

**UNIVERZA V LJUBLJANI  
EKONOMSKA FAKULTETA**

**MAGISTRSKO DELO**

**BORIS ŠUŠMAK**



**UNIVERZA V LJUBLJANI  
EKONOMSKA FAKULTETA**

**MAGISTRSKO DELO**

**IZGRADNJA INFORMACIJSKEGA SISTEMA  
ZA POTREBE GIBANJA MODRA ZASTAVA V SLOVENIJI**

**LJUBLJANA, MAREC 2004**

**BORIS ŠUŠMAK**

### **IZJAVA**

Študent Boris Šušmak izjavljam, da sem avtor tega magistrskega dela, ki sem ga napisal pod mentorstvom prof. dr. Borke Jerman-Blažič in skladno s 1. odstavkom 21.člena Zakona o avtorskih in sorodnih pravicah dovolim objavo magistrskega dela na fakultetnih spletnih straneh.

V Ljubljani, dne \_\_\_\_\_

Podpis:

\_\_\_\_\_

# KAZALO

|   |    |
|---|----|
| 1. UVOD .....   | 1  |
| 2. MODRA ZASTAVA IN NJENE ZAHTEVE PRI ANALIZI<br>KAKOVOSTI VODE .....   | 3  |
| 2.1 MODRA ZASTAVA ZA MARINE.....  | 4  |
| 2.2 MODRA ZASTAVA ZA NARAVNA KOPALIŠČA.....   | 6  |
| 2.3 ZAHTEVE PRI ANALIZI KAKOVOSTI VODE.....   | 9  |
| 3. OPIS TRENUTNO DELUJOČEGA SISTEMA MORSKE BIOLOŠKE<br>POSTAJE PIRAN ZA ZAJEM, SHRAMBO IN PRENOS PODATKOV .....       | 11 |
| 3.1 OCEANOGRAFSKA BOJA.....   | 11 |
| 3.1.1 Oceanografska boja kot plovni objekt.....   | 12 |
| 3.1.2 Merilni instrumenti na oceanografski boji.....  | 13 |
| 3.1.3 Kontrolna enota na boji .....   | 15 |
| 3.2 POSTAJA ZA SPREJEM PODATKOV .....   | 17 |
| 3.3 PROGRAMSKA OPREMA .....   | 17 |
| 3.3.1 Programi za numerično obdelavo in prikaz podatkov .....   | 18 |
| 3.3.2 Strežniški programi .....   | 19 |
| 3.4 SISTEMSKI PODATKI .....   | 20 |
| 3.4.1 Zgradba datoteke z meritvami in statusne datoteke .....   | 20 |
| 3.4.3 Opis komunikacijskega protokola .....   | 21 |
| 3.4.4 Zgradba konfiguracijske datoteke .....  | 22 |
| 3.5 PREDLOG IN NAČRT OPERATIVNE NADGRADNJE OBSTOJEČEGA SISTEMA ZA POTREBE<br>MORSKE BIOLOŠKE POSTAJE PIRAN (MBP)..... | 23 |
| 4. IZKUŠNJE DRUGIH DRŽAV .....  | 25 |
| 4.1 POSEIDON .....  | 26 |
| 4.2 MAREL .....   | 28 |
| 4.3 SHARK .....   | 30 |
| 4.4 PUERTAS DEL ESTADOS .....   | 30 |
| 4.5 CANDHIS .....   | 31 |
| 4.6 M3A .....   | 31 |
| 4.7 MEDGOOS .....   | 32 |
| 4.8 EGOS .....  | 33 |
| 4.9 SEANET.....   | 33 |
| 4.10 FERRYBOX .....   | 33 |
| 5. METODOLOGIJE NAČRTOVANJA INFORMACIJSKIH SISTEMOV .....   | 33 |

|  |    |
|--|----|
| 5.1 INFORMACIJSKI SISTEM .....   | 33 |
| 5.2 RAZVOJ METODOLOGIJ IZGRADNJE INFORMACIJSKIH SISTEMOV .....                               | 34 |
| 5.3 OBJEKTNO USMERJENE METODOLOGIJE RAZVOJA INFORMACIJSKIH<br>SISTEMOV .....                 | 40 |
| 5.3.1 Osnovni koncepti objektnega pristopa .....   | 40 |
| 5.3.1.1 Objekt .....   | 40 |
| 5.3.1.2 Ograjevanje .....  | 41 |
| 5.3.1.3 Razred .....   | 42 |
| 5.3.1.4 Asociacije .....   | 43 |
| 5.3.1.5 Sporočilo .....  | 44 |
| 5.3.1.6 Mnogoličnost .....   | 44 |
| 5.3.1.7 Dedovanje .....  | 44 |
| 5.3.2 POENOTEN JEZIK ZA MODELIRANJE – UML .....  | 46 |
| 5.3.2.1 Osnovni gradniki UML .....   | 48 |
| 5.3.3 POENOTENI PROCES RAZVOJA PROGRAMSKE OPREME .....                                       | 50 |
| 5.3.3.1 Ključne aktivnosti procesa .....   | 51 |
| 5.3.3.2 Glavne faze procesa .....  | 56 |
| 6. IZGRADNJA INFORMACIJSKEGA SISTEMA, VEZANEGA NA<br>PROGRAM MODRA ZASTAVA .....             | 57 |
| 6.1 IDEJNI NAČRT .....   | 58 |
| 6.1.1 Zahteve in potrebe uporabnikov sistema .....   | 59 |
| 6.1.2 Analiza tehnologij in tehnik merjenja kakovosti vode .....                             | 61 |
| 6.2 NAČRTOVANJE INFORMACIJSKEGA SISTEMA S POMOČJO<br>POENOTENEGA JEZIKA ZA MODELIRANJE ..... | 64 |
| 6.2.1 Diagram primerov uporabe .....   | 64 |
| 6.2.2 Diagram zaporedja .....  | 66 |
| 6.2.3 Objektni diagram .....   | 67 |
| 6.2.4 Razredni diagram .....   | 68 |
| 6.2.5 Diagram aktivnosti .....   | 69 |
| 6.2.6 Diagram sodelovanja .....  | 70 |
| 6.2.7 Diagram stanj .....  | 71 |
| 6.2.8 Diagram porazdelitve .....   | 71 |
| 6.2.9 Diagram komponent .....  | 72 |
| 6.3 SESTAV INFORMACIJSKEGA SISTEMA, PROTOKOLI IN VAROVANJE PODATKOV .....                    | 73 |
| 6.3.1 Boja .....   | 73 |
| 6.3.2 Izbira sistema za upravljanje z bazo podatkov .....                                    | 76 |
| 6.3.3 Strežnik in periferna oprema .....   | 76 |
| 6.3.4 Programska oprema in uporabniški vmesnik .....   | 77 |
| 6.3.4.1 Uporabniški vmesnik Okna .....   | 78 |
| 6.3.4.2 Program Mbp_boja .....   | 88 |
| 6.3.4.3 Program Vpis .....   | 89 |

|  |     |
|--|-----|
| 6.3.4.4 Uporabniški vmesnik za končne uporabnike .....                 | 89  |
| 6.4 VZPOSTAVITEV BAZE PODATKOV ZA POTREBE INFORMACIJSKEGA SISTEMA..... | 91  |
| 6.4.1 Izdelava logičnega načrta .....                                  | 92  |
| 6.4.2 Izdelava fizičnega načrta .....                                  | 94  |
| 6.4.3 Gradnja in vzdrževanje baze podatkov .....                       | 96  |
| 6.5 EKONOMSKI VIDIK IZGRADNJE INFORMACIJSKEGA SISTEMA.....             | 97  |
| 7. SKLEP .....   | 99  |
| LITERATURA.....  | 100 |
| VIRI.....  | 103 |
| SLOVARČEK TUJIH IZRAZOV .....  | I   |
| PRILOGA.....   | 1   |





# 1. Uvod

Okoljska osveščenost in trajni razvoj zahtevata posebno obravnavo vodnih virov, ki so danes ne samo izredno redka dobrina, temveč tudi nujno potrebna za življenje na Zemlji.

Mednarodni program Modra zastava, ki ima za nalogo, da so marine in naravna kopališča čista, varna, okoljsko prijazna in da uporabniki uživajo v prijetnem okolju, želi zato spodbujati pri ljudeh aktiven odnos do naravnega okolja. Upravitelji marin in naravnih kopališč ter lokalne oblasti, ki izkazujejo posebno skrben odnos do lastnega ter širšega okolja, lahko prejmejo okoljski znak modra zastava. Pomembna osnova za pridobitev vsakoletne modre zastave je izpolnjevanje pogojev o kakovosti kopalne vode v naravnih kopališčih. Predpisane mikrobiološke ter fizikalno-kemijske podatke analizira in zbira v času kopalne sezone v Sloveniji pooblaščen javna ustanova – Zavod za zdravstveno varstvo.

Trenutno vzorčenje vode poteka na klasični, ročni način. Zavod za zdravstveno varstvo jemlje vzorce na vzorčnih mestih s pomočjo čolna in epruvet oz. drugih pripomočkov. Kasneje vzorce v laboratoriju analizirajo.

Avtomatizirano vzorčenje določenih parametrov za potrebe gibanja Modra zastava zaenkrat ne obstaja. Prav tako ni zgrajen informacijski sistem prenosa informacij med Zavodom za zdravstveno varstvo in dobitniki modre zastave ter drugimi zainteresiranimi subjekti (turistične organizacije, okoljevarstveniki, morjeplovci, drugi).

Z avtomatskim zbiranjem določenih parametrov morske vode s pomočjo boje se v Sloveniji ukvarja le Morska biološka postaja v Piranu. Modra zastava zahteva vrsto parametrov, ki pa jih obstoječa boja ne meri.

Avtomatske merilne naprave v svetu so v zadnjih letih izredno napredovale. Problemi, ki so pri avtomatskem zbiranju tovrstnih podatkov še vedno prisotni, pa so certificirana kalibracija instrumentov in zagotovitev neprestanega delovanja informacijskega sistema s povezavo med plovcem in ustrezno biološko postajo.

Izgradnja avtomatiziranega informacijskega sistema za potrebe modre zastave bi omogočila vsem vpletenim subjektom takojšnji vpogled v stanje kakovosti vode. Velik problem samega kriterija čistosti kopalne vode v primeru modre zastave je v tem, da je potrebno vzorce jemati s frekvenco največ 18 dni. V Sloveniji je praksa, da vzorce jemljejo tedensko. Znano je, da se voda v morju in drugih naravnih kopališčih

neprestano meša in je dobljeni podatek zato zavajajoč. S pomočjo informacijskega sistema in avtomatskega zbiranja vzorcev bi bila dejanska kakovost vode jasna. Prav tako bi tako zbrane podatke lahko uporabili tudi v primeru izrednih onesnaženj in za druge okoljevarstvene, turistične in informativne namene.

Zaradi večanja pomembnosti programa Modra zastava v svetu in vedno večjih potreb uporabnikov kopališč in marin sem, kot nacionalni koordinator tega programa v Sloveniji, prišel na idejo, da bi izdelal projekt izgradnje informacijskega sistema za potrebe gibanja Modra zastava in drugih zainteresiranih subjektov, s ciljem, da bi idejo po predstavitvi tudi udejanili v praksi. Idejo sem predstavil tudi delovni skupini prof. dr. Vlada Malačiča, ki deluje na Morski biološki postaji v Piranu in se ukvarja z kontinuiranim merjenjem oceanografskih podatkov s pomočjo boje pred Piranom, kjer so jo kot tako podprli in so za pomoč pri realizaciji tudi zainteresirani. Zaradi mednarodno priznanih dolgoletnih izkušenj Morske biološke postaje pri merjenju oceanografskih podatkov s pomočjo boje so določene rešitve uporabne tudi v samem projektu.

Projekt temelji na naslednji hipotezi: Izgradnja popolno avtomatiziranega informacijskega sistema bi olajšala delo analitikov, nudila večjo transparentnost dobljenih podatkov, večjo možnost analitičnega spremljanja rezultatov in dodatno »on-line« zagotovilo uporabnikom naravnih kopališč z modro zastavo, da se kopajo v čisti in neoporečni vodi. Projekt sem predstavil v magistrski nalogi, ki vsebuje sedem poglavij.

Drugo poglavje podaja kratek opis programa Modra zastava in podrobnosti v zvezi s kriteriji za podelitev tega okoljskega znaka. Tretje poglavje priča o najbolj znanem in delujočem sistemu vzorčenja oceanografskih podatkov s pomočjo boje v Sloveniji, ki lahko služi za pomoč pri izgradnji specifičnega informacijskega sistema za potrebe modre zastave. V četrtem poglavju so opisani delujoči sistemi merjenja kakovosti vode in drugih hidrometeoroloških parametrov s pomočjo boj v Evropi. V petem poglavju spregovorimo o metodologijah izgradnje informacijskih sistemov. V šestem poglavju je podana izgradnja informacijskega sistema za potrebe gibanja Modra zastava. Predstavljen je idejni načrt z zahtevami in potrebami uporabnikov ter morebitnimi omejitvami. V nadaljevanju je podan pregled tehnologij in tehnik merjenja kakovosti vode, model informacijskega sistema, sestav informacijskega sistema, načrt vzpostavitve baze podatkov in ekonomski vidik izgradnje tovrstnega sistema. V sedmem poglavju je naloga zaključena s sklepnimi mislimi.

## **2. Modra zastava in njene zahteve pri analizi kakovosti vode**

Program Modra zastava koordinira Fondacija za okoljsko vzgojo (angl. Foundation for Environmental Education - FEE International, v nadaljevanju FEE), izvaja pa ga koordinacija za Modro zastavo (angl. Blue Flag Coordination), ki ima sedež v Kopenhagenu na Danskem (The Blue Flag Campaign, 10. 1. 2003). Cilj programa je zagotavljanje kakovosti naravnih kopališč in marin z modro zastavo. Gre za uveljavljen in cenjen ekološki znak. To potrjuje tudi priznanje, ki ga je programu Modra zastava podelil nemški organizator potovanj TUI na turistični borzi v Berlinu v letu 2001, za svojega pa sta ga prevzeli UNEP (okoljski program Združenih narodov) in svetovna turistična organizacija.

V FEE je vključenih že več kot 33 držav: Belgija, Bolgarija, Ciper, Estonija, Danska, Finska, Francija, Grčija, Hrvaška, Irska, Italija, Južnoafriška republika, Združenje karibskih držav, Nemčija, Nizozemska, Portugalska, Romunija, Slovenija, Španija, Švedska, Turčija in Velika Britanija, interes pa izkazujejo še druge države, tako v Evropi kot izven nje.

V državah članicah FEE poteka več celovitih programov: Eko schools - Eko šole, Blue Flag - Modra zastava in Young Reporters for Environment - Mladi poročevalci o okolju ter Learning about Forests - Spoznavajmo gozdove. Okoljski znak modra zastava podeljuje Evropska žirija na predlog Nacionalne komisije vsake države članice FEE. Za pridobitev priznanja modra zastava lahko na podlagi vsakoletnega razpisa kandidirajo kraji ter njihova naravna kopališča na morski obali, na obrežjih stoječih voda v notranjosti dežele in marine. Nacionalna komisija in Evropska žirija za podeljevanje lahko znak modre zastave (v nadaljevanju modra zastava) odzamera že med sezono, če bi se ugotovilo neizpolnjevanje kriterijev programa Modra zastava. Njihovi upravljalci zagotavljajo izpolnitev številnih kriterijev, ki so razdeljeni v naslednje štiri skupine:

- okoljska vzgoja ter informiranje,
- okolju prijazno poslovanje,
- kakovost kopalne vode,
- varnost in storitve.

Dobitniki modre zastave zagotavljajo uporabnikom čisto, varno in prijetno okolje, namenjenemu rekreaciji v vodi in ob njej. O tem morajo svoje uporabnike tudi

obveščati in izobraževati.

Evropska žirija je na predlog Nacionalnih komisij za modro zastavo, za leto 2003 podelila 2161 modrih zastav naravnim kopališčem ter 729 marinam, skupno torej 2890 modrih zastav, kar je največ doslej. V Sloveniji je v letu 2003 pridobilo modro zastavo sedem naravnih kopališč (Centralna plaža Okolje Portorož, Kopališče Krka – Zdravilišče Strunjan, Kopališče Simonov zaliv, Kopališče Žusterna, Mestno kopališče Koper, Kopališče Šobčev bajer in Grajsko kopališče Bled) in tri marine (Marina Portorož, Marina Izola, Marina Koper).

## **2.1 Modra zastava za marine**

Modra zastava za marine ne potrebuje vzorčenja vode, saj kriteriji pri marinah tega ne zahtevajo. Osnovne zahteve za pridobitev modre zastave za marine pa so slednje:

Marina z modro zastavo (v nadaljevanju: marina) je marina s pomoli ali s pontoni, namenjena plovilom za oddih. Lahko je del večjega pristanišča, ki izvaja tudi druge aktivnosti. V tem primeru mora biti marina jasno ločena od takih drugih aktivnosti. Opremljena mora biti tako, kot to zahtevajo kriteriji programa Modra zastava. V njej mora biti določena odgovorna oseba za izvajanje aktivnosti programa Modra zastava. V marino mora biti dovoljen dostop predstavniku programa Modra zastava zaradi tudi nenapovedanega kontrolnega ugotavljanja izpolnjevanja veljavnih kriterijev za modro zastavo.

Spodaj je prikazanih 21 osnovnih kriterijev, razdeljenih v štiri podskupine, za pridobitev modre zastave za marine, ki bodo stopili v veljavo leta 2004. »I« pomeni obvezno izpolnjevanje kriterija, »G« priporočeno izpolnjevanje kriterija, ki ga mora marina izpolnjevati, če želi pridobiti znak modra zastava.

### **A. Okoljska vzgoja in informiranje**

1. Uporabnikom marine so na razpolago okoljske informacije o občutljivih območjih na prostoru marine ali v njeni soseščini (I).
2. V marini je izobešen kodeks o okoljskem obnašanju (I).
3. V marini so javno predstavljene informacije o programu Modra zastava za marine in/ali kriteriji modre zastave za marine (I).
4. Marina mora izkazati izvajanje najmanj treh okoljskih vzgojnih aktivnosti, ki so namenjene njenim uporabnikom in zaposlenim (I).

5. Marina nudi lastnikom plovil individualne modre zastave (I).

## **B. Okoljski menedžment**

6. Marina ima sprejeto okoljsko politiko in načrt. Okoljski načrt mora obravnavati odnos do vode, do odpadnih voda in porabe energije, naloge s področja zdravstva in varnosti ter usmeritve za uporabo okolju prijaznih sredstev, ko so ta na razpolago (I).
7. Na razpolago morajo biti ustrezni, vidno označeni in ločeni zabojniki za zbiranje nevarnih odpadkov (barv, razredčil, ostrgane barve, čistilnih sredstev, baterij, odpadnih olj, ...). Z odpadki mora ravnati pooblaščen organizacija in jih odlagati na javnem odlagališču (z licenco) (I).
8. Marina mora razpolagati z ustreznimi in redno vzdrževanimi smetnjaki in/ali zabojniki. Z odpadki mora ravnati pooblaščen organizacija in jih odlagati na javnem odlagališču (z licenco) (I).
9. Marina razpolaga z zabojniki za odlaganje odpadnih materialov, namenjenih recikliranju, kot so steklo, prazne konzerve, papir, plastika, organski materiali ipd. (I).
10. V marini je na razpolago oprema za zbiranje kalužnih voda (G).
11. Marina zagotavlja opremo za praznjenje stranišč s plovil (G).
12. Vse zgradbe in oprema v marini morajo biti ustrezno vzdrževane, kot to zahtevajo nacionalni predpisi. Marina je ustrezno umeščena v bližnje naravno in kulturno okolje (I).
13. Marina razpolaga z ustreznimi, čistimi in dobro označenimi sanitarijami, ki vključujejo tudi kopalniške naprave, ter zagotavlja pitno vodo. Odplake odvajajo v javni kanalizacijski sistem (I).
14. Če marina izvaja popraviljanje in pranje plovil, mora zagotoviti, da se odpadki in odpadna voda ne iztekajo v javno kanalizacijo, na območje kopnega ali vodnega dela marine ali v bližnje naravno okolje (I).
15. Marina promovira trajnostne zvrsti prevoza (G).
16. V marini ni dovoljeno parkiranje/vožnja, razen na za to določenih predelih (I).

## **C. Varnost in storitve**

17. V marini je zagotovljena ustrezna in dobro označena reševalna in protipožarna oprema ter oprema prve pomoči. Oprema mora biti dovoljena s strani pristojnih nacionalnih oblasti (I).
18. Marina mora razpolagati z varnostnim načrtom za primere onesnaženja, požara ali za primere drugih nesreč. V marini morajo biti izobešena ustrezna navodila (I).
19. Na privezih morajo biti na razpolago priključki za elektriko in vodo. Instalacije

morajo ustrezati nacionalnim predpisom (I).

20. V marini so naprave, prilagojene potrebam invalidnih oseb (G).

21. V marini je izobešen plan marine, ki označuje razmestitev vseh ustreznih naprav (I).

#### **D. Kakovost vode**

22. V akvatoriju marine je voda vizualno čista (na površini ni oljnih madežev, niti smeti ali sledi odpadnih voda ter drugih vidnih posledic onesnaženja (I).

### **2.2 Modra zastava za naravna kopališča**

Za pridobitev modre zastave se lahko poteguje vsako naravno kopališče, z vsaj enim vzorčnim mestom za analizo kakovosti vode, skupaj z občino, v katerem se le-to nahaja in ki izpolnjuje kriterije programa Modra zastava. Naravno kopališče je vsako kopališče, ki leži na obrežju jezera ali morja. Zaradi tehničnih problemov vzorčenja vode na tekočih vodah se rečna kopališča zaenkrat ne morejo potegovati za pridobitev znaka modra zastava.

Spodaj je prikazanih 27 osnovnih kriterijev, razdeljeno v štiri podskupine, za pridobitev modre zastave za naravna kopališča. »I« pomeni obvezno izpolnjevanje kriterija, »G« priporočeno izpolnjevanje kriterija, ki ga mora naravno kopališče izpolnjevati, če želi pridobiti znak modra zastava.

#### **A. Kakovost kopalne vode**

1. Kakovost kopalne vode mora ustrezati zahtevam in standardom, predpisanim v Direktivi EU za kopalno vodo (I)
  - ni večjih onesnaženj zaradi kanalizacijskih iztokov ali drugih vzrokov (I),
  - ni vidnih onesnaženj zaradi ogljikovodikov (I).
2. Industrijske odplake ne smejo onesnaževati kopališča (I).
3. Za ukrepanje v primerih onesnaženj je sprejet občinski in/ali regionalni načrt (I).
4. Na kopališču se ne smejo nabirati ali gniti ostanki alg ali druge vegetacije; to je dovoljeno le na za to posebej določenem območju kopališča (G).
5. Občina mora izpolnjevati zahteve o ravnanju z odpadki in kakovosti izpustov, komunalnih odpadnih vod, skladno z opredelitvami Direktive EU (I).

## **B. Okoljska vzgoja in obveščanje**

6. V primeru večjega onesnaženja ali druge nevarnosti, ki preti kopališču ali delu kopališča, je zagotovljen sistem takojšnjega obveščanja javnosti. Postopek takega obveščanja mora biti v skladu z določili načrta za primere nevarnosti, kot ga zahteva kriterij 3 (I).
7. Informacije o naravnih občutljivih območjih in obalnih/obrežnih prostorih, vključujoč njihovo favno in floro, morajo biti javno predstavljene in vključene v turistične informacije. Te morajo vsebovati tudi nasvete o obnašanju obiskovalcev teh območij (I).
8. Upravljallec kopališča je dolžan (I):
  - a. objavljati aktualne informacije o kakovosti kopalne vode s pomočjo enostavno razumljivih tabel in števil,
  - b. v neposredni bližini droga z modro zastavo predstavljati informacije o modri zastavi ter navesti odgovornega na lokalni in nacionalni ravni,
  - c. sneti modro zastavo v primeru, ko kopališče ne izpolnjuje več katerega izmed obveznih kriterijev (I).
9. Občina in upravljallec kopališča morata skupaj zagotavljati in prikazati izvajanje najmanj pet okoljsko vzgojnih aktivnosti (I).
10. Predpisi, ki obravnavajo uporabo kopališč, morajo biti javnosti na njeno željo lahko dostopni, na primer v turistični informativni pisarni, na sedežu občine ali na kopališču. Na kopališču mora biti izobešen kopališki red (I).
11. V občini je organiziran Center za okoljsko vzgojo ali podoben stalni javni prostor, namenjen okoljski vzgoji, ki je vsebinsko vezan na okolje obale/obrežja. Tak prostor se v primeru, ko je v njem možno dobivati informacije o programu Modra zastava in predstavlja osrednje mesto vsem dostopnih vzgojnih aktivnosti o vodnem okolju, lahko imenuje Center modre zastave (G).

## **C. Upravljanje z okoljem**

12. Občina mora imeti sprejet prostorski in razvojni plan za svoje obalno/obrežno območje. Plan in tekoče aktivnosti občine, ki se nanašajo na ta prostor, morajo ustrezati planskim predpisom in predpisom, ki urejajo zaščito obalnega/obrežnega območja. V primeru zelo majhne občine je lahko njeno območje zajeto v širšem regionalnem planu (I).
13. Na kopališču mora biti na razpolago dovolj ustrezno zaščitene ter redno vzdrževane smetnjakov. Odstranjevanju odpadkov, alg in drugih polutantov, ki se nabirajo na kopališču, mora biti zagotovljena ustrezna skrb. Odvoz smeti s kopališča je dovoljen le z ustreznimi sredstvi (I).
14. Po potrebi mora biti v času kopalne sezone zagotovljeno vsakodnevno čiščenje

kopališča (I).

15. Na kopališču je prepovedano (I):

- vožnja, razen v posebej določenih primerih,
- tekmovanja s kolesi ali z avtomobili,
- odmetavanje odpadkov,
- kampiranje.

Kopališča, na katera je dovoljen pristop z avtomobili, morajo imeti označene parkirne prostore ter območja, kamor vozilom ni dovoljen pristop; avtomobilom ni nikoli dovoljen pristop na obalni/obrežni rob (I).

16. Dostop do obale kopališča mora biti varen (I).

17. Zaradi preprečevanja morebitnih sporov in nezgod med različnimi uporabniki mora biti kopališče ustrezno organizirano. Če s kopališčem mejijo zaščitena naravna območja, morajo biti sprejeti ustrezni ukrepi zaradi preprečitve negativnih vplivov uporabe kopališča ter njegovih voda na tako območje (I).

#### **D. Varnost in storitve**

18. Na kopališču morajo biti na razpolago ustrezne in čiste sanitarije, priključene na kanalizacijsko omrežje, ki ustreza zahtevam Direktive EU o odpadnih vodah (I).

19. V kopalni sezoni morajo biti zaposleni kopališki reševalci in/ali na razpolago mora biti ustrezna reševalna oprema, ki vključuje tudi opremo za reševanje plavalcev ter navodila za njeno uporabo; zagotovljen mora biti hiter dostop do telefona. Ustreznost reševalne opreme mora odobriti pristojna državna ustanova. Pri opremi morajo biti navodila za njeno uporabo ter vedno in hitro dostopna. Oprema mora biti redno pregledovana in ugotovljena brezhibnost njenega delovanja. Kopališki reševalci morajo biti strokovno usposobljeni in imeti ustrezno dovoljenje, kot to določajo predpisi državnih organov ali strokovnih združenj (I).

20. Na kopališču mora biti na razpolago in enostavno dosegljiva oprema za prvo pomoč (I).

21. Na kopališču morajo biti strogo spoštovani nacionalni predpisi, ki se nanašajo na pse in druge domače živali. Njihova prisotnost in vedenje mora biti na kopališču vedno nadzorovana (I).

22. Na kopališču je brezhiben vir pitne vode (G).

23. Zagotovljen mora biti enostaven in hiter dostop do telefona (G). V primeru, da kopališče ne razpolaga s kopališkimi reševalci, je ta kriterij obvezen (I).

24. Najmanj eno kopališče v občini mora imeti invalidnim osebam ustrezno prilagojen dostop v (kopalno) vodo in njim namenjene sanitarije; razlog za odstopanje od teh zahtev so lahko le neustrezni topografski pogoji. V občini s samo enim kopališčem z modro zastavo mora navedene zahteve izpolnjevati to kopališče (tudi v tem primeru velja zapisana izjema) (I).



25. Vse kopališke zgradbe in oprema morajo biti ustrezno vzdrževane (I).
26. Kopališče ima zabojnike za odlaganje odpadnih materialov, ki jih lahko reciklirajo (na primer steklo, pločevinaste konzerve, papir ipd.) (G).
27. Občina na območju kopališč spodbuja uporabo sonaravnih oblik prometa in gibanja, kot so kolesarjenje, javni prevoz in pešačenje (G).

## 2.3 Zahteve pri analizi kakovosti vode

Pomembna osnova za pridobitev vsakoletne modre zastave je izpolnjevanje pogojev o kakovosti kopalne vode v naravnih kopališčih. Prepisane mikrobiološke ter fizikalno-kemijske podatke analizira v času kopalne sezone pooblaščen javna ustanova. Za pridobitev modre zastave v tekočem letu je potrebno pri razpisni vlogi priložiti rezultate analize vode iz preteklega leta. Na slovenski morski obali nadzoruje kakovost kopalne vode že tretje desetletje Zavod za zdravstveno varstvo Koper na 29 plažah. V tabelah si podrobneje pogledjmo zahteve pri analizi vode, ki jih morajo izpolnjevati dobitniki modre zastave za naravna kopališča.

**Tabela 1: Zahtevane vrednosti mikrobioloških parametrov**

| <i><b>Parameter</b></i>               | <i><b>Priporočene vrednosti</b></i> | <i><b>Sprejemljiv % vzorčnih rezultatov, večjih od priporočene vrednosti</b></i> | <i><b>Zahtevane vrednosti</b></i> | <i><b>Sprejemljiv % vzorčnih rezultatov, večjih od zahtevane vrednosti</b></i> |
|---------------------------------------|-------------------------------------|--|-----------------------------------|--|
| skupne koliformne bakterije           | 500/100 ml                          | 20 %   | 10.000/100 ml                     | 5 %  |
| koliformne bakterije fekalnega izvora | 100/100 ml                          | 20 %   | 2.000/100 ml                      | 5 %  |
| streptokoki fekalnega izvora          | 100/100 ml                          | 10 %   | -                                 | -  |

*Vir: International Blue Flag Co-ordination, 2003*

**Tabela 2: Zahtevane vrednosti fizikalno-kemijskih elementov**

| <b>Parameter</b>   | <b>Zahtevane vrednosti</b>   | <b>Sprejemljiv % vzorčnih rezultatov, ki je večji od zahtevanih vrednosti</b> | <b>Priporočene vrednosti</b>   | <b>Sprejemljiv % vzorčnih rezultatov, ki je večji od priporočene vrednosti</b> |
|--|--|---|--|--|
| <b>vrednost pH</b>   | 6 to 9   | 5 %   |  |  |
| <b>barva in prozornost vode</b>  | Brez posebnosti  | 5 %   |  |  |
| <b>vsebnost mineralnih olj</b>   | <p><i>Voda:</i><br/>Oljne prevleke ni videti, prav tako ni zaznati vonja</p> <p><i>Tla:</i><br/>Plaža mora biti nadzirana v primerih takega onesnaženja, za ukrepanje v primerih polucije je sprejet občinski in/ali regionalni načrt.</p> | 5 %   |  |  |
| <b>Površinsko aktivne substance</b>  | Na površini ni trajne pene   | 5 %   | < 0.3 mg/L   | 10 %   |
| <b>Fenoli</b>  | Ni specifičnega vonja  | 5 %   |  |  |
| <b>Transparentnost</b>   | Vidljivost v globino >1m ali zaradi geografskih posebnosti <1m.  | 5 %   | Globinska vidljivost > 2m  | 10 %   |
| <b>Umazani ostanki in plavajoči materiali kot so les, plastični izdelki, steklenice, zabojniki za steklenice, plastika, guma ali katerakoli druga substanca.</b> | Odsotnost trdih odpadkov tako v vodi kot na tleh.  | 5 %   | Odsotnost katerekoli od naštetih materialov tako v vodi kot na tleh. | 10 %   |

Vir: International Blue Flag Co-ordination, 2003

### **3. Opis trenutno delujočega sistema Morske biološke postaje Piran za zajem, shrambo in prenos podatkov**

Avtomatizirani informacijski sistem za zajem in obdelavo podatkov za potrebe modre zastave potrebuje programsko in strojno opremo za zajem in obdelavo podatkov. V primeru sistema modra zastava je to oceanografska boja in računalniški sistem na kopnem. Za razumevanje razvoja informacijskega sistema v naslednjih poglavjih sem opisal bistvene elemente sistema, ki omogoča zajem, shrambo in prenos podatkov.

Med projekti Morske biološke postaje v Piranu (MBP), ki je sestavni del Nacionalnega instituta za biologijo (NIB), se nahaja tudi oceanografska boja, ki zbira množico podatkov z različnih merilnih instrumentov, nameščenih na boji. Podatki se preko radijske povezave prenašajo na računalniški sistem na kopnem, kjer se ustrezno obdelajo, shranijo in tudi prikažejo na spletnih straneh na naslovu URL: <http://buoy.mbss.org>. Vrednost opreme na oceanografski boji Piran znaša približno 20 MIO SIT, na kopnem pa več kot 2,5 MIO SIT.

Zaradi lažjega razumevanja predstavitve sem sistem predstavil v sklopu štirih podskupin, in sicer:

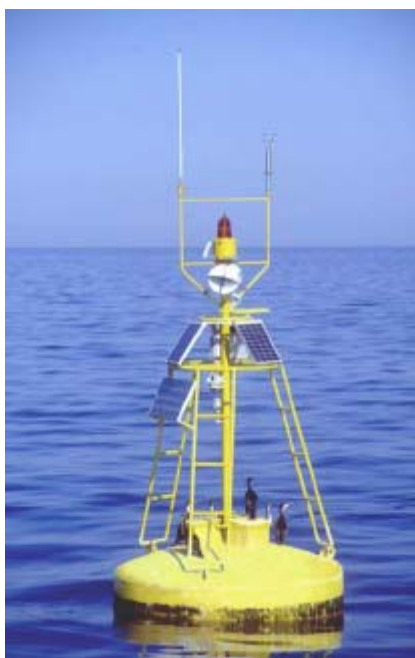
- boja z merilnimi instrumenti in drugo elektroniko,
- postaja za sprejem oceanografskih podatkov,
- programska oprema in
- sistemski podatki.

#### **3.1 Oceanografska boja**

Oceanografska boja, ki jo uporablja Morska biološka postaja Piran, je sestavljena iz naslednjih sestavnih elementov:

- plovnega objekta,
- merilnih instrumentov in
- kontrolne enote.

**Slika 1: Oceanografska boja**



**Vir: Tihomir Makovec, 2001**

### **3.1.1 Oceanografska boja kot plovni objekt**

Bojo je kot plovni objekt izdelalo podjetje Sirio, d.o.o., iz Kopra. Osnovana je na njihovem standardnem modelu boje LIGHTBUOY, premera 2,5 m.

Potrebna plovnost je zagotovljena s trupom boje, ki ima premer 2,5 m ter višino 1,55 m. Trup je narejen iz 8 mm debele ladijske jeklene pločevine. V njem se nahajajo akumulatorske baterije skupne kapacitete 390 Ah. Na trup je pritrjena nadgradnja boje, ki je narejena iz INOX AISI 304 materiala, višine 2,3 m. Nadgradnja je nosilec za regulacijsko elektroniko, bleskač, sončne celice, ultrazvočni merilec hitrosti vetra, elektronski kompas, namenski računalnik in za komunikacijsko opremo.

Paneli sončnih celic imajo površino, ki ustreza električni moči 125 W, in so z regulatorjem električnega toka KTE 04 priključeni na akumulatorske baterije. Regulator skrbi za omejevanje električnega toka. To je pomembno v dolгих sončnih dneh, saj bi neprekinjeno polnjenje v takih pogojih preobremenilo akumulatorje, obstaja pa tudi nevarnost eksplozije. Boja je opremljena s senzorjem za zaznavanje svetlosti. Ta senzor ob zmanjšani svetlosti poskrbi za vklop bleskača SIRIO 2000, ki je nameščen v svetilu premera 0,17 m. Fokusna višina svetila znaša 4 m, domet pa 5 NM.

Geografske koordinate oceanografske boje so 45°32'90" N in 13°33'00" E. Za postavitev je MBP pridobila vsa potrebna dovoljenja od Ministrstva za promet in zveze. Slovenija je pozicijo boje sporočila Hrvaškemu hidrografskemu inštitutu dne 16. januarja 2001, ki je informacijo objavil v svojih oglasih za pomorščake, hkrati pa je pozicijo z obalno postajo Navtex sporočil pomorščakom. Na omenjeno lokacijo so bojo namestili s pomočjo diferencialnega globalnega pozicijskega sistema (v nadaljevanju DGPS), za katerega so korekcijski signal pridobili od podjetja Harpha, d.o.o., iz Kopra s pomočjo GSM sprejemnika. Globina morja na tej poziciji je med 22 in 24 m. Boja je sidrana na tri betonske bloke. Vsak blok ima maso 1800 kg. Ob napetih verigah bi bil kot med verigo in navpično osjo boje enak 45°. Veriga pa mora biti rahlo ohlapna zaradi plimovanja gladine.

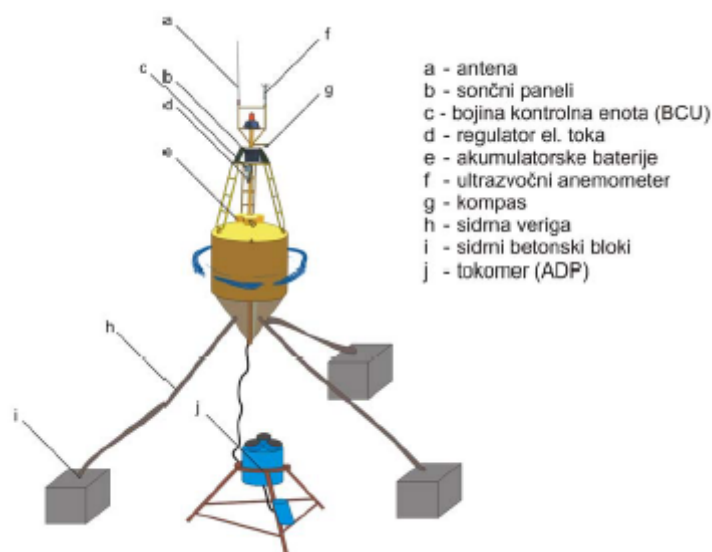
### **3.1.2 Merilni instrumenti na oceanografski boji**

Boja Morske biološke postaje Piran je trenutno opremljena z instrumenti za zajemanje naslednjih podatkov (po instrumentih):

- temperaturo in relativno vlažnost zraka,
- temperaturo in slanost morske vode 2 m pod morsko gladino,
- temperaturo, slanost in tlak morske vode v globini približno 23 m,
- hitrost in smer vetra,
- smer in hitrost morskih tokov v 20 različnih globinah,
- trenutno lego morske boje (smer in oba bočna naklona),
- točen čas in zemljepisno lego (GPS).

Sestavne dele boje z merilnimi instrumenti prikazuje slika 2.

**Slika 2: Boja z merilnimi instrumenti, zasidrana na treh betonskih blokih**



**Vir: Tihomir Makovec, 2001**

Instrumenti merijo fizikalne parametre, kot je:

#### **a. Hitrost in smer vetra:**

Hitrost, vertikalno komponento vetra in smer vetra je merjena z ultrazvočnim anemometrom WindObserver podjetja Gill Instruments. Instrument potrebuje za normalno delovanje enosmerno napajalno napetost v območju med 9 V in 30 V. Instrument posreduje izmerjene podatke po vmesniku RS422 (19200 baud, 8N1 - Način kodiranja podatkov – 19200 – 19.200 poslanih bitov v sekundi; 8 – podatek sestavljen iz osmih bitov; N – preverjanje paritete [none parity]; 1 – znak za konec podatka [one stop bit].). Podatki so kodirani po protokolu NMEA v obliki \$IIMWV,DDD,R,SSS.SS,U,A\*hh, kjer pomeni:

|                 |  |
|-----------------|--|
| <b>\$IIMWV:</b> | oznaka podatkov,   |
| <b>DDD:</b>     | smer vetra v stopinjah,                                      |
| <b>R:</b>       | ASCII znak R, ki pomeni relativno merjenje vetra,            |
| <b>SSS.SS:</b>  | hitrost vetra,   |
| <b>U:</b>       | enote (M = m/s, N = vozli, P = mph, K = kph),                |
| <b>A:</b>       | pravilnostna kontrola (A = pravilen, V = nepravilen),        |
| <b>*:</b>       | zaključni znak za kontrolo števila zlogov,                   |
| <b>hh:</b>      | število, ki predstavlja seštevek števila zlogov med \$ in *. |

Na izhodu instrumenta dobimo štiri vrednosti v eni sekundi. Meritev trenutne hitrosti traja 25 ms. Vsaka izhodna vrednost je povprečje devetih izmerjenih trenutnih

vrednosti.

**b. Nagib in kot zasuka boje ter temperatura zraka:**

Te količine meri elektronski kompas KVH-C100 / TCM2. Za normalno delovanje potrebuje enosmerno napajalno napetost v območju med 8 V in 18 V. Podatke posreduje po vmesniku RS232 (9600 baud, 8N1). Temperatura je merjena v samem ohišju kompasa. Kompas lahko posreduje podatke po protokolu NMEA, vendar v tem primeru ne daje vseh izmerjenih vrednosti, zato je nastavljen za podajanje podatkov v njegovem lastnem protokolu. Instrument vzorčuje neprestano s frekvenco 4 Hz, meri pa azimut, vzdolžni naklon boje (vzdolž smeri sever - jug, angl. pitch), prečni naklon boje (angl. roll) in temperaturo zraka v kompasu.

**c. Morski tokovi, temperatura in tlak pri morskem dnu:**

Morski tokovi se merijo z akustičnim Doplerjevim merilcem tokov Current profiler, NORTEK 3 axis Acoustic Doppler Profiler (v nadaljevanju ADP). Ta instrument potrebuje za svoje delovanje enosmerno napajalno napetost v območju med 12 V in 24 V. Podatke posreduje po vmesniku RS422 (9600 baud, 8N1). Instrument povprečuje podatke v izbranem časovnem intervalu. Po končanem vzorčevanju jih posreduje v binarnem formatu. Začetek merjenja oz. vzorčevanja z električnim pulzom sproži bojina kontrolna enota (angl. Buoy Control Unit, v nadaljevanju BCU). ADP vsebuje tudi senzorja za temperaturo in tlak morja.

**d. Nivo napetosti baterij:**

Napetost akumulatorskih baterij spremljamo z A/D pretvornikom v BCU

### **3.1.3 Kontrolna enota na boji**

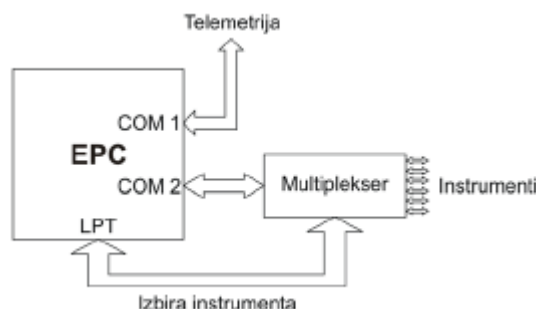
Kontrolna enota temelji na mikroprocesorju 68HC11 podjetja Motorola. Elektroniko komunikacijskega multiplekserja je razvilo slovensko podjetje Ditel, d.o.o., iz Kopra. Poleg zbiranja podatkov iz instrumentov in pošiljanja le teh preko radijskega modema na postajo na kopnem komunikacijski multiplekser tudi krmili napajanje instrumentov z električno energijo. Programska oprema v komunikacijskem multiplekserju je sestavljena iz sistemkega dela ali BIOS-a (angl. Basic Input/Output System), ki skrbi za ustrezno začetno nastavitve elektrone na boji in nudi uporabniškemu programu programski vmesnik za dostop do zmožnosti (resursov) na boji. BIOS tako nadzira: uro realnega časa, UART (angl. Universal Asynchronous Receiver Transmitter) komunikacijska vezja, CompactFlash pomnilniško kartico, vezje za vklop/izklop napajanja posameznih instrumentov in merilno elektroniko (A/D pretvornik v 68HC11), ki spremlja pogoje delovanja, tako da meri temperaturo in

vlago v ohišju ter trenutno napetost baterije in skupno porabo toka vseh sistemov na boji. Uporabniški del programske opreme predstavlja program za zajem, hranjenje in prenos zbranih meritev po radijski zvezi na kopno.

Delovanje podsistema za zajem podatkov na boji opravlja naslednje funkcije:

- V določenih intervalih (začetna nastavitvev je vsake pol ure, in sicer dvajset minut po polni uri in deset minut do polne ure) vključi instrumente na boji in prične z zajemom meritev. Začetna nastavitvev za dolžino zajema podatkov je deset minut.
- Meritve shrani na CompactFlash kartico.
- Po končanem zajemu podatkov procesor vklopi radijski modem in čaka na vzpostavitev zveze. Ko je zveza vzpostavljena se podatki prenesejo v računalniški sistem na kopnem (v Morsko biološko postajo Piran).

**Slika 3: Blok shema komunikacije kontrolne enote: povezava podatkovnega zajemalnika (angl. Embedded Processing Computer, v nadaljevanju EPC) s perifernimi napravami**



**Vir: Deželjin, 2001, str. 7**

Komunikacijska vata COM1 so priključena na radijski modem SATELLINE-3AS. Slednji ima naslednje tehnične lastnosti:

- frekvenčno območje: 380 – 470 MHz, razdeljeno na kanale po 12,5 kHz ali 25 kHz (torej 160 oz. 80 kanalov), uporabljenih je 80 kanalov, širine 25 kHz;
- hitrost prenosa: 9.600 bit/sek. (angl. pbs, v nadaljevanju pbs) pri 160-ih kanalih in 19.200 bps pri 80-ih kanalih;
- napajanje: od 9 do 30 V enosmerne napetosti.

Radijski modem je namenjen izmenjavi podatkov med Morsko biološko postajo Piran in bojo.

Dovoljenje za uporabo radijske frekvence 451,4 MHz je Morska biološka postaja Piran pridobila od Ministrstva za promet in zveze. Vse porabnike električnega toka napajajo akumulatorske baterije oceanografske boje.



### **3.2 Postaja za sprejem podatkov**

Na Morski biološki postaji v Piranu na kopnem se nahajata antena in radijski modem, ki sta enaka tistim na boji. Modem je priključen na komunikacijska vrata COM1 namenskega računalnika (angl. Data Receiving Unit, v nadaljevanju DRU). Na računalniku teče operacijski sistem Linux<sup>1</sup>, in sicer različica Debian 3.0. Računalnik služi tudi kot podatkovni strežnik in poganja MySQL<sup>2</sup> podatkovno bazo. MySQL je relacijski sistem za upravljanje baz podatkov. Del kratice SQL pri MySQL pomeni Structured Query Language - najbolj pogost standardiziran jezik, ki se uporablja za dostopanje do baze podatkov. MySQL je klient/strežnik sistem za upravljanje baz podatkov. Zanesljivost zajema podatkov je zagotovljena z neprekinjenim napajanjem UPS, zrcalnimi diski RAID, angl. Redundant Array of Independent Disks, ter z dodatnim računalnikom, ki služi v primeru okvare prvega in kot numerični računalnik, ki deluje v programskem okolju Windows. Računalnika sta medsebojno povezana preko 10 Mb UTP lokalnega omrežja Ethernet. Strežnik vsebuje tudi požarni zid (angl. firewall), ki ščiti podatke in lokalno omrežje pred nezaželenimi vdori z interneta.

Podsistem na Morski biološki postaji Piran ima za nalogo, da:

- stalno preizkuša (1 ping vsako sekundo), ali je radijski modem na boji vključen,
- ob vzpostavitvi zveze prenese vse še neprenesene meritve in jih zapiše v datoteko,
- po končanem prenosu meritev sproži vnos podatkov v MySQL bazo podatkov in
- ob vzpostavitvi zveze omogoči tudi terminalski priklop na instrumente na boji preko posebnega programa Okna.

### **3.3 Programska oprema**

Dobavitelj sistema za zajemanje podatkov je razvil tudi programsko opremo zanj.

Uporabniško programsko opremo sistema za zajem oceanografskih podatkov na kopenski sprejemni postaji sestavljajo trije programi:

- Mbp\_boja,
- Vpis in
- Okna,

pri čemer se prva dva samodejno zažene ob vklopu računalnika in delujeta neprenehoma kot dva procesa v ozadju. Program Mbp\_boja skrbi za celotno komunikacijo s podsistemom na boji in tako seveda tudi za sprejem podatkov. Naloga programa Vpis je, kot že ime namiguje, obdelava in vpis sprejetih podatkov v

MySQL2 podatkovno bazo surovih meritev. Program Okna pa je uporabniški program in služi kot uporabniški vmesnik za nastavitve parametrov delovanja boje in kot vmesnik za terminalski dostop do instrumentov na boji. Vsi trije programi izmenjujejo podatke med sabo preko skupnega pomnilnika, sinhronizacija pa poteka s pomočjo semaforjev in sporočil (signalov).

