

UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Manca Štrucelj

**Izdelava spletnega vmesnika za opravljanje poizvedb nad
Oracle podatkovno bazo**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: Aljaž Zrnec, PhD

Ljubljana, 2015

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco *GNU General Public License*, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses>.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo: Izdelava spletnega vmesnika za opravljanje poizvedb nad Oracle podatkovno bazo

Tema naloge: Sistem za nadzor misije (ang. Mission Control System - MCS) skrbi za komunikacijo s satelitom ESTCube-1, ki je zaenkrat edini aktivni satelit izdelan v okviru projekta ESTCube. Zaradi spremljanja tujih in načrtovanja izstrelitve novih ESTCube satelitov se je pojavila potreba po lažjem in bolj preglednem načinu iskanja shranjenih podatkov za nadaljnjo analizo. V okviru diplomske naloge najprej predstavite vse tehnologije, ki se uporabljajo za delovanje spletnega vmesnika ter uporabljene sisteme za hrambo podatkov. Implementirajte vmesnik, ki pridobi podatke, shranjene v Oracle podatkovni bazi in jih izvozi v csv datoteko. Podatke o satelitih in zemeljskih postajah, po katerih sortiramo, so shranjene v MongoDB. Predstavite vse aplikacijske vmesnike in knjižnice, ki jih boste v okviru izdelave vmesnika uporabili, in natančno analizirajte arhitekturo in delovanje uporabljenih podatkovnih baz.

Izjava o avtorstvu diplomskega dela

Spodaj podpisana Manca Štrucelj sem avtor diplomskega dela z naslovom:

Izdelava spletnega vmesnika za opravljanje poizvedb nad Oracle podatkovno bazo

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom Aljaža Zrneca, PhD,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 09.02.2015

Podpis avtorja:

Kazalo vsebine

POVZETEK

ABSTRACT

POGLAVJE 1	UVOD	1
POGLAVJE 2	PROJEKT ESTCUBE	2
2.1.	O ELEKTRIČNIH SOLARNIH JADRIH	3
2.2.	ESTCUBE-1	5
2.2.1.	<i>Znanstveni namen</i>	5
2.2.2.	<i>Orbita satelita</i>	5
2.2.3.	<i>Komunikacija</i>	5
2.2.4.	<i>Specifikacije</i>	6
POGLAVJE 3	PREDSTAVITEV PROBLEMATIKE IN PREDLAGANA REŠITEV	8
POGLAVJE 4	ARHITEKTURA MCS STREŽNIKA IN UPORABLJENE TEHNOLOGIJE	9
4.1.	ARHITEKTURA IN POSAMEZNI MODULI	10
4.2.	KNJIŽNICA HUMMINGBIRD	12
4.3.	OGRODJE SPRING	13
4.3.1.	<i>Jedro Spring ogrodja</i>	13
4.3.2.	<i>Ogradje za dostop do podatkov</i>	14
4.4.	ACTIVEMQ	15
4.5.	ATLASSIAN CROWD	16
4.6.	APACHE CAMEL	17
4.7.	APACHE MAVEN	18
POGLAVJE 5	UPORABLJENE PODATKOVNE BAZE	20
5.1.	PODATKOVNA BAZA ORACLE	20
5.1.1.	<i>Struktura podatkov</i>	20
5.1.1.1.	<i>Fizične strukture</i>	20
5.1.1.2.	<i>Logične strukture</i>	21
5.1.2.	<i>Programski jezik za nadzor baze</i>	22
5.1.3.	<i>Arhitektura Oracle baze v sistemu za nadzor misije</i>	22
5.2.	MONGODB	24
5.2.1.	<i>Struktura podatkov</i>	24
5.2.2.	<i>Indeksi v MongoDB</i>	24
5.2.3.	<i>Podvajanje</i>	25
5.2.4.	<i>Programski jeziki za nadzor baze</i>	25
5.2.5.	<i>MongoDB v MCS</i>	25
POGLAVJE 6	TEHNOLOGIJE, UPORABLJENE PRI IZDELAVI VMESNIKA	28
6.1.	AJAX	28
6.1.1.	<i>Dojo</i>	28
6.1.1.1.	<i>Asinhrona komunikacija</i>	29
6.1.1.2.	<i>Hramba podatkov na uporabnikovi strani</i>	29
6.1.1.3.	<i>Hramba podatkov na strežniku</i>	29
POGLAVJE 7	POMEN PAKETOV, SHRANJENIH V ORACLE PODATKOVNI BAZI	30

7.1.	BEACON	30
7.1.1.	<i>Struktura EstCube-1 beaona</i>	30
7.1.2.	<i>Vrednosti beaona v normalnem načinu</i>	31
7.2.	TNC PAKETKI.....	32
7.3.	AX.25	32
7.3.1.	<i>Struktura okvirja na EstCube-1 satelitu</i>	32
7.3.2.	<i>Opis polj v sporočilu</i>	33
POGLAVJE 8	PREDSTAVITEV VMESNIKA ZA OPRAVLJANJE POIZVEDB V ORACLE PODATKOVNI BAZI	34
8.1.	ZAKAJ SE VMESNIK ZA OPRAVLJANJE POIZVEDB POTREBUJE?	34
8.2.	GRAFIČNI UPORABNIŠKI VMESNIK ZA OPRAVLJANJE POIZVEDB	35
8.3.	PROGRAMSKA LOGIKA	38
POGLAVJE 9	ZAKLJUČEK.....	42

Seznam uporabljenih kratic

kratica	angleško	slovensko
JMS	Java Message Service	javanski sistem za prenos sporočil
XML	extensible markup language	razširljiv označevalni jezik
http	hypertext transfer protocol	protokol za prenos nadbesedil
URI	uniform resource identifier	enotni identifikator virov
RDBMS	relational database management system	sistem za upravljanje relacijskih podatkovnih baz
CSS	cascading style sheets	kaskadne stilske podloge
IT	information technology	informacijske tehnologije
ADCS	attitude determination and control system	sistem za nadzor in določanje pozicije satelita
CDHS	command and data handling system	sistem za upravljanje s podatki in ukazi
EPS	electrical power system	električni napajalni sistem
PL	payload	tovor

Povzetek

Cilj diplomske naloge je bil razvoj spletnega vmesnika za opravljanje poizvedb v Oracle podatkovni bazi. Glavni poudarek je omogočiti enostavno opravljanje poizvedb s preglednim prikazom rezultatov in možnostjo izvoza v csv format. V ta namen je bil razširjen modul Archiver, ki poleg shranjevanja arhivskih podatkov sedaj omogoča še iskanje po arhivu.

V prvem delu naloge je predstavljen projekt ESTCube, v sklopu katerega je bil razvit sistem za nadzor misije razprtja električnega solarnega jadra. Nato sledi predstavitev sistema za nadzor misije in tehnologij, uporabljenih za njegovo realizacijo ter predstavitev paketov, ki se uporabljajo pri komunikaciji s satelitom. V zadnjem sklopu je predstavljen spletni vmesnik za opravljanje poizvedb, za kaj se potrebuje in kako je realiziran.

Ključne besede: spletni vmesnik, Oracle, MongoDB, sistem za nadzor misije, Java, JavaScript, Dojo

Abstract

The purpose of this thesis was to produce a web interface for querying in Oracle database. The thesis mainly focuses on the most effective querying with explicit display of results and export to csv option. For that purpose we expanded Archiver class, which beside storing archive data now supports retrieving data from the archive.

The first part of the thesis describes ESTCube project which includes development of mission control system for deployment of electric solar sail. Follows a presentation of mission control system and technologies, used for its realisation and explanation of the packets, used for communication with the satellite. Last part of the thesis answers the question why query interface is needed and ends with a presentation of its implementation.

Keywords: web interface, Oracle, MongoDB, mission control system, Java, JavaScript, Dojo

Poglavje 1 Uvod

Hitremu razvoju računalniške tehnologije je v drugi polovici 20. stoletja sledil nagel razcvet vesoljske tehnologije. V petdesetih letih od izstrelitve prvega satelita¹ je uporaba vesoljske tehnologije postala dostopna širši javnosti, razvoj in izstrelitev satelitov pa iz leta v leto cenejša. Državam s svojim satelitom se bo v letu 2016 pridružila tudi Slovenija².

Razvoj odprtokodnih sistemov je zelo pomemben za izobraževalne namene, saj si univerze ne morejo privoščiti nakupa pravic za uporabo komercialnih projektov. Sistem za nadzor misije testiranja električnega solarnega jadra, uporabljen v projektu ESTCube, je razvit s strani študentov in vsebuje samo odprtokodne tehnologije. To omogoča uporabo in nadgradnjo sistema vsem, ki ga potrebujejo in ki želijo prispevati k cenejšemu razvoju vesoljskih tehnologij. V diplomski nalogi so predstavljene uporabljene tehnologije z opisom uporabljenih funkcionalnosti.

Glavni cilj diplomske naloge je bil razviti uporabniški vmesnik za opravljanje poizvedb v Oracle podatkovni bazi. Vmesnik je del sistema za nadzor misije, ki se uporablja za komunikacijo z ESTCube-1 satelitom, razvitim v estonskem Tartu Observatoriju. Komunikacija s satelitom poteka preko paketov dveh vrst; njihova funkcionalnost je predstavljena v nadaljevanju naloge.

Raziskovalci so že dlje časa prosili za vmesnik, ki jim bo pomagal pri pridobivanju paketov iz arhiva. Avgusta 2014 jim je bila želja izpolnjena in del sistema za nadzor misije je postal vmesnik, ki omogoča vnos iskalnih parametrov in prikaže vse pakete, ki jim ustrezajo, ter jih prikaže v tabeli, urejene po času sprejema. Rezultat iskanja je mogoče izvoziti v formatu csv, kar omogoča uvoz paketkov v druge aplikacije za nadaljnjo analizo.

Vmesnik za pridobivanje podatkov iz Oracle podatkovne baze sem razvila v času opravljanja prakse, ki sem jo naredila v okviru projekta Erasmus. Prvi mesec prakse sem iskala napake v sistemu in opravljala enostavnejše naloge. V tem času sem se dodobra spoznala s sistemom in nekaterimi izmed uporabljenih tehnologij. Zaradi velike želje po vmesniku za opravljanje poizvedb sem prevzela nalogo načrtovanja in realizacije vmesnika.

¹ Sputnik 1, izstrelitev leta 1957

² Razvoj se je pričel leta 2010 v Centru odličnosti vesolje – SI v sodelovanju z Univerzo v Torontu.

Poglavje 2 Projekt ESTCube

ESTCube projekt je nastal pod okriljem Tartu Observatorija³, neprofitne raziskovalne ustanove, ki deluje znotraj estonskega Ministrstva za izobraževanje in razvoj. Je naslednik Univerzitetnega observatorija v Tartuju in estonskega meteorološkega observatorija. Primarne aktivnosti instituta so raziskava in eksperimentalni razvoj za namen promocije znanosti na področjih astronomije, oddaljenega zaznavanja in vesoljske tehnologije, zagotavljanje storitev, ki temeljijo na raziskovalnem delu, ter izobraževanje mladih znanstvenikov.

ESTCube projekt se je pričel poleti leta 2008, njegov cilj pa je vključiti študente v vesoljske projekte. Dodatni cilj je spodbuditi razvoj vesoljske in visokotehnološke industrije z izobraževanjem strokovnjakov in širjenjem znanja o vesoljski tehnologiji [26], [15].

Oddelki Tartu Observatorija, ki sodelujejo pri ESTCube projektu, so:

- oddelek za astrofiziko, skupina za teoretično astrofiziko,
- oddelek za kozmologijo,
- oddelek za oddaljeno zaznavanje,
- oddelek za vesoljsko tehnologijo,
- laboratorij za testiranje in kalibracijo [30].

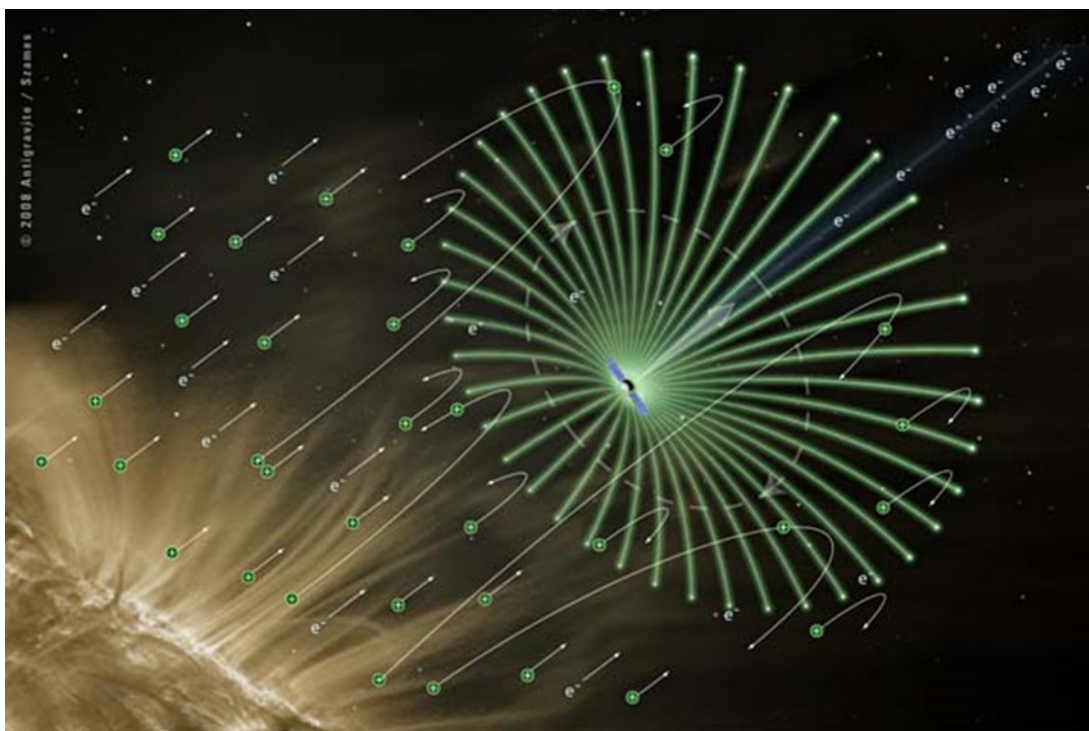
³ Tartu Observatorij je največji estonski astronomski observatorij. Ustanovljen je bil leta 1802. Leta 1946 se je odcepil od Univerze v Tartuju in prešel pod okrilje estonske Akademije za znanost. Leta 1963 je bila končana gradnja nove stavbe, v sklopu katere deluje največji optični teleskop v severnoevropskih državah.

