sistemu na odjemalce in strežnike³³. Zakaj gre pri tej delitvi? Uporabniki preko svojih aplikacij z računalnikom odjemalcem od računalnika strežnika zahtevajo določene podatke. Na uporabniški strani imamo neko aplikacijo (klienta - odjemalca), ki preko aplikacijskega sloja in po pravilih tega sloja vzpostavi povezavo z nekim računalnikom (strežnikom), ki ima podatke in ta ji podatke tudi pošlje. En strežnik lahko streže številnim klientom. Za nas

³² Oziroma prvi, ker je vrstni red odvisen od strani s katere se slojevitost protokolov opazuje.

³³ Server/client se pravzaprav pojavi prvič pri IP protokolu z strežniki DNS, vendar to za nas ni pomembno.

najpomembnejši klient je brskalnik (IE, Navigator, Opera ipd.) na računalniku morebitne stranke, ki si ogleduje spletno trgovino. Mesojedec to pojasnjuje veliko bolj precizno takole (Mesojedec, 1996: 50):

"Aplikacijski sloj je tisti del omrežne arhitekture, s katero neposredno komunicira uporabnik. Zanj se programi, ki delujejo na tem sloju, vedejo kot storitve (poudaril MU) omrežja. Vse internetovske storitve pa so organizirane po načelu odjemalcev in strežnikov (poudaril MD)."

Pri tem je potrebno poudariti, da poteka s strežniške strani le priprava in pošiljanje podatkov v skladu z navodili odjemalca. Sam odjemalec pa lahko v nadaljevanju opravi obdelavo dobljenih podatkov. Odjemalec je v tej komunikaciji prav tako aktiven kot strežnik, le da dobiva podatke za svoje obdelave od strežnika. To dejstvo nam omogoča "client side scripting" oziroma izvajanje skriptov na odjemalčevi strani. Bodoča spletna trgovina bo torej neke vrste storitev strežniškega programa na nekem strežniku in bo kot taka na razpolago odjemalcem (user agentom-browserjem, programom za elektronsko pošto ipd.). Storitve ponuja strežnik na določenih vratih (vhodih v gostiteljski sistem), ki so standardizirana. Če torej želimo od nekega strežnika neko storitev, potem najprej navedemo za katero storitev gre (po katerem protokolu jo dobimo), nato strežnikov naslov in nato vrata na katerih se storitev nudi. Če želimo npr. s strežnika prenesti neko datoteko (file transfer protokol), potem izgleda to nekako tako:

Protokol --> strežnikov naslov: --> vrata/ --> pot do datoteke na strežniku

<ftp://193.3.311.123:21/datoteka.dat>

ali <ftp://mojadomena.si/datoteka.dat>

Vrata lahko izpustimo, če gre za privzeta vrata izbranega protokola. User agent programi vedo, na katera vrata naj naslovijo zahteve. Na ta način smo prišli do splošnega krajevnikarja ali URL (Uniform Resource Locator), z njim na internetu določimo protokol, strežnik ter pot in datoteko povezano z neko storitvijo. WWW lahko definiramo kot z URL definirana strežniška mesta (site) in krajevne (datotečni sistem) vire, dosegljive preko spletnih brskalnikov po protokolu HTTP. Ker omogoča protokol HTTP prenašanje in ogled spletnih strani in se spletna trgovina lahko postavi samo kot spletna stran, bom opozoril na nekaj njegovih značilnosti. Glede na to, da so vse internetne storitve organizirane po

načelu odjemalcev in strežnikov, je po tem načelu organiziran tudi protokol HTTP, preko katerega spletni strežnik dostavi brskalniku spletno stran.

Komunikacija med obema poteka v štirih fazah:

1. Povezava (connection). Odjemalec se poveže s strežnikom na vratih 80. Strežnik sprejme povezavo.
2. Zahteva (Request). Odjemalec pošlje zahtevo po vsebini.
3. Odgovor (Response). Strežnik posreduje zahtevano vsebino ali pa sporočilo o napaki, če zahtevi ni mogoče ugoditi.
4. Odklop (disconnection). Po prenosu zahtevane vsebine strežnik odklopi povezavo. (Mesojedec, 1996: 55)

Glede na to, da poteka komunikacija preko omrežnih povezav, je za posredovanje zahtev in odgovorov nanje *potreben določen čas*. Da bi z vidika uporabnika program tekel čim hitreje (brez prekinitev), je najbolje, da se od strežnika poberejo samo nujni podatki, za obdelavo le-teh pa naj bi zadolžili skripte, ki se do brskalnika prenesejo skupaj s spletno stranjo.

Kot vidimo, poteka komunikacija v obliki zahteve, ki se ji ugotovi z odgovorom, ki mu sledi **odklop**. Strežnik ugotovi vsaki zahtevi zase in torej **ne pomni izida zahtev (stateless protocol)**. To pomeni, da pošlje podatke in takoj za tem pozabi na vse skupaj. Deloma je to povezano z zgoraj navedenim connectionless načelom protokola IP. To dejstvo pa je ključnega pomena za izdelavo spletne trgovine. Zakaj? Ker si strežnik ne zapomni izida zahtev, mu ni mogoče npr. ukazati: "Poglej na stran, ki si jo poslal npr. dve zahtevi nazaj in pošlji ime izdelka št. 5 po protokolu mailto na ta in ta e-mail naslov, dodaj še ceno izdelka in naslov IP s katerega je prišla zahteva." Strežnik ne more pogledati na stran, ki je bila poslana 2 zahtevi nazaj in poiskati izdelek št. 5, ker je že sistematično pozabil, da je sploh kdaj to stran poslal. Če bi vodil evidenco o vseh zahtevah, potem bi zelo hitro napolnil disk in se ustavil ("obesil"). Teoretično bi se mogoče dalo beležiti zahteve po specifičnih straneh z izdelavo programa za spremljanje, vendar pa se da to doseči na lažji način s piškotki.

Do sedaj sem obravnaval nosilne protokole, po katerih se preko interneta "vozijo" spletni dokumenti oziroma podatki o njih. Nadaljeval bom z opisom jezika HTML, s katerim je na tak ali drugačen način opisan skoraj vsak dokument, ki ga pošiljamo preko WWW. HTML (Hyper Text Markup Language) je označevalni jezik,

kar pomeni, da vsaj v času, ko je nastajal, ni bil namenjen programiranju, temveč označevanju vsebine. Označeno vsebino naj bi programi, ki razumejo oznake, predelali v obliko, kot jo vidimo na zaslonu. Namen jezika je torej določiti obliko vsebini. Ker lahko programe za iskanje in uporabo oznak napišemo v katerem koli programskem jeziku, lahko zato za vsak posamičen operacijski sistem napišemo program, ki bo ustrezno prevajal oznake v standardizirano obliko. Gre torej za platformsko neodvisnost pri standardizaciji oblikovanja. Če poenostavimo: dokument HTML izgleda približno enako na kakršnem koli operacijskem sistemu. V principu poteka označevanje oblike s pomočjo parov označb, ki so kot označbe definirane z znakoma za večje in manjše s tem, da je zaključni oznaki dodana še poševnica. Tako lahko na primer dosežemo odebeljen tekst z označbama `<B>odebeljen tekst</B>` (B kot bold oz. odebeljeno v angleščini). Program za pregledovanje dokumentov HTML in brskanje po internetu izlušči oznake in vse kar je vmes, ustrezno predela in oblikuje. Brskalnik torej v zgornjem primeru na zaslon vpiše: **odebeljen tekst**. Vendar pa to ni vse, sicer bi bil brskalnik samo urejevalnik teksta. Poleg možnosti oblikovanja z oznakami, ki so seveda standardizirane in veljajo za vse enako, lahko uporabimo tudi oznake za tvorjenje hiperbesedila. Hiperbesedilo so tisti deli dokumenta, ki se navezujejo na druge vsebine oz. dokumente. To pomeni, da se na mestu, kjer stoji oznaka za hipervsebino, prikaže poleg teksta tudi slika, film ali pa se predvaja zvok (večpredstavnost). Strežnik dostavi te druge (pravimo jim večpredstavnostne vsebine) preko URL (Uniform Resource Locator), ki je naveden običajno kot atribut oznake v sicer že dostavljenem dokumentu. Pomembna lastnost hiperbesedila je tudi možnost navezovanja - napotitve brskalnika na druge dokumente s spletnega mesta. Uporabniku ni potrebno poznati URL-ja povezanega dokumenta, ampak pač klikne na povezavo in brskalnik zahteva od strežnika nov dokument. Ko ga od njega dobi, odstrani oznake, njim ustrezno oblikuje besedilo, in postavi večpredstavnostne vsebine. Če strnemo do sedaj povedano, je brskalnik program, ki zna oblikovati besedilo (pozna tudi druge elemente kot npr. tabele) v skladu z oznakami HTML, lahko predstavi večpredstavnostne vsebine (multimedijo, slike ipd.), razume URL in zna z njegovo pomočjo uporabiti protokol HTTP za pošiljanje ustreznih zahtev strežniku.

Kljub bogastvu, ki ga lahko prikaže označevalni jezik HTML, pa je zanj značilna določena statičnost, namreč vsebina, ki jo je brskalnik zinterpretiral, je v brskalnikovem oknu nespremenljiva. Tudi neznatne oblikovne spremembe bi zahtevale ponoven prenos celotnega spremenjenega dokumenta, ki bi ga moral vnaprej pripraviti urejevalec spletnih strani. Ob tem pa so uporabniki običajnih programov navajeni na določeno interaktivnost. Programi se odzivajo na dejanja uporabnikov običajno na vsaj dva načina:

1. Oblikovno, z spreminjanjem izgleda aplikacije. (Primer: spreminjanje kazalca miške, podlage, označevalnih elementov ipd.)
2. Z ozirom na podatke, vnesene v program, oz. input/output odzivnost (primer: ob vnosu podatka se v skladu z vnesenim odpre novo okno).

Te vrste interaktivnosti sam označevalni jezik HTML vsaj v začetku ni omogočal. Z razvojem specifikacije jezika HTML so vanj vključili oznake in attribute, ki so omogočali izvrševanje programske kode in s tem tudi določeno interaktivnost.

3.3. Skripti na uporabniški in strežniški strani internetne povezave

Posebej se mi zdi potrebno izpostaviti dejstvo, da razvoj označevalnega jezika HTML ni šel v smeri razvijanja programerskih elementov (ukazov ipd.), temveč je šel v smeri razvijanja objektnega modela dokumenta HTML. Ta objektni model je omogočal vključitev kateregakoli skriptnega ali običajnega jezika v siceršnjo HTML kodo, pod pogojem, da zna jezik ta objektni model uporabiti. Vsak brskalnik, ki je razumel ukaze v kodi vsebovanega jezika, je lahko to kodo izvršil, kar pomeni, da je moral brskalnik vsebovati tudi interpreter v kodo dodanega jezika. V praksi različni proizvajalci vključujejo interpreter le za 2(3) jezike in sicer JavaScript in Jscript kot Microsoftovo implementacijo ECMA standardiziranega JavaScripta. Tema dvema ob bok pa lahko postavimo še VBScript. Refsnes Data (Refsnes Data, 2003: 1) je v zvezi s tem objavila raziskavo, po kateri je julija 2003 uporabljalo IE kot brskalnik 98% internetnih surferjev, IE pa interpretira VBScript. Drugi viri kažejo med 85% in 95% zastopanost IE med brskalniki.

Strežniško/odjemalska arhitektura interneta ima svoj vpliv tudi na izvajanje programov (skriptov), ki jih vsebujejo dokumenti HTML. Ti nimajo narave klasičnih aplikacij, ki so običajno samozadostne celote. Izvajajo se namreč lahko na

strežniški oziroma uporabniški strani internetne povezave. Ker poteka prenos podatkov preko interneta relativno počasi, so običajno spletne aplikacije razdeljene na vsaj dva dela in sicer na strežniško in uporabniško komponento. Tako strežniška komponenta priskrbi podatke, komponenta na brskalnikovi strani pa te podatke obdela, poskrbi za grafični vmesnik in jih po potrebi vrne. Ker pa so lahko baze, iz katerih strežniška komponenta črpa podatke, zelo obsežne, se običajno klientu pošljejo le tisti podatki, ki so nujno potrebni za obdelavo. Če so potrebni dodatni podatki, se vzpostavitev povezave ponovi s popravki poizvedbe oziroma z novo poizvedbo. S tem so podatki v prvotnem obsegu in vsebini izgubljeni. *Poenostavljeno bi lahko rekli, da ima vsaka prenesena spletna stran tudi svoje podatke.* To pomeni, da HTML nima sebi lastnih mehanizmov za hrambo vrednosti spremenljivk, ki bi se npr. hranile kot spremenljivke spletnega mesta. Seveda obstajajo tudi poti, s katerimi se temu izognemo. Več o tem kasneje. Skripti oz. programčki, vključeni na spletno stran, ki jo razumemo kot spletni dokument oz. datoteko, poskrbijo za oblikovno odzivanje spletne aplikacije na dogodke, ki jih povzroči uporabnik, omogočajo osnovne inupt/output tokove ter deloma procesiranje - obdelavo podatkov. To se dogaja lahko tudi na strežniški strani.

3.3.1. Posebnosti skriptov na strežniški strani internetne povezave

Izvajanje skriptov spletnih dokumentov na strežniški strani ima svoje posebnosti. Te so odvisne predvsem od:

1. operacijskega sistema, na katerem teče strežniški program
2. in od vrste strežniškega programa ali spletnega strežnika.

