

**UNIVERZA V MARIBORU
FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO**

Izidor Mlakar

DIPLOMSKO DELO

Maribor, September 2008



UNIVERZA V MARIBORU



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

Diplomsko delo univerzitetnega študijskega programa

**KOGNITIVNA MOBILNA PLATFORMA »DATA« ZA INTELIGENTNA
OKOLJA**

Študent: Izidor Mlakar
Študijski program: Univerzitetni, Telekomunikacije
Mentor: doc. dr. Rojc Matej
Somentor: red. prof. dr. Zdravko Kačič

Maribor, September 2008

ZAHVALA

Mentorju dr. Mateju Rojcu in somentorju dr. Zdravku Kačiču se zahvaljujem za vso strokovno svetovanje, potrpežljivost in spodbujanje v času nastajanje diplomske naloge, za omogočeno aktivno sodelovanje pri raziskovalnem delu v laboratoriju in strokovno usmerjanje tekom študija na programu Telekomunikacije.

Iskrena zahvala tudi dragi sestri, mami in očetu za vso ljubezen in podporo, ki ste mi jo namenili in za zaupanje, ki ste ga izkazovali v vseh mojih odločitvah.

Hvala tudi tebi Tjaša, ki me spremljaš in izkazuješ razumevanje kljub dnevnim muham. Hvala ti za vso podporo!

Zahvaljujem se tudi vsem ostalim, ki so mi stali ob strani tako pri raziskovalnem delu, kot tudi študiju!

KOGNITIVNA MOBILNA PLATFORMA »DATA« ZA INTELIGENTNA OKOLJA

Ključne besede: inteligentno okolje, multimodalni vmesniki človek-stroj, mobilna robotska enota, porazdeljeno programsko ogrodje, mikrokrmilnik Atmel ATmega 8, Java Media Framework, RTP prenos podatkov, TCP/IP prenos podatkov, XML prenos podatkov, strojni protokoli, končni avtomat, dogodkovno usmerjeni sistemi

UDK: 004.5(043.2)

Povzetek

Diplomsko delo vključuje analizo tehnologij in pristopov, ki omogočajo razvoj sistemov sposobnih kontekstnega procesiranja dogodkov v okolici in po drugi strani generiranje kognitivnih odzivov. Takšni sistemi postavljajo uporabnika v ospredje in se ne ukvarjajo zgolj z zajemanjem informacije, avtomatskim ali ukaznim krmiljenjem. Področje raziskav je v tem kontekstu usmerjeno predvsem v tako imenovane multimodalne tehnologije (npr. zajemanje avdio/video informacije, procesiranje avdio/video informacije in generiranje multimodalnih odzivov), umetno inteligenco (npr. statistično modeliranje in tvorjenje uporabniških profilov) in robotiko (npr. mobilna robotska enota, v celoti samostojno gibanje enote v danem prostoru in multimodalnost robotskih enot).

COGNITIVE MOBILE PLATFORM “DATA” FOR INTELLIGENT AMBIENCE

Key words: intelligent ambience, multimodal human-machine interface, mobile robotic unit, distributed framework, Atmel ATmega 8 microcontroller, Java Media Framework, RTP data transmission, TCP/IP data transmission, XML data transmission, hardware protocols, finite-state machine, event-based systems

UDK: 004.5(043.2)

Abstract

The presented diploma thesis includes analysis of technologies and approaches that enable development of systems, capable to process cognitive events in the environment and on the other hand capable to generate cognitive responses. Such systems are user oriented and don't deal with only capturing information, or driving techniques performed automatically or by using commands. The research field is in this context directed towards so-called multimodal technologies (e.g. audio/video acquisition, audio/video processing and generation of multimodal responses), artificial intelligence (e.g. statistical modeling and generation of user profile) and robotics (e.g. mobile robotic unit, fully autonomous unit's movement and multimodality of mobile robotic units).

VSEBINA

1	UVOD.....	10
2	INTELIGENTNA OKOLJA	14
	2.1 Arhitekture inteligentnih okolij	18
	2.1.1 EU projekt Amigo (Ambient Intelligence for the Networked Home).....	20
	2.1.2 EU projekt CHIL (Computers In the Human Interaction Loop)	22
	2.1.3 Drugi pomembnejši projekti	24
	2.2 Komponente inteligentnih okolij	25
	2.2.1 Senzorji	25
	2.2.2 Kamere in mikrofonska polja	26
	2.2.3 Mobilna robotska enota	27
	2.3 Multimodalnost.....	33
3	ZASNOVA KOGNITIVNEGA INTELIGENTNEGA OKOLJA	37
	3.1 Kognitivna arhitektura	37
	3.2 Arhitektura kognitivnega inteligentnega sistema »DATA«	39
4	ZASNOVA NARAVNEGA VMESNIKA ČLOVEK-STROJ (Human-Machine Interface – HMI).....	41
	4.1 SmartKom.....	41
	4.1.1 Arhitekturne zahteve sistema za multimodalno interakcijo.....	42
	4.1.2 Prilagodljiv sistem za multimodalni dialog	43
	4.1.3 Primeri aplikativne uporabe generične arhitekture.....	48
	4.2 Funkcionalna zasnova kognitivnega sistema »DATA« za inteligentna okolja	52
5	UMETNA INTELIGENCA (Artificial Intelligence - AI).....	56
6	ZAJEMANJE IN PRENOS AVDIO IN VIDEO INFORMACIJE	60
	6.1 RTP protokol (Real-time Transport protocol)	61
	6.2 RTP modul.....	64
7	UPORABA PODATKOVNIH BAZ.....	68
	7.1 Programski paket DATA SQL.....	69
	7.1.1 Avtomatski vnos podatka	69
	7.1.2 Poizvedba na zahtevo	71
8	KOMUNIKACIJSKI PROTOKOLI V SISTEMU »DATA«	72
	8.1 Strojni protokoli	73
	8.1.1 I2C (TWI) – povezava po dveh linijah (ATmega8)	74
	8.1.2 10 - bitna ADC pretvorba (ATmega8).....	75
	8.1.3 SPI – serijska povezava (ATmega8).....	77
	8.1.4 USART ali UART - univerzalni asinhroni serijski protokol (ATmega8)	80
	8.2 Mrežni protokoli	80
	8.2.1 TCP/IP protokol.....	81

8.2.2	Serijski protokol.....	83
8.2.3	Osnovni XML protokol	84
9	DISTRIBUIRANO PROGRAMSKO OGRODJE »DATA«.....	86
9.1	Specifikacije za distribuirano programsko ogrodje »DATA«.....	86
9.1.1	Sestava kompleksnega distribuiranega sistema z uporabo DATA ogrodja... ..	90
9.1.2	Java UniMod programsko ogrodje	91
9.1.3	Izvedba scenarijev	92
9.1.4	Gradniki sistema »DATA«.....	94
9.2	Konfiguriranje in profiliranje kognitivne mobilne plaforme »DATA«.....	104
9.2.1	Specifikacije za distribuirano programsko ogrodje »DATA«.....	105
10	PROGRAMSKO OGRODJE ZA ZAJEMANJE IN PRENOS AVDIO/VIDEO INFORMACIJE.....	107
11	RAZVOJ KOGNITIVNE MOBILNE PLATFORME »DATA«.....	112
11.1	Strojni nivo	112
11.1.1	Telemetrični modul (Vremenska postaja)	113
11.1.2	Krmilni modul	114
11.1.3	Centralni modul	119
11.1.4	Vmesnik na oddaljenem računalniku.....	121
11.2	Krmilni sistem mobilne robotske enote	123
11.2.1	Krmilni modul	125
11.2.2	Modul za upravljanje z motorji.....	129
11.2.3	Telemetrični modul.....	131
11.2.4	Nadgradnja modula Vremenska postaja	134
12	APLIKATIVNI SCENARIJI UPORABE SISTEMA »DATA«.....	139
12.1	Oddaljeno upravljanje z mobilno robotsko enoto.....	140
12.2	Sočasna komunikacija z mobilnimi robotskimi enotami.....	145
12.3	Nenadzorovano delovanje sistema DATA	148
13	SKLEP	151
	LITERATURA	154
	PRILOGE	157

Seznam slik

Slika 2-1: Konceptualna shema inteligentnega okolja.	15
Slika 2-2: Modularna arhitektura procesov inteligentnega okolja.....	17
Slika 2-3: Arhitektura storitvenega nivoja.	19
Slika 2-4: Sistemska arhitektura projekta AMIGO[8].....	20
Slika 2-5: Sistemska arhitektura CHIL[9].....	23
Slika 2-6: Tipična karakteristika senzorja (vzorec žarka).	26
Slika 2-7: Splošni "beamformer" tipa filtriraj & seštej.....	26
Slika 2-8: Prototip mobilne robotske enote.	28
Slika 2-9: Model mobilne robotske enote.....	28
Slika 2-10: Holonomski vozni sistem.....	30
Slika 2-11: Izbira alternativne poti.	31
Slika 3-1: Zasnova kognitivne arhitekture.....	37
Slika 3-2: Sistemska arhitektura DATA.....	38
Slika 4-1: Visoko nivojska arhitektura multimodalne interakcije.	42
Slika 4-2: Funkcionalna shema arhitekture sistema SmartKom	44
Slika 4-3: Modularna arhitektura informacijskega kioska SmartKom-Public.	47
Slika 4-4: Modularna arhitektura sistema SmartKom – Home.	49
Slika 4-5: Komponentna arhitektura sistema SmartKom – Mobile.	51
Slika 4-6: Funkcionalna zasnova kognitivnega sistema za inteligentna okolja.....	53
Slika 5-1: Funkcionalna shema inteligentnega agenta.	57
Slika 6-1: Avdio/vizualna podpora modulom v sistemu »DATA«.....	60
Slika 6-2: Struktura RTP paketa.....	62
Slika 6-3: Primer implementacije RTP modula.....	65
Slika 7-1: Grafični vmesnik DATA SQL programskega paketa.....	69
Slika 7-2: Funkcionalna shema procesa shranjevanja v bazo.....	70
Slika 8-1: Arhitektura sistema »DATA« na protokolnem nivoju.	72
Slika 8-2: TWI povezava.....	74
Slika 8-3: Prenos podatkov po TWI povezavi [38].....	75
Slika 8-4: Blokovna shema ADC pretvorbe[38].....	76
Slika 8-5: Blokovna shema SPI povezave[38].....	77
Slika 8-6: Blokovna shema SPI povezave gospodar – suženj[38].....	78
Slika 8-7: Blokovna shema UART modula[38].....	79
Slika 8-8: Struktura TCP/IP paketa.....	81
Slika 8-9: Enojna serijska povezava (polovični dupleks).....	84
Slika 8-10: Dualna serijska povezava (polni dupleks).....	84
Slika 8-11: Zgradba XML paketa.....	84
Slika 9-1: Inteligentno okolje z uporabo ogrodja »DATA«.....	86
Slika 9-2: Sistem inteligentnega okolja za upravljanje z informacijo/nadzorom.....	89
Slika 9-3: Enostaven primer scenarija uporabljenega za pogon DATA odjemalca.....	93
Slika 9-4: Funkcionalna in logična shema DATA odjemalca.....	95
Slika 9-5: DATA odjemalec: sprejemanje RTP tokov in ločeno predvajanje.....	96
Slika 9-6: DATA odjemalec: sprejemanje RTP tokov in združeno predvajanje.....	96
Slika 9-7: Funkcionalna in logična shema DATA strežnika.....	98
Slika 9-8: Načini predvajanja RTP toka na DATA strežniku.....	99
Slika 9-9: Razpošiljanje multimedijskega toka z uporabo RTP protokola.....	100

Slika 9-10: Post-procesiranje oddaljenega RTP toka.	101
Slika 9-11: Funkcionalna shema DATA modularnega strežnika.	102
Slika 9-12: Shema procesa zajemi → predvajaj/razpošlji.	103
Slika 9-13: Shema procesa prejmi → predvajaj/shrani.	104
Slika 10-1: Funkcionalna shema procesa zajemanja informacije iz USB kamere.	107
Slika 10-2: Samostojni grafični vmesnik RTP programskega paketa v sistemu DATA.	108
Slika 10-3: Funkcionalna shema procesa zajemanja informacije RTP toka.	109
Slika 10-4: Funkcionalna shema razpošiljanja RTP toka.	111
Slika 11-1: Funkcionalna shema sistema DATA na strojnem nivoju.	112
Slika 11-2: Funkcionalna shema modula »vremenska postaja«.	113
Slika 11-3: Vežje krmilnega modula.	114
Slika 11-4: Funkcionalna shema merjenja tlaka okolja.	115
Slika 11-5: Funkcionalna shema merjenja temperature okolja.	116
Slika 11-6: Funkcionalna shema elektronskega kompasa.	118
Slika 11-7: Centralni modul.	119
Slika 11-8: Funkcionalna shema vmesnika v MATLAB orodju.	121
Slika 11-9: Grafični vmesnik telemetričnega sistema.	122
Slika 11-10: Funkcionalna shema modula upravljanja mobilne robotske enote.	123
Slika 11-11: Funkcionalna shema krmilnega modula.	124
Slika 11-12: RC5 sprejemno vežje.	125
Slika 11-13: Programska logika krmilnega modula.	126
Slika 11-14: Modul za upravljanje z motorji.	129
Slika 11-15: Vežje telemetričnega modula.	131
Slika 11-16: Programska logika telemetričnega modula.	132
Slika 11-17: Vremenska postaja II.	135
Slika 11-18: Priklop temperaturnega senzorja.	137
Slika 11-19: Priklop barometra.	138
Slika 12-1: Grafični vmesnik sistema DATA.	139
Slika 12-2: Scenarij uporabe sistema: oddaljeno krmiljenje.	141
Slika 12-3: Komunikacijski diagram v primeru scenarija oddaljenega krmiljenja.	142
Slika 12-4: Sočasno prejemanje dveh RTP tokov.	144
Slika 12-5: Scenarij sočasne uporabe sistema DATA.	144
Slika 12-6: Komunikacijski diagram sočasnega upravljanja v sistemu DATA.	146
Slika 12-7: Scenarij avtonomnega delovanja sistema DATA.	148
Slika 12-8: Komunikacijski diagram avtonomnega delovanja.	149

Seznam tabel

Tabela 2-1: Nivoji avtonomnosti mobilne robotske enote.	29
Tabela 4-1: Vir znanja sistema SmartKom.....	46
Tabela 4-2: Vir znanja sistema SmartKom (nadaljevanje).....	47
Tabela 5-1: Objektive umetne inteligence.....	56
Tabela 6-1: Zgradba glave RTP paketa.	62
Tabela 6-2: Zgradba glave RTP paketa (nadaljevanje).	63
Tabela 6-3: Avdio/Video kodeki RTP protokola.	63
Tabela 6-2: Avdio/Video kodeki RTP protokola (nadaljevanje).....	64
Tabela 6-3: Osnovni RTP gradniki.....	66
Tabela 7-1: Tipični SQL ukazi.	68
Tabela 8-1: Struktura tipnih TCP/IP paketov.	81
Tabela 8-2: Struktura tipnih TCP/IP paketov (nadaljevanje).	82
Tabela 8-3: Primeri serijskih paketov.....	83
Tabela 11-1: PWM vrednosti.	133

Uporabljene kratice in oznake

RTP	Real-time Transport Protocol
MAC	<u>Media Access Control</u>
CMOS	Complementary metal–oxide–semiconductor, tehnologija uporabljena pri izgradnji mikroprocesorjev
AVR	Modified Harvard architecture 8-bit RISC single chip microcontroller
PWM	pulzno širinska modulacija
SDA	Podatkovna povezava dvožične serijske povezave
SCL	Urna povezava dvožične serijske povezave
MIPS	Milijonov inštrukcij na sekundo
RISC	Reduced instruction set computer
SQL	Strukturirani povpraševalni jezik za delo s podatkovnimi bazami (angl. Structured Query Language)
RDBMS	Sistem za relacijsko upravljanje s podatkovno bazo (Relational database management system)
HMI	vmesnik človek-stroj (human-machine interface)
ASR	razpoznavaletnik govora (automatic-speech recognition)
TTS	sintetizator govora (text-to-speech synthesis system)
ASCII	ameriški standardiziran kodni zapis za izmenjavo informacij "American Standard Code for Information Interchange"
RISC	reduciran nabor ukazov "Reduced Instruction Set Compute"
RS232	asinhrona serijska povezava
TCP	podatkovni kontrolni protokol "Transmission Control Protocol"
WAN	prostrano omrežje "Wide Area Network"
A/D	analogno/digitalni pretvornik
CMOS	polprevodniška tehnologija "Complementary Metal Oxide Semiconductor"
D/A	digitalno/analogni pretvornik
EEPROM	programirljiv ROM "Electrical Erasable Programmable Read Only Memory"
IR	infra rdeča svetloba "Infrared"
ISP	način programiranja "In-System Programming"

SPI	serijski periferni vmesnik "Serial Peripheral Interface"
UART	asinhroni prenos "Universal Asynchronous Receive and Transmit"
USART	asinhroni in sinhroni prenos "Universal Synchronous and Asynchronous Serial Receiver and Transmitter"
USB	standard za zaporedni prenos podatkov "universal serial bus"
LAN	lokalno omrežje "Local-Area Network"
Wi-Fi	brezžična vernost Wireless Fidelity
IP	Internet Protokol
Li	Litij
Ni-Cd	Nikelj – Kadmij
XML	razširljiv razčlenjevalni jezik " Extensible Markup Language"

1 UVOD

Inteligentna okolja predstavljajo novo paradigmo razvoja tehnologij v 21. stoletju. Vse več domačih in delovnih okolij vsebuje najrazličnejše računalniške sklope in v smislu avtomatskega izvajanja nalog, vse bolj "inteligentne" elektronske naprave. Tako dandanes v gospodinjstvih že lahko najdemo "pametne" pralne stroje, "pametne" hladilnike, naprednejše električne vtičnice itd. Avtomatizacija okolja tako postaja realnost v našem vsakdanjem življenju in bo prav gotovo narekovala smer razvoja vedno bolj naprednih inteligentnih okolij. Razvoj inteligentnih okolij temelji na izvedbi sofisticiranih namenskih inteligentnih okolij, pri čemer skuša izkoristiti funkcionalne prednosti posameznih implementacij, ter dodajanjem vedno novih in novih storitev. Glavno vprašanje, ki si ga pri razvoju takšnih sistemov moramo zastaviti ni več: "Kaj lahko okolje stori namesto nas?", temveč, "Kako lahko okolje izboljša kvaliteto našega življenja in v danem trenutku odreagira na človeku naraven način in celo z ustreznim kontekstnim obnašanjem?".

Razvoj inteligentnih okolij je torej usmerjen v implementacijo čim bolj vsestranske kognitivne sisteme, ki se »zavedajo« konteksta, v katerem prihaja do najrazličnejših akcij in dogodkov v okolici, ter znajo tudi ustrezno samodejno reagirati. Razvoj je še posebej usmerjen v iskanje inovativnega združevanja posameznih enot inteligentnih sistemov, kar bi tvorilo bolj smiselno, fleksibilno in predvsem bolj učinkovito modularno arhitekturo inteligentnih sistemov. V kontekstu izvedbe čimbolj naravne interakcije med človekom in napravami v inteligentnih okoljih, se pojavljata dve glavni veji razvoja.

Prva veja je usmerjena v raziskave in razvoj inteligentnih pogovornih avatarjev in sicer v okviru statičnih vmesnikov inteligentnega okolja. Avatarji dandanes že lahko izvajajo človeku podobne animacije posameznih delov obraza in telesa (neverbalna komunikacijska sredstva), služijo pa tudi kot dodaten navdih za razvoj hologramskih animacij tako oseb, kot akcij, ki jih izvajajo.

Druga veja je usmerjena v razvoj novih konceptov robustnih avtonomnih mobilnih robotskih enot, ki so sposobne opravljati vedno bolj inteligentne in kompleksne naloge, za delovanje pa potrebujejo vse manj energije. Takšne robotske enote morajo, podobno

kot inteligentna okolja, vključevati napredne in čimbolj naravne interakcijske vmesnike, ki vključujejo multimodalno interakcijo in kontekstno razumevanje uporabnikov, okolja in dogodkov v inteligentnem okolju.

Glavna področja razvoja inteligentnih okolij predstavljajo: umetna inteligenca, kognitivnost in multimodalne interakcije oz. vmesniki. Pravimo, da neka naprava v inteligentnem okolju poseduje lastnost *umetne inteligence*, če izkazuje sposobnost »razmišljanja«. To pomeni, da je sposobna spremljati dogodke v okolju, se učiti in posledično samostojno ali delno samostojno odločati o svojem nadaljnjem obnašanju ali nadaljnjem obnašanju drugih naprav, s katerimi upravlja. Umetna inteligenca je torej predvsem sposobnost naprav, da znajo posnemati delovanje človeškega uma [1].

S pojmom *kognitivnost* v znanosti označujemo kakršnokoli umsko operacijo ali strukturo, ki jo lahko preučujemo pod točno določenimi pogoji [2]. Kognitivna znanost se kot disciplina ukvarja s pridobivanjem znanja, na kakšen način živa bitja (stroji) pridobivajo svoje znanje, znanje predstavljajo in kako lahko s temi predstavitvami manipulirajo oz. jih procesirajo [3]. Kognitivna naprava je tako naprava, ki je sposobna pridobljeno informacijo pretvoriti v nov vir znanja, to znanje predstaviti oz. ga na primeren način uporabiti v nadaljnjih procesih delovanja.

Multimodalna interakcija uporabniku omogoči nove in naprednejše načine interakcije z računalniškim sistemom, ki presegajo tradicionalno interakcijo z uporabo miške ali tipkovnice, ki jo dandanes največ uporabljamo [4]. Običajno multimodalni interakcijski vmesniki združijo že poznane tradicionalne načine interakcije z avdio/vizualno interakcijo v obliki govora, obrazne mimike, gest rok in govornih animacij (pogovorni avatarji). Multimodalni vmesniki prinašajo tako bistveno večjo stopnjo naravnosti v komunikaciji človek-stroj (oz. v komunikaciji z napravami v inteligentnih okoljih), hkrati pa omogočajo uporabo sistemom tudi bistveno širšemu krogu ljudem (različnih sposobnosti).

V predstavljenem diplomskem delu, se implementacije inteligentnega sistema lotevamo na inovativen pristop povezovanja poljubnega inteligentnega okolja in kognitivne osebne mobilne robotske enote. Pri tem sama ideja o vzporednem in povezanem razvoju kognitivne osebne mobilne robotske enote in inteligentnega okolja, ter posledično nove arhitekture inteligentnega sistema, izhaja iz naslednje predpostavke:

V inteligentnih okoljih se lahko nahaja ena ali več mobilnih robotskih enot. Mobilne robotske enote so si med seboj lahko različne glede namembnosti in funkcionalne vloge v inteligentnih okoljih. Glede na svoje karakteristike razpolagajo v splošnem z različnimi kapacitetami napajanja. To posredno vpliva na odločitve, kako glede na procesorsko zmogljivost, ki si jo lahko privoščimo na mobilnih robotskih platformah, učinkovito porazdeliti breme procesiranja kompleksnih podatkov in informacij v samem inteligentnem okolju. Kontekstno razumevanje in procesiranje podatkov (npr. avdio/video), ki omogočijo kognitivno obnašanje, se tako lahko izvajajo na določenih mobilnih robotskih platformah lokalno. Na mobilnih robotskih platformah, ki razpolagajo z manjšo procesorsko zmogljivostjo, pa moramo zagotoviti sicer enako funkcionalnost, kljub temu, da se kognitivni procesi večinoma implementirajo zunaj mobilne robotske enote, kot del celotnega inteligentnega okolja.

V prvem delu diplomske naloge tako predstavljamo smer razvoja inteligentnih okolij, tako v smislu uporabniških in sistemskih zahtev inteligentnih okolij. V tem kontekstu podrobneje predstavimo najbolj odmevne projekte, ki na svojem področju dandanes tudi najboljše sledijo naslednjim ciljem razvoja kognitivnih, kontekstno usmerjenih inteligentnih okolij:

- modularnost sistemske arhitekture,
- visoko kvalitetno zajemanje multimodalne informacije,
- napredno procesiranje multimodalne informacije,
- naravna multimodalna interakcija, ki daje občutek naravnosti komunikacije z napravami.

Prvi del diplomske naloge je hkrati izhodišče za drugi implementacijsko naravnan del diplomske naloge. V drugem delu naloge so tako predstavljene tiste modularne rešitve, ki predstavljajo inovativen pristop k reševanju problemov, s katerimi se dandanes srečujemo v inteligentnih okoljih. V kontekstu sistemske arhitekture za inteligentna okolja, je predstavljena inovativna arhitektura, ki je podprta s programskim ogrođjem, ki je bilo razvito v okviru diplomske naloge. V kontekstu te arhitekture se predstavijo različni programski gradniki sistema, ki omogočajo predvsem bolj enostavno in bolj fleksibilno integracijo ene ali več mobilnih robotskih enot, in sočasno prilagodljivo

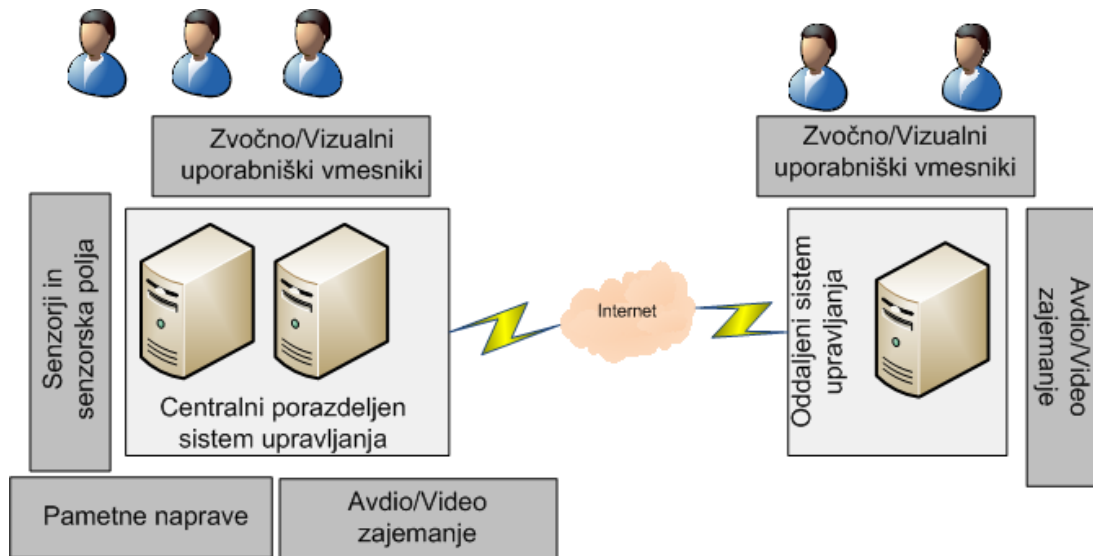
obnašanje sistema, tudi za več uporabnikov hkrati. Predstavljena diplomska naloga predstavlja tudi izvedbo samih komunikacijskih protokolov (ki omogočijo učinkovito komunikacijsko povezovanje med gradniki) in obnašanja sistema. Kot osnova multimodalne interakcije, se predstavi sistem zajemanja multimodalne informacije z uporabo različnih senzorjev in primarna fuzija, ter abstrakcija zajete informacije v obliki predstavitve in uporabe rezultatov meritev. Diplomska naloga v kontekstu mobilnosti, varovalnih mehanizmov in multimodalnega zajemanja informacije, v končnem implementacijskem delu predstavi tudi osnovo mobilne robotske platforme v inteligentnih okoljih. Diplomsko delo zaključimo s samo predstavitvijo celotnega opravljenega dela, pri čemer izpostavimo prednosti in slabosti posameznih gradnikov, ki ga razvit sistem lahko doprinese k razvoju inteligentnih okolij. Le te so predvsem pogojene s stanjem razvoja razpoložljivih in v diplomski nalogi uporabljenih tehnologij. Na koncu predlagamo še smernice nadaljnjega razvoja predstavljenega kognitivnega sistema za inteligentna okolja.