Vse tri programe bom podrobneje opisal v poglavju 6.4, saj so programi z manjšimi nadgradnjami primerni tudi za informacijski sistem za potrebe programa Modra zastava.

### **3.3.1 Programi za numerično obdelavo in prikaz podatkov**

Programi za obdelavo in prikaz podatkov so napisani s pomočjo knjižnice programov Matlab in jih izvajamo v okolju Windows. V strežniku (Linux) se vsake pol ure sproži sistemski program »matlab.sh«, ki preveri zadnji sklop datotek s podatki in si v dnevnik zapiše njihova imena ter dolžino. Ko naleti na sklop podatkov, ki še ni vpisan v dnevnik, tedaj ta sporoči podpornemu osebemu računalniku, ki te podatke obdela. To stori tako, da se v računalniškem omrežju poveže na nastavljena vrata internetnega protokola TCP/IP (angl. Transmission Control Protocol / Internet Protocol, v nadaljevanju TCP/IP) podpornega osebnega računalnika, v nadaljevanju numerični PC. Za numerični PC so v programskem okolju Delphi razvili sprožilni program (v nadaljevanju sprožilec). Sprožilec posluša na nastavljenih TCP-vratih v računalniku, v katerem program izvajamo. Ko se na ta vrata poveže oddaljen računalnik, ta najprej prebere internetni naslov oddaljenega računalnika, takoj za tem pa prekine povezavo. Če ta internetni naslov ustreza nastavljeni številki (torej številki strežnika), zažene ukazni program »start.bat« s trdega diska. Pognani ukazni program najprej poskrbi, da z omrežnega diska kopira podatkovne datoteke na lokalni disk, nato pa zažene še program Matlab, ki obdela podatke in nariše slike podatkov. Te slike shrani na omrežni disk. Ker je knjižnica programov Matlab interpretacijsko okolje (ukaze sproti prevaja in izvaja), je spominsko ter časovno potraten. Zato so začeli razvijati alternativno programsko opremo, ki je v nadaljevanju na kratko opisana.

Trenutni razvoj programske opreme kaže na bolj smotrno rabo programskega okolja Octane (Deželjin, 2001) za numerično obdelavo podatkov ter uporabo programov v programskem jeziku Java (angl. applet) za grafično prikazovanje na spletnih straneh. Tudi Octane temelji na interpretacijskem programskem jeziku, za razliko od drugih pa porabi bistveno manj procesorske moči, saj mu ni treba risati slik, poleg tega pa ga bodo izvajali v varčnejšem operacijskem sistemu Linux. Pripravljajo tudi novo metabazo podatkov, kjer bodo vpisovali obdelane oziroma prepračunane surove

podatke. Ukvarjajo se tudi s prognoziranjem plimovanja morja, za kar uporabljajo program prof. Franca Stravisija z Univerze v Trstu.

Pri nas uporabljamo dve uri – zimsko in poletno. Pri meteorologiji in oceanografiji pa se uporablja sončna oz. zimska ura. Tu velja omeniti, da je preklap med zimsko in poletno uro prisoten le v Evropi. Ta podatek je pomemben pri prikazu podatkov in ga je zato potrebno upoštevati.

### **3.3.2 Strežniški programi**

Strežnik Morske biološke postaje Piran funkcioniра v okolju Linux. Ukazni programi poganjajo programe, ki so dostopni pod pogoji GPL-licence. Za tako rešitev so se odločili zaradi stabilnosti in varčnosti operacijskega sistema. Drugi razlog, ki je botroval njihovi odločitvi, pa je v enostavnem odpravljanju napak v programih, ki so del odprte kode. Nenazadnje takšna programska zasnova omogoča prilagoditev programske opreme zastavljenemu projektu. Za programe odprte kode praviloma velja, da so pogosto dobro podprti z dopisnimi seznamami. Tudi komercialna ponudba pomoči (angl. support) je vedno bolj razširjena.

V delovni skupini dr. Malačiča so se odločili za distribucijo RedHat operacijskega sistema Linux zaradi kakovostne komercialne podpore. Poleg tega je večina namenskih programov, ki so narejeni za okolje Linux, testiranih in podprtih ravno za to distribucijo.

Za streženje podatkov v spletu pa uporabljajo programsko opremo Apache 1.3.x. Kot vmesnik za komunikacijo z SQL relacijsko bazo podatkov (v nadaljevanju SQL) so mu dodali PHP-modul z MySQL-podporo. Tako lahko prikazujejo podatke iz SQL-a na domačih spletnih straneh.

Omrežni disk na MBP upravljajo s programom Samba. To je programski paket, ki v okolju UNIX omogoča skupno rabo datotek, združljivo s tisto v okolju Windows. Poleg skupne rabe pa Samba omogoča tudi centralno omejevanje dostopnih pravic do map v skupni rabi (angl. exportov). Strežnik uporabljajo tudi za druge servisne dejavnosti, kot je elektronska pošta, sistem imenskega prostora itn. V prihodnosti bodo oceanografski postaji dodali tudi lasten strežnik.

Za datotečni sistem na mrežnem disku uporabljajo program Journalled file system. Ključna prednost teh datotečnih sistemov pred drugimi (npr. Ext2fs – angl. second extended filesystem) je v evidentiranju sprememb datotek v računalniškem dnevniku.

Ob morebitni okvari datotečnega sistema lahko tako ponovno vzpostavimo hierarhijo podatkov. Ti datotečni sistemi nudijo tudi razširjene lastnosti (angl. extended attributes) datotek. Te uporabljajo predvsem pri dodatnem omejevanju dostopa do datotek.

Razvoj programske opreme je potrebno dolgoročno načrtovati. Pri tem je potrebno upoštevati organizacijo izvirne kode, zato na MBP uporabljajo sistem CVS (angl. Concurrent Version System). CVS je sistem za nadzor revizije kode, ki omogoča pooblaščenemu uporabniku dostop do poljubne prejšnje verzije izbrane datoteke, ki je shranjena v CVS.

### **3.4 Sistemski podatki**

#### **3.4.1 Zgradba datoteke z meritvami in statusne datoteke**

Zgradba datoteke z meritvami, ki jo zapiše program Mbp\_boja ob sprejemu podatkov iz boje, lahko vidimo v spodnji tabeli.

**Tabela 3: Zgradba datoteke z meritvami**

| Odmik<br>v zlogih | Dolžina<br>v zlogih | Opis  |
|-------------------|---------------------|---|
| 0                 | 4                   | ASCII niz   |
| 4                 | 4                   | UTC čas zajema podatkov kot ga vrne UNIX funkcija time, vendar čas 0 ne predstavlja 1.1.1970 0:00 ampak 1.1.2000 0:00 |
| 8                 | N                   | Podatki o meritvah  |
| N+8               | 4                   | ASCII niz »STOP«  |

**Vir: Šonc, 2003, str. 25**

Podatki o meritvah so v obliki:

Prvi zlog je znak instrumenta, ki je:

- »V« za vetromer
- »1« za Idronaut 1
- »2« za Idronaut 2
- »W« za vetromer
- »K« za kompas
- »G« za GPS

- »P« za meritve procesorske plošče
- 0xA5 (hex koda) za ADP

Če je prišlo pri meritvi na posameznem instrumentu do napake, se znaku instrumenta prišteje še vrednost 128 (postavljen bit7), razen pri vetromeru, kjer je koda napake navedena posebej, in pri ADP, kjer posebne kode napake ni, ker so vsi podatki zapisani v podatkovni strukturi, ki je enaka strukturi iz navodil za ADP.

Nato se nahajajo podatki v eni od struktur, ki so podane v nadaljevanju, pri čemer podatek »tip\_velja«, »tip« ali »SyncChar« ustreza znaku instrumenta. Razen pri ADP-ju so vsi podatki zapisani v fiksni vejici kot celoštevilčne spremenljivke. Število mest za decimalno piko je implicitno (programsko) določeno za vsak podatek posebej. Podatki pri ADP so zapisani v obliki, kot je navedena v navodilih za ta instrument.

Zgradbo statusne datoteke »status.dat« pa si lahko ogledamo v prilogi 1.

### **3.4.3 Opis komunikacijskega protokola**

Protokol za prenos podatkov med bojo in Morsko biološko postajo Piran je enostavni ARQ (angl. Automatic repeat request) protokol, ki za vsak poslan paket pričakuje potrditev o prejemu paketa. Sistem na boji deluje kot strežnik zahtev, ki jih pošilja računalnik v Morski biološki postaji. V primeru uspešno prejetega paketa strežnik vrne paket s potrditvijo (angl. Acknowledgement), ACK (ASCII koda 25) v ukaznem zlogu. Če je strežnik sprejel paket, kjer CRC koda ne ustreza vsebini paketa ali paket z nepoznanim ukazom vrne paket z NACK (ASCII koda 26) v ukaznem zlogu, v prvem zlogu podatkov pa kodo napake. Odjemalec (računalnik na Morski biološki postaji) ima vgrajeno še dodatno kontrolo s časovnikom, ki sproži ponovno pošiljanje zadnjega paketa v primeru, če od strežnika v nekem maskimalnem vnaprej določenem času ni prejel odgovora.

Zgradbo paketov komunikacijskega protokola za prenos podatkov med bojo in MBP podaja tabela 4 na strani 22.

**Tabela 4: Zgradba paketov komunikacijskega protokola med MBP in bojo**

| Odmik<br>v zlogih | Dolžina<br>v zlogih | Opis  |
|-------------------|---------------------|---|
| 0                 | 1                   | Znak STX – ASCII koda 02  |
| 1                 | 1                   | Dolžina podatkovnega dela v zlogih (N)  |
| 2                 | 1                   | Ukazni zlog   |
| 3                 | N                   | Podatki (od 0 do 255 zlogov)  |
| 4+N               | 2                   | 16 bitna CRC koda paketa od odmika 1 do 4+N (dolžina, ukazni zlog in podatki) |
| 6+N               | 1                   | Znak ETX – ASCII koda 03  |

**Vir: Šonc, 2003, str. 29**

### 3.4.4 Zgradba konfiguracijske datoteke

Datoteka »buoy.cfg« je klasična besedilna datoteka, kjer lahko v vsaki vrstici zapišemo določeno nastavitveno možnost (opcijo). Poseben pomen ima znak »#«, ki označuje komentarje in vse besedilo od tega znaka pa do konca vrstice programi Mbp\_boja, Vnos in Okna zanemarijo. Datoteka je skupna vsem trem programom, posamezna vrstica z nastavitvami pa ima naslednjo zgradbo:

```
IME_NASTAVITVE = vrednost
```

Vrednost je lahko ali ASCII niz, ki mora biti omejen z navednicami. Primer:

```
SERIAL_DEVICE = "/dev/ttyS0"
```

Lahko je število ali zaporedje števil ločeno z vejicami:

```
SERIAL_SPEED = 19200
TERMINAL_INPUT_FILTERS = 0, 2, 1, 0, 2, 0, 3, 2, 4, 2, 5, 1, 6, 2
```

Logični vrednosti pravilno (true) in napačno (false) sta predstavljeni s številoma 1 in 0:

```
DELETE_DATAFILE = 0
```

### **3.5 Predlog in načrt operativne nadgradnje obstoječega sistema za potrebe Morske biološke postaje Piran (MBP)**

Morska biološka postaja Piran se vedno pogosteje vključuje v mednarodne projekte na področju oceanografije. Vsi našteti projekti silijo v nadgradnjo obstoječega informacijskega sistema te morske biološke postaje v smeri povezljivosti z večjimi sistemi. Trenutno potekajo izmenjave podatkov samo na podlagi dvostranskih pogovorov med inštitucijami, kar povzroča nepotrebno podvajanje podatkov. Komunikacija med posameznimi inštitucijami in MBP poteka na podlagi XML formata ali na podlagi .txt formata. Glede izmenljivosti podatkov je sistem piranske morske biološke postaje relativno zelo odprt. Nekatere smernice so bile seveda že začrtane, gre dejansko samo za priporočila. Skupaj z drugimi inštitucijami bo potrebno zato določiti skupne protokole prenašanja podatkov. Sodelovanja v takih projektih dajo vpogled v konkretne zahteve posameznih sistemov in iskanje skupnih točk med letimi. Za te potrebe bo potrebno razviti dodatno programsko orodje, ki bo v dogovorjenih časovnih frekvencah pošiljalo dogovorjene podatke v dogovorjeni obliki po dogovorjenih standardih prenosa podatkov v skupno bazo podatkov vseh morskoboloških postaj. V nadaljevanju predstavljam nekatere projekte, v katerih aktivno sodeluje tudi Morska biološka postaja Piran.

Nedavno se je zaključil projekt Phare CBC **Sistem za čezmejno izmenjavo ekoloških in oceanografskih podatkov**. Ta sistem podatkov, ki so pridobljeni z meritvami na stacionarnih strukturah (bojah, platformah) v Jadranskem morju, je namenjen tako širši javnosti kot strokovnim inštitucijam, ki sodelujejo pri izmenjavi podatkov.

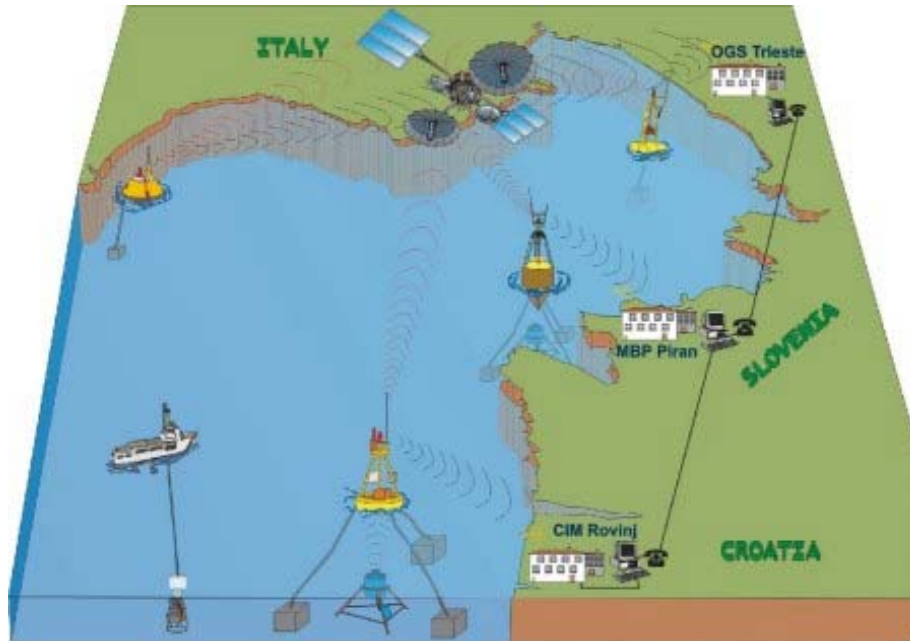
Dejavnosti projekta Sistem za čezmejno izmenjavo ekoloških in oceanografskih podatkov so bile:

- Razvoj programske opreme za izmenjavo podatkov v skoraj realnem času po mednarodnih standardih Medvladne oceanografske komisije Organizacije združenih narodov za izobraževanje, znanost in kulturo - Unesco-IOC (angl. United Nations Educational, Scientific and Cultural Organizations - Intergovernmental Oceanographic Commission), z institucijami v Italiji, ki podatke lahko ponudijo in sprejmejo.
- Instalacija nove merilne opreme za merjenje oceanografskih in ekoloških parametrov na obstoječo obalno oceanografsko bojo pred Piranskim rtom. Izvedene so meritve parametrov blizu morske gladine, kjer so časovne in

prostorske variacije parametrov običajno največje.

- Testiranje delovanja merilne in programske opreme ter zagon sistema za čezmejno izmenjavo podatkov, ki v grafični obliki predstavlja merske podatke tudi širši javnosti preko spletnih strani.

**Slika 4: Sistem za čezmejno izmenjavo ekoloških in oceanografskih podatkov**



**Vir: Tihomir Makovec, 2001**

Morska biološka postaja Piran sodeluje tudi pri izvajanju EU projekta **MAMA** (angl. Mediterranean network to Assess and upgrade Monitoring and forecasting Activities), ki je namenjen infrastrukturni pomoči institucijam za izvajanje operativne oceanografije in izgradnji avtonomnega sistema za oceanografske meritve v Sredozemlju s pomočjo boj. V projektu sodeluje MedGOOS (angl. Mediterranean Global Ocean Observing System), UNESCO/IOC in UNEP-MAP. Projekt koordinira italijanski IMC (International Marine Centre) s pomočjo MedGOOS sekretarijata na Malti. Osnova za izdelavo projekta je bila deklaracija (Memorandum of Understanding) o sodelovanju pri promociji operativne oceanografije v Sredozemskem morju.

Še en razvojni projekt, ki poteka na Morski biološki postaji, je **Adriacosm**, katerega namen je vzpostaviti povezan priobalni sistem merjenja kakovosti voda in cirkulacije tokov v skoraj realnem času (NRT – angl. Near Real Time) v Jadranskem morju in ustjih rek. Uporaba tega sistema je predvidena predvsem v menedžmentu odpadnih voda. Projekt vključuje inštitute iz: Italije, Slovenije in Hrvaške. Projekt podpirajo tudi



francoske inštitucije. Financiranje projekta je zagotovljeno s strani italijanskega ministrstva za okolje in prostor.

Poleg povezljivosti z drugimi sistemi bi bilo potrebno izboljšati tudi nekatere druge elemente sistema na Morski biološki postaji Piran, in sicer predvsem:

- zagotoviti sistemsko arhiviranje podatkov,
- zagotoviti izračune neposredno iz tekočih podatkov, saj se trenutno izračunavajo na podlagi tekočih podatkov samo srednje vrednosti,
- na boji vsak merilni instrument dopolniti še z enim in tako preprečiti morebitni izpad zbiranja podatkov na določenem instrumentu zaradi okvare le-tega,
- na spletni strani so določeni diagrami slabo razvidni ali pa jih sploh ni, zato bi bilo potrebno prenoviti spletno stran tako, da bi omogočala prikaz tekočih podatkov in izračunanih podatkov ter dodati nove skripte za jasno prikazovanje vseh podatkov, ki jih meri Morska biološka postaja Piran.

Povezljivost z drugimi sistemi ni odvisna samo od piranske morske biološke postaje temveč od vseh subjektov na področju oceanografije in meteorologije. Naštete pomanjkljivosti na samem sistemu Morske biološke postaje Piran pa bi bilo potrebno odpraviti. Največji problem pri tem je pomanjkanje sredstev in tehničnega osebja.

## **4. Izkušnje drugih držav**

Vsi strokovnjaki, ki se ukvarjajo z raziskovanjem morja, menijo, da obstajajo potrebe po tekočih podatkih o morski vodi, opisanih v poglavju 3.1.2, in nekaterimi drugimi podatki, ki jih merijo druge oceanografske ali morske biološke postaje, in sicer zaradi zagotovitve informacij s (Legrand, et. Al., 2002):

- podatki o morju in fizičnih lastnosti le-tega: tokovi, valovanja, slanost, SST,
- osnovnimi hidrološkimi in biokemičnimi podatki: vsebnosti kisika, pH, motnost, vsebnost klorofila, vsebnost hranil,
- meteorološkimi podatki, kot so: veter, temperatura zraka, zračni pritisk.

Ti podatki skupaj s podatki iz drugih virov (ročno, druga vzorčna mesta ipd.) tvorijo celoto podatkov, ki so potrebni za izboljšanje zanesljivosti in točnost napovedi in proučevanje oceanografskih podatkov.

V državah EU je zakonsko uokvirjeno, z direktivo 2000/60/EC - 200/10/23, da so države dolžne meriti svoje kopalne vode z namenom zagotavljanja informacij vsem uporabnikom kopališč o neoporečnosti oziroma ustreznosti kopalne vode. Prav tako

je podatkov vedno več, zahtevane so vedno bolj podrobne informacije. Zahteva se tudi ažurnost in takojšnja razpoložljivost podatkov. Zaradi tega trajno merjenje podatkov z dolgimi časovnimi vrstami zahteva uporabo avtomatizacije meritev.

Avtomatsko zbiranje podatkov ima tudi nekaj resnih omejitev, ki so posledica ne-vzdrževanja naprav in drugih vplivov (Legrand, et. Al., 2002):

- pojav alg na instrumentih,
- (ne)stabilnost senzorjev, problem (de)kalibracije,
- komunikacijski problemi,
- dostopnost instalacij,
- vandalizem,
- izpostavljenost ekstremnim vremenskim vplivom in pogojem.

Na kratko bom predstavil, poleg že opisanega slovenskega sistema Morske biološke postaje Piran, nekatere že delujoče sisteme avtomatiziranega merjenja voda v Evropi, ki delujejo na principu merjenja podatkov z bojo. Na teh osnovah se lahko boljše in lažje odločimo za najprimernejše programsko-tehnološke rešitve, ki bi bile v uporabi pri realizaciji projekta avtomatiziranega informacijskega sistema za potrebe gibanja Modra zastava. Iz podrobne analize posameznih že delujočih avtomatiziranih informacijskih sistemov merjenja morske vode spoznamo tudi katere meritve opravljajo posamezni sistemi. Ta informacija je pomembna predvsem zato, ker bi v primeru mednarodne uporabe avtomatiziranega informacijskega sistema za potrebe programa Modra zastava lahko določene obstoječe sisteme nadomestili oziroma dopolnili z novimi merilnimi bojami. Tako bi pomembno prispevali k še podrobnejšem spremljanju in analiziranju morja. Seveda pa je potrebno poudariti, da je informacijski sistem za potrebe Modre zastave priobalni sistem merilnih boj, saj je zahtevano vzorčenje v skladu z določili EU pravilnika za vzorčenje kopalnih voda, in kot tak ne more nadomestiti tistih sistemov, ki spremljajo in merijo lastnosti vode in ozračja na odprtem morju.

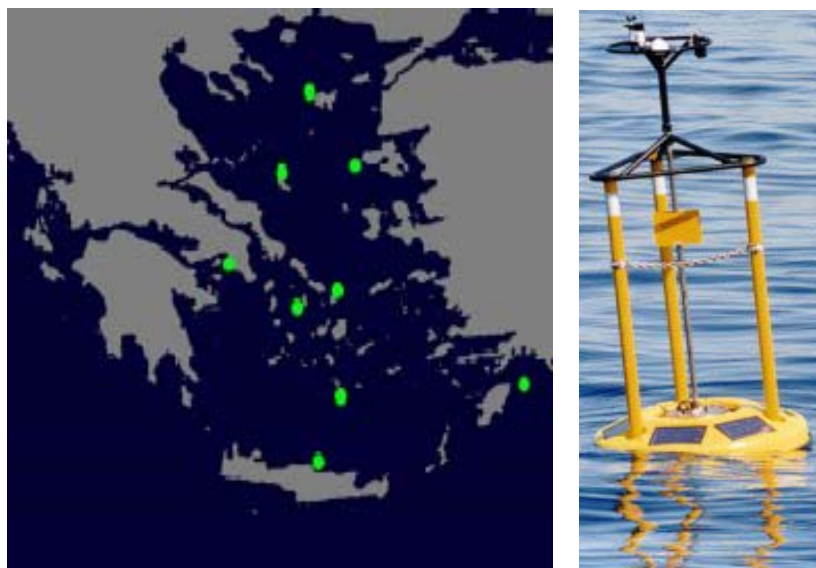
## **4.1 POSEIDON**

Na Egejskem morju ima grška institucija postavljenih devet boj, ki merijo fizikalne in biološko-kemične podatke. Z obalo komunicirajo z GSM moduli in satelitskim sistemom Inmarsat-C. Boje in sistem senzorjev redno vzdržujejo s frekvenco treh mesecev. Podatki so na voljo na internetu v realnem času in jih uporabljajo tudi v 72-urnem napovedovanju meteoroloških in oceanografskih podatkov. V prilogi 2 si lahko tako ogledamo primer napovedovanja temperature na površini morja za 72 ur, v prilogi 3 pa primer prikaza podatkov za eno izmed boj sistema Poseidon.

Napovedujejo:

- atmosfere, razmere,
- višino in smer valovanja na odprtem morju,
- splošno cirkulacijo voda v 3D tehniki,
- karakteristike valovanj priobalnih, plitkih voda,
- premikanje plavajočih onesnaženj.

**Slika 5: Pozicije boj in slika boje sistema Poseidon**



**Vir: National Center for Marine Research - Institute of Oceanography Athens, 23. 1. 2003**

Uporabniki sistema Poseidon le-tega uporabljajo kot vir informacij pri zagotavljanju:

- večje varnosti pomorskega prometa,
- preprečevanja morebitnih naravnih katastrof in onesnaženj,
- razvoja turizma,
- povečanja produktivnosti industrije, povezane z morjem,
- upravljanja in zaščite priobalnega pasu,
- učinkovitega planiranja priobalnih infrastruktur,
- raziskav in razvoja sodobnih tehnologij.

Informacijski sistem Poseidon je sestavljen iz (National Center for Marine Research - Institute of Oceanography Athens, 23.1.2003):

- SENZORJEV na boji za merjenje:
  - zračnega pritiska,
  - temperature vode,

- smeri in moči vetra,
  - višine, smeri in periode valovanja,
  - površinske slanosti in temperature,
  - površinske vsebnosti kisika,
  - svetlobne prepustnosti vode,
  - slanosti in temperature do globine 50 metrov,
  - vsebnosti klorofila A,
  - vsebnosti hranil,
  - radioaktivnosti.
- KOMUNIKACIJSKIH SISTEMOV BOJA-CENTER, ki temeljijo na:
    - satelitski komunikaciji preko satelita Inmarsat-C,
    - radijski UHF (angl. Ultra-high radio frequency) povezavi in
    - GSM povezavi.
  - STROJNE OPREME V CENTRU, sestavljene iz:
    - večprocesorskim računalnikom SGI-ORIGIN 2000, ki zadostuje za matematično-analitično kalkulacijo in predvidevanje razmer,
    - delovno postajo, ki temelji na UNIX in MS WINDOWS platformi in služi za analizo podatkov in prikazovanje le-teh.
  - PROGRAMSKE OPREME V CENTRU, ki jo sestavlja:
    - ORACLE-ova baza podatkov,
    - program za procesiranje in statistično obdelavo podatkov,
    - program za prikazovanje statističnih podatkov in napovedovanje le-teh,
    - Geografski informacijski sistem (GIS) za beleženje operacij Poseidona v realnem času.

## **4.2 MAREL**

Marel je sistem ene ali treh boj, ki merijo:

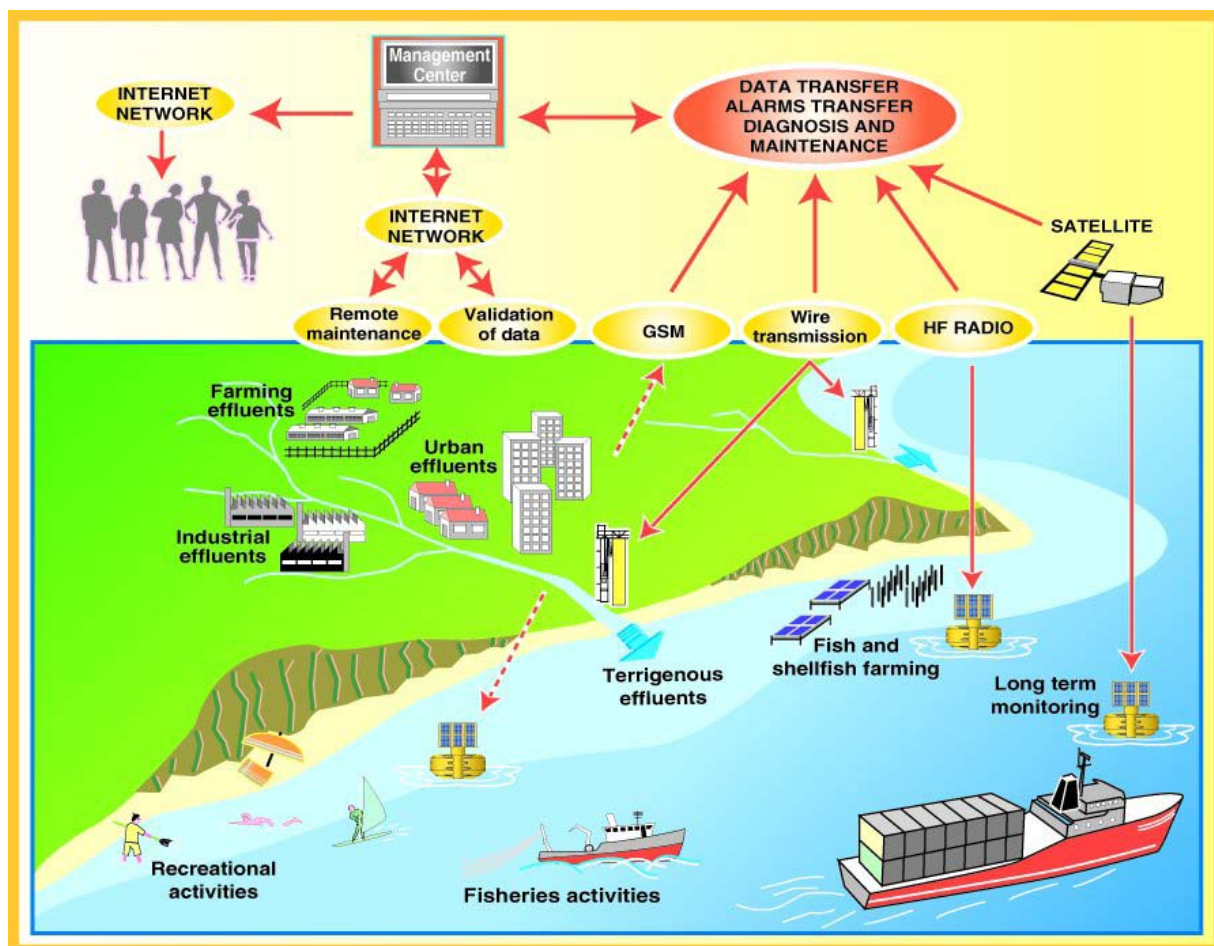
- fizikalne,
- meteorološke in
- biološko-kemične lastnosti voda,

in je operativen že od sredine leta 1999. S centrom na kopnem so povezane z GSM, satelitsko in VHF povezavo, ki poteka po TCP/IP protokolu. Vzdrževanje poteka s trimesečnimi presledki. Podatki so na razpolago na internetu šele po kvalitativnem

pregledu le-teh, kar pomeni, da so objavljeni z enodnevno zamudo.

Medsebojno povezanost vseh zainteresiranih subjektov sistema Marel si lahko nazorno ogledamo spodnji sliki:

**Slika 6: Sistem Marel**



Vir: Ifremer, 20. 1. 2003

Temperatura, prevodnost, raztopljeni kisik, kalnost vode, pH, klorofil, nitrati, veter, zračni pritisk in valovanje so merjeni vsako uro na večih plasteh morske gladine.

Sistem je sestavljen iz štirih komponent:

- »MAREL« boje,
- upravljalnega centra,
- dvosmerne podatkovne povezave in
- meteorološkega laboratorija.

Marel boja je posebne tipa, saj ima vse senzorje v zgornjem delu boje, kar

preprečuje nabiranje alg na senzorjih. Vzorce vode s pomočjo črpalke prenesejo do senzorjev. To omogoča poenostavljeno čiščenje senzorjev. Boja je opremljena s sončnimi celicami, kar omogoča neprestano napajanje naprav v boji. Vsi podatki so poslani z boje s pomočjo GSM modula in TCP/IP protokola v upravljalni center, kjer opravijo obdelave podatkov, kvalitativen pregled le-teh in morebitno alarmiranje in obveščanje vzdrževalcev.

### **4.3 SHARK**

Švedske ustanove imajo postavljen sistem boj v Baltskem morju, ki deluje od leta 2001. Tako kot vsi do sedaj opisani sistemi tudi ta zbira fizikalne, meteorološke in biološko-kemične podatke o lastnosti voda. Raziskujejo tudi pojav cvetenja morja. Komunikacija med bojami in centrom poteka preko Orbocom satelita. Boje imajo tudi lastno spominsko enoto, kjer hranijo podatke. Vzdrževanje boj poteka štiri- do šestkrat letno. Podatki so v realnem času na voljo na Internetu večinoma v preprostem ASCII formatu.

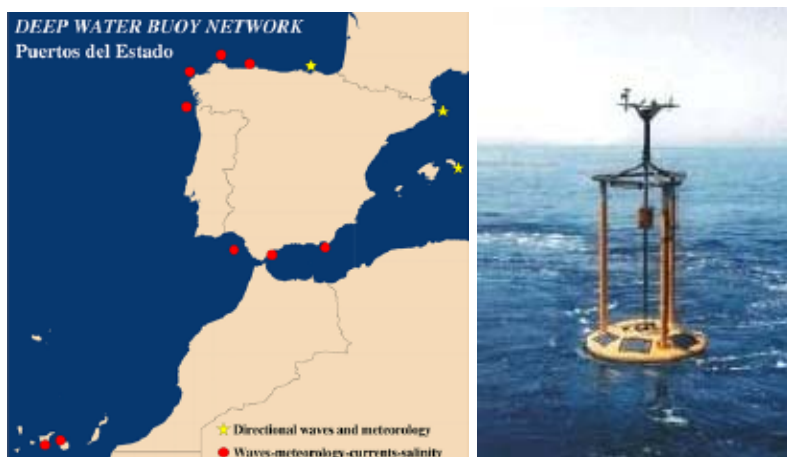
### **4.4 Puertas del Estados**

Španski sistem dvanajstih boj, ki so zasidrane ob njihovi obali, deluje že od leta 2000 in meri:

- fizične,
- hidrološke in
- meteorološke podatke.

Komunikacija poteka preko satelita Inmarsat-C. Vzdrževanje opravljajo štirikrat letno. Podatki so objavljeni na internetu po avtomatski kvalitativni obdelavi.

**Slika 7: Pozicije in slika boje sistema »Puertas del Estados«**



Vir: Puertas del Estados, 5. 2. 2003

## **4.5 CANDHIS**

Združen sistem Candhis ima 22 boj, ki so razporejene okoli francoskih obal ob Atlantskem oceanu, Mediteranskem morju in na Antilih. Sistem deluje že od leta 1972. 7 boj je prosto plavajočih. Merijo višino plimovanja, periodo in smer tokov. Komunikacija med bojami in centrom poteka preko radijske VHF postaje. Vzdrževanje izvajajo enkrat letno. Podatki so na voljo na internetu v realnem času.

**Slika 8: Prostoplavajoča boja**



Vir: Cetmef, 2003

## **4.6 M3A**

Grški M3A je poseben sistem treh boj, dveh podpovršinskih in eno površinsko, ki so blizu ena drugi. Ta sistem je opremljen za merjenje fizičnih, bio-kemičnih, meteoroloških in ADCP parametrov. V prilogi 4 prikazujemo grafični prikaz sistema.

Komunikacija med podvodnima bojama, torej t.i. linijo 1 in linijo 2, poteka preko hidroakustičnega modema ORCA-BASM na tri ure, površinska boja pa komunicira s centrom preko GSM modula in satelitov Argos ali Inmarsat-C.

Vzdrževanje mora potekati vsake tri mesece. Podatki so prikazani na internet strani v skoraj realnem času s triurnimi presledki in uporabljeni za osnovo pri različnih ekoloških modelih.

Osnovni namen je preučevanje tokov in kakovosti vode na odprtem morju v okolici otoka Kreta.

Boja Medusa je opremljena s sončnimi celicami, s katerimi napaja tudi druge boje, in akumulatorji, ki zagotavljajo vsaj 14-dnevno napajanje instrumentov. Zajemalnik podatkov na boji je sestavljen iz 80486 PC sistemom, ki deluje s frekvenco 33 MHz in tehnologijo CMOS za zmanjšanje porabe. Boja vsebuje tudi dva A/D pretvornika s katerimi 16 vhodnih podatkov pretvori v ustrezne digitalne izhodne podatke.

Programski del je napisan v Borlandovem C++ in MS-DOS 6.2 okolju.

Komunikacija med bojami in centrom poteka preko dvosmerne komunikacije preko UHF (433 MHz) sprejemno-oddajne postaje, ki ima doseg 20 nm. Podatke, ki jih zajemajo na dvajset minut, prenesejo v centralo vsake tri ure. Boja hrani podatke tudi na lokalnem trdem disku, velikosti 250 Mb.

Centralna postaja na obali je sestavljena iz računalnika PC, ki deluje v okolju Windows. Predstavitve podatkov so pripravljene s programom Matlab, komunikacije nadzoruje program PC-Anywhere. Primer izpisa si lahko ogledamo v prilogi 5. Strategija zbiranja podatkov sistema M3A pa je predstavljena v prilogi 6.

## **4.7 MedGOOS**

MedGOOS (angl. The Mediterranean Global Ocean Observing System) je nastal na pobudo UNESCO. Pod svojim okriljem združuje več projektov ali iniciativ, kot npr. že opisan projekt MAMA, MFSP (angl. Mediterranean Forecasting System Pilot Project), MEDAR/MEDATLAS (angl. Mediterranean Data Archaeology and Rescue of Temperature, Salinity and Bio-chemical Parameters) in MEDIR (angl. Mediterranean Directory of Marine Scientists and Professionals). Združiti poskuša strokovnjake in sisteme za merjenje morskih voda sredozemskih držav v enovit sistem. V prilogi 7 si lahko ogledamo »on-line« napoved temperature morja sistema MedGOOS za tri dni v naprej.



## **4.8 EGOS**

EGOS je sistem za proučevanje severnega dela Atlantskega oceana, ki zbira meteorološke podatke iz zasidranih in prostoplavajočih boj. Podatke prenašajo preko satelitov GEOS in Meteosat. Vzdrževanje poteka enkrat letno. Podatki so na razpolago v skoraj realnem času, in sicer po kvalitativni obdelavi. V prilogi 8 je predstavljen izpis premikov prosto plavajočih boj v decembru 2002.

## **4.9 SeaNet**

SeaNet je še eden izmed mnogih poskusov povezave različnih sistemov merjenja lastnosti voda v enovit sistem. Ta sistem so v začetku tvorili danski Marnet z dvajsetimi bojami v Severnem morju in Baltskem morju ter angleško-nizozemska mreža CEFAS. Dandanes SeaNet združuje tudi sisteme iz drugih držav iz severnoatlantske regije. Analiza in prikaz podatkov je možen v skoraj realnem času. Prikaz pozicij sistema boj in primer izpisa podatkov je razviden iz priloge 9.

## **4.10 FerryBox**

FerryBox je eden izmed alternativnih sistemov, kjer so merilne naprave nameščene na trajektnih ladjah in tako nimajo problemov z napajanjem in vzdrževanjem boj. V Evropi so si izbrali osem rednih linijskih poti, na katerih merijo podatke, kot so slanost, temperatura, fluorescentnost, prevodnost ipd. Lažje je vzdrževanje, nižji so stroški prenosa podatkov, načeloma ni nevarnosti poškodb ipd. Sam sistem je nazorno predstavljen v prilogi 10.

# **5. METODOLOGIJE NAČRTOVANJA INFORMACIJSKIH SISTEMOV**

## **5.1 INFORMACIJSKI SISTEM**

Količina informacij z dneva v dan raste, kar je razlog, da težko ali nepravočasno najdemo prave podatke. Za lažje izkopavanje potrebnih in želenih informacij je potrebna ustrezna računalniška tehnologija in dobro zasnovan informacijski sistem. Informacijski sistem je sistem, v katerem se ustvarjajo, shranjujejo in pretakajo

informacije. Informacijski sistem pa lahko definiramo tudi kot sistem človeških aktivnosti, ki lahko vsebuje, ali pa tudi ne, uporabo računalniških sistemov. (Avison, Fitzgerald, 1996, str. 13).

Informacijski sistem je osnovni pogoj za delovanje organizacije. Sestavljen je iz formalnega dela, t.i. informacijske infrastrukture, in neformalnega dela. Informacijska infrastruktura nam pove, kako so organizirana in povezana računalniška mreža, strojna oprema, programska oprema in baze podatkov (Turban, 1999, str. 53), neformalni del, pa predstavljajo posamezniki in odnosi med njimi. V njem potekajo procesi, ki omogočajo razrešitev treh temeljnih skupin problemov (Gradišar, 2001, str. 338):

- *Problem premostitve časovne pregrade*

Večina podatkov se ponavadi ne uporabi v istem trenutku, ko so nastali, ampak v nekem časovno odmaknjenem trenutku. Za premostitev razlike v času, podatke shranjujemo na različne materialne nosilce podatkov.

- *Problem transformacije podatkov*

Transformacija je proces, pri katerem pridobivamo iz različnih dejstev uporabne informacije. Problemi nastanejo zaradi izbire in priprave postopkov za generiranje informacij (na metodološkem delu) in zaradi izvedbe postopkov za generiranje ob uporabi različnih tehnik in sredstev (na tehničnem delu).

- *Problem premostitve prostorske pregrade*

Dogodki v nekem sistemu, obdelava podatkov o teh dogodkih in uporaba informacij ponavadi potekajo na različnih prostorsko odmaknjenih lokacijah. Informacijski sistem mora zagotoviti pogoje za primeren pretok podatkov med temi različnimi lokacijami.

## **5.2 RAZVOJ METODOLOGIJ IZGRADNJE INFORMACIJSKIH SISTEMOV**

V začetni fazi uporabe računalnikov v poslovne namene so imeli računalniki glede na današnji čas majhno zmogljivost, zato je bilo težišče raziskovanja v iskanju učinkovitih programskih algoritmov, ki so optimalno izkoriščali razpoložljive zmogljivosti v okviru proizvedenih programskih rešitev (angl. applications) omejenega obsega. Take programske rešitve so bile po vsebini omejene, izdelovali pa so jih programerji, katerih znanja so bila predvsem tehnične narave, manj pa so poznali poslovne probleme, ki jih je bilo treba rešiti. Poleg tega so jim pogosto manjkale tudi komunikacijske sposobnosti. Uporabniki so tako imeli težave pri prenašanju svojih potreb programerjem (Avison, Fitzgerald, 1996, str.18). Posledično so se pojavili problemi:

- Rešitve pogosto niso ustrezno zadovoljevale uporabniških zahtev.
- Četudi je bila rešitev ustrezna, ker je programer zelo dobro poznal problem in strukturo rešitve, je bilo izvajanje sprememb v kolikor toliko obsežni rešitvi dolgotrajno in zamudno zaradi posledic, ki so se lahko pojavile v drugem delu sistema.
- Rešitve so bile pogosto slabo dokumentirane.

Iz navedenih razlogov so podjetja kmalu postala odvisna od posameznikov, ki so podrobno poznali strukturo rešitev, saj so bile le-te za druge programerje nepregledne zaradi pomanjkanja dokumentacije in uporabe različnih tehnik programiranja.

Kot odgovor na navedene probleme se je pojavila nadaljnja delitev dela pri izgradnji informacijskih sistemov:

- analiza poslovnih zahtev – opravi jo sistemski analitik,
- načrtovanje sistema, ki naj izpolni definirane poslovne zahteve,
- programiranje.

Tako se poleg same implementacije pojavita še dve pomembni fazi v razvoju informacijskih sistemov; to sta analiza in načrtovanje (angl. design). Naslednja reakcija na opisane probleme je pojav formaliziranih metodologij za razvoj računalniških aplikacij (Avison, Fitzgerald, 1996, str. 20).

**Metodologija** za razvoj informacijskih sistemov je po definiciji zbirka postopkov, tehnik, orodij in dokumentacijskih pripomočkov, ki bo pomagala razvijalcem sistema v njihovih naporih za implementacijo novega informacijskega sistema. Pod pojmom »metodologija gradnje informacijskih sistemov« si vsaj v praksi največkrat predstavljamo organizacijsko-tehnično znanje, ki ga uporabljamo pri zasnovi in izdelavi računalniških rešitev (Kovačič, Vintar, 1994, str. 29). Pri tem je potrebno upoštevati, da se metodologije spreminjajo v skladu s hitrostjo razvoja in uporabe novih tehnologij, kar pomeni, da metodologija ne more obstajati neodvisno od tehnologije. Metodologija mora zato biti razvita in posodobljena v skladu z najboljšimi izkušnjami v določeni tehnološki veji (Clegg, 1994, str. 1). Kakovostno metodologijo razvoja informacijskega sistema sestavlja niz korakov - faz, za katere so definirana orodja in navodila, kako jih na projektnem delu uporabiti, druga pomembna zadeva pa je porazdelitev odgovornosti in vlog med glavnimi udeleženci procesa (Galliers, 1999, str. 69). Metodologija torej sestoji iz faz, ki se nadalje sestojijo iz podfaz, ki vodijo razvijalce sistema v njihovi izbiri tehnik, ki bi bile lahko primerne v vsaki fazi projekta in jim pomagajo načrtovati, upravljati, nadzirati in ocenjevati projekte izgradnje informacijskih sistemov (Avison, Fitzgerald, 1996, str. 10). Metodologija

tipično vsebuje predpisane korake razvoja, tehnike za pripravo izdelkov, orodja, integrirana z metodologijo, in vzorce, ponovno uporabne dokumente in sezname za preverjanje (angl. Checklists) (Murch, 2001, str. 139).

**Namen uporabe metodologije** pri izgradnji informacijskih sistemov je doseganje različnih ciljev, ki naj jih uporaba metodologije omogoča (Avison, Fitzgerald, 1996, str. 11):

- Natančno beleženje informacijskih zahtev. Metodologija naj pomaga pri opredeljevanju in analizi uporabniških zahtev.
- Podana mora biti sistematična metoda razvoja tako, da lahko napredek učinkovito nadzorujemo. Nadzor je posebnega pomena pri projektih velikega obsega, zato so nujne kontrolne točke in dobro definirane faze, ki omogočajo uporabo tehnik načrtovanja projektov.
- Metodologija naj omogoča razvoj informacijskega sistema v primernem časovnem okviru in ob sprejemljivih stroških.
- Izdelan informacijski sistem naj bo dobro dokumentiran in enostaven za vzdrževanje. Potreba po prihodnjih spremembah sistema je neizbežna zaradi sprememb v organizaciji in okolju. Spremembe naj bodo izvedljive s čim manjšim vplivom na preostali del sistema.
- Zagotovljena mora biti indikacija potrebnih sprememb v čim zgodnejši fazi razvojnega procesa. Strošek spremembe se povečuje, ko izgradnja sistema napreduje od začetne faze analize proti implementaciji sistema.
- Izdelan informacijski sistem naj bo všečen ljudem, na katere vpliva. Tako je zagotovljena uspešnost uporabe takega informacijskega sistema. Kot tipičen primer metodologije, ki naj izpolnjuje navedene zahteve, je v nadaljevanju opisana metodologija tradicionalnega življenjskega cikla razvoja informacijskega sistema.

Pri izbiri metodologije, ki jo bomo uporabili, se lahko poslužujemo naslednjih kriterijev (Lam, 2001):

1. kriterij: ustreznost metodologije za definiranje zahtev in pokrivanje faz razvoja programske opreme;
2. kriterij: koliko vplivajo razvijalčeve izkušnje z metodologijo na uporabnost metodologije;
3. kriterij: kakšno podporo ima metodologija in
4. kriterij: enostavnost metodologije za uporabo in razumevanje.

**Dobra metodologija** razvoja novega informacijskega sistema mora (Warren, 1998, str. B3- 4):

- imeti strukturiran koncept,
- omogočati njenim uporabnikom enostavno uporabo,

- omogočati efektivno vodenje projekta,
- zato je pri izbiri ustrezne metodologije potrebno biti pozoren, da:
- deli življenjski cikel projekta v faze,
  - ima natančno definirane postopke, zaporedje izvajanja postopkov in definirane izhodne rezultate postopkov,
  - omogoča razvid statusa projekta v vsakem trenutku,
  - vsebuje tehnike, ki izboljšujejo kakovost sistema in ima opredeljene postopke potrjevanja ustreznosti,
  - vsebuje postopke planiranja in projektnega vodenja,
  - ima opredeljene vloge in odgovornosti vseh sodelujočih na projektu,
  - določa zahteve za dokumentacijo in njeno standardizacijo,
  - vključuje postopke za zagotavljanje varnosti, nadzora in revizorske sledi v novem informacijskem sistemu in
  - obstaja orodje, ki podpira metodologijo in zvišuje produktivnost programerja.

Metodologija naj ima ločeno opredeljene:

- izhodne rezultate posameznih aktivnosti (torej kaj mora biti narejeno),
- postopke, kako izdelati izhodne rezultate (kako mora biti narejeno),
- standarde izvajanja postopkov in izhodnih rezultatov.

Obstaja več vrst metodologij, ki jih različne skupine uporabljajo glede na svoje potrebe in zahteve:

- metodologije kot gotovi izdelki, ki so podprti z ustreznimi orodji in informacijsko tehnologijo,
- interne metodologije, razvite v posameznih organizacijah in
- kombinacija obeh.