V zadnjem času pa je vpliv teh dejavnikov čedalje manjši, ker skušajo izdelovalci spletnih strežnikov omogočati svojim strankam vse storitve, ki jih omogoča tudi konkurenca. Izvajanje skriptov na strežniški strani je omogočeno na več načinov, od katerih so najvažnejši naslednji:

1. CGI (Common Gateway Interface),
2. SSI (Server Side Includes),
3. ISAPI (Internet Service Application Programming Interface),
4. ASP (Active Server Pages),
5. JSP (Java Server Pages).

CGI je standardiziran programski vmesnik med spletnim strežnikom in zunanjim programom. To pomeni, da ima vsak strežnik ne glede na izdelovalca vmesnik, ki omogoča strežniku zagon programov na zahtevo brskalnika. Po domače bi lahko rekli, da je CGI podprogram spletnega strežnika za zagon drugih programov in vračanje rezultatov teh programov brskalniku. Brskalniki lahko s pomočjo ukaza POST ali GET zahteva od strežnika zagon določenega programa, pri tem pa mu posreduje tudi parametre, ki jih program nato vključi v obdelavo. Rezultat vrne program strežniku, ta pa ga posreduje brskalniku. Če je rezultat zapisan v obliki HTML, potem bo brskalniki prikazal spletno stran. CGI tako omogoča celovitost interaktivnosti spletne aplikacije in kot tak pomeni prelomnico v oblikovanju spletnih strani.

SSI je zbirka funkcij, ki so jih dodali spletnim strežnikom. Za nas sta zanimivi predvsem funkcija `exec cgi` oziroma `exec com`, ki omogoča izvajanje programov CGI in je tako SSI neke vrste podvrsta CGI programiranja³⁴. Zaradi varnosti jih "Free Hosting providerji" neradi dopuščajo. Skripti se namreč izvajajo lokalno, kar je včasih varnostno vprašljivo in tudi korist početja je vprašljiva. Skripti CGI se lahko izvajajo na nek način globalno.

CGI ne omogoča, da se izvede samo del kode na spletni strani, temveč zahtevajo generiranje celotne spletne strani. S SSI tako na primer lahko oblikujemo spletno stran in v mesta s programom strežnik SSI vstavi aktualne podatke. Spletna stran je v tem primeru že oblikovana, koda za oblikovanje se ne generira, strežnik generira le podatke in jih vstavi. Pri skriptih CGI pa je potrebno tudi generiranje kode samega oblikovanja. To je lahko za oblikovalce spletnih strani nerodno, saj otežkoča enotnost izgleda spletne strani. Če na primer spremenimo barvo ozadja, moramo spremeniti tudi kodo CGI, ki bo vsebovala novo ozadje. Obstajajo seveda načini, s katerimi se temu izognemo (CSS³⁵). Pri SSI to generiranje kode HTML ni potrebno. Na kratko, SSI omogočijo ločevanje programerske od oblikovalske logike spletnih strani. Osebno mislim, da je to velika prednost.

ISAPI deluje kot vmesnik strežnikov na nivoju kode. To pomeni, da se program ne interpretira, temveč se v obliki izvršilne datoteke izvaja. Izvaja se strojna

³⁴ `exec com` omogoča izvajanje lupinskih programčkov ali funkcij. `com` in `cgi` sta parametra funkcije `exec`.

³⁵ Cascading style sheets.

koda, ki se kliče lahko s kodo strežnika. To so zelo učinkoviti in hitri programi, vendar pa izdelava le-teh zahteva znanje programiranja v jezikih kot so C, C++, Pascal, Delphi. Poleg tega je običajno za izdelovanje teh programov potrebno več časa. Tudi ISAPI tako kot CGI ne omogoča lahkega ločevanja med programiranjem in oblikovanjem spletne strani.

ASP je uvedel Microsoft na svojem Internet Informacijskem Strežniku, s tem da mu je dodal 7 objektov, ki so na nek način sestavni del strežnika. Zdi se, da združuje najboljše lastnosti svojih predhodnikov. Omogoča izvajanje le dela kode in s tem ohranja oblikovanje (ločitev programske od oblikovalske funkcije), omogoča neposreden dostop do obstoječih objektov na strežniku in s tem hitrost, ki je sicer nekaj manjša kot pri ISAPI pa vendar zadovoljivo hitra, krivulja učenja je zelo strma (ni se potrebno učiti novih sintaktičnih pravil), enostavnost kodiranja pa zelo velika, ker je vključena v siceršnjo sintakso programskega jezika.

JSP Javine strežniške strani so Sunov odgovor na ASP in omogočajo isto funkcionalnost skozi različne vmesnike Java. Za razliko od ASP je programski jezik omejen na Javo in je tako krivulja učenja položna. Zdi se, da so pravzaprav podvrsta ISAPI oz. Java ISAPI, ker delujejo preko strežniških vmesnikov.

3.3.2. Raznolikost jezikov na strežniški strani internetne povezave

Za razliko od skriptov na uporabniški strani so skripti na strežniški strani jezikovno veliko bolj raznoliki. Napišemo jih lahko v katerem koli skriptnem jeziku. Izvajanje strežniških skriptov je odvisno predvsem od tega, ali vsebuje spletni strežnik interpreter skriptnega jezika in ali je interpreter za določen skriptni jezik za določeno okolje sploh napisan oziroma ali ga je sploh mogoče napisati. Od tega odstopa Java kot programski in ne skriptni jezik, ki se lahko izvaja v katerem koli okolju. Zdi se, da vezanost na operacijski sistem in strežniški program programerje omejuje. Vendar temu ni tako. Programerji in njihove stranke imajo namreč veliko možnosti izbire med strežniki različnih podjetij, ki ponujajo storitve spletnega gostovanja. Vsak tako lahko najde sebi ustreznega ponudnika, katerega operacijski sistem in na njem postavljen strežnik bosta podpirala skriptni jezik spletne aplikacije. Standardizacija skriptnega jezika tako ni potrebna. Raznolikost v tem primeru uporabnikom (lastnikom spletnih strani oz.

uporabnikom storitev spletnega gostovanja) koristi, saj povečuje konkurenčnost in zbija cene.

Poleg objektivnih dejavnikov OS in strežniškega programa ter načina komunikacije med spletno stranjo in strežniškim okoljem tako o izbiri programskega skriptnega jezika odločajo predvsem subjektivni dejavniki, kot so subjektivna usmerjenost programerjev k določenemu programu, želje naročnika ali cena gostovanja.

3.3.2. Baze podatkov in SQL

Spletna trgovina ne more delovati brez podatkov o izdelkih, cenah in podobno. Te podatke je načelno mogoče vnesti v samo spletno stran, v praksi pa je to za trgovino, ki ima npr. 1000 izdelkov, zelo težko, nepraktično in tudi nevarno. Podatke je najučinkoviteje pridobiti iz baze podatkov, ki se bodisi ažurira ročno, bodisi preko XML povezanih dokumentov in spletnih uslug.

Poizvedbe in spremembe v baze podatkov opravljamo večinoma z jezikom SQL. SQL (standardni poizvedovalni jezik) je standarden jezik za dostop in upravljanje z bazami podatkov. Poizvedbe, narejene v tem jeziku, naj bi razumela praktično vsaka baza podatkov, zato lahko enako poizvedbo uporabimo na bazah različnih proizvajalcev, seveda pa morajo biti pri tem podatki enako formatirani. Če se formati podatkov razlikujejo, se strukturi podatkov prilagodi le ukaz SQL, ne pa celoten program. SQL je na tak način pravzaprav standard za podatkovne baze in kot tak v veliki meri olajšuje prilagajanje aplikacij različnim podatkovnim bazam.

Do podatkov lahko dostopa in jih ureja poseben temu namenjen strežnik (v smislu računalnika ali domene) ali strežniški program na strežniku, ki je hkrati tudi spletni strežnik, lahko pa poizvedbe opravi (za manjše baze) preko objektov komponent COM³⁶ sam web strežnik po nalogu skripta na spletnem mestu. Do podatkov se lahko dostopa neposredno preko strežniškega programa ali pa preko gonilnikov ODBC³⁷, bodisi preko OLE DB³⁸, ki je na nek način ovoj za objektno doseganje programskih vmesnikov ODBC podatkovnih strežnikov. Tehnologija ADO (Microsoft) je neka vrsta integrirane in standardizirane, od proizvajalca neodvisne povezave z nekaterimi bazami podatkov.

³⁶ COM common object model je microsoftov standard za povezljivost programskih objektov.

³⁷ ODBC - open database connectivity je standard za povezavo z bazami podatkov.

³⁸ OLE DB - data base object linking and embedding običajno programska komponenta za povezavo z bazami podatkov.

Mala podjetja, za katera gre v tej nalogi, si običajno svojih strežnikov ne morejo privoščiti. Investicija bi bila previsoka in tudi neučinkovita, saj lahko zaželen učinek dosežemo s cenejšimi sredstvi. To bi bilo podobno, kot če bi muhe uničevali z bombami. Zanje so torej dovolj učinkovite rešitve, ki jih omogoča npr. ADO.

3.3.3. Izbira skriptnega jezika na strežniški strani internetne povezave

Programer skriptov na strežniški strani lahko izbira med številnimi skriptnimi jeziki, kot so: Perl, PHP, Jscript (JavaScript), VBScript, Java, ipd. Izbira temelji na podlagi odločitve o načinu poganjanja strežniških skriptov (Cgi, Asp ipd.) ter glede na opremljenost jezika z moduli za komunikacijo z bazo podatkov, v kateri so ti zapisani.

Zdi se, da je Perl najboljša izbira. Omogoča namreč povezavo z bazami podatkov, je izredno učinkovit, še posebej pri obdelavi besedil, kar pride prav pri morebitnem CGI oblikovanju HTML strani, združljiv je z ASP in še bi lahko naštevali. Poudariti velja njegovo splošno naravo, ki je njegova prednost in hkrati tudi slabost. Ta splošnost omogoča programerju, da ga uporabi za številne različne naloge, vendar pa mora za vsako nalogo pripraviti vsaj neko rutino, če ne že celoten sklop rutin oz. modul. V zadnjem času ga nekako dohiteva PHP, ki je bil ustvarjen prav za skriptanje spletnih strani. V sam jezik so zato vgrajeni ukazi, ki bi v Perlu zahtevali celotne module. Ker gre pri spletni trgovini za spletne strani, se zdi, da bi bil PHP Perlu vsaj enakovreden. Še posebej to drži, če imata oba enako strmo krivuljo učenja. PERL omogoča uvoz modulov za pridobivanje podatkov iz podatkovnih baz številnih proizvajalcev, pri PHP pa je dostop do baz celo vgrajen v sam jezik.

Java kot izbira zahteva ISAPI, možen bi bil verjetno tudi CGI. Krivulja učenja je zelo položna, kar pomeni, da je za učenje ta jezik težak. Za poznavalce jezika pa zaradi platformske neodvisnosti Jave izbire ni.

Na Microsoftovih strežniških sistemih pa je uporaba ASP brez dvoma najučinkovitejši način za izvajanje strežniških skriptov. Uporablja tehnologijo COM³⁹, ki je enostavna za uporabo (ASP vsebuje le 7 objektov) in tudi zelo hitra, saj se izvaja (objekti, ne pa sami skripti) preko strojne kode. Pri tem pa preko

³⁹ Component object model.

ADO (access data objects) omogoča dostop do vseh mogočih podatkovnih baz vključno z uporabo poizvedb SQL. Vse to je doseženo preko celovitega objektnega sistema, ki omogoča enotne specifikacije za kodiranje. Tudi učenje osnov ni zahtevno in samo kodiranje poteka hitro, saj je razhroščevanje lahko. ASP lahko uporabljamo s Perlom, Jscriptom in VBScriptom. Jscript (v bistvu Microsoftov JavaScript) in VBScript sta v rahli prednosti. Uporabljata se namreč lahko tudi za skripte, ki se bodo izvajali v uporabnikovem brskalniku. To pomeni, da so lahko vsi skripti napisani le v enem skriptnem jeziku. To je za programerja prednost, ker se mora za izdelavo skriptov in njihovo vzdrževanje naučiti le enega skriptnega jezika. Java Script se lahko uporablja samo za pisanje programov, ki se bodo izvajali v internetnem okolju bodisi na strežniški strani bodisi na strani klienta. VBScript pa je različica Visual Basic, ki se uporablja za izdelovanje Windows aplikacij. Ob zgoraj navedenem dejstvu, da od 85% do 97% uporabnikov interneta le-tega ogleduje z IE, je odločanje med Jscriptom in VBScriptom za poznavalce VB bolj akademska. VB se lahko uporablja takorekoč za vse.

3.3.4. Hramba vrednosti spremenljivk med spletnimi dokumenti

Prej smo ugotovili, da HTML označevalni jezik oziroma skripti na spletni strani ne morejo shranjevati oziroma izmenjevati vrednosti spremenljivk med različnimi spletnimi stranmi. Načelno so vrednosti spremenljivk izgubljene takoj, ko v brskalnikovem oknu naložimo novo spletno stran. Ne obstajajo torej možnosti za uporabo spremenljivk z dosegom celotnega spletnega mesta. Na novo naložena stran tako ne more uporabljati vrednosti spremenljivk, ki so jih te dobile, kot posledica uporabnikovih dejanj na prejšnji strani. Primer: omogočimo na primer uporabniku, da na 1. strani izbere barvo ozadja. Njegov izbor naložimo v spremenljivko strBarva, ji torej priredimo vrednost uporabnikove izbire, ki je recimo "bela". Ko naloži drugo stran našega spletnega mesta, s skripti ne moremo več ugotoviti, kakšno vrednost za ozadje je izbral na 1. strani, strBarva in njena vrednost sta z naložitvijo nove strani izgubljeni.

Seveda se je možno tej težavi izogniti na več načinov.