2 INTELIGENTNA OKOLJA

Inteligentna okolja predstavljajo novo paradigmo v razvoju komunikacije človek-stroj. Do nedavnega je inteligentno okolje predstavljalo avtomatizirano okolje, ki lahko samostojno opravlja določene naloge. Dandanes smo tako priča razvoju avtomatizacije industrijskih sistemov, sistemov sposobnih upravljanja z napravami v bivalnem okolju, sistemov avtomatiziranega ali pol-avtomatiziranega nadzora, ter drugih podobnih sistemov. Skupna lastnost vseh teh sistemov je predvsem točno določena namembnost in sicer omogočiti uporabnikom upravljanje in vnaprej predvideno obnašanje naprav in okolja v katerem bivajo ali delajo.

V zadnjih letih je razvoj na področju inteligentnih okolij šel predvsem v združevanje posameznih povsem ločenih in modularnih sistemov, v izboljšave obstoječih tehnologij ter v ustvarjanje okolij, ki so ljudem vsakdanja. Pametna soba/hiša predstavlja prvi tak korak v smeri izdelave okolja, katere namen je pomagati ljudem pri njihovih vsakdanjih opravilih. Tradicionalna pametna okolja v osnovi vsebujejo uporabniške vmesnike, ki omogočajo povezovanje uporabnika in naprav v okolju. Vmesnik se lahko izvede z uporabo terminalskega zaslona (*touch screen*), mikrofonskih polj in nadzorno-krmilnega strežniškega sistema. Uporabnik lahko v takem primeru s pritiskom na zaslon ali govornimi ukazi upravlja z napravami v okolju. Še tako kvalitetno opremljena hiša in kompleksno izvedena avtomatizirana okolja, pa predstavljajo dandanes še vedno zelo »primitivno« obliko končne ideje, to je ideje o povezavi inteligence in življenjskega okolja v katerih delamo in bivamo. Želja je, da bi bilo okolje (hiša/soba) adaptivno, in kot tako ne bi temeljilo na vnaprej programirljivem oz. fiksnem obnašanju. V skladu s tem naj bi okolje bilo sposobno z opazovanjem svoje obnašanje tudi samo prilagoditi željam, potrebam in življenjskemu slogu uporabnikov [5][6]. Ustrezna umetna inteligenca takšnih okolij bi omogočala različno obnašanje okolja posameznim uporabnikom in prilagajanje njihovim trenutnim zahtevam. Uporabniki želijo upravljati z napravami v takšnih okoljih npr. prižigati, ugašati luči, se sprehajati med oddajami na televiziji itn. Želijo biti mobilni in upravljati z okoljem kjerkoli, neodvisno od tega, kje se naprave nahajajo. Po drugi strani mora sistem inteligentnega okolja spremljati/razpoznati obnašanje, kar predstavlja pomembne vhodne podatke za učenje

in adaptacijo sistema na določenega uporabnika. Na podlagi pravil in vzorcev, ki se jih sistem lahko nauči, se potem lahko sistem v različnih situacijah tudi različno odziva. Inteligentna okolja (npr. dom, pisarna, bolnišnica) so dodatno opremljena s senzorji in računalniki. Vse skupaj pa je povezano v skupno lokalno omrežje in priključeno na internet. S komponentami takšnega sistema dandanes že lahko upravljajo virtualni agenti - programska oprema, ki ima definirane izbire uporabnikov in lahko okolje oblikuje v skladu z njihovimi željami in zahtevami [7]. Na sliki 2-1, je predstavljena konceptualna shema inteligentnega okolja. Kot je razvidno iz slike, predstavlja inteligentno okolje združitev dveh sistemov: centralnega, ki omogoča porazdeljeno procesiranje informacije in oddaljenega sistema, ki omogoča oddaljen nadzor, krmiljenje oz. upravljanje z napravami v inteligentnem okolju.



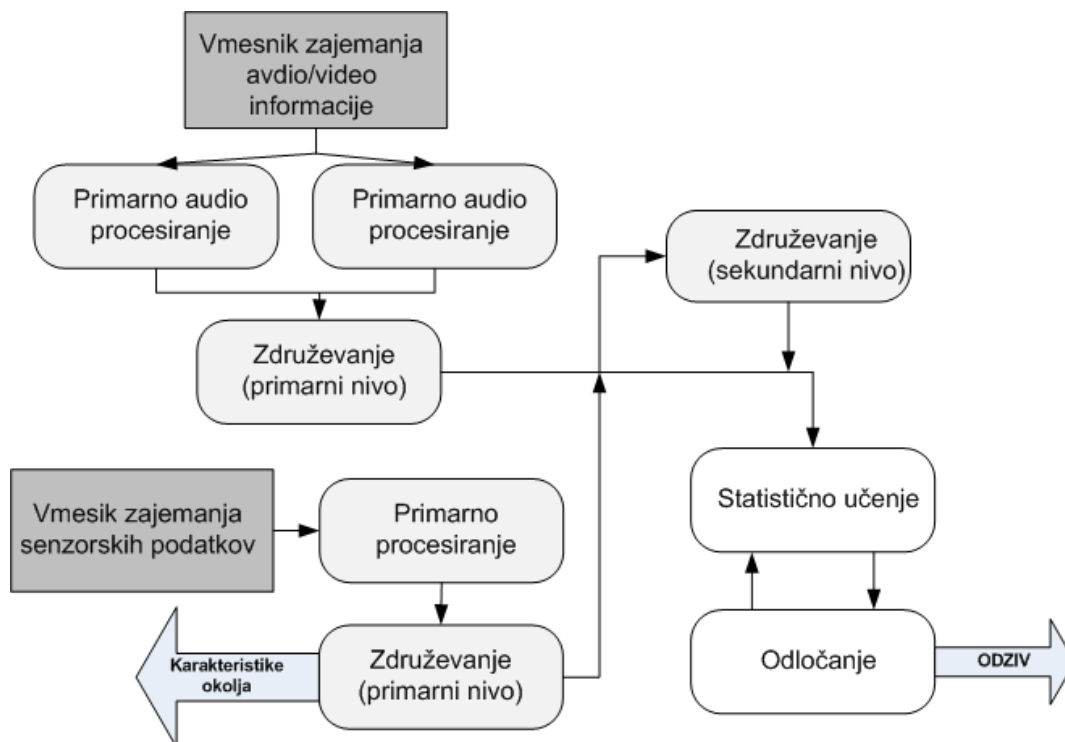
Slika 2-1: Konceptualna shema inteligentnega okolja.

Centralni porazdeljen sistem upravljanja predstavlja jedro upravljalnega sistema inteligentnega okolja. Sistem gosti multimodalne vmesnike za izvajanje multimodalne interakcije (dialogov) z uporabniki. Na centralni porazdeljen sistem upravljanja priključujemo tudi »pametne« naprave, ki s svojimi naprednimi funkcionalnostmi, obogatijo inteligentno okolje z novimi rešitvami, informacijami in storitvami, hkrati pa razbremenijo centralni porazdeljen sistem odvečnega procesiranja, ker ga lahko opravljajo same. *Oddaljeni sistem upravljanja* prav tako vključuje uporabo multimodalnih vmesnikov, ki niso del lokalnega omrežja inteligentnega okolja, temveč

integriran del odjemalca. Namenjeni so oddaljenemu nadzoru, upravljanju oz. krmiljenju naprav v inteligentnem okolju. Uporabniki oddaljenih sistemov upravljanja, lahko koristijo večino funkcionalnosti, ki jih ponuja dana rešitev inteligentnega okolja. Z drugimi besedami povedano, »centralni porazdeljen sistem« upravljanja in »oddaljeni sistem upravljanja«, skupaj tvorijo navidezno privatno omrežje (VLAN) Navidezno privatno omrežje se tvori avtomatsko, strežnik pa dovoljuje povezave le prijavljenih odjemalcev. Bivalno okolje in umetna inteligenca predstavljata dve tesno povezani področji dela v smeri vizije naprednejših inteligentnih okoljih in zagotavljanja kvalitetnejšega in učinkovitejšega življenja ljudi. Pri tem različnost posameznika predstavlja enega izmed najpomembnejših in hkrati najtežje rešljivih parametrov pri zagotavljanju kvalitetnega, človeku prijaznega in koristnega bivalnega okolja. »Centralni porazdeljen sistem upravljanja« predstavlja programsko jedro inteligentnega okolja, ki vključuje: module, ki so se sposobni učiti in znajo posledično delovati na osnovi opazovanja in statističnih pristopov; module generiranja odzivov; in module upravljanja s »pametnimi« napravami v okolju. Da bi inteligentno okolje sledilo svoji osnovni ideji o prilagodljivosti in uporabniški prijaznosti, mora po priporočilih Evropske robotske platforme (EUROP – European Robotics Platform) [42] zagotavljati naslednje:

- visoko kvalitetno zajemanje multimodalne informacije,
- združevanje zajete multimodalne informacije,
- distribuirano procesiranje in statistično obdelavo podatkov v realnem času,
- inovativno in napredno komunikacijo človek-stroj in generiranje odzivov v skladu s karakteristikami ciljne osebe (ali skupine),
- osnovo za kolektivno obnašanje posameznih modulov.

Slika 2-2 prikazuje dalje funkcionalno arhitekturo inteligentnega okolja, ki smo jo narisali v skladu z zahtevami in predpostavkami zapisane v priporočilih EU (EUROP). Slika predstavlja arhitekturo inteligentnega okolja v obliki logično združenih funkcionalnostih, ki omogočajo izvedbo modularnosti in ki sledijo postavljenim kriterijem po prilagodljivosti in uporabniški prijaznosti. Dodane so tudi visoko nivojske povezave med njimi.



Slika 2-2: Modularna arhitektura procesov inteligentnega okolja.

Multimodalno zajemanje avdio/video podatkov (Multimodal audio/video acquisition) v inteligentnih okoljih lahko izvajamo z uporabo mikrofonskih polj in stereo kamer, ki morajo zagotavljati visoko kvalitetno zajemanje. Dodatno lahko procese inteligentnega okolja obogatimo tudi s podatki in informacijami, pridobljenimi iz senzorjev v okolju. **Primarno avdio/video procesiranje** predstavlja procesiranje avdia in videa na signalnem nivoju. Zvočno/slikovno informacijo med procesiranjem očistimo in pretvorimo v obliko, ki predstavlja dalje vhod v module za učenje in odločanje. V osnovi procesiranje služi izboljševanju kvalitete signala (npr. zmanjševanje prisotnosti šuma v signalu), postopkom detektiranja dogodkov (detekcija gibanja, detekcija govora, detekcija raznih specifičnih akustičnih dogodkov itn.) in razpoznavanju (razpoznavanje govora, razpoznavanje oseb, razpoznavanje čustev itn.). **Združevanje (fusion)** predstavlja proces združevanja raznih modalitet, oziroma fuzijo med različnimi tipi informacije, ki jo sistem procesira pri opazovanju uporabnikov. Tako lahko z uporabo razpoznavanja govora (ASR – automatic speech recognition) in obraza na primer prepoznamo osebo, ali z uporabo ASR-ja in razpoznavanja gest opredelimo ukaz, ki ga želi uporabnik izvesti, itn. Fuzija v osnovi predstavlja

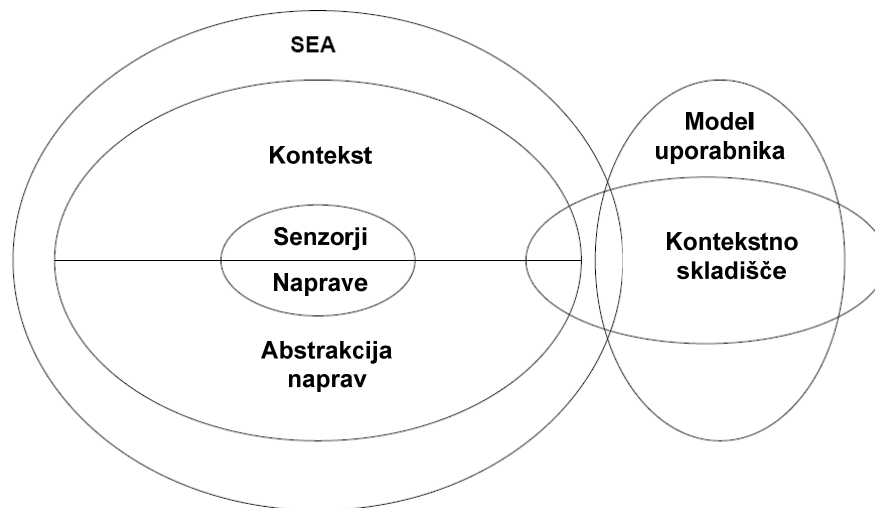
združevanje koristnih podatkov v takšne funkcionalne modele, s katerimi lahko zvišamo kvaliteto razpoznavanja sistema. Na primarnem nivoju se navadno omejimo na združevanje sorodnih informacij (npr. slika in zvok, temperatura in tlak), na sekundarnem nivoju pa skušamo rezultatom primarne fuzije, dodati še informacijo, ki ni povsem sorodna (npr. sliki in zvoku dodamo recimo še temperaturo in tlak). **Multimodalni odzvi** predstavljajo v inteligentnih okoljih takšne systemske odzive, ki so posredovani v obliki slike in govora. Govor tvorimo na osnovi sinteze govora (TTS), odzive v obliki slike pa preko pogovornih avatarjev.

V nadaljevanju bomo predstavili pomembnejše arhitekture inteligentnih sistemov, ki v določeni meri že sledijo ciljem razvoja modernih kognitivnih inteligentnih okolij. V okviru teh arhitektur, bomo na kratko predstavili dosežke nekaterih EU projektov, ki uporabljajo najnovejše tehnologije na področju razvoja inteligentnih okolij in najnovejše tehnologije za procesiranje informacij, ki tudi omogočajo implementacijo novih in naprednejših funkcionalnosti inteligentnih okolij.

2.1 Arhitekture inteligentnih okolij

Veliko dandanašnjih študij je skušalo zapolniti vrzel med nizko nivojskimi sistemski napravami kot so senzorji, ter kontekstno usmerjenimi visoko nivojskimi aplikacijami na vrhu. Projekt "The Aware Home Project" je realiziral inteligentno okolje, ki je sposobno zaznati in spremljati aktivnosti prebivalcev. Sistem je sestavljen iz več blokov, ki vključujejo med drugim sistem za sledenje in razpoznavanje aktivnosti znotraj prostorov, velike projekcijske zaslone, brezžična senzorska omrežja, itn. Projekt postavi komponente kontekstnega razumevanja na orodni programski sistem. Izbrana arhitektura sicer omogoča raznovrstnost v domačih inteligentnih okoljih, vendar pa ne more zadostiti potreb, ki jih imajo večja in bolj strukturirana inteligentna okolja.

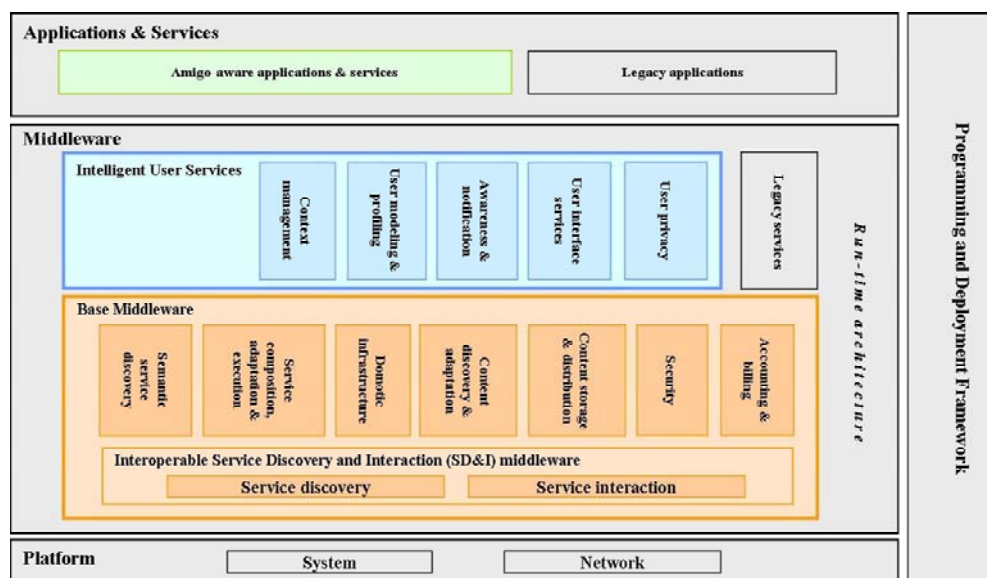
Pri zasnovi arhitekture inteligentnega okolja, je nujno potrebno predpostaviti predvsem ustrezne nivoje abstrakcije za storitve prodornih sistemov z zavedanjem konteksta (*context-aware pervasive systems*). Uveljavila se je zasnova več nivojske kontekstne arhitekture (gre za dvo nivojsko strukturo), ki je predstavljena na sliki 2-3 [43].



Slika 2-3: Arhitektura storitvenega nivoja.

Kot je razvidno iz slike, je arhitektura razdeljena v dva dela. Leva stran je sestavljena iz nivojev abstrakcije. Ti nivoji upravljajo s prenosom in transformacijo informacije med strojnimi napravami v okolju in aplikacijami, ki opravljajo digitalizirane storitve, ki jih uporabniki lahko uporabljajo v inteligentnem okolju. Drugi del arhitekture pa predstavljata kontekstno skladišče in model uporabnika. **Senzorski nivo** sestavlja nabor senzorjev in naprav, ki nadzirajo fizično ali virtualno okolje (npr.: senzorji detekcije gibanja, temperaturni senzorji, itn.) Senzorski nivo torej poskrbi za zajem informacije iz okolja in jo kot vhod podaja kontekstnemu nivoju. **Kontekstni nivo** mora (na sliki 2-2 primarno procesiranje in primarno združevanje) v osnovi procesirati (filtriranje) in uporabiti (združevati) zajeto senzorsko informacijo. Informacijo enega ali več tipov senzorjev lahko nato združi in pretvori v informacijo višjega nivoja. Preoblikovana in združena informacija se nato shrani v kontekstno skladišče, ki predstavlja ključni element arhitekture. **Kontekstno skladišče** združuje (na sliki 2-2 sekundarno združevanje in odločanje) informacijo celotnega okolja in hkrati z okoljem tudi upravlja. Kontekstni nivo shranjuje informacijo v obliki zbirke arbitrarnih kontekstnih objektov. Pametni agent SEA (*Smart Environment Agent*), se uporablja za prenos informacije iz kontekstnega nivoja, preko kontekstnega skladišča na nivo modela uporabnika (na sliki 2-2 - učenje). SEA agent predstavlja tako najvišji nivo abstrakcije arhitekture inteligentnega okolja, kjer se obravnava najvišje nivojske kontekstne informacije. Navadno se uporablja več pametnih agentov, saj lahko vsak agent na svoj

način interpretira isto informacijo. Na primer razpoznavanje osebe potrebuje večjo mero gotovosti razpoznavanja, da se sistem odloči da je oseba razpoznana, kot pa v primeru SAE agenta, ki izvaja detekcijo govora in mora govor le zaznati. Uporabniški model vsebuje informacijo o ljudeh in senzorsko informacijo, s katero lahko modeliramo aspekt osebe. Dokler je informacija anonimna, se hrani v kontekstnem skladišču. Ko pa se informacija poveže z določeno osebo, se mora prestaviti v uporabniški model (proces profiliranja). Ločevanje informacije s profiliranjem omogoči obravnavanje specifičnih osebnih informacij, s katerimi lahko to osebo dovolj dobro tudi opišemo. Nivoja naprav in abstrakcije pošiljata informacijo dalje nizko nivojskim napravam. Vse razen najenostavnejših naprav, imajo svoj objekt, ki vsebuje attribute naprave v kontekstnem skladišču. Karakteristike vključujejo IP naslov, MAC naslov ali USB – port, na katerega so priključene. V primeru, da okolje želi komunicirati s to napravo, se to izvede preko nivoja abstrakcije naprav. V nadaljevanju predstavljamo nekaj najbolj znanih projektov inteligentnih okolij, financiranih s strani EU.



Slika 2-4: Sistemska arhitektura projekta AMIGO[8].

2.1.1 EU projekt Amigo (*Ambient Intelligence for the Networked Home*)

Projekt AMIGO (Ambient Intelligence for the Networked Home) [8], predstavlja enega izmed boljših pristopov gradnje inteligentnih okolij. Amigo uporablja dobro definirano mrežno strukturo, ki inteligenco domačega okolja izvaja z integracijo inteligentnih naprav in njihovih gostiteljskih storitev. Na storitvenem nivoju AMIGO

vpeljuje dobro poznano sistemsko arhitekturo črne table ("blackboard"), ki temelji na prenosu informacije strežnik-odjemalec. Projekt AMIGO je zgradil multimodalno interakcijsko okolje, ki služi kot uporabniški vmesnik. Vmesnik podpira tako govorno kot slikovno usmerjene operacije zajemanja in generiranja informacije. Aplikacije in modalnosti lahko na različnih nivojih združujejo in povezujejo informacijo v skupni rabi. Shema sistemske arhitekture je predstavljena na spodnji sliki:

Kot je razvidno iz slike 2-4, **vmesno opremo** (*Middleware*), v osnovi gradijo funkcionalnosti, ki so nujno potrebne za realizacijo omreženega okolja. Vmesna oprema vključuje semantiko za komunikacijo in zaznavanje storitev in naprav v omrežju, vključujoč obstoječe standarde odkrivanja (UPNP, WS ali SLP). Dodatno pa so v sistem vmesne opreme implementirani še mehanizmi avtentikacije, avtorizacije in enkripcije. **CMS** (*Context Management Service*) je storitev kontekstnega upravljanja in predstavlja odprto infrastrukturo za upravljanje s kontekstno informacijo. Vloga CMS je zajemanje informacije iz različnih virov in združevanje ali abstrahiranje te informacije v višje nivojske kontekstne objekte (kontekstno informacijo), ki služi kot vhod kontekstno zavednim storitvam. Modul **ANS** (*zavedanje in obveščanje*) predstavlja osnovno funkcionalnost, ki omogoča razvoj aplikacij, ki ljudem ali aplikacijam omogočajo z minimalnim naporom zavedati se o pomembnih kontekstnih spremembah. ANS lahko sledi spremembam na različnih kontekstnih nivojih (aktivnosti, stanje okolja, prisotnost ljudi itd.) in z obveščanjem omogoča storitvam aplikacijskega nivoja, sposobnost zavedanja o spremembah. Vsaka aplikacija registrira pravila, s katerimi določa, o katerih spremembah želi biti obveščena. Z uporabniškega vidika, pa ANS uporabnika obvešča s pomočjo intenzitete izrisovanja (rendering), primerne uporabnikovim željam in trenutnemu kontekstu.

Modul za **uporabniško modeliranje in profiliranje**, poskrbi za metodologijo z namenom povečanja učinkovitosti in uporabnosti storitev in vmesnikov. Uporablja se za:

- oblikovanje prikazane informacije glede na uporabnika in kontekst,
 - predvidevanje obnašanja uporabnika v prihodnosti,
 - pomoč pri iskanju relevantne informacije,
 - prilagajanje vmesnikov uporabniku in kontekstu v katerem so uporabljeni,
-

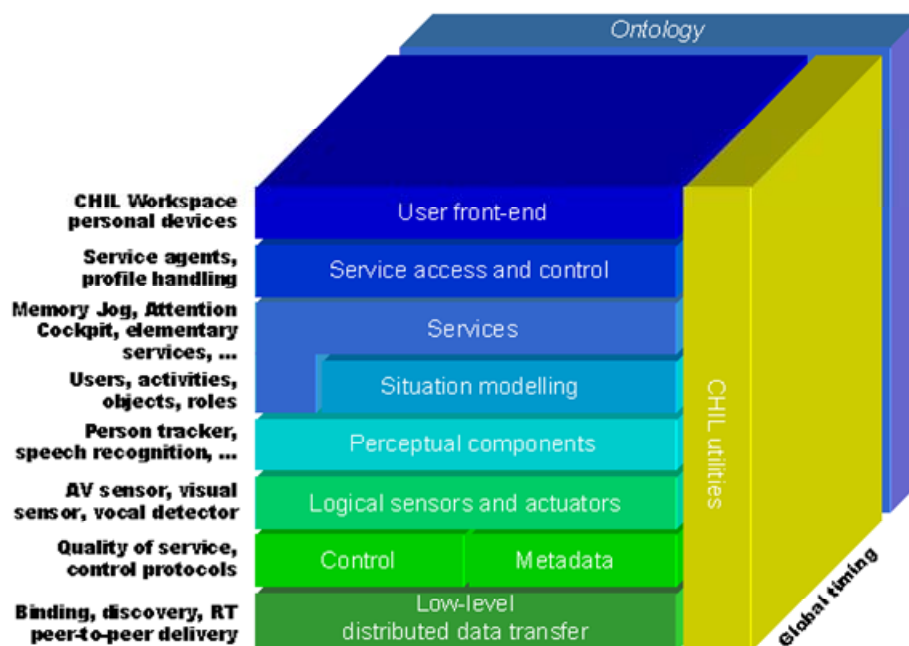
- indikacijo lastnosti vmesnikov in prikaza informacije z namenom prilagoditve več-uporabniškemu okolju.

Predstavljena arhitektura zadostuje potrebam dandanašnjih inteligentnih okolij in hkrati predstavlja arhitekturno zgled delu, ki je bilo izvedeno v okviru diplomskega dela. Vendar pa AMIGO projekt implementira tehnološko zaprt sistem. V več-uporabniških sistemih ni pomembno le zagotoviti množico uporabnih storitev, marveč je nujno tudi ta nabor storitev primerno uporabljati. Vsak uporabnik želi uporabljati druge storitve in v splošnem pričakuje drugačno obnašanje sistema, pri nadzoru in upravljanju z napravami v okolju. Zato je pomembno uporabniku ponuditi takšen sistem, ki ga je mogoče učinkovito in zelo fleksibilno prilagajati uporabnikovim potrebam in željam, brez nepotrebnih zapletenih konfiguracij, ki smo jim pogosto priča. Dodatno mora biti takšen distribuirani sistem, tudi sposoben hkratnega poganjanja več instanc, ki določajo uporabniku prilagojeno obnašanje in delovanje sistema. Dogodki v inteligentnem okolju se navadno prožijo naključno in asinhrono zato je pomembno, da se vse dogodke neprestano zaznava, primerno procesira in na njih primerno odziva in to v realnem času. Projekt AMIGO zgoraj omenjenih lastnosti nima, kar nam je tudi bilo izhodišče pri izvedbi lastne arhitekturne rešitve, ki jo obravnavamo tudi v okviru naloge.

2.1.2 EU projekt CHIL (*Computers In the Human Interaction Loop*)

Projekt CHIL (*Computers In the Human Interaction Loop*) [9], je pomemben predvsem zaradi tehnologij in inovacij, ki jih uvaja v izgradnjo multimodalnih ogrodij namenjenih uporabi v različnih inteligentnih okoljih. Sistemska arhitektura projekta naj bi zagotavljala možnost implementacije interaktivnih scenarijev in je grajena strukturirano. Arhitektura dodatno omogoča še upravljanje multimodalne interakcije. S stališča programske izvedbe, arhitektura vključuje komunikacijske komponente in komponente multimodalne interakcije. Shema arhitekture je predstavljena na sliki 2.5. EU projekt CHIL je pomemben predvsem zaradi načinov in postopkov, ki jih uporablja pri zajemanju, procesiranju multimodalne informacije, sama arhitektura sistema pa predstavlja zelo namensko orientirano okolje in ni najbolj primerna domačim inteligentnim okoljem. **Modul detekcije govora (SAD)** je v projektu CHIL izveden z uporabo linearne analize diskriminant, ki klasificira Mel-kepstalne koeficiente nad

vhodnim avdio signalom in ga klasificira na govor ali ne govor. Za obravnavo ne stacionarnih akustičnih okolij, pa so v projektu vključili metode adaptivnega določanja mej mere zaupanja. Za *akustično detekcijo (AED)*, projekt CHIL uporablja FBK-AED komponente, ki bazirajo na mikrofону z enojnim vhodom. Implementacija predlaganega pristopa pa želi le nazorno pokazati potencial metode za detekcijo in klasifikacijo izoliranega akustičnega dogodka v realnem času.



Slika 2-5: Sistemska arhitektura CHIL[9].