Nobena metodologija, uporabljena pri izgradnji ali prenovi informacijskega sistema, sama po sebi ne zagotavlja uspeha, lahko pa s svojimi priporočili, opisi posameznih korakov, ciljev in tveganj zviša kakovost končnega rezultata ter zmanjša tveganja.

Vse večji in kompleksnejši informacijski sistemi in hitro spreminjanje tehnologije zahtevajo vedno večjo formalizacijo razvojnega procesa informacijskega sistema. Pozornost pri razvoju informacijskih sistemov se pomika v zgodnje razvojne faze, strateško načrtovanje in logično zasnovo informacijskega sistema. Formalizacija zgodnjih razvojnih faz omogoča uporabo različnih računalniško podprtih orodij, tako imenovanih **CASE** (angl. Computer Aided Software Engineering) orodij. CASE orodja pomenijo računalniško podprto gradnjo informacijskih sistemov in predstavljajo uporabo računalniško podprtih sistemov za bolj učinkovit, natančen in celovit proces analize, načrtovanja, razvoja in vzdrževanja informacijskega sistema. CASE orodja

ponavadi vsebujejo module za opis poslovnih procesov, baze podatkov in strukture programa, omogočajo oblikovanje zaslonских slik, poročil in testnih podatkov (Gradišar, Resinovič, 2001, str. 184-185).

Zaradi vse večje zmogljivosti informacijske tehnologije se povečujejo tudi zahteve uporabnikov informacijskega sistema, ki zahtevajo pri svojem delu vse močnejšo informacijsko podporo, več informacij, bolj obdelane in predstavljene informacije, iz vedno večjega števila podatkovnih baz. To pa zahteva spremembo narave informacijskih sistemov. V zadnjem času postajajo informacijski sistemi vse bolj kompleksni, povezani in vseobsegajoči. Pogosto tak informacijski sistem sestoji iz cele množice različnih aplikacij, bodisi po naročilu organizacije ali drugih standardih, ki se dobijo na trgu. Vedno bolj se uveljavljajo celoviti informacijski sistemi, ki povezujejo vse funkcije neke organizacije, omogočajo uresničitev in vpeljavo novih konceptov organizacije dela in proizvodnih procesov v proizvodnjo in poslovanje (Kovačič, Vintar, 1994, str. 51).

Za uspešno **uvajanje celovite rešitve** v večje podjetje je zaradi obsežnosti projekta uvedbe pred odločitvijo o nakupu, oziroma lastnem razvoju, smiselno skrbno analizirati obstoječi sistem, stroške in čas do uvedbe ter se zavedati pozitivnih in negativnih učinkov nakupa že izdelanega sistema v primerjavi z lastnim razvojem (Dahlén, Elfsson, 1999, str. 13-14).

Pri izbiri standardnega informacijskega sistema podjetje pravzaprav izbira dve stvari, in sicer programski paket, ki bo omogočil spremembe in izboljšave poslovnih procesov, ter poslovnega partnerja, ki bo podjetju ponujal podporo in izboljšave sistema v prihodnosti. Zaradi velikega števila ponudnikov celovitih rešitev, je pomembno, da podjetje analizira ponudnike, ter ugotovi lastnosti in funkcionalnosti posameznih sistemov, glede na potrebe podjetja. Nekaj splošnih kriterijev za odločitev o izbiri ustreznega standardnega sistema je podanih v nadaljevanju (Shields, 2001, str. 68-71):

- sistem mora ustrezati kulturi in poslovnim procesom podjetja,
- sistem mora zagotavljati funkcionalnost s poslovnega področja podjetja,
- sistem mora zagotavljati fleksibilnost pri spremembi poslovnega okolja,
- pomembna je povezljivost z ostalimi sistemi v podjetju,
- razpoložljivost podpore ponudnika pri uvajanju,
- sistem mora biti dovršen in stabilen.

V praksi so se uveljavile različne **tehnike modeliranja**, na katerih temelji postopek zasnove in gradnje informacijskega sistema. Ne glede na izbrano metodologijo vsak informacijski sistem slej kot prej preide skozi etape življenjskega cikla, ki zajema faze

od analize, načrtovanja, gradnje do izkoriščanja in vzdrževanja informacijskega sistema (slika 9, str. 41). Od izbora metodologije pa je odvisno, kakšni modeli nastanejo v posamezni fazi in na kakšen način so predstavljeni.

Razvoj metodologij pri gradnji informacijskih sistemov je sledil razvoju informacijske tehnologije, njenih tehničnih in drugih značilnosti, po drugi strani pa tudi vlogi informatike v poslovnih sistemih.

Starejše metodologije so temeljile na tako imenovanem linearnem pristopu. Ta izhaja iz predpostavke, da je gradnja informacijskega sistema zaporedni proces, torej niz aktivnosti, ki si sledijo v vnaprej določenem zaporedju in pripeljejo do končnega cilja – uvedene programske rešitve.

**Slika 9: Etape življenjskega cikla informacijskega sistema**



**Vir: Kovačič, Vintar, 1994, str. 44.**

Kot posledica slabosti linearnega pristopa, je nastal prototipni pristop. Pri le-tem gre za razvojni pristop, ko se v sodelovanju z uporabniki najprej izdelata prototip sistema, nato pa se ga dopolnjuje in razvija, dokler ne izpolnjuje vseh uporabnikovih želja in zahtev. V zadnjih letih se je začel vse bolj uveljavljati objektni pristop gradnje informacijskih sistemov. Glavni cilj objektnega pristopa je razvijanje programske rešitve iz množice sestavnih delov (objektov). Pri tem je vsak sestavni del lahko uporabljen večkrat in v različne namene. Objektni pristop naj bi skrajšal čas razvoja novih rešitev, izboljšal zanesljivost in kakovost rešitev ter poenostavil vzdrževanje informacijskih sistemov (Kovačič, Vintar, 1994, str. 44-48).

V naslednjem poglavju, si podrobneje poglejmo temeljne značilnosti Objektno usmerjenih metod razvoja informacijskega sistema.

### **5.3 OBJEKTNO USMERJENE METODOLOGIJE RAZVOJA INFORMACIJSKIH SISTEMOV**

Objektna tehnologija se tudi v praksi vse bolj uveljavlja kot vodilna paradigma razvoja informacijskih in programskih sistemov. Vzrok leži v temu, da ta tehnologija v primerjavi s svojimi predhodniki zagotavlja boljše in učinkovitejše mehanizme za doseganje temeljnih ciljev informatike, kot so npr. višja kakovost, večja produktivnost, višja stopnja ponovne uporabe (Silič et al., 2000, str. 2).

Metodologije razvoja informacijskih sistemov pred pojavom objektno orientiranih metodologij dosledno ločujejo podatkovni in procesni pogled. Tako se podatki in procesi modelirajo v različnih modelih, v fazi implementacije pa se podatki nahajajo v podatkovni bazi, proces pa se kodira v programske procedure in rešitve, ki manipulirajo s temi podatki. Problematična je sprememba podatkovne strukture, ki obvezno povzroči iskanje programskih modulov, ki uporabljajo spremenjeno strukturo.

#### **5.3.1 Osnovni koncepti objektnega pristopa**

Osnovna ideja objektnega pristopa razvoja informacijskih sistemov je v čim bolj neposredni preslikavi objektov realnega sveta (npr. stranka, artikel) v model informacijskega sistema, ki obravnava dotično poslovno področje. Model je tako poenostavitev realnosti, narejen za boljše razumevanje sistema, ki ga kreiramo (Booch, Roubagh, Jacobson, 1999, str. 463). Model lahko opredelimo tudi kot abstrakcijo, predstavitev dela realnega sveta (Avison, Fitzgerald, 1996, str. 62). Osnovni koncepti takega modela: objekt, ograjevanje, razred, sporočilo, mnogoličnost, dedovanje, bodo predstavljeni v nadaljevanju.

##### **5.3.1.1 Objekt**

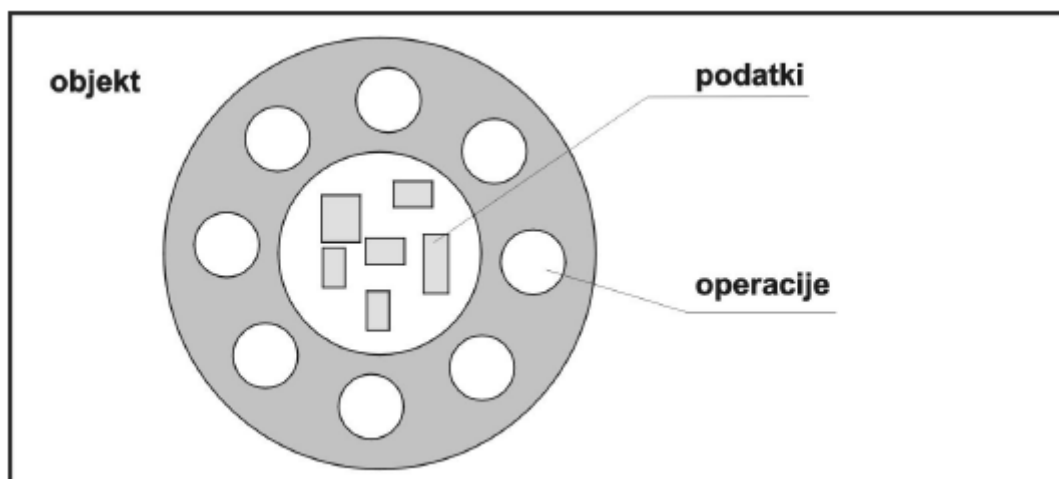
Objekt je entiteta, ki je sposobna shranjevati stanje (informacijo) in ponuja operacije za pregledovanje ali spreminjanje tega stanja (Jacobson et al., 1994, str. 44). 41 Entiteto lahko opredelimo kot stvar ali dogodek, ki ga opazujemo (Grad, Jaklič, 1996, str. 76). Objekt je katerakoli stvar (entiteta) realnega sveta, o kateri hranimo podatke



in obnašanje (Grad, Jaklič, 1996, str. 117). Objekti vsebujejo podatke in načine spreminjanja ter doseganja podatkov. Stanje objekta opisujejo atributi, določeno pa je z vrednostjo atributov. Obnašanja objekta opisujejo operacije, ki predstavljajo način za pregledovanje in spreminjanje vrednosti atributov in s tem stanja objekta. V tem je bistvena novost objektnega pristopa, saj se na enem mestu združujejo podatki in načini za spreminjanje podatkov. Vsak objekt ima enolični identifikator (angl. OID – object identifier), ki ga loči od ostalih objektov, čeprav imajo morda enake vrednosti atributov.

### 5.8.1.2 Ograjevanje

**Slika 10: Objekt vsebuje podatke in operacije**



**Vir: Črv, 2000, str. 9**

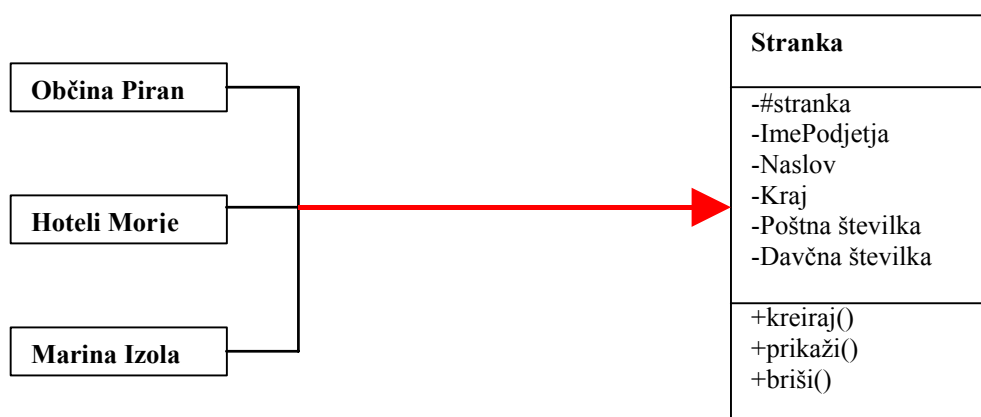
V objektno orientiranem sistemu so podatki shranjeni v notranjosti objekta. Edini način doseganja in spreminjanja podatkov je uporaba operacij, ki so definirane v tem objektu. Objekt vsebuje vse potrebne operacije za manipuliranje s podatki znotraj objekta. Operacije so implementirane z metodami, ki predstavljajo programsko kodo, ki izvaja opredeljene spremembe nad podatki. Objekt se navzven predstavlja le s svojimi operacijami, notranja struktura objekta in način implementacije operacij pa sta zunanjemu svetu (drugim objektom) skrita. Mehanizem skrivanja informacij pred zunanjim svetom (angl. information hiding) z izključnim dostopom do podatkov preko operacij objekta se imenuje ograjevanje (angl. encapsulation). Drugi objekti lahko pridobijo informacijo ali spremenijo stanje objekta le s pomočjo klica operacije zadevnega objekta. Tak način dela ščiti vsebino objekta pred zunanjim vdorom, odpravlja pa tudi potrebo po tem, da vsi objekti poznajo notranjo strukturo drugih

objektov (Taylor, 1992, str.50). Na ta način je omogočena lokalizacija sprememb, saj se morebitne spremembe načina obdelave podatkov implementirajo le na enem mestu.

### 5.3.1.3 Razred

Objekte, ki imajo podobne lastnosti in obnašanje, lahko združujemo v objektne tipe. Objekte, ki imajo poleg enakih operacij tudi enako notranjo strukturo, združujemo v razrede. Razred je ena od (morda mnogih) implementacij tipa (Jacobson et al., 1994, str. 50) . Razred je opis niza objektov, ki imajo enake attribute, operacije, povezave in semantiko (Booch, Jacobson, Rumbaugh, 1999, str. 18). Razred tako predstavlja vzorec za objekte in opisuje njihovo interno strukturo. Objekti istega razreda imajo enako definicijo operacij in informacijske strukture (Jacobson et al., 1994, str. 50). Tako so posamezni objekti primerki (angl. Instance) razreda.

**Slika 11: Primerki razreda**



Vir: avtor

Iz slike je razvidno, da so posamezni objekti – konkretna podjetja – primerki razreda Stranka, ki ima opredeljene attribute, ki ga opisujejo, in operacije. Atribut je imenovana lastnost razreda, ki opisuje razpon vrednosti, ki ga primerki lastnosti lahko vsebujejo. Predstavlja neko lastnost stvari, ki jo modeliramo, ki je skupna vsem objektom razreda ( Booch, Jacobson, Rumbaugh, 1999, str. 50). Operacija je implementacija storitve, ki jo lahko drugi objekti zahtevajo od razreda, z namenom vplivanja na obnašanje ( Booch, Jacobson, Rumbaugh, 1999, str. 51).

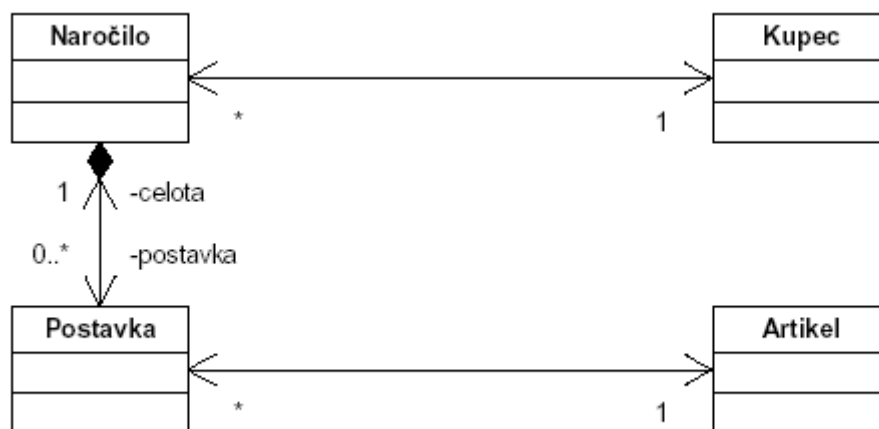
### 5.3.1.4 Asociacije

Asociacije predstavljajo povezave med primerki razredov (določena oseba dela za določeno podjetje). Na konceptualnem nivoju predstavljajo asociacije konceptualne povezave med razredi (Fowler, Scott, 2000, str. 52). Asociacija je strukturna povezava, ki pove, da so objekti ene stvari povezani z objekti druge (Booch, Rumbaugh, Jacobson, 1999, str. 65). Asociacija med seboj lahko povezuje dva ali več razredov. V prvem primeru jo imenujemo binarna asociacija, v drugem, izjemnem primeru pa n-arna asociacija. Grafično jo prikazujemo kot polno črto, ki povezuje razreda med seboj. Osnovne vrste asociacij predstavljajo naslednje strukture:

- »je-del« (angl. is-part-of),
- »je-povezan-z« (angl. is-associated-with),
- »je« (angl. is-a ).

Asociaciji lahko priredimo ime in smer asociacije. Asociacija je lahko eno-ali dvosmerna.

**Slika 12: Primer asociacij med razredi**



**Vir:** Prirejeno po Shelly et al., 2001, str. 5.20

Na primeru lahko vidimo tip asociacije »je povezan z« med kupcem in naročilom ter postavko naročila in artiklom. V obeh primerih je asociacija dvosmerna, kar v implementaciji pomeni, da mora imeti v prvem primeru razred Naročilo metodo, ki ugotovi, kateri kupec je dal naročilo, prav tako pa mora razred Kupec imeti metodo, ki ugotovi, katera naročila pripadajo določenemu kupcu. Opredeljena je tudi števnost (Uveljavljena sopomenka je kardinalnost, angl. cardinality, multiplicity), ki nam pove, da eno naročilo pripada natanko enemu kupcu, en kupec pa ima lahko več naročil.

Primer asociacije med naročilom in postavko naročila je vrste »je del«, ki predstavlja dejstvo, da so postavke sestavni del naročila. Lahko jo imenujemo agregacija. Prav tako je opredeljena števnost in obojestransko usmerjenost asociacije. Konkreten primer je posebna oblika agregacije, t.i. sestavljena agregacija (angl. composite aggregation), ki ima dodatno lastnost, da deli brez celote ne morejo obstajati, in je torej ob brisanju naročila treba izbrisati tudi vse postavke tega naročila. Tip povezave »je« predstavlja generalizacijo, ki bo podrobneje predstavljena v okviru opisa dedovanja.

### *5.3.1.5 Sporočilo*

Objekti komunicirajo med seboj s sporočili. Prenos sporočil je edini način, s katerim objekti sodelujejo med seboj. Objekt s sporočilom zahteva od drugega objekta izvedbo neke operacije. Tako je običajna vsebina sporočila:

- identifikacija objekta, ki naj izvede operacijo – prejemnik sporočila,
- ime operacije, ki naj se izvede,
- parametri operacije, če obstajajo.

Prejemnik izvede operacijo in vrne rezultat objektu, ki je zahtevo posredoval (pošiljatelju). Na ta način se omogoča ograjevanje in nedotakljivost notranjosti objekta s strani drugih objektov.

### *5.3.1.6 Mnogoličnost*

Kadar ima operacija z istim imenom v različnih razredih različen pomen in implementacijo, govorimo o mnogoličnosti ali polimorfizmu. Sporočilo lahko sproži različen odziv glede na razred, v katerem je klicana operacija implementirana. Na način izvedbe torej vpliva razred prejemnika sporočila.

### *5.3.1.7 Dedovanje*

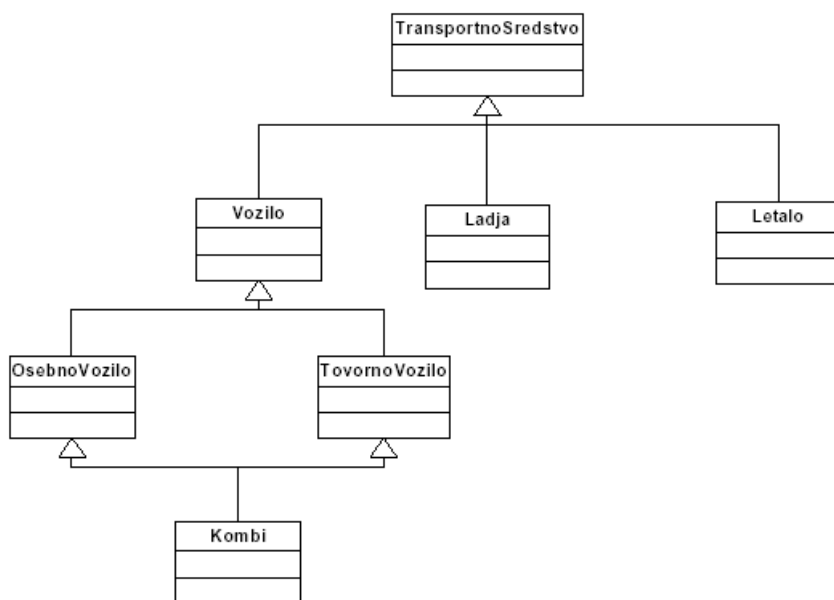
V primeru, ko imamo v našem modelu dva ali več razredov s podobnimi atributi in operacijami, lahko s postopkom generalizacije izločimo skupne attribute in operacije v poseben razred, ki ga z uporabo mehanizma dedovanja povežemo z razredi, ki ohranijo le tiste lastnosti, ki so jim lastne in jih tako izdelani nadrazred ali prednik (angl. superclass ali ancestor) nima. Obratno lahko s postopkom specializacije iz

razreda, ki ima ustrezne lastnosti kreiramo podrazred (angl. subclass);

Sopomenka je potomec (angl. Descendent), ki mu dodamo posebne operacije ali attribute, ki jih potrebujemo. Včasih prav z namenom kreiranja novih podrazredov kreiramo t.i. abstraktni razred, ki ima le funkcijo združevanja lastnosti, ki jih dedujejo njegovi potomci. Tak razred navadno nima primerkov, imajo jih le njegovi potomci. Znotraj specialnega razreda lahko poleg dodajanja atributov in operacij spremenimo tudi pomen in implementacijo določene operacije. Postopku pravimo predefiniranje (angl. overriding) Postopek je zelo učinkovit, vendar nam poslabša preglednost celotnega modela, saj na prvi pogled ni vidna različna funkcionalnost operacije z enakim imenom v dveh, z mehanizmom dedovanja povezanih razredih.

Potomec lahko deduje lastnosti od enega ali več prednikov. V prvem primeru govorimo o enkratnem, v drugem pa o večkratnem dedovanju. Večkratno dedovanje prav tako zmanjšuje preglednost modela in jo je treba posebej previdno uporabljati. Ob nepravilni uporabi lahko pride do ponavljajočega se dedovanja (angl. repeated inheritance), kjer enake operacije obstajajo v obeh prednikih izbranega razreda, ko imata oba prednika skupnega prednika, od katerega sta podedovala te operacije (Jacobson et al., 1994, str. 67).

**Slika 13: Primer dedovanja**



**Vir: Blanc, 2002, str. 36.**

Na sliki je prikazan primer nepravilne uporabe večkratnega dedovanja na primeru razreda »Kombi«, ki naj bi podedoval lastnosti osebnega in tovornega vozila. Lastnosti, ki so podedovane iz razreda »Vozilo«, se bodo dvakrat dedovale v enaki obliki. Situacijo lahko rešimo tako, da na novo definiramo te operacije (predefiniranje), izberemo ene ali druge, ali pa spremenimo strukturo dedovanja npr. tako, da »Kombi« postane direkten potomec razreda »Vozilo«. Glede na potrebno preglednost strukture dedovanja je najboljša zadnja možnost. Klasifikacija je dejanje odstranjevanja različnosti med objekti, da lahko vidimo skupne lastnosti (Odell, 1999, str. 122). Pozornost je treba nameniti razločevanju klasifikacije in generalizacije.

»Janez je človek (klasifikacija), človek je vrsta (generalizacija) in vrsta je življenjska oblika (generalizacija).« Na primeru teh izjav vidimo, da ima besedna zveza »človek je življenjska oblika« ustrezen smisel, zveza »Janez je življenjska oblika« pa ne. Generalizacija je tranzitivna, klasifikacija pa te lastnosti nima (Fowler, Scott, 2000, str. 95). Razločevanje lahko izvedemo, če si namesto povezave »je« poskusimo vprašati »je vrsta« (angl. is kind of).

Dedovanje ob pazljivi uporabi omogoča učinkovito vzdrževanje, saj se spremembe operacij vršijo na enem mestu, na najvišjem mestu v hierarhiji razredov. Tako spremenjeno operacijo podedujejo vsi potomci tega razreda. Dedovanje omogoča obsežno ponovno uporabo, saj se z dedovanjem lastnosti izbranega razreda in dodajanjem potrebnih funkcionalnosti bistveno skrajšuje čas implementacije.

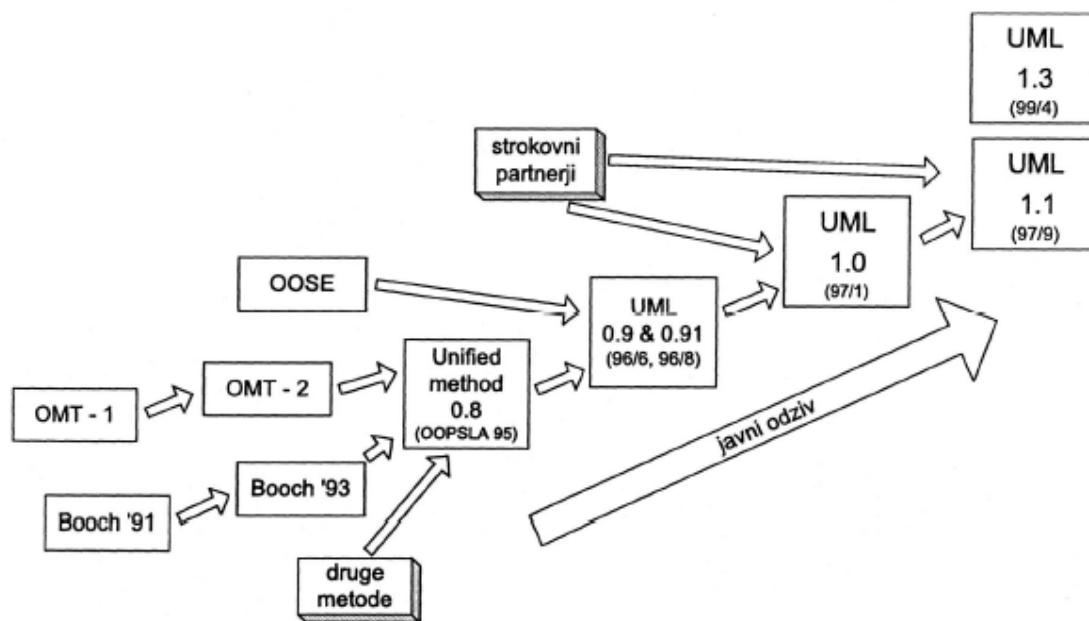
### **5.3.2 POENOTEN JEZIK ZA MODELIRANJE – UML**

Poenoten jezik za modeliranje (angl. unified modelling language, v nadaljevanju UML) je nastal z združitvijo več metodologij za objektno orientirano analizo in načrtovanje, med katerimi so bile najpomembnejše:

- OMT, avtorja Jamesa Roumbaugh,
- OOSE, avtorja Ivarja Jacobsona,
- Boochova metodologija, avtor: Grady Booch.

UML je bil v okviru OMG (Object Management Group) standardiziran in je postal standardni jezik za modeliranje.

Slika 14: Razvoj jezika UML



Vir: Živkovič, Heričko, Rozman, 2000, str.201

UML je jezik za modeliranje in ne metodologija za razvoj informacijskih sistemov. Metodologije običajno poleg jezika vsebujejo tudi proces izgradnje sistema z opredeljenimi fazami in njihovimi izdelki. Avtorji (Roumbaugh, Jacobson, Booch) poleg jezika, ki je metodološko neodvisen, predlagajo še svoj proces razvoja informacijskih sistemov, ki temelji na postopno naraščajočem modelu in se imenuje Poenoten proces razvoja programske opreme (angl. The unified software development process). UML, uporabljen skupaj s tem procesom, lahko označimo za metodologijo.

UML kot jezik za modeliranje omogoča izgradnjo različnih vrst modelov, ki nam omogočajo (Booch, Jacobson, Roumbaugh, 1999, str. 6):

- vizualizacijo trenutnega ali bodočega sistema (angleška izraza: as-is, to-be),
- prikaz strukture in obnašanja sistema,
- izdelavo vzorcev, ki nas vodijo pri konstrukciji sistema,
- dokumentiranje naših odločitev.

### 5.3.2.1 Osnovni gradniki UML

UML sestavljajo osnovni gradniki:

- stvari (angl. Things), imenovali jih bomo elementi,
- povezave (angl. relationships),
- diagrami.

Opis gradnikov je povzet po viru The unified modeling language user guide (Booch, Jacobson, Rumbaugh, 1999, str. 18 – 26 ).

#### **Elementi**

Elementi so lahko namenjeni opisovanju strukture (angl. structural things), obnašanja (angl. behavioral things), združevanja (angl. grouping things) ali opomb (angl. annotational things).

Med strukturne elemente spadajo razred, vmesnik (angl. Interface), sodelovanje (angl. Collaboration), primer uporabe (angl. use case). Razred je opis niza objektov, ki imajo enake attribute, operacije, povezave in semantiko. Razred implementira enega ali več vmesnikov. Grafično je predstavljen kot pravokotnik, ki običajno vsebuje ime, attribute in operacije.

Vmesnik je zbirka operacij, ki opredeljuje storitve, ki jih ponujata razred ali komponenta. Vmesnik opredeljuje specifikacijo in parametre operacije; izvedba operacije je skrita. Grafično je prikazan kot krog.

Sodelovanje opredeljuje interakcijo in je združba vlog in ostalih elementov, ki skupno delujejo v kooperativnem obnašanju. Razred je lahko del več sodelovanj. Sodelovanje je grafično predstavljeno kot črtkasta elipsa.

Primer uporabe je opis niza zaporednih korakov, ki jih izvaja sistem, in določenemu akterju prinašajo rezultat, ki ga lahko opazujemo. Grafično je prikazan kot elipsa. Komponenta je fizični del sistema, ki zagotavlja uresničitev niza vmesnikov. Komponenta tipično predstavlja fizično pakiranje sicer logičnih elementov, kot so razredi, vmesniki in sodelovanja. Grafično je prikazana kot pravokotnik z zankama.

Vozlišče (angl. Node) je fizični element, ki obstaja v času izvajanja in predstavlja računalniški vir, ki ima lasten spomin in včasih tudi procesorsko moč. Grafično je prikazan kot kocka, ki vsebuje lastno ime.



Elementa za opis vedenja sta interakcija in diagram stanj (angl. state machine). Interakcija predstavlja obnašanje, ki vsebuje niz sporočil, izmenjanih med objekti v določenem kontekstu za doseganje določenega namena. Interakcija vsebuje različne elemente, ki vključujejo sporočila, obnašanje, ki ga sporočilo sproži, in povezave med objekti. Grafično predstavljamo sporočilo kot usmerjeno daljico, ki skoraj vedno vsebuje ime operacije, ki jo sproža.

Diagram stanj predstavlja obnašanje, ki opredeljuje zaporedje stanj objekta, ki se menjajo v interakciji, kot odgovor na dogodke. Vključuje stanja, prehode (tok iz stanja v stanje), dogodke (ki sprožajo prehode) in aktivnosti (odgovore na prehode). Grafično stanje prikazujemo kot pravokotnik z zaobljenimi koti, ki vsebuje ime in podstanja, če obstajajo.

Mehanizem za grupiranje objektov v UML se imenuje paket (angl. Package). Paket je splošni mehanizem grupiranja elementov. Obstoja na konceptualnem nivoju, za razliko od komponente, ki obstaja v času izvajanja. Grafično je prikazan kot mapa. Za dodajanje opomb in opisov v UML uporabljamo ustrezne elemente opomb, ki se imenujejo zapiski (angl. Notes). Zapisek lahko pripnemo na element ali skupino elementov. Grafično je prikazan kot zavihan list.

## **Povezave**

Povezave v UML lahko razdelimo na:

- odvisnosti,
- asociacije,
- generalizacije,
- realizacije.

Odvisnost (angl. Dependency) je semantična povezava med dvema elementoma, pri kateri sprememba neodvisnega elementa lahko povzroči spremembo odvisnega elementa. Grafično jo prikažemo kot črtkano puščico.

Asociacija je strukturna povezava, ki opisuje niz zvez (angl. Links), ki povezujejo objekte. Poseben primer asociacije je agregacija, ki je opisana med osnovnimi koncepti na začetku poglavja. Asociacijo modeliramo kot polno črto med dvema strukturnima elementoma. V primeru agregacije je na strani črte, ki je pri elementu, ki predstavlja celoto, votel romb, v primeru sestavljene agregacije (angl. composite aggregation, je posebna oblika agregacije, kjer deli brez celote ne morejo obstajati) pa poln romb. Primer sestavljene agregacije so postavke dokumenta in dokument.

Generalizacija predstavlja povezavo generalizacije in specializacije, ki sta prav tako opisani na začetku poglavja. Grafično se prikaže kot puščica z votlo konico, usmerjeno proti predniku. Uresničitev je semantična povezava med elementoma, kjer prvi opredeljuje funkcijo, ki jo drugi izvede. Srečamo jo med vmesniki in razredi ter komponentami, ki jih uresničujejo, in med primeri uporabe ter sodelovanji, ki jih uresničujejo. Grafično je prikazana kot črtkana puščica z votlo konico.

## Diagrami

UML ponuja različne diagrame, ki nam služijo za prikaz različnih pogledov na obravnavani sistem. Tako predstavimo enoten model na več načinov, kar omogoča ločeno obravnavanje posameznih problemov in olajšuje modeliranje sistema. Na voljo imamo devet različnih diagramov. Diagrami, s katerimi prikazujemo statično strukturo sistema, so:

- razredni diagram,
- objektni diagram,
- diagram komponent,
- diagram porazdelitve (angl. deployment diagram).

Obnašanje sistema modeliramo z naslednjimi vrstami diagramov :

- diagram primerov uporabe
- diagrama interakcij sta:
  - diagram zaporedja (angl. sequence diagram)
  - diagram sodelovanja
- diagram stanj
- diagram aktivnosti

Diagrami vsebujejo osnovne gradnike in jih uporabljamo za prikaz različnih pogledov na enoten model sistema, ki ga obravnavamo.

Podrobneje bomo diagrame obravnavali v poglavju 6.2, kjer bomo s pomočjo jezika UML pristopili k izgradnji informacijskega sistema za potrebe Gibanja Modra zastava.

### 5.3.3 POENOTENI PROCES RAZVOJA PROGRAMSKE OPREME

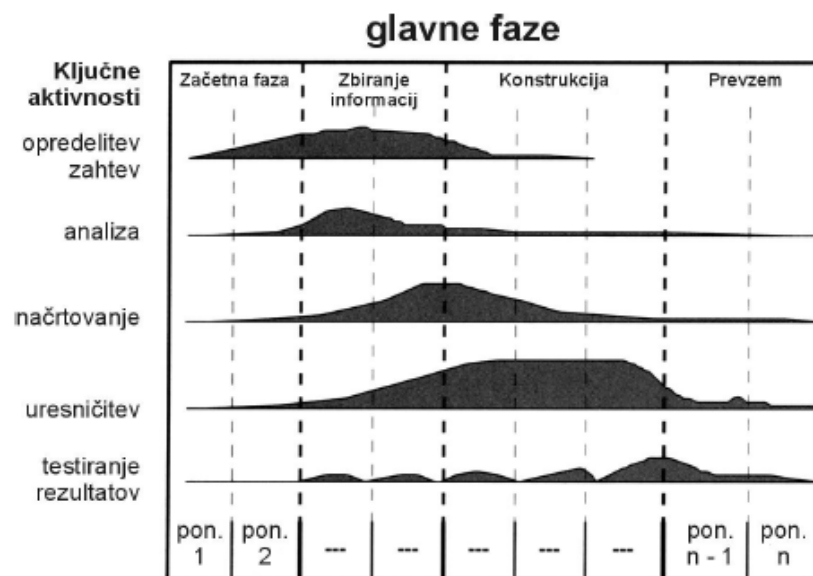
Avtorji jezika UML kot način izgradnje informacijskih sistemov predlagajo poenoten proces razvoja programske opreme (angl. The unified software development process), ki so ga opredelili kot drugi del metodologije, skupaj s poenotenim jezikom za modeliranje.

Proces razvoja programske opreme predstavlja opredelitev niza aktivnosti, ki so potrebne za spreminjanje potreb uporabnikov v konsistentni niz izdelkov, ki predstavljajo programsko rešitev in za spreminjanje dodatnih sprememb teh zahtev v nov, ponovno konsistentni niz izdelkov (Booch, Jacobson, Roumbaugh, 1999, str. 24). Izdelki poleg delujoče programske opreme obsegajo še niz med seboj konsistentnih modelov, ki prikazujejo različne poglede na sistem in so zapisani s pomočjo UML jezika. Proces je predstavljen kot iterativen in inkrementalen. Tako se celoten proces razvoja deli po dveh razsežnostih: časovno po fazah razvoja in po aktivnostih, ki jih opravljamo znotraj vsake ponovitve.

Po časovni razsežnosti so opredeljene faze razvoja, ki zajemajo celoten cikel razvoja informacijskega sistema:

- začetna faza (angl. inception),
- zbiranje informacij (angl. elaboration),
- konstrukcija (angl. construction),
- prevzem (angl. transition).

**Slika 15: Faze in aktivnosti poenotenega procesa**



Vir: Booch, Jacobson, Roumbaugh, 1999, str. 104

### 5.3.3.1 Ključne aktivnosti procesa

Celoten razvoj se razbije na vrsto ponovitev glavnih faz procesa. Rezultat vsake ponovitve je nov prirastek (sopomenka: inkrement) izdelkov, ki se kaže kot dodaten

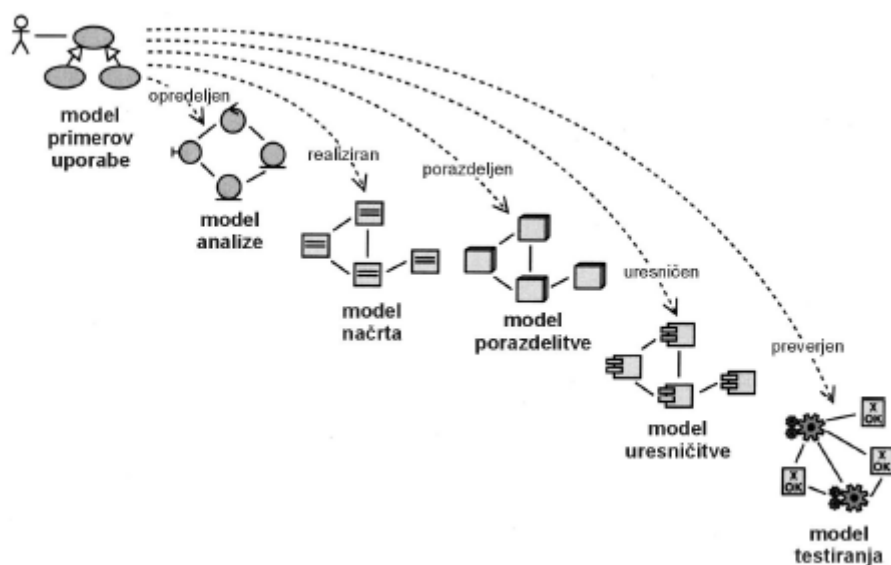
del kode, ki je pripravljena za produkcijo in polno integrirana, in razširjeni modeli, ki opisujejo različne poglede delovanja ter omogočajo enostaven prehod v naslednjo ponovitev, saj je delovanje sistema jasno prikazano. Znotraj vsake faze in ponovitve izvedemo zaporedje aktivnosti, ki predstavlja drugo razsežnost:

- a. opredelitev zahtev,
- b. analiza,
- c. načrtovanje,
- d. uresničitev (sopomenka: implementacija),
- e. testiranje rezultatov.

Obseg porabe virov za vsako od teh aktivnosti je različen za različne glavne faze in je grafično predstavljen kot osenčeno področje pod krivuljami znotraj slike procesa. Za različne aktivnosti potrebujemo ustrezne modele, znotraj katerih prikažemo rezultate vsake aktivnosti.

Tako so opredeljeni naslednji modeli:

**Slika 16: Modeli poenotenega procesa**



**Vir: Booch, Jacobson, Rumbaugh, 1999, str. 10**

Slika poudarja odvisnost med posameznimi modeli. Sprememba enega modela vpliva na druge. Tako se pri opredeljevanju zahtev uporablja model primerov uporabe, pri analizi model analize, za načrtovanje model načrta in model porazdelitve. Model uresnitve ustreza aktivnosti uresničevanja, model testiranja pa

aktivnosti testiranja. Modeli vsebujejo diagrame, opredeljene znotraj poenotnega jezika za modeliranje.

## **Opredelitev zahtev**

Namen opredelitve zahtev je s pomočjo primerov uporabe opredeliti model bodočega sistema.

Zahtevano funkcionalnost opredelimo, ko opišemo načine uporabe sistema in rezultate, ki jih sistem nudi uporabnikom. Pomembno je identificirati vse različne vloge, ki komunicirajo s sistemom, s čimer opredelimo akterje. Nefunkcionalne zahteve, kot je na primer odzivnost sistema so prav tako različne pri različnih primerih uporabe in jih opišemo v njihovem okviru (Booch, Jacobson, Rumbaugh, 1999, str. 131). Primere uporabe povežemo v model s pomočjo diagramov primerov uporabe. Poleg primerov uporabe, ki so glavno orodje pri opredeljevanju zahtev, si pomagamo še z diagrami za opisovanje obnašanja sistema, predvsem z diagrami zaporedja. Poleg tega lahko naredimo tudi nekatere prototipe uporabniških vmesnikov, ki pomagajo k podrobnemu razumevanju načina interakcije uporabnika in sistema. Glede na to, da je model primerov uporabe orientiran proti uporabniku in njegovemu pogledu na delovanje sistema, pri opredeljevanju primerov uporabe pa lahko srečamo izrazoslovje, ki je specifično obravnavani problematiki, si v takem primeru pomagamo tako, da napišemo definicijo uporabljenih izrazov - pojmovnik.

Primerom uporabe opredelimo pomembnost in jih razvrstimo v vrstni red za nadaljnji razvoj. Tako so glavni izdelki opredeljevanja zahtev naslednji:

- podrobno opredeljeni primeri uporabe,
- diagram primerov uporabe,
- ostali diagrami, ki pojasnjujejo posamezne posebnosti,
- seznam nefunkcionalnih zahtev,
- morebitni prototipi uporabniških vmesnikov,
- opredeljena prioriteta primerov uporabe,
- pojmovnik uporabljenih izrazov.

Na koncu opišemo model primerov uporabe kot celoto.

## **Analiza**

Analiza predstavlja pogled na način realizacije zahtevane funkcionalnosti. Treba je podrobno analizirati uporabniške zahteve, opredeljene v predhodnem koraku. Model analize se razlikuje od modela primerov uporabe, saj (Booch, Jacobson,

Roumbaugh, 1999, str. 175) :

- je opisan v formalnem jeziku, namenjenem razvijalcu sistema;
- prikazuje notranji pogled na sistem;
- sestavljen je iz razredov in paketov;
- ne vsebuje ponavljanj (sopomenka: redundanca) in je konsistenten;
- prikazuje način izvedbe funkcionalnosti znotraj sistema, je osnova za načrtovanje;
- opredeljuje način realizacije primerov uporabe s strani sistema.

Tako se v koraku analize opredelijo razredi in njihove odgovornosti, ki jih zagotavljajo operacije, vmesniki ter paketi, ki bodo zagotavljali določene funkcionalnosti. Opredelijo se tudi povezave med razredi ter njihovo združevanje v pakete. Na miselnem nivoju se preizkusi delovanje primerov uporabe skozi tako opredeljeni model. Tako določeni razredi in paketi omogočajo izvajanje primerov uporabe. Opredelimo tudi povezavo med primeri uporabe in elementi modela analize, ki omogočajo njihovo izvajanje. Za opisovanje modela analize uporabljamo:

- razredni diagram,
- diagram sodelovanja.

Odvisnosti med paketi, ki realizirajo posamezne primere uporabe, prikažemo v razrednem diagramu na višjem nivoju abstrakcije, kjer prikažemo pakete in njihove medsebojne odvisnosti. Model analize je prikazan na konceptualnem nivoju, podrobnosti uresničitve niso prikazane.

## **Načrtovanje**

V koraku načrtovanja opredelimo fizični model sistema, ki bo osnova za uresničevanje. Opredelimo model načrta in model porazdelitve. V modelu načrta podrobno opredelimo razrede, povezave med njimi in dinamično obnašanje sistema. Dokončno opredelimo attribute razreda, opredelimo vidljivost (Vidljivost lahko opredelimo kot javno, privatno in omejeno - angl. public, private, protected) in opišemo metode. Metode lahko opišemo v naravnem jeziku ali pseudo kodi ter prepustimo programiranje naslednjemu koraku, lahko pa jih takoj programiramo v izbranem programskem jeziku. Povezave med objekti dokončno opredelimo glede števnosti, smeri navigacije in imenovanja vlog glede na predvideno okolje uresničitve. Prav tako je treba upoštevati predvideno tehnologijo shranjevanja podatkov, ko načrtujemo razrede, ki izvajajo zapisovanje podatkov v okviru transakcije. Pakete preoblikujemo v podsisteme. Treba je zagotoviti čim manjšo odvisnost med podsistemi, opredeliti ustrezne vmesnike in zagotoviti, da podsistem ponuja pravilno realizacijo operacij, ki so opredeljene z njegovimi vmesniki.

Dinamično obnašanje sistema podrobno opišemo z diagrami zaporedja, diagrami prehodov stanj in diagrami aktivnosti. Tako je model načrta podrobna predstavitev sistema z upoštevanimi lastnostmi okolja, v katerem bo uresničen.

Model porazdelitve vsebuje vozlišča, njihove lastnosti in povezave ter začetno razporejanje razredov in podsistemov v vozlišča. Za opisovanje uporabimo diagram porazdelitve.

## **Uresničitev**

V koraku uresnitve izdelamo posamezne programske komponente, ki implementirajo razrede in podsisteme, ki smo jih opredelili v modelu načrta. Razrede implementiramo kot posamezne komponente, ki vsebujejo izvorno kodo. Uresničitve poteka v več ponavljanjih in vzporedno, zato je pomembno načrtovati integracijo po vsaki ponovitvi. Predvidene so ponovitve, v okviru katerih se izdelajo obvladljivi kosi kode, ki se sproti integrirajo z že razvitim delom sistema iz prejšnjih ponovitev. V okviru uresnitve izvedemo testiranje posameznih komponent, preden jih integriramo in namestimo na ustrezno strojno opremo - vozlišča. Integralni in sistemski test izvedemo v koraku testiranja. V modelu uresnitve uporabljamo diagrama komponent in porazdelitve. Glavni rezultat je koda, ki uresničuje izdelke predhodnih korakov in omogoča delovanje sistema.

## **Testiranje rezultatov**

Namen testiranja je preizkus delovanja sistema, tako glede pokrivanja zahtevanih funkcionalnosti kot tudi pravilnosti delovanja in izpolnjevanja nefunkcionalnih zahtev. Testiranje na nivoju komponent je opravljeno v koraku uresnitve, v koraku testiranja izvajamo integracijski in sistemski test. Za urejen način testiranja pripravimo model testiranja, ki vsebuje (Booch, Jacobson, Rumbaugh, 1999, str. 313):

- testne primere, ki opredeljujejo vsebino testiranja,
- testne postopke, ki opredeljujejo način testiranja,
- testne komponente, ki omogočajo avtomatizacijo testnih postopkov.

Treba je narediti načrt testiranja, ki opredeljuje urnik testiranja, in osebe, ki bodo testiranje izvajale. Po izvedenem testiranju ocenimo uspešnost testiranja. V primeru uspešnega testa celotnega sistema je sistem pripravljen za predajo. Model testiranja dopolnjujemo in ga uporabimo ob vsaki novi ponovitvi znotraj življenjske dobe sistema.

### 5.3.3.2 Glavne faze procesa

#### **Začetna faza**

V začetni fazi je treba opredeliti poslovno potrebo in obseg načrtovanega sistema. Opredeliti je potrebno glavne funkcionalnosti bodočega sistema. Opredelimo jih s primeri uporabe. Poiskati poskusimo vse primere uporabe, podrobno pa opredelimo le najvažnejše. V okviru primerov uporabe identificiramo akterje, ki bodo sodelovali s sistemom. Poleg definiranja funkcionalnosti je treba opredeliti potrebne vire in oceniti pričakovano trajanje in stroške izgradnje. Pripravimo začetni načrt projekta. Tako opredeljene stroške primerjamo s pričakovanimi koristmi, ki jih naj sistem prinese, in se odločimo za potrditev ali zavrnitev nadaljnjega razvoja.