1. Lahko oblikujemo spletno stran z okviri in v glavnem okviru ustvarimo skrito vnosno polje, v katerega vnašamo spremenljivke in njihove vrednosti. Pogoji je razdeljenost spletne strani na okvirje, ki omogoča prebiranje

spremenljivk v istem procesu oz. niti. Ta rešitev nas obvezuje na določeno oblikovno rešitev.

2. Težavo lahko rešimo tudi tako, da preko URL-ja poženemo ASP, CGI podprogram, ki shrani podatke v bazo podatkov na strežniku. Ob naložitvi nove strani preberemo ustrezne podatke iz baze. Rešitev je v Perlu relativno preprosta, vendar nasprotuje osnovnemu principu strežnik/odjemalec arhitekture, saj opravlja strežniški proces naloge, ki bi jih lahko izvajal uporabnikov brskalnik. S tem zmanjšujemo prosta sredstva strežnika.
3. Objektni model HTML so za rešitev tega problema nadgradili z objektom Cookie (Piškotek). Piškotek je majhna tekstovna datoteka, ki jo lahko ustvarimo in shranimo na uporabnikovem računalniku, ter seveda vanjo vpisujemo vrednosti. Ker se prenaša v glavi HTML dokumenta, jo je težje ponarediti, ob tem, da lahko načelno vanjo vpisuje podatke le domena, ki ga je ustvarila. Temu se je sicer mogoče izogniti z dovoljenji, ki pa jih spet v parametrih piškotka dodeljuje domena, katere del je spletni dokument. Zloraba piškotkov je mogoča torej le na dveh ravneh in sicer preko zapisovalca piškotkov in preko samega uporabnika, ki bi lahko teoretično piškotek odprl in vanj vpisal neresnične podatke. Zapisovalec piškotkov pa bi lahko podatke, pridobljene od uporabnika posredoval tretji strani. Piškotki ne morejo biti izvršilna datoteka. Težava nastopi le takrat, kadar je uporabnikov brskalnik nastavljen tako, da ne dovoljuje zapisovanje piškotkov. V takih primerih pač spletna trgovina ne deluje, morebitnega kupca pa se z opozorilom zaprosi, naj za ogled spletne strani začasno spremeni nastavitve brskalnika.

3.3.5. Zaključek

Glede na dejstvo, da mi je za postavitve spletne strani in razhroščevanje kode na razpolago strežnik Windows 2000 z nameščenim IIS (Internet Information Server), je najučinkoviteje in tudi najlažje kot skriptni jezik za strežniške skripte uporabiti VBScript, uporabiti objekte ASP za dostop do strežnika in ADO za dostop do baze podatkov (Access). Uporabljam torej samo en jezik, znanje le-tega pa lahko uporabim tudi v običajnih aplikacijah Windows.

Trgovino bo lahko uporabljalo od 85% do 97% uporabnikov interneta, ostale bi lahko pridobil, če bi se odločil za Jscript oziroma JavaScript. Ker je naloga učni pripomoček in ne komercialni izdelek, se mi zdi, da je koristneje pridobiti znanje VBScripta, ki je mogoče uporabiti tudi v aplikacijah Windows, kot pa uporabiti Jscript, ki je omejen na internetno okolje.

Upoštevati je potrebno še neko dejstvo. Z uvedbo .NET izvajalnega sistema je mogoče deloma popravljeno kodo VBScript uporabiti v tem okolju in ob tem integrirati program z vsemi novimi tehnologijami kot so XML, SOAP, EDI in podobno. Jscript zaenkrat tega ne omogoča na tako enostaven način. V tem smislu stavim na prihodnost .net arhitekture, ki omogoča po mojem mnenju do sedaj najboljšo in najenostavnejšo integracijo različnih informacijskih sistemov.

Odločil sem se za Accessovo bazo podatkov, ki lahko deluje preko objektov ADO v okviru procesa IIS. Zato ne potrebujem posebnega strežnika in tudi ne posebne povezave zanj. Za mala podjetja pa je Accessova baza podatkov dovolj uporabna. Ker bomo uporabili ADO in poizvedbe SQL, lahko bazo z minimalnimi spremembami zamenjamo.

4. Analiza programa

V zadnjem delu naloge bom obdelal program, ki sem ga izdelal na podlagi zgornjih ugotovitev. Eno od glavnih vodil pri oblikovanju je bila ločitev programske in oblikovalske logike spletnih strani, na katerih deluje program. Od elementov HTML sem v sam program vključil pravzaprav le tabelo in obrazec. To pomeni, da se lahko v oblikovalnem orodju HTML oblikuje izgled in se v zvezi z njim postavijo ustrezni oblikovalni skripti, nato pa se v stran samo prekopira skript z tabelo. Sam program je sestavljen iz dveh komponent in sicer iz kataloga in košarice.

Košarica je zasnovana kot dialoška spletna stran z določenimi merami in tudi oblikovanjem. Vendar pa oblikovanja ni dosti in se lahko zelo enostavno prilagodi siceršnjemu izgledu bodoče spletne strani. Ker je košarica zasnovana na principu dialoga, v času ko je odprta kataloške spletne strani, ki je sprožila odpiranje košarice, ni mogoče uporabiti. To onemogoča odpiranje več kot ene košarice naenkrat in olajšuje vodenje evidenc o naročilu. Številne spletne trgovine tega ne omogočajo. Tako lahko pride do situacije, ko se neizkušen spletni uporabnik po nesreči dotakne ozadja (kataloške spletne strani), košarica se zato minimizira in izgine iz ekrana. Uporabnik jo še enkrat pokliče, ker misli, da se je zaprla sama od sebe in naenkrat imamo odprtih vendar minimiziranih več košaric včasih celo z enim in istim izdelkom. Rešitev je preštevanje odprtih brskalnikovih instanc v trenutni niti, kar je s skriptnim jezikom praktično nemogoče. Lahko bi se košarico zastavilo kot ActiveX komponento, s tem pa ustvarimo nove, še večje probleme. Večina programerjev se zanaša na srečo in upa, da do teh nerodnosti ne bo prišlo. Na sami košarici je tudi obrazec, ki postane viden šele ob pritisku na gumb pošlji. Ker menim, da je v tej fazi razvoja spletnega trgovanja zaupanje v prodajalca odločilno pri odločanju o nakupu preko spletne strani, ne bom omogočal plačevanja vnaprej preko kartice, temveč po prejetju naročila, ki ga trgovec telefonsko preveri. Potencialni kupec dobi tako tisto za nakup odločujočo stopnjo varnosti, saj ve, da mu ne bo nihče pobral denarja z računa in potem "pozabil" dostaviti naročeno blago. Gre za utrjevanje zaupanja v sicer neznanega prodajalca. Proces nakupovanja pa je bližji vsakodnevni nakupovalni izkušnji

bodočega kupca, saj vemo, da se običajno ob plačilu naročenega blago tudi takoj prevzame. Uporabnik tako ob naročilu vnese le svoje ime in priimek ter telefonsko številko za potrditev naročila. Podatki o kupcu in naročenem blagu v košarici se nato preko mailto protokola dostavijo na e-mail trgovca oz. lastnika spletne strani. Podatki so oblikovani in se lahko spremenijo po potrebi v oznake XML. Ker bo to prosta koda, na razpolago vsem, ki bodo kodo hoteli uporabiti, je vanjo vgrajena omejitev, ki izkorišča hrošča v brskalniku IE. Naročilo vsebuje lahko zato le približno 400 znakov. Rešitev je preprosta. V ACTION Parameter obrazca je potrebno navesti skript (datoteko s skriptom), ki bo prevzel podatke in spremeniti parameter METHOD v "POST". Metodo POST je potrebno uporabiti, ker je varnejša, saj pošilja podatke v telesu sporočila, ne pa kot del URL-ja. Iz izkušenj vem, da bo od šestdesetih "hackerjev" le eden poskušal zlorabiti post metodo. Okno košarice ob vzpostavitvi iz piškotka prebere naročilo. Piškotek je nosilec podatkov, ki omogoča izmenjavo dodanih naročenih izdelkov med katalogom in košarico, hkrati pa hrani vrednosti naročila.

Kataloška spletna stran je veliko bolj enostavna kot košarica. V principu deluje tako, da skripti ASP iz baze podatkov preberejo izbrane izdelke in jih zapišejo (dobesedno) v spremenljivke. Te potem obdelajo skripti na uporabniški strani in oblikujejo tabelo s tremi vrsticami in tremi stolpci. V vsako polje tabele se vstavi pomanjšana slika izdelka, njegova cena, ime in gumb "V košarico". Ob kliku na gumb se modalno zažene (kot dialog) brskalniskovo okno z dodanim izdelkom (košarica).

Ime izdelka bi bilo lahko oblikovano kot "link" (z oznakami SPAN) na dinamično izdelano stran, ki bi opisovala izdelek. Iz praktičnih razlogov tega nisem izpeljal. Problem je namreč v slikah in opisih. Ti bi morali biti shranjeni v posebni tabeli. Za npr. 200 izdelkov bi moral za demonstracijo obdelati (skeniranje ipd.) 200 slik. Konceptualno pa bi to izgledalo tako, da bi se spremenljivka z imenom izdelka uporabila kot osnova za poizvedbo SQL, ki bi v bazi poiskala URL slike izdelka in njegov opis. To dvoje bi se dinamično vstavilo, v sicer že oblikovano spletno stran ASP. Na spletno stran kataloga bi bilo potrebno dodati enostaven obrazec z pomanjšanim oz. skritim tekstovnim poljem in v parameter Onklik "linka" (oznake SPAN) določiti proceduro, ki bi:

1. določila vrednost (ime izdelka) lastnosti `text` tekstovnega polja,

2. in to nato preko metode POST poslala strani ASP (parameter ACTION obrazca).
3. Zgornja procedura bi sprožila POST ACTION s klicem metode SUBMIT obrazca (`imeObrazca.submit`). Tako bi poslali vrednosti obrazca (`ime izdelka` v skritem tekstovnem polju) kot parameter h klicu ASP strani.

Ta stran ASP bi dejansko opravila zgornjo poizvedbo SQL in se oblikovala v skladu z rezultatom. Stran ASP z opisom izdelka bi preko objekta Request in njegove kolekcije `forms` (`imeObrazca.form("imeTextPolja")`) prebrala vrednost za ime izdelka in jo uporabila v SQL poizvedbi. Iz vrnjenega "Recordseta" bi v stran ASP z opisom izdelka v nadaljevanju dinamično zapisali URL slike in tudi opis. Kljub vsemu temu pa bi še vedno morali pripraviti vse slike.

Kataloška stran je zasnovana za prikazovanje delov baze izdelkov. Deli, ki jih bo prikazala konkretna stran, so določeni s poizvedbo SQL. Če hočemo prikazati druge izdelke, dupliciramo stran in spremenimo parametre SQL, vse ostalo ostane isto z oblikovanjem vred. Kompletan katalog je tako sestavljen iz večjega števila popolnoma enakih spletnih strani, ki se razlikujejo samo po poizvedbi SQL in naslovu. Vse strani, ki smo jih določili za prikazovanje celotne baze izdelkov, se vključijo kot sestavni del spletnega mesta. Na spletni strani kataloga sta še dva navigacijska gumba za premikanje med izdelki, kadar šteje blagovna skupina več kot 9 izdelkov, ki gredo na eno stran. Gumba sta povezana z ustreznima rutinama. Zadnji, tretji gumb omogoča prikaz košarice.

Košarica je v bistvu tabela, ki se dinamično gradi v skladu s podatki, ki jih pridobi iz piškotka. Obsežnost kode večajo predvsem preračunavanja stolpcev, preverjanja podatkov, oblikovanja in odzivnost na akcije kupca. Ta se gradi na dva načina: klasično in z uporabo funkcije `getRef`. Ob spreminjanju košarice - odstranjevanju izdelka iz nje poskrbi košarica za zapis popravljene vsebine v piškotek.

4.1. Katalog, zgradba in delovanje procedur

Katalog je sestavljen iz strežniških in uporabniških procedur. Strežniške procedure izvaja strežnik in zapiše rezultate neposredno v spletno stran. To pomeni, da se vrednosti spremenljivk ne prenašajo kot take, temveč se na brskalnikovo stran internetne povezave prenese samo rezultat v obliki zapisa. Uporabniške skripte pa izvršuje uporabnikov brskalnik. Vkolikor mora dobiti

določene vrednosti od strežnika, prebere zapis, ki ga je ustvaril strežnik. Tako na primer iz baze potegnemo ime določenega izdelka, recimo, da je ime izdelka majica. Če uporabniški skript potrebuje to vrednost za spremenljivko, dobesečno prebere besedo majica, ki jo je zapisal strežniški skript. V kodi ASP bi to izgledalo takole:

```
lmlzd = <%procedura%>          ---> procedura vrne besedo majica
```

V brskalnikovi kodi bi ta koda izgledala takole:

```
lmlzd = "majica"
```

Strežniški skripti, ki jih obdajajo oznaki `<%` in `%>`, torej v brskalniku niso vidni, viden pa je rezultat, ki ga vrne procesirana procedura ASP. Katalog z bazo podatkov zato ne komunicira neposredno in neprestano, temveč opravi poizvedbo in vrnjen "recordset objekt" spremeni v besedo (string). Podatke loči z nekim poljubnim ločilom, ter ta z ločili razdeljen "string" zapiše kot vrednost za določeno proceduro v skriptu na uporabniški strani. Vse procedure na strani kataloga so zapisane v enem skript bloku. To nam omogoča najavo nekaterih spremenljivk, katerih vrednosti lahko spremeni katerakoli procedura, vključena v ta blok. Lahko bi rekli, da imajo vrednosti teh v začetku script bloka najavljenih spremenljivk nekako globalen doseg strani.