Razpoznavanje gest v okviru projekta CHIL, poteka na podlagi detekcije in razpoznavanja roke. Modul je razvit za namene razpoznavanja kazalnih gest v kompleksnih poslovnih okoljih (npr. sestanki). Sama detekcija in sledenje roki, je izvedeno v treh fazah:

- sledenje glavi in roki,
- detekcija kazalne geste,
- ocena točke v katero gesta kaže.

3D pozicija glave in roke je določena s pomočjo slikovne sekvence, dobljene s stereo kamero. Proces adaptivno klasifikacijo barve kože in stereo informacijo, združuje v ogrodje za sledenje. Ker tako implementiran sledilnik ni odvisen od statičnosti ozadja,

se kamera lahko premika. S samo interpretacijo gest (razen določanja smeri v katero kaže kazalna gesta), pa se projekt CHIL ni ukvarjal. V okviru *razpoznavanja čustev/odnosa na nivoju signalov*, je CHIL implementiral multimodalno detekcijo čustev, ki temelji na analizi obrazne mimike in klasifikacije čustveno obarvanega govora. V okviru razpoznavanja oseb (MPI), modul v projektu CHIL upošteva rezultate akustične in video razpoznavne, ki jih združi in tako pridobi multimodalno informacijo o uporabniškem modelu. Sama fuzija temelji na visoko kvalitetnem vektorskem sistemu in linearni kombinaciji vsakega mono-modalnega rezultata, ki je obtežen z utežjo identifikacije.

2.1.3 *Drugi pomembnejši projekti*

Področje razvoja tako inteligentnih okolij, kot tudi posameznih modulov za inteligentna okolja, je zelo široko in zajema polno obetavnih projektov. Naj omenimo le nekatere med njimi:

- Augmented Multiparty Interaction (AMI) - projekt je pomemben predvsem zaradi inovativnosti v smeri interaktivnosti [10],
- Diagnostic and Intrinsic Variability in Natural Speech (DIVINES) - projekt podaja dodatne predloge za razvoj večjezničnih sistemov za razpoznavanje govora [11],
- Interaction Network on Emotion (HUMAINE) - projekt predstavlja uporaben pristop k implementaciji razpoznavanja čustev [12],
- NICE (Natural Interactive Communication for Edutainment) - projekt predstavlja dobro izhodišče za nadaljnji razvoj multimodalnih krmilnikov dialoga [13].

Glede na postavljene zahteve o funkcionalnosti inteligentnih okolij, bomo v nadaljevanju predstavili glavne fizične komponente inteligentnega okolja, ki omogočajo okolju procesirati in generirati tudi multimodalno informacijo. Največji poudarek je namenjen mobilni robotski enoti, ki se kot novost vpeljuje v takšna okolja v vlogi osebnih robotskih enot. Na videz trivialna povezava inteligentnega okolja in mobilne robotske enote, predstavlja velik korak v smislu optimizacije procesov in dviguje raven učinkovitosti in fleksibilnosti upravljanja z okoljem ter dviguje tudi naravnost komunikacije med uporabnikom in napravami v inteligentnih okoljih. Zato bomo v poglavju o mobilni robotski enoti, predstavili tudi samo predlagano idejno zasnovo takšne enote in želene značilnosti.

2.2 Komponente inteligentnih okolij

V fizičnem smislu lahko inteligentno okolje razdelimo na tri logične enote:

- zajemanje podatkov (senzorji, kamere, mikrofoni itd.),
- procesiranje podatkov (strežniki, računalniki),
- generiranje podatkov v obliki odzivov (obnašanje nekaterih naprav, multimodalni odzivi v obliki pogovornih avatarjev).

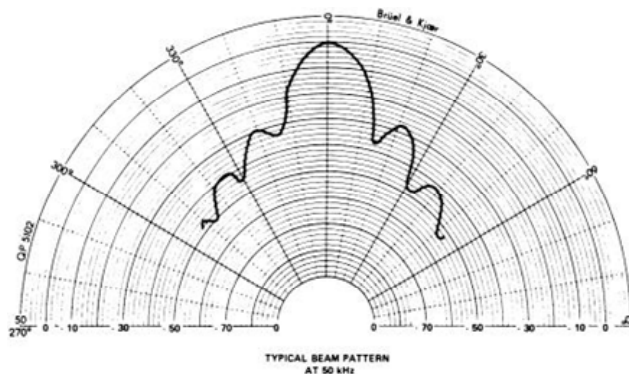
V tem poglavju se bomo omejili predvsem na logično enoto zajemanja podatkov, v okviru katere bomo izpostavili določene senzorje in avdio/video naprave. Inteligentna okolja namreč multimodalne podatke zajemajo z uporabo senzorjev in različnih naprav za zajemanje avdia in videa. Ti podatki predstavljajo pravzaprav multimodalni vhod sistema, na podlagi te informacije pa se tudi definira obnašanje sistema. Kvaliteta multimodalnega vhoda (točnost senzorjev, fokusiranost zvoka in slike, SNR (signal-to-noise ratio), ...) je namreč ključna za uspešno procesiranje in generiranje informacije v inteligentnih okoljih.

2.2.1 Senzorji

a) *Zvočni (ultrasonic) detektorji* – so znani tudi kot prevodniki, ki lahko sprejemajo in pošiljajo podatke delujejo na principu radarja ali sonarja, ki oceni karakteristike opazovanega objekta na podlagi odboja radijskih ali zvočnih signalov. Zvočni senzorji generirajo visoko frekvenčne zvočne valovne fronte in ocenijo odziv na podlagi sprejetega odboja. Za oceno razdalje senzorji merijo čas med izhodnim valom in prejetim odzivom. Zvočni detektorji se v inteligentnih okoljih uporabljajo predvsem za merjenje razdalj (detekcija ovir) in karakterističnega opisa posameznega objekta (predikcija o obliki objekta). Pri izbiri senzorjev pa moramo biti pozorni predvsem na usmerjenost karakteristike senzorja (**Slika 2-6**). Bolj ko je karakteristika senzorja usmerjena, bolj natančen bo rezultat meritve.

b) *Pasivni IR senzorji PIR* – delujejo na principu merjenja infrardeče radiacije objektov. V inteligentnih okoljih se uporabljajo predvsem za gradnjo sistemov detekcije gibanja. Gre za princip ko nek objekt, ki seva določen IR spekter (ima določeno temperaturo), stopi pred objekt z drugačno temperaturo (drugačnim spektrom sevanja).

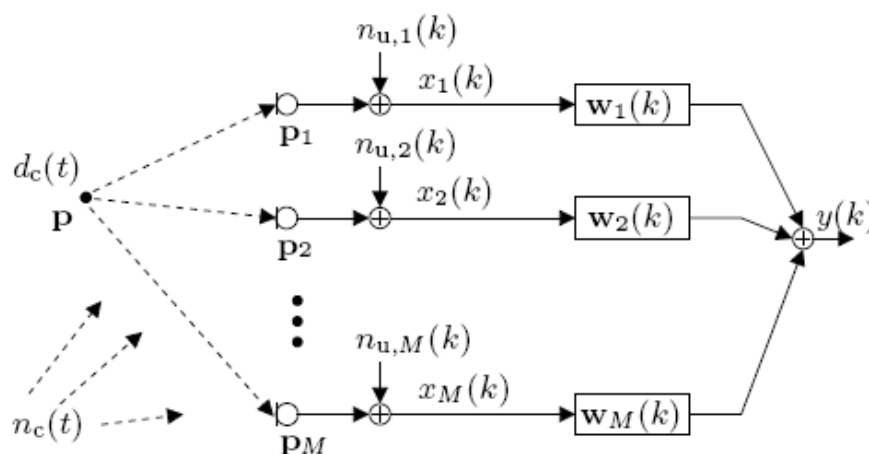
Poleg zgoraj omenjenih tipov senzorjev so v inteligentnih okoljih največkrat uporabljeni še odometri, barometri, temperaturni senzori in "gyros" senzori, emitorski senzori in detektorji, bio senzori, RFID značilke itn.



Slika 2-6: Tipična karakteristika senzorja (vzorec žarka).

2.2.2 Kamere in mikrofonska polja

S pomočjo mikrofonskih polj in visoko kvalitetnih stereo kamer, zagotavljamo kvalitetno zajemanje avdio in video informacije. Kvaliteta zajemanja avdia, se lahko dodatno izboljša z uporabo t.i. prilagodljivega generatorja snopa (adaptive beam former) (Slika 2-7), to je naprave, ki zmore za dano geometrijo polja iz nestacionarnih vmesnikov optimalno izločiti tako želene nestacionarne signale, kot tudi šum. V rezultirajočem signalu je NSR razmerje zmanjšano.



Slika 2-7: Splošni "beamformer" tipa filtriraj & seštej.

Ob linearnem mikrofonskem polju lahko uporabljamo še manjše mikrofone oblike T, ki se v praksi uporabljajo za akustično lokalizacijo vira. Kot dodatek za kvalitetno zajemanje ali prenos zvoka v zelo šumnih okoljih, lahko inteligentna okolja omogočajo tudi brezžične ali Bluetooth mikrofone. Za zajemanje slike v realnem času uporabljamo barvne stereo kamere. Video se zajema v toku, saj je procesiranje hipnih sprememb lahko z uporabo toka slik oteženo. Sistem po potrebi video okviri in procesira samo posamezen okvir.

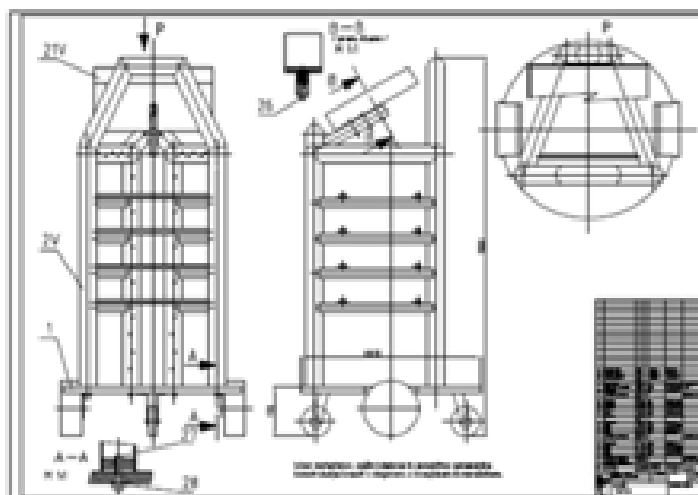
2.2.3 Mobilna robotska enota

Na prvi pogled trivalna povezava, vendar z veliko dodano vrednostjo, ki prispeva predvsem k bolj naravnemu načinu komunikacije v inteligentnih okoljih. Običajno razvoj kljub sorodnosti inteligentnih okolij in mobilnih robotskih enot (*Mobilna robotska enota lahko predstavlja namensko inteligentno okolje na kolesih*) poteka ločeno, vendar vzporedno. Za ločen razvoj se glavni razlog skriva predvsem v specifičnih pogojih pod katerimi enota deluje. Gibanje in omejena moč (tako procesorska kot napajalna) predstavljata največji problem v razvoju mobilnih enot. Tako obstajajo ločene namenske enote za vožnjo po težkem terenu (mobilne enote za reševanje), spremljevalne enote (life companion) itn. Razvoju vseh tipov enot pa je skupna zahteva po avtonomnosti (delne ali popolne), multimodalnosti in razumevanju in inteligenci. Skupaj z Roboti c.s. d.o.o smo prišli do naslednjega spoznanja:

Mobilna robotska enota mora biti avtonomna in kognitivna, sposobna skupnega delovanja v večjih enotah (avtonomno porazdeljevanje dela med komponentami) in nujno mora ustrezati psiho-sociološkimi standardom okolja ali skupine, kateri je namenjena (EUROP). V okviru oblike trenutne študije kažejo, da evropski prostor želi zaobljeno enoto, ki bo izražala osebnost. Linije ne smejo biti preveč agresivne ali ostre, saj se enota s takšno obliko lahko enostavneje giblje tudi v zelo dinamičnem okolju. Najbolj primerni materiali za izgradnjo ohišja enote pa so les v kombinaciji z umetnimi masami in kovino. Na podlagi poprej omenjene študije smo zasnovali model mobilne robotske enote, ki ustreza normativom povprečnega zahodno evropskega prebivalca in je predstavljen na spodnjih slikah:



Slika 2-8: Prototip mobilne robotske enote.



Slika 2-9: Model mobilne robotske enote.

Zgornja slika predstavlja model mobilne robotske enote in implementacijski diagram enote izdelav v programskem okolju AutoCad 2008[44].

Karakteristike mobilne robotske enote

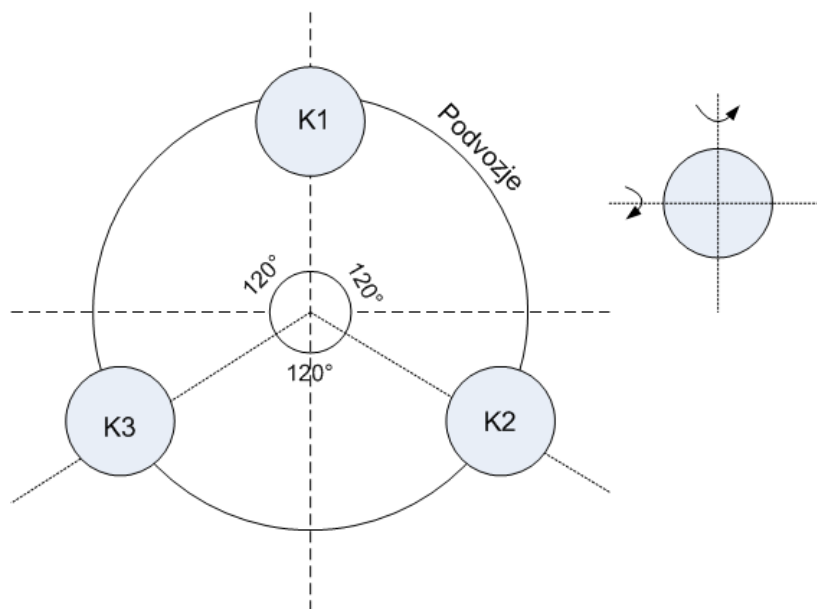
Avtonomnost mobilne robotske enote lahko opišemo na dveh nivojih, primarnem in sekundarnem. Primarni nivo avtonomnosti poskrbi, da enota pridobi informacije o okolju v katerem se nahaja in da se enota v tem okolju lahko prosto giba. Sekundarni nivo avtonomnosti, predstavljajo za delovanje enote ne-nujne funkcionalnosti, ki

predstavljajo dodano vrednosti, predvsem v smislu enostavnosti uporabe in ceni izdelave. Nivoja avtonomnosti sta predstavljena v spodnji tabeli:

Nivo avtonomnosti	Karakteristika	Izvedba
Primarni	<i>Gibanje</i>	dobro zasnovan pogonski sistem, aplikacija za avtonomno gibanje (lokalizacija, načrtovanje poti, izogibanje ovir)
	<i>Polnjenje</i>	procedure lokalizacije in načrtovanja poti
Sekundarni	<i>Samostojno vzdrževanje</i>	logična in mehanska diagnostika, procedure za odpravo napak
	<i>Učenje</i>	procedure učenja primernega odziva v primeru napake, vmesnik človek-stroj (HMI – human machine interface)
	<i>Skupinsko obnašanje</i>	procedure za proizvodbo in pridobivanje le najnujnejše informacije

Tabela 2-1: Nivoji avtonomnosti mobilne robotske enote.

Za namene gibanja mobilne robotske enote, se uporablja lahko **simetrični pogonski** sistem. To je kolesni sistem, ki je postavljen simetrično na vsaki strani enote in v njeni sredini. Vsako kolo deluje samostojno in omogoča enoti gibanje v naslednje smeri: naprej, nazaj in rotacija na mestu. Vsako kolo mora biti dovolj veliko, da lahko enota neovirano premaguje ukrivljenost in neenakomernost višine tal prisotne v domačem okolju. Drugi način implementacije pogonskega sistema predstavlja **holonomski sistem**. Trokolesni sistem uporablja "omni" kolesa, ki so locirana 120° relativne pozicije glede na ostala kolesa. "Omni" kolo je kolo, ki se lahko vrti hkrati okrog dveh pravokotnih osi. Za razliko od simetričnega voznega sistema, ki omogoča gibanje koles naprej in nazaj, holonomski sistem omogoča gibanje enote v katero koli smer. Holonomski pogonski sistem iz zgornjega opisa je predstavljen na sliki 2-10.



Slika 2-10: Holonomski vozni sistem.

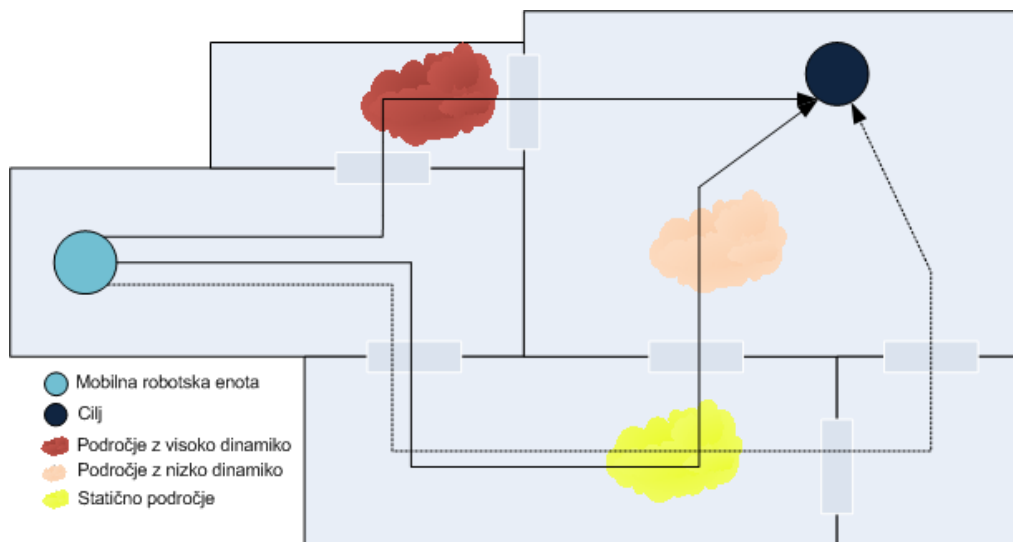
Pogonski sistem enote se izvede z uporabo brez krtačnih (brushless) DC motorjev, ki jih odlikuje visoka stopnja učinkovitosti in večji pospeški (torque). Ti motorji so hkrati cenovno primerni in imajo nižjo stopnjo hrupa pri visokih hitrostih in nizkih napetostih. Napajanje enote se lahko izvede z več tipi baterij (akumulatorjev). Primerni so predvsem Li, Ni-Cd in svinčeni akumulaterji oz. baterije. Pomembno pri izbiri baterije je tako njena kapaciteta kot tudi hitrost polnjenja.

Avtonomijo gibanja predstavljajo mehanizmi lokalizacije, mapiranja z navigacijo in izogibanja ovir. V smislu lokalizacije je najbolj primerna uporaba kombinacij oddajnih senzorjev na vhodih v sobo in sprejemnega senzorja na mobilna enoti. Z vsakim prehodom senzorja se na podlagi identifikacijske številke sobe (labele), enota "zaveda" v kateri prostor vstopa. Za točnost lokalizacije v sobi, pa dodatno poskrbi modul za mapiranje (mapping engine). Procedura za izogibanje ovir, ki predstavlja še eno izmed nabora procedur primarnega nivoja avtonomnosti, se izvede lahko tudi z uporabo evolucijskih algoritmov. V primeru detekcije ovire mora enota ponovno izračunati najbolj primerno pot in pri tem upoštevati tako oviro na njeni trenutni poti, kot vse ovire, ki se lahko nahajajo na ostalih poteh (vnešene v model prostora).

Modul za mapiranje prostorov, na mapi modela sobe označi trenutno pozicijo enote v prostoru. Uporablja se kot pripomoček pri lokalizaciji in navigaciji enote. Mapiranje

vključuje klasifikacijo objektov na podlagi obteževanja objektov glede na stopnjo dinamičnosti premikanja. Tako so objekti, ki so v prostoru statični, najbolj uteženi, objekti katerih pozicije pa ni moč predvideti (visoka dinamika gibanja), pa bodo imeli najmanjšo uteženost. Začetno uteženost definira enota sama v sodelovanju s podobnimi procedurami, implementiranimi v programskem okviru inteligentnega okolja. Med procesom nenehnega učenja tako prostor kot mobilna enota skupaj prilagajata uteženost posameznih predmetov.

Sledenje poti predstavlja postopek, ki mobilni enoti omogoča pomniti trenutno opravljeno pot in shraniti vse anomalije, ki jih najde na tej poti (novi objekti, ki jih še ni na mapi itd.). Enota objekte razpozna s pomočjo metod za razpoznavanje objektov implementiranih na sami enoti in tudi kot del inteligentnega okolja. Trenutna informacija se prilagodi ali nadomesti z "novo" informacijo glede na pomembnost predmeta in lokacijo, kjer se je dinamična sprememba zgodila in zaznala (npr. stol je namesto za mizo, postavljen na sredino jedilnice).



Slika 2-11: Izbira alternativne poti.

Mobilna robotska enota lahko s pomočjo vnaprej omenjenih procedur lokalizacije, mapiranja in sledenja poti, vnaprej predvidi najprimernejšo pot. Procesu pravimo načrtovanje poti (route planning). Enota za ta namen hrani 3D model prostora (mapo) v katerem se nahaja. Mobilna robotska enota sicer lahko načrtuje pot sama, kadar je cilj

znotraj mej trenutnega prostora ali kadar želi načrtovati le pot do meje trenutnega prostora (3D model naslednjega prostora še ni bil izveden). V primeru, da se cilj nahaja zunaj mej trenutnega 3D modela, mobilna enota pot načrtuje v sodelovanju z inteligentnim okoljem (ki hrani vse izdelane 3D modele prostorov danega okolja). Izhod iz prostora se določa glede na relativno pozicijo enote in možnih prehodov v prostor, kjer se cilj nahaja. Na prehodu v nov prostor mobilna robotska enota pridobi nov 3D model okolja, in v primeru ovire na poti, ponovno načrtuje in prilagodi nadaljnjo pot.

Slika 2-11 prikazuje poti, ki jo na primer mobilna enota izbere (črtkano) in dve alternativni poti do cilja. Na podlagi podatkov o dinamiki območja, ki ga enota mora preiti v sodelovanju z inteligentnim okoljem, enota izbere najboljšo možno pot. Glavno utež izbire predstavlja prav podatek o dinamiki, vendar pa na izbiro vpliva dodatno še z utežmi za dolžino opravljene poti, za število potrebnih zavojev, za predvideno število 3D modelov, ki jih bo potrebno izmenjati z okoljem (predvideno število prehodov skozi oddajne senzorje).

Sekundarni nivo avtonomnosti se ukvarja s samostojnostjo mobilne robotske enote v smislu vzdrževanja in z metodami učenja in pristopi k skupinskemu oziroma kolaborativnemu obnašanju. Samostojnost v okviru vzdrževanja predstavlja zmožnost avtonomnega napajanja enote. Algoritmi temeljijo na principu "najdi najbližji vir napajanja in se zapelji tja". Tako enota uporablja vnaprej opisane metode sledenja in načrtovanja poti, kot dodatno utež pa je vključena predvidena poraba moči na dani poti. Hkrati lahko tudi prilagodi svoje obnašanje in po potrebi začasno izklopi ne-aktivne module in s tem zmanjšuje porabo energije (npr. merjenje temperature, zajem zvoka; v primeru da se nahaja v isti sobi kot je terminal, lahko enota tudi začasno prekine povezavo z inteligentnim okoljem, itn.).

Logično in fizično diagnosticiranje sta postopka s katerima enota ocenjuje in preverja pravilnost delovanja in sposobnost delovanja. Enota jih običajno opravlja periodično.

- *Logična lastna diagnoza* predvideva uporabo modularnih verifikacijskih algoritmov namenjenih preverjanju pravilnosti delovanja kode. Glede na karakteristični vhod lahko mobilna robotska enota primerja predviden izhod z izračunanim. V primeru napake, lahko enota začasno onemogoči delovanje modula. Pristop logične diagnoze in učinkovitega obravnavanja napak, je omogočen z uporabo XML opisov

obnašanja, ki se uporabljajo v distribuiranem spletnem ogrodju (web based distributed framework).

- *Fizična diagnoza* skrbi za preverjanje pravilnosti točnosti senzorjev, pravilnosti delovanja električnih komponent (motorji, mikrokrmilniki, itn.) in stanja komunikacijske povezave. V primeru odkrite napake ali delovanja modula na meji pravilnosti, enota začne proceduro testiranja dotičnega modula z uporabo testne sekvence za dani modul, ki poda oceno o potrebi in zmožnosti odprave napake (začasna uporaba nadomestnega modula ali rekombinacija delujočih modulov). V primeru nezmožnosti poprave napak, enota to samodejno javi v obliki poročila o nedelovanju modula, na kar se na podlagi testnega vzorca lahko predvidi vzroke za nastalo napako in definira postopke za ustrezno reševanje napak.

V nadaljevanju bomo predstavili pojem multimodalnosti, ki inteligentnim sistemom omogočijo naprednejše kognitivno kontekstno obnašanje. Pojem multimodalnosti predstavljamo v obliki pristopov, ki so v stroki dandanes predmet raziskav in razvoja in so uporabljeni v okviru projektov, ki predstavljajo najnovejše tehnologije na področju inteligentnih okolij. Nabor predstavljenih pristopov zajema tako nizko nivojska procesiranja na vhodu sistema (npr. detekcija govora), kot tudi visoko nivojska procesiranja, ki zahtevajo abstrakcijo informacije in uporabo raznih metod združevanja različnih monomodalnih vhodov (npr. razpoznavanje emocij).

2.3 Multimodalnost

Multimodalnost v inteligentnih okoljih omogoči visoko stopnjo kognitivnega kontekstno usmerjenega razumevanja in odzivanja sistema. Poleg predhodno omenjenih pristopov, ki so uporabljeni v Amigo in Chil projektih, naj omenimo še nekatere druge multimodalne pristope, ki so pomembni za kvalitetno procesiranje multimodalne informacije.

Detekcija govora (SAD), uporablja mikrofonska polja in predstavlja obetaven pristop za izboljševanje avtomatskega razpoznavanja govora (ASR) oddaljenega govora. Zaradi prisotnosti odmeva en sam oddaljen mikrofonski težko omogoči razpoznavanje govora z zadovoljivo gotovostjo, razen če lahko zagotovimo popolnoma enake akustične pogoje (v kontekstu akustike okolja, lokalizacije govornika in postavitve) tako v fazi učenja kot v fazi testiranja. Poleg odmeva je kritična komponenta akustičnega signala razmerje

signal-šum (SNR). SNR se v primeru prisotnosti dodatnega vira zvoka (televizija, računalnik, hladilnik, drugi govorec) občutno zmanjša, saj vsak dodaten zvok predstavlja šum v zajetem signalu.

Mikrofonska polja so z uporabo poprej omenjenega principa tvorjenja snopov zelo dobro usmerjena in zagotavljajo višjo kvaliteto zajetega zvoka. Razvoj algoritmov detekcije govora teži k uporabi poljubnega nabora polj akustičnih senzorjev, ki zahteva uporabo alternativnih pristopov pri združevanju zajete informacije. Eden najbolj obetavnih pristopov predstavlja fuzijo v hipotetični domeni in ne več na nivoju signalov. Fuzija se izvede na osnovi valčnih (Wavelet) klasifikatorjev in klasifikatorjev z uporabo nevronske mreže (ANN). Projekt CHIL na področju SAD, predlaga uporabo linearne diskriminantne analize (LDA) z uporabo Mel-kepstralnih koeficientov vhodnih avdio okvirjev, pri klasifikaciji na govorne in negovorne signale. Zaradi uporabe SAD v ne-stacionarnih okoljih, pa je predlagana prilagoditev tehnik adaptivnega določanja mej.