2.1. O električnih solarnih jadrih

Leta 2006 je finski znanstvenik Pekka Janhunen predstavil novo pogonsko tehnologijo za premikanje vesoljskih plovil, ki temelji na interakciji med hitro gibajočimi ioni v solarnem vetru in električnim poljem, ki ga generira satelit.

Električno solarno jadro polnega obsega je sestavljeno iz 50 do 100 prevodnih žic, imenovanih tetri (ang. tether), dolgih 20 kilometrov in z debelino 25 mikronov. Posamezni teter tehta samo nekaj sto gramov in zavzema zelo majhno prostornino. Vesoljsko plovilo vsebuje elektronski top, ki ga napaja solarna energija in vzdržuje plovilo ter tetre na visokem pozitivnem potencialu (20kV). Električno polje se razteza nekaj deset metrov okoli tetra v okoliško solarno plazmo.

Dinamični pritisk solarnega vetra se spreminja, njegova povprečna vrednost pa je 2 nPa na Zemljini distanci od sonca, kar je približno 5000-krat šibkejše od solarnega radioaktivnega pritiska. Zaradi velike površine in zelo majhne teže posamezne kovinske žice je električno solarno jadro kljub temu še vedno zelo učinkovito. Protoni v solarnem vetru se odbijajo od pozitivno nabitih tetrov in s tem povečujejo rotacijsko hitrost plovila. Nasprotno pa naboj tetrov anione privlači, ki tako dodatno ustvarjajo električni tok na njih - Slika 1, zato električni top ne deluje s konstantno močjo, ampak po potrebi, da vzdržuje željeni potencial. Smer premikanja se uravnava z napetostjo posameznih tetrov; kjer je napetost manjša, je tudi pospešek manjši [11], [12].



Slika 1: Prikaz delovanja električnih solarnih jader.

Posamezni teter proizvede več kvadratnih kilometrov efektivne površine, ki je zmožna pridobiti okoli 10 milinewtnov sile. Tako bi 1000-kilogramsko plovilo, opremljeno s 100 žicami proizvedlo pospešek okoli 1 mm/s^2 , po enem letu delovanja pa bi bila končna hitrost okoli 30 km/s. Vesoljska plovila z manjšim tovorom bi lahko potovala hitro; potovanje do Plutona bi tako vzelo manj kakor 5 let [12].

Električna solarna jadra bi se lahko uporabljala tudi za vračanje satelitov iz orbite. Če je naboj tetrov nasproten povprečnemu naboju solarnega vetra, večji del ionov tetri privlačijo in na satelit delujejo v nasprotni smeri trenutnega premikanja [3].

2.2. ESTCube-1

ESTCube-1 je prvi estonski satelit, izdelan s strani študentov Univerze v Tartuju, Estonske letalske akademije, Tehnološke univerze v Tallinu in Univerze za znanost o življenju. Projekt so nadzorovali raziskovalci Tartu Observatorija in profesorji Univerze v Tartuju.

Zasnova, razvoj in izdelava satelita so potekali od začetka projekta v letu 2008 do izstrelitve v Zemljino orbito sedmega maja 2013. Od takrat se satelit aktivno uporablja v raziskovalne in izobraževalne namene [7].

2.2.1. Znanstveni namen

Glavna naloga satelita je razprtje 10 metrov dolgega in 20 do 50 mikrometrov širokega električnega solarnega jadra, slikanje le-tega in izmeritev električne solarne energije, ki jo proizvede. Razprtje jadra se izvede s pomočjo centrifugalne sile, ki jo povzročijo elektromagnetne tuljave. Za razvitje jadra se mora rotacijska hitrost satelita povečati na dva obrata na sekundo. Uspešnost razprtja se bo preverila z vgrajeno kamero in s preverjanjem hitrosti rotacije. Če se hitrost rotacije satelita zmanjša, se je teter uspešno sprostil.

Sekundarni cilj je bil slikanje Zemlje; v enem letu je bilo posnetih in shranjenih več kot 200 slik [7], [2].

2.2.2. Orbita satelita

Satelit je vtirjen v orbito na višini 670 kilometrov, potuje s hitrostjo 7.5 kilometrov na sekundo in vsak dan sedemkrat obkroži Zemljo [15].

2.2.3. Komunikacija

Komunikacija s satelitom je izvedena z uporabo dveh frekvenc, ki sta registrirani pri Mednarodni zvezi radio amaterjev:

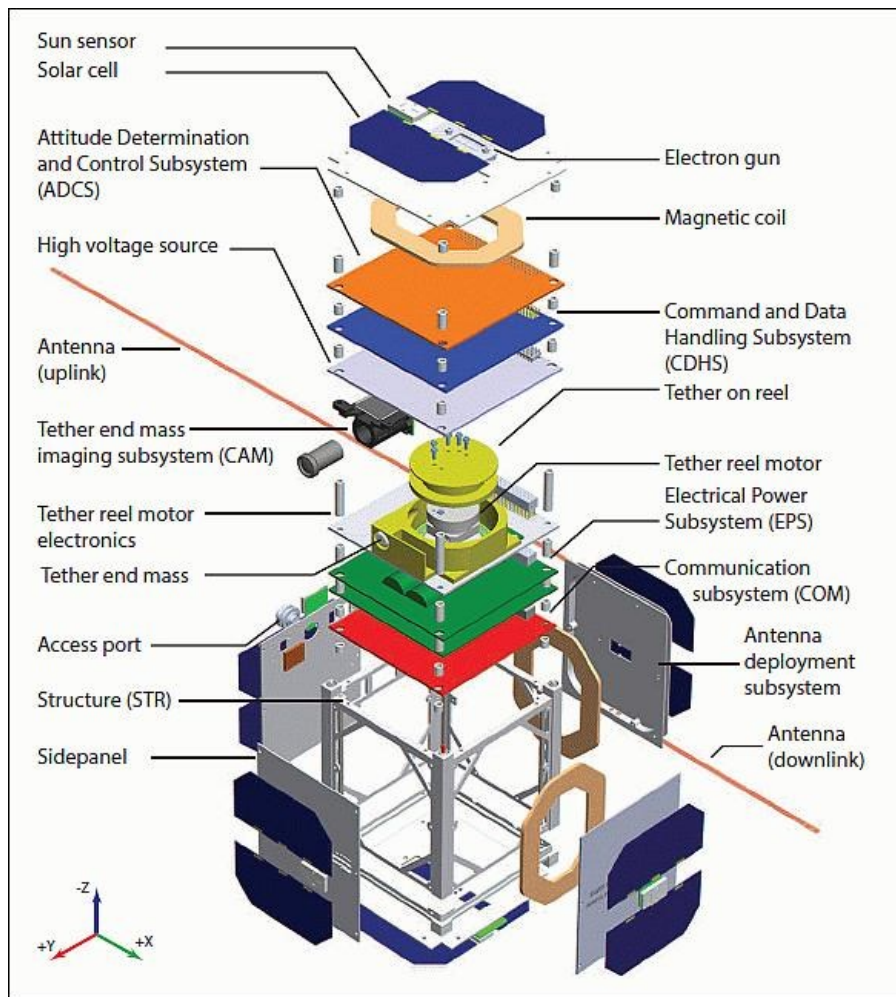
- 437.250 MHz in
- 437.505 MHz.

Periodična, a počasna, komunikacija je narejena z 18 besedami na minuto na frekvenci 437,250MHz. Na tej frekvenci se prenašajo najbolj pomembni parametri satelita, vsake tri do deset minut. Za hiter prenos se uporablja modulacija s frekvenčnimi skoki na frekvenci 437,505MHz, s povezovalno hitrostjo 9600 baudov, po kateri se prenašajo paketi standarda AX.25. Hitra povezava se uporablja le, kadar satelit prejme določen ukaz. Ukazi se pošiljajo preko 145MHz amaterske pasovne širine [7].

2.2.4. Specifikacije

Satelit je kockaste oblike dimenzij 10cm x 10cm x 10cm z 1.05 kg mase, izmerjene pred izstrelitvijo [26].

Slika 2 prikazuje vse dele satelita in vsebovane module, kakor so razporejeni na satelitu.



Slika 2: Sestavni deli satelita [26].

Sistemi na satelitu:

- ADCS: sistem za določitev in nadzor pozicije satelita,
- CAM: kamera za zajem slik Zemlje in razprtega tetra,
- CDHS: sistem za upravljanje s podatki in ukazi, glavni računalnik na satelitu,
- COM: komunikacijski sistem,
- EPS: sistem, ki skrbi za napajanje na satelitu, pridobivanje in distribucijo električne energije,
- PL: eksperimentalni modul, ki vsebuje teter in vse ostalo, potrebno za primarno misijo [26], [16].

Poglavje 3 Predstavitev problematike in predlagana rešitev

Pred izstrelitvijo satelita v orbito so bili opravljeni obsežni laboratorijski testi, katerih cilj je bil simulirati razmere v brezračnem prostoru in preveriti pravilnost delovanja ter vzdržljivost vseh modulov. Ker ni mogoče upoštevati vseh dejavnikov, ki vplivajo na delovanje satelita v Zemljini orbiti, so po izstrelitvi potrebna dodatna testiranja in posodobitve programske opreme.

Edini način, na katerega se lahko opazuje stanje modulov in opravi nadaljnje diagnostike, je preko podatkov, ki jih preko radijske povezave posreduje satelit. Po enem letu od ustalitve satelita v orbiti je bila potrebna kalibracija senzorjev, popravki sistema za meritev in nadzor višine, pokazale pa so se tudi težave pri dovajanju električne energije v posamezne module [29].

Vsa komunikacija s satelitom se shrani v arhive v sistemu za nadzor misije (ang. Mission Control System – MCS). Uporabljeni sta dve vrsti paketov: beacon nam poda informacije o stanju sistema, AX.25 paketi pa se uporabljajo za prenos ukazov satelitu in odgovorov na njih. Do sedaj so raziskovalci podatke za nadaljnjo analizo iskali neposredno po Oracle podatkovni bazi s pomočjo poizvedb, ker pa je to zelo zamudno in zahteva predznanje, so izrazili željo po grafičnem vmesniku za opravljanje poizvedb in prikaz rezultatov.

Zahteve uporabnikov:

- iskanje vseh vrst paketov, shranjenih v arhivu,
- hitro in intuitivno definiranje iskalnih parametrov,
- izvoz rezultata iskanja v csv format.

Skupaj z analitiki, ki so končni uporabniki, se je zasnoval vmesnik, ki jim bo omogočal čim lažje iskanje in sortiranje podatkov po vseh parametrih ter izvoz v format, primeren za nadaljnjo obdelavo.

Poglavje 4 Arhitektura MCS strežnika in uporabljene tehnologije

Sistem za nadzor misije služi kot programska podpora projektu ESTCube. MCS omogoča spremljanje orbite satelita, pošiljanje ukazov in sprejemanje podatkov s satelita. Vse podatke je možno spremljati v realnem času, po novem pa je omogočen tudi pregled prejetih podatkov od začetka misije. MCS omogoča tudi spremljanje in nadzor zemeljskih postaj, ki služijo za komunikacijo in upravljanje s satelitom.

Razvoj odprtokodnega sistema za nadzor misije se je pričel septembra 2011 v sodelovanju Univerze v Tartuju in podjetja Logica Estonia, ki je sedaj del skupine CGI⁴. V okviru CGI-ja je bila razvita odprtokodna infrastruktura Hummingbird, ki je hrbtenica ESTCube sistema za nadzor misije [6]. Razvoj sistema je potekal dve leti in se še nadaljuje. V tem času so se zamenjale štiri skupine študentov, ki so nadaljevale delo prejšnjih skupin. Znanje razvijalcev se iz semestra v semester spreminja, vsaka nadaljnja skupina je običajno manj izkušena od prejšnjih, saj so predhodniki opravili kompleksnejše naloge.