#### **Zbiranje informacij**

V tej fazi podrobno opredelimo večino primerov uporabe in jih analiziramo. Pretežne aktivnosti te faze so dokončno opredeljevanje zahtev, analiza in načrtovanje. V okviru načrtovanja načrtujemo sistem kot celoto in upoštevamo posebnosti opreme, ki jo nameravamo uporabiti. Izvedemo tudi prototip do te mere, da lahko prikažemo delovanje najpomembnejših primerov uporabe. Preverimo realnost projektnega načrta, kot smo ga opredelili v začetni fazi, in ga po potrebi popravimo. Sproti je treba oceniti morebitna nova tveganja, ki so se pokazala z boljšim razumevanjem obravnavane problematike. Na koncu faze imamo pripravljene naslednje izdelke:

- opredeljena je večina primerov uporabe,
- modela analize in načrtovanja za večino primerov uporabe,
- model uresničitve in testiranja v obsegu prvega prototipa,
- dopolnjen načrt projekta in oceno tveganja.

Ko smo prepričani, da izdelani modeli napovedujejo uresničitev sistema, ki bo izpolnjeval zahtevane funkcionalnosti, nadaljujemo z naslednjo fazo.

#### **Konstrukcija**

V fazi konstrukcije je pretežni del aktivnosti izveden v okviru načrtovanja, uresničitve in testiranja. Fazo izvedemo v več ponavljanjih, znotraj katerih postopno dograjujemo sistem. V vsaki ponovitvi dodamo funkcionalnost nekaj primerov uporabe, izvedemo testiranje in integracijo. Uporabniki sodelujejo pri testiranju vsake ponovitve in lahko vplivajo še na zadnje korekcije izdelanih funkcionalnosti. Ob izdelani programski opremi pripravimo tudi ustrezno tehnično in uporabniško dokumentacijo. Dopolnimo tudi vse nedokončane modele iz prejšnjih faz. Na koncu faze je potrebno oceniti, ali



je programska oprema primerna za vsakodnevno uporabo in predajo uporabnikom.

## **Prevzem**

V fazi prevzema predamo izdelek v uporabnikovo okolje. Opravimo še zadnja testiranja in popravke. Po zagonu se lahko pokažejo težave, ki jih v fazi testiranja nismo uspeli identificirati. V kolikor so odstopanja večja, lahko odpravo problema posredujemo v naslednjo ponovitev izdelave sistema. Manjše popravke izvedemo kar znotraj faze prevzema. Tako dopolnimo model uresničitve in testiranja ter zabeležimo pridobljene izkušnje za izboljšanje procesa v naslednjih ponovitvah.

Razvijanje programske opreme na opisan način omogoča kontinuirano dopolnjevanje in spreminjanje funkcionalnosti glede na spremenjeno poslovno okolje in uporabniške zahteve. Ključnega pomena je postopnost in iterativnost ter močna povezanost posameznih modelov, ki omogoča enostavno odkrivanje potrebnih sprememb v modelih, ki so bližji implementaciji, glede na spremembe v modelu primerov uporabe.

Metodologija omogoča modeliranje na dokaj različne načine z uporabo različnih kombinacij diagramov. Pri razvoju programske opreme s pomočjo te metodologije je neobhodna uporaba ustreznega CASE orodja, ki sproti preverja konsistentnost posameznih modelov ter sprejetje načelnih usmeritev glede uporabe posameznih diagramov, v različnih fazah izgradnje informacijskega sistema.

## **6. Izgradnja informacijskega sistema, vezanega na program Modra zastava**

K izgradnji informacijskega sistema, ki bo omogočal zbiranje podatkov za potrebe programa Modra zastava pristopimo postopoma, in sicer s pomočjo treh faz:

- pripravimo idejni načrt, v katerem prikažemo idejno zasnovo projekta, pregledamo zahteve in potrebe uporabnikov in analiziramo obstoječe tehnologije, potrebne za izgradnjo tovrstnega informacijskega sistema,
- načrtujemo informacijski sistem s pomočjo poenotenega jezika za modeliranje,
- predlagamo sestav informacijskega sistema, in sicer posebej za bojo z merilnimi napravami, za strežnik in periferno opremo ter izbiro uporabniškega vmesnika, in izberemo sistem za upravljanje z bazo podatkov ter vzpostavimo bazo podatkov in ustrezno sistem integriramo.

V postopku izgradnje informacijskega sistema pa ne smemo pozabiti na ekonomski

vidik postavitve in vzdrževanja takega sistema. Pri izgradnji sistema izhajamo iz že delujočega vzorčenja voda v sistemu Morske biološke postaje Piran, ki smo ga opisali v tretjem poglavju.

## 6.1 Idejni načrt

Opisani problemi, ki se pojavljajo ob samem vzorčenju kopalne vode, vplivajo na smer izgradnje informacijskega sistema, ki bi rezultate vzorčenja prenesel do uporabnikov v trenutku. Na tak način bi uporabnik lahko prišel do ažurnih in verodostojnih podatkov.

Idejni načrt sem si zamislil tako, kot ga prikazuje spodnja slika:

**Slika 17: Idejna struktura informacijskega sistema za potrebe programa Modra zastava**



**Vir: avtor**

Na sliki je prikazan sistem z okvirno 20 manjšimi bojami, ki bi bile locirane v priobalnem pasu, natančneje v akvatorijih naravnih kopališč ali na obrobju le-teh. Kakovost kopalne vode in druge podatke bi preko senzorjev, nameščenih na boji, merili po določilih Direktive EU o kakovosti kopalne vode ter drugih priporočilih. Podatke bi periodično pošiljali preko GSM omrežja na strežnik, ki bi bil predvidoma postavljen na Zavodu za zdravstveno varstvo Koper, ki je zadolžen in pooblaščen za vzorčenje kopalne vode. S pomočjo sprožilnika bi podatke vnesli v bazo podatkov in

jo tako dopolnjevali in ažurirali. Programska oprema bi poskrbela za to, da bi se neobdelani podatki nemudoma ustrezno obdelali in pripravili za prikaz na spletu. Preko interneta bi bili pripravljeni podatki dosegljivi uporabnikom na spletnih straneh Zavoda za zdravstveno varstvo ali spletnih straneh gibanja Modra zastava.

### **6.1.1 Zahteve in potrebe uporabnikov sistema**

Zahteve gibanja Modre zastave glede vzorčenja kopalnih voda, ki so osnova za pridobitev tega ekološkega znaka, smo opisali že v drugem poglavju. Z dodatnimi merilnimi napravami (veter, temperatura, morski tokovi) pa lahko učinkovito merimo tudi parametre, ki jih potrebujejo pri delu drugi subjekti, ki niso neposredno povezani s programom Modra zastava.

Osnovne zahteve uporabnikov lahko strnemo v naslednjih alinejah:

- vsako kopališče ima lahko eno ali več fotografij kopalne vode in kopališča,
- posamezno podjetje (ali občina) lahko upravlja eno ali več kopališč,
- v podjetju je za spremljanje podatkov zadolžena ena ali več oseb, ki imajo lahko različne omejitve pri vpogledih v podatke merilnih postaj,
- vrsta uporabnika določa do katerih podatkov ima posamezni uporabnik dostop,
- vsako kopališče ima lahko eno ali več merilnih boj,
- vsaka boja ima lahko enega ali več senzorjev za merjenje podatkov,
- vsak senzor lahko opravi eno ali več meritev, ki jih je zaradi naknadne obdelave potrebno shranjevati v podatkovni bazi,
- vsak senzor meri določeno vrsto podatka (namembnost senzorja), ki je lahko za objavo ali pa ima vpogled v podatek le določena vrsta uporabnika,
- vsako kopališče ima objavljene določene, izbrane podatke, ki so lahko tudi izračunane vrednosti iz posameznih surovih podatkov.

Uporabnike sistema lahko delimo na dve skupini, in sicer:

- a. nosilci programa Modra zastava (komisija) in raziskovalne inštitucije,
- b. drugi uporabniki, kot so upravljalci naravnih kopališč/marin, uporabniki naravnih kopališč/marin in obiskovalci internet strani gibanja Modra zastava.

Glede na to delitev lahko tudi opredelimo zahteve in potrebe posameznih uporabnikov.

#### **A. Nosilci programa Modra zastava (komisija) in raziskovalne inštitucije**

Nosilci programa Modra zastava (komisija) in raziskovalne inštitucije potrebujejo

nefiltrirane, surove podatke. Vsi podatki morajo biti na voljo, saj je le tako moč dobiti vse potrebne informacije.

Zaradi specifičnosti zahtev uporabnikov v tem primeru niso potrebne obdelave podatkov. Uporabniki v tej skupini imajo večinoma izdelane lastne aplikacije in bi dobljene podatke samo uredili v skladu z zahtevami njihovih aplikacij.

Črpanje podatkov bi izvajali neposredno iz baze surovih podatkov.

## **B. Drugi uporabniki in filtriranje podatkov**

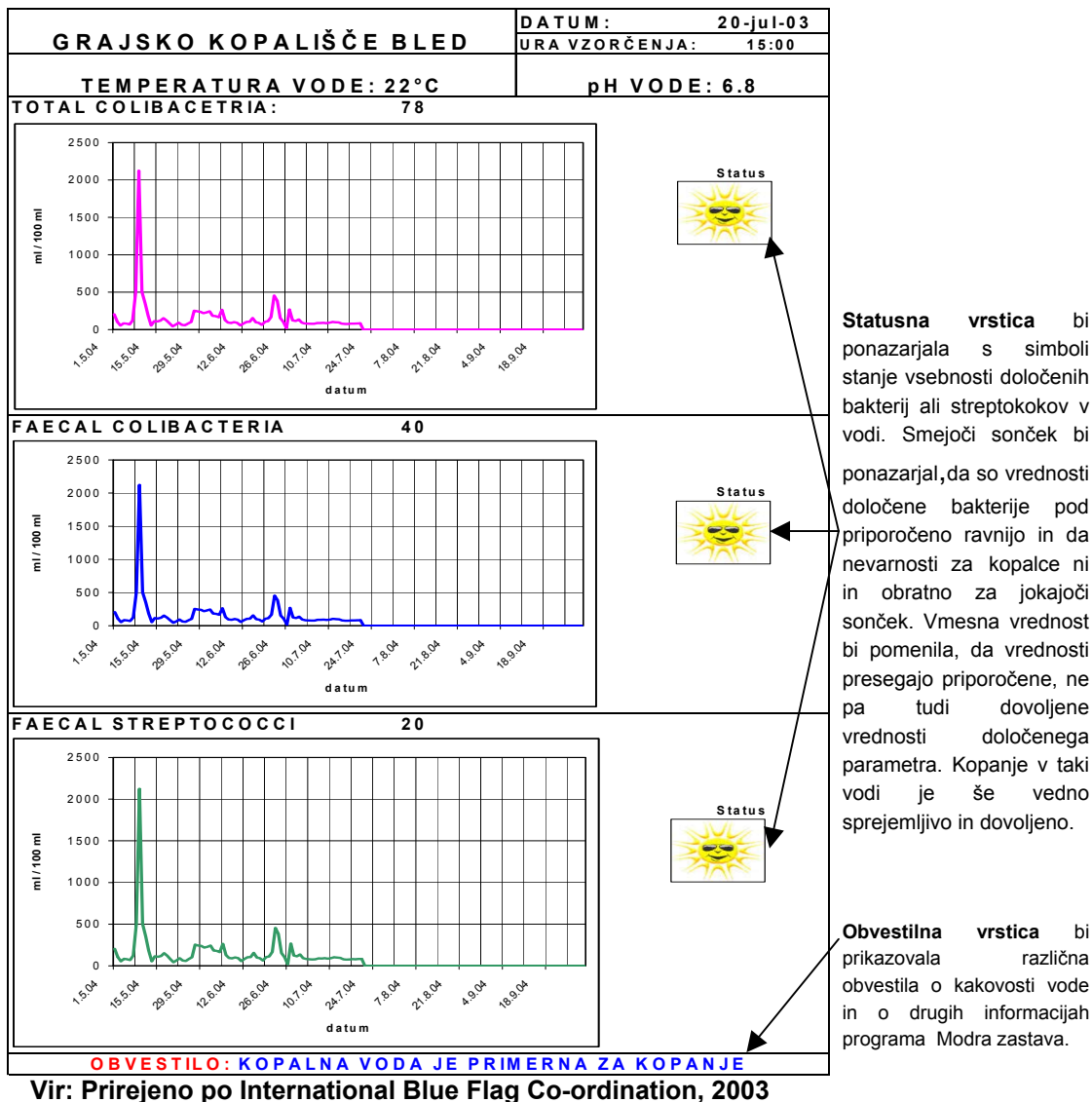
Potrebe gibanja Modra zastava glede kriterijev za ustreznost kakovosti kopalne vode moramo prilagoditi zahtevam tega mednarodnega gibanja. V pravilniku o vzorčenju, ki je v skladu z Direktivo EU o kopalnih vodah, je potrebno zagotoviti vsaj 14-dnevno vzorčenje v času kopalne sezone. V našem primeru bi bilo vzorčenje v realnem ali skoraj realnem času, kar bi pomenilo, da bi imeli namesto vsaj dveh vzorcev mesečno najmanj 240 vzorcev, ob predpostavki vzorčenja na tri ure. Teh podatkov je za osnovno potrebo samega gibanja veliko preveč. Potrebna bi bila filtracija podatkov tako, da bi upoštevali samo podatke ob določenih urah, na primer ob 9.00, ob 12.00 ter ob 15.00, kar bi število podatkov zmanjšalo okvirno na 90 vzorcev mesečno. Za potrebe gibanja pa bi bilo potrebno omejiti tudi obdobje vzorčenje kopalne sezone, v skladu z Direktivo EU o kopalnih vodah, in sicer bi za Slovenijo določili ta termin od 1. maja do 30. septembra. Osnovne zahteve te skupine uporabnikov bomo opredelili v naslednjem poglavju. Zahteva upravljalcev kopališč in nacionalnega koordinatorja za program Modra zastava pa je tudi avtomatsko obveščanje le-teh o morebitnem onesnaženju kopalne vode preko elektronske pošte in SMS sporočil.

## **C. Potrebna dokumentacija**

Kljub kontinuiranemu avtomatskem vzorčenju bi Zavod za zdravstveno varstvo zaradi preteklih ročnih analiz in dolgoročnih spremljanj kakovosti kopalne vode potreboval pisni izpis vzorca za posamezno vzorčno mesto v 7-dnevnih intervalih po ustaljenih metodah vzorčenja.

Upravljalci kopališč bi potrebovali tudi posebno spletno stran, kjer bi se prikazovali zahtevani podatki, ki bi jih namesto v trenutno papirnati obliki objavljali na vhodu ali na večih lokacijah kopališča, in sicer na zaslonih iz tekočih kristalov ali na zaslonih s katodno cevjo. Na spodnji sliki prikazujemo osnutek tovrstne strani.

**Slika 18: Osnutek osnovnih elementov, potrebnih v računalniškem prikazu podatkov za naravna kopališča in kopalce/unorahnika**



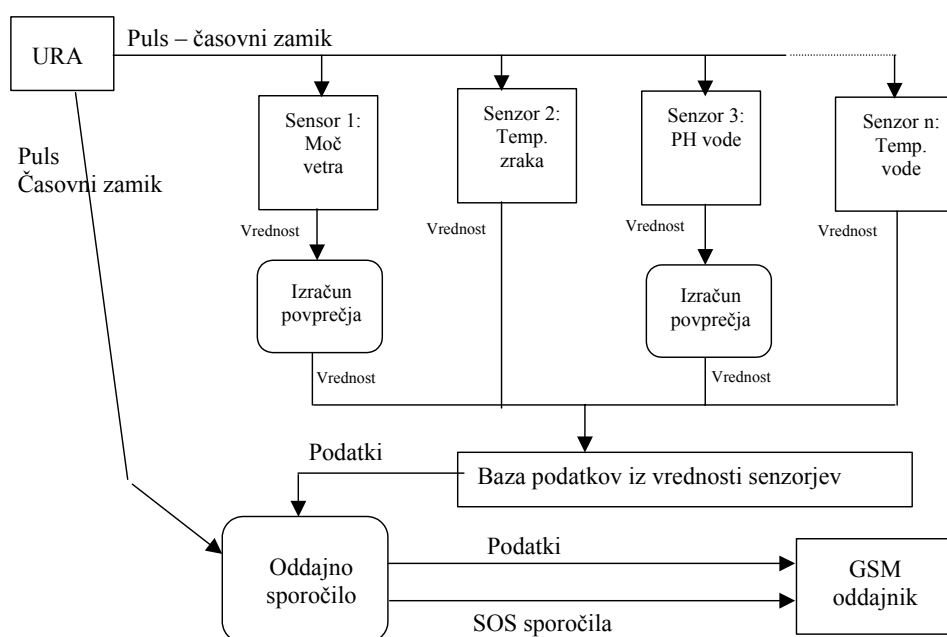
## 6.1.2 Analiza tehnologij in tehnik merjenja kakovosti vode

Kakovost vode ugotavljamo z analizo vzorcev. Te lahko jemljemo ročno, avtomatsko ali polavtomatsko. Ročno jemanje vzorcev poteka s pomočjo epruvet, in to po točno določenih pravilih. Polavtomatsko vzorčenje poteka s posebnimi napravami – avtomatskimi vzorčniki, ki jemljejo vzorce na določenih mestih po vnaprej določenem času. Te vzorce mora fizično prevzeti človek. Avtomatsko lahko jemljemo vzorce vode **s pomočjo senzorjev**, ki so lahko: biosenzorji, optični, akustični ali elektromagnetni. Senzorji so sestavljeni iz t.i. merilnih tipal, ki so lahko uporovna,

induktivna, kapacitivna, elektrodinamična, piezoelektrična ali fotoelektrična.

Vse na boji vpete merilne naprave oziroma senzorji sporočajo podatke **kontrolni enoti na boji**. Ta nadzira in upravlja delovanje posameznih merilnih naprav, izračunava povprečja, kjer je to potrebno, sporoča morebitne okvare in pripravi podatke za pošiljanje. Na sliki številka 19 si pogledjmo shemo delovanja kontrolne enote na boji.

**Slika 19: Shema delovanja kontrolne enote na boji**



**Vir: avtor**

Problem merjenja kakovosti vode in drugih oceanografskih parametrov je v kalibraciji merilnih naprav – senzorjev. Zaradi tega je potrebno senzorje kalibrirati v povprečju vsaj dvakrat letno, kar zahteva dodatna finančna sredstva. V času kalibracije merilna naprava ne deluje, saj kalibracije izvajajo v laboratorijih. Rešitev za to obstaja v podvojevanju merilnih naprav, kar pa dodatno zahteva sredstva, ki si jih večina institucij ne more zagotoviti. Preverjanje kakovosti podatkov, ki jih senzorji pošiljajo, lahko poteka s pomočjo dveh postopkov in sicer s pomočjo avtomatske kontrole kakovosti in s pomočjo vrednotenja oceanografskih podatkov. Na podlagi preverjanja podatkov lahko tudi ugotovimo, če je senzor poškodovan ali nekalibriran.

Na kratko si pogledjmo oba opisana postopka kontrole kakovosti oceanografskih podatkov (Petelin, Malačič, 2001):

**a. Avtomatsko kontrolo kakovosti**, kjer obravnavamo surove ali obdelane podatke, največkrat izvaja računalnik. Sem sodijo postopki za:

- ugotavljanje napak v časovnih intervalih vzorčenja,
- omejitve pri vrednostih podatkov,
- ugotavljanje neutemeljene konstantne vrednosti,
- ugotavljanje hitrosti spreminjanja vrednosti,
- identificiranje manjkajočih podatkov (»podatkovne vrzeli«).

Kontrola kakovosti surovih podatkov je namenjena predvsem ugotavljanju pravilnosti delovanja merilnih naprav ter morebitnih nestabilnosti in motenj pri meritvah. Kontrola kakovosti na obdelanih podatkih, ki jih dobimo šele, ko surove podatke povprečimo, pa ugotavlja napačne podatke in anomalije ter predstavlja množico minimalnih zahtev, ki morajo biti istočasno med seboj združljive in enostavne za uporabo.

**b. Vrednotenje oceanografskih podatkov**, kjer ugotavljamo »smiselnost« podatkov. Pri tem si pomagamo z:

- grafi,
- trendi in vzorci,
- drugimi razpoložljivimi podatki.

Vrednotenje oceanografskih podatkov ugotavlja njihovo »smiselnost« in se v glavnem izvaja na dveh nivojih:

- nižji nivo – postopek izvajamo ročno s preučevanjem diagramov časovnih vrst, ugotavljanjem vzorcev in trendov ter s primerjavo z drugimi razpoložljivimi podatki in znanimi lastnostmi pojavov;
- višji nivo – izvajamo harmonično analizo ter primerjavo posameznih podatkov z drugimi znanimi podatki itd.

Vsakemu podatku v podatkovni bazi dodamo posebno oznako oziroma polje, ki pove, kako kakovosten je ta podatek oziroma za kakšno vrsto napake gre. Pri avtomatski kontroli podatkov vpisujemo tako imenovane oznake (angl. flag). Te oznake lahko takoj izpišemo oziroma prikažemo na diagramih. Popravljanje podatkov pa mora biti omejeno na naslednje:

- »prazne« vrednosti (angl. null) smejo biti pripisane le podatkom, ki so zavrnjeni v grobem preverjanju mejnih vrednosti. Nadalje so prazne vrednosti pripisane tistim, ki so zavrnjeni v testu stacionarnosti (preveč zaporednih enakih vrednosti) in pri identificiranih podatkovnih »vrzelih«;
- interpolacija je dovoljena samo v primerih posameznih »praznih« vrednosti

in je sprejemljiva pri določenih intervalih vzorčenja. Uporabljamo linearno interpolacijo med dvema sosednjima točkama;

- lahko uporabljamo podatke, ki so bili kako drugače izmerjeni na isti lokaciji;
- če pri kontroli podatkov ugotovimo sistematično napako, katere vzrok poznamo, potem je treba popraviti vse prizadete podatke. Vzrok sistematične napake in popravljanje podatkov mora biti dokumentiran.

Na koncu podatke kvalificiramo, tj. jim dodamo validacijske oznake (poleg »flag«), ki povedo, kakšne vrste je podatek s stališča kontrole kakovosti (dober, slab, nezanesljiv, brez vrednosti, interpoliran, vstavljen s pomočjo drugih podatkov).

## **6.2 NAČRTOVANJE INFORMACIJSKEGA SISTEMA S POMOČJO POENOTENEGA JEZIKA ZA MODELIRANJE**

### **6.2.1 Diagram primerov uporabe**

Diagram primerov uporabe prikazuje primere uporabe in akterje, ki so z njimi povezani. Izraža uporabniški pogled na sistem. Primeri uporabe predstavljajo potek dogodkov, ki prinašajo želene rezultate akterju, ki jih izzove. Akter je oseba ali drug sistem izven obravnavanega, ki vstopa v interakcijo s sistemom. Akter ni konkretna oseba, ampak predstavlja določeno vlogo, ki jo le-ta predstavlja (npr. prodajni predstavnik). Primer uporabe predstavlja opis določenega dogajanja, ki nastane v interakciji akterja in sistema. Opišemo ga tekstualno, vendar opredelimo naslednje sestavine (Sturm, 1999, str. 60):

- ime primera uporabe,
- kratek opis namena,
- primarni akter - je tisti, zaradi katerega sistem gradimo,
- sekundarni akter, ki tudi sodeluje,
- začetek - je prva aktivnost primera uporabe,
- konec - je zadnja aktivnost primera uporabe,
- merljivi rezultat,
- potek dogodkov, kjer opišemo pozitivni scenarij,
- alternativni tok dogodkov, ko pogoji za izvedbo niso izpolnjeni,
- morebitni primer uporabe, ki se izvede iz opisanega,
- odprte zadeve.



**Tabela 5: Tekstualni opis primera uporabe**

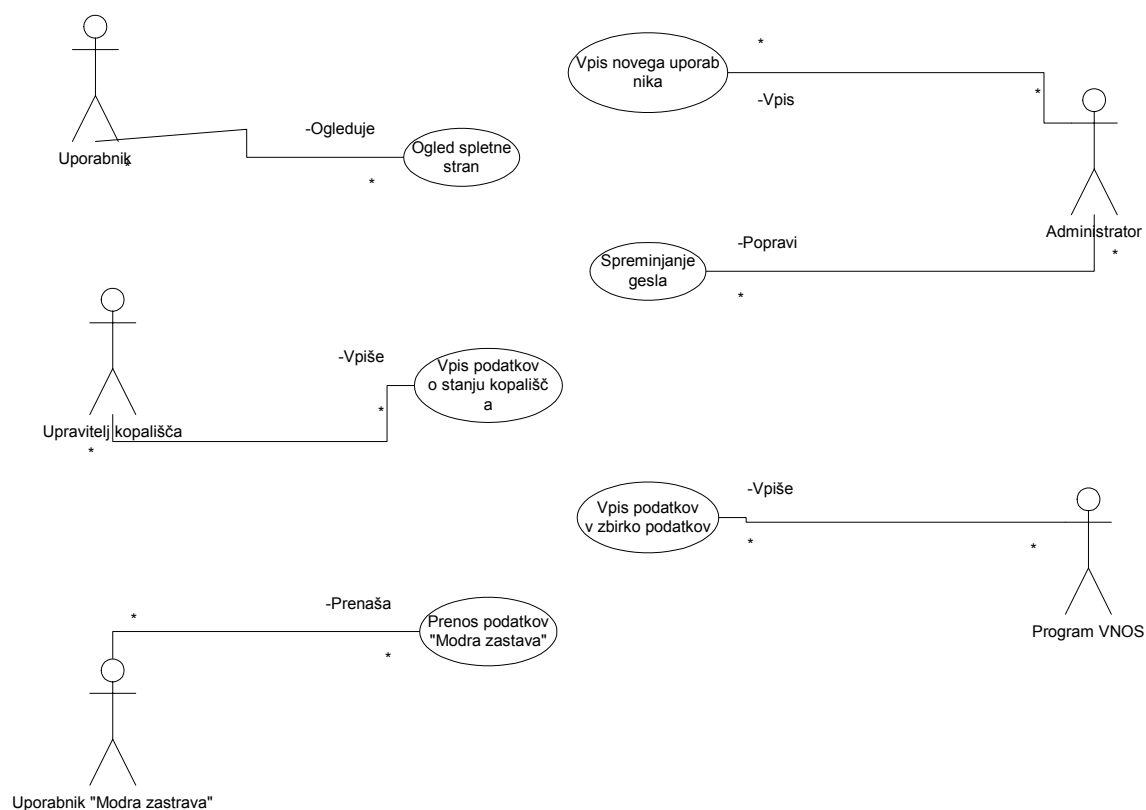
|                                    |  |
|------------------------------------|--|
| <b>Ime</b>                         | Vpis podatkov o novemu uporabniku  |
| <b>Kratek opis</b>                 | Administrator vpiše podatke o novemu uporabniku (upravitelj ali uporabnik "Modra zastava")   |
| <b>Primarni akter</b>              | Administrator  |
| <b>Sekundarni akter</b>            | -  |
| <b>Začetek</b>                     | Administrator izbere spletno stran   |
| <b>Konec</b>                       | Podatki o novem uporabniku so zapisani v zbirko podatkov   |
| <b>Merljivi izid</b>               | Zapisani podatki o novemu uporabniku v zbirki podatkov   |
| <b>Potek dogodkov</b>              | Pred vstopom na stran se mora administrator identificirati in autenticirati (uporabniško ime in geslo). V spletnem pregledovalniku se pojavi stran na kateri lahko administrator vpiše podatke o novemu uporabniku. Vpisuje naslednje podatke: uporabniško ime, geslo, podatke o uporabniku (ime, priimek,...), ali je uporabnik del programa modra zastava in ali je uporabnik upravitelj kopališč. Če je uporabnik upravitelj kopališč mu administrator še priredi kopališča katera upravlja. Po vpisu podatkov administrator zahteva zapis podatkov v zbirko podatkov. Zahteva se skupaj s podatki prenese na spletni strežnik, kjer se zapiše v zbirko podatkov. |
| <b>Alternativni potek dogodkov</b> | -  |
| <b>Povezani primeri</b>            | -  |
| <b>Odprta vprašanja</b>            | -  |

Vir: avtor

Na ta način se opišejo vsi primeri uporabe (Priloga 11) . Diagram primerov uporabe služi prikazovanju povezav med različnimi primeri uporabe in akterji, zato na enem diagramu prikažemo več povezanih primerov uporabe.

Diagrame primerov uporabe uporabljamo za omejitev, kaj naj sistem dela; pravzaprav so način opredeljevanja zahtev do sistema. Primeri uporabe predstavljajo posamezne zaključene funkcionalnosti sistema.

**Slika 20: Diagrami primerov uporabe za spletni strežnik**



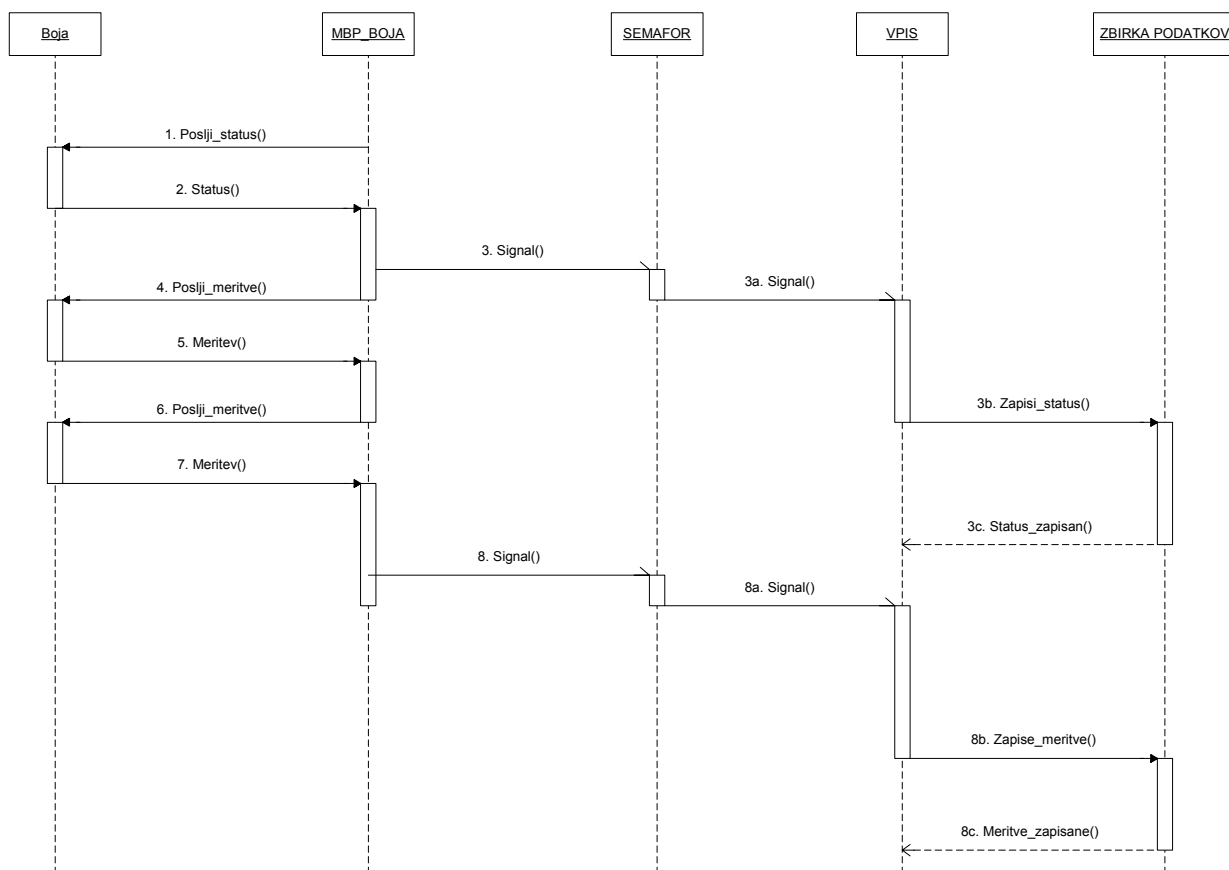
**Vir: avtor**

Za podroben opis izmenjevanja sporočil v sistemu uporabljamo interakcijske diagrame. Ločimo diagram zaporedja in diagram sodelovanja, ki sta dva pogleda iste vsebine in ju lahko med seboj neposredno prevedemo.

## 6.2.2 Diagram zaporedja

Diagram zaporedja (angl. sequence diagram) prikazuje časovno zaporedje izmenjave sporočil med objekti sistema in zunanjimi akterji. Vsebuje objekte, zveze in sporočila. Na spodnji sliki je prikazan diagram zaporedja za prenos podatkov iz boje v zbirko podatkov.

**Slika 21: Diagram zaporedja- prenos podatkov iz boje v zbirko podatkov**



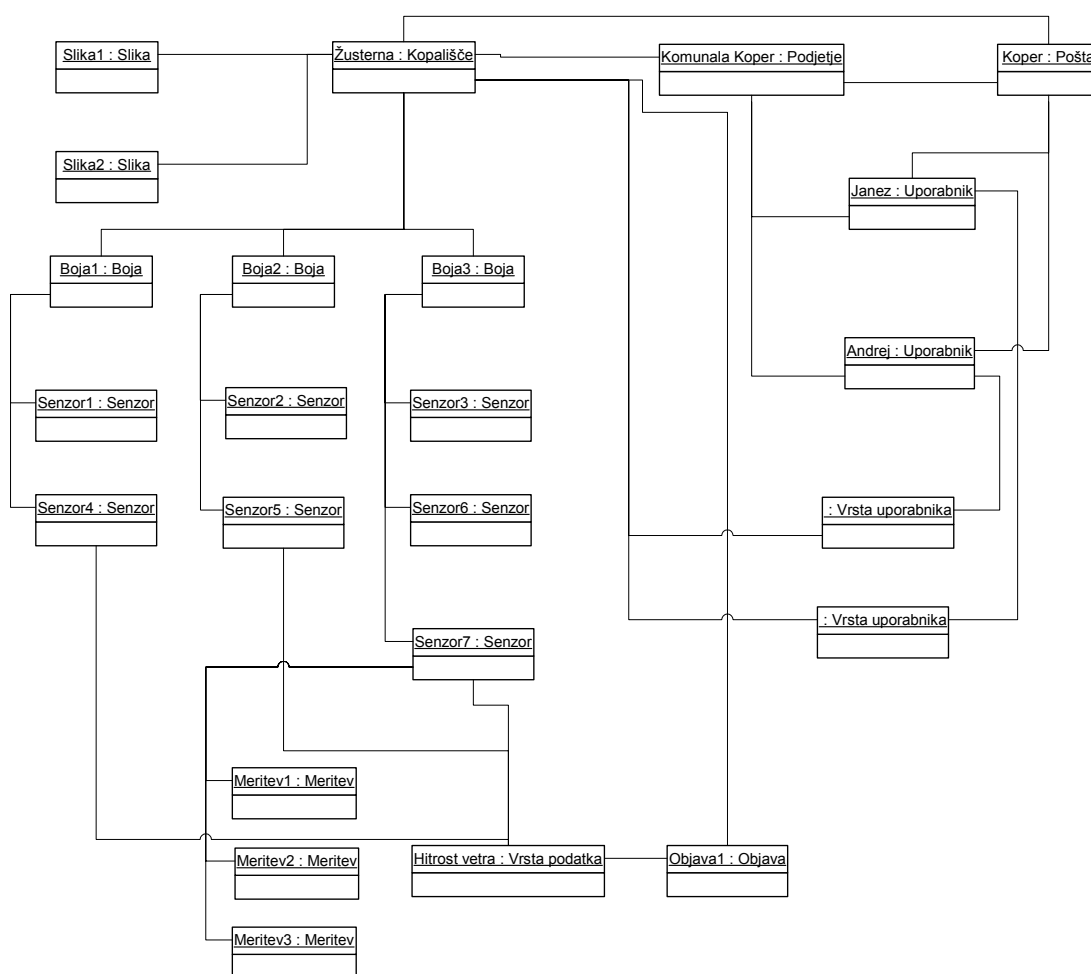
**Vir: avtor**

Navpična črtna črta predstavlja življenjsko črto objekta in prikazuje čas sodelovanja objekta z drugimi objekti. Življenjska črta se konča z uničenjem objekta, ki jo označuje znak »X« na mestu brisanja. Navpični ozki pravokotniki so namenjeni označevanju časa, ko je objekt aktiven in kontrolira izvajanje dogajanja. Usmerjene puščice predstavljajo sporočila, ki se izmenjujejo med objekti. Sporočila v implementaciji naslavljaajo metode komponent, ki izvajajo zahtevane manipulacije notranje strukture objektov. Preostali diagrami zaporedja so prikazani v prilogi 12.

### 6.2.3 Objektni diagram

Objektni diagram prikazuje objekte, predstavlja trenutni posnetek (angl. Snapshot) sistema v nekem trenutku delovanja. Vsebuje objekte in zveze (angl. Link).

**Slika 22: Primer objektnega diagrama**



Vir: avtor

Notacija za prikaz konkretnega objekta je lahko objekt:razred, če pa prikazujemo splošni objekt kot predstavnika razreda, to zapišemo z :razred. Objektni diagrami se redkeje uporabljajo, predvsem za predstavitev povezav med konkretnimi primerki posameznih razredov.

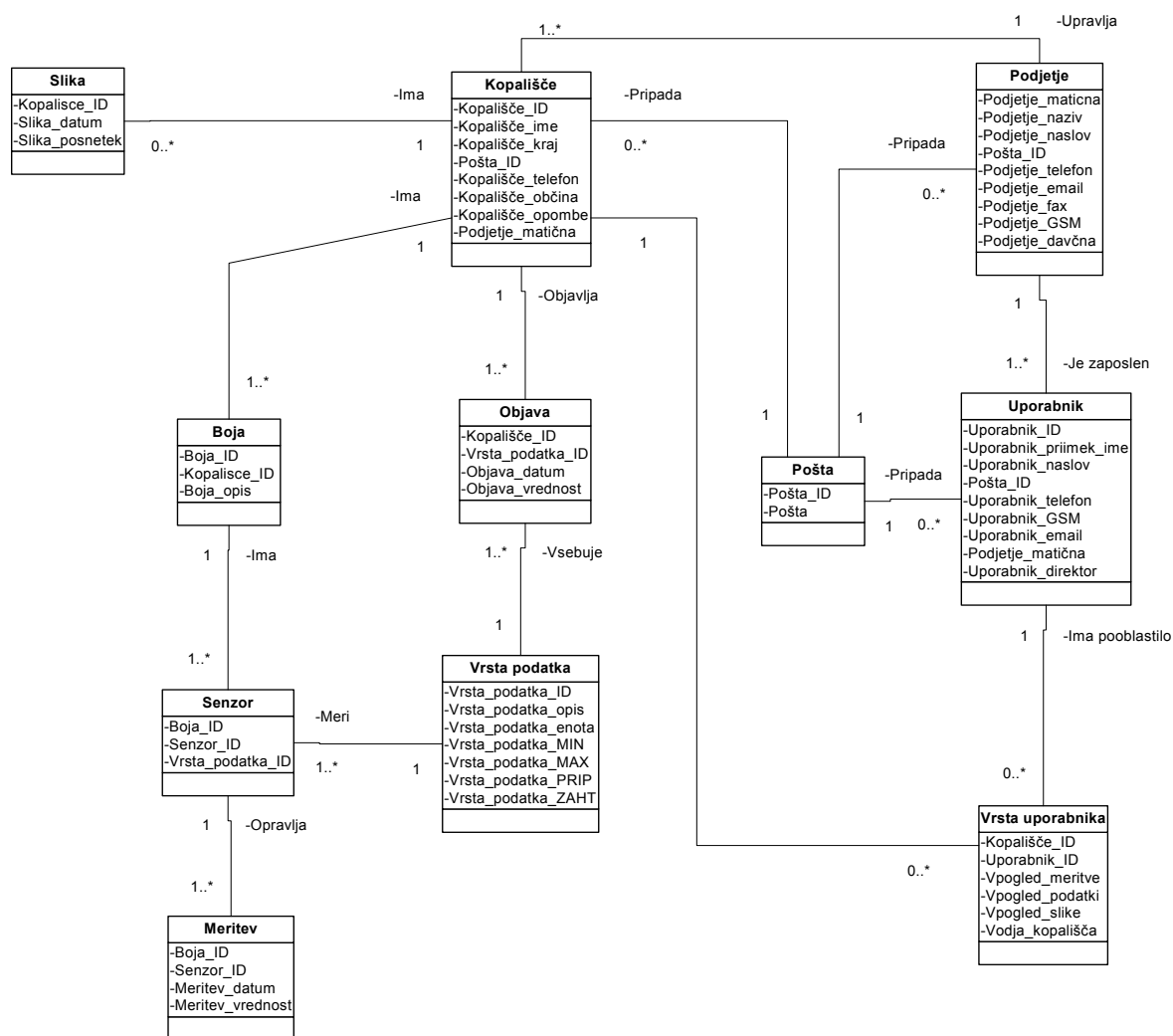
## 6.2.4 Razredni diagram

Razredni diagram prikazuje razrede, vmesnike, sodelovanja in njihove povezave.

Poleg navedenih elementov lahko razredni diagram vsebuje tudi pakete, ki so namenjeni grupiranju elementov večjega modela v več manjših zaokroženih celot. V okviru modela prikazujemo razred kot pravokotnik, ki mu priredimo ime ter navedemo

njegove attribute in operacije. Povezave med razredi so lahko asociacije, agregacije, sestavljene agregacije in generalizacije. Povezavam opredelimo števnost in smer, lahko jim tudi opredelimo vlogo (angl. Role), ki jo ima razred na posameznem zaključku povezave.

**Slika 23: Primer razrednega diagrama – zbirka podatkov za poročanje**



Vir: avtor

## 6.2.5 Diagram aktivnosti

Diagram aktivnosti uporabljamo za prikaz zaporednih in vzporednih korakov v izvajanju procedure. Diagram vsebuje aktivnosti, prehode med njimi, odločitvene aktivnosti, ki glede na postavljene pogoje prehajajo v različne naslednje aktivnosti, ter uskladjitvene črte (angl. synchronization lines), v katerih se vzporedni tokovi izvajanja uskladijo in se opravi prehod do naslednje aktivnosti, ko so izpolnjeni

določeni pogoji.

Uskladitvena črta je prikazana kot debela vodoravna daljica, v katero vodita dva prehoda, iz nje pa vedno le en prehod. Uporaba uskladitvenih črt omogoča modeliranje vzporednega izvajanja procesov. Kadar z diagramom aktivnosti modeliramo prehajanje procesa med več različnimi enotami organizacije, ko gre za prehod odgovornosti, lahko z navpičnimi črtami (angl. Swimlines) prikažemo tovrstno delitev.

Elementi, povezave in diagrami so osnovni gradniki poenotenega jezika za modeliranje, s katerim izdelujemo različne modele želenega informacijskega sistema glede na posamezno fazo razvojnega procesa.

Preostali diagrami aktivnosti so predstavljeni v prilogi 12.

**Slika 24: Diagram aktivnosti v primeru ogleda spletne strani**

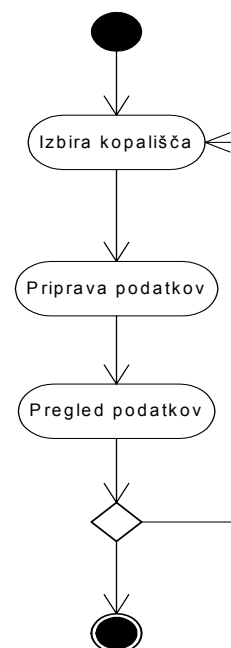


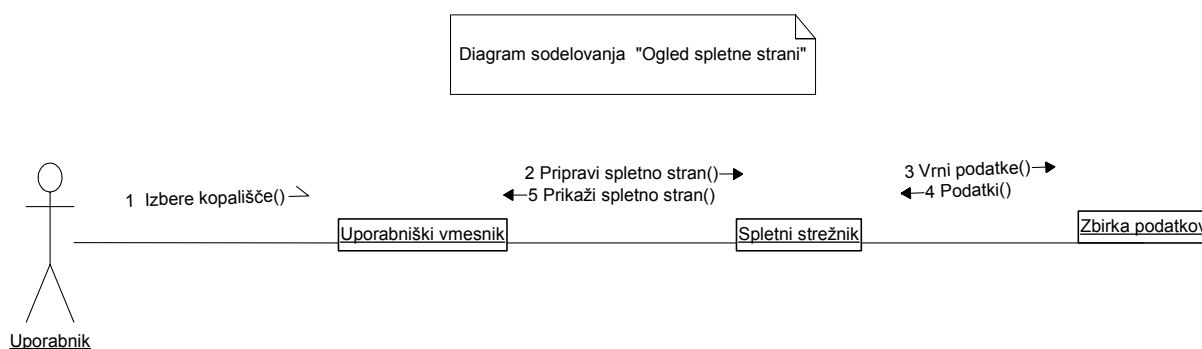
Diagram aktivnosti "Ogled spletne strani"

**Vir: avtor**

### 6.2.6 Diagram sodelovanja

Diagram sodelovanja predstavlja izmenjavo sporočil na drugačen način, ki bolj poudarja organizacijo objektov, ki sodelujejo v interakciji. V tem primeru je zaporedje pošiljanja sporočil opredeljeno z zaporedno številko sporočila, ki je pripisana sporočilu. Na diagramu sodelovanja so označene zveze (angl. Link) med objekti, ki na diagramu zaporedja niso posebej prikazane.

**Slika 25: Diagram sodelovanja v primeru ogleda spletne strani**



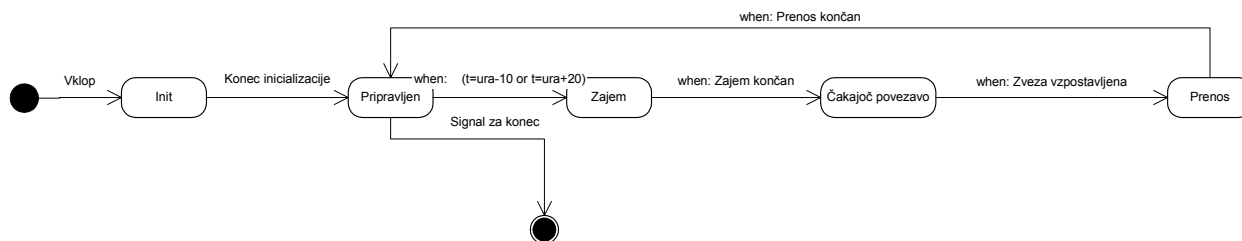
Vir: avtor

Preostali diagrami sodelovanja so prikazani v prilogi 12.

## 6.2.7 Diagram stanj

Diagram stanj prikazuje stanja, prehode, dogodke in aktivnosti. Uporablja se za prikaz spreminjanja stanja izbranega objekta glede na dogodke, ki spremembo stanj povzročajo.

**Slika 26: Diagram stanj objekta - boja**



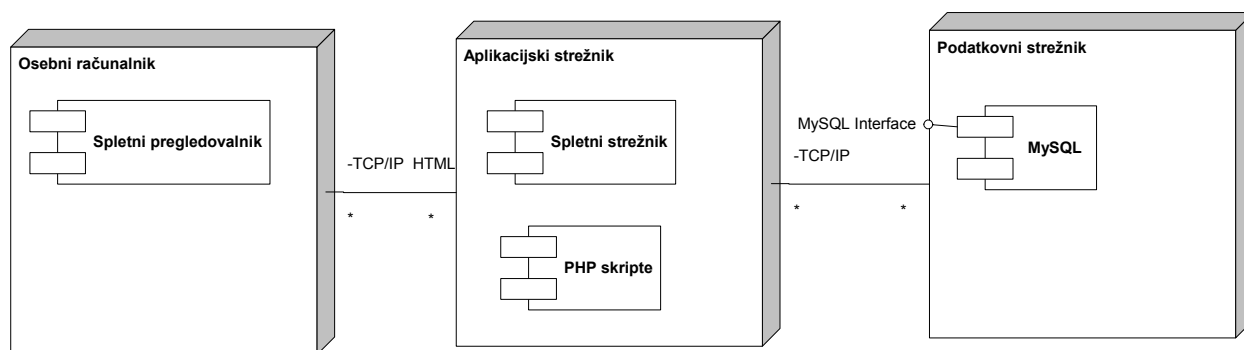
Vir: avtor

Preostali diagrami stanj so prikazani v prilogi 12.

## 6.2.8 Diagram porazdelitve

Diagram porazdelitve (angl. deployment diagram) prikazuje namestitve komponent v vozlišča, ki predstavljajo strojno opremo, na kateri se komponente izvajajo. Vsebuje vozlišča in povezave med vozlišči, lahko pa prikazujemo tudi porazdelitev komponent med vozlišča.