Razpoznavanje govora in naravno razumevanje jezika predstavljata pomemben proces v sklopu razumevanja kognitivnega okolja. Dominantni princip, ki se uporablja za razpoznavanje govora, so t.i. prikriti modeli Markova – HMM [15] (). Algoritmi avtomatskega razpoznavanja vključujejo monofonske, trifonske ali besedne statične Markove modele (akustični modeli), ki temeljijo na Bayesovi verjetnostni in klasifikacijski teoriji [21]. Informacija, ki jo nosi jezikovni model pa je uporabna predvsem pri razpoznavanju tekočega govora. Kvalitetni sistemi za razpoznavanja in razumevanje jezika vključujejo eno ali večjezični modul, ki je zmožen razpoznavati izolirane besede ali besedne zvezne (npr. ukazi napravam: "NAPREJ", "NAZAJ", "POMIK NAPREJ", "KAMERA LEVO") in do določene mere lahko razpozna tudi tekoč govor (domensko omejeno neodvisno razpoznavanje tekočega govora) v različnih kontekstih. Razpoznavanje se prilagodi omejenim domenam preko akustičnega učenja, optimizacije ASR parametrov, jezikovnega modeliranja in gradnje slovarja. V primeru večjezične podpore so glavni parametri, ki zaradi posebnosti procesa razpoznavanja vplivajo na kvaliteto razpoznavanja, način govora, mogoče disfluence in govor v ozadju. Posledično je eden izmed boljših načinov razpoznavanja večjezičnega govora grajen na osnovi "ASR N-best liste" ali hipotetičnih besednih grafov in uporabe sintaksno-sematičnega razčlenjevanja besed na osnovi domenske ontologije. Enega

robustnejših razčlenjevalnikov s končnim številom stanj je razvilo podjetje NISLab[14]. Razpoznavalnik se uporablja za razpoznavanje nepopolnega govora, kjer se N-najboljših izhodov poda kot vhod v postopek fuzije.

Projekt DIVINES na področju razpoznavanja govora predstavlja inovativne pristope k izboljševanju mere zaupanja razpoznavalnikov (pravilnosti razpoznanih besed) naravnega govora. V proces razpoznavanja dodaja inovativno pristope za izbiro mer zaupanja brez potrebe po ponovni ponovitvi besede, ki jo je uporabnik izgovoril [22].

Naravna sinteza govora, ki naj bo čim bolj naravna, mora v generiranje govora vključiti tudi čustvene komponente, karakterizacijo govorca in ekspresivnost generiranega govora. Te komponente se vključujejo v fazi post-procesiranja govornih signalov z uporabo raznih tehnik preoblikovanja. Naravnost govora bistveno izboljšamo z uporabo prozodičnega modeliranja, ki je odvisno od semantične informacije jezikovnega procesorja. Dodatno zahtevo procesa sinteze govora, pa predstavlja večjezičnost, ki jo dosežemo s sintaktično analizo, ki vpeljuje psiho-lingvistično združevanje besed v jezikovne skupine. Trenutna tehnologija naravne sinteze govora uporablja predvsem konkatenacijsko tehniko, to je zlepljanje enot govora [16], ki jih pridobimo iz velikih govornih baz, običajno nevtralnega načina izgovarjave. Modifikacija prozodije, ekspresivnosti ali načina govora se izvaja z uporabo parametrov, ki karakterizirajo produkcijske modele govora. Čustvene komponente (prozodija čustev) v govoru označujejo individualnost govorca [17][18]. Klasični pristopi prozodije temeljijo na tonskih modelih [19], novejši pristopi pa skušajo vpeljati uporabo korpusa prozodične informacije, iz katerega se izbere prozodična kontura, ki je najbolj primerna za dano jezikovno strukturo [20].

V sklopu multimodalnosti je naslednje pomembno področje *prepoznavanje čustev/odnosa na nivoju signalov*. Projekt HUMAINE je definiral šest tematskih področij razpoznavanja emocij: teorija emocij, signalni/znakovni vmesniki, struktura emocijsko obarvane interakcije, kognitivnih emocij in dejanjih. Obstaja več principov, ki se uporabljajo za tovrstno razpoznavanje. Principi se v osnovi ločujejo po številu emocijskih stanj, ki jih razpoznavajo oz. klasificirajo. Za kognitivno inteligentno okolje primeren in relativno poenostavljen princip uporablja 6 stansko klasifikacijo emocij, in sicer: veselje, jezo, strah, zgražanje, žalost in presenečenost. Osnovni model

razpoznavanja uporablja dvo-nivojsko strukturo HMM klasifikatorjev in upošteva kratkočasovne in dolgočasovne značilnosti informacije. Na prvem nivoju se določijo nizko-nivojske lastnosti, na drugem nivoju pa se določajo visoko-nivojske lastnosti. Pri implementaciji sistema se vedno pojavlja dilema med uporabo kratkočasovnih ali dolgočasovnih lastnosti signalov, raziskave pa so usmerjene bolj v uporabo dolgočasovnih lastnosti. Dobra primera uporabe tako enih kot drugih lastnosti sta:

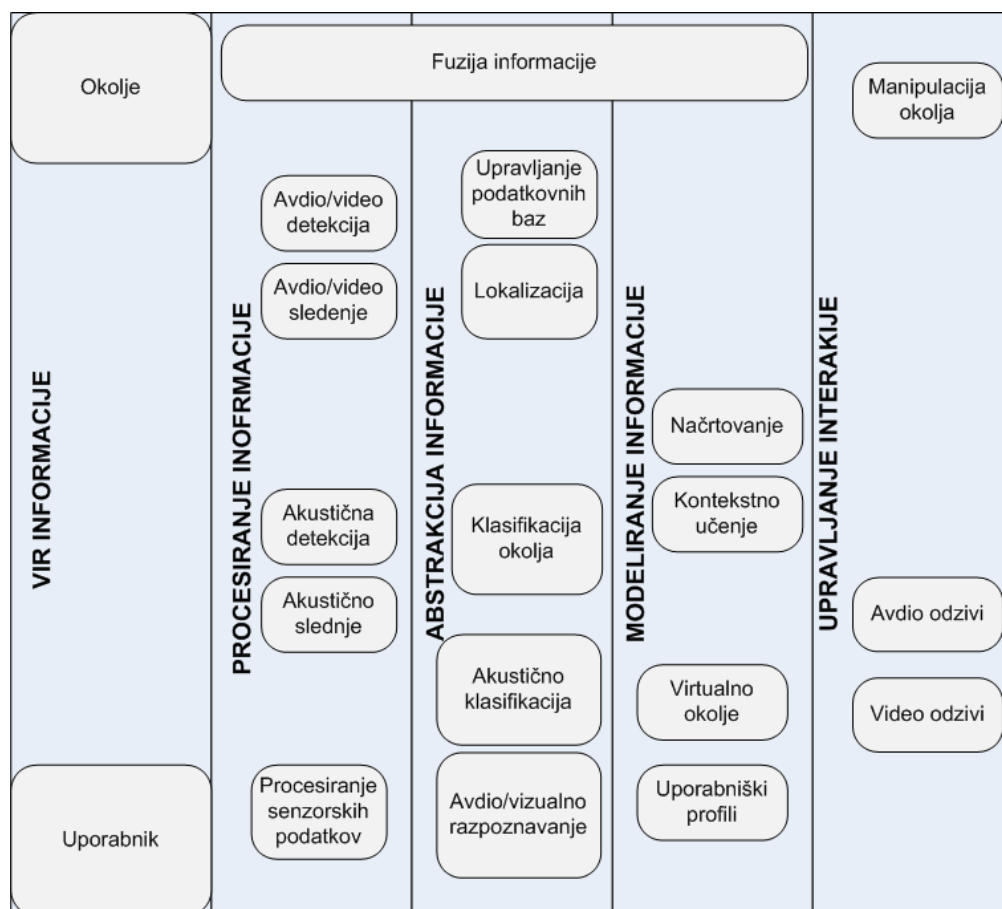
- V [23] se za razpoznavanje osnovnih emocij uporablja kratkočasovne lastnosti in prikrite modele Markova (HMM).
- V [24] se za razpoznavanje emocij uporablja dolgočasovne lastnosti, nevronske mreže in meritve razdalje med emocijskimi značilkami.

Informacija, ki jo nosi človeški obraz predstavlja ključni element v procesu razpoznavanja čustev, obrazna mimika pa vsebuje bogat nabor parametrov, ki govorijo o človekovih občutjih in mislih. Izmenjava informacije, ki jo nosi človeški obraz zato predstavlja enega izmed pomembnejših informacijskih kanalov in ključen korak v smeri uporabne, intuitivne in multimodalne interakcije človek-stroj. Informacijo, ki jo nosi obrazna mimika [25], na podlagi obdelave statičnih slik, zajetih ob največji izraženosti posameznih emocij (ko so izražene emocije najbolj izrazite), razdelimo v šest osnovnih emocijskih razredov. Večina pristopov vključuje izgradnjo video baz s posnetki, kjer uporabnik na zahtevo izrazi določeno čustvo, šele nato pa se lotijo problema klasifikacije. Ti pristopi v večini primerov temeljijo na kodnem sistemu FACS (*Ekman and Friesen Facial Action Coding System*), oštevilčenju vseh področij obraza, ki določajo obrazno mimiko (gibanje obraza). Kot rezultat dobimo velik nabor možnih obraznih izrazov. Namesto FACS sistema pa se lahko za razpoznavanje obrazne mimike uporabi tudi fizične modele glav, ki vključujejo mišice in kožo. Takšni pristopi rezultirajo v kvalitetnem prepoznavanju obrazne mimike, kadar so pogoji delovanja in razpoznavne striktno določeni [26]. V naslednjem poglavju bomo predstavili zasnovo kognitivne arhitekture, ki jo razvijamo za razna inteligentna okolja. Zasnovana arhitektura predstavlja nov pristop k snovanju in oblikovanju arhitektur inteligentnih okolij in omogoči izvedbo funkcionalnosti, ki jih inteligentno okolja mora ponujati (*glej poglavje 2*).

3 ZASNOVA KOGNITIVNEGA INTELIGENTNEGA OKOLJA

3.1 Kognitivna arhitektura

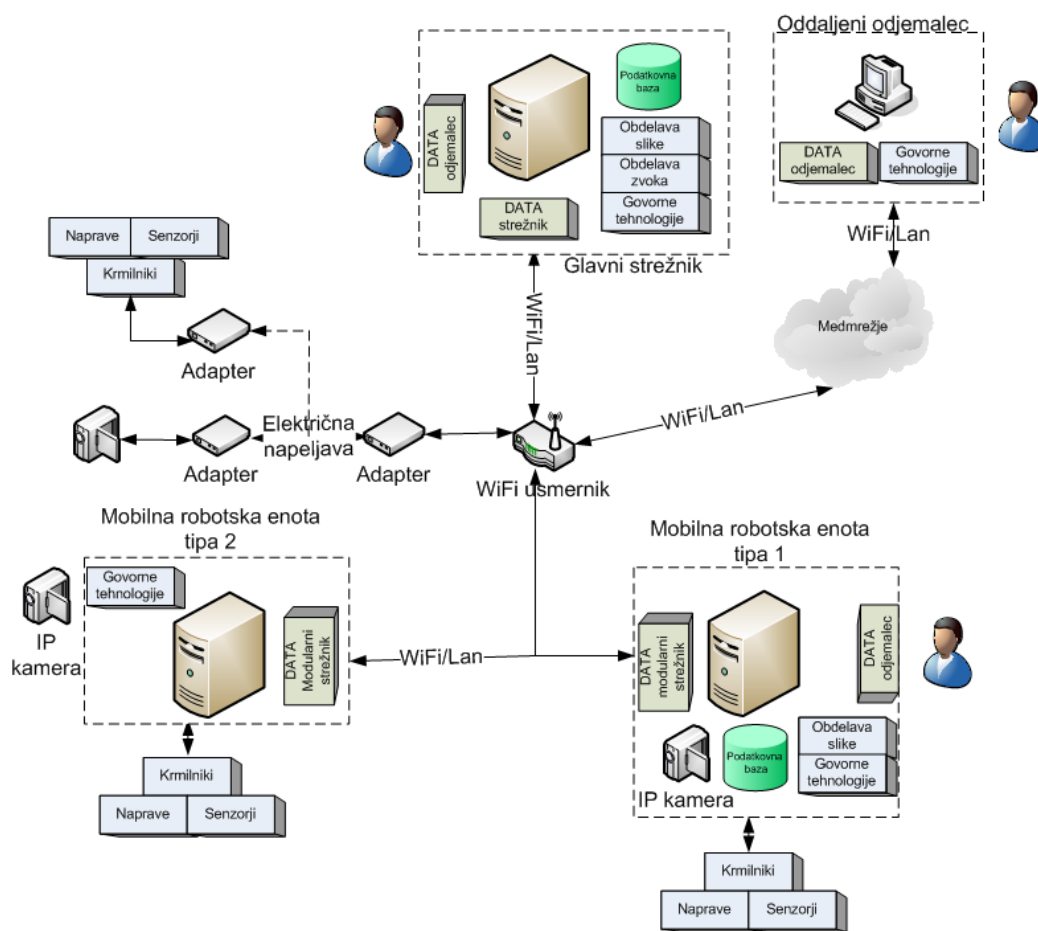
V okviru diplomske naloge smo zasnovali kognitivno arhitekturo za razna inteligentna okolja. Kot je prikazano na sliki 3-1, se prikazana arhitektura deli na pet logičnih slojev.



Slika 3-1: Zasnova kognitivne arhitekture.

Na sliki so prikazane logične povezave med posameznimi komponentami kognitivne arhitekture inteligentnega okolja. Uporabnik predstavlja glavni vir multimodalne informacije. Sekundarni vir informacije v obliki akustičnih signalov in senzorskih meritev pa zagotavlja okolje. Procesiranje zajete informacije se izvede s procesi kot so: detekcija, sledenje in procesiranje senzorskih podatkov. V tej fazi sistem že logično združuje posamezne podatke v manjše logične enote. Tako se na primer fuzija izvede nad podatki, dobljeni z detekcijo zvoka/gibanja, senzorsko informacijo o objektih

(dimenzije objektov, barva objektov, itn.), avdio/video signalov, katerih vir in posledično pomen je znan. Primerno združeni podatki se nato prenesejo v nivo *abstrakcije*, kjer se formirajo skupine podatkov drugega nivoja. Združena informacija na tem nivoju je zahtevnejša za razumevanje in za uporabo že zahteva metode kognitivnega razumevanja. Višje nivojske skupine so tako npr.: ime uporabnika, besedni ukaz, klasifikacija modela okolja, itn. Brez kontekstne informacije so ti podatki več ali manj neuporabni. Podatki drugo nivojske abstrakcije se prenesejo v nivo modeliranja. V tem nivoju se tvorijo/posodablajo uporabniški modeli, okoljski modeli ter generirajo akcije, ki naj se ob nekem dogodku izvedejo (določa se njihova oblika in primernost). Zadnji nivo arhitekture predstavlja upravljanje interakcije (krmilnik dialoga). Na tem nivoju se oblike odzivov pretvorijo v akcijo, vizualne in/ali zvočne signale.



Slika 3-2: Sistemska arhitektura DATA.

3.2 Arhitektura kognitivnega inteligentnega sistema »DATA«

Sistemska arhitektura kognitivnega inteligentnega sistema »DATA« je predstavljena na sliki 3.2 in jo sestavljajo naslednji moduli: en glavni strežnik, ena ali več mobilnih robotskih enot in ena ali več uporabniških vmesnikov (odjemalcev). Gradniki sistema so med seboj povezani preko WiFi ali LAN povezav. Za namene programskega ogrodja je predvidena uporaba prilagojene zvezdne topologije omrežij.

Zvezdna topologija omogoča učinkovit tok podatkov med posameznimi moduli/gradniki in varovan dostop do modulov znotraj lokalnega omrežja inteligentnega okolja. Topologija hkrati zagotavlja hkraten dostop do modulov kadarkoli in od koderkoli. Brezžični usmerjevalnik služi kot centralna komunikacijska točka in omogoča implementacijo pravil in omejevanja dostopa. Vse oddaljene povezave pa najprej obravnava glavni strežniški modul, ki jih posreduje ostalim modulom znotraj omrežja, katerim so povezave namenjene. Kot prikazuje zgornja shema, model sistema DATA sestavljajo DATA strežnik (DS), ena ali več DATA modularnih strežnikov (DMS) in ena ali več modulov DATA odjemalec (DC). DATA strežnik predstavlja glavno vhodno točko sistema. Vse povezave med DATA klienti in DATA modularnimi strežniki (razen povezave $DC \leftrightarrow DMS$ na sami enoti) se vzpostavljajo preko DATA strežnika, ki na podlagi prejetega scenarija odloči na kateri DATA modularni strežnik se bo odjemalec povezal. DATA odjemalec lahko z napravami in mobilnimi enotami v inteligentnem okolju upravlja glede na uporabniški profil, ki ga določuje izbrani scenarij, ki ga uporablja DATA odjemalec. DATA strežnik ima direktni dostop do vseh naprav v inteligentnem okolju (ip kamere, krmilniki naprav, inteligentne naprave, itn.) in posredni dostop do naprav nameščenih na mobilne enote, ki delujejo v okviru inteligentnega okolja. DATA strežnik lahko hkrati streže več DATA klientom in pridobiva ter implementira različne uporabniške scenarije. Hkratnost delovanja se odraža tudi na področju hkratnega nadzora več mobilnih enot in naprav, direktnega in posrednega zajemanja podatkov ter upravljanja (v obliki procesiranja in/ali shranjevanja) z njimi. Storitve, ki zahtevajo večje sistemske vire (podatkovna skladišča, procesiranje slik in videa, učenje) se lahko opravljajo porazdeljeno na več fizičnih strežniških enotah (gručenje (clustering) – generiranje virtualne delovne postaje). DATA modularni strežnik ima dostop do naprav in senzorjev priključenih na

posamezno enoto in opravlja nalogo sekundarne komunikacijske točke. Glavna razlika med DATA strežnikom in DATA modularnim strežnikom je v sposobnosti vzpostavljanja povezave. DATA strežnik lahko vzpostavi in prekinja povezavo z DATA modularnim strežnikom, DATA modularni strežnik pa le čaka na "ukaze" posredovane s strani DATA strežnika in ne more vzpostavljati direktne povezave z drugim DATA modularnim strežnikom. Vloga DATA odjemalca DATA sistema v inteligentnem okolju, je v izvajanju napredne multimodalne interakcije človek-stroj. Uporabniki lahko z uporabo DATA odjemalca komunicirajo s sistemom na naraven način (govor, miška tipkovnica, igralna palica, itn.) , saj DATA odjemalec vsebuje ustrezno podporo in ima pripravljeno osnovo za vključevanje naprednih multimodalnih interakcijskih rešitev (npr. geste). Ker so žične povezave med vmesnikom in uporabnikom nezaželeni, podpira DATA odjemalec IR, WiFi, BlueTooth in radijske tehnologije vhodnih naprav za upravljanje z inteligentnim okoljem in napravami, ki jih okolje s predstavljeno DATA arhitekturo lahko gosti. Izmenjava podatkov v inteligentnem okolju z DATA arhitekturo lahko poteka tudi preko električnega omrežja. Komunikacija poteka po istih protokolih kot WiFi in LAN komunikacija. Komunikacija preko električnega omrežja služi kot podporni način izmenjave podatkov in lahko razbremeni obstoječe komunikacijske povezave predvsem v času prometnih konic in zmanjšuje potrebo po dodatnih žičnih instalacijah v inteligentnem okolju, saj nekatere naprave ne podpirajo nujno tudi WiFi povezav. Omrežje s komunikacijo preko električnega omrežja se izvede lahko z uporabo omrežnih pretvornikov (npr. Home Plug PLI-2030), s katerimi tvori hibridno žično lokalno omrežje (LAN + električne povezave). Več informacij o žičnih omrežjih preko električnega omrežja se nahaja v diplomskem delu [27].

4 ZASNOVA NARAVNEGA VMESNIKA ČLOVEK-STROJ (Human-Machine Interface – HMI)

Naravni vmesnik med človekom in strojem običajno implementiramo z uporabo adaptivnih uporabniških nivojev, ki predstavljajo funkcionalno implementacijo multimodalne interakcije[46]. Multimodalni nivo se lahko implementira na številnih različnih aplikativnih scenarijih. Naslednja generacija uporabniških vmesnikov teži k razvoju takšnih inteligentnih sistemov, ki bodo sposobni posnemati oz. izvajati takšne dialoga, ki so značilni za ljudi, in tako omogočiti naravno interakcijo z raznimi napravami v inteligentnih okoljih. V primerjavi s tradicionalnimi uporabniškimi vmesniki (zaslon na dotik, uporaba miške in tipkovnice, tablični računalniki, itn.), bodo napredni inteligentni vmesniki zmožni inovativnega generiranja tako verbalnih kot tudi neverbalnih komunikacijskih modalnosti, kar pa bo zahtevalo tudi razvoj kompleksnejših sistemskih arhitektur. Implementacija takšnih inovativnih vmesnikov mora slediti naslednjim ciljem:

- Naravno interakcijsko obnašanje: časovna uporabniško prilagojena interakcija,
- fleksibilnost: procesiranje velikega števila modalnosti in aplikacij,
- odprtost: dovoljevanje različnih pristopov procesiranja,
- upravljivost: podpora distribuiranemu razvoju.

V nadaljevanju bomo na osnovi projekta SmartKom [28][29] predstavili tiste pristope, cilje in izvedbe storitev, ki jih lahko ponudijo takšni napredni HMI vmesniki. Študija projekta SmartKom predstavlja tudi eno od izhodišč pri razvoju kognitivnega sistema »DATA«, ki ga predstavljamo v diplomski nalogi.

4.1 SmartKom

Arhitektura inteligentnega okolja, ki je nastala v okviru projekta SmartKom, predstavlja takšno zasnovo, ki omogoča izvedbo praktičnega programskega ogrodja za uspešno realizacijo naprednih aplikacij multimodalnega dialoga. Multimodalni dialog za sistem SmartKom, ne vključuje le integracijo osnovnih modalnosti in njihove sinhroniziranosti, marveč obravnava celoten spekter fenomenov v pogovoru, ki jih lahko zasledimo pri multimodalni interakciji. Glede na cilje razvoja, ki so usmerjeni v

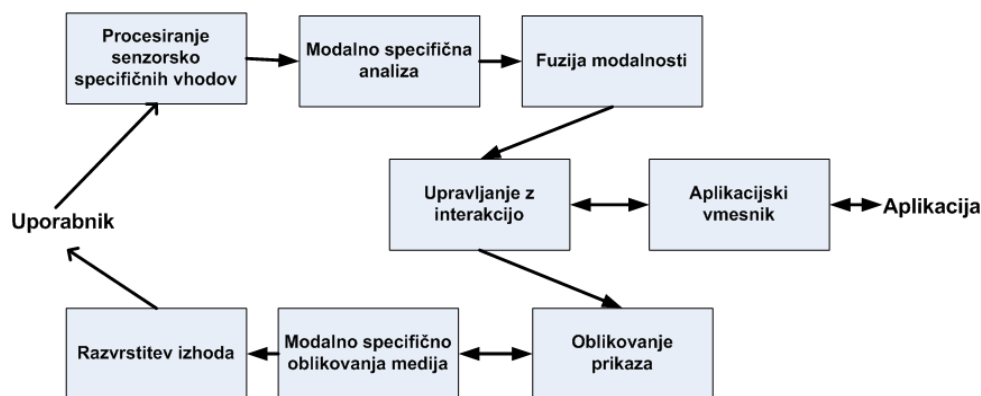
možnost uporabe kjerkoli, je SmartKom zasnoval takšen sistem, ki ga lahko vključujemo v različne tipe nalog z različnimi uporabniškimi scenariji.

4.1.1 Arhitekturne zahteve sistema za multimodalno interakcijo

Konceptualne arhitekture, kot so opisane v [30][31][32][33][34][35], predstavljajo sistemsko arhitekturo, ki povezuje arhitekture tradicionalnih vmesnikov (zasloni na dotik, zasloni za prikaz, tipkovnica in/ali miška, daljinski upravljalniki, itn.) in inteligentnih uporabniških vmesnikov. Znotraj klasičnega Seeheimovega referenčnega modela [36], se notranja struktura komponent uporabniškega vmesnika, ki deluje kot posrednik med končnim uporabnikom in podpornim aplikacijskim sistemom, deli na naslednje tri enote:

- Prezentacijski nivo ali upravljanje prikaza, določa obliko in nizko nivojsko obnašanje uporabniškega vmesnika,
- krmiljenje dialoga je sestavljeno iz vmesne faze procesiranja, ki upravlja tok interakcije in interakcijo zahtev,
- aplikacijski vmesnik predstavlja tretjo logično komponento, ki omogoča semantični model aplikacije, ki ga potrebujemo za dostopanje do funkcionalnega jedra sistema.

Slika 4-1 predstavlja strukturo višje nivojske arhitekture multimodalne interakcije.



Slika 4-1: Visoko nivojska arhitektura multimodalne interakcije.

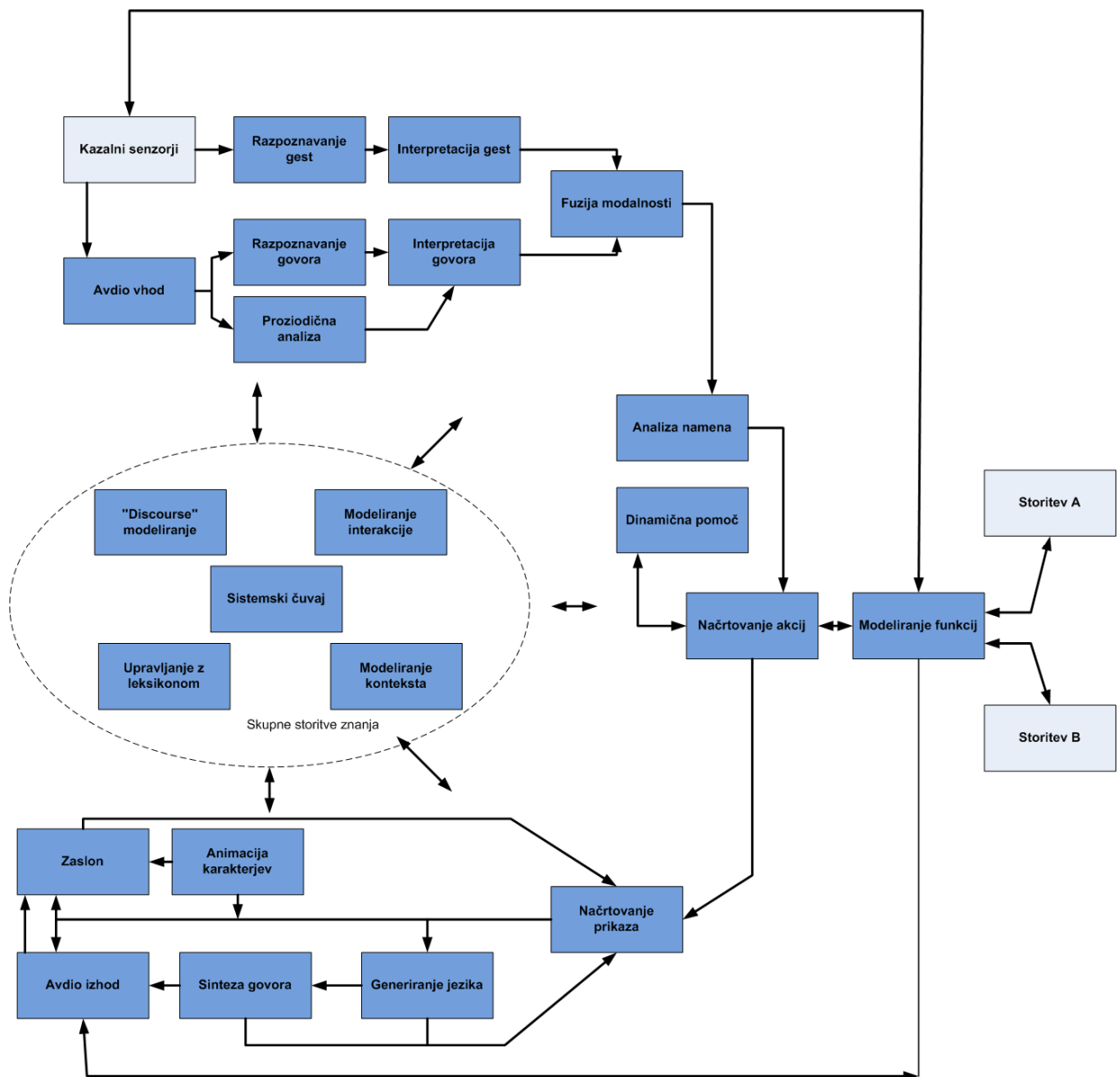
Nivo procesiranja specifične vhodne informacije senzorjev, združuje vse tehnološke vhodne naprave, kot tudi mikrofone in module za razpoznavanje gest. Ponuja standardizirano strojno neodvisno procesiranje na nivoju signalov ali simbolov. *Nivo*

specifične modalne analize je sestavljen iz razpoznavalnikov, ki transformirajo signale s senzorjev in jih pretvorijo v simbolično informacijo, ter analizatorjev, ki znajo te signale tudi opisati. *Fuzija modalnosti* združi pomen zajetih modalnosti v eno koherentno predstavitev. *Modul za opravljanje interakcije* identificira namene uporabnika, določa naslednje korake, ki naj jih sistem opravi in kliče razne aplikacijske funkcije. *Aplikacijski vmesnik* definira abstraktne vmesnike za interakcijo z aplikacijami v ozadju. Ta nivo skrije tehnične podrobnosti aplikacij in tako zagotovi generični način upravljanja z aplikacijami. *Nivo oblikovanja predstavitev*, glede na namene sistema informacijo transformira v generiranje koordiniranega multimodalnega izhoda. Funkcionalni blok sestavlja prva faza fizije modalnosti, ki se ukvarja s splošno organiziranostjo predvidene multimodalne interakcije. V splošnem organiziranost interakcije vključuje razne pod naloge, kot so: izbira vsebine, a-lokacija medijskih vsebin in modalnosti, oblikovanje videza in koordinacija. *Oblikovanje specifičnega modalnega medija*, se opravi v sodelovanju z *Nivojem oblikovanja predstavitev*. Procesiranje na tem nivoju se ukvarja predvsem s konverzijo strukture abstraktnih vsebin v medijske objekte, ki so primerne za predstavitev. *Razvrstitev izhoda* predstavlja zadnji funkcionalni blok višje nivojske arhitekture. Blok združuje tehnične metode za izvajanje predstavitve koordinirane systemske reakcije po definiranih medijskih kanalih (zasloni, zvočniki, itn.). *Podatkovni viri znanja*, ki niso predstavljeni v zgornjem diagramu, predstavljajo enega izmed vitalnih delov sistema in lahko zahtevajo združen dostop različnih komponent. Pomembni viri znanja za inteligentno multimodalno interakcijo vključujejo eksplicitne modele uporabnikov, domeno, kontekst, naloge, različne medije ter modalnosti. Predstavljen arhitekturni diagram prikazuje glavni tok informacije po linearni strukturi, ki sestavlja osrednji cevovod procesiranja. Arhitektura ne omejuje implementacije dodatnih medsebojnih povezav komponent, saj lahko posamezne komponente obravnavamo tudi kot komunikacijske procese in ne le faze v samem cevovodu procesiranja.