MCS je prvi odprtokodni sistem, ki temelji na spletnih tehnologijah. Odprtokodnost omogoča ostalim študentom, da ga uporabijo v svojih misijah. Ker sistem MCS temelji na spletnih tehnologijah, lahko dostopamo do njega s poljubne lokacije, zahtevan je samo dostop do svetovnega spleta. Tako lahko ustvarimo svetovno mrežo zemeljskih postaj in s tem dosežemo pogostejšo komunikacijo z ESTCube-1 in ostalimi sateliti, shranjenimi v sistemu [4], [5].

⁴ CGI Group Inc. je multinacionalno podjetje s sedežem v Kanadi, ki se ukvarja s svetovanjem in ponujanjem rešitev na področju informacijskih tehnologij [32].

4.1. Arhitektura in posamezni moduli

Najpomembnejši del sistema za nadzor misije je spletni strežnik, kjer s pomočjo grafičnega vmesnika dostopamo do vseh ostalih modulov. Avtentikacija na strežniku poteka preko Atlassian Crowd strežnika.

Večina modulov, potrebnih za spremljanje in upravljanje satelita, je vzeti iz knjižnice Hummingbird.

Uporabljene funkcionalnosti iz knjižnice Hummingbird:

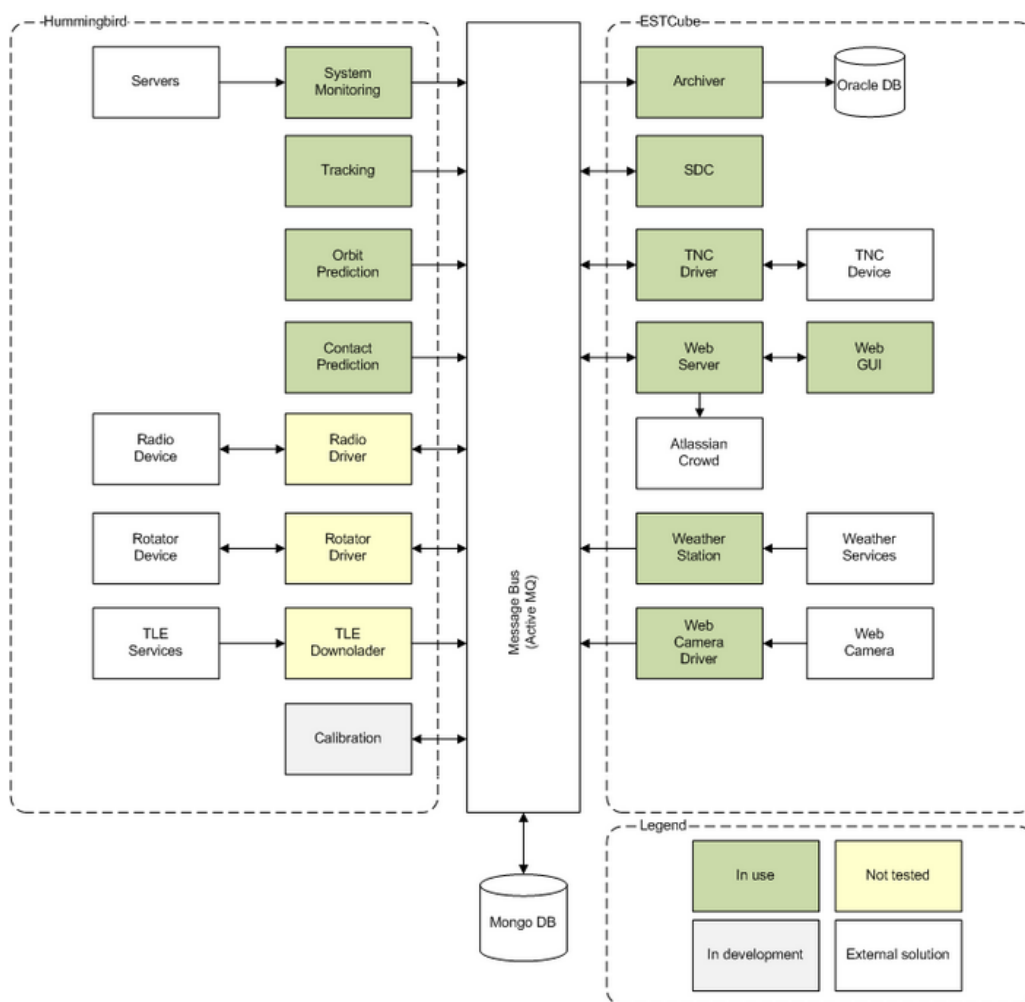
- opazovanje sistema,
- sledenje satelita,
- predvidevanje orbite satelita,
- predvidevanje naslednjega kontakta s satelitom,
- gonilniki za radio,
- gonilniki za rotator anten,
- prenašalnik TLE (ang. Two-Line Element) paketov.

Ostale module je razvila ESTCube skupina. Najbolj pomemben je gonilnik za TNC (ang. Terminal Node Controller) napravo, ki skrbi za pošiljanje in prejemanje TNC paketov. Poti za komunikacijo med moduli so implementirane z uporabo programske knjižnice Apache Camel, sporočila pa se pošiljajo s pomočjo ogrodja ActiveMQ. MongoDB baza se uporablja za hranjenje sistemskih podatkov. Podatki, shranjeni v bazi MongoDB:

- ukazi, ki se preko AX.25 paketov pošiljajo satelitu,
- zemeljske postaje,
- sateliti, s katerimi je bila vzpostavljena komunikacija,
- podatki o vzpostavitvi povezave s satelitom,
- podatki o izgubi povezave s satelitom,
- podatki o orbiti in hitrosti premikanja satelita,
- podatki o parametrih, ki sestavljajo beacon,
- podatki o zagonu modulov sistema za nadzor misije.

Slika 3 prikazuje module, uporabljene v sistemu za nadzor misije. Ločeno so prikazani moduli iz Hummingbird knjižnice in moduli, razviti znotraj projekta ESTCube. Komunikacija med moduli poteka preko Apache ActiveMQ posrednika sporočil. Prikazani so vsi moduli, ki so:

- že bili implementirani (z zeleno barvno),
- že implementirani, vendar niso še dokončno stestirani (z rumeno barvo),
- ki so trenutno v razvoju (z modro barvo),
- uporabljene zunanje rešitve (z belo barvo).



Slika 3: Arhitektura sistema za nadzor misije

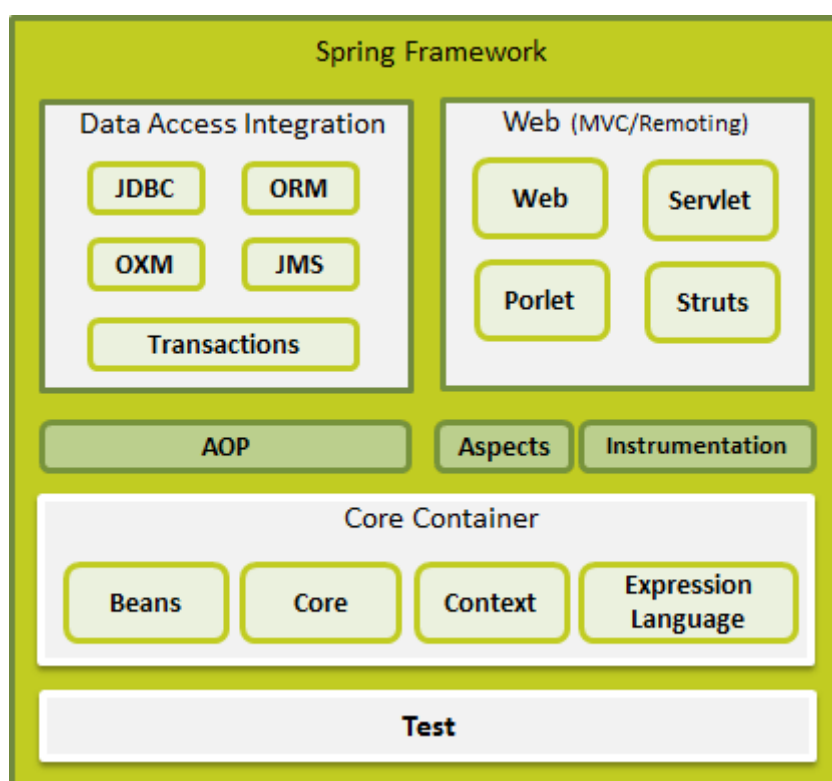
4.2. Knjižnica Hummingbird

Hummingbird je odprtokodna knjižnica za spremljanje in nadzor oddaljenih sistemov, kot so sateliti, rakete ali zemeljske postaje. Priskrbi prilagodljivo integracijsko okolje s standardnimi funkcionalnostmi, kot so npr. kodiranje/dekodiranje, kalibracija in navigacija. Do sedaj so bili v okviru omenjene knjižnice razviti moduli za:

- preverjanje telemetrije in kalibracijo,
- konfiguracijo, preverjanje in avtomatsko pošiljanje ukazov,
- pisanje skript z jeziki JavaScript in Python z možnostjo avtomatizacije,
- napovedovanje kontaktov s satelitom,
- spremljanje stanja sistema,
- obveščanje o dogodkih,
- kodiranje/dekodiranje paketov [17].

4.3. Ogradje Spring

Javansko ogradje Spring je odprtokodno, aplikacijsko ogradje in vsebuje več kot 10 različnih modulov, ki zagotavljajo različne storitve. Osnovne funkcionalnosti lahko uporablja katerakoli Javanska aplikacija, dodatne razširitve pa so namenjene razvoju spletnih aplikacij na platformi Java EE. V nadaljevanju so opisani moduli, ki so uporabljeni v sistemu za nadzor misije.



Slika 4: Moduli Spring ogradja [27].

4.3.1. Jedro Spring ogradja

Jedro Spring ogradja vsebuje 4 različne module:

- **osnovni modul** – vsebuje temeljne dele ogradja, vključno z inverzijo nadzora in vstavljanjem odvisnosti,
- **modul fižolčki** (ang. bean) – poskrbi za napredno implementacijo tovarniških metod (ang. factory methods), ki omogočajo kreiranje objekta brez definiranja razreda objekta,
- **modul kontekst** – zgradi projekt na podlagi zgornjih dveh modulov in omogoča dostop do kateregakoli definiranega objekta,

- **modul izrazni jezik** – priskrbi zmogljiv jezik za opravljanje poizvedb in manipulacijo objekta v realnem času.

Glavna funkcionalnost Spring ogrodja je inverzija nadzora, ki nam omogoča dosledno konfiguriranje ter upravljanje z Javanskimi objekti, uporabljajoč preslikave. Osnovni modul poskrbi za upravljanje z življenjskimi cikli objektov, t. j. jih ustvari, pokliče njihove inicializacijske metode in konfigurira te objekte z njihovim povezovanjem.

Kreiranim objektom se reče tudi *upravljani objekti* ali *fižolčki*. Lahko jih konfiguriramo v XML datoteki ali z določenimi Javanskimi notacijami pri definiranju razredov. Tako izvorni kodi dodamo potrebne podatke za kreacijo fižolčkov.

Upravljane objekte lahko pridobimo s pregledom odvisnosti ali z vstavljanjem odvisnosti. Pri pregledu odvisnosti poizvedovalec komunicira neposredno z vsebovalnikom objekta in zahteva specifičen objekt točno določenega tipa. Pri vstavljanju odvisnosti pa vsebovalnik vstavi objekte glede na poimenovanje s pomočjo konstruktorjev, lastnosti ali tovarniške metode.

4.3.2. Ogrodje za dostop do podatkov

Ogrodje za dostop do podatkov razvijalcem omogoča lažji dostop do podatkov v podatkovnih bazah. Podpira uporabo vseh pogosto uporabljenih ogrodij za dostop do podatkov v povezavi z Java: JDBC, Hibernate, JDO, Oracle TopLink, Apache OJB, Apache Cayenne, ipd.

Za vsa podprta ogrodja Spring omogoča:

- **upravljanje z viri**, t. j. avtomatično pridobivanje in sproščanje podatkovnih sredstev,
- **upravljanje z izjemami**, t. j. preslikovanje izjem, povezanih z dostopom do podatkov, v Springovo hierarhijo,
- **sodelovanje pri transakcijah**, t. j. pregledno sodelovanje pri tekočih transakcijah,
- **pridobivanje sredstev**, t. j. pridobivanje objektov podatkovne baze iz ovojnice bazena povezav
- **abstraktnost** za upravljanje z BLOB in CLOB objekti.

Za vsako podprto ogrodje Spring priskrbi predloge razredov, s katerimi nam omogoči uporabo vseh funkcionalnosti [28], [27].

4.4. ActiveMQ

Apache ActiveMQ je odprtokodni posrednik sporočil, napisan v Javi z JMS odjemalcem, in omogoča komunikacijo z več kot enim odjemalcem ali strežnikom. Upravljanje s sporočili se izvaja s pomočjo povezovanja računalnikov v gruče ali z možnostjo uporabe katerekoli podatkovne baze za zagotavljanje JMS vztrajnosti poleg virtualnega spomina in predpomnilnika.