**Slika 27: Diagram porazdelitve**

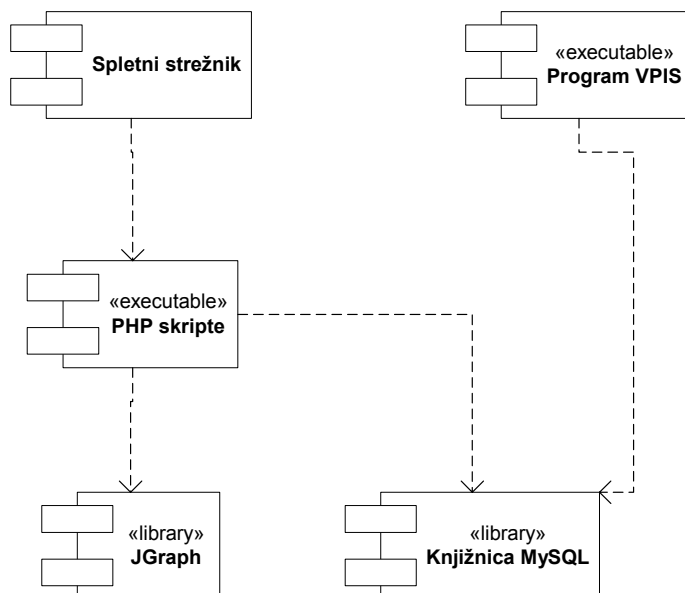


Vir: avtor

## 6.2.9 Diagram komponent

Diagram komponent prikazuje komponente in povezave med njimi. Vsebuje komponente, vmesnike, povezave odvisnosti, asociacije, generalizacije in uresničitve. Z njim lahko modeliramo zaključene dele programske kode, ki implementirajo funkcionalnost izbranih razredov, in povezave med njimi. Komponente lahko združujemo v pakete za razdelitev sistema v več manjših zaključenih celot. Prikazane so odvisnosti med komponentami in vmesniki kot način komuniciranja med njimi.

**Slika 28: Diagram komponent za spletni strežnik**



Vir: avtor



## 6.3 Sestav informacijskega sistema, protokoli in varovanje podatkov

### 6.3.1 Boja

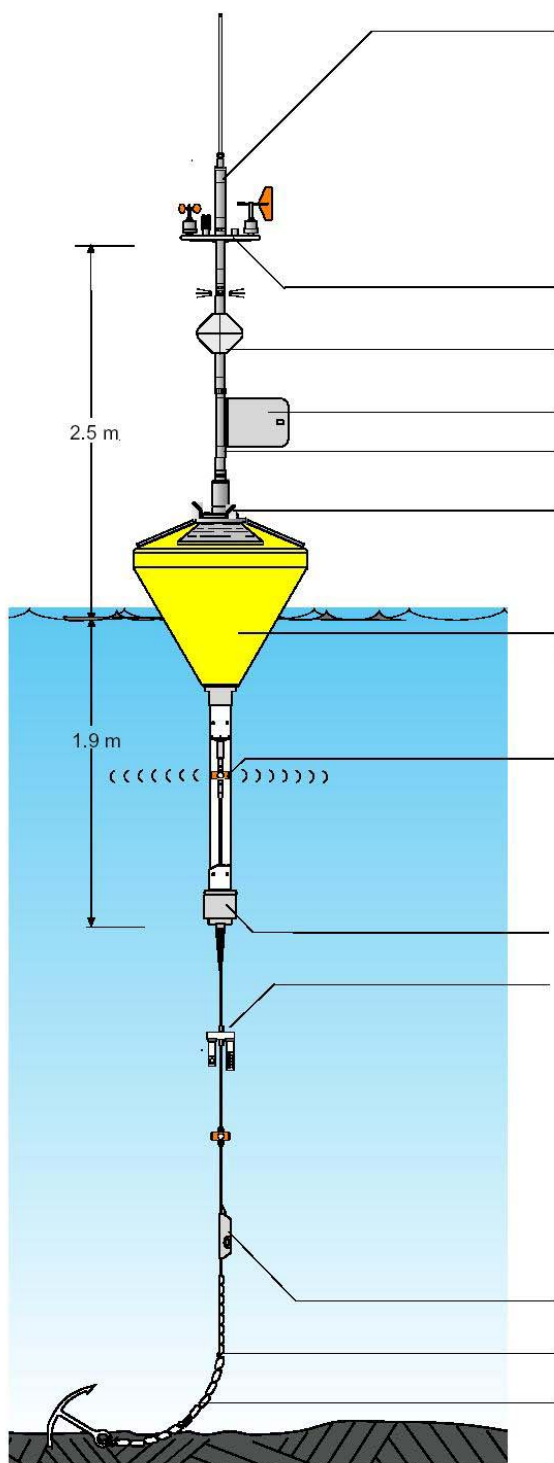
Na boji (slika 29, str. 75) imamo senzorje, ki jih bomo uporabili v sistemu, in sicer:

- senzor za merjenje **hitrosti vetra** pošilja podatke o povprečni moči vetra in sunke v določenem intervalu. Deluje na intervalu do 79 m/s z natančnostjo  $\pm 2\%$ ;
- senzor za merjenje **smeri vetra** je sestavljen iz vrtljive vetrnice, ki je vpeta na nosilec. Meri smer vetra od moči vetra 0,3 m/s dalje z natančnostjo  $\pm 5\%$ ;
- senzor za merjenje **temperature ozračja** v razponu od  $-44^{\circ}\text{C}$  do  $+49^{\circ}\text{C}$  in natančnostjo  $\pm 0,1\%$ ;
- senzor **zračnega pritiska** meri pritisk v intervalu od 920 do 1080 hPa z natančnostjo  $\pm 0,2$  hPa;
- »**dataloger**« ima 31 kanalov in na katerega so priključeni senzorji. Določimo lahko intervale vzorčenja, in sicer 0.5, 1, 2, 5, 10, 20, 30, 60, 120, 180-minutne, in to glede na zasedenost kanalov. Vsak zaseden kanal potrebuje namreč 4 sekunde za prenos podatkov;
- Dopplerjev senzor vodnega toka meri vodni tok na principu Dopplerjeve metode. Meri tudi temperaturo vode. Lastnosti tega senzorja so:
  - **hitrost vodnega toka** meri v intervalu 0-300 cm/s z natančnostjo  $\pm 0,55$  cm/s,
  - **smer vodnega toka** meri v intervalu 0-360° z natančnostjo  $\pm 5\%$ ,
  - **temperaturo vode** meri v intervalu  $-10^{\circ}\text{C}$  do  $+43^{\circ}\text{C}$  z natančnostjo  $\pm 0,1^{\circ}\text{C}$ ;
- senzor za merjenje **slanosti vode** meri slanost v intervalu 0-40 ppt z natančnostjo  $\pm 0,2$  ppt;
- senzor za merjenje **motnosti vode** deluje na principu senzorja s kombinacijo LED in foto diode in meri motnost na podlagi števila odbojnih delcev v vodi v

intervalu 0,1-500 NTU in natančnostjo  $\pm 2\%$ ;

- senzorji za merjenje **skupnih koliformnih in fekalnih koliformnih bakterij ter fekalnih streptokokov** zaenkrat še niso na voljo, zato je trenutno mogoče te meritve opraviti zgolj ročno. Število koliformnih in fekalnih koliformnih bakterij ter fekalnih streptokokov izražamo z MPN (angl. Most Probable Number = najverjetnejše število), ki nam pove, koliko bakterij ali streptokokov je v 100 ml ali 1 litru kopalne vode. Število koliformnih bakterij lahko prikažemo tudi s koli litrom, kar pomeni najmanjšo količino vode, v kateri je vsaj ena bakterija, njihovo prisotnost pa določamo tudi z metodo membranske filtracije. Slednja poteka tako, da filter, skozi katerega smo precedili odpadno vodo, položimo na specifično selektivno gojišče, kjer porastejo koliformne bakterije v značilni obliki in barvi. Bakterijske kolonije preštajemo in jih izrazimo v količini na 100 ml ali na 1 liter vzorca. Po navedbah španskega podjetja URSAT (URSAT, 20. 6. 2003) končna testiranja na teh novih senzorjih že potekajo;
- senzor za ugotavljanje **vsebnosti fenolov** lahko temelji na več vrst tehnologij, kot je metal-oksidni senzor (MOS), prevodno-polimerni senzor, kristalno kvarčni mikrosenzor ter optični senzor (Stuetz, 2003). Tudi tej senzorji so še v zgodnji razvojni fazi;
- senzor za **ugotavljanje transparentnosti vode** temelji na posebnem potopnem optičnem senzorju. Transparentnost vode meri s pomočjo infrardeče svetlobe v razponu od 2 do 100 cm;
- boja bi imela tudi manjšo, na vodo odporno oziroma zaščiteno **digitalno foto-kamero** za ugotavljanje oziroma evidentiranje plavajočih delcev in oljnih madežev v okolici boje.

**Slika 29: Primerna boja za uporabo v sistemu - boja AANDERAA CMB 4280**



**GSM oddajnik** z modemom 900/1800 Mhz, doomet večji od 20 km, SIM kartica lokalnega operaterja.

Možnost kombinacije/nadgradnje z VHF 141-143 MHz, UHF 400-500 MHz z dometom do 10 km, Argos oddajnika za povezavo preko satelita ali radijskega oddajnika z dometom do 20 km.

**Nosilec** senzorjev za meteorološke parametre: hitrost vetra, smer vetra, temperatura zraka

**Utripajoča svetilka**

**Veternica**

**Kontrolna enota** s spominsko enoto za shranjevanje podatkov

**BOJA** premera 90 cm, nosilnosti 35 kg in lastne teže 400 kg, ki vsebuje »**dataloger**«, akumulatorsko baterijo 12 V z napetostjo 21 Ah ter z možnostjo nadgradnje s senzorji GPS in foto-kamero

**Osnovna oprema** boje vsebuje 6 W sončne celice s skupno močjo 30 W in največjim tokom polnjenja 2 A. Izmerjena napetost 15 V.

**Dopplerjev senzor** vodnega toka, ki poleg smeri meri tudi moč toka in temperaturo vode.

**35 kg utež**

**Nosilec** dodatnih senzorjev:

- za pH
- za skupne koliformne bakterije
- za fekalne koliformne bakterije
- za fekalne streptokoke
- za ugotavljanje vsebnosti fenolov

**Sponka** s senzorjem za ugotavljanje transparentnosti vode

**Veriga** (do 6 m)

**Sidro** z verigo (do 3 m). Namesto sidra lahko uporabimo za pričvrstitev boje na betonski blok.

**Vir: Prirejeno po Aandera Instruments, 25. 3. 2003**

### 6.3.2 Izbira sistema za upravljanje z bazo podatkov

Sistem za upravljanje baze podatkov (angl. Database Management System – DBMS, v nadaljevanju SUBP) je zbirka programov, ki omogočajo tvorjenje, uporabo in vzdrževanje baz podatkov (Jaklič, 1999).

Za sistem za upravljanje baz podatkov izberemo MySQL, ki se uporablja za dodajanje, dostop in procesiranje podatkov, shranjenih v bazi. Izbira MySQL-a za SUBP je povsem enostavna. MySQL je brezplačen, gre za t.i. »Open Source« programsko opremo, in zelo hiter, zanesljiv in enostaven za uporabo v spletnih aplikacijah, ki potrebujejo dostop do baz podatkov na internetu. MySQL uporablja GPL (angl. General Public License) ([URL: <http://www.gnu.org/>], 5.2.2003), ki definira, kaj lahko (brezplačno) naredimo s programsko opremo v različnih primerih.

Tak SUBP uporablja tudi Morska biološka postaja Piran in po njihovih navedbah le-ta popolnoma ustreza njihovim potrebam in zahtevam.

### 6.3.3 Strežnik in periferna oprema

Podatki z boj se bodo prenašali v določenih časovnih intervalih enkratno, zato je potrebno dobro poskrbeti za varnost. Zato postavimo dvonivojsko konfiguracijo, sestavljeno iz dveh računalnikov, enega izključno za sprejem podatkov, drugega kot računalnik, ki poskrbi za prenos podatkov in je povezan z internetom.

Dostop do prvega računalnika, kjer je postavljena baza MySQL, ima izključno samo vzdrževalec strežnika. Prvi računalnik je povezan z drugim računalnikom preko lokalnega omrežja. Dostop do tega računalnika je možen samo pooblaščenim osebam in to samo lokalno. Iz interneta lahko dostopamo samo do drugega računalnika in hkrati samo do dovoljenih storitev, ker je prvi računalnik zaščiten s požarnim zidom. Do lokalnega omrežja, kjer se nahaja podatkovni strežnik, dostop iz interneta ni mogoč. S tem zagotovimo dodatno varnost. Zaradi varnosti podatkov je priporočljivo, da prvi računalnik, t.i. podatkovni strežnik, vsebuje še:

- RAID diske, postavljene v zrcalnem načinu in
- natančno določeno politiko ažuriranja in vzdrževanja podatkov.

Drugi računalnik ima replicirane podatke iz prvega računalnika. Na tem računalniku bi deloval spletni strežnik Apache, replicirana baza MySQL, skriptni jezik PHP (angl. Hypertext Preprocessor) in tako bi potekala povezava z internetom.

Priklopna konfiguracija v računalniku: »P3« Procesor hitrosti vsaj 600MHz, z RAM spominom, velikosti vsaj 256Mb, 3Ware IDE HW RAID kontroler, za začetek tri trde diske, IDE velikosti vsaj 40Gb in hitrosti 7.200 rpm, ki bi jih priklopili v »Raid mirror« načinu tako, da bi bil tretji »hot-spare drive«, torej rezervni oziroma podporni disk. Ustrezna mrežna kartica bi bila lahko Intelova Express PRO 100.

Na strežnik bi namestili Debian Stable distribucijo Linux-a s kerneljem 2.4.19, uporabili LVM (angl. The Logical Volume Manager) za logične particije, blokirali skoraj ves prihajajoč promet, v MySQL-u bi vklopili funkcijo binarni dnevnik (angl. binary logging), kar je tudi nujno za replikacijo in omogoča tudi zaščito oziroma »backup«. Uporabili bi tudi Ext3fs – datotečni podsistem – ki ima funkcijo preprečevanja izgube podatkov in pisanja dnevnika.

Podoben sistem imajo tudi na Morski biološki postaji Piran, zato bi bilo smotrno za potrebe Modre zastave uporabljati že obstoječi in delujoči strežnik. To bi omogočilo obojestranske koristi: Morski biološki postaji dostop do dodatnih oceanografskih podatkov, samemu programu Modra zastava pa zmanjšanje stroškov investicije in vzdrževanja.

#### **6.3.4 Programska oprema in uporabniški vmesnik**

Programsko opremo lahko delimo na sistemsko in uporabniško. Najpomembnejši del sistemske programske opreme je operacijski sistem, ki je v našem sistemu Linux. Uporabniški programi, ki jih uporabljamo v našem sistemu, pa so internet brskalnik (npr. Netscape Navigator) na uporabnikovi strani, na strežnikovi strani pa spada med programsko opremo program Apache za serviranje spletnih strani, program Qmail za pošiljanje (in sprejemanje) elektronske pošte ter antivirusni program, npr. TrendMicro. Kot vmesnik za komunikacijo z MySQL bazo uporabljamo PHP-modul z MySQL-podporo. Teh programov uporabniki ne uporabljajo neposredno, so pa glavni nosilci servisov, ki jih nudi strežnik. Ne smemo pozabiti na programsko opremo za zajem in izmenjavo podatkov med bojami in strežnikom ter programa za avtomatsko osveževanje baze podatkov slovenskega podjetja Ditel. Te programe predstavljamo najprej, saj predstavljajo izredno pomembno sestavino samega informacijskega sistema. Te programe uporablja tudi Morska biološka postaja Piran, ki je z delovanjem teh programov več kot zadovoljna. Uporabniški vmesnik Okna, program Mbp\_boja in program Vpis so kot takšni primerni za uporabo v avtomatiziranem informacijskem sistemu za potrebe programa Modra zastava.

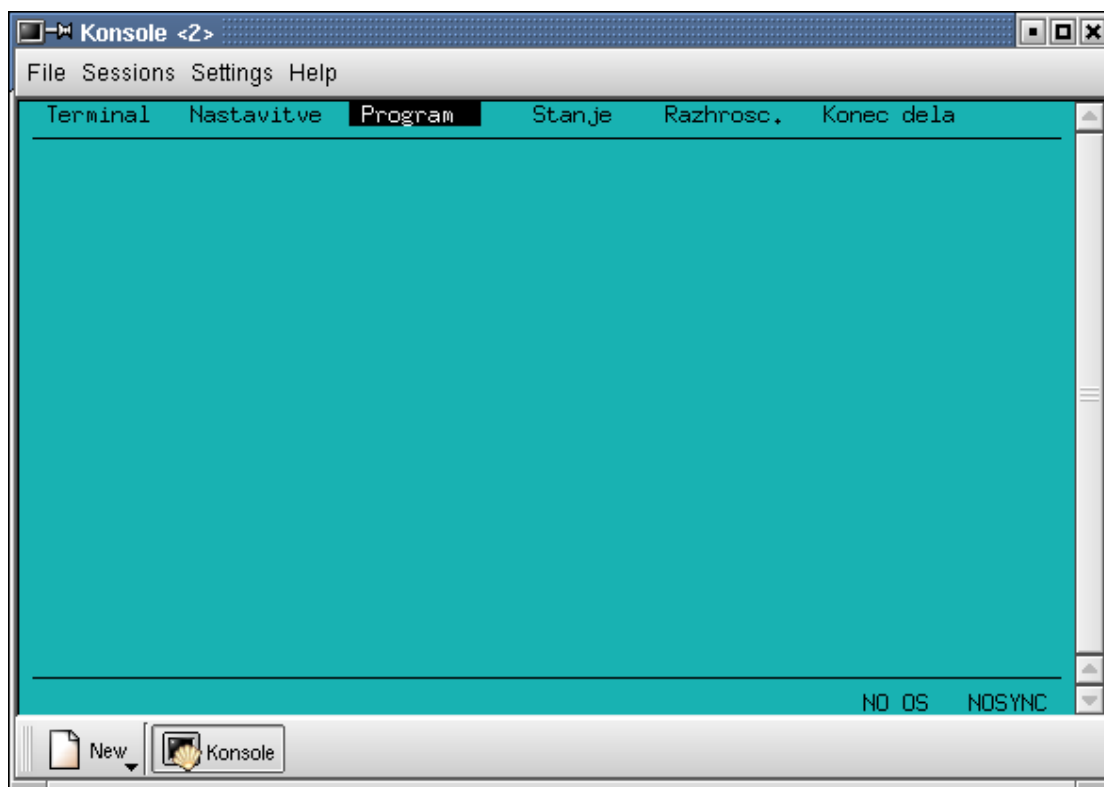
#### 6.3.4.1 Uporabniški vmesnik Okna

Uporabniški vmesnik za sistem zajema oceanografskih podatkov uresničuje program Okna. Glavni namen programa je nastavitve parametrov delovanja sistema, omogoča pa tudi terminalsko povezavo s posameznimi merilnimi instrumenti na boji, daljinsko nalaganje in zagon nove uporabniške programske opreme na boji in spremljanje delovnih pogojev boje (napetost baterij, skupna poraba toka, temperatura in relativna vlaga v ohišju ter status uporabniške programske opreme). Program podpira tudi določene funkcije razhroščevanja programske opreme na boji, kot so: branje podatkov neposredno iz pomnilnika komunikacijskega multiplekserja na boji, vpis podatkov v prej omenjeni pomnilnik ter spremljanje nepravilnih meritev posameznih instrumentov na boji.

Po sinhronizaciji zveze med oceanografsko bojo in sistemom na kopnem se zveza vzpostavi tako, da se v zgornji vrstici izpišejo vse opcije glavnega menija programa, spodnja vrstica pa služi kot statusna vrstica. V desnem delu statusne vrstice so prikazani do štirje statusni podatki:

- skrajno desno se nahaja statusni napis »SYNC« ali »NOSYNC«, ki pomeni, da zveza je oziroma ni vzpostavljena;
- eno mesto bolj v levo je napis »OS OK« ali »NO OS«, ki označuje, da uporabniška programska oprema na boji deluje oziroma ne deluje;
- še malo bolj v levo se lahko pojavi napis »DACQ«, ki je aktiven takrat, ko programska oprema na boji zbira podatke instrumentov;
- na istem mestu kot napis »DACQ«, se lahko pojavi napis »TRANS«, ki sporoča, da je v teku prenos podatkov iz boje na kopno.

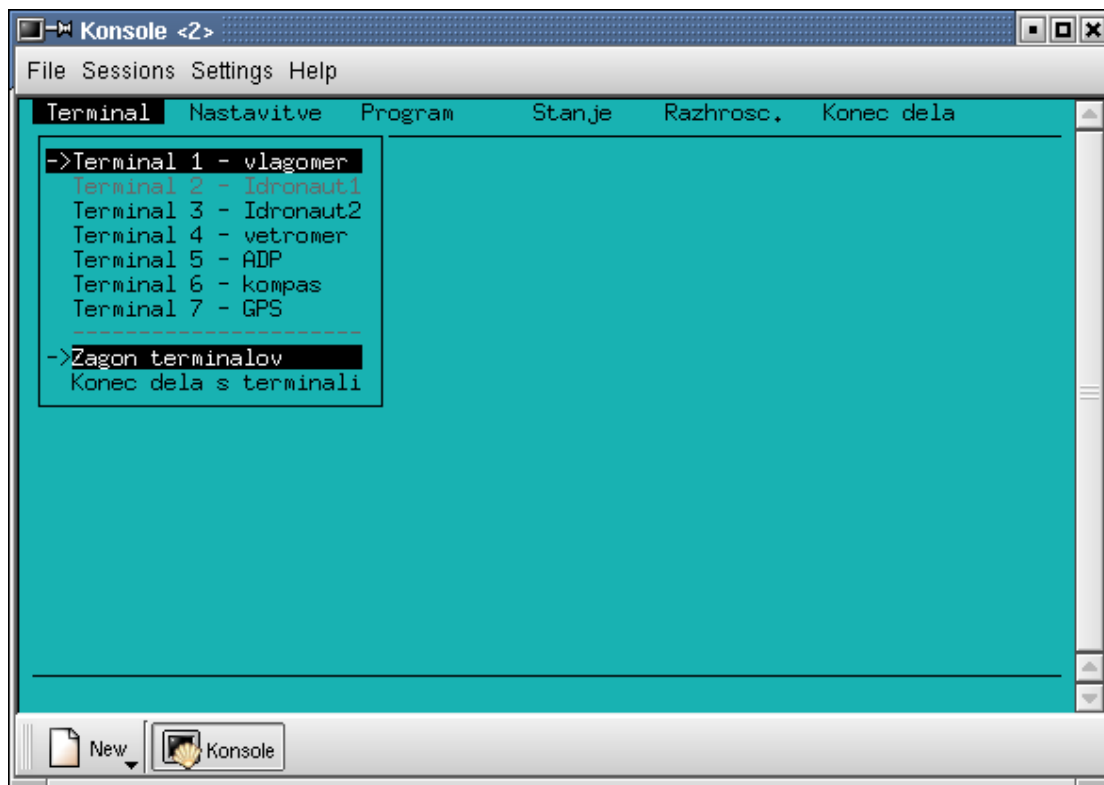
**Slika 30: Glavni meni**



**Vir: Šonc, 2003, str. 8**

Upravljanje delovanja instrumentov poteka preko podmenija za terminalski dostop, ki se nekoliko razlikuje od drugih podmenijev, saj lahko izberemo več instrumentov. Izbrani instrumenti so v podmeniju osvetljeni, hkrati pa lahko izberemo največ tri instrumente, izbiramo pa lahko le med tistimi, ki niso v merilnem krogu in so v podmeniju zapisani s črno barvo. Če poskušamo izbrati več kot tri instrumente, nas bo program opozoril na to z dodatnim oknom z opozorilnim napisom in našo dodatno izbiro zanemaril. Instrumentov, ki so v merilnem krogu in jih nadzira program za zajem podatkov na boji, ne moremo izbrati, terminalski dostop do njih je onemogočen in v podmeniju so označeni s sivo barvo. Posamezne instrumente lahko vključimo ali izključimo iz merilnega kroga v podmeniju »Nastavitve«. Ko smo zaključili z izbiro instrumentov, se s tipko za pomik navzdol pomaknemo na opcijo »Zagon terminalov« in jo potrdimo s tipko Enter (slika 31).

Slika 31: Podmeni za terminalski dostop

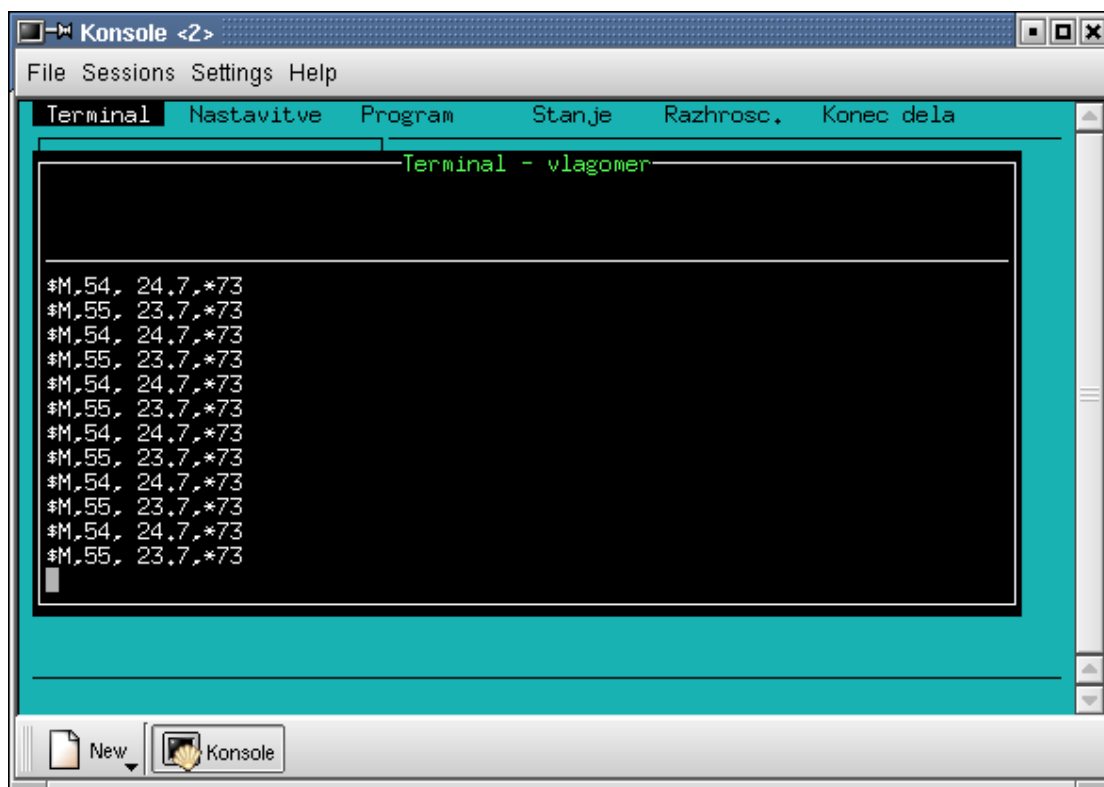


Vir: Šonc, 2003, str. 9

Odrplo se bo eno ali več novih oken. Na zgornjem robu vsakega takega terminalskega okna je označen instrument, ki ga upravljamo. Vsi znaki, ki jih bomo vtipkali in jih bo program poslal instrumentu na boji, se bodo prikazali v zgornjem, manjšem delu okna, vsi znaki, ki jih je vrnil posamezen instrument, pa se bodo prikazali v spodnjem, večjem delu okna. Na vtipkane znake se bo odzivalo le sprednje okno, okna v ozadju pa bodo sprejemala le znake z instrumentov. Med posameznimi okni se sprehajamo in jih tako postavimo v ospredje s tipko »Tab«. Če želimo končati delo s terminalskimi okni ali vklopiti dodatna okna, pritisnemo najprej tipko F10. S tem se vrnemo nazaj v podmeni za delo s terminalskimi okni. Nato lahko vključimo ali izključimo dodatne instrumente po zgoraj opisanem postopku ali pa izberemo opcijo »Konec dela s terminali«. Če smo izbrali slednjo, se bodo vsa terminalska okna zaprla, instrumenti na boji pa se bodo izklopili.



Slika 32: Terminalsko okno



Vir: Šonc, 2003, str. 10

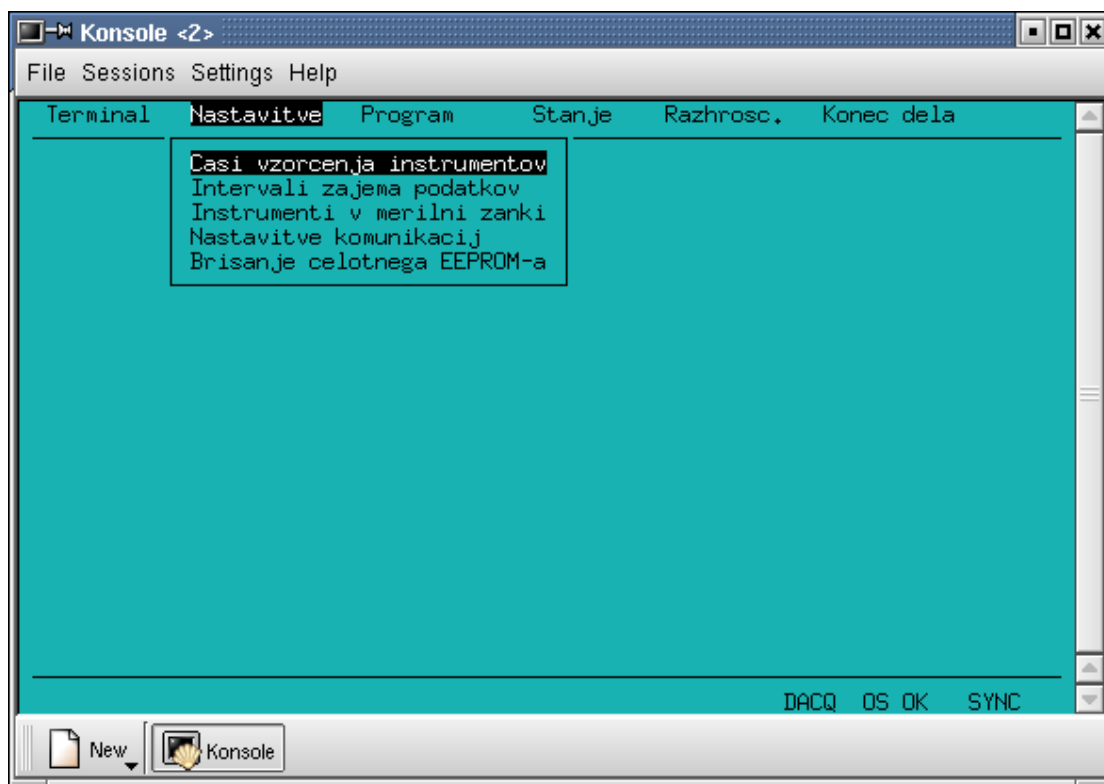
Vsi sprejeti znaki iz instrumentov med terminalskim dostopom bodo zapisani tudi v datoteko surovih podatkov na glavnem strežniku, in sicer za vsak instrument svojo. Pri vsakem novem dostopu do posameznega instrumenta se datoteka ustvari na novo in stari podatki se zbrisejo. Če želimo podatke ohraniti, moramo datoteko pred ponovnim dostopom prepisati drugam.

Nastavljanje parametrov vzorčenja instrumentov izvajamo v podmeniju »Nastavitve« programskega sistema Okna, in sicer (slika 33):

- čase oziroma hitrost vzorčenja posameznih instrumentov,
- časovne parametre, ki določajo čas zajema podatkov, čas mirovanja instrumentov in druge časovne parametre, ki jih potrebuje program za zajem podatkov,
- postavljamo instrumente v/iz merilne zanke,
- komunikacijske parametre in
- brišemo lahko vse nastavitve.

Zadnji opciji sta na voljo le sistemskemu (angl. root) uporabniku, ker se običajno nikoli ne uporabljata in vplivata na pravilno delovanje sistema na zelo nizkem nivoju.

**Slika 33: Podmeni nastavitve sistema za zajem podatkov**

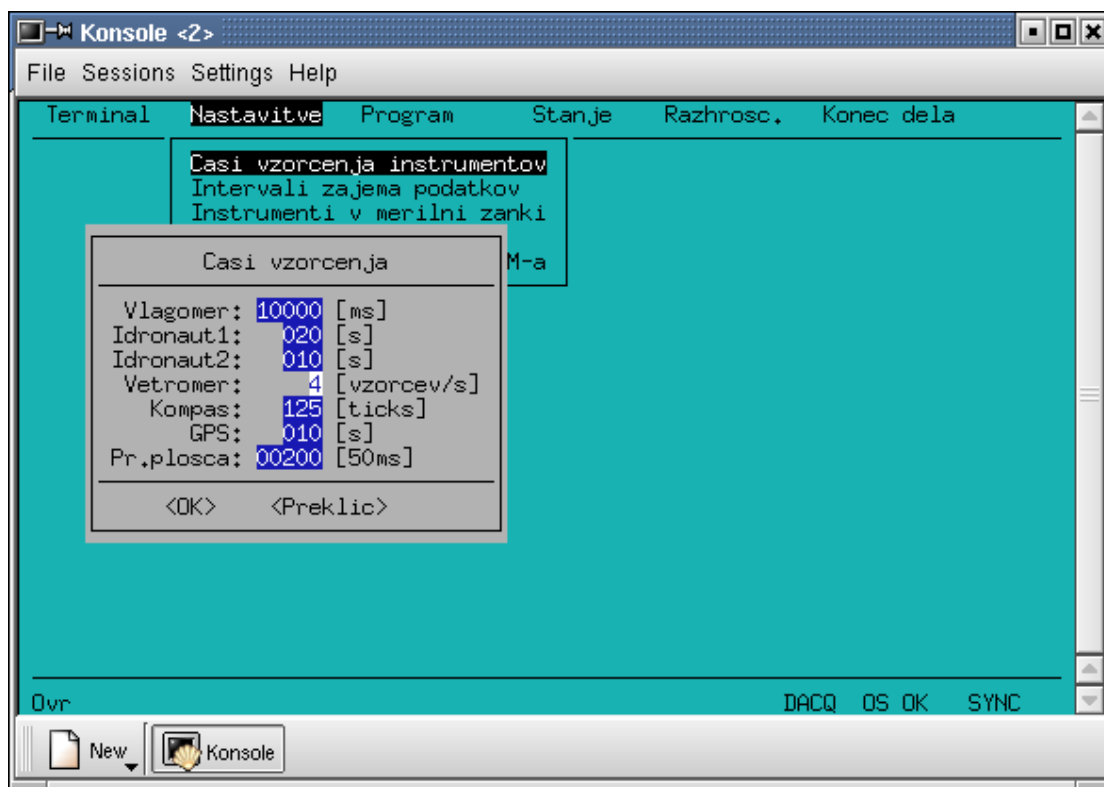


Vir: Šonc, 2003, str. 11

Vnos hitrosti vzorčenja posameznih instrumentov (slika 34) poteka na običajen način. Med vnosi se pomikamo s tipko Enter ali pa s tipkama za pomik dol in gor. Preverjanje pravilnosti vnosa poteka sproti in program vam ne bo dovolil prehoda na naslednji vnos, dokler tekoči ni pravilno vnesen. Pri vsakem instrumentu je podana časovna enota, omejitve pa so pri posameznem instrumentu naslednje:

- hitrost vzorčenja vlagomera je lahko med 500 in 30000 milisekundami,
- hitrost vzorčenja obeh Idronaut sond (slanost, temperatura, tlak) vlagomera je lahko med 1 in 999 sekundami,
- vetromer dopušča le 1 ali 4 vzorce na sekundo,
- hitrost vzorčenja za kompas je podana v internih periodah ure v kompasu (tikih), podamo jo lahko le v vrednostih, deljivih s 25 (0, 25, 50, ... 225, 250), v realni čas pa jih preračunamo po formuli  $\text{čas[ms]} = 150[\text{ms}] + \text{število\_tikov} * 0.8[\text{ms}]$ ,
- hitrost vzorčenja GPS-a je lahko med 1 in 999 sekundami,
- hitrost vzorčenja za meritve na procesorski plošči komunikacijskega multiplekserja (napetost, tok, temperatura, vlaga) pa je lahko med 20 in 60.000 enotami po 50 ms.

Slika 34: Vnos hitrosti vzorčenja za posamezne instrumente



Vir: Šonc, 2003, str. 12

Začetni vnos podatkov poteka v prepisovalnem načinu (novi znak se prepíše čez starega), kar lahko vidimo v levem robu spodnje statusne vrstice glavnega okna, kjer se pokaže napis prepíši - »Ovr«. Če želimo kak novi znak vriniti, s tipko vnesi - »Ins« preklapimo v vstavljalni način in v statusni vrstici glavnega okna se napis »Ovr« zamenja z napisom »Ins«. Ko smo vnesli vse podatke, se osvetli gumb z napisom v redu - »OK«. S tipkami za pomik levo in desno lahko takrat izbiramo med gumboma z napisom »OK« in napisom Preklic. Če pritisnemo potrditveno tipko »Enter«, ko je osvetljen gumb preklic, se vsi vnosi zavržejo. Enak učinek ima pritisk na tipko preklica - »Esc«, ki pa deluje kadarkoli med vnosom. Med vnosom je aktivna še tipka za premike med polji - »Tab«, ki preklaplja med izbiro gumbov (OK, Preklic) in med samim vnosom.

Če smo potrdili gumb »OK« s tipko »Enter«, se na zaslonu pojavi dodatno okno, kjer so izpisane vse spremembe, ki smo jih vnesli. Tu imamo ponovno možnost, da potrdimo vnesene spremembe ali celoten vnos prekličemo. Če smo vnesene spremembe potrdili, začne program pošiljati podatke na bojo in v posebnem oknu izpisuje status vpisa vsake spremembe (OK ali Napaka). Od tega trenutka dalje so podatki zapisani v pomnilniku EPROM na boji in vsa naslednja merjenja bodo

upoštevala nove vnesene čase.

Vnos podatkov pri določanju intervala zajema podatkov poteka na podoben način kot pri nastavitvah hitrosti vzorčenja instrumentov in tudi tu imamo sprotno preverjanje vnosov. Omejitve pri posameznih nastavitvah pa so naslednje:

- čas vzorčenja je lahko med 60 in 499.999 sekundami,
- čas mirovanja je lahko med 60 in 43.140 sekundami,
- skupni čas merilnega cikla (vzorčenje in mirovanje) je lahko med 120 in 43.200 sekundami,
- čas zagona instrumentov, ki predstavlja čas, ki je potreben, da se vsi instrumenti ustrezno nastavijo pred vzorčenjem, je lahko med 20 in 60 sekundami,
- širina startnega okna je lahko 2 in 10 sekundami,
- čas predpriprave je lahko med 2 in 40 sekundami.

Dodaten pogoj za pravilni vnos je še, da mora biti čas mirovanja obvezno krajši od skupnega časa merilnega cikla. Čeprav izgleda nenavadno, je lahko čas vzorčenja precej daljši od skupnega merilnega cikla.

Kako »razume« programska oprema na boji navedene systemske čase? Program na boji preverja realni čas po modulu skupnega časa merilnega cikla v sekundah in glede na druge časovne parametre zažene vzorčenje. Na primer: če je dolžina skupnega merilnega cikla enaka recimo 15 minut (900 sekund), čas mirovanja pa 7 minut (420 sekund), bo program zagnal vzorčenje 7, 22, 37 in 52 minut čez vsako polno uro. Vzorčenje pa bo trajalo toliko časa, kot je določen čas vzorčenja, in to je lahko več kot 8 minut. Seveda pa se vzorčenje v tem primeru ne bo vedno začelo ob prej navedenih časih. Uporabnik mora paziti, da konec vzorčenja ne pade v bližino zagonskih časov vzorčenja, saj bo na ta način moral čakati do naslednjega zagonskega časa. Če nastavimo čas zagona instrumentov na 30 sekund, se bodo vsi instrumenti vključili 30 sekund pred zagonskim časom vzorčenja in v tem času bo program izvedel začetne nastavitve na instrumentih. Novi merilni cikel se bo pričel le, če je program končal prejšnjega dovolj zgodaj, in sicer vsaj toliko časa pred novim zagonskim časom kot znaša čas zagona instrumentov minus širina startnega okna. V tem skrajnem primeru program za zajem časa mirovanja sploh ne upošteva. V praksi moramo merilne cikle določiti z dovolj rezerve, tako da ima program po končanem vzorčenju dovolj časa, da pošlje zbrane podatke na kopno, ker med vzorčenjem pošiljanje podatkov na kopno ne deluje.

Če ima program na boji premalo časa na razpolago za pošiljanje zbranih meritev na kopno, se podatki kopičijo na spominski kartici »CompactFlash« in po določenem času se začnejo stare meritve izgubljati.

Dodatni parameter »čas predpriprave« določa, koliko prej bo program prekinil morebitni nedokončan prenos podatkov, preden se prične novi zagon instrumentov. Morebitni nedokončan prenos podatkov se torej prekine toliko sekund pred novim začetkom vzorčenja, kolikor znaša vsota časa zagona instrumentov in časa predpriprave.

Podatki iz posameznega instrumenta bodo vključeni v avtomatsko zbiranje podatkov le, če je instrument vključen v merilno zanko. Če pa hočemo do posameznega instrumenta dostopati v terminalskem načinu, pa mora biti izključen iz merilne zanke.

Določanje, kateri instrument je v merilni zanki ali izven nje, poteka enostavno, in sicer s tipkama za pomik gor in dol izberemo instrument, izbiro ali neizbiro pa izvedemo s pritiskom na tipko za presledek. Ostali del postopka je enak kot pri vnosu drugih nastavitev.

Nastavitve parametrov komunikacij so dostopne le sistemskemu uporabniku. Tu lahko nastavljamo parametre zaporednih vrat komunikacijskega multiplekserja na boji. Nastavitve prvih vrat oziroma vrat s številko 0 lahko le vidimo, ker predstavljajo nastavitve za komunikacijo z radijskim modemom, in jih nikakor ne smemo spreminjati. Druge nastavitve lahko spreminjamo, smiselno je to le pri priključitvi drugačnega instrumenta ali izjemoma pri spremembi hitrosti komuniciranja obstoječega. Za pravilne nastavitve za vsak instrument uporabljamo ustrezne tovarniške listine.

Postopek za vnos je enak kot pri vnosu drugih nastavitev. Omejitve pri posameznih nastavitvah pa so naslednje:

- dovoljene vrednosti za hitrost so: 0 – vrata neaktivna, 150, 300, 600, 1200, 2400, 4.800, 9.600, 14.400, 19.200, 28.800, 38.400, 57.600 in 115.200,
- število podatkovnih bitov je lahko med 5 in 8,
- parnost je lahko soda – S, liha – L ali brez parnosti – N,
- število stop bitov pa je lahko le 1 ali 2.

Brisanje vseh nastavitev je možno izvesti le sistemskemu uporabniku. Poleg vseh nastavitev se pobriše tudi uporabniški program na boji in sistem se ponastavi (resetira), tako da je potrebno to opcijo uporabljati zelo previdno.

V glavnem meniju programskega sistema Okna je na voljo opcija, ki omogoča nalaganje nove uporabniške programske opreme na komunikacijski multiplekser na boji in je dostopna le sistemskemu uporabniku. Pred samim nalaganjem novega programa moramo na računalnik na Morski biološki postaji Piran prepisati datoteko s

prevedenim programom. V okno, ki se odpre po izbiri opcije »Nalaganje programa«, vpišemo pot do prej prepisane datoteke s programom.

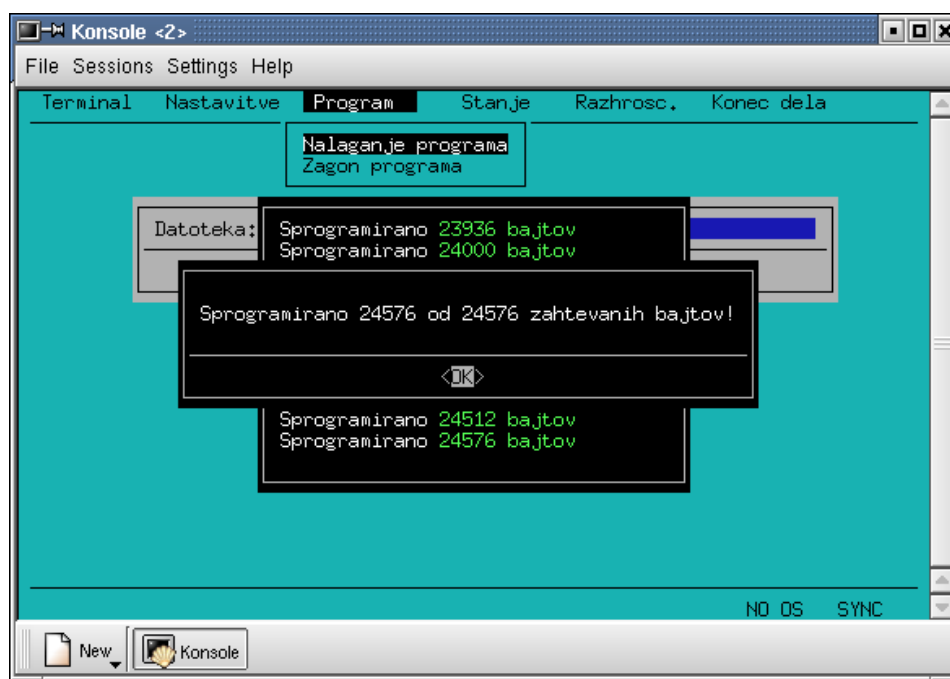
Preko radijske zveze se izbrana programska datoteka prenese v sistem na boji in tam zapiše v zunanji programski FLASH pomnilnik. Sistem se nato samodejno ponastavi, vendar se še ne zažene. Ob uspešnem prenosu se na zaslonu prikaže ustrezno sporočilo. Novi program moramo nato še posebej zagnati z izbiro opcije »Zagon programa«. Tak način zagona je uporabljen zato, da lahko pred samim zagonom uporabnik v terminalskem načinu preizkusi posamezne sklope sistema in instrumente na boji. Po zagonu novega programa se radijska zveza prekine do konca prve periode vzorčenja. V tem času, dokler se ne pojavi napis »SYNC« v spodnji statusni vrstici glavnega okna, ne zaganjamo nobenih opcij programa Okna.

Opcija »Prikaz stanja« glavnega menija omogoča prikaz okna, ki stalno prikazuje čas, ki ga meri ura realnega časa na boji, napetost baterij, trenutno poraba toka, temperaturo in relativno vlago v ohišju komunikacijskega multiplekserja na boji. V oklepaju pod napisom »stanje na boji« je prikazan tudi status uporabniške programske opreme na boji in kjer so možni trije napisi:

- program za zajem zagnan – kar pomeni, da je program za zajem naložen in da deluje,
- program za zajem ni zagnan – kar pomeni, da je program za zajem sicer naložen, da pa ne deluje,
- program za zajem ni naložen – kar pomeni, da je uporabniški program v Flash programskem pomnilniku na boji zbrisan.

Zadnje sporočilo se lahko med normalnim delovanjem pojavi le pri brisanju vseh nastavitev, predzadnje pa po nalaganju in pred zagonom novega programa. Če se predzadnji napis pojavi v katerikoli drugih okoliščinah, pomeni, da se je sistem ponastavil (resetiral) ali zaradi napake v programu, ali zaradi nenadnega padca napetosti baterij, ali katerega drugega vzroka, ki ga je potrebno odpraviti. Če se pa pojavi zadnji napis in če nismo uporabili opcije »Brisanje vseh nastavitev« podmenija »Nastavitve«, lahko sumimo na okvaro elektronike v komunikacijskem multiplekserju.