4.1.2 Prilagodljiv sistem za multimodalni dialog

Sistem SmartKom je zasnovan na osnovi t.i. situacijske, k uporabnikom usmerjene dialoške paradigme [37] in vključuje antropomorfičen in emocijsko odziven uporabniški vmesnik. Interakcijska metafora je osnovana na ideji, da uporabnik delegira nalogo

virtualnemu komunikacijskemu agentu, ki se v obliki pogovornega avatarja vizualizira kot živ karakter. Vmesniški agent razpoznava namene in cilje uporabnika, v primeru potrebe po kolaboraciji tako na primer vpraša uporabnika za povratno informacijo, v vlogi uporabnika lahko dalje dostopa do različnih aplikativnih storitev in prikazuje rezultate v ustreznih oblikah. Na implementacijskem nivoju je sistem usmerjen predvsem v okolja, kot so pametne sobe, kioski ali mobilne aplikacije.



Slika 4-2: Funkcionalna shema arhitekture sistema SmartKom .

Zanimiva lastnost sistema multimodalnega dialoga SmartKom, predstavlja možnost integracije različnih storitev (modularen samostojen razvoj različnih aplikacijskih funkcij v koherenten sistem z dodano vrednostjo). Dodatno sistem implementira sposobnost samonadzora, ki neprestano preverja stanje posameznih komponent in tako določa generalno procesorsko zmogljivost sistema. Takšen pristop meta-nivojskega razmišljanja in nadzora, se lahko uporabi pri generiranju ustrezne povratne informacije sistema, ki se nanaša na sistemsko aktivnost in potencialne napake v delovanju sistema. Na sliki 4-2 so prikazani osnovni gradniki sistema z multimodalno interakcijo. Vsaka komponenta sistema predstavlja enega izmed procesorskih enot izvajalnega sistema. Osvetljene komponente se uporabljajo v vseh primerih aplikacij. Ker je sistem SmartKom implementiran porazdeljeno, predstavlja vsaka komponenta neodvisen proces. Podporna arhitektura je zasnovana na komunikacijskih procesih, ki delujejo na način: "objavi - prijavi se". Ta pristop podpira dogodkovno usmerjeno interakcijo z uporabo usmerjenih komunikacijskih povezav med pošiljateljem sporočila, ki predstavlja generator podatkov in nabora prejemnikov, ki so porabniki podatkov.

Osnovni sistem podpira multimodalni vhod v obliki govora in gest. Poleg tega integrira dodatne aplikacijsko specifične storitve. Tip senzorjev namenjen procesiranju gest, je odvisen od tehnične priprave aplikacijskega scenarija. Ločevanje procesiranja senzorsko specifičnih vhodov v ločene komponente, ki na sistem povezujejo določeno napravo in enkapsulirajo dostop do podporne strojne opreme, predstavlja prvi dekompozicijski pristop arhitekture sistema. Za doseganje maksimalne fleksibilnosti sistema se sledi istemu pristopu tudi na izhodnih napravah. Analiza specifičnega modalno vhoda se lahko razdeli v ločene komponente in tako omogoči nadalje razločevanje med modalno specifičnim razpoznavanjem (npr.: procesiranje senzorskih podatkov na leksikalnem in semantičnem nivoju; posledična semantična interpretacija, narejena na osnovi simbolične interpretacije). Interakcija pri upravljanju sistema SmartKom, temelji na združevanju in uporabi več komponent. Glavna naloga analize je, izbira najbolj verjetne interpretacije vhoda glede na dani nabor hipotez. Načrtovanje akcij predstavlja osnovo krmiljenja dialogov in je podprto s nadomestno komponento zagotavljanja pomoči, ki se aktivira v primeru težav, ki lahko nastopijo med samo interakcijo, ali pa se aktivira v primeru potrebe po dodatni pomoči. Glavne faze

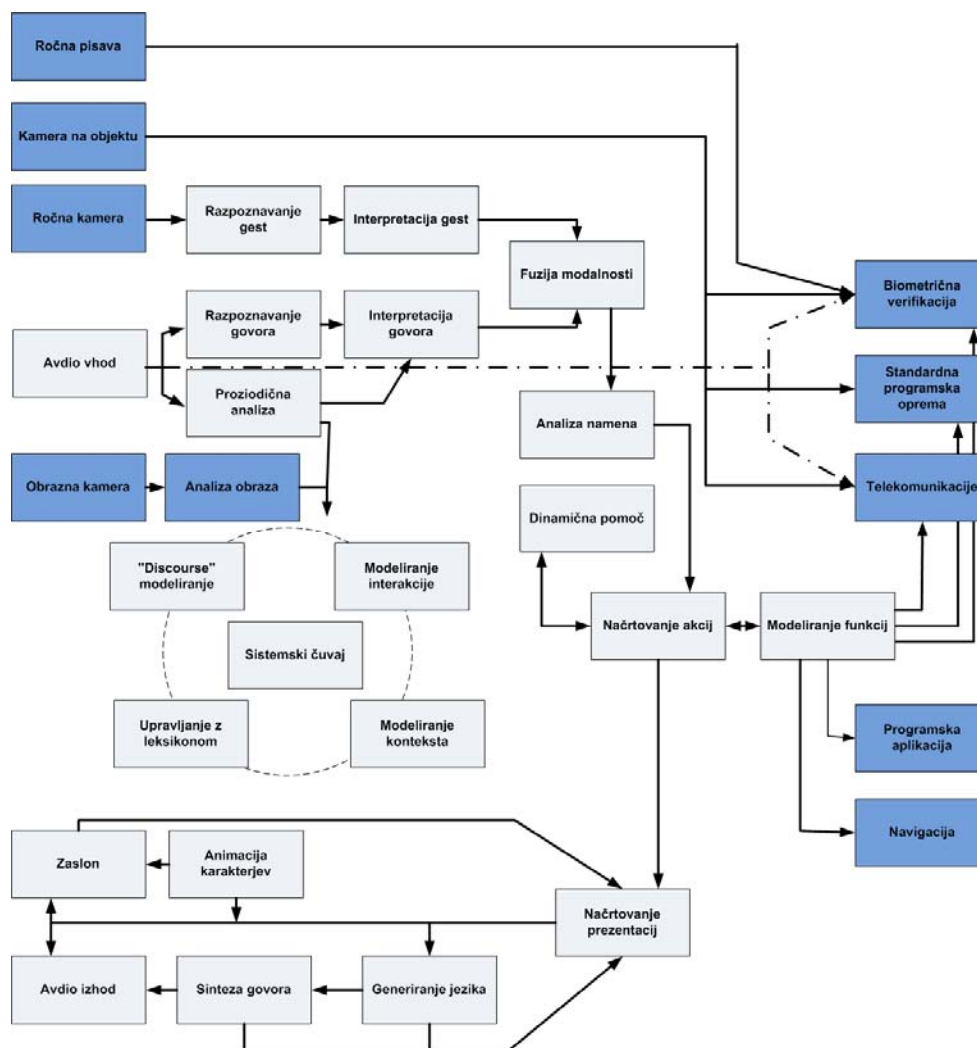
procesiranja so podprte z uporabo različnih komponent, ki aktivno vzdržujejo vire znanja v skupni rabi. Viri znanja so predstavljeni v spodnji tabelah 4-1 in 4-2.

Vir znanja	Uporaba/Vzdrževanje
Multimodalni diskurz (discourse) model	Izveden je za namene sematične in pragmatične interpretacije v času procesiranja vhodov in izhodov. Dinamično se posodablja med generiranjem izhoda in vključuje kontekstno »razmišljanje« in ocenjevanje.
Kontekstna informacija	Uporablja se za procesiranje situacijskih parametrov kot sta: trenuten prostor in čas, ki ga podaja kontekstni model.
Modeliranje interakcije	Obravnava različne aspekte, kot so modalnosti in uporabniške preference specifičnih oblik komunikacije in emocijsko stanje uporabnika. Interakcijski model omogoča dinamično prilagoditev komunikacijskega obnašanja sistema.
Leksikon	Predstavlja dinamični podatkovni vir, ki se osvežuje z novimi vnosi, odvisnimi od dinamičnosti aplikacijskih podatkov, ki jih sprejema od zunanjih informacijskih storitev. Osveževanje leksikona se posreduje vsem komponentam, ki procesirajo naravni jezikovni vhod in izhod.

Tabela 4-1: Vir znanja sistema SmartKom.

Vir znanja	Uporaba/Vzdrževanje
Sistemski paznik	Nadzira stanje procesiranja vseh individualnih komponent in ponuja realno informacijo o trenutnem stanju sistema. Takšna informacija se uporablja za takojšnjo povratno informacijo ali inicializacijo pomožnih reakcij v primeru težav pri procesiranju.

Tabela 4-2: Vir znanja sistema SmartKom (nadaljevanje).



Slika 4-3: Modularna arhitektura informacijskega kioska SmartKom-Public.

Komponenta funkcionalnega modeliranja, realizira aplikacijski vmesnik in izvaja dodaten nadzor nad vsemi vhodnimi in izhodnimi napravami, ter koordinira dostop do naprav odvisno od eksplicitnega stanjskega modela. Aplikacijski nivo je modulariziran v obliki množice storitvenih komponent, izmed katerih vsaka realizira specifično aplikacijsko funkcionalnost. Preostali elementi iz nabora komponent, upravljajo s fizijo modalnosti. Druga komponenta načrtovanja je odgovorna za oblikovanje prikaza. Del oblikovanja modalno specifičnega izhoda, opravi komponenta animacije karakterjev. Le-ta realizira izhod v obliki pogovornega avatarja.

4.1.3 Primeri aplikativne uporabe generične arhitekture

Arhitekturno ogrodje SmartKom je oblikovano tako, da lahko podpira širok nabor kolaborativnih ali multimodalnih dialogov, ki uporabniku omogočajo intuitiven in učinkovit dostop do raznih funkcionalnosti, ki so jim na voljo za opravljanje različnih nalog. Implementacija sistema omogoča tri različne tipe implementacije scenarijev, ki zaradi specifičnih karakteristik, tudi definirajo samo arhitekturo sistema.

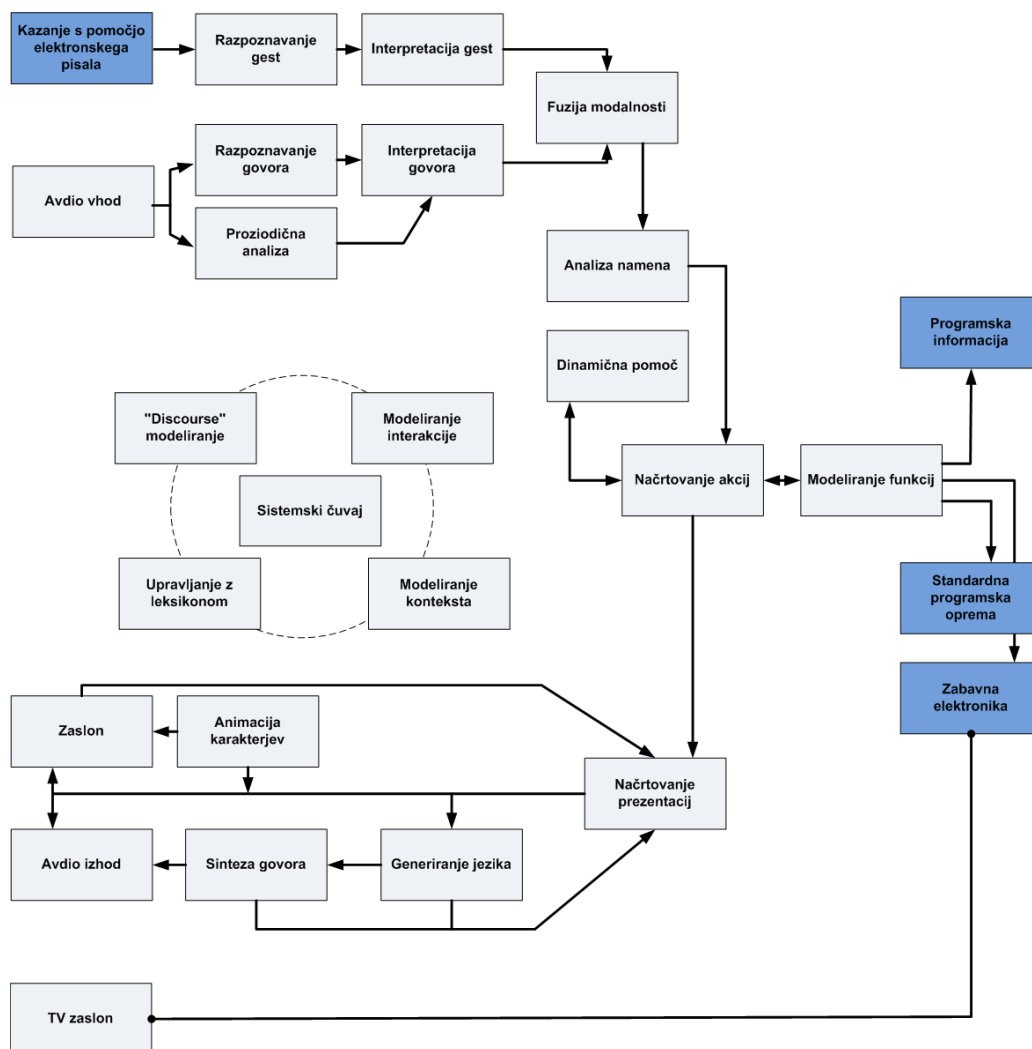
Informacijski in komunikacijski kiosk

Kiosk je realiziran kot napredna multimodalna komunikacijska postaja, ki je namenjena železniškim postajam, letališčem in drugim javnim prostorom. Kiosk je namenjen lahko raznim posredovanjem informacije, ki se nanašajo npr. na predvajanje filmov, izvajanja raznih rezervacij itd. Z njim lahko izvedemo tudi napredno personalizirano komunikacijsko storitev z uporabo telefona, faksa ali elektronske pošte. Arhitektura sistema je predstavljena na sliki 4-3. Komponente ki so specifične za dan scenarij so v diagramu osvetljene.

Multimodalni vhod za geste, je realiziran na infrardečem sistemu kamer, ki med premikanjem roke po grafični projekciji vmesnika sledijo roki uporabnika. Takšen senzor ne potrebuje fizičnega kontakta roke in horizontalne projekcije, kar omogoča bolj naraven način uporabe gest. Kot dodatno vhodno modalnost sistem SmartKom implementira analizo izrazov na obrazu. V kombinaciji z akustičnimi indikatorji, pridobljenimi s pomočjo prozodične analize, se razpoznavanje obrazne mimike uporablja za določanje trenutnega stanja uporabnika. Informacijski kiosk je tako opremljen z dvema aplikacijsko specifičnima vhodnima komponentama za sisteme, ki

podpirajo uporabo elektronskega pisala, ter dodatnim sistemom kamer, ki zajemajo digitalno sliko objektov postavljenih na projekcijskem panelu. V danem scenariju, so možne naslednje funkcije ob uporabi glavnih storitvenih komponent:

- Biometrična avtentikacija z uporabo konture zvoka, podpisa ali roke,
- dostop do standardnih aplikacij, kot so imenik in prenos e-mail sporočil,
- telefonske povezave in prenos faks sporočil,
- storitev pregleda sporeda za kino in rezervacije predstave.
- pregledovanje zemljevidov, ki vsebujejo označene lokacije kinematografov in označene poti do zaželene lokacije.



Slika 4-4: Modularna arhitektura sistema SmartKom – Home.

Informacijski asistent za dnevne sobe SmartKom-Home

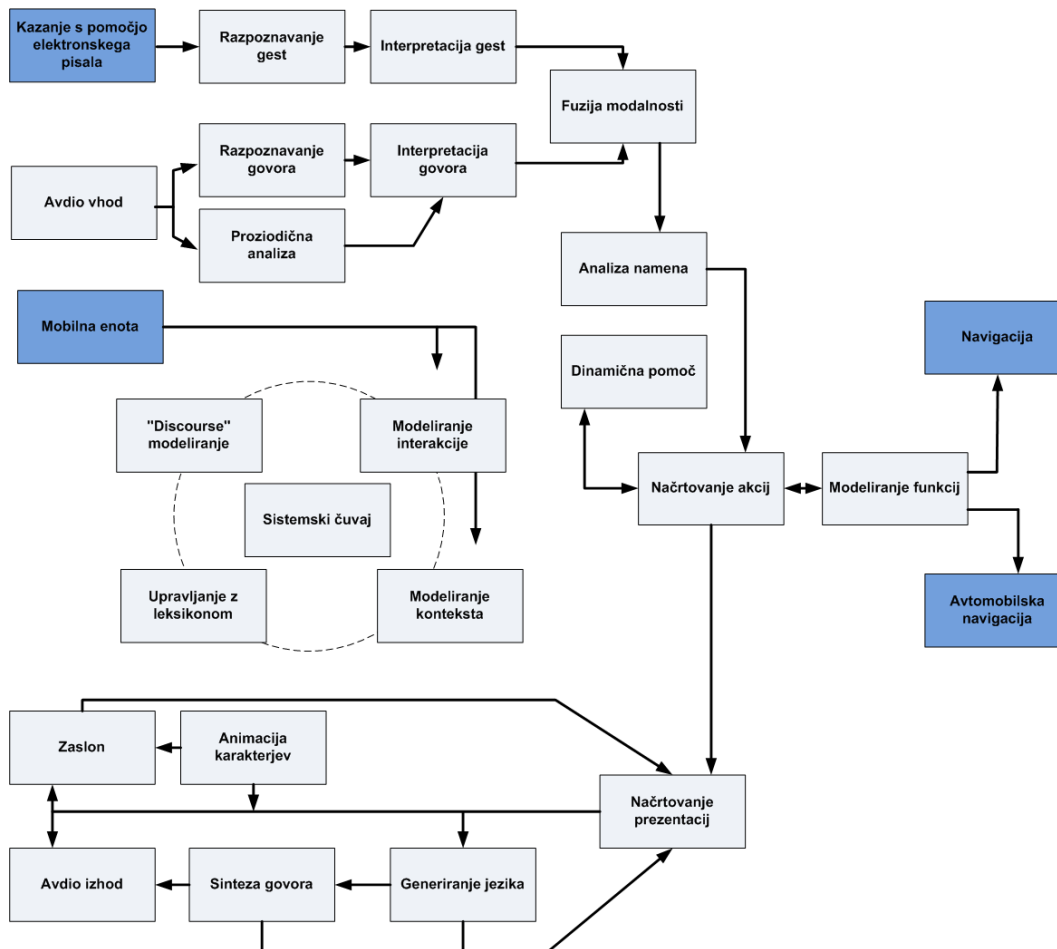
V osnovni konfiguraciji predstavlja SmartKom-Home multimodalni portal za razne storitve doma. Z uporabo prenosnih tabličnih računalnikov, lahko uporabnik uporablja sistem kot elektronski programski vodič ali za enostavno upravljanje z napravami zabavne elektronike, kot so televizor, digitalni HiFi radio, digitalni DVD predvajalnik itn. Modularna arhitektura sistema SmartKom – Home je predstavljena na Sliki 4-4.

Sistem podpira osnovne modalnosti, ki so povezane z multimodalnim sistemom. Uporabniki lahko krmilijo tudi z narisanimi gestami (elektronsko pisalo). V kontekstu scenarijev za dom, podpira sistem trenutno dva različna pristopa interakcije. V t.i. načinu "usmerjen naprej" (lean-forward mode), se kot multimodalni vhod lahko uporabljata koordinirano govor in geste, ki tako skupaj oblikujeta multimodalno interakcijo s sistemom. Alternativni t.i. način "usmerjen nazaj" (lean-backward mode), pa omogoča krmiljenje brez uporabe prenosnih tabličnih računalnikov. Za upravljanje s sistemom se v takšnem načinu lahko uporablja samo verbalna komunikacija. Zaradi ne razpoložljivosti grafičnega vmesnika v tem načinu delovanja, sistem dinamično prireja multimodalni izhod, v okviru katerega uporabi bolj naravno generiranje govora. Sistem sicer uporablja dve enaki storitveni komponenti, kot sistem SmartKom – Public, vendar pa podaja različne aplikativne metode teh storitev. Podatkovna baza programa se primarno uporablja za pridobivanje informacije o televizijskih oddajah, orodje za delo s koledarjem pa je implementirano tako, da omogoča nastavitev časa snemanja posameznih oddaj. Tretja storitvena komponenta ponuja uporabnikom visoko nivojski vmesnik za upravljanje z različnimi napravami. Npr.: storitev zabavne elektronike omogoča oddaljeno krmiljenje televizije, videorekorderja in satelitskega sprejemnika. Sistem tako ponuja fleksibilno in odprto integracijo različnih aplikacij. Storitve tvorijo povezovalne komponente, ki implementirajo dobro definirano povezavo do aplikacijsko specifičnih funkcionalnosti in pogosto enkapsulirajo celoten in kompleksen ter aplikacijsko specifičen podsistem. Tipičen primer takšne integracije predstavlja integracija komercialne rešitve za nakup produktov preko televizije.

Mobilni asistent SmartKom – Mobile

Sistem SmartKom – Mobile je mobilni asistent, ki ga lahko uporabljamo na primer med hojo ali v avtomobilu. Scenarij združuje tipične storitve kot so: planiranje

potovanja in usmerjanja po izbrani poti na podlagi GPS. Zunaj avtomobila uporabnik upravlja s sistemom preko ročnega računalnika, znotraj posebej pripravljene avtomobila pa se posebej vgrajena oprema uporablja za zajemanje in procesiranje multimodalnega vhoda in izhoda. Komponente sistema SmartKom – Mobile, so predstavljene na sliki 4-5. Multimodalna interakcija se opravlja preko elektronskih pisal na manualnih napravah. Zaslone v avtomobilu pa ni občutljiv na dotik.



Slika 4-5: Komponentna arhitektura sistema SmartKom – Mobile.

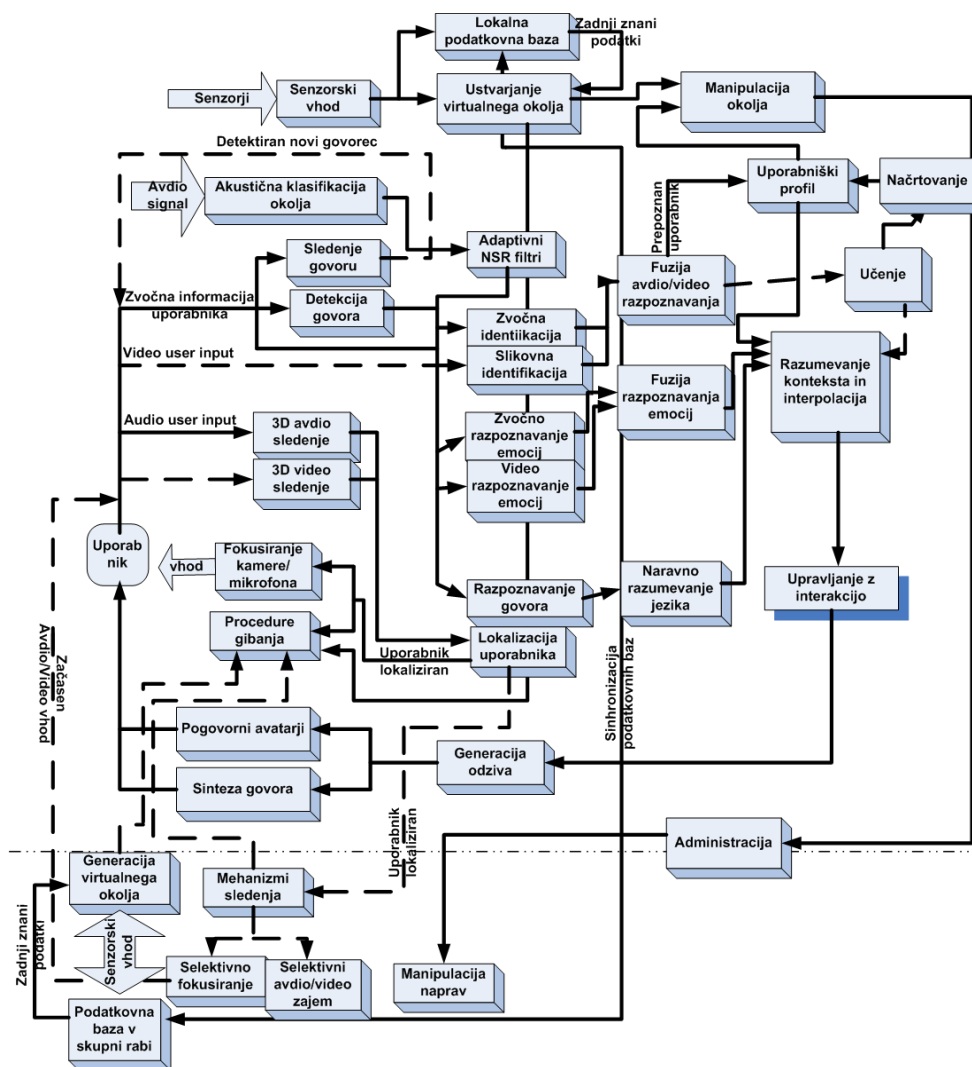
Komponenta z imenom "mobilna naprava", ponuja generično rešitev za daljinsko upravljanje različnih naprav. V izogiba časovno potratnemu ponovnemu zagonu in izklopu sistema med delovanjem, se avdio, video in druge komponente za prikazovanje izvedejo z razpošiljanjem informacij v obliki RTP tokov. Dvosmerna RTP povezava omogoča oddaljen dostop do komponente za prikazovanje. Po pritisku na gumb za aktivacijo na krmilni napravi ali v avtomobilu, komponenta "mobilna naprava"

posreduje karakteristike nove čelne (front-end) naprave. Velika razlika med zaslonom v avtomobilu in tistim na krmilni napravi, prisili sistem v avtomatsko prilagajanje vizualne predstavitve glede na tip naprave, ki se uporablja za prikaz. Navigacijska komponenta v vozilu lahko vključuje običajen komercialni sistem navigacije, ki generira zemljevid z označenimi voznimi potmi. Dodatno lahko zemljevid omogoča tudi označevanje parkirnih mest. Manualna navigacijska naprava lahko tudi omogoča generiranje zemljevidov, na katerih so označene turistične zanimivosti, načrtovanje potovanja in vodenje oz. usmerjanje po izbrani poti. Takšna komponenta enkapsulira kompleksni podsistem, ki je implementiran z uporabo ogrodja, ki ponuja storitve več agentov. Glede na specifiko in zahteve modernih multimodalnih HMI vmesnikov, smo v okviru diplomske naloge predvideli zasnovo kognitivnega sistema, ki ga predstavljamo v naslednjem poglavju.