Tok programa teče takole:

1. Procedura ASP (1) na strežniški strani odpre bazo podatkov, s stavkom SQL iz baze pobere izdelke, ki ustrezajo blagovni skupini, ki jo predstavlja in spravi vse skupaj v `recordset` objekt.
2. procedura ASP (2) še vedno na strežniški strani iz objekta `recordset` skozi zanko, s pomočjo katere pregleda vse zapise objekta `recordset`, oblikuje string (besedo) za tri spremenljivke in sicer `strIzdIme`, `strIzdCena` in `strSlika`. Podatki za posamičen izdelek so v besedi ločeni z znakom `$`. Zatem se objekt `recordset` zapre, ker nočemo imeti baze v neskončnost odprte. Te besede (stringi) se nato dobesečno zapišejo kot vrednosti spremenljivk `pljSlika`, `pljIzdIme`, `pljIzdCena` v proceduri `window_onLoad`.
3. Na tej točki izvajanja programa se preselimo na uporabnikovo stran internetne povezave. Brskalnik je prejel tok HTTP s kodo HTML in jo procesira skupaj z vpisanimi vrednostmi za podatke iz baze podatkov.

Ob nalaganju okna spletne strani se vedno sproži dogodek `onLoad()`, ki ga prestrežemo z dogodkovno proceduro `window_onLoad()`. Z njo pravzaprav nadaljujemo izvajanje programa na uporabnikovi strani internetne povezave. Skozi proceduro `window_onLoad()` nato postavimo tabelo in v vsako polje tabele sliko izdelka, ime izdelka in ceno ter gumb "V košarico". To storimo dinamično skozi gnezdeno zanko `for`, ker ne vemo vnaprej katere blagovne skupine bomo naložili. Koda je neodvisna od rezultatov poizvedbe SQL. Zunanja zanka `for` je namenjena izdelavi vrstic tabele, notranja zanka `for` pa izdelavi celice v "trenutni" vrstici. Gumbovemu dogodku `onClick` priredim `Klik()` nadzornika dogodka (event handler). To storim z zgoraj omenjeno funkcijo `GetRef()`. Na tej točki se avtomatizem aplikacije ustavi. Izvajanje se nadaljuje v odvisnosti od uporabnikove akcije. Od vseh mogočih uporabnikovih akcij nadaljujemo izvajanje v štirih primerih:

- a. Uporabnik klikne gumb "Naprej!"
- b. Uporabnik klikne gumb "Nazaj!"
- c. Uporabnik klikne gumb "Pokaži košarico!"
- d. Uporabnik klikne gumb "V košarico!"

Prvi trije gumbi so vnaprej vneseni na spletno stran in jim priredimo dogodkovne nadzornike `Naprej()`, `Nazaj()` in `pkzKosarco()`. Prvi dve proceduri sta namenjeni brskanju po blagovni skupini, če obsega več kot 9 izdelkov. Pri tem za "nove" izdelke ne naredimo nove poizvedbe in ne odpiramo nove povezave, temveč enostavno zbrisem tabelo in kličem deklico za vse z imenom `window_onLoad()`, ki izriše tabelo v skladu z globalnima spremenljivkama `dodani` in `StvPlj`. Obe spremenljivki sta števca, ki beležita mesto zadnjega prikazanega izdelka iz blagovne skupine. `Dodani` beleži koliko izdelkov je bilo dodanih (kje smo v trenutni tabeli) v tabelo pri njenem zadnjem izrisu polja (celice), `StvPlj` pa, koliko jih je bilo vsega skupaj dodanih (mesto, kje smo med vsemi izdelki v blagovni skupini). `PkzKosarco` je v bistvu ukaz za modalno odpiranje košarice z naročenimi izdelki in je identična proceduri `odpri()`, kot taka pa je koristna pri razhroščevanju, ko je potrebno, da se prvi klic izvede pravilno, pri drugem klicu iste funkcije pa sem malo eksperimentiral s parametri. Ostane le še dogodkovni nadzornik za gumb "V košarico!" z imenom `Klik()`. S to

proceduro pravzaprav zapišemo piškotek in potem odpremo košarico. Košarica bo najprej prebrala piškotek in potem izrisala tabelo, ki bo prikazovala izbrane izdelke. Če piškotka ni, se pravi da je uporabnik prvič izbral enega od ponujenih izdelkov, potem piškotek ustvarimo, če pa je prej že izbral kak izdelek, potem izdelek dodamo v piškotek. Piškotku nisem priredil atributa `expires` zato, da ga ni potrebno brisati, saj obstaja kot piškotek samo v delovnem spominu brskalnika. Na ta način zaobidem varnostne nastavitve in omogočam zapisovanje, četudi bi imel uporabnik zelo stroge varnostne omejitve. Še boljše bi bilo uporabiti objekt `session`, ki prav tako zapiše piškotek. Piškotek sestavlja polje treh lastnosti in sicer `ImeIzd`, `Cena`, `Količina` (ta je privzeto 1 in se lahko prilagodi v košarici). Podatke o izbranem izdelku dobi program iz imena gumba, ki ga sestavlja ravno ime izdelka in njegova cena, ločena s tilde (~). To pomeni, da bo ime gumba (lastnost `name`) pod izdelkom, ki prikazuje majico s ceno 20,00 sit, majica~20. Če bi hotel dodati novo lastnost izdelka, bi pač za novo tilde dodal to lastnost tako, da bi bila lastnost (property) `name` gumba npr. majica~20~rdeča. Zapisi v piškotku pa niso urejeni po izdelkih, temveč po lastnostih. Piškotek tako ni sestavljen v smislu

```
Izdelek1=lastnost1~lastnost2~lastnost3; _
Izdelek2=lastnost1~lastnost2~lastnost3
```

temveč:

```
Lastnost1=Izdelek1~Izdelek2; Lastnost2= _
Izdelek1~Izdelek2; Lastnost3= Izdelek1~Izdelek2
```

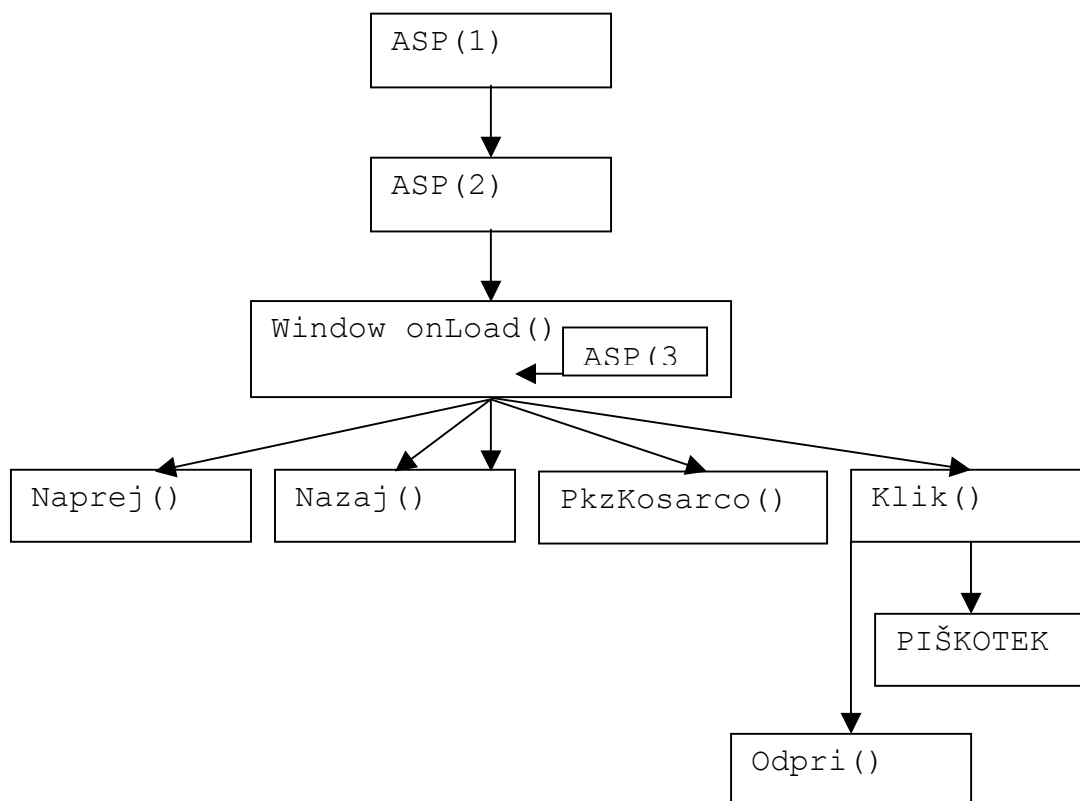
To se zdi na prvi pogled nerodno, vendar je pri prebiranju podatkov iz piškotka v košarico to lažje, saj se tabela piše po vrsticah in se v vrstico dodajajo vrednosti za tekoči stolpec. Je pa res, da bi šlo tudi obratno. Gre za popolnoma subjektivno odločitev. Piškotek za dva izdelka iz baze tako izgleda takole:

```
Bolajcek;ImeIzd=vžigalnik ploščat mat~vžigalnik ploščat
mlečni;Cena=90~80;Kolicina=1~1;ASPSESSIONIDFFFETDCB=CLGPFHIA _
CDGEJOFFDOFCHIKF
```

Spremenljivka za količino je `ASP session id`, objekta `session`, ki ga IE avtomatsko tvori za našim hrbtom. Vidimo torej, da je v tem piškotku zabeležen nakup dveh izdelkov:

1. vžigalnik ploščat mat s ceno 90,00 sit in količino 1 kom,
2. vžigalnik ploščat mlečni s ceno 80,00 sit in količino 1 kom.

Na spletni strani je zaradi demonstracijskih razlogov dodan gumb "Pokaži kuki!", ki kliče proceduro `pkzKuki()`, ki pokaže piškotek. Če zaključimo, katalog torej ne prenaša Košarici vrednosti spremenljivk (npr.: količina, cena ipd), temveč jo zapiše v piškotek. Tega pa prebere Košarica in pridobi potrebne podatke za izris tabele z izdelki. Shema procedur izgleda nekako takole:



Shema 4.1: Shema procedur v katalogu

4.2. Košarica: zgradba in medsebojno delovanje procedur

Spletna stran košarica vsebuje precej več skriptne kode, kot stran katalog. Zaradi lažjega pregleda nad dogajanjem, ki se nadzoruje s kodo, je najbolje, da se glede na vlogo, ki jo opravljajo med delovanjem skriptov, procedure razdelijo na tri skupine.

1. Procedure, ki imajo več ali manj oblikovno vlogo in opravljajo naloge spreminjanja spletne strani, ki je posledica dejanj uporabnika. S temi procedurami torej spreminjam izgled košarice in s spremenjenim izgledom opozarjam na možnost uporabnikove akcije (spreminjanje barve ozadja in napisov ter oblike miškinih kazalcev).
2. Procedure, ki opravljajo razna tabelarna preračunavanja. Ker je osnova košarice tabela z naročenimi izdelki, opravljajo procedure te skupine preračunavanja zmnožkov po vrsticah med količino in ceno, da dobimo

vsoto za plačilo po vrstici in seštevke vrstičnih vsot kot skupno vsoto za plačilo vseh izdelkov skupaj. Ker lahko uporabnik spreminja količine in odstranjuje izdelke iz košarice, je potrebno izvesti hkratna preračunavanja skupnih vrstičnih in stolpčnih vsot in to prilagoditi spremembam. Gre torej za neke vrste neposredno odzivno proceduro na izbiro uporabnika. V resnici so to funkcije, ki opravljajo le naloge množenja in seštevanja, ob tem da so tempirane na nek dogodek, več ali manj zaključen vnos uporabnika. Problem so le poimenovanja celic. Ta imenovanja sledijo tabelnemu objektnemu modelu in izgledajo zapleteno.

3. Tretjo skupino procedur pa sestavljajo procedure, ki tvorijo osnovne elemente programa in pravzaprav postavijo objektni model spletne strani ali pa opravljajo dela v zvezi s tem modelom. To so pravzaprav nosilne procedure košarice in njim bom posvetil največ pozornosti in jih tudi malce podrobneje obdelal.

Procedure iz prvih dveh skupin bom več ali manj naštel in pojasnil njihovo nalogo, ter opozoril na mesto, kjer se kličejo, to pa je tudi vse.

Oglejmo si kako izgleda spletna stran košarica:

Zap.št.	Ime izdelka	Cena	Količina	Za plačilo	Pošlji
1	Vžigalnik nav. mlečni	20,00 SIT	1	20,00 SIT	odstrani
2	Vžigalnik elektronski mlečni	50,00 SIT	1	50,00 SIT	odstrani
3	Vžigalnik ploščat mlečni	80,00 SIT	1	80,00 SIT	odstrani
ZA PLAČILO:				150,00 SIT	V redu

Košarica je izdelek podjetja ANKLEM; tel.: 041/507-485

slika 4.1: Košarica

Kot vidimo, je sestavljena iz tabele, katere število vrstic je odvisno od števila naročenih izdelkov. Podatki o izdelkih se pred renderiranjem strani naložijo iz piškotka.