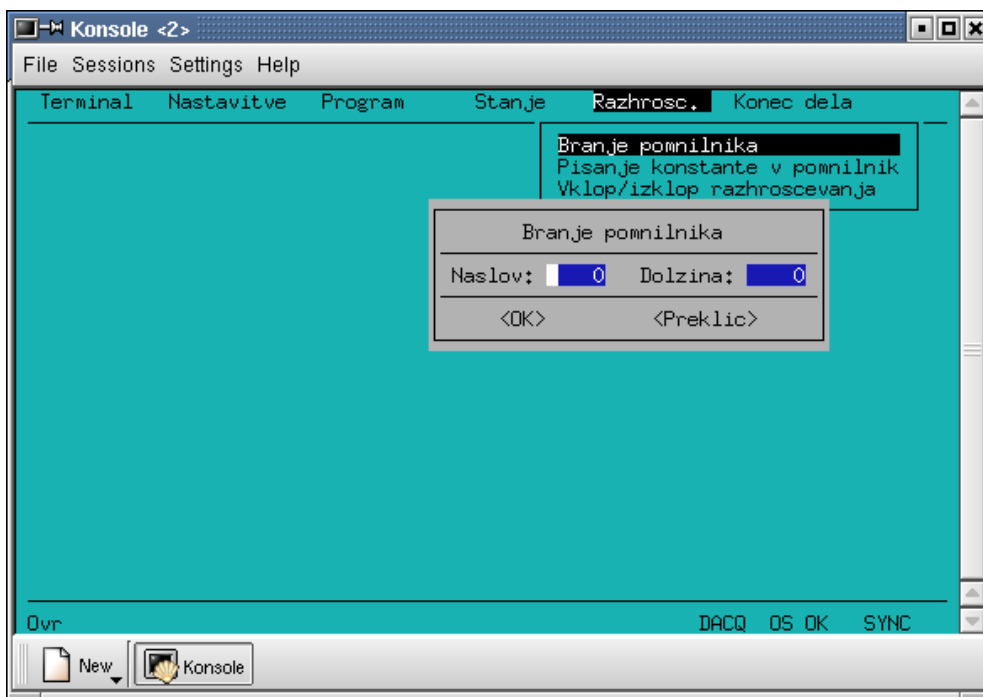
**Slika 35: Statusno sporočilo po uspešnem nalaganju programa**



**Vir: Šonc, 2003, str. 19**

Razhroščevanje je ena od uporabniških funkcij. Običajnemu uporabniku je dostopna le prva opcija tega podmenija (slika 36), ostali dve lahko uporablja le sistemski uporabnik. Prva opcija omogoča branje podatkov iz poljubnega bloka pomnilnika komunikacijskega multiplekserja, druga opcija omogoča vpis enozlogne konstante v poljubni blok delovnega pomnilnika RAM (angl. Random Access Memory) komunikacijskega multiplekserja, tretja opcija pa omogoča lovljenje meritev z napako. Pri prvi opciji vpišemo začetni naslov v pomnilniku in dolžino bloka v zlogih, prebrana vsebina pa se prepíše v datoteko, katere ime določimo v konfiguracijski datoteki. Pri drugi opciji – pisanje konstante v pomnilnik – vpišemo začetni naslov v RAM pomnilniku komunikacijskega multiplekserja, dolžino bloka in vrednost konstante. Pri tretji opciji označimo instrumente, pri katerih želimo loviti meritve z napako. Po določenem času (odloči uporabnik) vse instrumente odznačimo (vsi oglati oklepaji za instrumenti prazni) in takrat se vsebina razhroščevalnega pomnilnika, ki hrani meritve z napako in ki je zelo majhen (255 zlogov), prepíše v datoteko, katere ime določimo v konfiguracijski datoteki. Pred uporabo opcij tega podmenija preberite še tovarniške listine komunikacijskega multiplekserja in izvorno kodo programske opreme v komunikacijskem multiplekserju.

**Slika 36: Branje bloka podatkov iz pomnilnika v komunikacijskem multiplekserju**



Vir: Šonc, 2003, str. 22

#### 6.3.4.2 Program Mbp\_boja

Program deluje kot proces v ozadju (angl. daemon), njegova glavna funkcija pa je odjemalni del komunikacijskega protokola za prenos meritev z boje na kopno. Ob vsaki vzpostavitvi zveze najprej zahteva statusne informacije iz boje, ki jih potem zapiše v datoteko »status.dat«. Hkrati preko semaforja obvesti program Vpis, ki statusne informacije pregleda in vpiše v MySQL bazo podatkov. Zatem zahteva nove merilne podatke od zadnjega prebranega zapisa naprej in jih zapiše v datoteko. Ko so vse meritve prenesene, preko semaforja spet obvesti program Vpis, ki jih nato, tako kot statusne podatke, obdela in zapiše v MySQL bazo podatkov. Če je na računalniku zagnan tudi program Okna, program Mbp\_boja to zazna preko spremenljivk v skupnem pomnilniku in prenaša zahteve programa Okna na bojo in povratne informacije nazaj. S pomočjo sporočil oz. signalov tudi obvešča program Okna, če je zagnan seveda, o bistvenih spremembah, ki se tičejo komunikacij in statusa boje, in te so:

- vzpostavitev zveze (sinhronizacija),
- ali program za zajem podatkov na boji deluje,
- ali je trenutno v teku zajem podatkov na boji in



- ali trenutno teče prenos podatkov na kopno.

Program se zažene avtomatsko ob zagonu računalnika s pomočjo skripte »bcu« na imeniku /etc/init.d (Linux distribucija Debian). Seveda mora sistemski administrator ustrezno povezati »bcu« skripto na ustrezen zagonski nivo.

#### 6.3.4.3 Program Vpis

Program deluje kot proces v ozadju (daemon), njegova glavna funkcija pa je vpis podatkov v MySQL podatkovno bazo. Program večino časa čaka pritajen v pomnilniku in postane aktiven šele, ko od programa Mbp\_boja dobi preko semaforja sporočilo, da so prispeli podatki, ki ji je potrebno vpisati. Program vpisuje le surove podatke v bazo surovih podatkov, katerih struktura je podobna tisti, opisani v poglavju 3.4.1, in ne opravlja nobenih dodatnih obdelav. Te so naloga drugih programov.

Tako kot program Mbp\_boja se tudi program Vpis zažene avtomatsko ob zagonu računalnika s pomočjo skripte.

#### 6.3.4.4 Uporabniški vmesnik za končne uporabnike

Internetni brskalnik uporabljamo kot grafični uporabniški vmesnik za internet. S pomočjo brskalnika lahko uporabniki dostopajo do določene oziroma iskane spletne strani in tako pridejo do želenih informacij.

Pri oblikovanju uporabniškega vmesnika oziroma, v tem primeru, internet strani programa Modra zastava je potrebno biti pozoren na naslednje elemente (Turk, 1. 3. 2003):

- **Enostavnost in konsistentnost.** Osnovni problem uporabnikov je pomanjkanje občutka, kje se trenutno nahajajo v okviru predstavitve nekega podjetja ali produkta. Povezave v hipertekstu lahko kažejo tudi na druge strežnike in predstavitve drugih podjetij. Zato je pomembno, da zagotovimo konsistenten videz strani v neki predstavitvi. Tako bo lahko, če bo videl, da se je videz spremenil, ko je kliknil neko hipertekst povezavo, že na prvi pogled ugotovil, da je pristal na kakšni drugi predstavitvi. Drugi razlog za enostavnost in konsistentnost je večja hitrost sprehajanja po menijih, če so le-ti vedno na istem mestu in podobno oblikovani.
- **Pomoč pri navigaciji.** Uporabniku moramo vedno zagotoviti, da brez težav

pride na matično stran in druge ključne strani predstavitev. Te povezave mu moramo zagotoviti na vsaki strani. Običajno jih oblikovalci dajo na stran kar v obliki gumbov.

- **Uporaba gumbov za prehajanje med stranmi v želenem zaporedju.** Problem vseh hipertekstnih predstavitev je, da (običajno) niso urejene lepo zaporedno kot npr. knjige. Če obiskovalec predstavitev lista z gumbi "Naprej" in "Nazaj" v brskalniku, se velikokrat težko znajde ali pa ne vidi vsebine v takem zaporedju, kot bi mi želeli. Če hočemo vsebino predstavitev urediti v neko logično zaporedje, si lahko pomagamo s tem, da gumbom za direktni dostop do pomembnejših strani dodamo še gumba "prejšnja stran" in "naslednja stran".
- **Slepe ulice.** Za uporabnike posebno slabo delujejo tako imenovane "slepe ulice" (dead-end) strani, ki ne vsebujejo nobene povezave. Takšne strani so najbolj zanesljivo sredstvo za odganjanje obiskovalcev.
- **Hiter dostop do iskanih informacij.** Cilj pri oblikovanju strukture predstavitev je zagotoviti uporabniku čim hitrejšo pot do informacij, ki jih išče. Predstavitev je treba oblikovati v učinkovito hierarhijo informacij, da je število korakov skozi menije čim manjše. Raziskave kažejo, da imajo uporabniki rajši goste menije z manj nivoji kot pa veliko število poenostavljenih menijev. Večina meni, da mora biti na meniju vsaj od pet do sedem povezav.
- **Dialog in feedback.** Spletna predstavitev mora imeti vgrajeno tudi možnost komuniciranja z uporabniki. Dialog je lahko vzpostavljen tudi s pomočjo rednih mesečnih novic, na katere se lahko obiskovalci naročijo.
- **Hitrost nalaganja strani.** Počasnost pri nalaganju izbrane spletne strani je zelo moteč dejavnik pri predstavitvi. Kaj je počasi in kaj je hitro, je sicer subjektivna ocena, raziskave pa kažejo, da je meja nekako pri 10 sekundah. Ker je čas nalaganja v veliki meri odvisen od hitrosti povezav, ki jih imajo uporabniki, je dobro poznati tehnično opremljenost ciljne skupine. Če računamo, da bo veliko uporabnikov dostopalo do strežnika preko navadne telefonske linije, ne kaže pretiravati z grafiko, animacijo in zvokom, saj bodo sicer obiskovalci hitro zapustili predstavitev.

Prav tako je potrebno paziti, da internet strani vsebujejo (Jerman-Blažič, 2001, str. 67-72):

- polno informacij in novosti, ki so pomembne in vzbujajo pozornost zainteresiranih,
- občasne nagradne igre, ki pritegnejo tudi načeloma nezainteresirane,
- pregledne, informativne in grafično privlačne spletne strani, pri katerih pa obiskovalci ne bodo izgubili potrpljenja bodisi s čakanjem ali s »sprehajanjem« po straneh,

- možnost ocene le-te, saj se lahko z mnenji obiskovalcev stran še izboljša in tako še bolj približa ciljni skupini ljudi. To oceno se lahko vključi tudi v nagradno igro in tako dodatno vzpodbudi obiskovalce, da strani ocenijo in predlagajo izboljšave.

Pri oblikovanju internet strani gibanja Modra zastava sem poskusil upoštevati napisana priporočila v čimvečji meri. Na spodnji sliki si lahko ogledamo osnovno stran spletne predstavitve gibanja Modra zastava.

**Slika 37: Prva stran spletne predstavitve gibanja Modra zastava**



**Vir: avtor**

## **6.4 Vzpostavitev baze podatkov za potrebe informacijskega sistema**

Na podlagi zbranih zahtev uporabnika in na podlagi razrednega diagrama začnemo z logičnim in fizičnim načrtovanjem baze podatkov za potrebe informacijskega sistema za potrebe gibanja Modra zastava.

### 6.4.1 Izdelava logičnega načrta

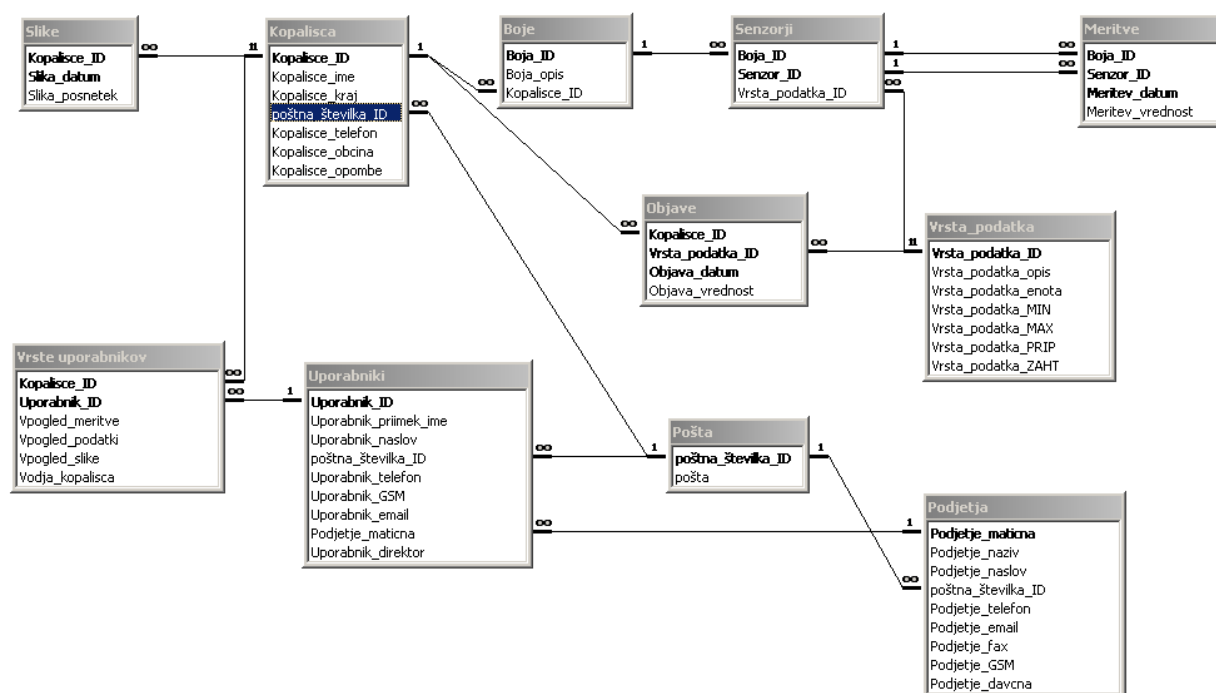
Logični načrt je načrt baze podatkov, ki ga bomo realizirali s pomočjo dveh korakov:

- najprej preslikamo razredni diagram, ki smo ga pripravili v fazi načrtovanja, v relacijski podatkovni model SUBP,
- načrt nato prilagodimo konkretnemu SUBP, saj lahko različni SUBP implementirajo podatkovni model na različne načine, z dodatnimi možnostmi in omejitvami.

S postopkom normalizacije nato preverimo kakovost logičnega načrta in ga po potrebi izboljšamo – odpravimo podvajanje podatkov (Grad et al., 1996, str. 230).

Tabele predstavljajo osnovo vsake relacijske podatkovne baze. Ko podrobno definiramo vse tabele, le-te medsebojno tudi povežemo. Povezave določimo, če so podatki ene tabele povezani s podatki druge tabele.

Slika 38: Prikaz tabel in povezav



Vir: avtor

MySQL Poizvedbe bomo v našem primeru vključili kar v samo internet stran, kjer jih bomo potrebovali, saj nam skriptni jezik PHP to omogoča.

Za spletno predstavitev rezultatov bomo potrebovali sledeče poizvedbe oziroma podatke:

- seznam vseh zadnjih izmerjenih podatkov za izbrano kopališče,
- seznam zadnjega izmerjenega podatka za izbrano kopališče in izbrani parameter,
- seznam vseh podatkov v izbranem obdobju za izbrano kopališče in izbran parameter,
- podobne poizvedbe pa potrebujemo tudi za objavljene podatke.

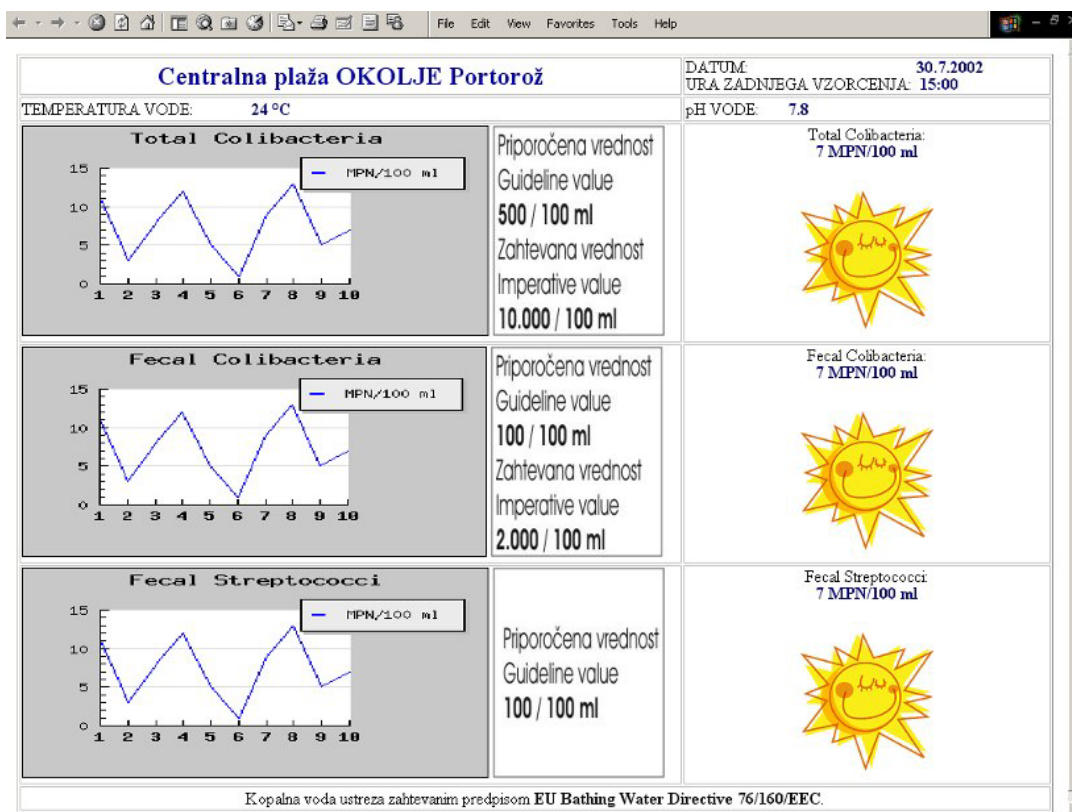
SQL poizvedbe za navedene kriterije prikazujemo v prilogi 13. Zaradi podobnosti poizvedb za objavljene podatke le-teh ne bomo podrobneje prikazali.

Pri določenih poizvedbah moramo paziti tudi na tiste podatke, v našem primeru kislost in bazičnost vode - pH, kjer je zahtevana vrednost omejena tako na zgornjem kot spodnjem področju. V tem primeru jemljemo entiteto »Vrsta\_podatka\_PRIP« za spodnjo mejo podatka, entiteto »Vrsta\_podatka\_ZAHT« pa jemljemo za zgornjo mejo podatka.

Prirejene poizvedbe bomo uporabili tudi pri risanju grafov, kjer bomo uporabili objektno usmerjeno grafično knjižnico za PHP, v našem primeru JpGraph (Aditus, 1. 2. 2003), in določene podatke iz MySQL baze. V prilogi 14 je prikazan osnutek PHP skripte za prikazovanje grafov.

Podobno skripto naredimo še za skupne koliformne bakterije in koliformne bakterije fekalnega izvora, dodamo še druge potrebne podatke in oblikujemo .php spletno stran za izbrano kopališče v skladu z zahtevami uporabnikov, kot je bilo razvidno na sliki 18. Na sliki 39 tako prikazujemo končni vmesnik za prikaz meritev na vhidih v kopališča.

**Slika 39: Prikaz končnega vmesnika za prikaz »on-line« meritev na vhodih v kopališča**



Vir: avtor

## 6.4.2 Izdelava fizičnega načrta

Pri izdelavi fizičnega načrta je predvsem pomembno vedeti, katere podatke in kako pogosto bodo uporabniki dosegali, spreminjali in dodajali le-te v bazo podatkov. Kriteriji pri izdelavi fizičnega načrta so (Grad et al., 1996, str. 230):

- poraba prostora na zunanjih pomnilnikih,
- odzivni čas oz hitrost izvršbe poizvedbe ali transakcije,
- prepustnost, ki pomeni povprečno število transakcij, ki jih lahko izvedemo na časovno enoto.

Izdelava fizičnega načrta je odvisna od izbranega SUBP oziroma kaj na tem področju omogoča. V splošnem se pri izdelavi fizičnega načrta odločamo o tem, na katerih nosilcih podatkov bodo shranjeni podatki in posamezni deli baze podatkov, katere indekse bomo izbrali in katere podatkovne strukture bomo izbrali za doseganje podatkov (Grad et al., 1996, str. 231).

Informacijski sistem za potrebe programa Modra zastava začnemo graditi s postavitvijo strežnika in lociranjem boj s pripadajočo opremo. Strežnik za potrebe

obravnavanega informacijskega sistema je lahko tudi obstoječi strežnik na Morski biološki postaji Piran ali pa lahko v istih prostorih postavimo novi podatkovni strežnik, internet strežnik pa delimo skupaj z že obstoječim. Na tak način zmanjšamo investicijske in vzdrževalne stroške. Zaradi podobnosti sistemov je tudi strošek izobraževanja manjši. Opis strežnika in programske opreme smo podrobneje predstavili v poglavjih 6.3.2. in 6.3.3.

Vsak računalnik, ki je neposredno vključen v internet, ima tudi svoj naslov. Poleg številčnega naslova ali naslova IP obstaja tudi njegova mnemonična preslikava (Jerman-Blažič, 1996, str. 19-21). Društvo za okoljevarstveno vzgojo Evrope v Sloveniji je že vložilo prošnjo za registracijo pod vrhno domeno .si in tudi registriralo svoje ime pod vrhno domeno .si, in sicer pod imenom *www.drustvo-doves.si*.

Najprej ročno vnesemo v bazo podatkov vse podatke o kopališčih, občinah, odgovornih osebah, vzorčnih mestih in senzorjih. Pri vnašanju pazimo na definicijo ključev v posameznih tabelah. Ključne zato določimo na sledeči način:

- ***kopalisce\_ID*** – vsako kopališče ima obratovalno dovoljenje in je zato le-to primerno za glavni ključ;
- ***boja\_ID*** – boje enostavno označujemo po vrstnem redu implementacije, torej od 1 do n;
- ***senzor\_ID*** – glavni ključ za označevanje senzorja sestavimo iz številke 1, ki ji nato pripišemo dvomestno številko (za potrebe slovenskega gibanja Modra zastava), ki označuje številko boje, nato dodamo še dvomestno številko, ki označuje tip senzorja, ter nazadnje še številko, ki jo uporabljamo v primeru, da imamo na boji dva enaka senzorja;
- ***vrsta\_podatka\_ID*** – glavni ključ v tem primeru zapišemo podobno kot ključ pri bojah, torej zaporedno številko od 1 do m;
- ***podjetje\_maticna*** – kot že samo ime pove, tukaj uporabljamo za ključ matično številko podjetja, ki kopališče upravlja;
- ***uporabnik\_ID*** – uporabnika enolično določa EMŠO, enotna matična številka občana;
- ***postna\_stevilka\_ID*** – kot že samo ime pove, za poštne številke uporabljamo uradni poštni seznam Pošte Slovenije.

Najboljši termin za začetek spremljanja kakovosti kopalne vode bi bilo obdobje pred začetkom kopalne sezone, saj bi tako lahko odpravili potrebo po ročnem vnašanju podatkov merjenj kopalne vode in bi od te točke dalje tudi ohranjali časovno vrsto in frekvenco vzorčenja podatkov. V skladu s tem datumom prilagodimo tudi vse aktivnosti v skladu z izgradnjo informacijskega sistema.

Po zagonu sistema pazimo tudi na redno zaščito podatkov pred izgubo ali poškodovanjem. Podatke po končani kopalni sezoni zapišemo tudi na zgoščenko, ki jo nato hranimo v fizično varovanem arhivu.

Bazi podatkov postavimo tudi programske zaščite, ki bodo onemogočile zlorabo podatkovne baze. Vsakemu uporabniku, ki ima pravico do uporabe elementov podatkovne baze, dodelimo uporabniško ime, štirimestno identifikacijsko številko in šifro za vstop v sistem. Glede na pravice, ki jih imajo uporabniki pri delu z bazo podatkov, ločimo različne skupine uporabnikov (Zdovc, 1993, str. 187):

- uporabniki (angl. users), ki imajo dostop do omejenega dela baze podatkov,
- sistemski uporabniki (angl. admins), ki imajo dostop do celotne baze podatkov, in
- gostje (angl. guests), ki imajo dostop do javnih podatkov.

Za skupino uporabnikov in gostov lahko nastavimo omejitve za vsak objekt podatkovne baze (npr. za tabelo, za poročilo itd.). Izberemo lahko med več oblikami zaščit. Prav tako lahko za vsako tabelo omejimo prikaz določenih polj le-te. Načeloma pa tudi velja, da se z večanjem števila vrst uporabnikov otežuje administriranje same baze podatkov. Večkrat je zato bolje opredeliti samo ene vrste uporabnikov, na primer uporabniki, ki pa imajo različne pravice, saj je tako administriranje lažje.

#### **6.4.3 Gradnja in vzdrževanje baze podatkov**

Implementacija oziroma gradnja baze podatkov predstavlja realizacijo logičnega in fizičnega načrta in vnos začetnih podatkov iz realnega sveta v fizično bazo podatkov. Razlikovati moramo dve fazi (Jaklič, 2002):

- definiranje strukture:
  - vnesemo strukturo, ki smo jo predvideli z logičnim načrtom,
  - definiramo zunanje poglede uporabnikov,
  - definiramo notranji model,
- vnos začetnih podatkov:
  - vnesemo podatke ali jih pretvorimo iz stare oblike v novo.

Ko realiziramo vse navedene faze, začnemo s fazo testiranja sistema. Na podlagi ugotovljenih napak le-te tudi ustrezno popravimo.

Informacijski sistem je živ organizem, ki se mora spreminjati in prilagajati. Nove zahteve (na primer spremembe zakonodaje, dodatne zahteve uporabnikov idr.) zahtevajo nove verzije programske opreme. Še več sprememb in izboljšav je lastnih razvoju in napredku informacijskega sistema, ki ga dela vedno bolj uporabnega, stabilnega in hitrega. Hkrati pa se spreminja tudi tehnologija, ki je podlaga informacijskem sistemu.



Zaradi vsega omenjenega je potrebno nadomeščati programe z novimi verzijami letih. Enako kot pri drugih sredstvih je tudi pri informacijski opremi po nakupu nujno vzdrževanje, ki ga delimo v dve kategoriji:

- vzdrževanje aplikacije (osveževanje), kjer nadomeščamo obstoječe verzije programov z novimi, popravljenimi, in
- vzdrževanje podatkov (vzdrževanje), kjer popravljamo, spreminjamo, analiziramo podatke, ki jih programi ali poizvedbe kreirajo.

## 6.5 Ekonomski vidik izgradnje informacijskega sistema

Okvirne stroške investicije postavitve ene boje z merilnimi instrumenti prikazujemo v Tabeli 6.

**Tabela 6: Ocena stroškov investicije boje z merilnimi instrumenti**

|  |                      |
|--|----------------------|
| 1. boja s pripadajočo opremo (solarne celice, nosilci)   | 13.000,00 EUR        |
| 2. oprema za sidranje boje (balastna veriga, vrv, sidro) | 400,00 EUR           |
| 3. senzorji po opisu                                     | 15.000,00 EUR        |
| 4. oprema za »on-line« meritve na boji                   | 2.500,00 EUR         |
| 5. oprema za GSM prenos podatkov boja-center             | 4.000,00 EUR         |
| <b>SKUPAJ:</b>   | <b>34.900,00 EUR</b> |

Vir: Aanderaa, 2003

Kot lahko vidimo, so stroški nakupa boje s pripadajočimi merilnimi napravami ocenjeni na 34.900,00 EUR. Tem stroškom je potrebno dodati še stroške nakupa in postavitve strežniške opreme, izdelave internet aplikacije in sprožilnega programa, ki osvežuje bazo podatkov in pripravlja določene poizvedbe po zahtevah naročnika, ki jih na podlagi razgovorov ocenjujem na okvirno 5.000,00 EUR.

V primeru, da bi za informacijski sistem za potrebe programa Modra zastava uporabljali obstoječi sistem na Morski biološki postaji Piran, bi del stroškov investicije nadomestili stroški najema prostora in opreme.

Iz tega lahko ocenimo, da celotna investicija postavitve (na začetku) dvajsetih boj znaša okvirno 700.000,00 EUR. Seveda bi z nakupom večjega števila boj in ob morebitni zainteresiranosti tudi drugih držav za ta projekt ta strošek ustrezno zmanjšali.

Tem investicijskim stroškom je seveda potrebno nato dodati še vzdrževalne stroške.

Po podatkih EuroGOOS-a stane vzdrževanje, ki vključuje kalibracijo instrumentov, stroške vzdrževalnih operacij in certificiranje podatkov ene boje od 15.000 do 80.000 EUR letno (Legrand, et. al., 2002), in sicer je ta strošek pretežno odvisen od:

- lokacije boje (odprto morje, priobalna, prostoplavajoča),
- števila ljudi in tehnične opreme, potrebne za dostop do boje, in vsa vzdrževalna dela,
- velikosti plovila, če je le-to sploh potrebno, za dostop do boje in
- lastnosti in število instrumentov in senzorjev v samem omrežju.

Ker gre v našem primeru za relativno dostopno bojo, ki ne potrebuje posebnih plovil in velikega števila ljudi, lahko ocenimo, da so stroški vzdrževanja celotnega sistema največ 300.000,00 EUR letno in vključujejo tako kalibracijo senzorjev s strani pooblaščenih ustanov, vzdrževanje boje, baze podatkov in programske ter strojne opreme.

V obstoječem sistemu vzorčenja je okvirna cena enega vzorčenja okvirno 300,00 EUR, kar pomeni, da mora upravitelj kopališča glede na kriterije programa Modra zastava letno v povprečju nameniti za vzorčenje kopalne vode v poletni kopalni sezoni okvirno 4.000,00 EUR. Ta cena se namreč določa na podlagi pogodbe med upraviteljem kopališča in Zavodom za zdravstveno varstvo. Tukaj moramo poudariti, da je v večini držav EU plačilo analiz vzorčenja kopalne vode bodisi na občinski bodisi na državni strani.

Iz povedanega sledi, da obstoječi način financiranja ne bi zadostoval niti za pokritje variabilnih letnih obratovalnih stroškov. Prav zato je sistem naravnan tako, da lahko služi tudi drugim subjektom, ki niso povezani neposredno s programom Modra zastava. Sistem je izrednega pomena tudi za preprečevanje morebitnih onesnaženj kopalne vode, saj omogoča takojšnje informiranje o morebitnih onesnaženjih in takojšnje ukrepanje pristojnih organov. Prav tako pa je sistem izrednega pomena za vse, ki se ukvarjajo z lastnostmi morja in napovedovanjem posameznih pojavov, kot so cvetenje morja, kroženje vode ipd.

## 7. Sklep

Delovanje zanesljivih in učinkovitih sistemov, sposobnih kontrolirati vodne vire na načine, ki bodo omogočali takojšnje odgovore na probleme onesnaženja okolja, razsipavanja z energijo ali povečanja količine produkcije, nam bo povrnilo zaupanje v prihodnost, v kateri je skrb za naravne vire povsem usklajena z ekonomskim in človeškim razvojem. Govorimo o trajnostnem razvoju, o razvoju, ki omogoča zadostitev potreb sedanjih generacij, ne da bi bile ob tem ogrožene možnosti prihodnjih generacij za zadostitev njihovih potreb, predvsem tistih, ki se dotikajo potreb po čistem in lepem okolju.

Program Modra zastava želi ljudem ozavestiti okoljevarstveno delovanje in to dosega s pomočjo različnih okoljevarstvenih aktivnosti, v katere ljudi tudi vključuje. Ena izmed zahtev samega programa Modra zastava je tudi kakovost kopalne vode. Prav kontinuirano spremljanje kopalne vode je lahko eden izmed pokazateljev učinkovitosti vseh okoljevarstvenih gibanj in posameznih nacionalnih prizadevanj v smeri trajnostnega razvoja.

Zaradi tega, ker samo spremljanje brez ukrepanja ni dovolj, je pomembno, da so sistemi kontinuiranega spremljanja tako učinkoviti in hitri, da omogočajo pristojnim organom takojšnji vpogled v stanje kakovosti kopalne vode in takojšnje reagiranje v primeru nenadnega dviga enega izmed kvalitativnih parametrov kakovosti vode. To je mogoče doseči izključno z merilnimi napravami, ki delujejo s trajno komunikacijsko povezavo in sistemi na kopnem.

V pričujoči magistrski nalogi sem zato predstavil izgradnjo informacijskega sistema za potrebe programa Modra zastava ter analiziral obstoječe sorodne informacijske sisteme. Ugotovil sem, da so investicijski stroški samega sistema izredno visoki, z razvojem merilnih tehnik pa lahko pričakujemo pocenitev tovrstne tehnologije. Zaradi tega trenutno, z ekonomskega stališča, ni moč upravičiti naložbe v izgradnjo tovrstnega sistema. Kljub temu menim, da je prihodnost merjenja kakovosti kopalne vode prav v podobno zasnovanih sistemih, ki omogočajo kontinuirano merjenje vseh pojavov v sami kopalni vodi.

## Literatura

- [ 1 ] Avison D. E., Fitzgerald G.: Information Systems Development: Methodologies, Techniques and Tools, 2nd ed. London: McGraw-Hill, 1996. 505 str.
- [ 2 ] Blanc Miloš: Razvoj informacijskega sistema logistike z uporabo metodologije tabelarni razvoj aplikacij (TAD). Magistrsko delo. Ljubljana: Ekonomska fakulteta, 2001. 102 str.
- [ 3 ] Booch Grady, Rumbaugh James, Jacobson Ivar: The Unified Modeling Language User Guide. Reading: Addison Wesley Longman, Inc. 1999. 482 str.
- [ 4 ] Clegg Dei, Barker Richard: CASE Method Fast Track, a RAD approach, Wokingam, England (etc.), Addison Wesley, 1994. 207 str.
- [ 5 ] Colijn Franciscus, Petersen Wilhelm: Collecting oceanography data from ferries. Geesthacht: GKSS Research Centre - Institute for Coastal Research. [URL: <http://www.ices.dk/marineworld/ferries.asp>], 15. 2. 2003.
- [ 6 ] Črv, Milan: Objektni pristop k prenovi poslovnih procesov in izgradnji informacijskega sistema - metodološki vidiki. Doktorska disertacija. Ljubljana: EF, 2000. 202 str.
- [ 7 ] Deželjin Damir: Obalna oceanografska postaja Piran kot del sistema varne obalne plovbe. Diplomaska naloga. Piran: Fakulteta za pomorstvo in promet, 2001. 52 str.
- [ 8 ] Fowler Martin, Scott Kendall: UML Distilled, Second Edition. New York: Addison Wesley Longman, 2000. 186 str.
- [ 9 ] Galliers R.D., Leidner D.E. and Baker B.S.H.: Strategic Information Management. Second Edition. B.k.: Butterworth Heinemann, 1999. 590 str.
- [ 10 ] Grad Janez, Jakli\_ Jurij: Baze podatkov. Ljubljana: EF, 1996. 260 str.
- [ 11 ] Grad Janez, Jaklič Jure: Baze podatkov. Ljubljana: Ekonomska fakulteta, 1996. 253 str.

- [ 12 ] Gradišar Miro, Resinovič Gortan: Informatika v poslovnem okolju. Ljubljana: Ekonomska fakulteta, 2001. 508 str.
- [ 13 ] Jacobson Ivar, Ericsson Maria, Jacobsson Agneta: The Object Advantage: Business process reengineering with object technology. New York: ACM Press, 1995, 347 str.
- [ 14 ] Jaklič Jure: Načrtovanje baz podatkov. Prosojnice predavanj. Ljubljana: Ekonomska fakulteta, 2002. 124 str.
- [ 15 ] Jerman-Blažič Borka, Klobučar Tomaž, Perše Zoran, Nedeljkovič Dragan: Elektronsko poslovanje na internetu. Ljubljana: Gospodarski vestnik, 2001. 205 str.
- [ 16 ] Jerman-Blažič Borka: Internet. Ljubljana: Novi Forum, 1996. 87 str.
- [ 17 ] Kovačič Andrej, Vintar Mirko: Načrtovanje in gradnja informacijskih sistemov. Ljubljana: Dzs, 1994. 316 str.
- [ 18 ] Legrand Jacques, Goasguen Gérard, Ribotti Alberto, Tziavos Christos, Bozzano Roberto, Lindh Henrik, Rodrigues Ignacio, Alvarez Fanjul Enrique: Monitoring the marine environment Operational practices in Europe - a survey of operational practices in use for the running of environmental monitoring installations in Europe. Atene: 3<sup>rd</sup> EuroGOOS – The European Global Ocean Observing System - Conference. [URL:<http://www.ifremer.fr/dtmsi/publications/presentation/monitoring.pdf>], 20. 12. 2002.
- [ 19 ] Murch Richard: Project Management: Best practices for IT professionals. Upper Saddle River: Prentice Hall, 2001. 247 str.
- [ 20 ] Odell James J.: Advanced Object-Oriented Analysis and Design Using UML. Cambridge: Cambridge university press, 1999. 246 str.
- [ 21 ] Petelin Boris, Malačič Vlado: Kontrola kakovosti podatkov za oceanografsko bojo. Piran: Morska biološka postaja Piran, 2001. 15 str.
- [ 22 ] Shelly B. Gary, Cashman J. Thomas, Rosenblatt J. Harry: System Analysis and Design – Fourth Edition. Boston: Course Technology – Thomson

Learning, 2001. 28 cm.

- [ 23 ] Silič Marin, Colnar Marko, Krisper Marjan, Rupnik Rok, Bajec Marko, Rozman Ivan, Heričko Marjan, Domanjko Tomaž, Jurič B. Matjaž, Živkovič Aleš, Beloglavec Simon, Kožman Mitja, Novakovič Aleksander, Stantič Mitja, Rubin Samo, Tomažič Roman, Jensterle Rado: Enotna metodologija razvoja informacijskih sistemov, 4. zvezek – Objektni razvoj informacijskih sistemov. Ljubljana: Vlada Republike Slovenije, Center vlade Republike Slovenije za informatiko, 2000. 198 str.
- [ 24 ] Stuetz Richard: Using sensor arrays for on-line monitoring of water and wastewater quality. Cranfield University. [URL: <http://www.iscpubs.com/articles/aln/n0101stu.pdf>], 1. 4. 2003.
- [ 25 ] Šonc Janez: Navodila za uporabo programske opreme na sistemu za zajem oceanografskih podatkov. Piran: Nacionalni inštitut za biologijo, 2003. 33 str.
- [ 26 ] Taylor David A.: Object-Oriented Information Systems: Planning and Implementation. New York: John Wiley & Sons, Inc., 1992. 357 str.
- [ 27 ] Turk Roman: Oblikovanje uporabniškega vmesnika pri spletnih predstavitev. Ljubljana: Gimnazija Vič. [URL: [http://gimvic.org/gradiva/pouk/navodila-strani/Uporabniski\\_vmesnik.html](http://gimvic.org/gradiva/pouk/navodila-strani/Uporabniski_vmesnik.html)], 1. 3. 2003.
- [ 28 ] Warren J. Donald Jr., Edelson Lynn W., Parker Xenia Ley: Handbook of IT Auditing. 1998 Edition. Warren, Gorham & Lamont, 1998. 1127 str.
- [ 29 ] Zdovc Andrej: Microsoft Access. Nova Gorica: Maya, 1993. 212 str.
- [ 30 ] Živkovič Aleš, Heričko Marjan, Rozman Ivan: Razvoj programske opreme z uporabo jezika UML. Uporabna informatika, številka 4, 2000, str. 200-207.

## Viri

- [ 1 ] Aanderaa Instruments. [URL: <http://www.aanderaa.com/>], 25. 3. 2003.
- [ 2 ] Aandera Instruments: Product catalog – Preferred products. Nesttun: Aanderaa Instruments, 2003. 31 str.
- [ 3 ] Aditus. JpGraph – OO Graph Library for PHP. [URL: <http://www.aditus.nu/jpgraph/index.php>], 1. 2. 2003.
- [ 4 ] CEFAS - Monitoring the health of our seas: SmartBuoy. [URL: <http://www.cefadirect.co.uk/monitoring/default.asp>], 26. 1. 2003.
- [ 5 ] Cetmef: Candhis – Centre d«Arhivage National de Donnees de Houles In-Situ. [URL: <http://www.equipement.gouv.fr/cetmef/candhis/>], 18. 1. 2003.
- [ 6 ] EGOS – European Group on Ocean Stations: [URL: <http://www.meteo.shom.fr/egos/>], 1. 2. 2003.
- [ 7 ] FEE International : Guidance Notes To The European Blue Flag Beach Criteria. Kopenhagen. Kopenhagen: International Blue Flag Co-ordination. [URL: <http://www.blueflag.org/pdf/BeachGuidanceNotes2003.pdf>], 15. 1. 2003.
- [ 8 ] GNU«s Not Unix: GPL – »General Public License«. [URL: <http://www.gnu.org/>], 5. 2. 2003.
- [ 9 ] Ifremer: Le systeme Marel. Plouzané: Ifremer. [URL: <http://www.ifremer.fr/dtmsi/programmes/marel/marel.htm>], 20. 1. 2003.
- [ 10 ] Jaklič Jure: Baze podatkov – modeliranje podatkov, načrtovanje in izgradnja BP, tehnologija, trendi in novosti – povzetki predavanj. Ljubljana: Ekonomska fakulteta, 1999.
- [ 11 ] Lam Josie: Object-Oriented Technology. [URL: <http://disc.cba.uh.edu/~rhirsch/spring97/lam1/hope.htm>], 25.12.2001.
- [ 12 ] Martedec S.A. - The Marine Technology Development Company: The Mediterranean Moored Multi-sensor Array (M3A). Athens: Eant. [URL:

- <http://www.martedec.gr/m3a.htm>], 20. 1. 2003.
- [ 13 ] MedGOOS – The Mediterranean Global Ocean Observing System: Project MAMA. [URL: <http://www.capemalta.net/medgoos/index.html>], 21. 1. 2003.
  - [ 14 ] Nacionalni inštitut za biologijo – Morska biološka postaja Piran. [URL: <http://buoy.mbss.org>], 1.12.2003.
  - [ 15 ] National Center for Marine Research – Institute of Oceanography Athens: Poseidon System. [URL: <http://www.poseidon.ncmr.gr/>], 23. 1. 2003.
  - [ 16 ] Puertas del Estados. [URL: <http://www.puertos.es/Rayo/indexeng.html> ], 5. 2. 2003.
  - [ 17 ] SeaNet: Monitoring Networks on the North Sea. [URL: <http://www.minvenw.nl/rws/projects/seanet/>], 17. 1. 2003.
  - [ 18 ] SMHI - Swedish Meteorological and Hydrological Institute. [URL: <http://www.smhi.se/en/index.htm>], 10. 2. 2003.
  - [ 19 ] Southampton Oceanography Centre: FerryBox Project. [URL: [http://www.soc.soton.ac.uk/ops/ferrybox\\_index.php](http://www.soc.soton.ac.uk/ops/ferrybox_index.php)], University of Southampton Waterfront Campus, 15. 2. 2003.
  - [ 20 ] The Blue Flag Campaign. [URL: <http://www.blueflag.org>], 10. 1. 2003.
  - [ 21 ] Tihomir Makovec: fotografije in risbe boje MBP Piran. 2001.
  - [ 22 ] URSAT. [URL: <http://www.ursat.com/ursat/boya/english/page2.html>], 20. 6. 2003



## Slovarček tujih izrazov

**ARQ** (angl. Automatic Repeat Request) - protokol, ki za vsak poslan paket pričakuje potrditev o prejemu paketa.

**A/D pretvornik** (angl. Analog / Digital converter) – pretvori trenutno vrednost analogne količine v digitalno.

**ADP / ADCP** (angl. Acoustic Doppler [Current] Profiler) – akustični tokomer, ki s Dopplerjevim principom meri tokove po vertikalnem profilu vodnega stolpa.

**ADP / NDP** (angl. Acoustic Doppler Profiler / Nortek Doppler Profiler) - akustični Dopplerjev merilec tokov.

**ANSI C** – standardni nabor ukazov, ki ga razume ANSI C kompatibilen prevajalnik.

**ASCII** (angl. American Standard Code for Information Interchange) – standardni nabor znakov, ki se uporablja za računalniško izmenjavo podatkov.

**BIOS** (angl. Basic Input/Output System) – program največkrat vgrajen v ROM, ki določa kako računalnik prepozna posamezne periferne opreme in zaganja programe.

**BCU** (angl. Buoy Control Unit) – bojina kontrolna enota.

**C++** - visokonivojski programski jezik, ki ga je razvil Bjarne Stroustrup (Bell Labs).

**CASE** (angl. Computer Aided Software Engineering) – računalniško podprta orodja za gradnjo informacijskih sistemov.

**CMOS** (angl. Complementary Metal Oxide Semiconductor) – stabilna pomnilniška tehnologija z nizko porabo.

**CVS** (angl. Concurrent Version System) - sistem za nadzor revizije kode, ki omogoča pooblaščenemu uporabniku dostop do poljubne prejšnje verzije izbrane datoteke, ki je shranjena v CVS.

**DGPS** (angl. Differential Global Positioning System) – izboljša pozicijo, ki bi jo dobil navaden GPS sprejemnik, s korekcijskim signalom, ki ga običajno dobi iz obalnega radijskega svetilnika.

**Distribucija** – skupek programov, ki pokrivajo široko področje uporabe. Ponavadi govorimo o distribucijah pri programih oz. operacijskih sistemih, ki so dostopni pod pogoji GPL-licence.

**DRU** (angl. Data Receiving Unit) – postaja za sprejem podatkov.

**EPC** (angl. Embedded PC) – namenski računalnik (uporabljen na boji in v radijski sprejemni postaji).

**EPROM** (angl. Erasable Programmable Read Only Memory) - brisljivo programibilni bralni pomnilnik.

**Ethernet UTP**– računalniško omrežje, na katerega je vsak računalnik priključen s svojim UTP (angl. Unshield Twisted Pair) kablom in tako povezan na aktivno mrežno opremo.

**EU** (angl. European Union) - Evropska unija.

**FEE** (angl. Foundation for Environmental Education) – Fondacija za okoljsko vzgojo.

**Flash RAM** (angl. Flash Random Access Memory) – je pomnilniški medij, katerega jedro so RAM-čipi in katerega vsebina se ne izbriše kljub prekinitvi električnega toka.

**GIS** (angl. Global Information System) – Globalni informacijski sistem.

**GOOS** (angl. Global Ocean Observing System) – Globalni oceanografski opazovalni sistem v okvirih katerega delujejo tudi MAMA, EuroGOOS, MedGOOS in drugi sistemi opazovanja morja, ki določa okvire delovanja posameznih članic tega

sistema.

**GPL** (angl. GNU Public Licence) – programska oprema, ki je izdana pod pogoji te licence, se lahko brezplačno uporablja (več na <http://www.gnu.org/>).

**GPS** (angl. Global Position System) – sistem za določanje pozicije s pomočjo geostacionarnih satelitov.

**GSM** (angl. Global System for Mobile Communication) - celularni sistem mobilne telefonije, ki deluje v določenih frekvenčnih območjih 900, 1800 in 1900 Hz.

**HTML** (angl. HyperText Markup Language) - jezik, ki je namenjen izdelavi spletnih strani, tj. povezavi z drugimi dokumenti, prikazovanju in oblikovanju podatkov.

**IDE** (angl. Intelligent Drive Electronics ali Integrated Drive Electronics) – vmesnik za trde diske pri katerih je kontroler vgrajen v sam disk.

**IOC** (angl. Intergovernmental Oceanographic Commission) - medvladna oceanografska komisija.

**IP-številka** – številka, ki je dodeljena samo enemu računalniku v računalniškem omrežju s TCP / IP.

**Java** – programski jezik. Ko kodo v Javi prevedemo, dobimo t.i. bycode, torej program, ki je neodvisen od platforme. Bycode program na ciljnem računalniku poženemo v virtualnem stroju JVM (angl. Java Virtual Machine).

**LVM** (angl. Logical Volume Manager) – dinamično upravljanje z diski.

**MAMA** (angl. Mediterranean Global Ocean Observing System) – Globalni opazovalni sistem mediteranskih držav, ki deluje v okvirjih GOOS-a.

**MPN** (angl. Most Probable Number) – najverjetnejše število, nam pove, koliko bakterij ali streptokokov je v 100 ml ali 1 litru kopalne vode.

**MS-DOS** (angl. Microsoft Disk Operating System) – operacijski sistem.

**MySQL** – SQL relacijska baza podatkov.

**NMEA protokol** – Protokol podatkov, namenjen izmenjavi podatkov med navigacijskimi instrumenti.

**OS** (angl. Operating System) – operacijski sistem. Osnovni program, ki nam omogoča delo z računalnikom oz. komunikacijo z njegovimi vhodno / izhodnimi enotami.

**PC** (angl. Personal Computer) – osebni računalnik.

**pH** (angl. Potential Hydrogen) – merilo za kislost ali bazičnost razstopin.

**PHARE CBC** (angl. Poland and Hungary: Assistance to the Restructuring of the Economy Cross-Border Co-operation) - Phare program prekomejnega sodelovanja. Program Phare je pobuda Evropske unije, ki zagoravlja državam srednje in vzhodne Evrope finančno pomoč za podporo procesa ekonomske preobrazbe, hkrati pa jim pomaga okrepiti njihove novooblikovane demokratične družbe.

**PHP** (angl. Hypertext Preprocessor) - je skriptni jezik, ki ga vključimo v opisni jezik HTML, kar nam omogoči izdelavo dinamičnih spletnih strani, ki so osnova za široko izbiro storitev, ki jih danes najdemo v spletu.

**RAD** (angl. Rapid Application Development) – programsko okolje za hiter razvoj aplikacij. Ponavadi opravimo večji del naloge s samim vlečenjem komponent (gumbkov) z miško iz menijev v okno. S spreminjanjem lastnosti komponent v t.i. nadzorniku lastnosti prikažemo videz programa.