4.2 Funkcionalna zasnova kognitivnega sistema »DATA« za inteligentna okolja

V okviru diplomske naloge smo zasnovali lastno rešitev kognitivnega sistema z imenom »DATA«. Slika 4-6 tako predstavlja predlagano zasnovo sistema na funkcionalnem nivoju[47]. Predstavljena funkcionalna shema predstavlja vizualizacijo procesov kognitivnega sistema za napredna inteligentna okolja, ki tudi vključujejo multimodalno procesiranje na vhodu in izhodu sistema. Vir vhodne informacije je vedno uporabnik ali okolje, izhodna informacija pa se poda kot odziv generiran v obliki akcije ali avdio/video signala. V primeru detekcije dogodka, ki ga generira uporabnik, se najprej izvedeta proces 3D avdio in 3D video sledenja[54], ki kot vhodni parameter metode lokalizacije uporabnika/dogodka, pravilno usmerita video kamere in mikrofone na področju, kjer se je dogodek zgodil[48]. Dodatno se izvede še akustično zajemanje in akustična klasifikacija okolja. Akustična klasifikacija okolja se določa s pomočjo modelov okolja, ki vsebuje tudi informacijo o akustičnem stanju okolja, ki jih sistem tvori/posodablja tekom delovanja. Modeli okolja poleg akustične informacije vsebujejo še informacijo o dinamiki predmetov v okolju. Ko detektor govora zazna govor v okolici, se sproži sledenje govoru v obliki pravilnega usmerjanja mikrofonov in se prične razpoznavanje govora. V adaptivno filtriranje zvočnega signala vključimo še parametre akustične klasifikacije in tako izboljšamo zvočni signal, ki ga posredujemo razpoznavalniku tekočega govora. Odvisno od operacije, ki jo kognitivni sistem

opravlja, se s pomočjo filtriranega avdio/video signala prožijo procesi razpoznavanja obraza in razpoznavanja govorca, ločenega avdio in video razpoznavanja čustev ali pa razpoznavanja tekočega govora in mimike. Ne glede na operacijo se izhodi procesov avdio/video rezultatov vedno primerno združujejo v obliki fuzije. Pri razpoznavanju osebe je lahko rezultat fuzije kar uporabnik, pri razpoznavanju čustev pa opis razpoznanega čustva. Na podlagi visoko nivojske abstrakcije lahko sistem sklepa o kontekstu informacije in akcijah potrebnih za doseg želenega obnašanja. S pomočjo učenja in izsledkov učenja, lahko sistem dodatno opredeli primernost odziva glede na predhodne odzive v podobnih situacijah.



Slika 4-6: Funkcionalna zasnova kognitivnega sistema za inteligentna okolja.

Funkcionalna zasnova predstavljena na sliki 4-6, omogoča izpolnjevanje zahtev inteligentnih okolij, o katerih smo govorili v poglavju 2. Kot multimodalni vhod imamo lahko naslednje tipe multimodalne informacije:

- Govor uporabnika in akustična informacija prisotna v govoru,
- Slika uporabnika, z zajeto obrazno mimiko,
- Manualne geste (neverbalna komunikacijska sredstva),
- Akustična in telemetrična informacija o okolju.

Sistem zajeto akustično informacijo iz okolja procesira v kontekstu akustičnega klasificiranja okolja[55][56]. Ta proces v okviru interakcije pripomore k boljšemu razumevanju sistema (boljše razpoznavanje avdio/video ukazov), saj se na podlagi akustičnih značilk lahko določeni avdio/video šumi iz vhodne informacije odstranijo, mogoča pa je tudi razpoznavna emocionalnega stanja uporabnikov. Zvočna in slikovna informacija o uporabniku, se koristi na večih nivojih kognitivnega sistema »DATA«. Na primarnem nivoju, se ta vir multimodalne informacije, uporablja za identifikacijo uporabnika. Za ta namen se iz zajete informacije izloči značilke, s katerimi nato želimo enolično določiti uporabnika (velikost in oblika obraza, postavitev oči uporabnika, višina glasu uporabnika, način govora uporabnika itn.)[52][53]. Sekundarni nivo avdio/vizualnega procesiranja, se koristi za razpoznavanje vsebine v interakciji človek - stroj. Ob uporabi razpoznavanja tekočega govora in razpoznavanja gest, lahko sistem razpoznavna tudi vsebino interakcije[51][55]. Tretji nivo koristi avdio/video informacijo za razpoznavanje emocij.

V procesu uporabe rezultatov operacij procesiranja avdio/video informacije na višjih nivojih procesiranja, kognitivni sistem »DATA« združi rezultate posameznih monomodalnih razpoznavanj in oceni vrednost abstraktnega rezultata (npr. končni rezultat razpoznavanja emocij je emocijsko stanje »veselje«). Abstraktni rezultati fuzije multimodalne informacije služijo nadaljnjim proceduram umetne inteligence. Umetna inteligenca[50] predstavlja še zadnjo stopnjo abstrakcije informacije, kjer se na podlagi nižje nivojskih procesov in predhodnih izkušenj, sistem odloči o kontekstu dogodka. Kombinacija ukaza in emocionalnega stanja uporabnika, lahko npr. pove veliko o smiselnosti/nujnosti operacije, ki jo uporabnik zahteva. Glede na odziv uporabnika na

akcijo, ki jo je sistem izvedel, sistem na tej stopnji posodablja svoje znanje v obliki posodabljanja uporabniškega profila. Na podlagi kontekstnega razumevanja dogodka, se sistem v fazi načrtovanja odloči za akcije, ki jih je potrebno izvesti. Upravljaavec multimodalne interakcije, s pomočjo pridobljenih značilnosti interakcije iz nabora akcij, ki jih mora izvesti in konteksta operacije, določi tako tipe odzivov, kot njihovo obliko. Sistem za generiranje odzivov implementira s pomočjo TTS tehnologije, manipulacije okolja in/ali pogovornih avatarjev, karakteristična navodila upravljalnika multimodalne interakcije, kot multimodalni odziv sistema v interakciji z uporabnikom. V prikazano arhitekturo sistema je vključena tudi mobilna robotska enota. V kontekstu uporabniške interakcije, le-ta predstavlja platformo za naprednejše in čim bolj naravno obnašanje sistema. Mobilna robotska enota implementira podobne komunikacijske metode in manipulacije, kot samo inteligentno okolje. Metode so navadno omejene in dopuščajo le določeno stopnjo procesiranja. Za višje stopnje procesiranja, mobilna robotska enota uporabi možnost kolaboracije in porazdeljenega procesiranja na močnejšem strežniškem sistemu, ki predstavlja osrednji "center" inteligentnega okolja. V nadaljevanju bomo podrobneje predstavili predlagani razvoj kognitivnega sistema, ki se ukvarja s kontekstnim razumevanjem dogodkov v inteligentnem okolju in učinkovitem in fleksibilnem upravljanju z napravami v okolju, tudi v primeru večih uporabnikov in velikih kompleksnostih takšnih okolij.

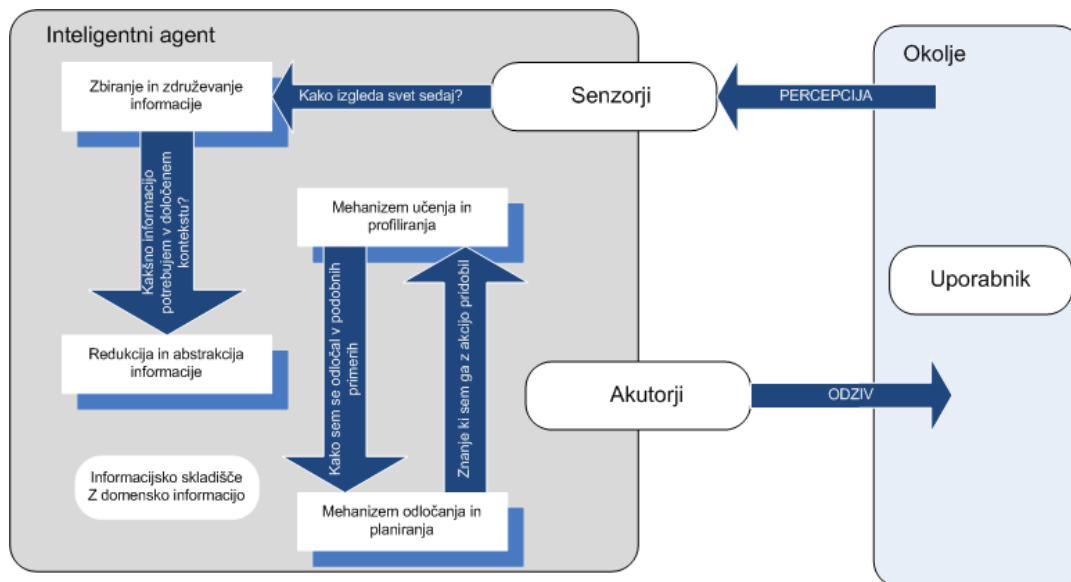
5 UMETNA INTELIGENCA (Artificial Intelligence - AI)

V inteligentnih okoljih pristopi umetne inteligence omogočajo kognitivno razumevanje okolja in uporabnikov, ter njihovega obnašanja v okolju. Glavne objektivne umetne inteligence v inteligentnih okoljih so predstavljene v spodnji tabeli:

Objektiva	Opis
<i>Zbiranje in združevanje informacije</i>	Uporaba distribuiranih senzorskih polj v inteligentnem okolju in na mobilni robotski enoti (zajemanje in združevanje multimodalne informacije – avdio, video, termično sevanje objektov v IR spektru, itn.).
<i>Redukcija in abstrakcija informacije</i>	Prilagoditev zbrane in združene informacije v tipske domene primerne za uporabo v procesih računalniškega modeliranja in učenja.
<i>Profiliranje uporabnikov</i>	Uporaba procesov učenja in modelov razpoznavanja vzorcev za implementacijo modeliranja profilov uporabnikov in njihovih navad v digitaliziranem okolju.
<i>Učinkoviti mehanizmi učenja</i>	Razvoj algoritmov, ki bodo učinkovito procesirali informacijo prejšnjih objektov in omogočili implementacijo mehanizmov intuitivnega in kognitivnega razumevanja uporabnikov.
<i>Inteligentna kolaboracija sistemov</i>	Učinkovito opredeliti funkcije posameznih sistemov v inteligentnem okolju (okolje in mobilna enota) in opredeliti ter razviti mehanizme za izmenjavo potrebnih podatkov, ki omogoča napredno skupinsko (porazdeljeno) reševanje problemov.

Tabela 5-1: Objektivne umetne inteligence.

Umetna inteligenca [56] inteligentnih okoljih je posredovana s pomočjo inteligentnih agentov. Implementacije agentov so odvisne predvsem od namembnosti okolja (domače, poslovno, konferenčno okolje,...), saj namembnost okolja določa tipe informacij (ki jih je potrebno zajeti), modele učenja (ki jih je potrebno postaviti), in tipe odzivov, ki jih mora okolje generirati (definiirajo tako tipe vhodne informacije kot modele učenja). Inteligentni agent v okviru AI predstavlja entiteto, ki lahko opazuje in usmerja delovanje inteligentnega okolja v smeri opravljanja nekega zadanega cilja. Inteligentni agenti pri opravljanju nalog za večjo učinkovitost uporabljajo metode učenja in uporabe znanja na višjih nivojih.



Slika 5-1: Funkcionalna shema inteligentnega agenta.

Slika 5-1 predstavlja osnovno idejo delovanja inteligentnega agenta. S senzori zajemamo podatke, ki nam podajajo informacijo o trenutnem stanju okolja ali opazovanega objekta. S procesom redukcije in abstrakcije vso zajeto informacijo najprej zmanjšamo v nabor, ki zajema le tisto informacijo, ki je bistvena za določeno nalogo npr. razpoznavanje osebe – iz nabora zajete informacije izločimo le avdio in video informacijo, ostala informacija okolja pa je za to nalogo nepomembna in predstavlja balast. Primerno reducirano informacijo nato abstrahiramo, procesiramo in pretvorimo v primerno domensko obliko in jo shranimo v podatkovno skladišče. Informacija je profilno in domensko razvrščena. S pomočjo mehanizmov učenja in odločanja lahko na podlagi trenutne abstrahirane informacije in predhodnega znanja generiramo primeren odziv.

Zajemanje podatkov in združevanje (fuzija)

Podatki se zajemajo iz večih tipov senzorjev postavljenih v inteligentno okolje ali na mobilno robotsko enoto, ki deluje v okviru okolja. Zbrana multimodalna informacija (avdio, video, laser, ultrazvočna, detekcija gibanja, lociranje, itn.) se na primeren način združi v domensko tipizirano informacijo, ki vsebuje samo nujno informacijo za implementacijo akcije v okviru domene. Za proces združevanja je najprimernejša uporaba pristopov na podlagi Kalmanove filtracije. Fuzija stanskih vektorjev uporablja skupino Kalmanovih filtrov[57], ki pridobijo podatke o stanju posameznih senzorjev.

Stanja posameznih senzorjev se nato združijo v skupno informacijo o njihovem stanju. Metode principa fuzije meritev direktno združujejo meritve posameznih senzorjev in pridobijo uteženo ali argumentirano meritev. Šele nato za pridobitev predvidevanja o končnem stanju nad združeno informacijo uporabijo en Kalmanov filter. V osnovi metode fuzije meritev ponujajo boljše ocene, fuzija stanskih vektorjev pa zahteva manjše procesorske zmogljivosti in večjo toleranco do napak. Fuzija stanskih vektorjev je uporabna le, kadar so Kalmanovi filtri konsistentni. V realnih aplikacijah (npr.: sledenje in navigacija), so nižji procesi pogosto nelinearni, Kalmanovi filtri pa so takrat zasnovani na lineariziranih procesnih modelih (Jacobian linearisation or neurofuzzy local linearisation) in so zato zaradi napak v modelih (posledica linearizacije) pogosto nekonsistentni. V realnih aplikacijah se zato največkrat uporablja fuzija meritev.

Redukcija in abstrakcija informacije

Namen redukcije in abstrakcije informacije je izločanje nezaželene informacije z namenom ohranjanja le potencialno zanimivih podatkov, odkrivanje skritih vzorcev in pomembnih lastnosti podatkov. Večina senzorjev je zasnovanih tako, da ponujajo točno določeno informacijo v eni domeni. Laserski senzor za določevanje razdalje bo lahko posredoval razdaljo merjeno s pomočjo laserskega žarka. Oblika informacije, ki jo senzor vrne pa direktno ni uporabna za proces razpoznavanja objektov. Procedure redukcije in abstrakcije podatkov bodo informacijo pridobljeno z laserskega senzorja v domeni razpoznavanja objekta predstavile kot daleč ali blizu, z uporabo fuzije večih senzorskih podatkov pa kot okroglo, kvadratno, itn. Redukcija in abstrakcija informacije vključuje naslednje procese:

- **Filtriranje informacije:** procesi, ki odstranijo nezaželeno informacijo (informacija ki je označena kot trivialna, šumna ali neuporabna za trenutno nalogo),
 - **Izbira lastnosti:** proces izbire, ki zagotovi da se potencialno uporabna informacija zadrži (informacija relevantna za trenutno nalogo; možnost je tudi, da ni nujno potrebna),
 - **Grupiranje informacije:** proces detekcije skritih vzorcev in pomembnih lastnosti v pridobljeni in prečiščeni informaciji.
-

Profiliranje in gradnja modelov

Namen profiliranja in gradnje modelov, je izdelava tipskega modela obnašanja in aktivnosti posameznega uporabnika na osnovi učenja. Najboljšo kombinacijo predstavlja uporaba tako nadzorovanega kot nenadzorovanega učenja. S takšno kombinacijo lahko definiramo vzorce normalnega in nenormalnega obnašanja. Najbolj primerna implementacija je implementacija z uporabo mehkih množic (Fuzzy-rough sets) in HMM modeliranja za izgradnjo modelov človeškega obnašanja. Profiliranje obnašanja pa se izvede s primerjavo trenutnih senzorskih podatkov in HMM modeli obnašanja. V primeru da se vhodna informacija ne sklada z naučenim HMM modelom, se obnašanje označi kot nenormalno, sistem pa lahko izvede akcije pomoči.

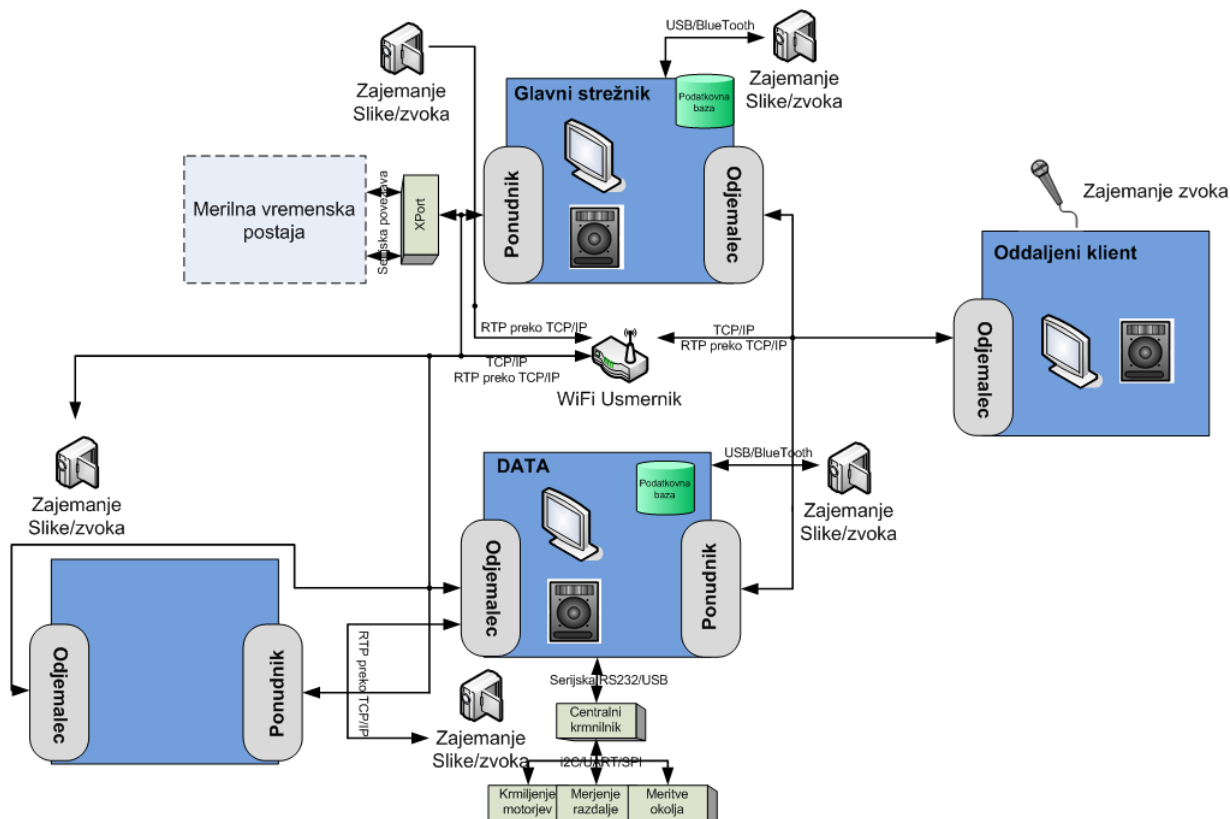
Mehanizmi učenja

Najprimernejši pristopi vključuje uporabo SVM (support vector-machines – metoda podpornih vektorjev), ki optimizirajo konstrukcijo modelov in proces posodabljanja glede na obstoječe modele in glede na novo pridobljen nabor podatkov. Naučeno obnašanje se pretvori v pravilo, ki se doda v bazo pravil. Baza pravil predstavlja nabor predhodnega znanja sistema.

Naslednji segment kontekstnega sistema, ki se nanaša na prenos avdio/vizualnih podatkov, predstavljamo v naslednjem poglavju. V njem podrobneje opišemo tudi programski modul, ki smo ga v okviru diplomskega dela razvili za namene zajemanja in prenosa takšnih informacij. V kognitivnem sistemu, takšni viri predstavljajo vhodno informacijo v module za avdio/video procesiranje. Predstavljen sistem omogoča tudi nizko-nivojsko procesiranja avdio/video signalov, npr. odstranjevanja šuma, pretvorbe slik v sivinsko paleto, detekcijo govora, itn.

6 ZAJEMANJE IN PRENOS AVDIO IN VIDEO INFORMACIJE

Avdio in video informacija v kognitivnem sistemu, predstavlja enega najpomembnejših informacij za multimodalni vhod kognitivnega sistema. Avdio/video informacija ne služi le za krmiljenje oz. upravljanje s sistemom (krmiljenje z gestami rok, govorno krmiljenje itd.), marveč lahko s pravilno manipulacijo in obravnavo, služi kot pomemben podatek pri pridobivanju kontekstne informacije z analizo nekega dogodka v okolju. Akustična informacija, skrita v govoru uporabnika in mimika obraza, ki jo uporabnik tvori med govorom, lahko na primer določita stopnjo nujnosti izvedbe določene operacije, ki jo uporabnik zahteva. S pomočjo fuzije avdio in video informacije za specifičnega uporabnika, lahko določamo psiho-fizično stanje uporabnika itn. Na sliki 6-1 predstavljamo realno implementacijo sistema DATA in njene osnovne gradnike. Na sliki smo posebej označili podporo zajemanju avdio in video informacije posameznega gradnika sistema DATA.



Slika 6-1: Avdio/vizualna podpora modulom v sistemu »DATA«.

Iz slike je razvidno, da sistem DATA ne predvideva enolične definicije avdio/video naprav za zajemanje. Sistem DATA definira tri tipe enot za zajemanje avdio/video informacije. Prvi tip predstavljajo naprave, ki so zmožne zajemati le avdio informacijo (mikrofoni), drugi tip predstavljajo naprave, ki so zmožne zajemati le video informacijo (določene USB kamere), tretji tip naprav pa predstavljajo hibridi, ki lahko zajemajo tako avdio kot tudi video informacijo. Na modulih sistema DATA, smo definirali naprave za zajemanje, predvsem glede na napajalne zmožnosti posameznih naprav in seveda tudi glede na predviden tip vhodne multimodalne informacije, na dani mobilni robotski enoti ali drugih modulih sistema DATA. Enote v sistemu DATA, ne uporabljajo le naprav za zajemanje, ki so direktno oz. lokalno priključene, marveč lahko uporabljajo tudi naprave, ki so dostopne preko mrežne povezave, čeprav upravlja z dano napravo za zajemanje enota, na kateri se le-ta nahaja. V kontekstu priključitve naprave za zajemanje, ločimo naprave z direktnim priklopom (USB kamere) in naprave z mrežnim priklopom (žične in brezžične IP kamere). V nadaljevanju bomo predstavili RTP protokol [58], ki ga uporabljamo v predlaganem sistemu DATA za prenos avdio/video informacije. Poleg tega predstavljamo tudi pripadajoči programski paket, ki smo ga razvili v okviru diplomskega dela in ki omogoča učinkovito zajemanje in prenos avdio/video informacije. Programski paket tudi predstavlja pomemben element pri izgradnji sistema, ki bo sposoben kognitivnega obravnavanja dogodkov v inteligentnih okoljih.

6.1 RTP protokol (Real-time Transport protocol)

Sistem DATA ne omejuje prenosa podatkov le v obliki ASCII znakov. Avdio in video informacija se v obliki RTP paketov prenaša med posameznimi gradniki sistema DATA. RTP protokol zagotavlja hiter in dovolj učinkovit prenos avdio in video informacije in zmanjšuje potrebo po uporabi balasta (ni dodatnega formiranja TCP/IP paketov, ki nosijo RTP informacijo). Z uporabo časovnega označevanja (time stamping) in 32 bitnega polja standardnega RTP paketa RTP protokol omogoča sinhronizacijo in računanje trepetanja (jitter).

RTP protokol se v sistemu DATA ne uporablja le za prenos, ter predvajanje/shranjevanje avdio/video informacije. Sistem DATA s pomočjo RTP

protokola omogoča tudi porazdeljeno procesiranje avdio/video informacije. Ker sistem DATA predvideva uporabo različnih naprav, z različnimi procesorskimi zmogljivostmi, predstavlja zmožnost oddaljenega (porazdeljenega) procesiranja, pomembno lastnost v smislu učinkovitosti in fleksibilnosti delovanja celotnega sistema. Na slika 6-2 je predstavljena zgradba tipičnega RTP paketa, ki se uporablja pri prenosu RTP tokov v sistemu DATA.



Slika 6-2: Struktura RTP paketa.

RTP protokol definira standardizirano obliko paketa za prenos avdio/video informacije preko interneta. RTP protokol za komunikacijo ne uporablja standardnih TCP ali UDP vrat. Edini standard, ki ga RTP protokol upošteva je, način komunikacije preko UDP protokola. Komunikacija se opravlja preko sodih vrat, naslednja liha vrata so namenjena kontrolnemu protokolu RTCP (RTP Control Protocol). RTP glavi sledi RTP vsebina, ki vsebuje zlogovni zapis avdio/video informacije. Najmanjša možna velikost glave RTP paketa je 12 zlogov. Polje »RTP glava« RTP paketa je opisano v Tabela 6-2 in 6-2.

Polje [velikost]	Pomen polja
Verzja (Version) [2 bita]	Označuje verzijo RTP protokola, ki se uporablja.
P [1 bit]	Uporablja se za označevanje dodatno pripetih zlogov na koncu RTP paketa.
X [1 bit]	Uporablja se za označevanje dodatne razširjene glave med standardno RTP glavo in RTP vsebino.
CC [4 bite]	Označuje število CSRC identifikatorjev, ki sledijo fiksni glavi.
M [1 bit]	Uporablja se na nivoju aplikacije. Definira ga profil aplikacije. V primeru, da je nastavljen, pomeni da ima RTP vsebina posebno lastnost za aplikacijo.
PT [7 bitov]	Definira format bremena in interpretacijo bremena v okviru aplikacije.
Številka sekvence (Sequence Number) [16 bitov]	Številka zaporedja se ob pošiljanju vsakega RTP paketa poveča za 1. Z njo lahko sprejemnik detektira izgubo paketa in ponastavi sejo.

Tabela 6-1: Zgradba glave RTP paketa.

Polje [velikost]	Pomen polja
Časovna značka (Timestamp) [32 bitov]	Časovna značilka reflektira instanco vzorčenja prvega podatka v RTP paketu. Instanca vzorčenja se mora določiti s pomočjo ure, katere vrednost narašča monotonno in linearno. Velikost koraka naraščanja mora dovoljevati sinhronizacijo in izračune za trepetanje (jitter).
SSRC (Synchronization source identifier)	Sinhronizacijski identifikator vira, enoznačno identificira vir RTP toka.
CSRC (Contributing source ID)	Identifikator prispevajajočih virov označuje vire, ki so prispevali k tvorjenju RTP toka večih virov.
Razširjena glava (Extension header)	Prva 32 bitna beseda glave vsebuje profilski identifikator (16 bitov) in dolžino specifikatorja dolžine (16 bitov), ki identificira dolžino razširjene glave v 32 bitnih enotah izključujoč 32 bitov razširjene glave

Tabela 6-2: Zgradba glave RTP paketa (nadaljevanje).