Procedure iz prve skupine vplivajo na dogajanje v stolpcu količina in na celice z napisom "odstrani". Na slednje vplivajo procedure:

```
objTabela_onMouseout(), objTabela_onMouseover(),
objTabela_onClick(), mover(), mout(), mclick() in klik().
```

S pritiskom na celico "odstrani" se odstrani trenutna vrstica. Za to poskrbi `objTabela_onClick()`, ki nato še sproži tabelarna preračunavanja, ki popravijo vsoto v celici, ki kaže skupno vsoto za plačilo. Procedura `objTabela_onMouseover()` poskrbi za to, da se v primeru, ko se miškin kazalec znajde nad celico z napisom "odstrani" spremeni oblika miškega kazalca in spremeni barva ozadja in napisa v celici. S tem opozorim uporabnika na možnost dejanja oziroma na možnost, da s pritiskom na levo miškino tipko odstrani izdelek iz košarice. Procedura `objTabela_onMouseout()` vrne barvo in ozadje črk ter obliko miškega kazalca, če uporabnik premakne miškin kazalec iz območja celice. Proceduri `mover()` in `mout()` veljata za celice z napisom "Pošlji" in "V redu" in opravljata popolnoma iste naloge, vendar pa se kličeta s funkcijo `GetRef()`, ker je to bolj praktično za ti dve celici. Procedura `mclick()` zapre košarico in je nastavljena tako, da se sproži, če uporabnik pritisne na celico "V redu" ali pa na gumb Pošlji (GLEJ na sliko 4.2). V to skupino procedur spada še procedura `klik()`, ki se sproži, če uporabnik klikne na vnosno polje za vnos količine in označi celotno vsebino polja, zato ni potrebno brisanje stare vsebine polja. Ostane še `brisemo()`, ki zbršiše kompletno vsebino kateregakoli vnosnega polja.

V drugo skupino funkcij spadajo procedure `izracunaj()`, `skupaj()` in `entr()`. Funkcija `skupaj()` vrne vsoto celic v stolpcu za plačilo in znesek vpiše v zadnjo celico stolpca za plačilo v zadnji vrstici. Pri tem to še oblikuje v finančnem formatu z dvema decimalnima mestoma in s SIT simbolom na koncu. Funkcija `izracunaj()` pa vrača zmnožek cene in količine po vrstici in prilagaja ustrezne vrednosti v stolpcu "Za plačilo". Funkcijo `izracunaj` kličejo s funkcijo `GetRef()` naslednji dogodki, ki jih lahko sproži uporabnik:

- deaktiviranje kateregakoli vnosnega polja za vnos količine v stolpcu "količina",
- pritisk na tipko <ENTER> - pod pogojem, da je katero koli od vnosnih polij za količino v žarišču (Procedura `entr()`),
- ali pritisk na celico "odstrani".

Procedura `entr()` opravlja podobne naloge kot funkcija `izracunaj`, izvrši pa se le, če je pritisnjena tipka <ENTER>.

Procedure zgornjih dveh skupin so tipične odzivne procedure na dejanja uporabnika in se naslanjajo na dogodke, ki jih registrira brskalnikovo okolje. Zanje je značilno, da bi lahko delovale v katerem koli skriptu s tabelo, na kateri koli spletni strani. Ne glede na dejstvo, da procedure zgornjih skupin formalno niso odzivne procedure, pa jih kot take vpelje uporaba funkcije `GetRef()`, ki omogoča, da se namesto sistemsko definirane funkcije izvrši poljubna funkcija. Če funkcije `GetRef()` ne bi uporabil, bi moral za vsak dogodek napisati ustrezno odzivno proceduro, ki pa bi imela popolnoma isto vsebino kot odzivne procedure podobnih dogodkov drugih objektov spletne strani.

Programski del spletne strani košarica opredeljujejo pravzaprav procedure tretje skupine. Te so:

1. `Window_onLoad()`,
2. `zapisiTabelo()`,
3. `Izdelki()`,
4. `PosNar()`,
5. `Form1_onsubmit()`,
6. `Window_onBeforeUnload()`,
7. `zapisiKuki()`.

Procedure `Window_onLoad()`, `Form1_onsubmit()` in

`Window_onBeforeUnload()` so klasične odzivne oz. dogodkovne procedure, ki

jih sproža dogodkovni mehanizem brskalnikov v skladu s specifikacijo W3C objektnega modela HTML. V program so vključene zato, ker nam omogočajo reagiranje (vstavljanje ustrezne kode) na za naše namene ključnih točkah in trenutkih programskega toka. To je še posebej značilno za proceduri

`Window_onLoad()` in `Window_onBeforeUnload()`, ki sta pravzaprav v program vključeni samo zato, da lahko v *točno določenem trenutku* kličeta proceduri, ki opravljata pravo delo. `Window_onLoad()` tako sproži izvajanje procedure `zapisiTabelo()`, `Window_onBeforeUnload()` pa

`zapisiKuki()`. Z njima tempiramo izvajanje določenih opravil na prave trenutke. Kako je torej zasnovana spletna stran "Košarica"? Sama po sebi vsebuje štiri objekte, ki sem jih ustvaril s kodo HTML. To so BODY (telo spletne strani) TABLE (tabela), TBODY (telo tabele) in FORM (obrazec). Ustrezno sem jih imenoval zaradi naslavljanja s skriptom. Vse ostale objekte ustvari skriptni blok. Ob

"renderiranju" spletne strani brskalnik ob dogodku `Onload` objekta `window` sproži izvajanje procedure `Window_onLoad()`, ki sproži izvajanje procedure `zapisiTabelo()`. Potrebno je povedati, da bi se lahko temu izognil in zapisal izvajanje kode procedure `zapisiTabelo()` v samo kodo HTML, pa tega zaradi lažjega razhroščevanja nisem storil. `zapisiTabelo()` prav v trenutku, preden uporabnik zagleda spletno stran, zgradi objektni model košarice v skladu s podatki, ki jih s pomočjo funkcije `Izdelki()` dobi iz piškotka. Opozarjam, da v trenutku pisanja programa ne moremo vedeti, koliko vrstic bo imela tabela in če bo sploh imela kako vrstico, torej, če je uporabnik prvič izbral izdelek iz kataloga. Vemo pa, da bo imela tabela glavo in nogo. Zato bi lahko oblikovanje le teh sprožila že koda HTML, vendar tega zaradi subjektivnih razlogov nisem storil. Tabeli preko skriptov torej ustvarim ustrezno število objektov "child" od `Thead` (Glava tabele) `Row` (Vrstica) `cell` (celica v vrstici) `Tfoot` (noga tabele), poleg tega še razne dodatne lastnosti `caption` (napis) in podobno. V tako ustvarjene celice sproti ustavljamo podatke o izdelkih, zabeleženih v piškotku. V piškotku mora biti vsaj en izdelek, saj smo neposredno, preden smo odprli okno "Košarica", vnesli podatke o izdelku (procedura `klik()` iz kataloga zapiše piškotek in takoj nato kliče `odpri()`, s katero odpremo košarico). Sama procedura `zapisiTabelo()` ne prebira vrednosti iz piškotka, temveč kliče funkcijo `Izdelki()`, ki to stori in se vrne v obliki dvodimenzionalnega polja spremenljivk. Prva dimenzija so izdelki, druga pa vrednosti (količina, cena in ime izdelka). Torej imamo v trenutku, ko se izvajanje funkcije `zapisiTabelo()` konča, izdelan kompleten objektni model tabele z vstavljenimi vrednostmi po posamičnih vrsticah, za kar poskrbi funkcija `Izdelki()`. Ta je sicer napisana v skladu z načelom "Kar ne ubije, krepi", pa vendar deluje. Uporabiti bi moral objekt `RegExp` in s pomočjo urejenih izrazov (Regular Expressions) iz piškotka potegniti ustrezne vrednosti. Zaradi kompatibilnosti s starejšimi verzijami IE tega nisem storil, IE 4. namreč `RegExp` ne pozna. Tudi samo proceduro `zapisiTabelo()` bi sedaj razdelil na posamične procedure in jih klical iz nje. Tako bi si olajšal spreminjanje tabele in vključevanja dodatnih stolpcev na primer za barvo izdelka. Ob izhodu iz procedure `zapisiTabelo()` dobimo Košarico, kot jo kaže slika 4.1. Uporabnik ima naslednje možnosti. Lahko:

- popravlja količine (naloge vnosnih polij stolpca Količina opravljajo procedure druge skupine),
- odstranjuje izdelke iz košarice (celica "Odstrani"),
- potrdi vnesene izdelke in količine in se vrne v katalog (celica "V redu"),
- odda naročilo.

Opravila, povezana z prvimi tremi možnostmi, opravljajo že opisane procedure iz prve in druge skupine procedur in predstavljajo pravzaprav procedure za urejanje tabel. S programom so povezane le zato, ker tabela prikazuje naše izdelke. Ključna je zadnja možnost oddaje naročila. To sproži uporabnik s klikom na celico "Pošlji". V tem trenutku se spletni strani razdela objektni model obrazca, ki je s kodo HTML samo nakazan. Vstavijo se torej vnosna polja za podatke o naročniku. Spletna stran izgleda kot na sliki 4.2.

Zap.št.	Ime izdelka	Cena	Količina	Za plačilo	Pošlji
1	Vžigalnik nav. mlečni	20,00 SIT	1	20,00 SIT	odstrani
2	Vžigalnik elektronski mlečni	50,00 SIT	1	50,00 SIT	odstrani
3	Vžigalnik ploščat mlečni	80,00 SIT	1	80,00 SIT	odstrani
ZA PLAČILO:				150,00 SIT	V redu

Košarica je izdelek podjetja ANKLEM; tel.: 041/507-485

Vaše ime: Vaš priimek: Vaš telefon:

Slika 4.2: Košarica z naročnikovimi podatki

V prejšnjih poglavjih sem se odločil za preverjanje naročila in plačilo po povzetju. Zato potrebujemo samo podatke o imenu in priimku stranke ter o telefonu, na katerem preverimo naročilo in dobimo podatke o naslovu. Naslov bi lahko dobili tudi iz obrazca, vendar s tem nočem obremenjevati kupca. Zato so potrebni so samo trije vnosi. Kaj se dogaja s programom v trenutku, ko uporabnik pritisne na celico "Pošlji"? Sproži se procedura `PosNar()`, ki ima dve nalogi. Najprej iz tabele pobere vse podatke o naročenih izdelkih in jih spravi v globalno spremenljivko `tblPodatki`. Nato pa ustvari tri vnosna polja in gumb "Pošlji!". Bolje bi bilo, če bi ga imenoval "Oddaj!". Obnašanje vnosnih polj uravnavajo procedure iz prve skupine in tu tega ne bom ponavljal. Izvajanje programa spet

steče le ob pritisku na **gumb** "Pošlji". Ta akcija sproži proceduro

`Form1_onsubmit()`, ki ima štiri naloge. Te so:

- da preveri pravilnost vnosa,
- da pripravi besedilo e-mail sporočila, ki ga bo prejel uporabnik,
- da zbriše vse izdelke iz košarice (na ta način izpraznimo kasneje piškotek),
- in da zapre košarico.

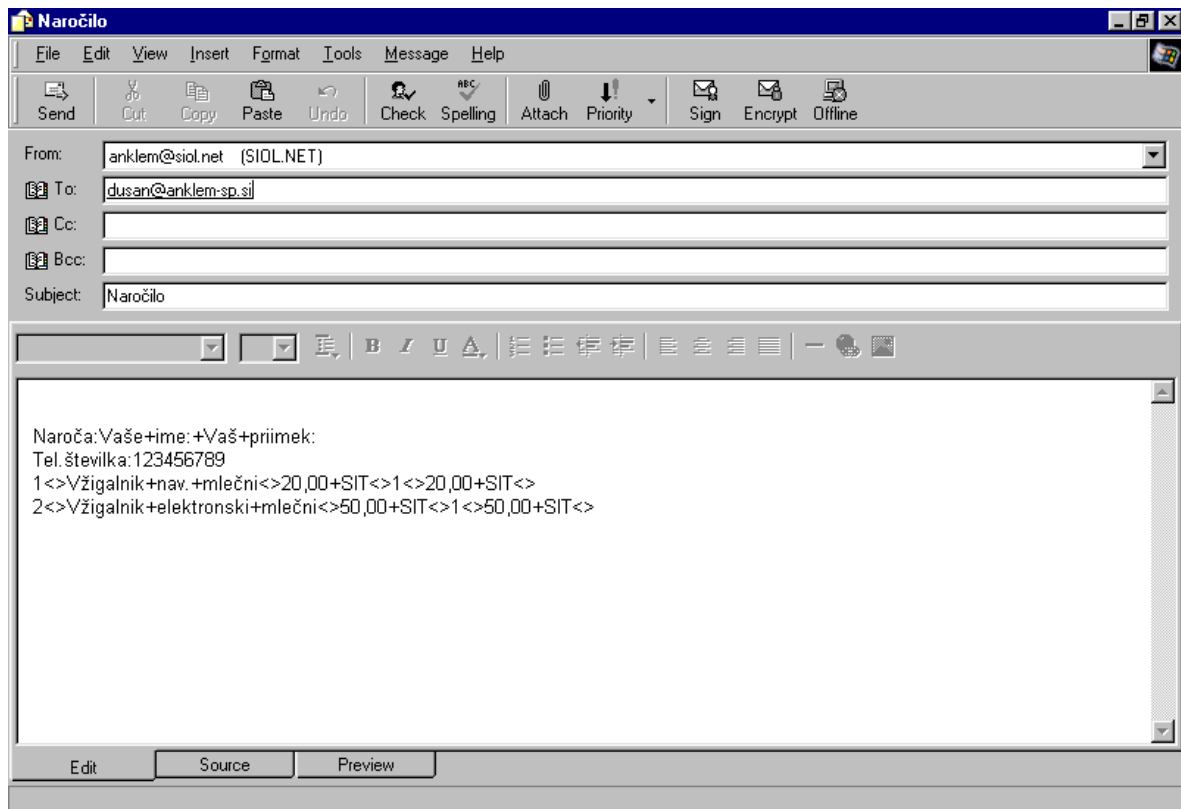
Zapiranje košarice sproži **vedno** tudi dogodek `onBeforeUnload` objekta `window`, ki sproži odzivno proceduro `Window_onBeforeUnload()`. Ta preveri stanje globalne logične spremenljivke `brezKukija` in pokliče proceduro `zapisKuki()`, ob tem pa ustrezno nastavi samo spremenljivko `brezKukija` za razhroščevanje. Vendar:

- Če je do zapiranja košarice prišlo zaradi klika na **gumb** "Pošlji!", bo to sprožilo proceduro `Form1_onsubmit()`, ki bo zbrisala tabelo, zato bodo vrednosti za lastnosti prazne in bo s tem v bistvu piškotek prazen.
- Če pa je do zapiranja košarice prišlo z zapiranjem košarice bodisi s klikom na celico "V redu" bodisi z standardnim zapiranjem brskalnikovega okna, kar tudi sproži proceduro `Window_onBeforeUnload`, **ne bo** klicana procedura `Form1_onsubmit()`, ki bi zbrisala tabelo, in v piškotek se bodo zato zapisali vsi izdelki.