**RAID** (angl. Redundant Array of Independent Disks) - skupina neodvisnih diskov s preobiljem podatkov. Pri tem pomeni redundanca ali preobilje večkrat zapisane enake podatke ali osnovnim podatkom pridružene nadzorne podatke, ki so iz njih izračunani po določenem algoritmu, tako da je možno odkrivanje in odpravljanje

napak.

**Real time OS** – je operacijski sistem v katerem je zagotovljeno, da se bo določen dogodek začel točno ob določenem času.

**SMS** (angl. Short Message Service) – servis, ki omogoča pošiljanje kratkih sporočil med uporabniki mobilnega telefona.

**SQL** (angl. Structured Query Language) – nabor standardnih ukazov za delo s podatkovnimi bazami SQL.

**TCP / IP** (angl. Transmission Control Protocol / Internet Protocol) – protokoli, ki se uporablja v večini računalniških omrežij in na internetu za prenos in kontrolo prenosa podatkov.

**UART** (angl. Universal Asynchronous Receiver Transmitter) - je naprava, ki po več žicah vzporedno prihajajoče podatke pretvori v en sam tok zaporedno odhajajočih podatkov, ki jih vodimo v modem.

**UHF** (angl. Ultrahigh Frequency) – radijski valovi v frekvenčnem območju od 300 MHz do 3 GHz.

**UML** (angl. Unified Modelling language) – poenoten jezik za modeliranje.

**UNEP** (angl. United Nations Environmental Program) – Okoljski program Združenih narodov.

**UNESCO** (angl. United Nations Educational, Scientific and Cultural Organizations) – Organizacija združenih narodov za izobraževanje, znanost in kulturo.

**UTP** (angl. Unshielded Twisted Pair) – vrsta kabla, ki se uporablja predvsem za povezave v lokalnih omrežjih.

**VHF** (angl. Very High Frequency) – gre za frekvenco, ki ima domet okoli 30 navtičnih milj in deluje v frekvenčnem območju od 156 do 176 MHz.

**VTMS** (angl. Vessel Traffic Management System) – sistem za nadzor in upravljanje plovbe. Vsebuje VTS-sistem in omogoča povezovanje z organizacijami, ki so vključene v pomorsko dejavnost za povečanje ekonomičnosti oz. izkoriščenosti razpoložljivih kapacitet.

**VTS** (angl. Vessel Traffic System) – sistem za nadzor plovbe



# PRILOGA

## Vsebina priloge:

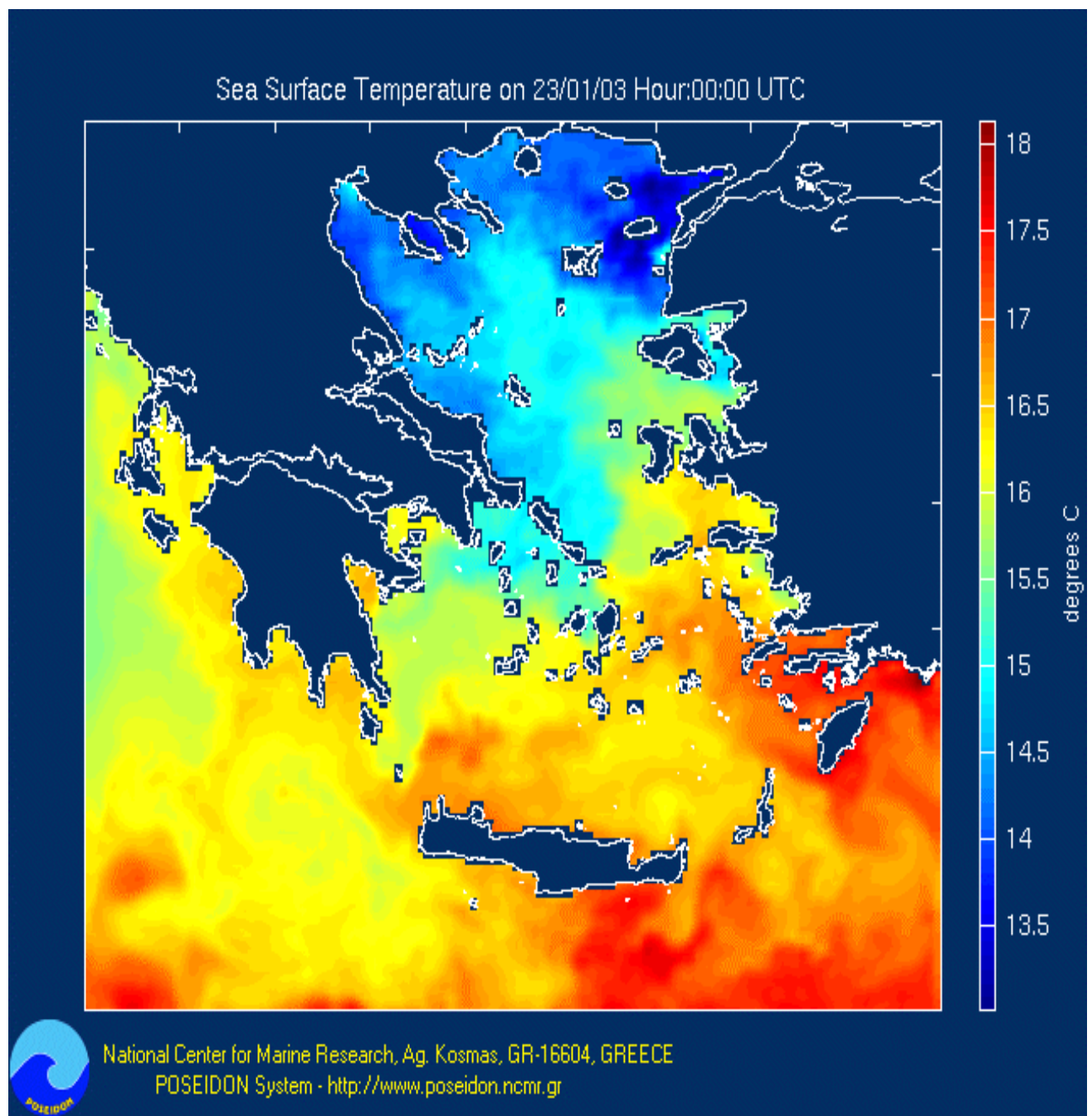
|  |    |
|--|----|
| Priloga 1: Zgradba datoteke »status.dat«   | 2  |
| Priloga 2: Napoved temperature na površini morja za 72 ur, kot jo prikazuje sistem Poseidon                            | 3  |
| Priloga 3: Prikaz podatkov za eno izmed boj sistema Poseidon   | 4  |
| Priloga 4: Prikaz sistema treh boj M3A   | 5  |
| Priloga 5: Prikaz izmerjenih podatkov sistema M3A  | 6  |
| Priloga 6: Shematičen prikaz strategije zbiranja podatkov sistema M3A  | 7  |
| Priloga 7: Primer »on-line« napovedi temperature morja za tri dni vnaprej  | 8  |
| Priloga 8: Premikanje prosto plavajočih boj v decembru 2002  | 9  |
| Priloga 9: Pozicije sistema SeaNet in prikaz izpisa izmerjenih podatkov  | 10 |
| Priloga 10: Sistem FerryBox - Zbiranje podatkov o kakovosti vode s pomočjo trajektne ladje in trajektne linije sistema | 11 |
| Priloga 11: Tekstualni opisi primerov uporabe  | 12 |
| Priloga 12: Zbirka vseh UML diagramov  | 15 |
| Priloga 13: SQL poizvedbe informacijskega sistema za potrebe gibanja Modra zastava                                     | 28 |
| Priloga 14: Primer skripte PHP za prikazovanje grafov  | 30 |

## Priloga 1: Zgradba datoteke »status.dat«

| Odmik v zlogih | Dolžina v zlogih | Opis  |
|----------------|------------------|---|
| 0              | 7                | Ura (1), minuta (1), sekunda (1), dan (1), mesec (1) in leto (2) vzpostavitve modemske povezave (UTC čas)   |
| 7              | 1                | Fina nastavitvev RTC ure na boji  |
| 8              | 1                | 1 – program za zajem podatkov teče<br>0 – program za zajem podatkov ne teče   |
| 9              | 4*1              | Meritve na procesorski plošči v surovi obliki (napetost, tok, temperatura, vlaga)   |
| 13             | 8*2              | V/I naslovi 8 zaporednih vrat na boji   |
| 29             | 8*4              | Nastavitve zaporednih vrat v obliki:<br>Zlog 0 – 1 vrata aktivna, 0 vrata neaktivna<br>Zlog 1,2 – Vrednost registra za hitrost prenosa (baud rate) v 16C654<br>Zlog 3 – Vrednost za LCR register v 16C654 (podatkovni biti, stop biti, parnost) |
| 61             | 40               | Podatki o CompactFlash kartici  |
| 101            | 1                | Fina nastavitvev RTC ure na boji  |
| 102            | 7                | Ura (1), minuta (1), sekunda (1), dan (1), mesec (1) in leto (2) zadnjega resetiranja proc. plošče na boji (UTC čas)  |
| 109            | 1                | Program za zajem podatkov naložen (1 – DA, 0 – NE)  |
| 110            | 1                | Celoten EEPROM zbrisan (1 – DA, 0 – NE, FF – neuporabljeno)   |
| 111            | 1                | Seznam vključenih naprav (bit0 – modem, bit1 – vlagomer, bit2 – Idronaut1, bit3 – Idronaut2, bit4 – vetromer, bit5 – ADP, bit6 – kompas, bit7 – GPS)  |
| 112            | 1                | Status zadnje bralno/pisalne operacije na CompactFlash kartici  |
| 113            | 1                | Stanje časovnika psa čuvaja (COP reset timeout)   |
| 114            | 4<br>6*1         | Številka zadnjega zapisanega odseka (sektorja) z meritvami na CompactFlash kartici ali<br>Če program za zajem podatkov ne teče, se tu nahajajo statusne BIOS spremenljivke (BIOS napake) in je s tem datoteke konec.                            |
| 118            | 4                | Številka zadnjega prebranega odseka (sektorja) z meritvami na CompactFlash kartici  |
| 122            | 2                | Čas vzorčenja A/D pretvornika na procesorski plošči v 50ms  |
| 124            | 2                | Čas vzorčenja vlagomera v ms  |
| 126            | 2                | Čas vzorčenja sonde Idronaut 1 v s  |
| 128            | 2                | Čas vzorčenja sonde Idronaut 2 v s  |
| 130            | 2                | Čas vzorčenja vetromera v 50ms  |
| 132            | 2                | Čas vzorčenja ADP (se ne uporablja)   |
| 134            | 2                | Čas vzorčenja kompasa v tihih (glej listine za kompas)  |
| 136            | 2                | Čas vzorčenja GPS v s   |
| 138            | 4                | Čas zajema podatkov v s   |
| 142            | 2                | Čas celotnega merilnega cikla v s   |
| 144            | 2                | Čas mirovanja v s   |
| 146            | 2                | Čas mirovanja minus čas priprave instrumentov minus čas startnega okna v s  |
| 148            | 2                | Čas mirovanja minus čas priprave instrumentov v s   |
| 150            | 2                | Čas mirovanja minus čas priprave instrumentov minus čas startnega okna minus čas predpriprave instrumentov v s  |
| 152            | 8*1              | Vključenost (1 vključen, 0 izključen) instrumentov v merilno verigo (procesorska plošča, vetromer, Idronaut1, Idronaut2, vetromer, ADP, kompas, GPS)  |
| 160            | 6*1              | Statusne BIOS spremenljivke (BIOS napake)   |
| 166            | 1                | Število OS napak (napake pri zajemu meritev)  |
| 167            | 1                | Prekoračitev medpomnilnika OS napak   |
| 168            | N                | Podatki o OS napakah  |
| 168+N          | 4*1              | Meritve na procesorski plošči v surovi obliki (napetost, tok, temperatura, vlaga)   |

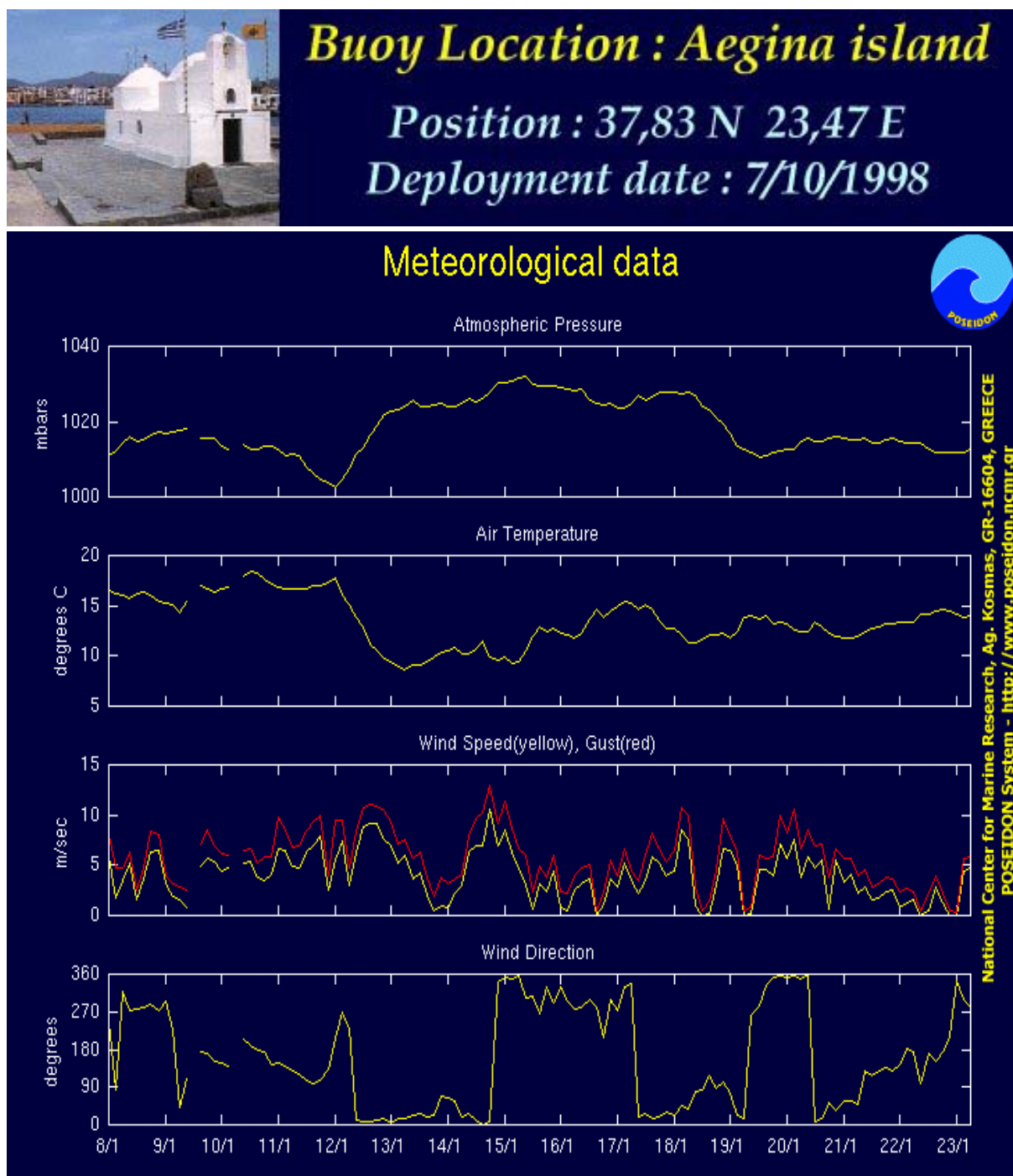
Vir: Šonc, 2003, str. 28-29

Priloga 2: Napoved temperature na površini morja za 72 ur, kot jo prikazuje sistem Poseidon



Vir: National Center for Marine Research - Institute of Oceanography Athens, 23. 1. 2003

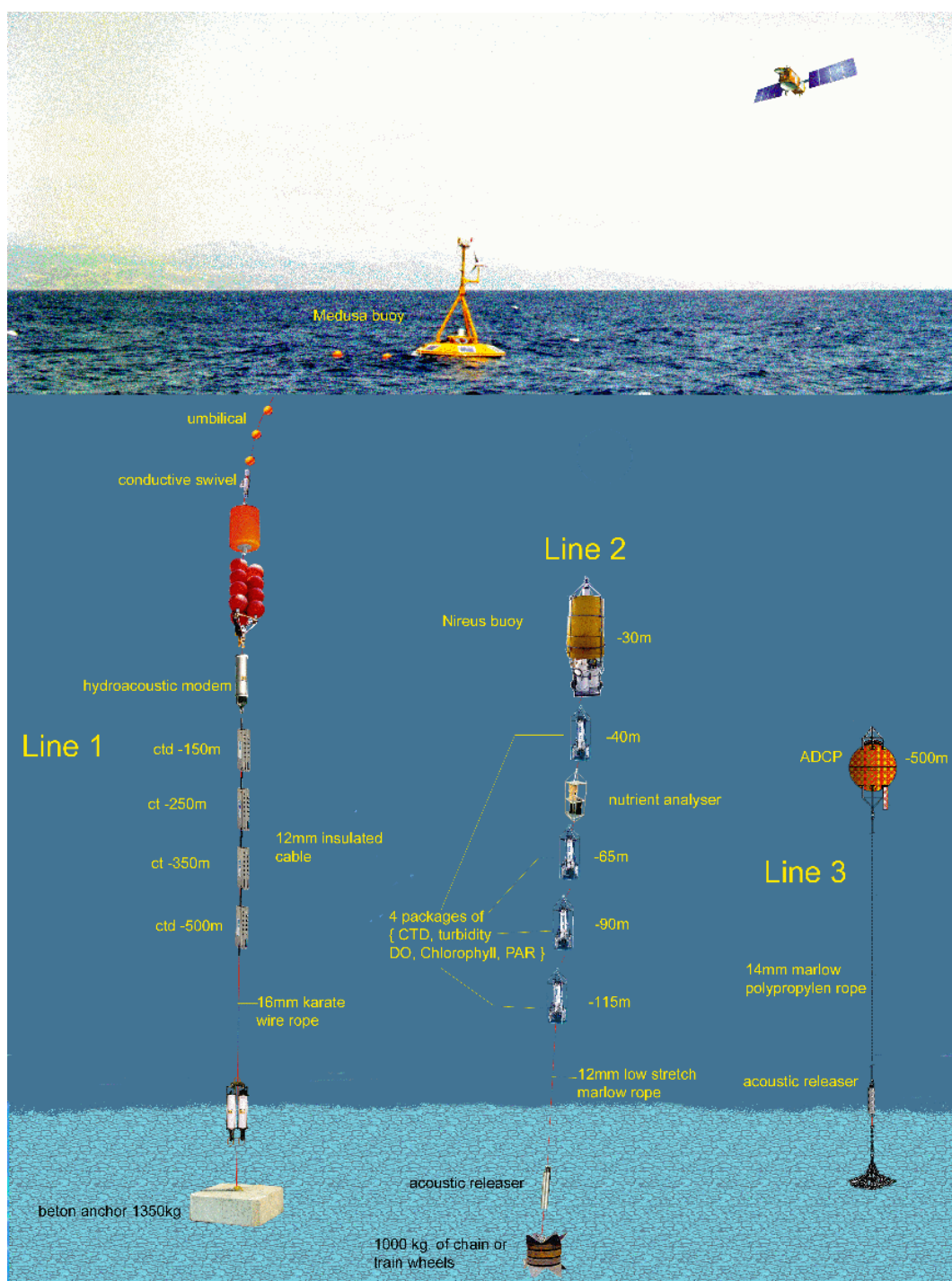
Priloga 3: Prikaz podatkov za eno izmed boj sistema Poseidon



Vir: National Center for Marine Research - Institute of Oceanography Athens, 23. 1. 2003



#### Priloga 4: Prikaz sistema treh boj M3A

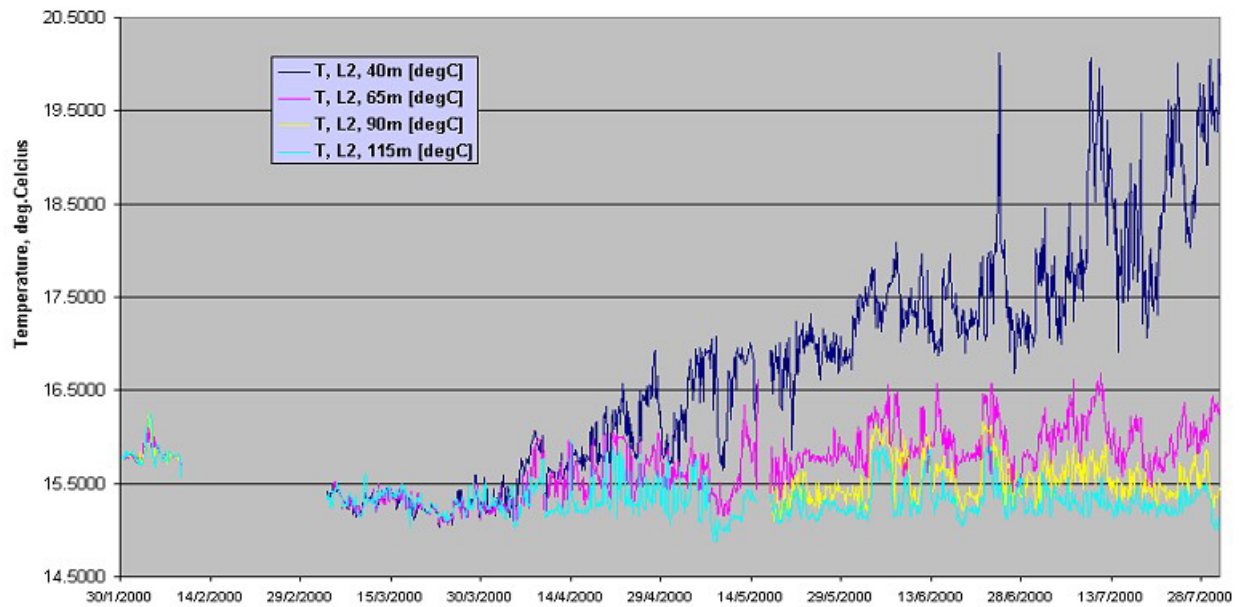


Vir: Martedec S.A. – The Marine Technology Development Company, 20. 1. 2003

## Priloga 5: Prikaz izmerjenih podatkov sistema M3A

### Temperature at 40, 65, 90 and 115 meters

Deployment period: 30/01/2000 - 31/07/2000



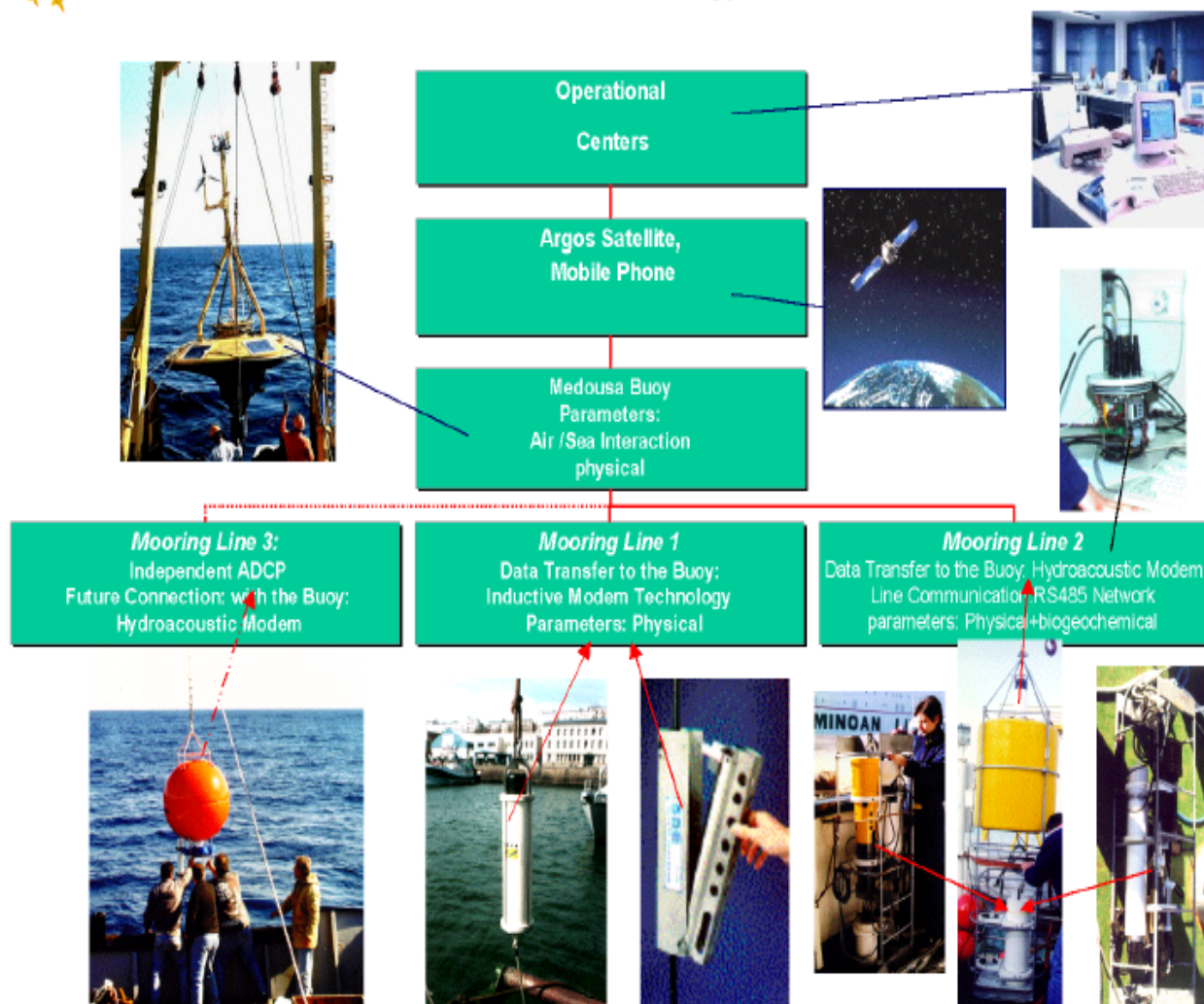
Vir: Martedec S.A. – The Marine Technology Development Company, 20. 1. 2003

## Priloga 6: Shematičen prikaz strategije zbiranja podatkov sistema M3A



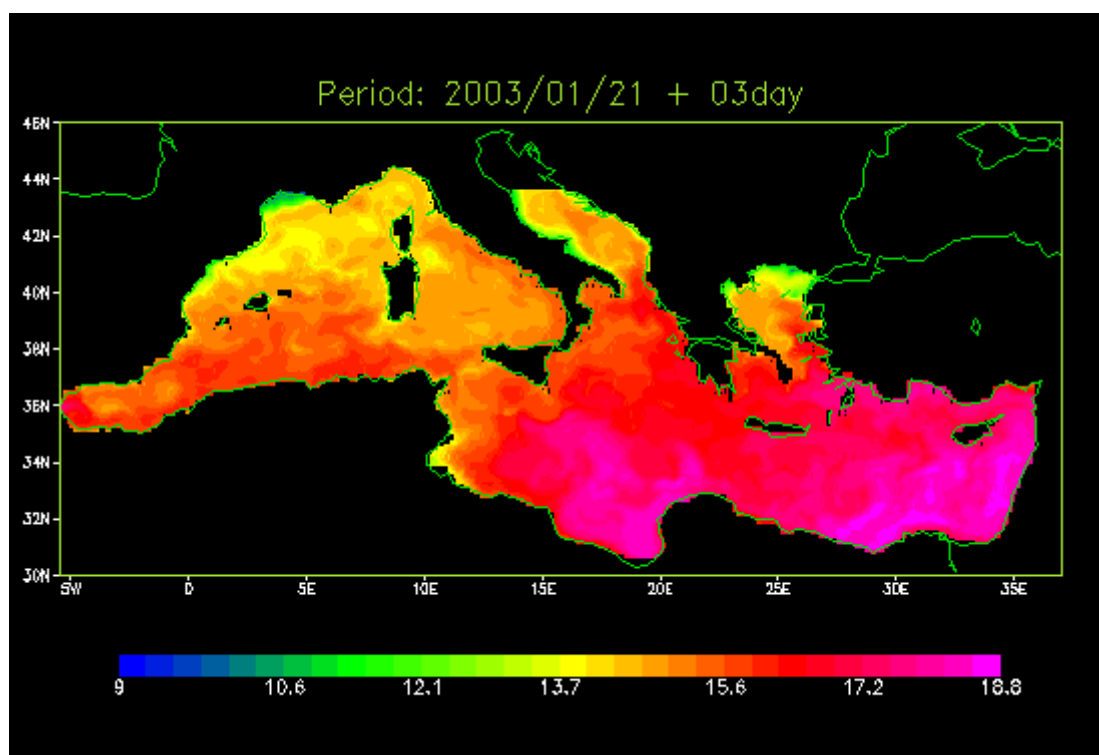
### Development, Construction and Deployment of the M3A system

#### Data Collection Strategy



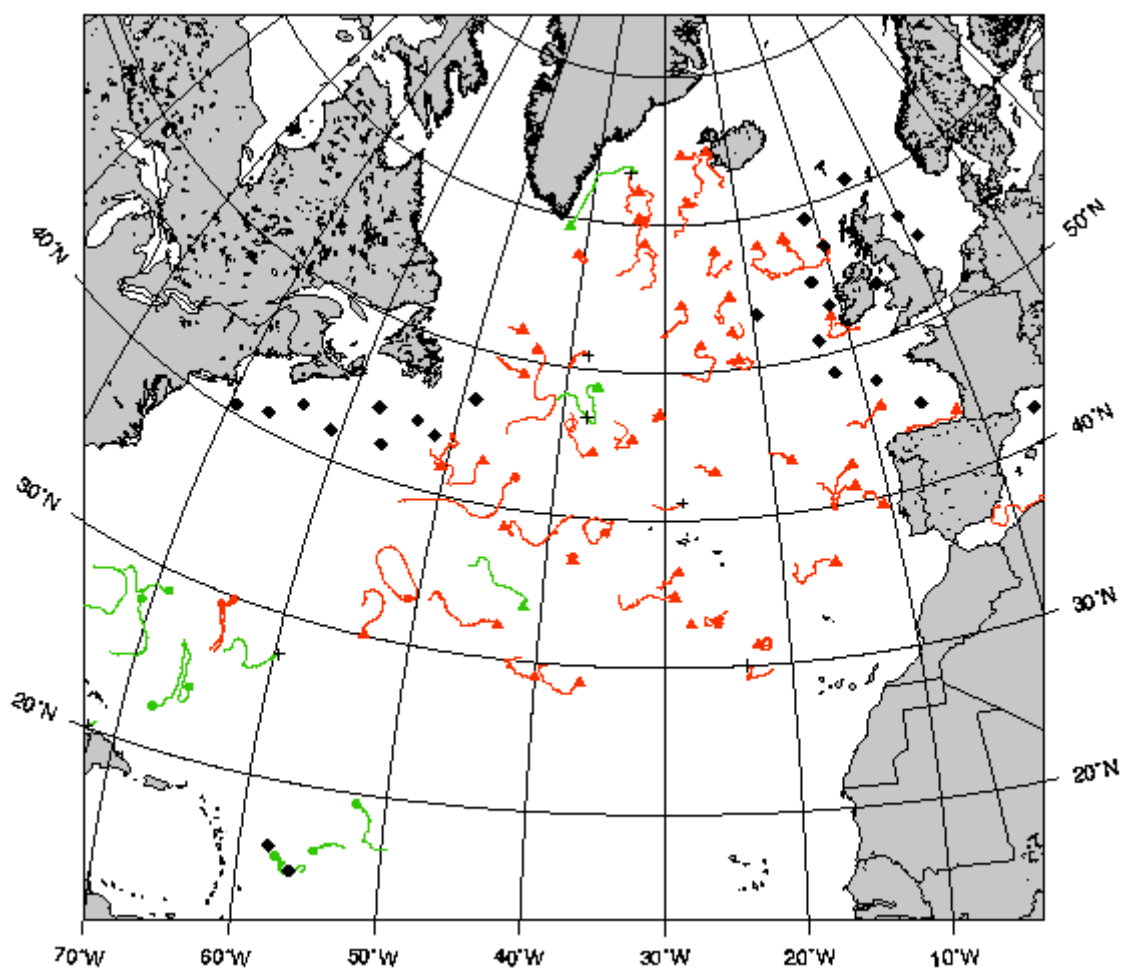
Vir: Martedec S.A. – The Marine Technology Development Company, 20. 1. 2003




**Priloga 7: Primer »on-line« napovedi temperature morja za tri dni vnaprej**



**Vir: MedGOOS – The Mediterranean Global Ocean Observing System, 21. 1. 2003**

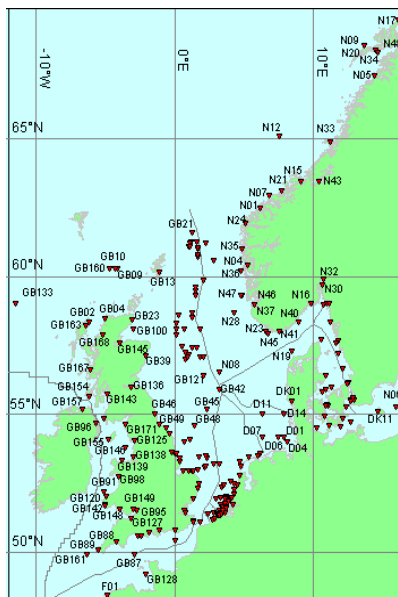
### Priloga 8: Premikanje prosto plavajočih boj v decembru 2002



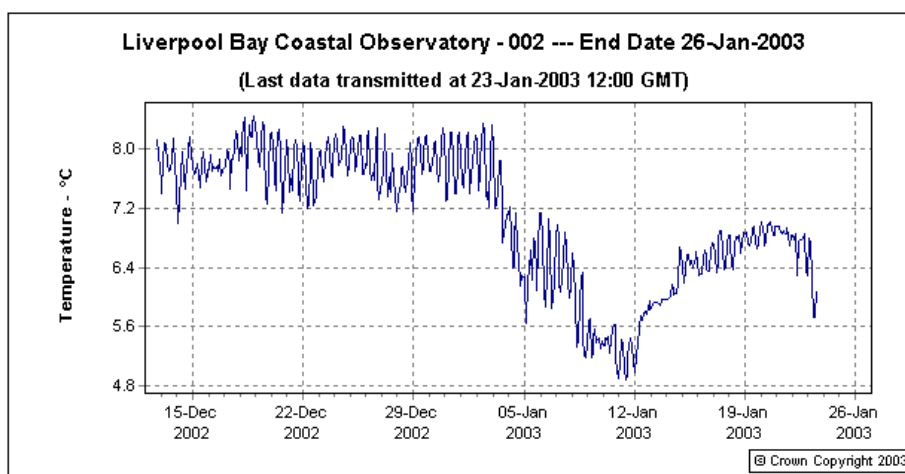
-  plavajoče boje, ki nudijo podatke o zračnem pritisku
-  plavajoče boje, ki nudijo podatke o vetru
-  zasidrane boje

Vir: EGOS – European Group on Ocean Stations, 1. 2. 2003

## A yellow towed buoy with a white sensor housing and a yellow cross-shaped frame, floating on the water. The buoy has the text "PROPERTY OF GOVERNMENT" and "CEFAS" printed on it, along with the number "1".



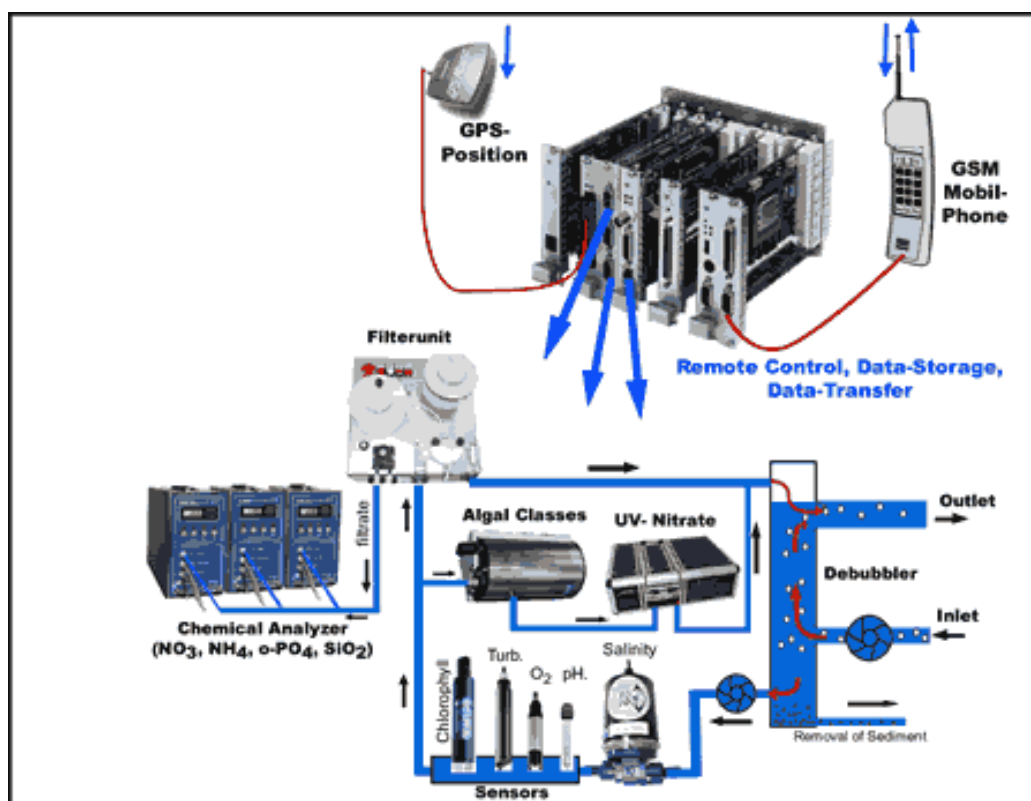
**Prikaz izmerjenih podatkov (temperature vode) sistema SeaNet:**



**Vir: CEFAS – Monitoring the health of our seas, 26. 1. 2003**



**Priloga 10: Sistem FerryBox - Zbiranje podatkov o kakovosti vode s pomočjo trajektne ladje in trajektne linije sistema**



Vir: Southampton Oceanography Centre 15. 2. 2003

**Priloga 11: Tekstualni opisi primerov uporabe**

|                                    |   |
|------------------------------------|---|
| <b>Ime</b>                         | Ogled spletne strani  |
| <b>Kratek opis</b>                 | Uporabnik si na spletni strani oleda podatke za izbrano kopališče   |
| <b>Primarni akter</b>              | Uporabnik   |
| <b>Sekundarni akter</b>            | -   |
| <b>Začetek</b>                     | Uporabnik izbere spletno stran  |
| <b>Konec</b>                       | Spletna stran je prikazana  |
| <b>Merljivi izid</b>               | Izdelava in prikaz spletne strani   |
| <b>Potek dogodkov</b>              | Uporabnik izbere spletno stran s podatki o kopališčih. Izbere kopališče, ki ga zanima. Spletni pregledovalnik pošlje spletnemu strežniku podatek o kopališču in le te iz podatkov v zbirki podatkov izdela spletno stran za izbrano kopališče. Spletna stran se prikaže v spletnem pregledovalniku. |
| <b>Alternativni potek dogodkov</b> | -   |
| <b>Povezani primeri</b>            | -   |
| <b>Odprta vprašanja</b>            | -   |

|                                    |  |
|------------------------------------|--|
| <b>Ime</b>                         | Vpis podatkov o stanju kopališča   |
| <b>Kratek opis</b>                 | Upravitelj kopališča vpiše opisne podatke o kopališču, ki ga upravlja  |
| <b>Primarni akter</b>              | Upravitelj kopališča   |
| <b>Sekundarni akter</b>            | -  |
| <b>Začetek</b>                     | Uporabnik izbere spletno stran za vpis podatkov  |
| <b>Konec</b>                       | Podatki o kopališču so zapisani  |
| <b>Merljivi izid</b>               | Podatki v zbirki podatkov  |
| <b>Potek dogodkov</b>              | Upravitelj izbere spletno stran za vpis podatkov o kopališču. Pred vstopom na stran se mora identificirati in autenticirati (uporabniško ime in geslo). V spletnem pregledovalniku se pojavi stran s seznamom kopališč, ki jih upravlja. Upravitelj izbere kopališče in na spletni strani se pojavi obrazec za vpis opisnih podatkov. Upravitelj vpiše podatke in izbere akcijo zapiši. Spletna stran na spletni strežnik pošlje zahtevo za vpis podatkov v zbirko podatkov. Če se podatki uspešno zapišejo strežnik to sporoči spletnemu pregledovalniku. |
| <b>Alternativni potek dogodkov</b> | V primeru, ko vpis podatkov ne uspe (podatki se ne vpišejo v zbirko podatkov) spletni strežnik to sporoči upravitelju s sporočilom v spletnem pregledovalniku.   |
| <b>Povezani primeri</b>            | -  |
| <b>Odprta vprašanja</b>            | -  |



|                                    |  |
|------------------------------------|--|
| <b>Ime</b>                         | Prenos podatkov "Modra zastava"  |
| <b>Kratek opis</b>                 | Uporabnik s spletnega strežnika prenese datoteko s podatki dogovorjenimi s programom "Modra zastava"   |
| <b>Primarni akter</b>              | Uporabnik "Modra zastava"  |
| <b>Sekundarni akter</b>            | -  |
| <b>Začetek</b>                     | Uporabnik izbere spletno stran   |
| <b>Konec</b>                       | Datoteka s podatki je prenešana na uporabnikov računalnik  |
| <b>Merljivi izid</b>               | Datoteka na uporabnikovem računalniku  |
| <b>Potek dogodkov</b>              | Uporabnik izbere spletno stran za prenos datoteke s podatki. Pred vstopom na stran se mora identificirati in autentificirati (uporabniško ime in geslo). V spletnem pregledovalniku se pojavi stran s seznamom kopališč in s parametri (časovno obdobje,...). Uporabnik izbere kopališče in vpiše vrednosti parametrov ter izbere akcijo prenos. Spletna stran na spletni strežnik pošlje zahtevo za izdelavo in prenos datoteke. Spletni strežnik izdelava datoteko (podatke pridobi iz zbirke podatkov) in jo pošlje na spletni pregledovalnik. Spletni pregledovalnik shrani datoteko na disk na uporabnikovem računalniku. |
| <b>Alternativni potek dogodkov</b> | -  |
| <b>Povezani primeri</b>            | -  |
| <b>Odprta vprašanja</b>            | -  |

|                                    |   |
|------------------------------------|---|
| <b>Ime</b>                         | Spreminjanje gesla  |
| <b>Kratek opis</b>                 | Administrator spremeni uporabnikovo geslo   |
| <b>Primarni akter</b>              | Administrator   |
| <b>Sekundarni akter</b>            | -   |
| <b>Začetek</b>                     | Administrator izbere spletno stran  |
| <b>Konec</b>                       | Spremenjeno geslo je zapisano v zbirki podatkov   |
| <b>Merljivi izid</b>               | Zapisano geslo (novo) v zbirki podatkov   |
| <b>Potek dogodkov</b>              | Pred vstopom na stran se mora administrator identificirati in autentificirati (uporabniško ime in geslo). V spletnem pregledovalniku se pojavi stran s seznamom uporabnikov sistema. Administrator izbere uporabnika in sproži zahtevo za spremembo gesla. Sistem mu ponudi stran na kateri lahko spremeni geslo in administrator vpiše novo geslo (pri vpisu podatkov o geslu se le to ne vidi na spletni strani). Po vpisu podatkov administrator zahteva zapis podatkov v zbirko podatkov. Zahteva se skupaj z novim geslom prenese na spletni strežnik, kjer se zapiše v zbirko podatkov. |
| <b>Alternativni potek dogodkov</b> | -   |
| <b>Povezani primeri</b>            | -   |
| <b>Odprta vprašanja</b>            | -   |

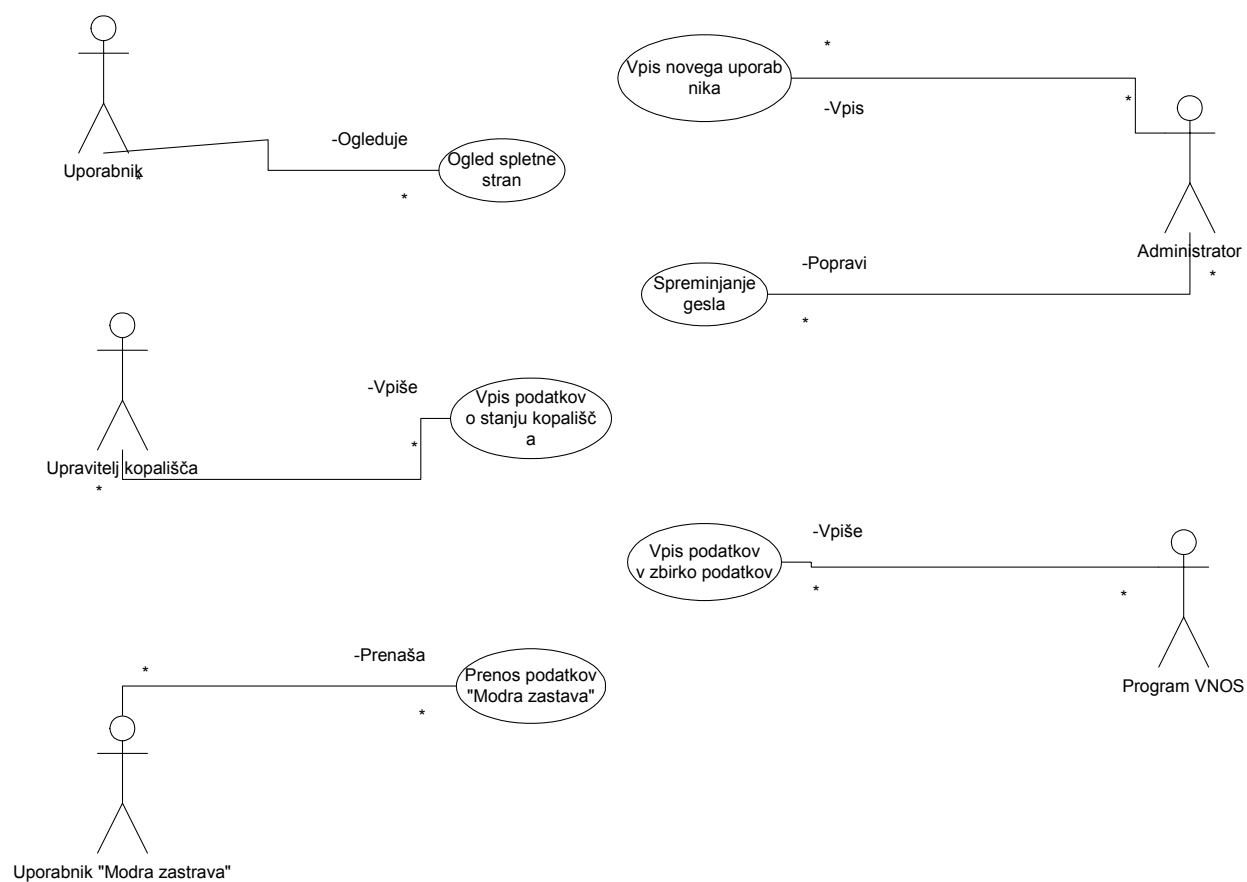
|                                    |  |
|------------------------------------|--|
| <b>Ime</b>                         | Vpis podatkov o novemu uporabniku  |
| <b>Kratek opis</b>                 | Administrator vpiše podatke o novemu uporabniku (upravitelj ali uporabnik "Modra zastava")   |
| <b>Primarni akter</b>              | Administrator  |
| <b>Sekundarni akter</b>            | -  |
| <b>Začetek</b>                     | Administrator izbere spletno stran   |
| <b>Konec</b>                       | Podatki o novem uporabniku so zapisani v zbirko podatkov   |
| <b>Merljivi izid</b>               | Zapisani podatki o novemu uporabniku v zbirki podatkov   |
| <b>Potek dogodkov</b>              | Pred vstopom na stran se mora administrator identificirati in autenticirati (uporabniško ime in geslo). V spletnem pregledovalniku se pojavi stran na kateri lahko administrator vpiše podatke o novemu uporabniku. Vpisuje naslednje podatke: uporabniško ime, geslo, podatke o uporabniku (ime, priimek,...), ali je uporabnik del programa modra zastava in ali je uporabnik upravitelj kopališč. Če je uporabnik upravitelj kopališč mu administrator še priredi kopališča katera upravlja. Po vpisu podatkov administrator zahteva zapis podatkov v zbirko podatkov. Zahteva se skupaj s podatki prenese na spletni strežnik, kjer se zapiše v zbirko podatkov. |
| <b>Alternativni potek dogodkov</b> | -  |
| <b>Povezani primeri</b>            | -  |
| <b>Odprta vprašanja</b>            | -  |

|                                    |  |
|------------------------------------|--|
| <b>Ime</b>                         | Vpis meritev v zbirko podatkov   |
| <b>Kratek opis</b>                 | Izmerjeni podatki se zapišejo v zbirko podatkov  |
| <b>Primarni akter</b>              | Program VPIS   |
| <b>Sekundarni akter</b>            | -  |
| <b>Začetek</b>                     | Program dobi signal, da so podatki pripravljeni za vpis v zbirko podatkov  |
| <b>Konec</b>                       | Program se pritaji in čaka na nov signal   |
| <b>Merljivi izid</b>               | Zapisani podatki v zbirki podatkov   |
| <b>Potek dogodkov</b>              | Program VPIS, ki je pritajen v pomnilniku od "semaforja" dobi signal, da lahko prične s prenosom podatkov v zbirko podatkov. Program prične brati pomnilniške lokacije, kjer so zapisani podatki in jih zapisovati v relacijsko zbirko podatkov. Podatke prepisuje toliko časa dokler ne prepiše vseh. Ko konča prepisovanje se program ponovno pritaji in čaka na naslednji signal. |
| <b>Alternativni potek dogodkov</b> | -  |
| <b>Povezani primeri</b>            | -  |
| <b>Odprta vprašanja</b>            | -  |