Nekaj funkcionalnosti:

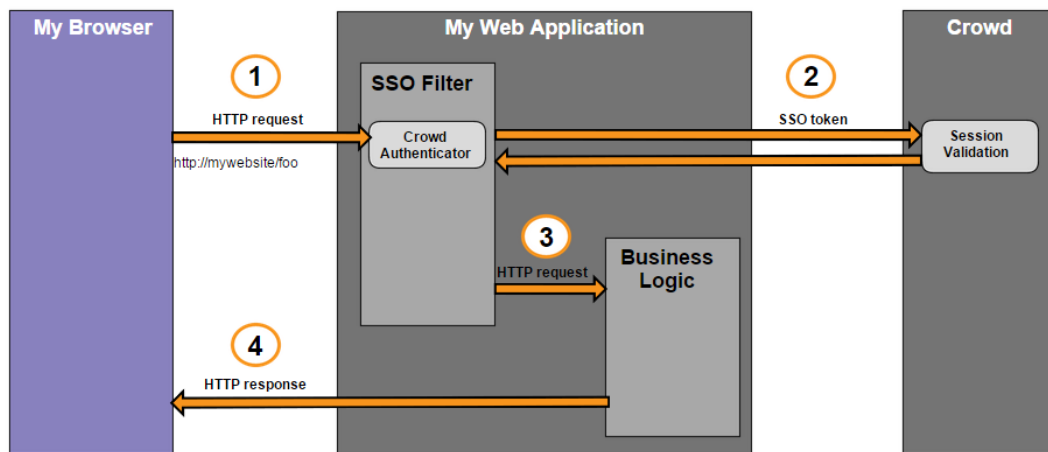
- podpora različnih medjezičnih klientov in protokolov,
- polna podpora poslovnih integracijskih vzorcev,
- podpora različnih naprednih funkcionalnosti, kot so skupine za sporočila, virtualne destinacije in združene destinacije,
- enostavna uporaba skupaj s Spring ogrodjem in možnost konfiguracije v Springovih XML konfiguracijskih mehanizmih,
- načrtovan za visoko zmogljive gruče in komunikacijo med klienti in strežniki ter komunikacijo, ki temelji na omrežnih vrstnikih (ang. peer based),
- REST aplikacijski vmesnik omogoča jezikovno nevtrarno sporočanje za spletne aplikacijske vmesnike,
- podpira asinhroni JavaScript in XML (ang. Asinhron JavaScript in XML – Ajax), ki omogoča, da spletni brskalniki postanejo del sporočila [8].

4.5. Atlassian Crowd

Crowd poskrbi za enkratni vpis (ang. Single Sign-On – SSO) v različne aplikacije. Uporabnik se vpiše samo enkrat in lahko dostopa do različnih aplikacij znotraj iste domene brez ponovnega vpisa.

Glavna funkcionalnost strežnika Crowd omogoča enkratni vpis z uporabo brskalnikovih piškotkov, ker pa brskalnik omejuje dostop do piškotkov samo gostiteljem znotraj iste domene, morajo biti vse aplikacije v isti domeni.

Slika 5 prikazuje, kako gre HTTP zahteva najprej skozi SSO filter, kjer pridobi SSO žeton, s katerim na Crowd strežniku preveri, ali je uporabnik že vpisan. Scenarij na sliki predvideva, da je uporabnik že vpisan, zato se HTTP zahteva posreduje aplikaciji, ki ustvari in pošlje odgovor brskalniku [25].



Slika 5: Prikaz avtentikacije s sistemom Crowd

4.6. Apache Camel

Camel nam omogoča, da definiramo usmerjevalne poti in pravila za posredovanje v različnih domensko specifičnih jezikih, kot so Fluent API, Spring ali Blueprint XML konfiguracije datoteke. Podpira ogrodja za vstavljanje odvisnosti, kot so Spring, Blueprint in Guice. Ima tudi obsežno podporo za testiranje usmerjevalnih poti.

Camel s pomočjo URI-jev prenaša sporočila v katerem koli formatu, naj bo to HTTP, ActiveMQ, JMS, JBI ... Apache Camel je majhna knjižnica z minimalnimi odvisnostmi za lažjo integracijo znotraj Javanskih aplikacij. Znotraj istega programa lahko uporabljamo različne tipe sporočil in dostopamo do različnih tipov komponent. Omogoča definiranje fizičkov (ang. bean biding), kjer povemo, katere metode naj se izvedejo in kako naj se sporočilo prevede v parametre razreda [9].

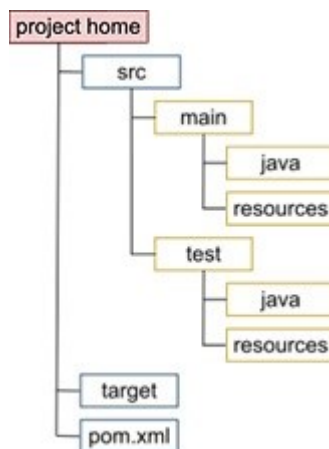
4.7. Apache Maven

Maven je orodje za avtomatsko izgradnjo projekta, primarno namenjeno Javanskim projektom in ima dva aspekta gradnje aplikacije: opiše, kako je aplikacija sestavljena, in definira vse njene odvisnosti. XML datoteka opiše, kako naj se projekt zgradi, njegove odvisnosti od zunanjih modulov in komponent, vrstni red izgradnje, direktorije in zahtevane vtičnike. Dinamično shrani Javine knjižnice in Mavenske vtičnike iz enega ali več repozitorijev.

Mavenov glavni cilj je, da razvijalcem omogoči razumevanje celotnega stanja projekta v najkrajšem času. V ta namen Maven stremi k:

- poenostavljanju gradnje projekta,
- enotnemu sistemu za gradnjo projektov,
- generaciji kvalitetnih podatkov o projektu,
- generaciji smernic za najboljši način razvoja,
- razvoju transparentnih migracijskih funkcionalnosti.

Objektni model projekta (ang. Project Object Model – POM) vsebuje celotno konfiguracijo za posamezni projekt. Splošna konfiguracija vsebuje ime projekta, lastnika projekta in odvisnosti od ostalih projektov. Lahko definiramo tudi posamezne fraze v postopku gradnje, ki se implementirajo kakor vtičniki.



Slika 6: Struktura Apache Maven projekta

Večji projekti so navadno razdeljeni v več modulov ali podprojektov, od katerih vsak vsebuje svojo POM datoteko. Slika 6 prikazuje strukturo posameznega podprojekta. Ko je projekt razdeljen na

module, lahko definiramo korenski POM, ki prevede vse module s samo enim ukazom. POM datoteka lahko tudi podeduje konfiguracijo drugih POM datotek. Privzeto vse POM datoteke podedujejo Super Pom, ki priskrbi osnovno konfiguracijo, npr. izvorne direktorije, vtičnike, ... [20].

Poglavje 5 Uporabljene podatkovne baze

To poglavje zajema pregled ključnih lastnosti uporabljenih podatkovnih baz in glavnih razlik med njimi. V sistemu za nadzor misije sta uporabljena dva tipa podatkovnih baz – relacijske baze in dokumentno orientirane baze. Razlikujejo se v strukturi podatkov, v mehanizmu proti izgubi podatkov in v podprtih jezikih za delo z bazo. Predstavljene so glavne lastnosti posamezne baze in njihova uporaba v MCS sistemu.

5.1. Podatkovna baza Oracle

Oracle podatkovna baza (Oracle RDBMS) je relacijski sistem za upravljanje s podatkovnimi bazami.

Baza je zbirka podatkov, ki se obravnavajo kot celota, katere namen je shranjevanje in pridobivanje podatkov. Strežnik podatkovne baze skrbi za upravljanje z podatki. Ponavadi strežnik skrbi za velike količine podatkov v večuporabniškem okolju; tako lahko več uporabnikov hkrati dostopa do istih podatkov.

Oracle RDBMS je prvi sistem, namenjen uporabi v poslovnih sistemih in je najbolj prilagodljiv ter učinkovit sistem za upravljanje s podatki [2].

5.1.1. Struktura podatkov

Podatkovna baza ima ločene logične in fizične podatkovne strukture, kar omogoča spreminjanje strojne opreme brez sprememb v logični podatkovni strukturi. Podatki se logično hranijo v podatkovnih tabelah in fizično v podatkovnih datotekah [2].

5.1.1.1. Fizične strukture

Vsaka Oraclova podatkovna baza ima eno ali več fizičnih *podatkovnih datotek*, ki vsebujejo vse podatke v bazi. Podatki logičnih struktur baze, kot so tabele in indeksi, so fizično shranjeni v pripadajoči datoteki.

Značilnosti podatkovne datoteke so:

- datoteka lahko pripada samo eni podatkovni bazi,
- datoteka se lahko avtomatično razširi, ko v bazi zmanjka prostora,
- ena ali več datotek tvori logično strukturo baze, imenovano podatkovno področje.

Vsaka baza ima pripadajočo **kontrolno datoteko**, ki povzema fizično strukturo baze. Vsebuje ime podatkovne baze, imena in lokacije podatkovnih ter datotek z dnevnikom in časovni žig o ustvaritvi baze. Zaradi zaščite proti napakam pri delovanju kontrolne datoteke lahko Oracle sočasno vzdržuje več enakih kopij le-te.

Vsaka baza prav tako vsebuje dve ali več **datotek z dnevnikom**, ki vsebujejo vse spremembe, ki so se zgodile nad bazo. Če pride pri shranjevanju podatkov v podatkovne datoteke do nesreče, se podatke lahko pridobi iz datoteke z dnevnikom, kar prepreči njihovo izgubo.

Oracle omogoča avtomatsko shranjevanje datotek z dnevnikom v **arhivske datoteke**.

Datoteke s parametri vsebujejo seznam konfiguracijskih parametrov za določeno instanco in bazo.

Varnostna kopija nadomesti datoteke, ki so se pokvarile zaradi napake v delovanju spominskega medija ali zaradi uporabniške napake [24].

5.1.1.2. Logične strukture

V okviru relacijske podatkovne baze so podatki na logičnem nivoju predstavljeni z uporabo tabel. V okviru ene tabele se shranjujejo podatki – vrstice, ki opisujejo enake objekte.

Najmanjša logična enota Oraclovih baz so **podatkovne strani**. Posamična stran ustreza določenemu številu bytov fizičnega spomina na disku. Standardna velikost strani se nastavi v inicializacijskih parametrih.

Naslednji nivo logičnega spomina je **področje**, ki vsebuje določeno število zaporednih podatkovnih strani in hrani specifičen tip podatkov.

Področja se povezujejo v **segmente**, ki so dodeljeni določeni logični strukturi. Ko v segmentu zmanjka prostora, se mu avtomatično dodeli novo področje, s tem pa nov spominski prostor. Različni tipi segmentov:

- **podatkovni segment**, ki hrani podatke tabele,

- **indeksni segment**, ki hrani podatke o indeksih v tabeli,
- **začasni segment**, ki hrani začasno tabelo pri opravljanju zahtevnih SQL poizvedb,
- **povrnitveni segment**, ki pomaga pri povrnitvi baze na prejšnje stanje [24].

5.1.2. Programski jezik za nadzor baze

Jedro Oraclovega aplikacijskega razvoja sta strukturirani poizvedovalni jezik (ang. Structured Query Language – SQL) in proceduralni jezik/strukturirani poizvedovalni jezik (ang. Procedural Language/Structured Query Language – PL/SQL). Večina zalednih delov sistemov lahko izvaja SQL, spletne aplikacije pa ga uporabljajo za dostop do podatkovnih baz. Je enostaven in lahko razumljiv, poenoten podatkovni model. V veliko aplikacijah je uporabljen samostojno, lahko pa ga kličemo tudi s pomočjo Java, Oraclovega klicnega vmesnika (Oracle Call Interface – OCI), XML SQL orodja (ang. XML SQL Utility – XSU), ... [24].

5.1.3. Arhitektura Oracle baze v sistemu za nadzor misije

V sistemu za nadzor misije se uporablja zastonska verzija sistema Oracle – Oracle Database Express Edition. Vsak tip paketkov ima v Oracle podatkovni bazi rezervirani dve tabeli: <name>_FRAMES in <name>_FRAME_DETAILS. Tabela FRAMES poleg razčlenjenih in surovih podatkov vsebuje tudi podatek o času sprejema paketa, v FRAME_DETAILS tabeli pa se hranijo podrobnosti o prenosu preko JMS (ang. Java Message Service). Povezovalni atribut med tabelama je ID paketa. Vsak prejeti paket ima približno 20 pripadajočih vrstic v FRAME_DETAILS tabeli.

Za primer si pogledjmo način arhiviranja AX.25 paketov in strukturo tabel (Slika 7 in Slika 8): Polje INFO vsebuje surove podatke, ostala polja pa razčlenjene podatke poslanih podatkov, ID številko prejetega paketa in čas prejema le-tega. Vsaka vrstica v tabeli AX25_FRAME_DETAILS je sestavljena iz imena shranjene lastnosti, njene vrednosti, ID paketa, ki mu pripada, in časa kreacije lastnosti (v sistemu za nadzor misije).