Če poenostavimo, procedura `Window_onBeforeUnload` vedno kliče metodo `zapisKuki()`, vendar se v primeru, da je do zapiranja prišlo zaradi klika na celico "V redu", zapiše tabela z vsemi vrsticami (Izdelki). Če pa je do zapiranja prišlo zaradi pritiska na **gumb** "Pošlji!", potem se sproži procedura `Form1_onsubmit()`, ki zbriše vrstice tabele s podatki o izdelkih, nato pa procedura `zapisKuki()` zapiše prazno tabelo brez vrstic. Torej so vrednosti za ključ v piškotku nič. Tako se lahko vpisovanje naročil preko piškotka nadaljuje.

Ob kliku na gumb "Pošlji!" se aktivira ACTION metoda obrazca, ki sproži v parametru naveden protokol `mailto` z e-mail naslovom trgovca, kar spet sproži privzetega poštnega odjemalca s pripravljenim novim e-mail sporočilom, v katerem je privzet poštni naslov kupca (uporabnik), v <zadevi> sporočila je beseda "naročilo", v <telesu> sporočila pa je po izdelkih, cenah in količini razdelano naročilo. Vse to skoraj brez programiranja, edino v proceduri `Form1_onsubmit()` smo nastavili lastnost `form1.body.value`. V obrazec pa

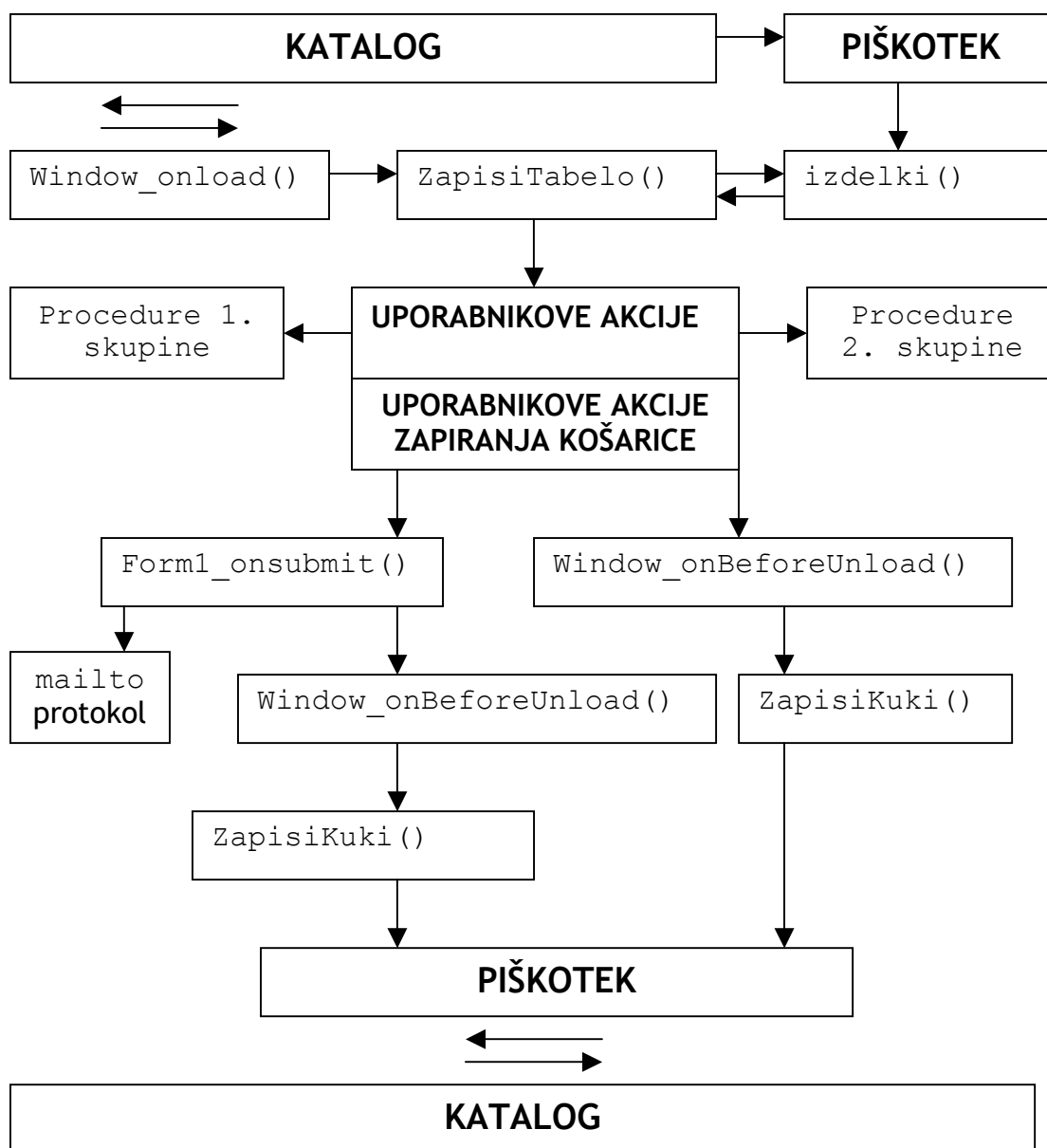
sem že med ustvarjanjem objektne strukture s proceduro `posNar()` vstavil dve skriti polji z imeni (lastnost: `name`) standardne vrednosti mail sporočila: `subject` in `body`. Protokol Mailto namreč uporabi imena kontrol obrazca, če ustrezajo zaglavljem elektronskih sporočil, tako da vrednosti v teh kontrolah vnese kot vrednosti oznak elektronskih sporočil. To ne velja za oznako `To`, saj se ta navede takoj za navedbo protokola in dvopičjem v parametru obrazca ACTION (v HTML kodi oznaka FORM).



Slika 4.3: primer elektronskega sporočila privzetega poštnega odjemalca z naročilom

Kot vidimo s slike (slika 4.3), bi načelno lahko v `form1.body.value` vstavili ustrezne oznake XML (npr.: `<cena>` namesto `<>` kot v primeru). Te so na sliki (slika 4.3) primera prazne in niso strukturirane v obliki drevesne strukture, kot demonstracija pa kažejo, da lahko zadeva lepo deluje. Kot vemo, bi lahko tako za elektronsko sporočilo uporabil ustrezen skladiščni program, izluščil naročilo s pomočjo oznak XML v telesu sporočila in sprožil nadaljnjo obdelavo oziroma postopek pošiljanja. Težava je le v protokolu mailto, ki v tem primeru prenaša sporočila v nekodirani obliki. Za varnostno izpopolnitev bi mogoče morali sprožiti certificiran podpis. Naj zaključim to poglavje in s tem nalogo s shemo toka dogajanja na spletni strani košarica:

Shema programskega toka spletne strani košarica:



Shema 4.2: Procedure v košarici

5. PRILOGA:

Prilogo naloge sestavljajo tri datoteke: scripti.htm, trg1.asp, trg2.asp. Prva predstavlja spletno košarico, drugi dve pa skupaj sestavljata katalog izdelkov z dvema blagovnim skupinama. Ker se datoteki trg1.asp in trg2.asp razlikujeta le v eni vrstici, bom datoteko trg2.asp priložil le do konca ASP procedure, v kateri bo razlika med datotekama (SQL stavek) označena odebeljeno. Priložena bi morala biti še access baza podatkov, vendar se je ne da zapisati v tekstovni obliki. Iz klica ASP procedure se lahko vidi zgradba tabele Izdelki, ki se nahaja v bazi anklem.mdb.

5.1. scripti.htm

```
<HTML>
<HEAD>
<META name=VI60_defaultClientScript content=VBScript>
<STYLE type="text/css">
  DIV {contenteditable:true; text-align:center;
    overflow:hidden;
    clip:rect(0px 20px 20px,0px);
    white-space:nowrap;}
</STYLE>
<SCRIPT LANGUAGE=VBScript>
<!--
Const stVnostnihMest = 2      'vnos števila mest
Const imKukija = "bolajcek"
const kolicina = "Kolicina"
dim pljLastnosti(1)
  pljLastnosti(0) = "ImeIzd"
  pljLastnosti(1) = "Cena"
dim glava, napis, vrstcGlv, clcGlv, noga, vrstcNoga, vrstica
dim obrazec
dim txtIme
dim txtPriimek
dim txtZadeva
dim txtStGsm
dim cmdPoslji
dim vsebina
dim tblPodatki
dim brezKukija
dim blnPosNar
'-----

Function izdelki ()
On Error Resume Next
dim kuki
kuki = "imeKukija=" & document.cookie
if InStr(kuki,ASP) <> 0 or eval(InStr(kuki,ASP) = Null) Then
  dim konec
  konec = InStr(kuki,"ASP")
  kuki = Left(kuki,konec-3)
End If
plj = split(kuki,";",-1,vbTextCompare)
```

```

dim pljIzd()
For m = 0 To UBound(plj)
    Redim Preserve pljIzd(m)
    pljIzd(m) = split(plj(m), "=", -1, vbTextCompare)
next
stevecpolja=split(pljIzd(1)(1), "~", -1, vbTextCompare)
If Err.Number <> 0 Then
    msgbox "Košarica je prazna!", 16, "      K O Š A R I C A"
    close
    exit Function
else
End if
dim kolone()
dim c
c=UBound(pljIzd)
for h=1 To UBound(pljIzd)
    Redim Preserve kolone(h-1)
    kolone(h-1)=split(pljIzd(h)(1), "~", -1, vbTextCompare)
next
dim stVrstic, stKolon
stVrstic = UBound(kolone(0)) 'stVrstic = CInt(fix(UBound(kolone(0))))
stKolon = Ubound(kolone) 'stKolon = CInt(fix(Ubound(kolone)))
dim Vrstice()
ReDim Vrstice(stVrstic, stKolon)
for p=0 to stKolon
    for r=0 to stVrstic
        vrstice(r,p) =kolone(p)(r)
        'pri tridemenzionalni tabeli bi bila tu še ena vrstica
        'cca vrstice(r,p,X) =kolone(p)(X) ?????????????prav vprašanje?
    next
next
izdelki = vrstice
End Function
'-----

Sub zapisiTabelo()
on error resume next
dim stvVrstic
dim stvKolon
set napis = objTabela.createCaption()
napis.align = "center"
napis.style.fontSize = "20"
napis.style.fontWeight="bold"
napis.style.Color = "red"
napis.align = "center"
napis.innerText="K O Š A R I C A"
'set noga = objTabela.createTFoot()
'noga.innerText = _
'"Kosarica je izdelek podjetja ANKLEM; Zaloska 178/a, Ljubljana"
set glava = objTabela.createTHead()
set vrstcGlv = glava.insertRow()
vrstcGlv.bgColor = "blue"
vrstcGlv.style.Color = "White"
pljNslVrst = array("Zap.št." _
, "Ime izdelka" _
, "Cena" _
, "Količina" _
, "Za plačilo" _
, "Pošlji")
for i=0 To 5
    set clcGlv = vrstcGlv.insertCell()

```

```

        clcGlv.align="center"
        clcGlv.style.fontWeight="bold"
        clcGlv.innerText = pljNslVrst(i)
    next
    vrstcGlv.cells(5).id = "clcPoslji"
    clcPoslji.onMouseOut = GetRef("mout")
    clcPoslji.onMouseOver = GetRef("mover")
    clcPoslji.onClick = GetRef("posNar")
    stvVrstic = UBound(izdelki(),1)
    If Err.Number <> 0 Then
        exit Sub
    else
    End if
    stvKolon = UBound(izdelki(),2)
    for j=0 To stvVrstic
        vrstica="vrstica"&j
        set vrstica = teloVrstice.insertRow(j)
        vrstica.bgColor = "lightskyblue"
        vrstica.id = "vrsticaID"&j
        'alert "sporočamo vam vrstični ID=" & vrstica.id & _
        '      " sirina tabele = " & objTabela.width
        dim a, vnos
        a=5
        for i=0 To a
            celica = "celica"&i
            set celica = vrstica.insertCell(i)
            select case i
                case 0
                    'zaporedna številka
                    celica.id= i&"celica"&j
                    celica.innerText = j+1
                case (a-3)
                    'cena
                    celica.id = i&"celica"&j
                    celica.innerText =formatCurrency _
                        (izdelki() (j,stvKolon-1),2) 'i*3
                case (a-2)
                    'količina
                    celica.bgColor = "LightSteelBlue"
                    celica.id = i&"kolicina"&j
                    vnos = "vnos"&a
                    set vnos = document.createElement _
                        ("<INPUT TYPE='text'>")
                    celica.appendChild(vnos)
                    vnos.value = izdelki() (j,stvKolon)
                    vnos.size = stVnostnihMest
                    vnos.maxlength = stVnostnihMest
                    vnos.tabindex = j
                    vnos.style.backgroundColor = LightSteelBlue"
                    set vnos.onclick = GetRef("klik")
                    set vnos.ondeactivate = GetRef("izracunaj")
                    set vnos.onKeyPress = GetRef("entr")
                case a-1
                    'vsota oz. za plačilo
                    celica.id = i&"celica"&j
                    celica.innerText = formatCurrency _
                        (vrstica.cells(a-3) _
                        .innerText*vnos.value)
                case a
                    'odstrani
                    celica.id = i&"celica"&j
                    celica.innerText = "odstrani"
                    celica.style.fontWeight="bold"
                case else
                    'ime Izdelka
                    celica.id = i&"celica"&j ' (j+1)&".izdelek"
                    celica.innerText = izdelki() (j,stvKolon-2