Del »RTP vsebina« RTP paketa, vsebuje breme v obliki sinhroniziranih vzorčenj avdio/video informacije. Spodnji tabeli prikazujeta kodiranje avdio/video informacije, ki jih podpira RTP protokol.

Ime	Tip	Frekvenca [Hz]	Število avdio kanalov	Reference
PCMU	Avdio	8000	1	RFC 3551
1016	Avdio	8000	1	RFC 3551
G721	Avdio	8000	1	RFC 3551
GSM	Avdio	8000	1	RFC 3551
G723	Avdio	8000	1	
DVI4	Avdio	8000	1	RFC 3551
DVI4	Avdio	16000	1	RFC 3551
LPC	Avdio	8000	1	RFC 3551
PCMA	Avdio	8000	1	RFC 3551
G722	Avdio	8000	1	RFC 3551
L16	Avdio	44100	2	RFC 3551
L16	Avdio	44100	1	RFC 3551
QCELP	Avdio	8000	1	
CN	Avdio	8000	1	RFC 3389
MPA	Avdio	90000	1	RFC 2250, RFC 3551
G728	Avdio	8000	1	RFC 3551
DVI4	Avdio	11025	1	

Tabela 6-3: Avdio/Video kodeki RTP protokola.

Ime	Tip	Frekvenca [Hz]	Število avdio kanalov	Reference
DVI4	Avdio	22050	1	
G729	Avdio	8000	1	
CellB	Video	90000		RFC 2029
JPEG	Video	90000		RFC 2435
nv	Video	90000		RFC 3551
H261	Video	90000		RFC 2032
MPV	Video	90000		RFC 2250
MP2T	Avdio/Video	90000		RFC 2250
H263	Video	90000		
GSM-HR	Audio	8000	1	
GSM-EFR	Audio	8000	1	
L8	Audio	varira	varira	
RED	Audio			
VDVI	Audio	varira	1	
BT656	Video	90000		
H263-1998	Video	90000		
MP1S	Video	90000		
MP2P	Video	90000		
BMPEG	Video	90000		

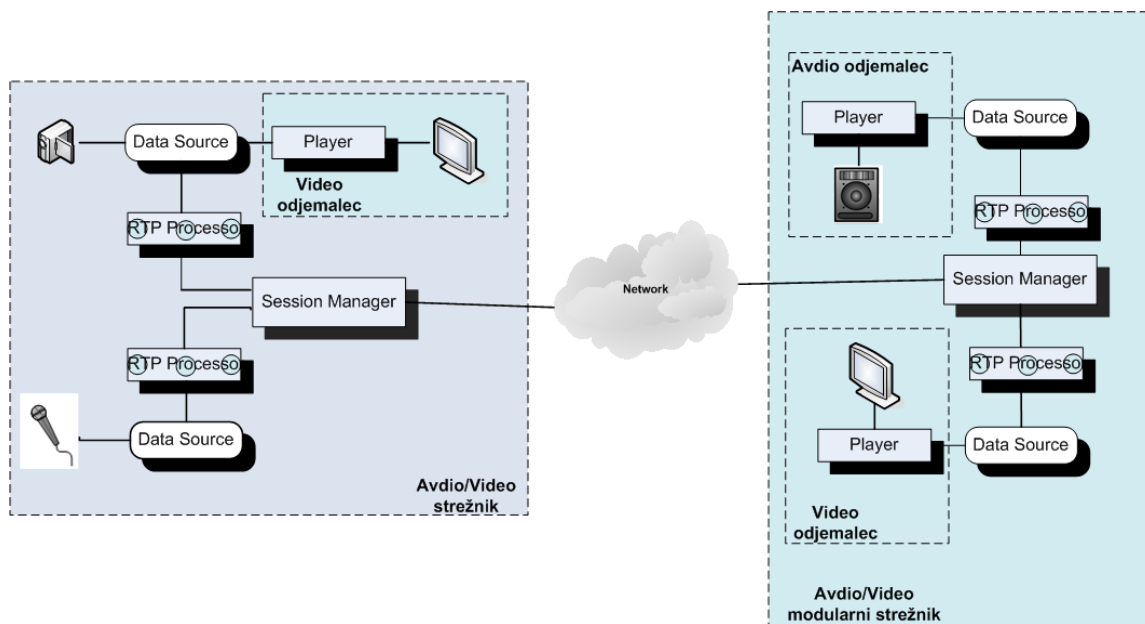
Tabela 6-4: Avdio/Video kodeki RTP protokola (nadaljevanje).

V nadaljevanju bomo predstavili aplikacijo, ki v okviru sistema DATA skrbi za učinkovito zajemanje, predvajanje in procesiranje avdio/video informacije ter zanesljiv prenos avdio/video podatkov v obliki RTP tokov.

6.2 RTP modul

Glede na zahteve inteligentnega okolja, se je v okviru kognitivnega sistema »DATA«, razvil samostojen RTP modul. Uporablja se lahko na katerem koli izmed gradnikov sistema »DATA« (DATA strežnik, DATA odjemalec ali DATA modularni strežnik). Po opravljenih analizah smo ugotovili, da okolje JMF 2.1.1e[59], ki ga ponuja Sun, ustreza vsem zahtevam izvedbe RTP modula. Modul se je implementiral kot podpora procesiranju avdio in video informacije, ter za potrebe prikaza avdio/video informacije brez prostorske omejitve. Uporabnik lahko ob uporabi DATA strežnika/DATA odjemalca procesira/pregleduje avdio/video informacijo, zajeto na kateremkoli DATA modularnem strežniku, ali pa pregleduje informacijo zajeto na DATA strežniku z uporabo oddaljenega DATA odjemalca. Oddaljeno procesiranje igra ključno vlogo pri enotah z malo ali brez procesorskih zmožnosti. Zajeta informacija se

v obdelavo pošlje oddaljenemu DATA strežniku, kjer se informacija procesira, rezultat ali dobljena multimedijska vsebina pa se pošlje nazaj k viru. Na spodnji sliki je predstavljen primer sistema izmenjave avdio/video informacije z uporabo RTP modula.



Slika 6-3: Primer implementacije RTP modula.

Iz sheme je razvidno da RTP modul trije pod moduli: avdio/video strežnik, avdio/video modularni strežnik in avdio/video klient. Avdio/video klient je namenjen predvajanju zajetih ali prejetih avdio/video tokov, avdio/video modularni in avdio/video strežnik se uporabljata za zajemanje, procesiranje, prejemanje in oddajanje avdio/video informacije. Spodnja tabela opisuje pomembnejše gradnike RTP modulov:

RTP gradnik	Slovensko poimenovanje	Opis objekta
<i>DataSource</i>	<i>Podatkovni vir</i>	Podatkovni vir predstavlja instanco, s katero zajemamo avdio/video informacijo iz naprav za zajemanje in jo pretvorimo v bitni tok podatkov, primeren za predvajanje ali nadaljnjo procesiranje. Podatkovni vir je možno neomejeno klonirati, s čimer se bitni tok podatkov hrani v originalnem podatkovnem viru in hkrati preusmerja v klonirane podatkovne vire. Združen podatkovni vir, predstavlja združitev dveh podatkovnih virov (originalni-kloniran, kloniran-kloniran ali originalni, originalni).

		V bitnem toku podatkov vira se prenaša združena informacija obeh podatkovnih virov.
<i>RTPProcessor</i>	<i>RTP procesor</i>	RTP procesor je objekt, ki se v okviru RTP modulov uporablja pri multipleksiranju/demultipleksiranju (v kombinaciji z upravljavcem sej), avdio/video procesiranju in avdio/video kodiranju (izvedba različnih kodekov nad avdio/video informacijo). Vhod RTP procesorja je lahko RTP tok (v primeru prejetja) ali pa bitni tok (kadar gre za procesiranje ali predvajanje avdio/video informacije).
<i>SessionManager</i>	<i>Upravljevec sej</i>	Objekt, ki je odgovoren za nadzor in upravljanje različnih RTP sej gradnika DATA sistema. Upravljevec sej informacijo RTP procesorja pretvori v RTP sejo (tvori RTP pakete s primerno obliko) ali prejema RTP tok podatkov, RTP pakete razčleni in posreduje pravilni instanci RTP procesorja. Skupaj z RTP procesorjem tvori modul multipleksiranja/demultipleksiranja RTP tokov.
<i>Player</i>	<i>Predvajalnik</i>	Objekt s katerim v okviru RTP modulov lahko predvajamo informacijo podatkovnih virov. Predvajalnik lahko dostopa do osnovnih, kloniranih ali združenih podatkovnih virov. RTP predvajalnik torej predstavlja objekt grafičnega vmesnika. Predvajamo lahko avdio, video ali združeno informacijo.
<i>Data Sink</i>	<i>Podatkovni ponor</i>	Objekt s katerim shranjujemo avdio/video informacijo v datoteko. Podatkovni ponor ima podobne lastnosti kot predvajalnik, le da ni namenjen direktnim vizualnim predstavitev. Z uporabo <i>DataSink</i> objekta, se opravi tudi shranjevanje v slikovne datoteke.

Tabela 6-5: Osnovni RTP gradniki.

Kot smo že omenili, implementiramo RTP modul na vseh treh gradnikih sistema »DATA« (DS, DMS in DC). V osnovi RTP modul podpira zajemanje, predvajanje, prejetje/pošiljanje avdio/video podatkov. Nekateri avdio/video strežniki in avdio/video modularni strežniki, pa lahko opravljajo še procesiranje informacije. Procesorske zmogljivosti so odvisne od tipa mobilne enote oz. gradnika na katerem se

RTP modul implementira. Informacijo zajemamo z uporabo avdio, video ali hibridnih (avdio/video) naprav za zajemanje ali iz medijskih datotek. V primeru, da naprava podpira zajemanje avdio in video informacije (je hibrid), se toka podatka lahko razdelita in procesirata ločeno. Zajeti tok podatkov se implementira v objekt podatkovnega vira (*DataSource*), in se z uporabo manj zahtevnega procesiranja pretvori v želeni medijski format (avi, mpeg4, mp3, mjpeg, itn.). Formati zajemanja in pošiljanja preko RTP protokola so polno nastavljivi (brez potrebe po ponovnem zagonu). Z RTP procesorjem v osnovi kodiramo (abstrahiramo) podatkovni vir v obliko, ki je primerna za prenos preko RTP protokola. Skupaj z upravljavcem sej, pa predstavljata tokovni multiplekser za pošiljanje podatkov oz. podatkovni de-multiplekser namenjen prejetju avdio/video informacije. RTP tokovi se lahko pošiljajo vsem poslušalcem v omrežju (broadband) ali pa so ciljno usmerjeni in imajo točno določen IP naslov in vrata kamor so namenjeni. V primeru, da je želeni medij sestavljen iz več tipov informacije, se vsak tip informacije prenaša po ločenem toku (drugačna številka vrat). Tako se avdio in video informacija enega medija prenašata ločeno v dveh tokovih. Tudi avdio/video strežnik in avdio/video modularni strežnik implementirata avdio/video odjemalca. V takšni implementaciji se avdio/video odjemalec uporablja za predvajanje de-multipleksiranih in dekodiranih podatkovnih tokov. RTP procesor na sprejemni strani lahko za namene predvajanja ponovno združi več tokov v multimedijski podatkovni vir, ki se direktno predvaja na vmesniku. Iz sheme enostavne implementacije RTP modula je razvidno, da se lahko zajeti podatki hkrati predvajajo na vmesniku in pošiljajo v obliki medijskih tokov. Podatkovni vir se v tem primeru klonira. Prednost opisane implementacije RTP modula je v sočasni uporabi nespremenjenega RTP modula na vseh objektih inteligentnega okolja. Želena funkcionalnost pa se definira z uporabo XML opisa. Kloniran podatkovni vir omogoča med seboj neodvisno procesiranje, predvajanje in pošiljanje medijskih tokov. Kadar potrebujemo dodatno funkcionalnost RTP modula, originalni vir (ki ga vedno hranimo v izvorni obliki) enostavno ponovno kloniramo in obravnavamo/procesiramo kot ločen vir.

7 UPORABA PODATKOVNIH BAZ

Sistem DATA za shranjevanje in statistično obdelavo podatkov potrebuje tudi nadgradljive in učinkovite podatkovne baze, ki smo jih implementirali v okviru MySQL 5.0 [40]. MySQL predstavlja zmogljiv relacijski upravljalni sistem podatkovnih baz RDBMS, ki je implementiran kot strežnik in omogoča hkratni dostop večih uporabnikov. MySQL podpira tudi osnovni SQL [41]. Pri tem SQL je standardni interaktivni programski jezik za upravljanje s podatkovnimi bazami, shranjevanje podatkov v podatkovne baze in izvajanje poizvedb iz podatkovnih baz. Jedro SQL jezika predstavlja ukazni jezik, ki omogoča vstavljanje, brisanje, posodabljanje in pridobivanje podatkov iz podatkovne baze. SQL ukazi se podajajo v obliki SQL stavkov. V spodnji tabeli so predstavljeni tipični začetki SQL stavkov in njihov pomen

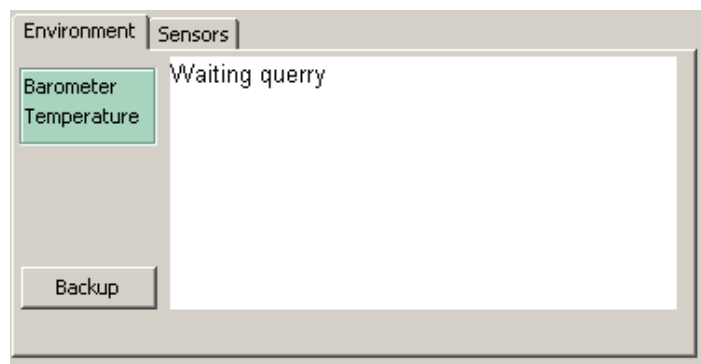
SQL ukaz	Pomen
SELECT	Izberi: ta stavek se uporablja za pridobivanje podatkov iz baze.
INSERT	Vstavi: ta stavek se uporablja z vnašanje podatkov v podatkovno bazo.
DELETE	Briši: ta stavek se uporablja za brisanje podatkov iz podatkovne baze.
UPDATE	Posodobi: ta stavek se uporablja za posodabljanje podatkov v podatkovni bazi.

Tabela 7-1: Tipični SQL ukazi.

SQL stavek formiramo na naslednji način: *SQL_ukaz polja_sql_tabele FROM ime_tabele*. Pogoji SQL stavka, podajamo z rezerviranimi besedama WHERE ali JOIN. V primeru, ko SQL ukazu podamo pogoj, se stavek izvede le nad vnosi v tabeli, ki ustrezajo pogoju. V okviru sistema DATA, smo razvili podatkovno bazo, ki trenutno služi predvsem statističnim analizam znotraj sistema DATA (shranjevanje meritev, beleženje RTP sej, itn.). V prihodnje, pa se bodo baze uporabljale tudi kot vir znanja raznim modulom z umetno inteligenco, za shranjevanje konfiguracijskih podatkov, itn. Za namene sistema DATA, smo razvili neodvisen, modularen Javanski paket DATA SQL, ki omogoča uporabo metod, ki so potrebne za upravljanje in manipulacijo s samo podatkovno bazo in ga bomo predstavili v nadaljevanju.

7.1 Programski paket DATA SQL

Programski paket DATA SQL vsebuje metode, ki omogočajo izvajanje operacij nad SQL bazami. V pripadajočih razredih tako paket implementira metode za vnos, brisanje, posodabljanje, pridobivanje podatkov, ter kreiranje in brisanja tabel. Za implementacijo DATA SQL paketa smo razvili javanski razred *java.sql*, ki vsebuje deklaracije za povezavo na podatkovno bazo in pošiljanje SQL ukazov podatkovni bazi. Grafični vmesnik DATA SQL programskega paketa, je predstavljen tudi na spodnji shemi.



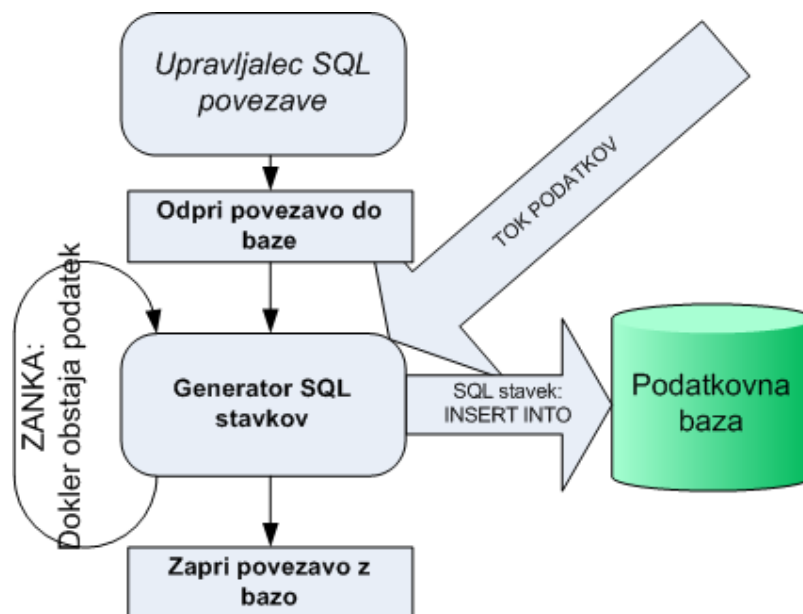
Slika 7-1: Grafični vmesnik DATA SQL programskega paketa.

S pomočjo standardnih vmesnikov (miška, tipkovnica ali igralna palica), lahko s pomočjo grafičnega vmesnika DATA SQL tako pridobivamo informacijo o zgodovini meritev s posameznih senzorjev in generiramo razne statistične diagrame o stanju v inteligentnem okolju. Rezultati poizvedb se nam trenutno shranjujejo dodatno v tekstovni obliki, v prihodnje pa bodo dodatno prikazani obliki raznih grafičnih animacij in tudi posredovani v govornih obliki z uporabo sintetizatorja govora (TTS) in pogovornih avatarjev. Prav tako tudi za DATA SQL načrtujemo implementacijo naprednejših načinov upravljanja, kot je npr. govor. V nadaljevanju bomo predstavili značilni operaciji, ki jih lahko s pomočjo DATA SQL programskega paketa opravljamo v okviru sistema DATA. Operaciji sta naslednji: avtomatski vnos podatka in poizvedba na zahtevo.

7.1.1 Avtomatski vnos podatka

Avtomatski vnos podatka, predstavlja tisto sposobnost sistema DATA, da samodejno in brez pomoči uporabnikov, shranjuje zajete podatke s senzorjev v inteligentnem

okolju v dano podatkovno bazo. Spodnja slika predstavlja funkcionalno shemo avtomatskega vnosa podatka. Podatek se lahko sprejme lokalno (v primeru, ko se podatkovna baza nahaja na enoti, ki direktno upravlja z virom podatkov) ali preko mrežne povezave (kadar podatek želimo shraniti na oddaljeno podatkovno bazo - centralni podatkovni vir).



Slika 7-2: Funkcionalna shema procesa shranjevanja v bazo.

Proces na sliki 7-2 pričakuje na vhodu DATA SQL nek tok podatkov, ki ga je potrebno neposredno shranjevati v podatkovno bazo. Za ta namen upravljalcev podatkovne baze odpre SQL sejo. Programsko se odpiranje SQL sej izvaja na naslednji način:

```

conn = getConnection(url+host+Data_base, user_name, password);
conn.createStatement();

```

Ob uporabi metode `getConnection` podrazreda `java.sql`, objekt `DriverManager` pripravi povezavo na določeno podatkovno bazo (`url+host+Data_base`). Podatki za avtentikacijo uporabnikov, se podajo v oblik uporabniškega imena in gesla. Ko je povezava pripravljena, se z metodo `createStatement()` vzpostavi seja s podatkovno bazo.

Seja traja vse dokler na vhodu DATA SQL še obstaja podatek, ki ga je potrebno vnesti v podatkovno bazo. V nasprotnem primeru se seja zapre, in sicer s klicem metode

close(). V času trajanja SQL seje, generator SQL stavkov najprej določi tip vhodnega podatka in na ta način attribute, ki jih mora za določen tip podatka podati ob vnosu v bazo (imena, tuji ključi, itn.). Pravilno formiran podatkovni objekt, se prenese v metodo za vnos, ki formira SQL stavek in ga izvede. Spodnji izsek iz programske kode, predstavlja vnos SQL podatka v podatkovno bazo.

```
sql_insert = "INSERT INTO ";
//tabela_vnosa = "mer_razd(senzor,razdalja,datum) ";
sql_values = " VALUES (" + marker + "," + value + "," + date + ")";
Statement insert_table = conn.createStatement();
String sql = sql_insert + table + sql_values;
insert_table.executeUpdate(sql);
```

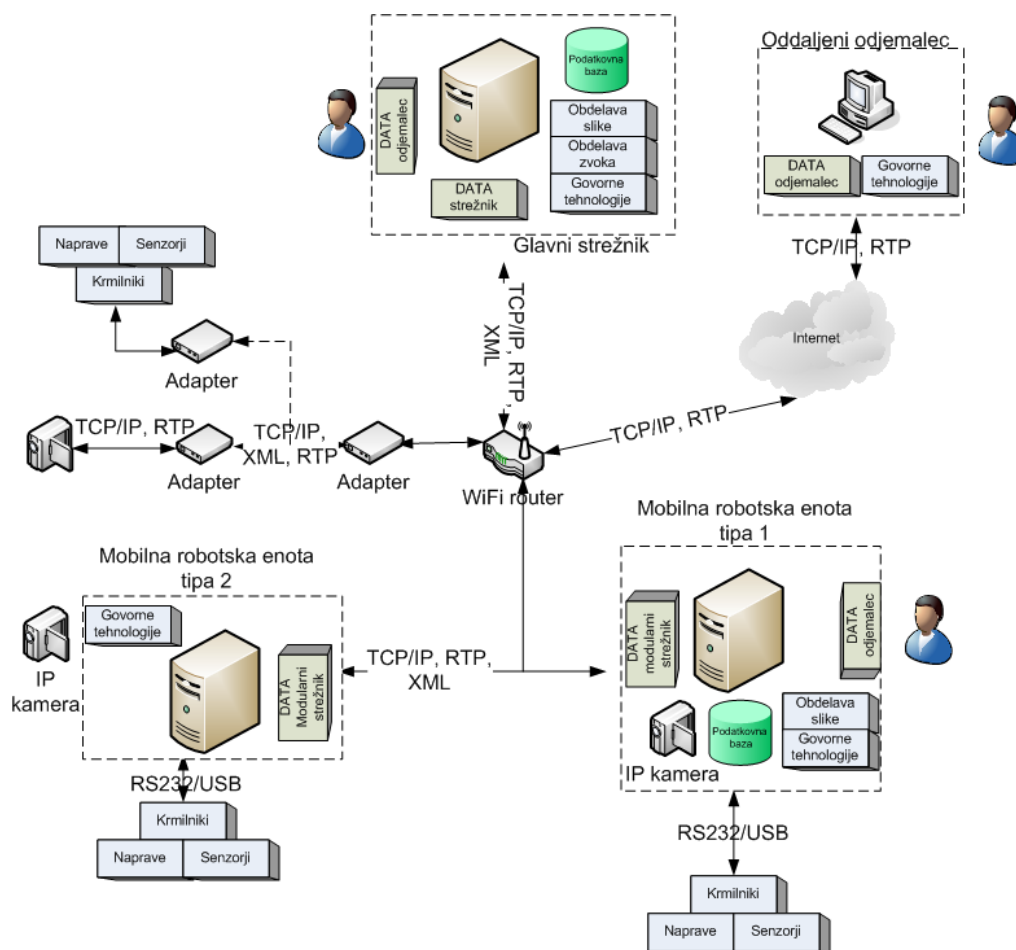
7.1.2 Poizvedba na zahtevo

Poizvedba na zahtevo, se izvede enako kot SQL SELECT stavek. Izvede se na identičen način, kot poprej predstavljen proces avtomatskega vnosa. SQL upravljalec ustvari novo sejo in jo odpre s pomočjo metod *getConnection* in *createStatement*. Generator SQL stavkov nato generira SQL stavek, glede na zahtevo, ki jo je prejel. V primeru poizvedbe na zahtevo, je to ime tabele, dodatni pogoji izbire (kot so časovni interval, velikost) itn. Po izvedbi SQL stavka, se rezultat poizvedbe prikaže na grafičnem vmesniku. Proces poizvedbe na zahtevo, se tako izvede v obliki naslednjega niza ukazov:

```
String sql = "SELECT " + value + " FROM " + table + " WHERE "
+ indicator + " LIKE " + "" + marker + "%" + "";
ResultSet result = send_query.executeQuery(sql);
while (result.next()) { // process results one row at a time
    mark = result.getString(1);
    size = result.getString(2);
    date = result.getString(3);
    SQL_query=SQL_query + mark + "\t" + size + "\t" + date + "\n";
}
```

8 KOMUNIKACIJSKI PROTOKOLI V SISTEMU »DATA«

Slika 8-1 prikazuje podrobno analizo sistema »DATA« na nivoju protokolov. Vse povezave, ki jih predvidevamo v sistemu »DATA«, uporabljajo za izmenjavo podatkov TCP/IP protokol. Osnovni protokol na osnovi XML, se uporablja za prenos ASCII podatkov in prenos XML opisov scenarijev, ki definirajo obnašanje posameznih modulov sistema »DATA«. RTP protokol o katerem smo že pisali v okviru poglavja za avdio/video prenos, se uporablja za učinkovit prenos avdio in video tokov med posameznimi gradniki sistema »DATA«. Senzorji, krmilnik in druge naprave so s sistemom »DATA« povezani preko serijskih povezav (USB2.0/Bluetooth/ZegBee/RS232 tehnologija). Električna linija predstavlja enega izmed ne mrežnih načinov prenosa podatkov.



Slika 8-1: Arhitektura sistema »DATA« na protokolnem nivoju.

Naprave, ki niso zmožne gostiti mrežne (žične ali brezžične povezave), lahko na sistem »DATA«, povežemo preko obstoječega električnega omrežja. Z uporabo posebnih adapterjev, lahko TCP/IP pakete preusmerimo na električne povezave in tako krmilimo naprave povezane v to obliko komunikacije. Prenos podatkov preko električnega omrežja, je pomemben tudi zaradi možnosti razbremenitve TCP/IP omrežja, ki jo ponuja. Če se sistem za prenos preko električnih povezav implementira kot dodaten način povezave med gradniki »DATA« sistema, lahko gradniki v primeru zasičenosti omrežja izberejo alternativno komunikacijsko pot, po kateri si izmenjujejo informacijo.

V nadaljevanju predstavljamo osnovne lastnosti posameznih komunikacijskih protokolov, ki smo jih uporabili znotraj sistema DATA. Glede na njihovo uporabo v sistemu, jih delimo v nadaljevanju na strojne (komunikacijski protokoli med posameznimi strojnimi komponentami sistema DATA) in višje nivojske mrežne protokole, ki implementirajo komunikacijo med posameznimi gradniki sistema DATA (DATA strežnik, DATA klient in DATA modularni strežnik) na aplikativnem nivoju.