1	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID	NUMBER(15,0)	No	(null)	1	(null)
2	DIRECTION	VARCHAR2(10 BYTE)	No	(null)	2	(null)
3	RECEPTION_TIME	NUMBER(15,0)	Yes	(null)	3	(null)
4	SATELLITE	VARCHAR2(50 BYTE)	Yes	(null)	4	(null)
5	DESTADDR	RAW	Yes	(null)	5	(null)
6	SRCADDR	RAW	Yes	(null)	6	(null)
7	CTRL	NUMBER(3,0)	Yes	(null)	7	(null)
8	PID	NUMBER(3,0)	Yes	(null)	8	(null)
9	INFO	RAW	Yes	(null)	9	(null)
10	FCS	RAW	Yes	(null)	10	(null)
11	ERROR_BITMASK	NUMBER(5,0)	No	(null)	11	(null)
12	CREATED	TIMESTAMP(6) WITH TIME ZONE	No	(null)	12	(null)

Slika 7: Struktura AX25_FRAME tabele

1	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID	NUMBER(15,0)	No	(null)	1	(null)
2	NAME	VARCHAR2(200 BYTE)	No	(null)	2	(null)
3	VALUE	VARCHAR2(4000 BYTE)	No	(null)	3	(null)
4	FRAME_ID	NUMBER(15,0)	No	(null)	4	(null)
5	CREATED	TIMESTAMP(6) WITH TIME ZONE	No	(null)	5	(null)

Slika 8: Struktura AX25_FRAME_DETAILS tabele

5.2. MongoDB

MongoDB je odprtokodna, dokumentno orientirana podatkovna baza. Klasificirana je kot NoSQL baza in se izogiba tradicionalni relacijsko orientirani strukturi. Posnema JSON (ang. JavaScript Object Notation) strukturo z dinamičnimi shemami in tako poenostavlja integracijo podatkov znotraj različnih aplikacij. Ne omogoča stikov tabel, kar je ključna lastnost relacijskih podatkovnih baz, ima pa zato boljše podporo definiranju primarnih in sekundarnih indeksov. MongoDB vsebuje podporo za različne, pogosto uporabljene programske jezike in razvojna okolja že v uradnih gonilnikih.

5.2.1. Struktura podatkov

Podatki v MongoDB imajo zelo prilagodljive sheme. Za razliko od SQL podatkovnih baz, kjer moramo vnaprej določiti shemo tabele, preden lahko shranjujemo podatke, zbirke v MongoDB nimajo opredeljene strukture. Ta prilagodljivost nam olajšuje povezovanje dokumenta z določeno entiteto ali objektom. Vsak dokument znotraj zbirke ima lahko enaka podatkovna polja kakor predstavljena entiteta, četudi se podatkovni tipi atributov v dokumentih razlikujejo. Prav tako ni nujno, da imajo dokumenti enako število atributov.

MongoDB podpira iskanje po poljih, povpraševanje po razponu (ang. range query) in iskanje s pomočjo regularnih izrazov. Poizvedbe lahko vrnejo celotni dokument ali posamezna polja dokumentov in lahko vključujejo tudi uporabniško definirane JavaScript funkcije [18].

5.2.2. Indeksi v MongoDB

Vsakemu polju v dokumentu lahko določimo primarni ali sekundarni indeks. Indeksi so posebne podatkovne strukture za lažje iskanje. Indeks hrani vrednosti enega ali več polj, ki so urejene po vrednosti polja. Če obstaja primerni indeks za opravljanje poizvedbe, MongoDB s pomočjo indeksov omeji dokumente, ki jih mora preiskati, in določi, kateri dokumenti ustrezajo poizvedbi [22].

Vrste indeksov:

- **enojni indeksi**, ki vsebujejo podatke samo enega polja v dokumentu,
- **sestavljene indeksi**, ki vsebujejo več kakor eno polje v dokumentu,

- **indeks z več ključi**, t. j. indeks tabele, ki za vsako polje v tabeli ustvari ključ v indeksu,
- **geoprostorski indeksi in poizvedbe**, ki podpirajo poizvedbe na osnovi lokacije, ki je shranjena kot GeoJSON objekt ali koordinatni par,
- **besedilni indeksi**, ki podpirajo iskanje besedilne vsebine v dokumentu,
- **razpršitveni indeksi**, ki vzdržujejo vnose s pomočjo razpršilnih tabel in vsebujejo vrednosti indeksiranih polj [21].

5.2.3. Podvajanje

Za razliko od Oracle podatkovnih baz, ki izgubo podatkov preprečujejo z uporabo varnostne kopije tabele, MongoDB omogoča shranjevanje več kopij istih podatkov. Vsaka kopija ima vlogo primarne ali sekundarne replike. Primarna kopija opravlja zapisovalne in bralne operacije. Sekundarne kopije vzdržujejo isto vsebino z vgrajeno logiko za repliciranje. Kadar pride do napake v primarni kopiji podatkov, se avtomatsko izvede proces izbire sekundarne replike, ki se zamenja s pokvarjeno primarno. Poleg zaščite proti izgubi podatkov nam podvajanje omogoča še boljšo dosegljivost shranjenih podatkov. Klient lahko pošlje ukaze za bralne in pisalne operacije na določen strežnik, vse kopije baze pa se avtomatsko poenotijo [23].

5.2.4. Programski jeziki za nadzor baze

MongoDB podpira uporabo različnih programskih jezikov, kakor so C, C++, .NET, Java, JavaScript, PHP, Python, ... Vsak od podprtih jezikov se prevede v enostaven povpraševalni jezik, razvit samo v ta namen [18].

5.2.5. MongoDB v MCS

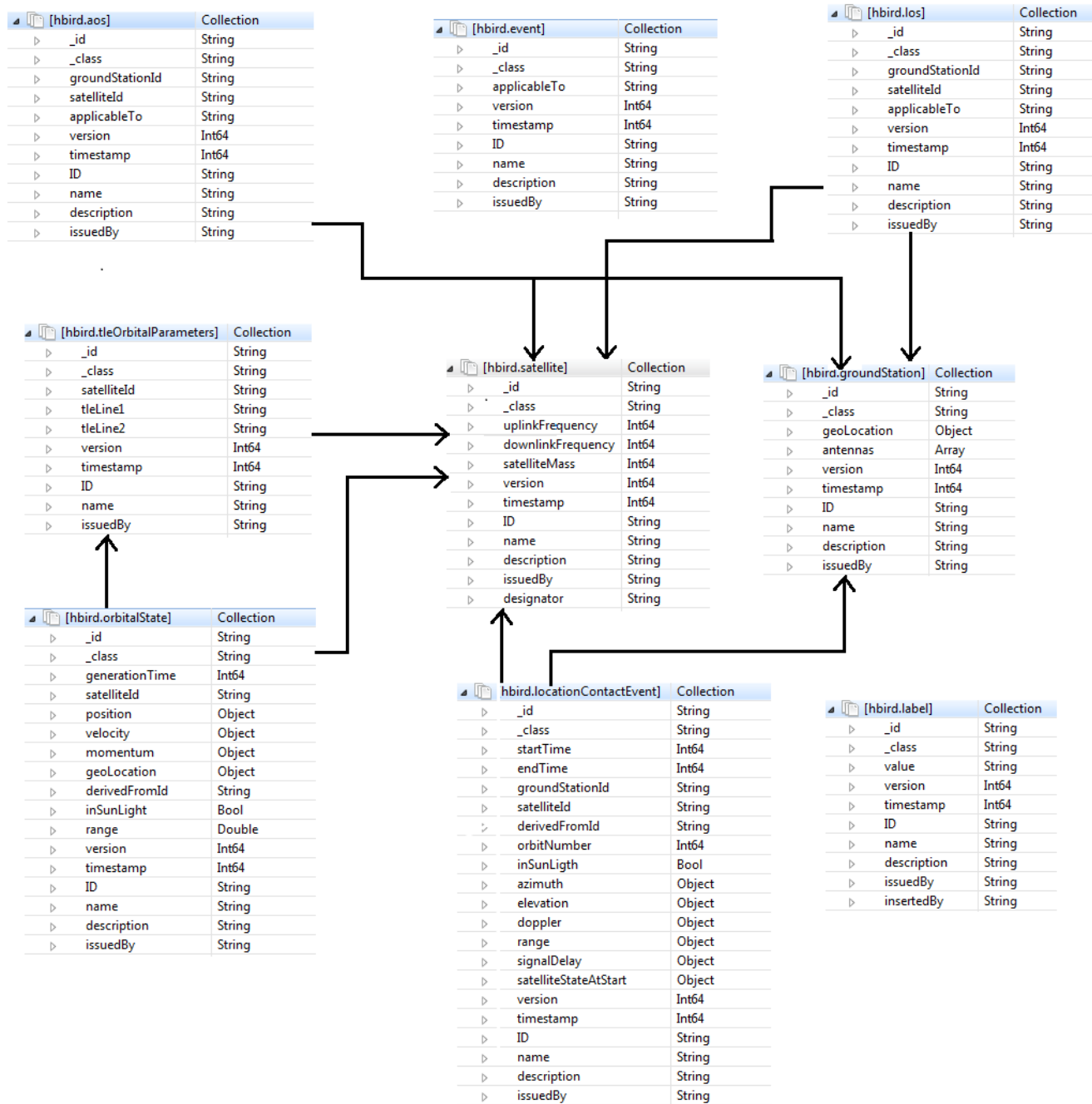
Razvojna ekipa se je odločila za hrambo sistemskih podatkov v MongoDB zaradi hitrosti opravljenih operacij v bazi pri pravilni uporabi indeksov. V primerjavi z sistemom Oracle se povpraševalni stavki izvedejo za več kakor polovico hitreje, vstavljanje vrstic v bazo pa je za več kakor 100-krat hitrejšo. MongoDB omogoča opravljanje poizvedb v JavaScriptu, kar pomeni, da lahko poizvedbe opravljamo v sklopu uporabniškega vmesnika in za poizvedbe ne potrebujemo programske logike v ozadju.

Podatki, shranjeni v bazi MongoDB:

- ukazi, ki se preko AX.25 paketov pošiljajo satelitu,
- zemeljske postaje,
- sateliti, s katerimi je bila vzpostavljena komunikacija,
- podatki o vzpostavitvi povezave s satelitom,
- podatki o izgubi povezave s satelitom,
- podatki o orbiti in hitrosti premikanja satelita,
- podatki o parametrih, ki sestavljajo beacon,
- podatki o zagonu modulov sistema za nadzor misije.

Slika 9 prikazuje podatkovno strukturo zbirk v mongoDB bazi.

Collection	Field	Type
[hbird.command]	_id	String
	_class	String
	arguments	Object
	Object	Object
	CDHSBlockIndex	Object
	commandId	Object
	source	Object
	CDHSSource	Object
	priority	Object
	destination	Object
	device	Object
	timestamp	Object
	sec1	Object
	sec2	Object
	flags	Object
	fs_type	Object
	fs_id	Object
	transferTime	Int64
	executionTime	Int64
	version	Int64
timestamp	Int64	
ID	String	
name	String	
description	String	
[hbird.binary]	_id	String
	_class	String
	rawData	Binary
	version	Int64
	timestamp	Int64
	ID	String
	name	String
	description	String
	issuedBy	String
	[hbird.metadata]	_id
_class		String
metadata		Object
Object		Object
CDHSBlockIndex		Int32
CDHSSource		Int32
source		Int32
priority		Int32
argumentsLength		Int32
ID		Int32
length		Int32
destination		Int32
insertedBy		String
applicableTo		String
version		Int64
timestamp		Int64
ID		String
name	String	
issuedBy	String	
[hbird.startComponent]	_id	String
	_class	String
	arguments	Object
	Object	Object
	entity	Object
	Object	Object
	name	String
	description	String
	type	Object
	value	Object
mandatory	Bool	
destination	String	
transferTime	Int64	
executionTime	Int64	
version	Int64	
timestamp	Int64	
ID	String	
[hbird.track]	_id	String
	_class	String
	arguments	Object
	Object	Object
	satelliteId	Object
	Object	Object
	name	String
	description	String
	type	Object
	unit	String
	value	Object
	mandatory	Bool
	contact	Object
destination	String	
transferTime	Int64	
executionTime	Int64	
version	Int64	
timestamp	Int64	
ID	String	
name	String	
description	String	
issuedBy	String	
[hbird.parameter]	_id	String
	_class	String
	value	...
	Int64	Int64
	Double	Double
	Int32	Int32
	unit	String
	version	Int64
	timestamp	Int64
	ID	String
	name	String
description	String	
issuedBy	String	
calibrated	Bool	



Slika 9: Podatkovni model mongoDB baze

Poglavje 6 Tehnologije, uporabljene pri izdelavi vmesnika

V sledečem poglavju bomo spoznali tehnologije, ki so uporabljene za delovanje spletnega vmesnika sistema za nadzor misije. JavaScript je vodilni odprtokodni programski jezik za razvoj spletnih vmesnikov z velikim naborom knjižnic. Pri razvoju MCS sistema se uporablja ena izmed njih – knjižnjica Dojo.

6.1. Ajax

AJAX (ang. asinhron JavaScript in XML) je interpretiran programerski jezik, ki temelji na JavaScript-u. Najbolj pogosto je uporabljen pri razvoju spletnih aplikacij, saj nam omogoča asinhrono komunikacijo, prilagajanje prikazane strani brez osveževanja in napredno interakcijo z uporabnikom. Uporablja se še pri programiranju mrež na strežnikih, pri razvoju iger ter namiznih in mobilnih aplikacijah.

JavaScript je klasificiran kot prototipno orientiran⁵ skriptni jezik s prvorazrednimi funkcijami⁶. Podpira objektno orientiran, postopkovni in funkcionalni programski slog [19].

6.1.1. Dojo

Zbirka orodij Dojo je odprtokodna, modularna JavaScript/Ajax knjižnica, namenjena razvoju aplikacij in spletnih strani.