Vir: avtor

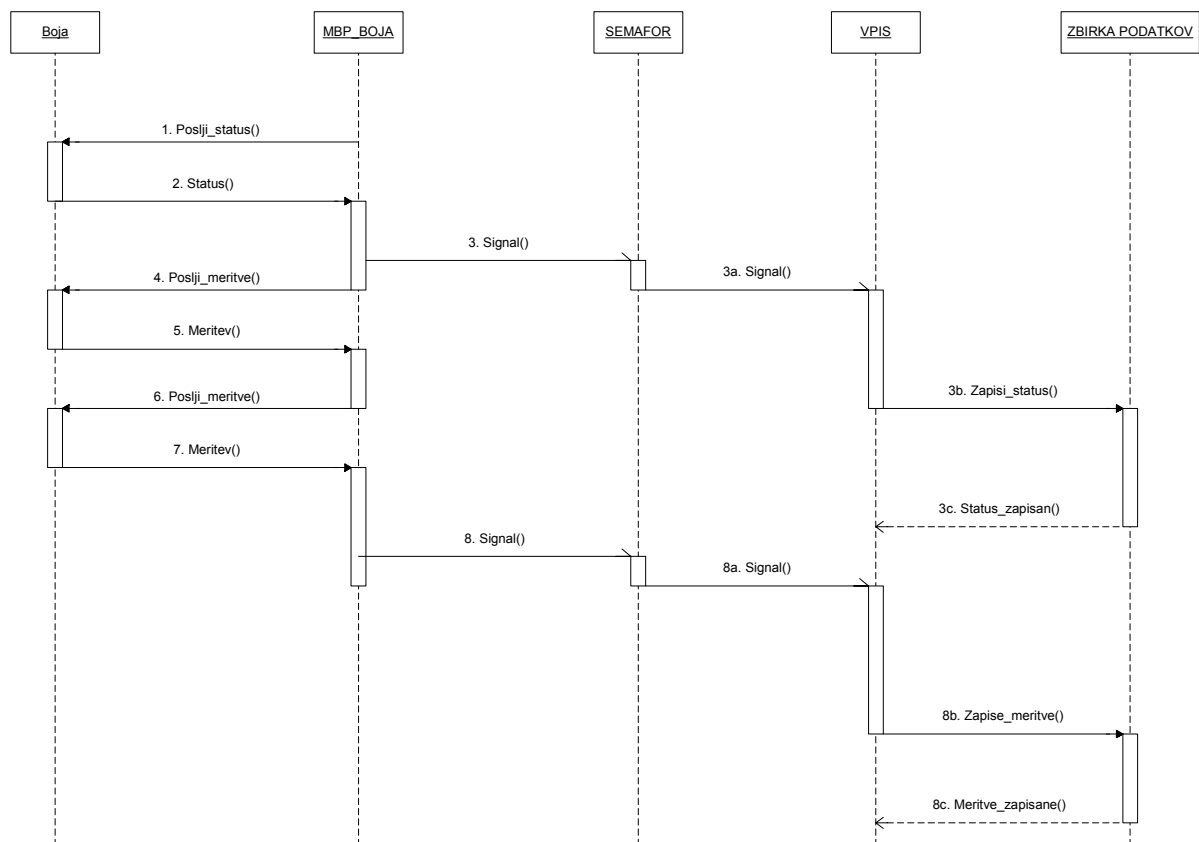
## Priloga 12: Zbirka vseh UML diagramov

### 12.1 Diagrami primerov uporabe za spletni strežnik

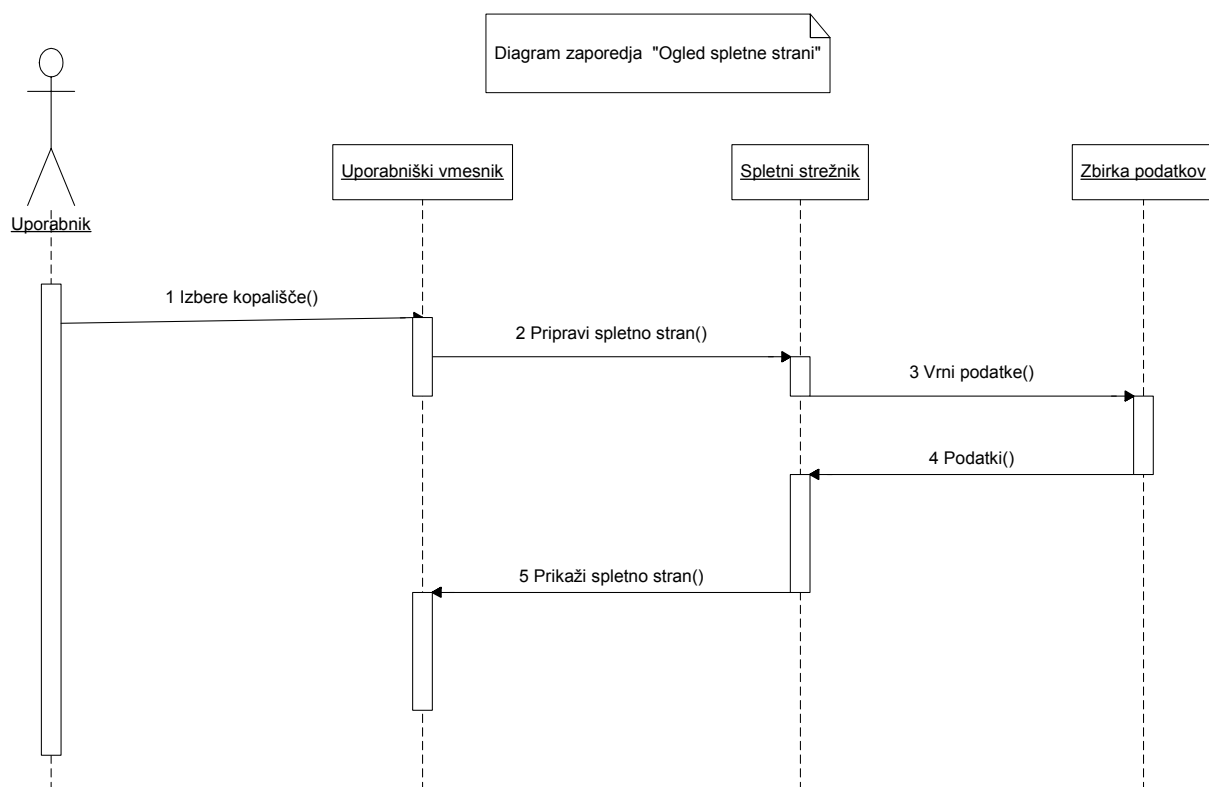


## 12.2 Diagrami zaporedja

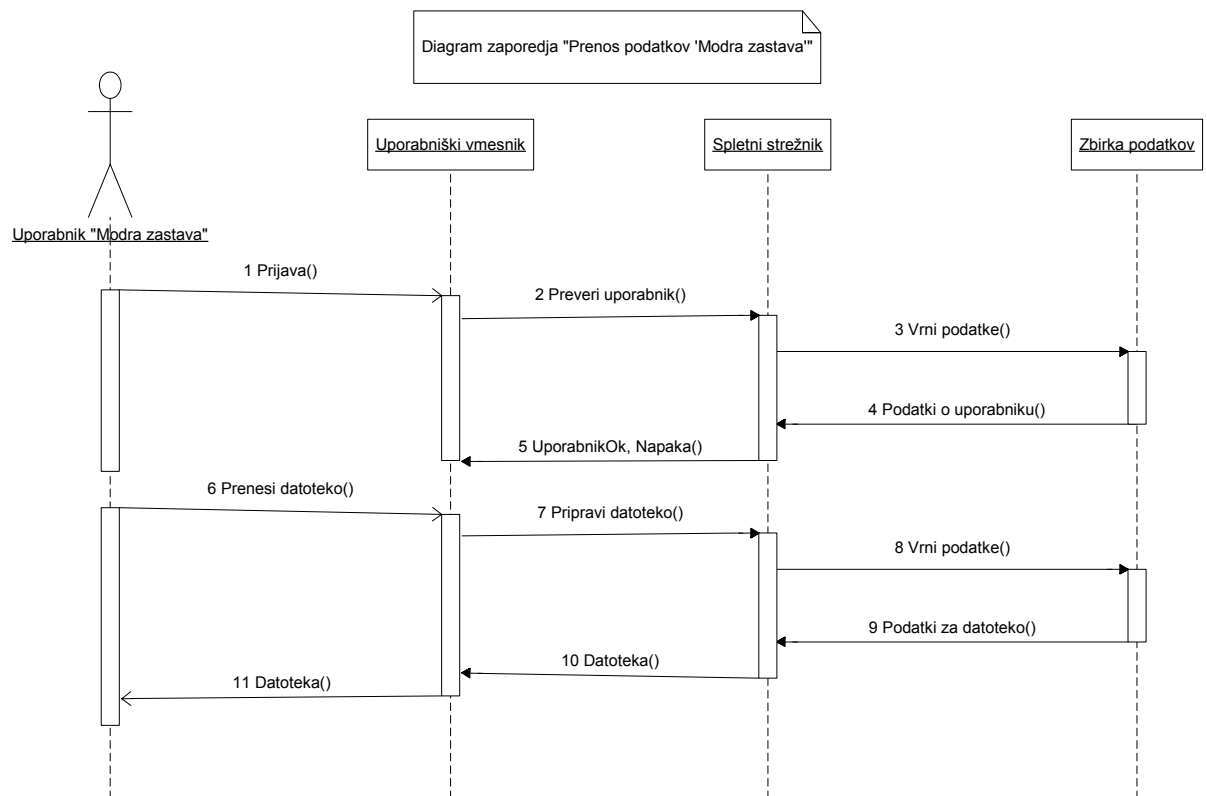
### Prenos podatkov iz boje v zbirko podatkov



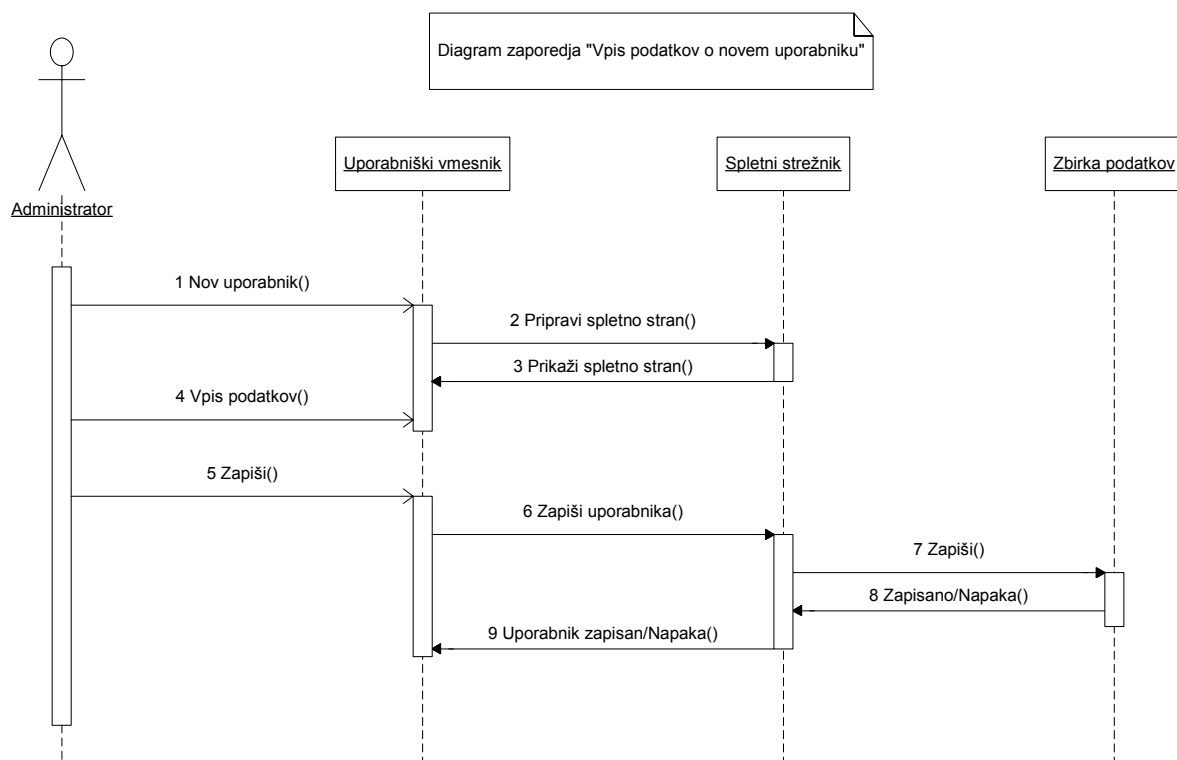
## Ogled spletne strani



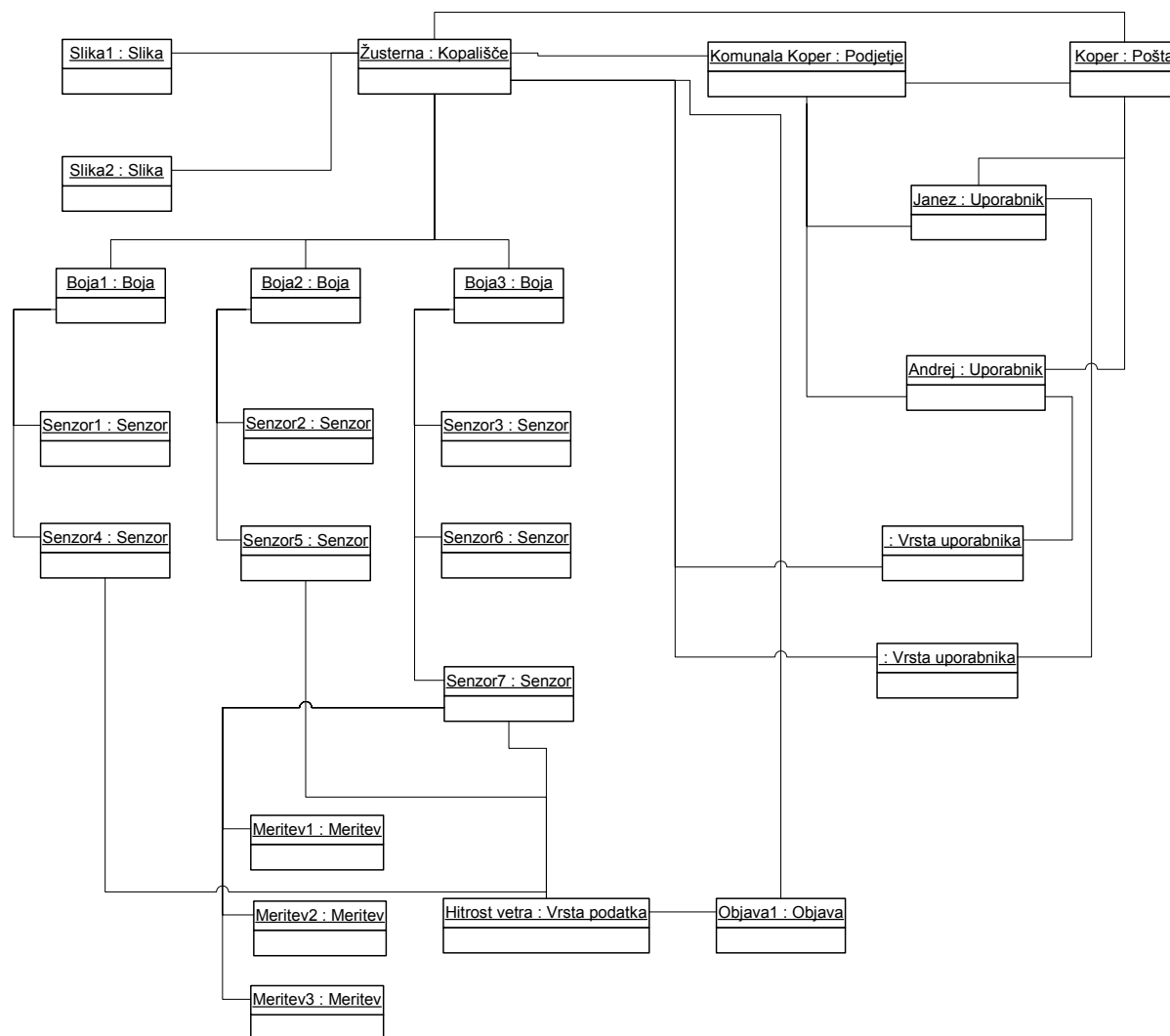
## Prenos podatkov 'Modra zastava'



## Vpis podatkov o novem uporabniku

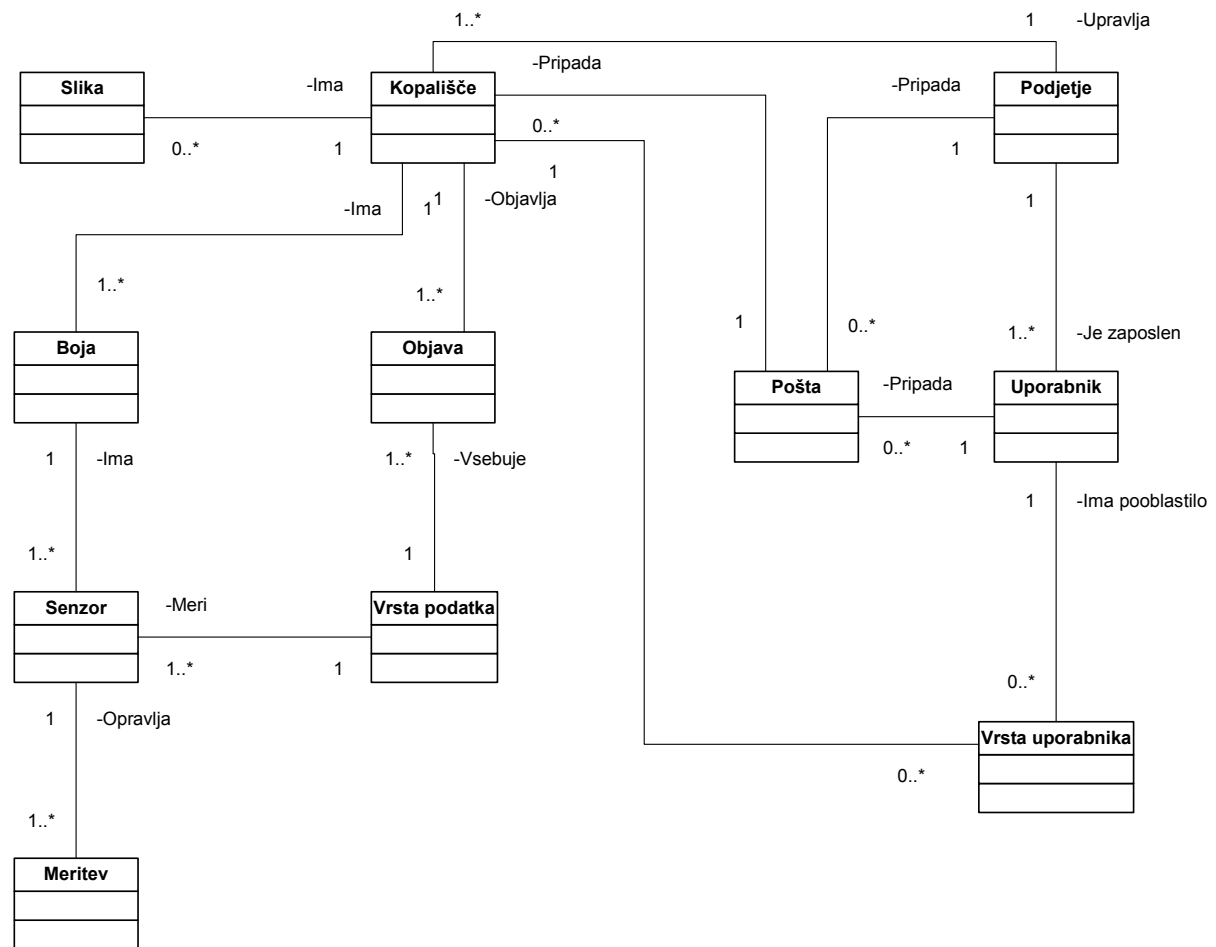


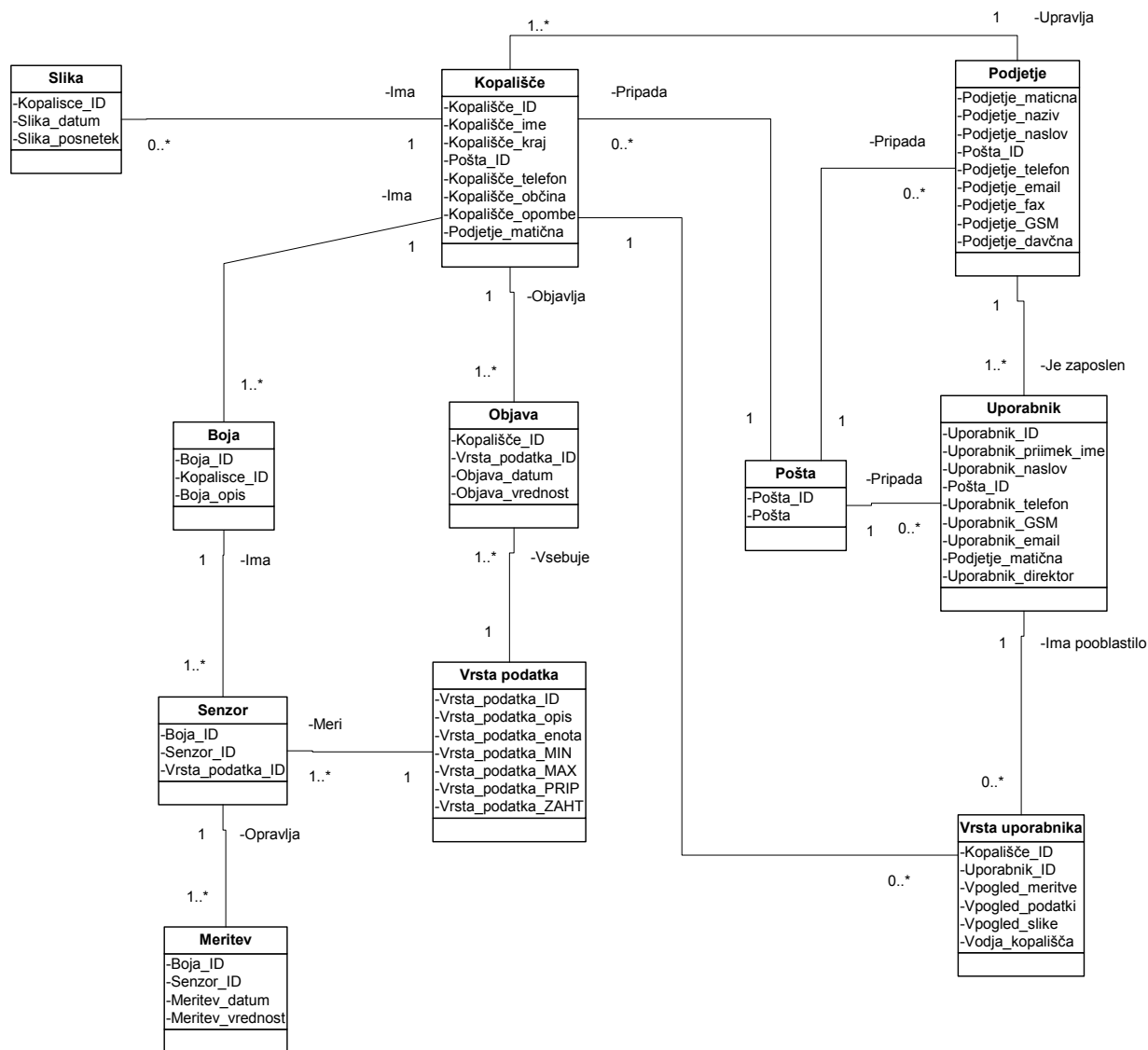
## 12.3 Objektni diagram – zbirka podatkov za poročanje





## 12.4 Razredni diagram brez in z atributi – podatki za poročanje





## 12.5 Diagram aktivnosti

Ogled spletne strani

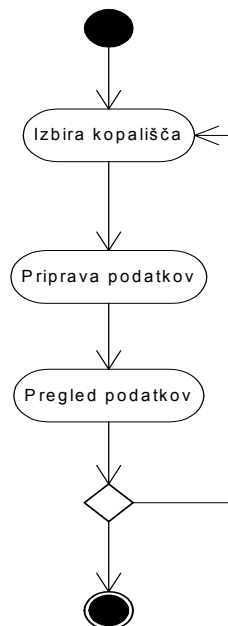


Diagram aktivnosti "Ogled spletne strani"

Prenos podatkov 'Modra zastava'

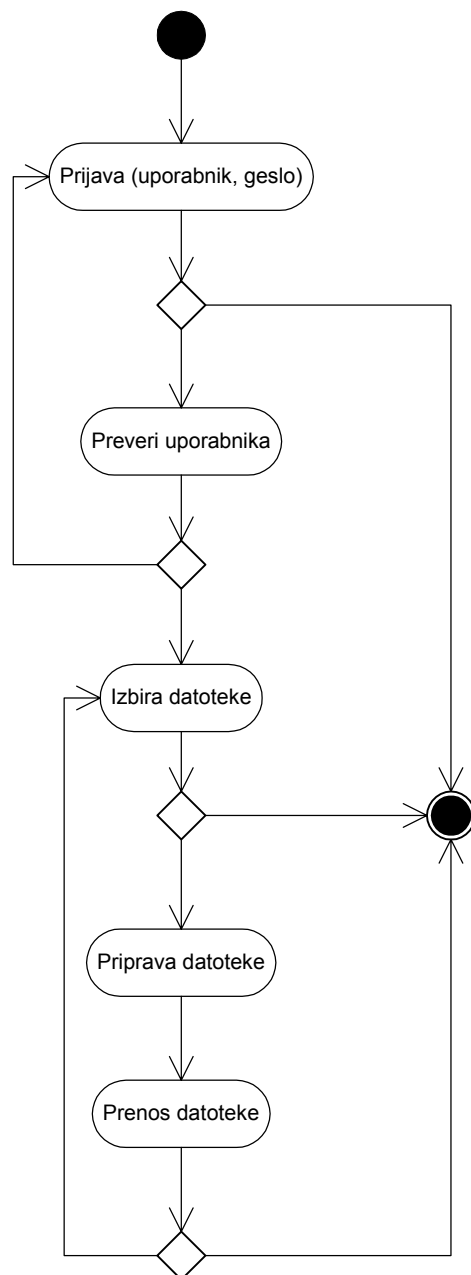


Diagram aktivnosti "Prenos podatkov 'Modra zastava'"

## Vpis novega uporabnika

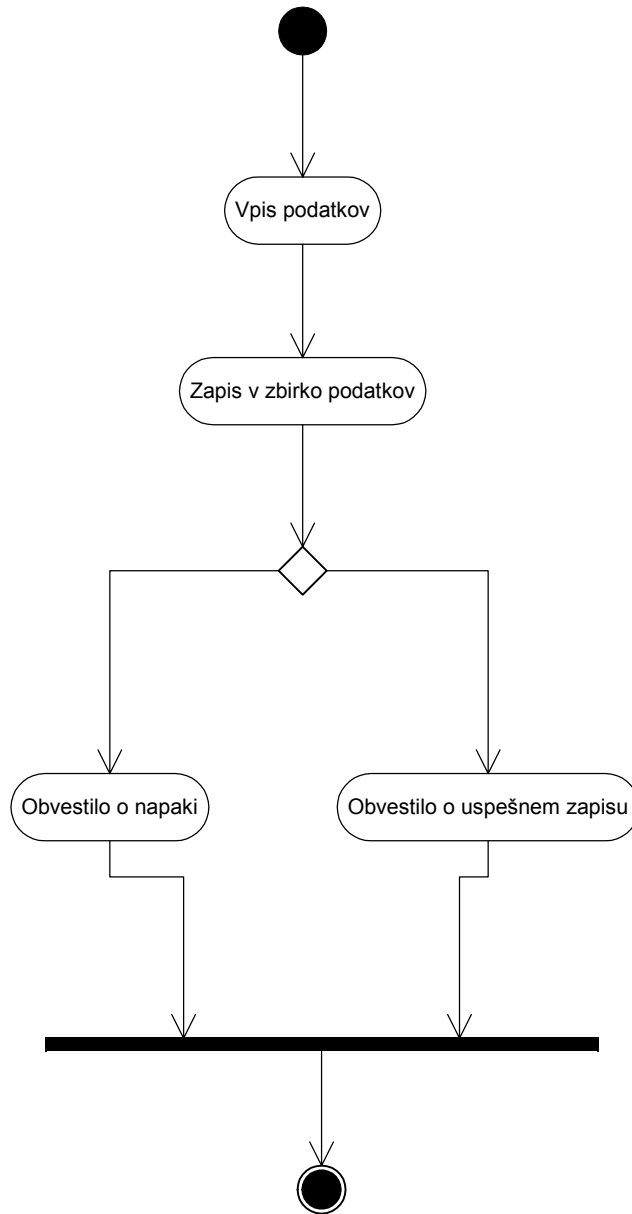
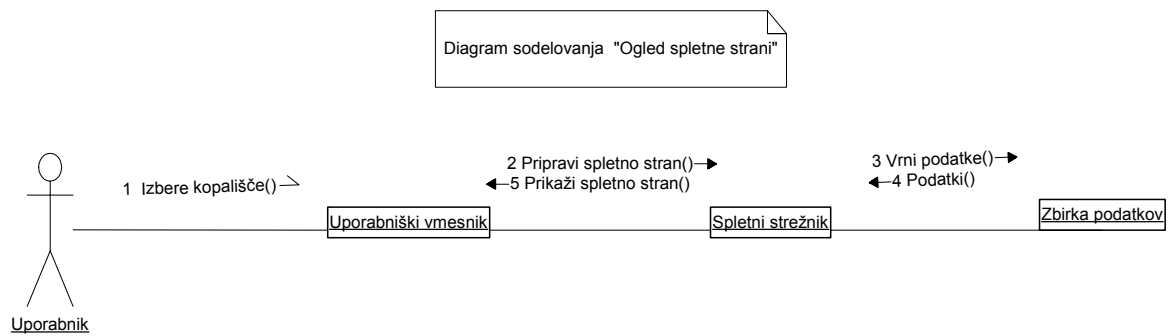


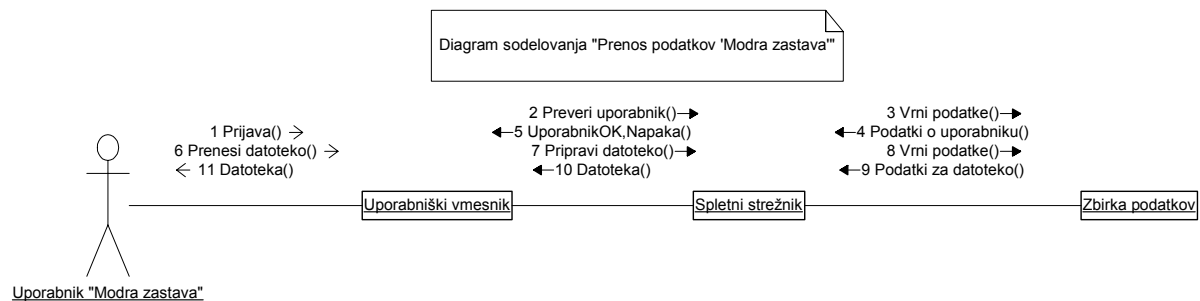
Diagram aktivnosti "Vpis novega uporabnika"

## 12.6 Diagram sodelovanja

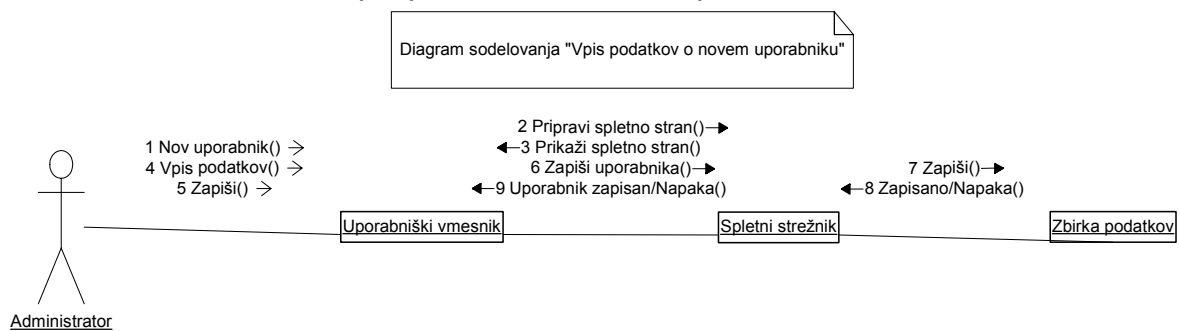
## Ogled spletne strani



## Prenos podatkov 'Modra zastava'

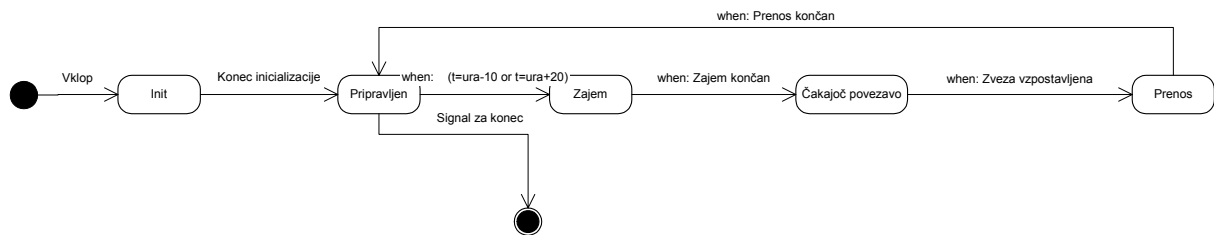


## Vpis podatkov o novem uporabniku

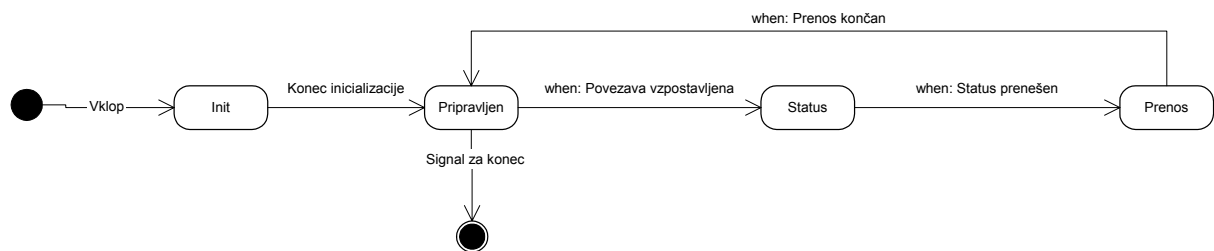


## 12.7 Diagram stanj

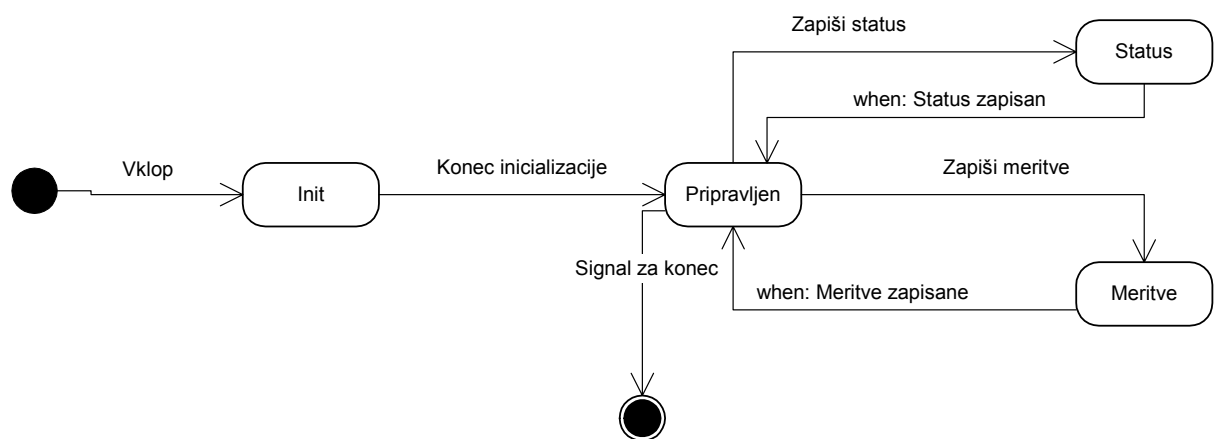
### Boja



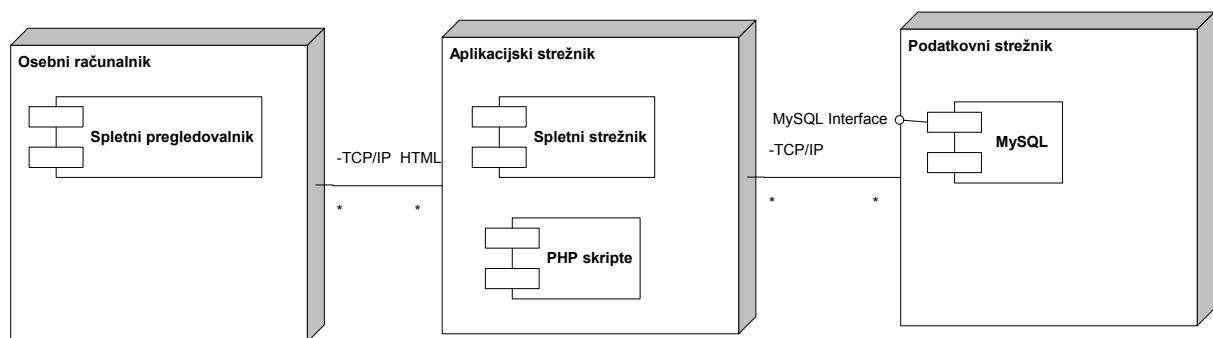
### program MBP\_BOJA



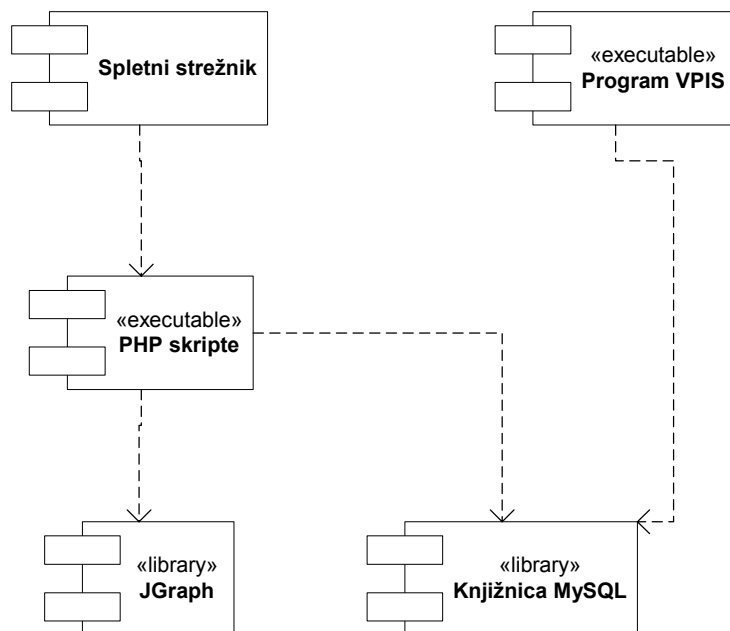
### program VPIS



## 12.8 Diagram porazdelitve



## 12.9 Diagram komponent za spletni strežnik



Vir: avtor

## Priloga 13: SQL poizvedbe informacijskega sistema za potrebe gibanja Modra zastava

SQL poizvedba za prikaz vseh zadnjih izmerjenih podatkov za izbrano kopališče:

```
SELECT Kopalisca.Kopalisce_ID, Boje.Boja_ID, Boje.Boja_opis,
Senzorji.Senzor_ID, Senzorji.Vrsta_podatka_ID,
Vrsta_podatka.Vrsta_podatka_opis, Vrsta_podatka.Vrsta_podatka_enota,
Vrsta_podatka.Vrsta_podatka_MIN, Vrsta_podatka.Vrsta_podatka_MAX,
Vrsta_podatka.Vrsta_podatka_PRIP, Vrsta_podatka.Vrsta_podatka_ZAHT,
Last(Meritve.Meritev_datum) AS LastOfMeritev_datum,
Last(Meritve.Meritev_vrednost) AS LastOfMeritev_vrednost
FROM Vrsta_podatka RIGHT OUTER JOIN ((Kopalisca LEFT OUTER JOIN (Boje LEFT
OUTER JOIN Senzorji ON Boje.Boja_ID = Senzorji.Boja_ID) ON
Kopalisca.Kopalisce_ID = Boje.Kopalisce_ID) LEFT OUTER JOIN Meritve ON
(Senzorji.Senzor_ID = Meritve.Senzor_ID) AND (Senzorji.Boja_ID =
Meritve.Boja_ID)) ON Vrsta_podatka.Vrsta_podatka_ID =
Senzorji.Vrsta_podatka_ID
GROUP BY Kopalisca.Kopalisce_ID, Boje.Boja_ID, Boje.Boja_opis,
Senzorji.Senzor_ID, Senzorji.Vrsta_podatka_ID,
Vrsta_podatka.Vrsta_podatka_opis, Vrsta_podatka.Vrsta_podatka_enota,
Vrsta_podatka.Vrsta_podatka_MIN, Vrsta_podatka.Vrsta_podatka_MAX,
Vrsta_podatka.Vrsta_podatka_PRIP, Vrsta_podatka.Vrsta_podatka_ZAHT
HAVING (((Kopalisca.Kopalisce_ID)=[Vnesi šifro kopališča]));
```

SQL poizvedba za prikaz zadnjih izmerjenih podatkov za izbran parameter in izbrano kopališče:

```
SELECT Kopalisca.Kopalisce_ID, Boje.Boja_ID, Boje.Boja_opis,
Senzorji.Senzor_ID, Senzorji.Vrsta_podatka_ID,
Vrsta_podatka.Vrsta_podatka_opis, Vrsta_podatka.Vrsta_podatka_enota,
Vrsta_podatka.Vrsta_podatka_MIN, Vrsta_podatka.Vrsta_podatka_MAX,
Vrsta_podatka.Vrsta_podatka_PRIP, Vrsta_podatka.Vrsta_podatka_ZAHT,
Last(Meritve.Meritev_datum) AS LastOfMeritev_datum,
Last(Meritve.Meritev_vrednost) AS LastOfMeritev_vrednost
FROM Vrsta_podatka RIGHT OUTER JOIN ((Kopalisca LEFT OUTER JOIN (Boje LEFT
OUTER JOIN Senzorji ON Boje.Boja_ID = Senzorji.Boja_ID) ON
Kopalisca.Kopalisce_ID = Boje.Kopalisce_ID) LEFT OUTER JOIN Meritve ON
(Senzorji.Senzor_ID = Meritve.Senzor_ID) AND (Senzorji.Boja_ID =
Meritve.Boja_ID)) ON Vrsta_podatka.Vrsta_podatka_ID =
Senzorji.Vrsta_podatka_ID
GROUP BY Kopalisca.Kopalisce_ID, Boje.Boja_ID, Boje.Boja_opis,
Senzorji.Senzor_ID, Senzorji.Vrsta_podatka_ID,
Vrsta_podatka.Vrsta_podatka_opis, Vrsta_podatka.Vrsta_podatka_enota,
Vrsta_podatka.Vrsta_podatka_MIN, Vrsta_podatka.Vrsta_podatka_MAX,
Vrsta_podatka.Vrsta_podatka_PRIP, Vrsta_podatka.Vrsta_podatka_ZAHT
HAVING (((Kopalisca.Kopalisce_ID)=[Vnesi šifro kopališča]) AND
((Senzorji.Vrsta_podatka_ID)=[Vnesi vrsto podatka]));
```



SQL poizvedba za prikaz izmerjenih vrednosti kopališča v izbranem obdobju za izbrano kopališče in izbran parameter:

```
SELECT Kopalisca.Kopalisce_ID, Kopalisca.Kopalisce_ime,
Kopalisca.Kopalisce_opombe, Boje.Boja_ID, Boje.Boja_opis,
Senzorji.Senzor_ID, Senzorji.Vrsta_podatka_ID,
Vrsta_podatka.Vrsta_podatka_opis, Vrsta_podatka.Vrsta_podatka_enota,
Vrsta_podatka.Vrsta_podatka_MIN, Vrsta_podatka.Vrsta_podatka_MAX,
Vrsta_podatka.Vrsta_podatka_PRIP, Vrsta_podatka.Vrsta_podatka_ZAHT,
Meritve.Meritev_datum, Meritve.Meritev_vrednost
FROM Vrsta_podatka RIGHT OUTER JOIN ((Kopalisca LEFT OUTER JOIN (Boje LEFT
OUTER JOIN Senzorji ON Boje.Boja_ID = Senzorji.Boja_ID) ON
Kopalisca.Kopalisce_ID = Boje.Kopalisce_ID) LEFT OUTER JOIN Meritve ON
(Senzorji.Senzor_ID = Meritve.Senzor_ID) AND (Senzorji.Boja_ID =
Meritve.Boja_ID)) ON Vrsta_podatka.Vrsta_podatka_ID =
Senzorji.Vrsta_podatka_ID
WHERE (((Senzorji.Vrsta_podatka_ID)=[Vnesi vrsto podatka: ]) AND
((Meritve.Meritev_datum)>=[Vnesi začetni datum: ] AND
(Meritev_datum)<=[Vnesi končni datum: ]) AND
(Kopalisca.Kopalisce_ID)=[Vnesi šifro kopališča]));
```

## Priloga 14: Primer skripte PHP za prikazovanje grafov

```
<?php
include ("jpggraph.php");
include ("jpggraph_line.php");
$jpggcache = APACHE_CACHE_DIR;
$graph_name = 'graf1.png';
// Podatki - povezava z MySQL bazo
    // povezava z bazo podatkov
$link = mysql_connect("localhost","boris", "") or die("Prijava ni uspela!
(napačno upor. ime ali geslo?)" );
mysql_select_db("DBmz");
    // SQL poizvedba
$sql = "SELECT Kopalisca.Kopalisce_ID, Boje.Boja_ID, Senzorji.Senzor_ID,
Senzorji.Vrsta_podatka_ID, Meritve.Meritev_datum, Meritve.Meritev_vrednost FROM
Vrsta_podatka RIGHT OUTER JOIN ((Kopalisca LEFT OUTER JOIN (Boje LEFT OUTER
JOIN Senzorji ON Boje.Boja_ID = Senzorji.Boja_ID) ON Kopalisca.Kopalisce_ID =
Boje.Kopalisce_ID) LEFT OUTER JOIN Meritve ON (Senzorji.Senzor_ID =
Meritve.Senzor_ID) AND (Senzorji.Boja_ID = Meritve.Boja_ID)) ON
Vrsta_podatka.Vrsta_podatka_ID = Senzorji.Vrsta_podatka_ID
WHERE (Senzorji.Vrsta_podatka_ID)= '1' AND (Kopalisca.Kopalisce_ID)= '526286')
ORDER BY Meritev_datum ASC";
// Definicija podatkov
$result=mysql_query($sql, $link);
    if ($myrow=mysql_fetch_array($result)) {
        do {
            $data[] = $myrow["Meritev_vrednosti"];
        }while ($myrow=mysql_fetch_array($result));
    }
// Postavitev grafa
$graph = new graph(400, 300, $graph_name, "auto");
$graph->SetScale('textlin');
// Definicija osi grafa
$graph->img->SetMargin(10, 10, 10, 10);
$graph->title->Set("streptokoki fekalnega izvora");
$graph->xaxis->title->Set("datum");
$graph->yaxis->title->Set("vrednost meritve na 100 ml");
// Definicija linije grafa
$lineplot=new LinePlot($data);
$lineplot->SetColor("blue");
// Dodajanje linije grafu
$graph->Add($lineplot);
// Prikaži graf
$graph->Stroke();
?>
```