```

```

        End Select
        celica.align = "center"
    next
    For z=0 To teloVrstice.rows(j).cells.length-1
        'alert teloVrstice.rows(j).cells(z).id
    next
next
set noga = objTabela.createTFoot()
set vrstcNoga = noga.insertRow()
vrstcNoga.backgroundColor = "blue"
vrstcNoga.style.Color = "White"
dim b, clcNoga
b=2
for k=0 To b
    clcNoga = "clcNoga"&k
    set clcNoga = vrstcNoga.insertCell(k)
next
vrstcNoga.cells(0).colspan = vrstcGlv.cells.length-2
vrstcNoga.cells(0).innerText = "Z A P L A Č I L O :"
vrstcNoga.cells(0).style.FontWeight = "bold"
vrstcNoga.cells(0).align = "Right"
vrstcNoga.cells(1).style.FontWeight = "bold"
vrstcNoga.cells(1).innerText = skupaj()
vrstcNoga.cells(1).align = "center"
vrstcNoga.cells(2).align = "center"
vrstcNoga.cells(2).innerText = "V redu"
vrstcNoga.cells(2).style.FontWeight = "bold"
vrstcNoga.cells(2).id = "potrdi"
potrdi.onclick = GetRef("mclik")
potrdi.onMouseout = GetRef("mout")
potrdi.onMouseover = GetRef("mover")
'alert potrdi.offsetWidth&" "&potrdi.parentElement.offsetWidth
End Sub
'-----

Sub mclik ()
window.close
end Sub
'-----

Sub mover ()
me.style.backgroundColor = "Red"
me.style.cursor = "hand"
me.style.color = "White"
end Sub
'-----

Sub mout ()
me.style.backgroundColor = "blue"
me.style.cursor = "default"
me.style.color = "White"
end Sub
'-----

Sub Window_onLoad ()
zapisiTabelo
End Sub
'-----

Function skupaj()
Dim skVsota

```

```

for d=1 To ObjTabela.rows.length-2
    skVsota = skVsota+CInt _
        (objTabela.rows(d).cells(objTabela.rows(d).cells.length-2) _
        .innerText) ' (objTabela.rows(d).length-2).innerText _
next
skupaj = formatCurrency(skVsota)
End Function
'-----

Sub objTabela_onMouseout()
if Window.event.srcElement.innerText="odstrani" Then
    Window.event.srcElement.style.backgroundColor = ""
    Window.event.srcElement.style.cursor = "default"
    Window.event.srcElement.style.color = "Black"
End If
End Sub
'-----

Sub objTabela_onMouseover()
if Window.event.srcElement.innerText="odstrani" Then
    Window.event.srcElement.style.backgroundColor = "Red"
    Window.event.srcElement.style.cursor = "hand"
    Window.event.srcElement.style.color = "White"
    'Window.event.srcElement.style.textDecorationBlink = true
End If
End Sub
'-----

Sub objTabela_onclick()
'alert Window.event.srcElement.id & _
' " " & Window.event.srcElement.parentElement.id
if Window.event.srcElement.innerText="odstrani" Then
    teloVrstice.deleteRow _
        (Window.event.srcElement.parentElement.sectionRowIndex)
    vrstcNoga.cells(1).innerText = formatCurrency(skupaj())
End if
End Sub
'-----

Function klik()
me.select()
klik = Window.event.srcElement.value
end Function
'-----

sub izracunaj()
if isNumeric(me.value) Then
    if me.value > 0 then
        me.parentElement.nextSibling.innerText = formatCurrency _
            (me.value*me.parentElement.previousSibling.innerText)
        vrstcNoga.cells(1).innerText = skupaj()
    else
        me.value=1
        me.parentElement.nextSibling.innerText = formatCurrency _
            (me.value*me.parentElement.previousSibling.innerText)
        vrstcNoga.cells(1).innerText = skupaj()
    end if
else
    me.value=1
    me.parentElement.nextSibling.innerText = formatCurrency _
        (me.value*me.parentElement.previousSibling.innerText)

```

```

        vrstcNoga.cells(1).innerText = skupaj()
End if
vrstcNoga.cells(1).innerText = skupaj()
End Sub
'-----

sub entr()
if window.event.keyCode = 13 Then
    if isNumeric(me.value) Then
        if me.value > 0 then
            me.parentElement.nextSibling.innerText = formatCurrency _
                (me.value*me.parentElement.previousSibling.innerText) _
            vrstcNoga.cells(1).innerText = skupaj()
        else
            me.value=1
            me.parentElement.nextSibling.innerText = formatCurrency _
                (me.value*me.parentElement.previousSibling.innerText) _
            vrstcNoga.cells(1).innerText = skupaj()
        end if
    else
        me.value=1
        me.parentElement.nextSibling.innerText = formatCurrency _
            (me.value*me.parentElement.previousSibling.innerText) _
        vrstcNoga.cells(1).innerText = skupaj()
    End if
End if
End Sub
'-----

Sub objTabela_onResize()
external.dialogWidth = 650 & "Px"
external.dialogHeight = (objTabela.clientHeight+100) & "Px"
dim ty, tx
tx= window.screen.availWidth/2-window.document.body.clientWidth/2
ty=window.screen.availHeight/2-window.document.body.clientHeight/2
'alert tx & "    " & ty
external.dialogTop = ty
external.dialogLeft = tx
End Sub
'-----

Sub zapisikuki ()
on error resume next
dim kukiText(1)
for e=1 To objTabela.rows(1).cells.length-4
    if e<>objTabela.rows(1).cells.length-4 Then
        for f=1 To objTabela.rows.length-2
            if f=1 Then
                kukiText(e-1) = objTabela.rows(f).cells(e).innerText
            else
                kukiText(e-1) = kukiText(e-1)&"~"& _
                    objTabela.rows(f).cells(e).innerText
            end if
        next
    else
        for f=1 To objTabela.rows.length-2
            if f=1 Then
                kukiText(e-1) = CInt(objTabela.rows(f).cells(e).innerText)
                If Err.Number <> 0 Then
                    'msgbox err.number
                    exit Sub
                End if
            end if
        next
    end if
end for
end sub

```

```

        End if
    else
        kukiText(e-1) = kukiText(e-1) & "~" & Cint _
            (objTabela.rows(f).cells(e).innerText)
    end if
    next
end if
next
'dolocamo količino
dim kukiKolicina
for f=1 To objTabela.rows.length-2
    if f=1 then
        kukiKolicina = objTabela.rows(f).cells _
            (objTabela.rows.item(1).cells.length-3).childNodes(0).value
    else
        kukiKolicina = kukiKolicina & "~" & _
            objTabela.rows(f).cells(objTabela.rows.item(1).cells.length-3). _
            childNodes(0).value
    end If
next
document.cookie = imKukija
for k=0 To UBound(kukiText)
    document.cookie = pljLastnosti(k) & "=" & kukiText(k)
next
document.cookie = kolicina & "=" & kukiKolicina
>alert document.cookie & vbcrLf & "V zapisiKuki script.html"
End Sub
'-----

Sub Window_onBeforeUnload ()
if brezKukija = true then
    document.cookie = imKukija
    brezKukija = false
else
    call zapisikuki()
end if
End Sub
'-----

sub posNar ()
if txtIme = Empty then
    set txtIme = document.createElement _
        ("<INPUT TYPE='text' id='ime' name='ime'>")
    form1.appendChild(txtIme)
    txtIme.value = "Vaše ime:"
    txtIme.onFocusin = GetRef("brisemo")
    set txtPriimek = document.createElement _
        ("<INPUT TYPE='text' id='priimek' name='priimek'>")
    form1.appendChild(txtPriimek)
    txtPriimek.value = "Vaš priimek:"
    txtPriimek.onFocusin = GetRef("brisemo")
    set txtStGsm = document.createElement _
        ("<INPUT TYPE='text' id='stgsm' name='stgsm' CausesValidation='true'>")
    form1.appendChild(txtStGsm)
    txtStGsm.value = "Vaš telefon:"
    txtStGsm.onFocusin = GetRef("brisemo")
    set txtZadeva = document.createElement _
        ("<INPUT TYPE='hidden' NAME='Subject' VALUE='Naročilo'>")
    form1.appendChild(txtZadeva)
    vsebina = txtIme.value & "+" & txtPriimek.value
    set txtVsebina = document.createElement _

```

```

        ("<INPUT TYPE='hidden' NAME='body' VALUE=" & vsebina & ">")
form1.appendChild(txtVsebina)
set cmdPoslji = document.createElement _
    ("<INPUT TYPE='Submit' id='cmdPoslji'
NAME='CmdPoslji'>")
form1.appendChild(cmdPoslji)
cmdPoslji.value = "Pošlji!"
form1.style.position = "relative"
form1.style.Top = -15
form1.style.width = 515
fx = window.document.body.clientWidth/2 - left _
    (form1.style.width, len(form1.style.width)-2)/2
form1.style.left = fx
dim stRows
dim stCelice
for stRows = 0 to teloVrstice.rows.length-1
    for stCelice = 0 To teloVrstice.rows(stRows).Cells.length-2
        if instr(1,teloVrstice.rows
            (stRows).Cells(stCelice).id,"kolicina",vbTextCompare) _
            <> 0 then
            'msgbox "Količina = " & _
            'teloVrstice.rows(stRows). _
            'Cells(stCelice).childNodes(0).value
            tblPodatki = tblPodatki & _
                teloVrstice.rows(stRows). _
                Cells(stCelice).childNodes(0).value & "<>"
        else
            'msgbox teloVrstice.rows(stRows). _
            'Cells(stCelice).id & _
            '"= " & teloVrstice.rows(stRows). _
            'Cells(stCelice).innerText
            tblPodatki = tblPodatki & _
            teloVrstice.rows(stRows). _
            Cells(stCelice).innerText & "<>"
        End If
    next
    stCelice=0
    tblPodatki = tblPodatki & vbcrLf
next
Else
End If
end Sub
'-----

Sub form1_onsubmit()
txtStGsm.value = replace(txtStGsm.value,"/", "",1)
txtStGsm.value = replace(txtStGsm.value," ", "",1)
txtStGsm.value = replace(txtStGsm.value,"-", "",1)
txtStGsm.value = replace(txtStGsm.value,"+", "",1)
txtStGsm.value = replace(txtStGsm.value,"(", "",1)
txtStGsm.value = replace(txtStGsm.value,")", "",1)
if IsNumeric(txtStGsm.value)=false or len(txtStGsm.value)>13 Then
    msgbox "Vnesli ste napačno telefonsko številko!" & vbcrLf & _
        "Prosimo vas, da jo ponovno vpišete", 16 _
        , "N A P A K A - pri vnosu telefonske številke!"
    txtStGsm.value = "Vaš telefon:"
    window.event.returnValue = false
else
    form1.body.value = "Naroča:" & txtIme.value & " " & _
        txtPriimek.value & vbcrLf & _
        "Tel.številk:" & txtStGsm.value & vbcrLf & _

```



```

                                tblPodatki
    dim   vrste
    For   each vrste in teloVrstice.rows
        teloVrstice.removeChild(vrste)
    next
    window.setTimeout "close", 10000
End   If
End Sub
'-----

Sub   brisemo()
me.value = ""
End   Sub
'-----

//-->
</SCRIPT>
</HEAD>
<BODY BGCOLOR="gold" ID="telo">
<TABLE ID=objTabela
width="85%"
ALIGN=center
border=2
cellspacing=0
cellpadding=1
COLOR="RED">
<TBODY ID=teloVrstice>
</TBODY>
</TABLE>
<CENTER><SPAN STYLE="FONT-SIZE: xx-small;
FONT-FAMILY: verdana">Košarica je izdelek podjetja ANKLEM; tel.:
041/507-485      </SPAN></CENTER>
<FORM ACTION="mailto:alena@anklem-sp.si" METHOD=Get id=form1 name=form1>
</FORM></CENTER>
</BODY>
</HTML>

```

5.2. trg1.asp

```

<%@ Language=VBScript %>
<HTML>
<HEAD>
<META name=VI60_DTCScriptingPlatform content="Server (ASP)">
<META name=VI60_defaultClientScript content=VBScript>

<%
NizPovezave="Provider=Microsoft.Jet.OLEDB.4.0;" & _
"Data Source =c:\Moji dokumenti\BazaAnklem\Anklem.mdb"
NizSQL="SELECT * FROM Izdelki "& _
        "WHERE Ime LIKE 'vžigal%' or" & _
        " Ime LIKE 'svin%' or " & _
        "Ime LIKE 'stek%' " & _
        "ORDER BY ID ASC"
Set Podatki = Server.CreateObject("ADODB.Recordset")
podatki.CursorLocation = 2
Podatki.CursorType = AdOpenStatic
Podatki.Open NizSQL, NizPovezave
Set izdIme = Podatki("Ime")
Set slika = Podatki("Slika")
Set izdCena = Podatki("Cena")
%>

<SCRIPT LANGUAGE="vbscript">