Dojo poskrbi za pravilno izvajanje spletnih aplikacij v različnih brskalnikih, vsebuje ogrodje za definiranje modulov in upravljanje z njihovimi odvisnostmi, priskrbi orodja za izgradnjo in optimizacijo JavaScript in CSS programov ter ustvarjanje dokumentacije in testnih razredov, podpira lokalizacijo in internacionalizacijo ter priskrbi širok nabor pogosto uporabljenih razredov in gradnikov uporabniškega vmesnika.

⁵ Prototipno orientirano programiranje je način objektno orientiranega programiranja, kjer se dedovanje izvede s procesom kloniranja obstoječih objektov, ki nato služijo kakor prototip novega objekta.

⁶ Če programski jezik vsebuje prvorazredne funkcije pomeni, da omogoča podajanje funkcij kakor argument drugim funkcijam, vračanje funkcij kakor rezultat funkcije in shranjevanje funkcij v spremenljivke in njihovo shranjevanje v podatkovne strukture.

Zbirko Dojo sestavljajo:

- jedro **dojo**, ki vsebuje temeljne in večino nevizualnih modulov,
- knjižnica **dijit**, ki vsebuje module z gradniki in postavitvami za uporabniški vmesnik,
- knjižnica **dojox**, ki vsebuje module v razvoju, ki še niso dovolj stabilni za priključitev v *dojo* ali *dijit*,
- orodja **util**, ki poskrbijo za optimizacijo, dokumentacijo, preverjanje stila in testiranje.

6.1.1.1. Asinhrona komunikacija

Pomembna funkcionalnost Ajax aplikacij je asinhrona komunikacija med spletnim brskalnikom in strežnikom. Podatki se izmenjajo in vsebina prikazane strani se posodobi brez osveževanja.

6.1.1.2. Hramba podatkov na uporabnikovi strani

Poleg podpore za branje in pisanje piškotkov Dojo omogoča tudi lokalno hrambo uporabniških podatkov. Tako lahko s strinjanjem uporabnika varno shrani podatke, način hrambe pa je odvisen od uporabljenega brskalnika.

6.1.1.3. Hramba podatkov na strežniku

Dojo vsebuje različne načine shranjevanja podatkov na strežniku:

- **CsvStore** omogoča samo branje podatkov iz datotek v formatu csv,
- **OpmlStore** omogoča samo branje podatkov iz datotek v formatu .opml,
- **YahooStore** omogoča branje iskalnih rezultatov iz storitve Yahoo Search!,
- **RdfStore** omogoča branje in pisanje podatkov, ki se uporabljajo za komunikacijo z RDF⁷ podatkovnimi strežniki [10].

⁷ Ogrodje za opis virov (ang. Resource Description Framework - RDF) se uporablja kakor splošna metoda za konceptualno opisovanje podatkov, uporabljenih za spletne vire.

Poglavje 7 Pomen paketov, shranjenih v Oracle podatkovni bazi

Preko dveh različnih frekvenc satelit ESTCube-1 prejema in oddaja dve različni vrsti paketov. S pomočjo beacon sporočil, ki jih satelit oddaja vsakih sedem minut, posreduje telemetrične podatke. V TNC paketih pa so zapakirani AX.25 paketi, ki služijo pošiljanju ukazov satelitu in sprejemanju podatkov o stanju podsistemov na satelitu. V sklopu tega poglavja bomo spoznali strukturo vseh treh paketov in podatke, ki jih vsebujejo.

7.1. Beacon

Vsi sateliti oddajajo "svetilne žarke", katerih namen je, da posredujejo meritve notranjih senzorjev in meritve za določitev položaja satelita na nebu [29].

Beacon je sestavljen iz šestnajstiških znakov, ki se prevedejo z Morsejevo abecedo.

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
CW	T	W	U	S	H	5	6	M	Z	N	A	B	C	D	E	F

Tabela 1: Preslikava iz šestnajstiškega zapisa v Morsejevo abecedo

7.1.1. Struktura EstCube-1 beacons

1. Glava beacons služi identifikaciji oddajajočega satelita. Pri EstCube-1 zavzema vrednost "ES5E/S".
2. Naslednji znak nam pove način beacons. Znak "E" pomeni normalni način, beacon se zaključi s "K" in vsebuje 43 znakov. "T" pomeni varni način in se zaključi s "KN", beacon pa vsebuje 52 znakov [13].

7.1.2. Vrednosti beacons v normalnem načinu

Časovni žig	
Napetost glavnega prevodnika	[V]
Povprečno razmerje moči	[W]
Napetost baterije A	[V]
Napetost baterije B	[V]
Temperatura baterije A	[V]
Hitrost vrtenja Z	[deg/s]
Moč sprejetega signala	[dBm] Decibel-milivat
Faza misije	[0="Ustalitev v orbito", 1="Spremljanje Zemlje", 2="Sprostitev tetra", 3="Merjenje sile na električnem jadru"]
Čas od zadnje ponastavitve	[Ure]
Moč toka na tetru	[mA]
Čas od zadnje napake	[Ure]
Stanje sistema za nadzor in ravnanje s podatki	
Stanje sistema EPS	
Stanje sistema za določitev in nadzor višine	
Stanje sistema za komunikacijo	

Tabela 2: Vrednosti beacons v normalnem načinu [13].

7.2. TNC paketki

Na satelit je vgrajen TNC (ang. Terminal Node Controller), ki poskrbi za asinhrono serijsko komunikacijo z zemeljskimi postajami. Podatki se prenašajo po KISS protokolu.

Protokol KISS omogoča preprosto implementacijo v enostavnih vgrajenih sistemih, ki so sposobni asinhrono serijske komunikacije. Na razpolago je osem ukazov in štirje posebni znaki, ki označujejo začetek in konec KISS paketa. Ukaz z imenom Podatkovni Okvir in z vrednostjo 0x00 omogoča prenos podatkov in se na satelitu EstCube-1 uporablja za pošiljanje AX.25 paketkov [31].

7.3. AX.25

AX.25 je protokol na sloju podatkovne povezave in je izpeljan iz X.25 protokola, ki ga je nadomestil internetni protokol (IP). Uporabljajo ga radio amaterji za vzpostavitev paketnih radio povezav. AX.25 je v glavnem odgovoren za vzpostavljanje povezave in prenos podatkov med vozlišči mreže ter za detekcijo napak na komunikacijskih kanalih [1].

EstCube-1 uporablja AX.25 pakete za prenos telemetrijskih podatkov posameznega podsistema in za pošiljanje ukazov satelitu.

7.3.1. Struktura okvirja na EstCube-1 satelitu

Glava paketa AX.25	
Izvor	uint8
Cilj	uint8
Dolžina sporočila	uint16
Sporočilo	uint32
Noga paketa AX.25	

Tabela 3: Struktura AX.25 paketa

Struktura sporočila je ista, neglede na to, ali gre za zahtevo po telemetriji ali za pošiljanje ukazov satelitu. V primeru zahteve po telemetričnih podatkih določena polja nimajo pomena.

7.3.2. Opis polj v sporočilu

- bit za takojšnjo izvedbo ukaza,
 - če želimo, da se ukaz izvede takoj, vrednost nastavimo na 1
- bit za določitev prioritete,
 - 1 pomeni visoko prioriteto, 0 nizko
- ciljni podsistem,
- ID ukaza,
- izvorni podsistem,
- indeks bloka, kateremu pripada ukaz,
- dolžina podatkov v bytih,
- podatki [14].

Poglavje 8 Predstavitev vmesnika za opravljanje poizvedb v Oracle podatkovni bazi

8.1. Zakaj se vmesnik za opravljanje poizvedb potrebuje?

EstCube-1 preko nizkih frekvenc vedno sporoča lokacijo in stanje satelita. Pakete z imenom beacon lahko sprejemajo vsi radio amaterji, ki jih nato posredujejo ESTCube ekipi, ta pa jih shrani v sistem MCS. Preko visokih frekvenc se satelitu pošilja različne ukaze in posodobitve, satelit pa odgovori o uspešnosti izvedenih operacij oz. posreduje zahtevane podatke. Vsi prejeti podatki se shranijo v arhive.

Analiza prejetih podatkov služi:

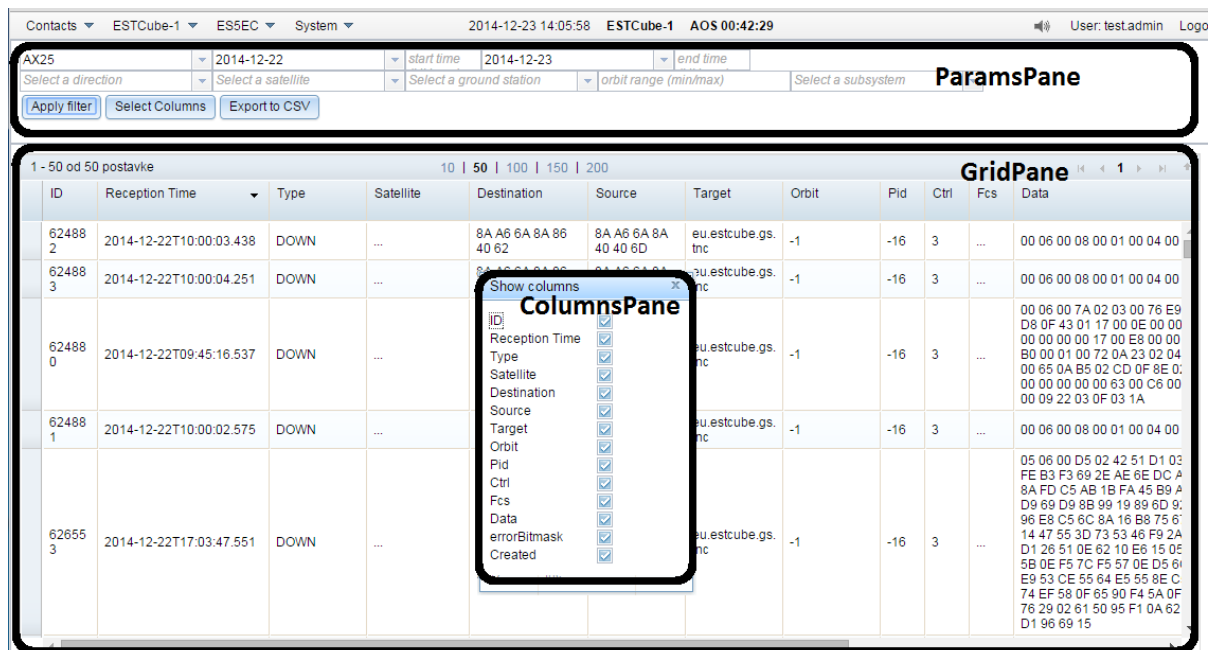
- opazovanju orbite satelita,
- določitvi popravkov orbite satelita,
- določitvi potrebne posodobitve programske opreme na satelitu,
- opazovanju poteka razvitja tetra in določitvi uspešnosti primarne misije,
- prejemanju posnetih slik in realizaciji sekundarne misije.

Skupina ESTCube nima ekipe, ki bi se ukvarjala samo z analizo in distribucijo prejetih podatkov s satelita. Zato mora vsaka skupina posebej dostopati do arhivov in pridobiti potrebne podatke, da preverijo pravilnost delovanja svojega segmenta. Večina študentov, ki sodelujejo pri projektu ESTCube, nima podlage v pridobivanju podatkov iz podatkovnih baz. Če bi vse želeli naučiti, kako pridobiti podatke iz arhivov, bi jim morali razložiti strukturo arhivskih podatkovnih baz in uporabo osnovnih poizvedbenih stavkov. Kljub temu pridobljeni podatki ne bi bili dovolj prečiščeni, saj so zaradi združevanja tabel poizvedbe preveč zahtevne za začetnike. Tako bi izgubljali čas z učenjem študentov, ki pa bi morali še vedno sortirati prejete podatke, zato se je IT razvojna ekipa odločila, da ob prvi priložnosti razvije vmesnik za pridobivanje podatkov iz arhivov, ki bo enostaven in intuitivno razumljiv vsem študentom, ki sodelujejo pri ESTCube projektu.

8.2. Grafični uporabniški vmesnik za opravljanje poizvedb

Vmesnik je zgrajen s komponentami iz knjižnic Dijit in Dojox. Do vmesnika dostopamo preko zavihka “Sistem”, kjer se nahajajo sistemske in diagnostične funkcionalnosti.

Ko prvič odpremo vmesnik, se izvede konstruktorska metoda, ki inicializira glavni vsebovalnik in v njem dve podokni: “paramsPane”, ki vsebuje elemente za določitev iskalnih parametrov, in “gridPane” s tabelo za prikaz rezultatov poizvedbe (Slika 10).