```

```

<!--
dim pljSlika, pljIzdIme, pljIzdCena
Dim StvPlj
dim dodani
dim tabela
'-----

sub odpri ()
tabela = showModalDialog _
("scripti.htm", _
0, _
"dialogwidth=650px;center:yes;resizable:on;status:no")
end sub
'-----

Public Sub klik()
dim lastnosti(2)
lastnosti(0) = "ImeIzd"
lastnosti(1) = "Cena"
lastnosti(2) = "Kolicina"
dim pljTmpIzdelki
pljTmpIzdelki = split(window.event.srcElement.name,"~")
dim izdelki
Redim izdelki(Ubound(pljTmpIzdelki)+1)
izdelki(Ubound(izdelki)) = 1
for j=0 To Ubound(izdelki)-1
    izdelki(j)= pljTmpIzdelki(j)
next
    if InStr(document.cookie,"bolajcek") and _
        InStr(document.cookie,lastnosti(0)&"=") then
        for i=0 To UBound(izdelki)
            zacetek = inStr(InStr _
            (document.cookie, lastnosti(i)),document.cookie,"=")+1
            konec = inStr(zacetek,document.cookie,";")
            document.cookie = lastnosti(i)&"="& _
                mid(document.cookie, zacetek, konec-zacetek)& _
                "~"&izdelki(i)
        next
    else
        document.cookie = "bolajcek"
        for i=0 To UBound(lastnosti)
            document.cookie = lastnosti(i)&"="&izdelki(i)
        next
    End If
'alert document.cookie & vbcrLf & "V KATALOGU"
call odpri()
End Sub
'-----

<%Dim j, strSlika, strIzdIme, strIzdCena
j=0
'Podatki.Sort "Ime ASC"Podatki.MoveFirst
Do While Not Podatki.EOF
    strSlika=strSlika&slika&"$"
    strIzdIme=strIzdIme&izdIme&"$"
    strIzdCena=strIzdCena&izdCena&"$"
    Podatki.MoveNext
    j=j+1
Loop
strSlika = Left(strSlika,len(strSlika)-1)
strIzdIme = Left(strIzdIme,len(strIzdIme)-1)
strIzdCena = Left(strIzdCena,len(strIzdCena)-1)

```

```

Podatki.close
set Podatki = nothing
%>
'-----

Sub window_onLoad()
dodani = 0
pljSlika=split("<%=strSlika%>", "$")
pljIzdIme=split("<%=strIzdIme%>", "$")
pljIzdCena=split("<%=strIzdCena%>", "$")
'msgbox UBound(pljSlika) & " " & join(pljSlika, ";") & vbCrLf & _
'UBound(pljIzdIme) & " " & join(pljIzdIme, ";") & vbCrLf & _
'UBound(pljIzdCena) & " " & join(pljIzdCena, ";")
dim tvrstice, tcelice, slikca, gumb, stvPlj
'stvPlj=0
for k=0 to 2
tvrstice = "tvrstice"&k
set tvrstice = teloTbl.insertRow()
    for m=0 to 2
        if stvPlj < UBound(pljSlika) + 1 Then
            tcelice="tcelice"&m
            set tcelice = tvrstice.insertCell()
            tcelice.style.fontFamily="Arial"
            tcelice.style.fontSize=12
            slikca = "slikca"&k&"_"&mp
            set slikca = document.createElement("IMG")
            tcelice.appendChild(slikca)
            'alert stvPlj
            slikca.src = pljSlika(stvPlj)
            slikca.style.visibility = visible
            slikca.style.display = "block"
            tcelice.innerHTML =tcelice.InnerHtml&pljIzdIme(stvPlj) &
            "
            "<br>Cena:
            "&formatCurrency(pljIzdCena(stvPlj)) &"<br>"
            gumb = "gumb"&k&"_"&mp
            set gumb = document.createElement("<INPUT
TYPE='button'>")
            'msgbox gumb.id
            gumb.value = "V košarico"
            gumb.name = pljIzdIme(stvPlj) &"~"&pljIzdCena(stvPlj)
            tcelice.appendChild(gumb)
            gumb.onClick = GetRef("klik")
            stvPlj = stvPlj+1
            dodani = dodani + 1
        End if
    next
next
End sub
'-----

Sub Naprej ()
if stvPlj > Ubound(pljSlika) Then
    'alert UBound(pljSlika) & " " &stvPlj
else
    dim vrstica
    For each vrstica in teloTbl.rows
        teloTbl.removeChild(vrstica)
    next
    call window_onLoad
End If

```

```

End Sub
'-----

Sub Nazaj ()
if stvPlj = 9 Then
else
    stvPlj = StvPlj - 9 - dodani
    dim vrstica
    For each vrstica in teloTbl.rows
        teloTbl.removeChild(vrstica)
    next
    call window_onLoad
End If
End Sub
'-----

sub pkzKosarco
tabela = showModalDialog("scripti.htm", _
0, _
"dialowidth=650px;center:yes;resizable:on;status:no")
end Sub

Sub pkzKuki ()
msgbox document.cookie
End Sub
-->
</SCRIPT>
</HEAD>
<BODY id=tjelo>
<TABLE border="0" id=tabla width="80%">
<TBODY ID=teloTbl name="telesce">
</TBODY>
</TABLE>
<P>
<INPUT type="button"
value="<"
id=cmdNazaj
name=cmdNazaj
onClick=Nazaj()
size=60
style="LEFT: 10px; WIDTH: 100px; TOP: 15px">
<INPUT type="button"
value=">"
id=cmdNaprej
name=cmdNaprej
onClick=Naprej()
size=60
style="WIDTH: 100px">
<INPUT id=cmdkosarca
style="WIDTH: 112px; HEIGHT: 24px"
type=button
size=38
value=Košarica
name=cmdKosarca
onClick=pkzKosarco()></P>
<P>Pojdi na <A href="http://dusan/Diploma/Trg2.asp">trg2</A>
<INPUT id=cmdPkzKuki
type=button
value="Pokazi kuki"
name=cmdPkzKuki
onclick=pkzKuki>
</P>

```

```
</BODY>
</HTML>
```

5.3. trg2.asp

```
<HTML>
<HEAD>
<META name=VI60_DTCScriptingPlatform content="Server (ASP)">
<META name=VI60_defaultClientScript content=VBScript>

<%
NizPovezave="Provider=Microsoft.Jet.OLEDB.4.0;" & _
"Data Source =c:\Moji dokumenti\BazaAnklem\Anklem.mdb"
NizSQL="SELECT * FROM Izdelki WHERE id>22 ORDER BY ID ASC"
Set Podatki = Server.CreateObject("ADODB.Recordset")
podatki.CursorLocation = 2
Podatki.CursorType = AdOpenStatic
Podatki.Open NizSQL, NizPovezave
Set izdIme = Podatki("Ime")
Set slika = Podatki("Slika")
Set izdCena = Podatki("Cena")
%>
```

VIRI:

1. (2002) "Nakupovanje na spletu" 2002/1, VIP - revija za potrošnike, Ljubljana
2. (2002) TNS Interactive - Global eCommerce report June-2002; Products Purchased / Reasons for not purchasing in 2002 - Across 37 countries, Taylor Nelson Sofres plc., London UK
<http://www.tns Sofres.com/ger2002/products purchased/index.cfm>
3. (april 2002) WOW! Facts 2002. Business Women's, Washington DC
4. (2002) 2002 Denver Scarborough Report. Scarborough Research, New York
<http://www.denvernewspaperagency.com/presentation/2002major-market.ppt>) slide 23
5. (2002/03) SIBIS Pocket Book 2002/03 Measuring the Information Society in the EU, the EU Accession Countries, Switzerland and the US, SIBIS - Empirica, Bonn
6. (Januar 2003) Targets for e-business policies for SMEs. SEDISI, Atene
7. (januar 2003) "Distribution by Top-Level Domain Name by Name". Internet Software Consortium - network wizzards <http://www.isc.org/ds/WWW-200301/dist-byname.html>, Redwood City CA 94063 USA
8. (2003) OECD Communications Outlook 2003, OECD Publications, Paris
9. (2003) RIPE Host Count, RIPE Network Coordination Centre, Amsterdam,
<http://www.ripe.net/ripenncc/pub-services/stats/hostcount/index.html>
10. Barua, Anitesh in Pinnell, Jon in Shutter, Jay in Whinston, B. Andrew (1999): Measuring the Internet Economy: An Exploratory Study. CONFERENCE ON THE MEASUREMENT OF ELECTRONIC COMMERCE, Singapur
11. Cabral, Luis M.B. (2000): Introduction to industrial organization. MIT press, Cambridge, Massachusetts
12. Cyveillance študija "Sizing the Internet" v Pastore, Michael (12.7.2000): "The Web: More Than 2 Billion Pages Strong". JupiterMedia, Darien CT
cyberatlas.internet.com/big_picture/traffic_patterns/article/0,,5931_4136_91,00.html
13. Damij, Talib (1998): Poslovna informatika. Ekonomska fakulteta, Ljubljana
14. Lebo, Harlan (februar 2003): The UCLA internet report - "Surveying the digital future" year three. UC Regents, LA
15. Lisser, Eleena de (2003): "Small Businesses Still Lag In E-Commerce Ventures"2, The Wall Street Journal on Line
<http://www.startupjournal.com/howto/soundadvice/199908180953-lisser.html>
16. Lah, Marko (1998): Marketinške implikacije ekonomskih teorij rasti podjetja. Akademija MM, Slovenska znanstvena revija za trženje, december 1998, št.3, str. 9-15.
17. Kirch, Olaf (1995): Linux Network Administrator's Guide. O'Reilly, Kalifornija
18. Kovačič, Andrej in Vintar, Mirko(1994): Načrtovanje in gradnja informacijskih sistemov. DZS, Ljubljana
19. Kramberger Anton, Ilič Branko, Kohont Andrej (2003): S strateško naravnanim menedžmentom do rasti in uspešnosti organizacije (v tisku)
20. Kutin, Breda in Lavrenčič, Jernej in Okoren, Boštjan in Velkavrh, Daša in Božičević, Urša (december 2001): Izvensodno reševanje potrošniških sporov

- na področju e . poslovanja - končno poročilo. Mednarodni inštitut za potrošniške raziskave in MID, Ljubljana
21. McGeachie, David (1999): Defining e-commerce. CONFERENCE ON THE MEASUREMENT OF ELECTRONIC COMMERCE, Singapur
 22. Mesojedec, Uroš (1996): Java programiranje za internet. Pasadena, Ljubljana
 23. O'Daniel, Thomas (1999): Modeling Electronic Commerce: Value-Added Roles. CONFERENCE ON THE MEASUREMENT OF ELECTRONIC COMMERCE, Singapur
 24. Oxa (2003): Study on Interoperability Service Diversity and Business Models in Digital Broadcasting Markets - Volume II. Oxford Economic Research Associates, Oxford str.: 83 tabela A3.5.1. Industry Evidence - Microsoft Približno 90% PC uporablja Microsoftov operacijski sistem
 25. Refsnes Data(2003): Browser statistics - http://www.w3schools.com/browsers/browsers_stats.asp. Refsnes Data, Norveška
 26. Sicherl, dr. Pavle (2001): Analiza časovnih distanc v letu 2001, Metodologija spremljanja časovnih distanc. MID, Ljubljana
[http://www2.gov.si/mid/mid.nsf/V/KBFC41C8C1DE8B278C1256C0C007230BA/\\$file/CD_metodologija.pdf](http://www2.gov.si/mid/mid.nsf/V/KBFC41C8C1DE8B278C1256C0C007230BA/$file/CD_metodologija.pdf)
 27. Sicherl, dr. Pavle (november 2002): Komparativna analiza izbranih indikatorjev položaja Slovenije s področja interneta, osebnih računalnikov in mobilnih telefonov. MID, Ljubljana
[http://www2.gov.si/mid/mid.nsf/V/K5AC827E791ADCBEB1256C85002A5BF5/\\$file/Sicherl_1_stevilo_PC.pdf](http://www2.gov.si/mid/mid.nsf/V/K5AC827E791ADCBEB1256C85002A5BF5/$file/Sicherl_1_stevilo_PC.pdf)
 28. Selhoffer, Hannes (november 2002) E-Business in EU in 2002, Results of an enterprise survey, e-business watch, Brussels
 29. Vehovar, Vasja (1999): Measuring Electronic Commerce with Sample Surveys - The Methodological Problems. CONFERENCE ON THE MEASUREMENT OF ELECTRONIC COMMERCE, Singapur
 30. Vehovar, Vasja in Kragelj, Boris (maj 2003): RIS 2003/2 - Gospodinjstva uporaba interneta, MID in Ministrstvo za šolstvo, znanost in šport, Ljubljana
 31. Vehovar, Vasja in Lavtar, Darja in Pfajfar, Alenka (April 2003): RIS2002 - Podjetja Internet in informacijske tehnologije. MID in Statistični urad RS, Ljubljana
 32. Vehovar, Vasja in Lobe, Bojana in Kovačič, Matej (junij 2003): Confidentiality concern and on-line shopping. FDV, Ljubljana
 33. Vehovar, Vasja in Pfajfar, Alenka in Podlogar, Mateja in Pucihar, Andreja (2003): RIS2002 Podjetja elektronsko poslovanje. MID in Ministrstvo za šolstvo, znanost in šport, Ljubljana
 34. Vehovar, Vasja in Remec, Matija (januar 2003): RIS2002 Informacijske tehnologije v gospodinjstvih, MID in Ministrstvo za šolstvo, znanost in šport, Ljubljana
 35. Vehovar, Vasja in Vukčević, Katja (oktober 2001): Digitalni razkorak - Slovenija 2001, MID, Ljubljana
 36. Waraker, Abigail (23.1.2003) European firms show interest in .biz. vnunet.com, <http://www.vnunet.com/News/1128587>
 37. Zook, Matthew (2001) Current statistics/Domain name Geography. Zooknic zaps, Berkeley - ZDA, <http://www.zooknic.com/Domains/index.html>