ID	Reception Time	Type	Satellite	Destination	Source	Target	Orbit	Pid	Ctrl	Fcs	Data
62488 2	2014-12-22T10:00:03.438	DOWN	...	8A A6 6A 8A 86 40 62	8A A6 6A 8A 40 40 6D	eu.estcube.gs. tnc	-1	-16	3	...	00 06 00 08 00 01 00 04 00
62488 3	2014-12-22T10:00:04.251	DOWN	eu.estcube.gs. tnc	-1	-16	3	...	00 06 00 08 00 01 00 04 00
62488 0	2014-12-22T09:45:16.537	DOWN	eu.estcube.gs. tnc	-1	-16	3	...	00 06 00 7A 02 03 00 76 E9 D8 0F 43 01 17 00 0E 00 00 00 00 00 00 17 00 E8 00 00 B0 00 01 00 72 0A 23 02 04 00 65 0A B5 02 CD 0F 8E 00 00 00 00 00 00 63 00 C6 00 00 09 22 03 0F 03 1A
62488 1	2014-12-22T10:00:02.575	DOWN	eu.estcube.gs. tnc	-1	-16	3	...	00 06 00 08 00 01 00 04 00
62655 3	2014-12-22T17:03:47.551	DOWN	eu.estcube.gs. tnc	-1	-16	3	...	05 06 00 D5 02 42 51 D1 03 FE B3 F3 69 2E AE 6E DC A 8A FD C5 AB 1B FA 45 B9 A D9 69 D9 8B 99 19 89 6D 9 96 E8 C5 6C 8A 16 B8 75 6 14 47 55 3D 73 53 46 F9 2A D1 26 51 0E 62 10 E6 15 0E 5B 0E F5 7C F5 57 0E D5 6 E9 53 CE 55 64 E5 55 8E C 74 EF 58 0F 65 90 F4 5A 0F 76 29 02 61 50 95 F1 0A 62 D1 96 69 15

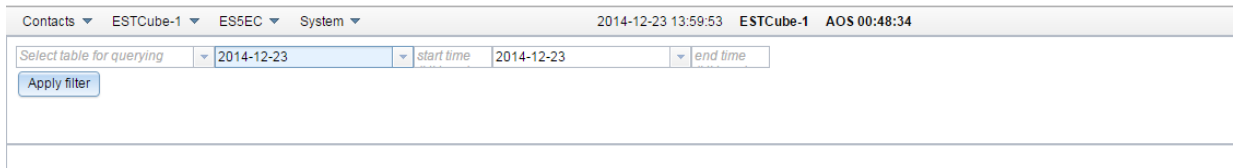
Slika 10: Vsebovalniki, ki sestavljajo vmesnik.

Za elemente v podoknu paramsPane skrbi razred QueryParametersProvider. Le-ta deli paramsPane na dodana tri podokna:

- commonContentPane, ki vsebuje parametre, skupne vsem paketom,
- customContentPane, ki spreminja elemente, glede na parametre izbranega paketa,
- buttonsContentPane, kjer se nahajajo gumbi.

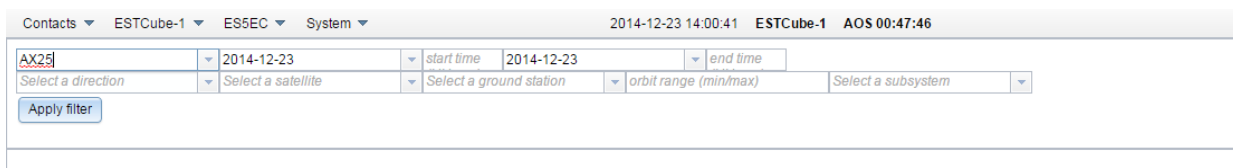
Vrednosti v spustnih seznamih so shranjene v Dojovem modulu za hrambo podatkov na strežniku – shrambi. Vrednosti nekaterih spremenljivk so zaradi lažje implementacije definirane znotraj kode (npr. podsistem, smer prenosa), nekatere pa se v shrambo shranjujejo med izvajanjem sistema in tako vsebujejo vse vrednosti, ki so bile uporabljene (npr. sateliti in zemeljske postaje).

Po inicializaciji strani so vsi elementi brez vrednosti, razen izbirnikov datuma, ki ne delujejo brez privzete vrednosti. Na sliki 10 je prikazan paramsPane s commonContentPane in buttonsContentPane, kakor se prikažejo pri ustvaritvi vmesnika:



Slika 11: ParamsPane po inicializaciji vmesnika

Ko izberemo tabelo paketov, ki nas zanimajo, se pojavi podokno customContentPane z dodatnimi elementi za izbiro vrednosti iskalnih parametrov. Če spremenimo iskalno tabelo, se spremeni tudi vsebina podokna customContentPane. Paketi TNC in AX.25 imajo večino lastnosti istih, beacon pa ima vse parametre edinstvene. Vsebina elementov se ob posodabljanju podokna ohranja. Slika 11 predstavlja parametre, ki jih lahko določimo pri iskanju med AX.25 paketi:



Slika 12: paramsPane, kadar iščemo med AX.25 paketi

Ob kliku na gumb “Apply filter” se ustvari ID poizvedbe in izvede HTTP GET metoda, ki izbrane vrednosti posreduje strežniku.

Stanje poizvedbe je vidno v tabeli. Če vmesnik še čaka na odgovor, se izpiše “Request sent. Waiting for response”. Kadar ni zadetkov za poizvedbo, se sporočilo spremeni v “No data to display”, sicer pa se tabela napolni s paketi, ki ustrezajo iskalnim parametrom. Ko je število vrnjenih paketov večje od 50, se rezultat poizvedbe razdeli na več strani. Nad tabelo imamo komponente, s katerimi določamo prikazano število paketov na eni strani, in komponente za premikanje med stranmi. Med stranmi se lahko premikamo s puščico, eno po eno, ali skočimo na določeno stran z izbiro številke strani.

Contacts ▾ ESTCube-1 ▾ ES5EC ▾ System ▾ 2014-07-28 16:53:41 ESTCube-1 -- NO AOS/ILOS DATA -- User: test.admin Logout

TNC: 2014-07-24 10:22 2014-07-28 end time
 none Select a satellite Select a ground station orbit range (min/max)

Apply filter Select Columns Export to CSV

ID	Reception Time	Type	Satellite	Target	Orbit	Data	Created
559363	2014-07-24T13:52:29.209	DOWN	...	eu.estcube.gs.tnc	-1	8A A6 6A 8A 86 40 62 8A A6 6A 8A 40 40 6D 03 F0 00 06 00 05 22 03 60 01 01	2014-07-24T13:52:29.000
559362	2014-07-24T13:52:28.432	UP	...	eu.estcube.gs.tnc	-1	8A A6 6A 8A 40 40 6D 8A A6 6A 8A 86 40 62 03 F0 06 00 00 8A 20 03 60 86 03 5D 12 01 0B 10 90 93 EF 09 80 93 EE 09 0E 94 0F 10 90 93 F1 09 80 93 F0 09 0E 94 13 10 90 93 F3 09 80 93 F2 09 0E 94 17 10 90 93 F5 09 80 93 F4 09 0E 94 1B 10 90 93 F7 09 80 93 F6 09 0E 94 1F 10 90 93 F9 09 80 93 F8 09 0E 94 6A 0F 90 93 FB 09 80 93 FA 09 0E 94 6E 0F 90 93 FD 09 80 93 FC 09 0E 94 72 0F 90 93 FF 09 80 93 FE 09 0E 94 32 10 90 93 01 0A 80 93 00 0A 0E 94 2C 10 90 93 03 0A 80 93 C8 AA 58 62	2014-07-24T13:52:28.000
559361	2014-07-24T13:52:28.041	DOWN	...	eu.estcube.gs.tnc	-1	8A A6 6A 8A 86 40 62 8A A6 6A 8A 40 40 6D 03 F0 00 06 00 05 22 03 60 01 00	2014-07-24T13:52:28.000
559360	2014-07-24T13:52:27.409	UP	...	eu.estcube.gs.tnc	-1	8A A6 6A 8A 40 40 6D 8A A6 6A 8A 86 40 62 03 F0 06 00 00 8A 20 03 60 86 03 5D 12 00 D9 09 80 93 D8 09 0E 94 E3 0F 90 93 DB 09 80 93 DA 09 0E 94 E7 0F 90 93 DD 09 80 93 DC 09 0E 94 EB 0F 90 93 DF 09 80 93 DE 09 0E 94 EF 0F 90 93 E1 09 80 93 E0 09 0E 94 F3 0F 90 93 E3 09 80 93 E2 09 0E 94 F7 0F 90 93 E5 09 80 93 E4 09 0E 94 FB 0F 90 93 E7 09 80 93 E6 09 0E 94 FF 0F 90 93 E9 09 80 93 E8 09 0E 94 03 10 90 93 EB 09 80 93 EA 09 0E 94 07 10 90 93 ED 09 80 93 EC 09 0E 94 C8 AA 8F 32	2014-07-24T13:52:27.000
559367	2014-07-24T13:52:30.539	DOWN	...	eu.estcube.gs.tnc	-1	8A A6 6A 8A 86 40 62 8A A6 6A 8A 40 40 6D 03 F0 00 06 00 05 22 03 60 01 01	2014-07-24T13:52:30.000
559366	2014-07-24T13:52:30.377	UP	...	eu.estcube.gs.tnc	-1	8A A6 6A 8A 40 40 6D 8A A6 6A 8A 86 40 62 03 F0 06 00 00 8A 20 03 60 86 03 5E 12 01 12 0A 93 2F 80 E0 82 2B 90 93 15 0A 80 93 14 0A 80 EA 99 E0 CF 91 08 95 FB 01 A0 E1 20 E0 30 ED 41 91 51 91 24 0F 25 1F A1 50 D1 F7 81 15 40 E2 94 07 38 F0 36 95 27 95 36 95 27 95 36 95 27 95 30 C0 DB 01 90 96 ED 91 FC 91 91 97 EE 0F FF 1F E6 0F F7 1F A0 81 B1 81 96 95 87 95 91 83 80 83 FB 01 80 A1 91 A1 FC 01 EE 0F FF 1F E6 0F F7 1F 40 81 51 81 4A 1B 5B 0B 24 0F 35 1F 01 96 80 31 DE 12 29 3C	2014-07-24T13:52:30.000
559365	2014-07-24T13:52:30.377	DOWN	...	eu.estcube.gs.tnc	-1	8A A6 6A 8A 86 40 62 8A A6 6A 8A 40 40 6D 03 F0 00 06 00 05 22 03 60 01 01	2014-07-24T13:52:30.000

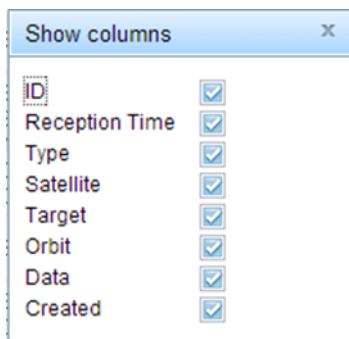
Slika 13: Prikaz vmesnika po opravljeni poizvedbi

Ko so prikazani paketi tipa beacon, se nam ob kliku na posamezno vrstico v tabeli odpre novo okno z razčlenjenimi podatki.

Ob prvi poizvedbi se pojavita še dva gumba:

- “Select Columns”, s katerim izbiramo prikazane stolpce v trenutno izrisani tabeli,
- “Export to CSV”, ki tabelo shrani v format csv in shrani na uporabnikov računalnik.

Pri kliku na gumb “Select Columns” se nam na sredini vmesnika odpre novo okno z vsemi lastnostmi iskanih paketov. Z izbirnimi polji določimo, katere lastnosti želimo opazovati. Vrstice v tabeli se posodobijo, ko s klikom na križec v zgornjem desnem kotu zapremo okno. Primer okna za izbor stolpcev pri prikazu AX.25 paketov je prikazan na sliki 13.

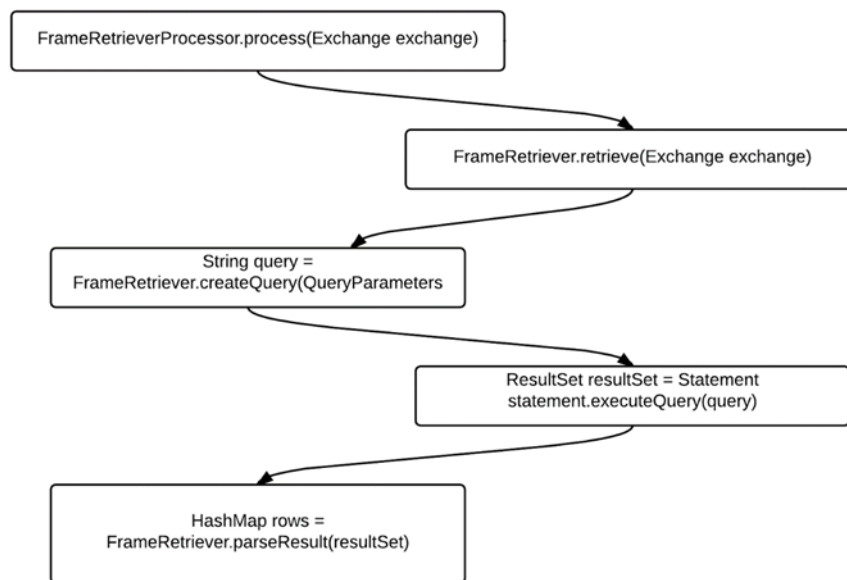


Slika 14: Okno za izbor prikazanih vrednosti v tabeli

Pri izvozu v format csv se paketi, prikazani v tabeli, shranijo v istem vrstnem redu, kakor so prikazani, enak ostane tudi vrstni red stolpcev. Izvozijo se samo stolpci, ki so izbrani za prikaz.

8.3. Programska logika

Strežnik od grafičnega vmesnika prejme HTTP GET zahtevo z vrednostmi vseh parametrov, četudi te niso določene. Razred CustomQueryServlet kreira objekt tipa QueryParameters in iskalne parametre preslika v attribute objekta. Ta objekt se nato preko usmerjevalnih poti posreduje modulu Archiver.



Slika 15: Prikaz izvršitve poizvedbe v Archiver modulu

Ob zagonu modula Archiver ta vzpostavi povezavo z podatkovno bazo Oracle. Za vsako vrsto paketov ustvari objekt tipa FrameRetrieverProcessor, ki pridobi rezultat poizvedbe in ga vrne strežniku.

Ko Archiver prejme novo zahtevo, le-to prejme primeren FrameRetrieverProcessor, saj je v glavi sporočila definirana tabela, nad katero se bo opravila poizvedba. Glede na glavo sporočila jih Camel sortira in posreduje primernemu FrameRetrieverProcessorju.

```
from(Constants.AMQ_RETRIEVE).choice()  
    .when(simple("${in.header.table}==AX25")).process(ax25FrameRetriever)
```

```

.when(simple("${in.header.table}=="TNC")).process(tncFrameRetriever)
.otherwise().process(beaconRetriever)
.end();

```

Za vsako vrsto paketov je definiran posamičen razred tipa FrameRetriever, ki kreira SQL poizvedbeni stavek in ga pošlje sistemu za upravljanje podatkovne baze. Upravljalca baze vrne rezultate v objektu tipa ResultSet, kjer je vsaka vrnjena vrstica iz baze shranjena v vrstici tabele, vsak stolpec pa predstavljen s parom ime stolpca/vrednost. Za pridobivanje vrednosti iz ResultSeta se uporablja iterator, ki se pomika od prvega do zadnjega rezultata poizvedbe.

```

public class Ax25Retriever extends FrameRetriever {
public void retrieve(Exchange exchange) throws ArchivingException, SQLException {
try {
    AX25QueryParameters params = (AX25QueryParameters)
exchange.getIn().getBody();
    String query = createQuery(params);
    df.setTimeZone(utc);
    super.retrieve(exchange);
    Statement statement = c.createStatement();
    ResultSet resultSet = statement.executeQuery(query);
    //get metadata
    ResultSetMetaData meta = null;
    meta = resultSet.getMetaData();
    //get column names
    int colCount = meta.getColumnCount();
    ArrayList<String> cols = new ArrayList<String>();
    for (int i=1; i<=colCount; i++){
        cols.add(meta.getColumnName(i));
    }
    //fetch out rows
    HashMap<Integer, AX25Object> rows = new HashMap<Integer,
AX25Object>();
    AX25Object row = null;
    while (resultSet.next()) {
        if(!rows.containsKey(resultSet.getInt("ID"))){
            row = new AX25Object();
            row.setRequestId(params.getReqId());
            for (String colName:cols) {
                if (resultSet.getObject(colName) == null);
                else if (colName.equals("ID")){

```

```

        row.setId(resultSet.getInt(colName));
    }else if (colName.equals("DIRECTION")){
        row.setDirection(resultSet.getString(colName));
    }
    ...
    ...
    ...
    }rows.put(row.getId(), row);
}
exchange.getOut().setHeader("RequestId", params.getReqId());
exchange.getOut().setBody(new ArrayList<Object>(rows.values()));
} catch (NumberFormatException e) {
    e.printStackTrace();
}finally{
    c.close();
}
}
}

```

Rezultat poizvedbe se shrani v HashMap in se v taki obliki preko usmerjevalne poti pošlje grafičnemu vmesniku, bolj natančno v shrambo tabele za prikaz rezultatov. Kadar je število paketov, ki ustrezajo iskalnim parametrom, večje od 1000, se seznam razbije na več kosov s 1000 elementi, ki se asinhrono pošljejo po usmerjevalni poti. S tem se skrajša čas prenosa rezultata od modula do uporabniškega vmesnika.

```

public void process(Exchange exchange) throws Exception {
    frameRetriever.retrieve(exchange);
    DefaultCamelContext context = new DefaultCamelContext();
    producer = context.createProducerTemplate();
    try {
        ArrayList<Object> array = exchange.getOut().getBody();
        if (array.size()>=1000){
            int partitionSize = 1000;
            ArrayList<List<Object>> partitions = new ArrayList<List<Object>>();
            for (int i = 0; i < array.size(); i += partitionSize) {
                partitions.add(array.subList(i, i + Math.min(partitionSize,
                    array.size() - i)));
            }
            Iterator<List<Object>> i = partitions.iterator();

```



```
while(i.hasNext()){  
    producer.sendBodyAndHeader("activemq:queue:customQueryReturn", i.next().toArray(), "requestId",  
    exchange.getOut().getHeader("RequestId"));  
}  
else{  
    producer.sendBodyAndHeader("activemq:queue:customQueryReturn",  
    array, "requestId", exchange.getOut().getHeader("RequestId"));  
}  
catch (Exception e) {  
    e.printStackTrace();  
}  
}
```

Poglavje 9 Zaključek

Vesoljska tehnologija postaja vse bolj javno dostopna. Število projektov, namenjenih v izobraževalne namene, se zadnja leta povečuje sorazmerno z dostopnostjo tehnologije. Večina komercialnih projektov ni javno dostopnih, kar pomeni, da morajo študentje odprtokodne sisteme razviti sami. Ker pa se znanje študentov ne more kosati z usposobljenostjo profesionalnih načrtovalcev in ker se razvijalci menjajo iz semestra v semester, razvoj poteka mnogo počasneje.

Cilj diplomske naloge je bil razvoj močno potrebnega vmesnika za opravljanje poizvedb v arhivih in prikaz rezultatov. Potreba po vmesniku se je pokazala po izstrelitvi ESTCube-1 satelita v orbito, vendar ga zaradi funkcionalnosti z višjo prioriteto implementacije ni bilo mogoče realizirati.

Največja ovira pri razvoju je bilo veliko število uporabljenih tehnologij v sistemu za nadzor misije, ki sem jih morala spoznati v postopku načrtovanja vmesnika, zato sem v treh mesecih, kolikor časa sem delala na razvoju vmesnika, implementirala le osnovne funkcionalnosti, ki so potrebne za tekoče delovanje in ki so jih zahtevali končni uporabniki. V okviru nadaljnjega razvoja bi lahko dodali sortiranje podatkov v tabeli, optimizirali kreiranje poizvedbenega stavka in omogočili shranjevanje iskalnih parametrov.

Sodelovanje v projektu me je še dodatno navdušilo nad razvojem programske opreme in mi vlilo zaupanje samo vase, saj sem videla, da lahko tudi neprofesionalni razvijalci veliko prispevamo k sistemu v razvoju.

Kazalo slik

Slika 1: Prikaz delovanja električnih solarnih jader	4
Slika 2: Sestavni deli satelita [26].	6
Slika 3: Arhitektura sistema za nadzor misije	11
Slika 4: Moduli Spring ogrodja [27].	13
Slika 5: Prikaz avtentikacije s sistemom Crowd	16
Slika 6: Struktura Apache Maven projekta	18
Slika 7: Struktura AX25_FRAME tabele	23
Slika 8: Struktura AX25_FRAME_DETAILS tabele	23
Slika 9: Podatkovni model mongoDB baze	27
Slika 10: Vsebovalniki, ki sestavljajo vmesnik.	35
Slika 11: ParamsPane po inicializaciji vmesnika	36
Slika 12: paramsPane, kadar iščemo med AX.25 paketi	36
Slika 13: Prikaz vmesnika po opravljeni poizvedbi	37
Slika 14: Okno za izbor prikazanih vrednosti v tabeli	37
Slika 15: Prikaz izvršitve poizvedbe v Archiver modulu	38

Kazalo tabel

Tabela 1: Preslikava iz šestnajstiškega zapisa v Morsejevo abecedo	30
Tabela 2: Vrednosti beacons v normalnem načinu [13]	31
Tabela 3: Struktura AX.25 paketa	32

Viri

- [1] W. A. Beech, D. E. Nielsen in J. Taylor, AX.25 Link Access Protocol, Tucson : Amateur Packet Radio Corporation, 1998.
- [2] J. E. Scott, S. Spendolini in T. Fox, Pro Oracle application express, New York : Apress, 2014.
- [3] P. Janhunen in A. Sandroos, „Simulation study of solar wind push on a charged wire: basis of,“ © European Geosciences Union, Helsinki, 2007.
- [4] K.-L. Kusmin, „Application of Scrum methodology in long-term student,“ 2012.
- [5] E. J. Okugbeni, „Security Implementation of Mission Control,“ 2014.
- [6] „Work on Estonia’s first satellite under way. “ *The Baltic Times*, 03 april 2013.

Spletni viri

- [7] „Amateur Radio – PEØSAT,“ [Elektronski]. Available: <http://www.pe0sat.vgnet.nl/satellite/cube-nano-picosats/estcube-1/>. [Poskus dostopa 08 december 2014].
- [8] „Apache ActiveMQ,“ 2011. [Elektronski]. Available: <http://activemq.apache.org/>. [Poskus dostopa 13 november 2014].
- [9] „Apache Camel,“ [Elektronski]. Available: <http://camel.apache.org/>. [Poskus dostopa 17 november 2014].
- [10] „Dojo Toolkit Reference Guide,“ 2012. [Elektronski]. Available: <http://dojotoolkit.org/reference-guide/1.10/index.html>. [Poskus dostopa 04 december 2014].

- [11] „Electric Solar Wind Sail (E-sail),“ [Elektronski]. Available: <http://www.electric-sailing.com/>. [Poskus dostopa 12 december 2014].
- [12] „Electric Solar Wind Sail (E-sail) - theory,“ [Elektronski]. Available: <http://www.electric-sailing.com/theory.html>. [Poskus dostopa 12 december 2014].
- [13] „ESTCube - Beacon Decoding,“ [Elektronski]. Available: <http://www.estcube.eu/en/radio/beacon-decoding>. [Poskus dostopa 11 december 2014].
- [14] „ESTCube - Telemetry Packet Description,“ [Elektronski]. Available: <http://www.estcube.eu/en/telemetry-packet-description>. [Poskus dostopa 11 december 2014].
- [15] „ESTCube - What is ESTCube-1,“ [Elektronski]. Available: <http://www.estcube.eu/en/mission/what-is-estcube-1>. [Poskus dostopa 18 december 2014].
- [16] „ESTCube-1 Subsystems,“ [Elektronski]. Available: <http://www.estcube.eu/en/subsystems>. [Poskus dostopa 19 december 2014].
- [17] „Hummingbird In 180 Seconds,“ [Elektronski]. Available: <http://www.hbird.de/?q=180sec>. [Poskus dostopa 16 december 2014].
- [18] „Introduction to MongoDB,“ [Elektronski]. Available: <http://www.w3resource.com/mongodb/introduction-mongodb.php>. [Poskus dostopa 28 november 2014].
- [19] „JavaScript Tutorial for begginers,“ [Elektronski]. Available: http://referencedesigner.com/tutorials/js/js_1.php. [Poskus dostopa 04 december 2014].
- [20] „Maven getting started guide,“ 07 november 2014. [Elektronski]. Available: <http://maven.apache.org/guides/getting-started/index.html>. [Poskus dostopa 13 november 2014].
- [21] „MongoDB - Indexes,“ [Elektronski]. Available: <http://docs.mongodb.org/manual/core/indexes/>. [Poskus dostopa 28 november 2014].
- [22] „MongoDB - Indexes Introduction,“ [Elektronski]. Available: <http://docs.mongodb.org/manual/core/indexes-introduction/>. [Poskus dostopa 28 november 2014].
- [23] „MongoDB - Replication Introduction,“ [Elektronski]. Available: <http://docs.mongodb.org/manual/core/replication-introduction/>. [Poskus dostopa 28 november 2014].

- [24] „Oracle Database - intro,“ 2014. [Elektronski]. Available: https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm. [Poskus dostopa 27 novemeber 2014].
- [25] „Overview of SSO,“ [Elektronski]. Available: <https://confluence.atlassian.com/display/CROWD/Overview+of+SSO>. [Poskus dostopa 05 november 2014].
- [26] „Sharing Earth Observation Resources - ESTCube-1,“ [Elektronski]. Available: <https://directory.eoportal.org/web/eoportal/satellite-missions/e/estcube-1>. [Poskus dostopa 12 december 2014].
- [27] „Spring Architecture,“ [Elektronski]. Available: http://www.tutorialspoint.com/images/spring_architecture.png. [Poskus dostopa november 2014].
- [28] „Spring Framework - Reference documentation,“ 04 november 2014. [Elektronski]. Available: <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/>. [Poskus dostopa 13 november 2014].
- [29] I. Sunter, „ESTCube forum,“ 18 april 2014. [Elektronski]. Available: <http://confluence.tudengisatelliit.ut.ee:8080/viewtopic.php?f=6&t=3&sid=8e69ce653a5d673f6dd59062be926178>. [Poskus dostopa 06 november 2014].
- [30] „Tartu Observatory space resarch centre - staff,“ [Elektronski]. Available: http://www.to.ee/eng/about_us/staff. [Poskus dostopa november 2014].
- [31] „Wikipedia - KIS (TNC),“ 28 september 2014. [Elektronski]. Available: http://en.wikipedia.org/wiki/KISS_%28TNC%29. [Poskus dostopa 11 december 2014].
- [32] „Wikipedia - CGI Group,“ 07 november 2014. [Elektronski]. Available: http://en.wikipedia.org/wiki/CGI_Group. [Poskus dostopa 12 januar 2015].