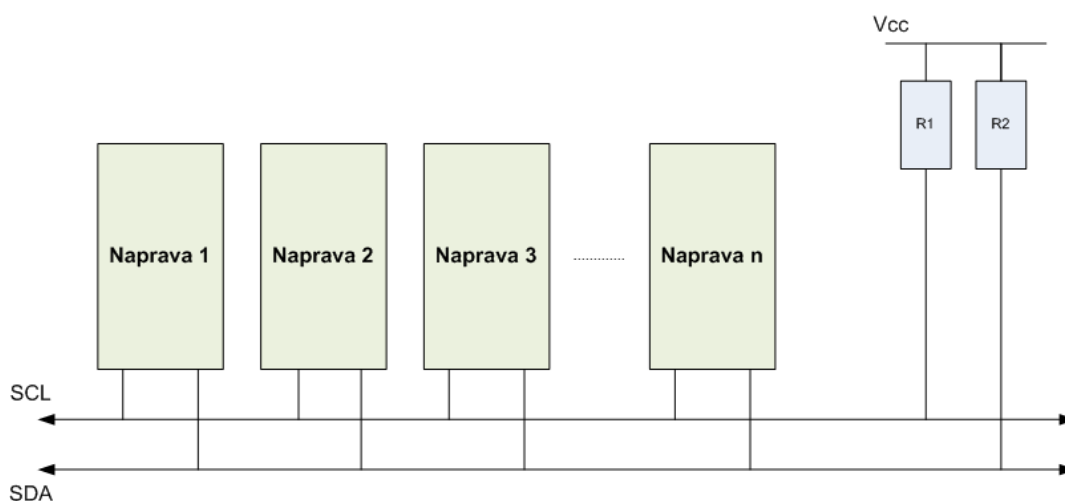
8.1 Strojni protokoli

Strojni protokoli skrbijo za učinkovito komunikacijo med strojnimi komponentami sistema DATA. Atmelov mikrokrmilnik ATmega8 [38], je 8 bitni CMOS energetsko nizko potrošni mikrokrmilnik grajen na AVR RISC arhitekturi. Z izvajanjem močnih ukazov v enem urinem ciklu, krmilnik dosega izhodno hitrost do 1MIPS na Mhz. Sistemski arhitekti lahko z uporabo krmilnika optimizirajo vezja glede na potrošnjo energije in hitrost delovanja. ATmega8 ponuja sledeče: 8K zlogovni sistemsko programibilni pomnilnik, z možnostjo branja med pisanjem, 512 zlogovni EEPROM, 1K zlogovni SRAM, 23 splošno uporabnih vhodno/izhodnih priključkov, 32 splošno uporabnih registrov, tri fleksibilne časovnike s primerjalnimi načini, notranje in zunanje prekinitve, serijsko programirljiv USART, zlogovno usmerjen I2C vmesnik, 6 kanalni ADC pretvornik z 10 bitno resolucijo, SPI serijskimi vrati itd. V okviru razvoja mobilne robotske platforme za inteligentna okolja smo ga uporabili v tej fazi razvoja predvsem zaradi hitrosti in nizke energijske porabe. Strojna podpora številnim serijskim protokolom, pa omogoča enostavno priključevanje in upravljanje s številnimi senzorji. Strojni protokoli v nadaljevanju so opisani v kontekstu uporabe na ATmega8

mikrokrmilniku. V istem kontekstu so poimenovani tudi posamezni registri, ki opisujejo delovanje protokolov.

8.1.1 I2C (TWI) – povezava po dveh linijah (ATmega8)

TWI je namenjen povezavi ene ali več naprav, pri čemer se za povezavo uporabljata le dve bi-direktni liniji (bus), imenovani SCL in SDA. SCL je linija, po kateri se prenašajo urni impulzi, SDA pa linija po kateri se izmenjujejo podatkovni biti. Vsak preneseni bit na podatkovni liniji, spremlja urni impulz na urni liniji. Razen za pogoj START in STOP, morajo biti podatki stabilni, ko je ura na visokem nivoju.

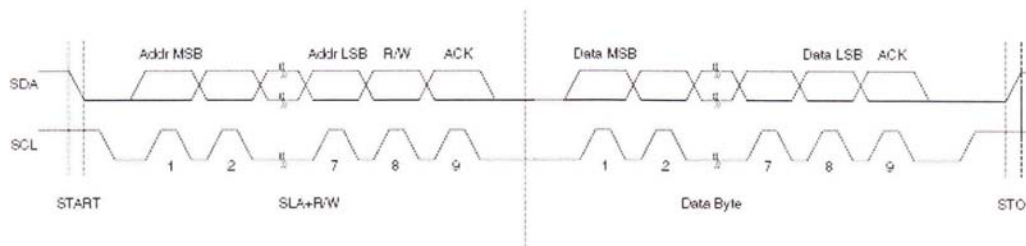


Slika 8-2: TWI povezava.

V I2C povezavah, se komponente delijo na gospodarje in sužnje (z uporabo programskih knjižnic je lahko komponenta hkrati gospodar in suženj, vendar pa mora za ta namen implementirati dve različni TWI povezavi). Gospodar TWI povezave je tista naprava, ki generira urni takt na liniji SCL. Samo gospodar lahko prične komunikacijo (podajanje pogoja START) in jo konča (podajanje pogoja STOP). Pogoja START in STOP se signalizirata s postavitvijo linije SDA na nizek nivo, medtem ko je linija SCL na visokem nivoju. Med pogojema START in STOP, se povezava obravnava kot zasedena in drugi gospodar nadzora nad takšno povezavo ne more prevzeti.

Vsi naslovni paketi na povezavi TWI, so posredovani v 9-bitnem načinu in so sestavljeni iz sedmih naslovnih bitov, kontrolnega bita READ/WRITE in bita ACK. V primeru da je bit READ/WRITE postavljen, se pričakuje operacija »branje iz naprave«.

V nasprotnem primeru je pričakovana operacija »pisanja v registre naprave«. V primeru da suženj detektira, da želi gospodar z njim komunicirati, mora najkasneje v času poteka cikla 9 bitov ACK, linijo SCL spustiti na nižji nivo.



Slika 8-3: Prenos podatkov po TWI povezavi [38].

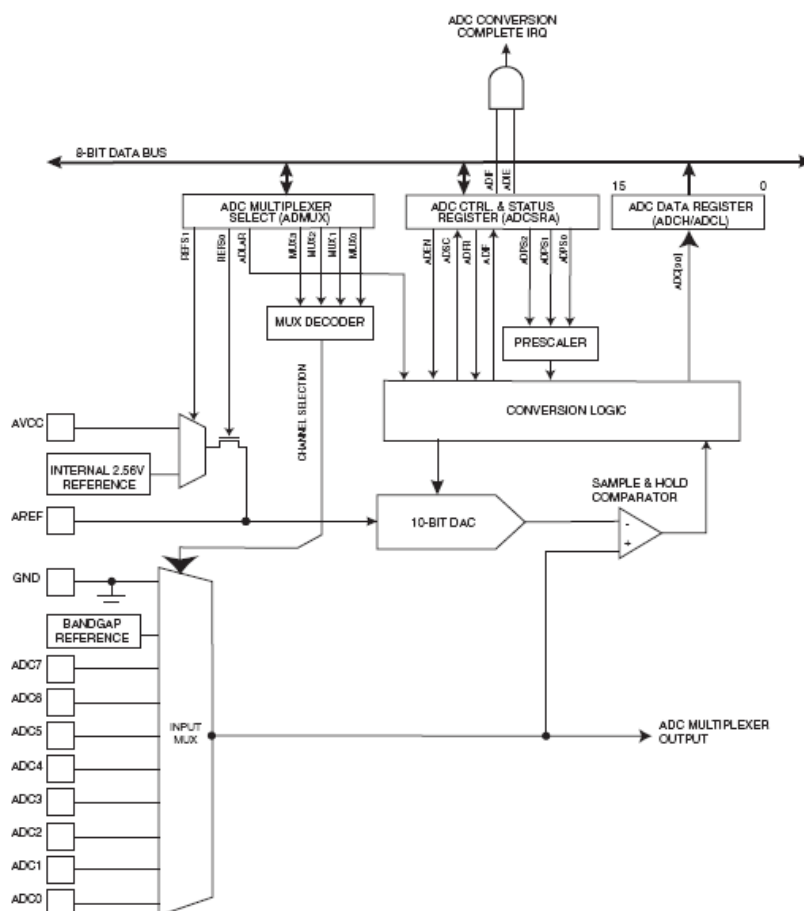
Naslovi suženjskih naprav so lahko poljubni, vendar pa je naslov 0000000 rezerviran za t.i. generalni klic, hkrati komunikacijo z vsemi napravami na I2C povezavi. Generalni klic se uporablja kadar gospodar želi zapisati neko vrednost v vse naprave povezave; generalno branje je nesmiselno, saj na tak klic odzovejo vse naprave hkrati (nemogoče saj prva naprava povezavo zaklene, ostale pa paketov ne pošiljajo več kot enkrat).

8.1.2 10 - bitna ADC pretvorba (ATmega8)

Analogno – digitalna pretvorba je ena izmed najpogostejših operacij, ki jih opravljamo nad analognimi električnimi signali. Operacija pretvori vhodno zvezno napetost v številčno (digitalno) obliko. Mikrokontrolnik ATmega8 podpira 10 bitno sukcesivno aproksimacijo ADC. ADC je povezan na 8 kanalni multipleksor, ki omogoča implementacijo osmih eno linijskih ADC povezav. Eno-linijska ADC povezava, kot referenco jemlje GND (0 V). ADC vsebuje vezje "vzorči" in "zadrži", ki zagotavlja, da je vhodna ADC napetost med pretvorbo zadržana na konstantnem nivoju. Referenčna napetost določi interval pretvorbe. Eno-linijske pretvorbe, ki presegajo referenčno napetost, bodo imele vrednost približno 0X3FF. Referenčna napetost je lahko notranja (2.56V), ali pa jo dovedemo na specifičen priključek krmilnika AREF. V primeru, da je na priključek AREF pripeljan fiksni vir napetosti, uporabnik v aplikaciji ne more uporabljati druge referenčne napetosti.

Tako nam AD pretvorba mikrokontrolnika z resolucijo 10 bitov nam pretvori analogno vhodno napetost v 10 bitno digitalno vrednost. Najnižja napetost je določena z

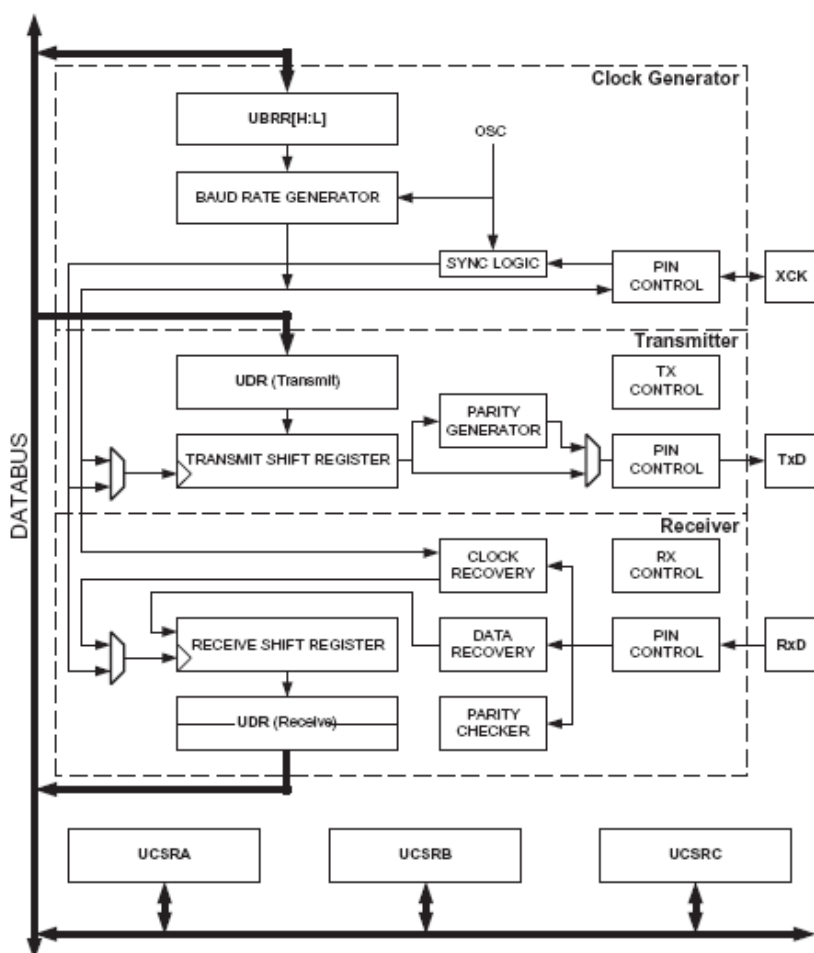
maso (GND), najvišja vrednost pa je napetost na referenčnem priključku (AREF). Resolucijo AD pretvorbe določimo z velikostjo razpona med najvišjo in najnižjo vrednostjo, ter bitnim razponom rezultata. V primeru uporabe mikrokrmilnika ATMEL Atmega8, dani analogni vhodni kanal izberemo z zapisom v bit MUX, registra ADMUX. AD omogočimo s postavitvijo bitov ADC, ADEN in ADCSRA. Vrednost pretvorbe se hrani v registrih ADCH in ADCL. Rezultat je desno poravnan (če postavimo bit ADALR v registru ADMUX, pa levo). V primeru leve poravnosti in 8 bitne resolucije, je potrebno prebrati le vrednost registra ADCH, drugače pa moramo najprej prebrati register ADCL in šele nato register ADCH. S tem zagotovimo, da podatkovni registri pripadajo isti pretvorbi. AD pretvorba se zaključi, ko se prebere ADCH register.



Slika 8-4: Blokovna shema ADC pretvorbe[38].

sinhronizacijo in vzdržuje suženjski števec bitov v sinhronizaciji z urnim generatorjem gospodarja.

V načinu delovanja kot »gospodar«, se nastavita bita MSTR in SPCR kontrolnega SPI registra, uporabnik pa lahko določi usmerjenost !SS priključka. Če je !SS priključek konfiguriran kot izhod, se priključek obravnava kot generalni izhod, ki ne vpliva na SPI povezavo. V nasprotnem primeru mora biti priključek !SS postavljen na visokem nivoju, saj le tako lahko zagotovimo način delovanja kot »gospodar«. V primeru, da neka druga naprava postavi priključek !SS na nizek nivo, medtem ko je priključek !SS v funkciji vhoda, SPI sistem to zazna kot prevzem SPI povezave nekega drugega »gospodarja«, ki je izbral sužnja, in takoj začne pošiljati podatke po povezavi.



Slika 8-7: Blokovna shema UART modula[38].

8.1.4 USART ali UART - univerzalni asinhroni serijski protokol (ATmega8)

UART modul delimo na tri logične enote: generator ure, oddajna enota in sprejemna enota. Blokovna shema UART modula je predstavljena na sliki 8-7. Logika enote generiranja ure, je sestavljena iz sinhronne logike za zunanji vnos ure, ki je uporabljena pri sinhronem suženjskem načinu delovanja. Enota za generiranje ure generira osnovno (referenčno) uro za oddajno in sprejemno enoto. UART podpira štiri načine delovanja: normalen asinhroni način, dvakratni asinhroni način (dvakratna hitrost), sinhroni način »gospodar« in sinhroni način »suženj«. Oddajno enoto sestavljajo pisalni register, pomikalni register, paritetni generator in kontrolna logika za obravnavanje različnih serijskih formatov. Prenos podatkov se inicializira z zapisom podatkov v oddajni register. Podatki za prenos se v pomikalni register prestavijo, ko je le-ta pripravljen za ponovno oddajo.

Sprejemna enota vsebuje za sprejem asinhronih podatkov modul obnovitve, paritetni prevajalnik, kontrolno logiko, pomikalni register in dvonivojski sprejemni register. Ko se sprejemna enota aktivira, se običajne operacije priključka RxD prepisejo, priključek pa postane serijski vhodni priključek UART komunikacije. Sprejemnik začne sprejemati, ko na povezavi zazna veljaven začetni bit. Podatki se zapisujejo v sprejemni pomikalni register vse dokler ni poslan prvi pogoj STOP (vsi nadaljnji pogoji STOP se ignorirajo). Takrat se vsebina pomikalnega registra prestavi v sprejemni register, ki se ga bere z branjem lokacije UDR I/O.

V nadaljevanju si pogledjmo še višje nivojske protokole, ki jih DATA sistem implementira za učinkovit prenos informacije med gradniki sistema DATA. RTP protokol smo že predstavili v poglavju 6 *ZAJEMANJE IN PRENOS AVDIO IN VIDEO INFORMACIJE* zato ga tukaj ne bomo več podrobneje predstavljali.

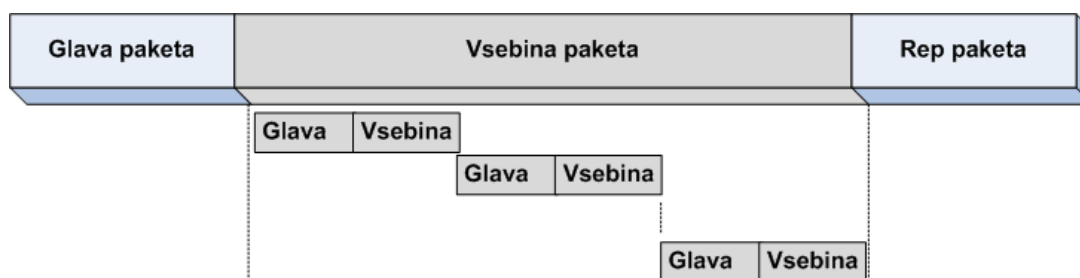
8.2 Mrežni protokoli

Sistem DATA podpira tako žične kot tudi brezžične povezave. V okviru omrežja inteligentnega okolja smo glede na različne tipe informacij, ki jih želimo prenašati, adaptirali in preoblikovali obstoječe protokole, ki jih podpirajo tako žične kot brezžične povezave. Za boljši nadzor in lažjo uporabo prejete informacije, smo v okviru

protokolne implementacije sistema DATA, uvedli še posebna kodiranja vsebine paketov DATA protokolov.

8.2.1 TCP/IP protokol

TCP/IP protokol v sistemu »DATA«, je izveden na principu »zahteva-odgovor«, kjer zahtevi sicer nujno ne sledi odgovor (še vedno se na najnižjem nivoju uporabljajo potrditveni paketi, ki pa za uporabnika niso vidni). Protokolna implementacija omogoča razvijalcu sistema definicijo različnih tipov paketov, ki se prenašajo po omrežju.



Slika 8-8: Struktura TCP/IP paketa.

TCP/IP paket je sestavljen iz glave, vsebine in repa paketa. Glava paketa identificira za kašen tip paketa gre in ali je nanj potrebno odgovoriti. Polje vsebine paketa lahko nosi enega ali več tipskih paketov. Pomen informacije se določi na podlagi glave paketa. Rep paketa je namenjen označevanju konca TCP/IP paketa in lahko vsebuje tudi statusna sporočila. V tabelah 8-1 in 8-2 so predstavljeni tipični TCP/IP paketi, ki so uporabljeni v sistemu DATA.

Naziv paketa	Glava paketa	Glava vsebine	Rep paketa	Namen
<i>Krmilni paket (paketi ne potrebujejo posebnega odziva)</i>	<COM>	<MOV> up	<END>	Krmiljenje mobilne enote brez posebne prioritete.
		<MOV>right	<HIGH>	Krmiljenje mobilne enote brez visoke prioritete.
	<CAM>	<LEFT>	<END>	Krmiljenje kamere za konfiguracijsko določeno število korakov brez posebne prioritete.
		<RIGHT> 50	<LOW>	Krmiljenje kamere 50 korakov (°) z nizko prioriteto.

Tabela 8-1: Struktura tipnih TCP/IP paketov.

Naziv paketa	Glava paketa	Glava vsebine	Rep paketa	Namen
<i>Nadzorni paket (paketi pričakujejo odgovor)</i>	<SEN>	<R>	<END>	Paket, ki definira zahtevo po trenutnih meritvah z vseh ultrazvočnih senzorjev.
		<R 1 2 >	<END>	Paket, ki definira zahtevo po trenutnih meritvah senzorjev označenih z 1, 2.
<i>Nadzorni paket (paketi pričakujejo odgovor)</i>	<SEN>	<T>	<START>	Paket, ki opredeljuje zahtevo po konstantnem poročanju o meritvah vseh temperaturnih senzorjev.
		<P 1 >	<STOP>	Paket, ki opredeljuje zahtevo po prenehanju konstantnega poročanja o meritvah senzorjev tlaka označenih z 1.
	<SEN>	<R1>data<R2>data	<END>	Paket, ki sledi kot odgovor na zahtevo po trenutnem statusu senzorjev 1 in 2.
<i>Poizvedbeni paketi (kot odgovor se pričakuje datoteka)</i>	<SQL>	<R 1 7 >	<5>	Paket, ki zahteva poročanje o statusu senzorjev 1 in 7 zadnjih 5 dni, vključno z današnjim dnem. Kot odgovor pričakuje datoteko.
			<END>	Paket, ki zahteva poročanje o statusu senzorjev 1 in 7 za celotno obdobje delovanja vključno z današnjim dnem. Kot odgovor pričakuje datoteko.

Tabela 8-2: Struktura tipnih TCP/IP paketov (nadaljevanje).

Na protokolnem nivoju predstavlja DATA odjemalec tisti modul, ki implementira TCP/IP povezavo preko Java vtičnic ("sockets") in pošilja zahteve, ter prejema odgovore DATA strežnika in obratno, saj je povezava polno dvosmerna. Kontrolna in ukazna sporočila se formirajo v obliko TCP/IP poprej omenjene strukture in tako prenašajo skozi omrežje. V ta namen se implementirajo TCP/IP odjemalci in oddajalci, ki podpirajo nitenje [60](vsaka TCP/IP povezava predstavlja eno instanco – eno nit). DATA strežnik v tem kontekstu predstavlja hibrid, saj implementira tako TCP/IP poslušalca in generator TCP/IP paketov, ki pakete razpošilja na določene destinacijo (definirane z IP naslovom in številko vrat) . Hkrati DATA strežnik predstavlja edini

gradnik, ki se lahko poveže direktno na enega ali več DATA odjemalcev, oziroma enega ali več DATA modularnih strežnikov. Na protokolnem nivoju predstavljajo DATA modularni strežniki, podobno kot DATA strežniki, vmesnik med ethernetom in napravami, ki so povezane na DATA modularni strežnik preko povezave USB, Bluetooth, ZigBee ali RS232 (serijska).

8.2.2 Serijski protokol

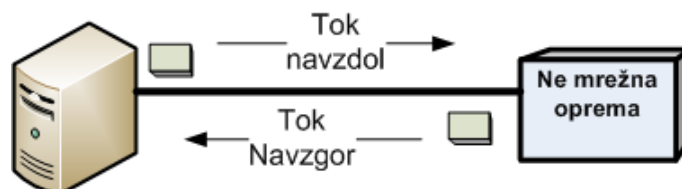
V Data sistemu lahko za prenos serijskih podatkov uporabljamo pretvornike ethernet – serijski, BlueTooth – serijski ali USB- serijski. Ti pretvorniki se uporabljajo za povezave naprav, ki ne podpirajo ethernet povezave, in DATA modularnih strežnikov ter DATA strežnikov. Preko teh povezav lahko s takšnimi napravami sistem upravlja, jih krmili in iz njih prebira podatke. Podobno kot TCP/IP seje lahko tudi seje serijskih povezav gostijo različne tipe paketov. Tipi paketov so odvisni od implementacije posameznih naprav, specificirajo pa se s pomočjo uporabe konfiguracijskih skript. Primeri paketov so podani v spodnji tabeli.

Naziv paket	Oblika paketa	Zahteva/Odgovor
<i>Statusni paket</i>	<S>S1 S2 S3<END>	Zahteva po statusu senzorjev S1, S2 in S3.
	<S11>120.0<S12>122.5<S13>120.5 <END>	Odgovor v obliki treh meritev senzorja S1, opravljenih na poziciji 1, 2, 3.
<i>Kontrolni paket</i>	<CON>M1 0<END>	Zahteva po postavitvi senzorskega modula M1 v spanje.
<i>Krmilni paket</i>	<COM>S 3<END>	Zahteva po premiku mobilne enote v smeri 3.
	<COM>R1 7<END>	Zahteva po premiku senzorja R1 na pozicijo 7.

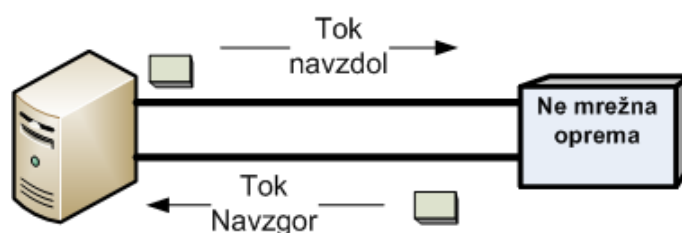
Tabela 8-3: Primeri serijskih paketov.

DATA sistem lahko gosti več serijskih povezav naenkrat. Število sej je odvisno le od števila razpoložljivih adapterjev. Serijske povezave lahko implementiramo kot polovični-dupleks (ena seja, podatki tečejo v vsako smer izmenično) ali kot polni dupleks (tok podatkov poteka v obe smeri hkrati, vsaka smer ima svojo serijsko sejo).

Implementacija obeh načinov serijske povezave je predstavljena na spodnjih dveh slikah.



Slika 8-9: Enojna serijska povezava (polovični dupleks).



Slika 8-10: Dualna serijska povezava (polni dupleks).

8.2.3 Osnovni XML protokol

Osnovni XML protokol [61] se v sistemu DATA uporablja predvsem za prenos ASCII podatkov med moduli sistema DATA in omogoča sočasne prenose v obe smeri (polni dupleks). Izmenjava informacij poteka na naslednji način:

- DATA odjemalec \leftrightarrow DATA strežnik
- DATA strežnik \leftrightarrow DATA modularni strežnik

Ko se vzpostavi povezava med DATA odjemalcem in DATA strežnikom, lahko DATA strežnik vzpostavi povezavo do drugih (predvidenih v XML opisu obnašanja klienta) DATA modularnih strežnikov. Po vzpostavljeni povezavi, pa izmenjava XML paketov med njimi, poteka direktno med DATA odjemalcem in DATA modularnim strežnikom. Za zagotavljanje učinkovite in enoumne komunikacije, ki uporablja XML pakete, smo formirali specifično zgradbo XML paketa, ki je predstavljena na **Slika 8-11**liki 8-11.



Slika 8-11: Zgradba XML paketa.

Polje *Glava XML paketa*, se začne z **<XML:**. S tem povemo, da vsebina paketa predstavlja XML obliko zapisa in ustreza pravilom formiranja XML vsebin. Nadaljujemo z poimenovanje operacije/akcije, ki jo želimo izvajati z vsebino XML paketa (npr.: *krmili*, *pošlji RTP*, *manipulacija baze*, itn.) in glavo zaključimo z **><Data>**. Polje *Prejemniki* v XML paketu, je definirano kot pod-struktura XML paketa in je označena z **<Receivers> </Receivers>**. Med definicijo začetka in konca polja **Receivers**, zapisujemo izbrane naslovnike. Naslovnike deklariramo znotraj XML oznak **<Receiver id='id_naslovnika' />**. Polje *Prejemniki*, ima lahko neomejeno število naslovnikov, v primeru, da pa polje *Prejemniki* nima vnosov, predstavlja vsebina XML paketa lokalno operacijo, ki jo mora izvesti lokalni sistem (npr. DATA strežnik ali DATA modularni strežnik). Polje *Vsebine*, deklariramo z naslednjimi XML oznakami **<Content></Content>**. To polje vsebuje nabor tistih akcij, ki naj jih naslovniki izvedejo. Posamezne akcije se podajajo v obliki **<Action do="akcija" objects="objekt_1, objekt_2" />**. V primeru da polje *Vsebine* ne vsebuje akcij, XML paket obravnavamo kot testni paket delovanja sistema. Parameter *objects* je lahko prazen oz. ni nujno da je sploh prisoten. Če XML paket vsebuje odziv na zahtevano operacijo, se odziv zapiše v obliki **<Response object="objekt_1">response</Response>**. Rep XML paketa lahko vsebuje opombe v obliki prioritete akcij **<Priority>Vrednost</Priority>**.

Zaradi ohranjanja XML strukture mora rep nujno vsebovati oznako **</Data>**. V nadaljevanju je prikazana tipična struktura XML paketa, ki povzema zgoraj navedene lastnosti XML paketa. Predstavljeni paket definira lokalno operacijo poizvedbe v bazo, zato XML objekt *Recivers* v tem primeru ne vsebuje vnosov.

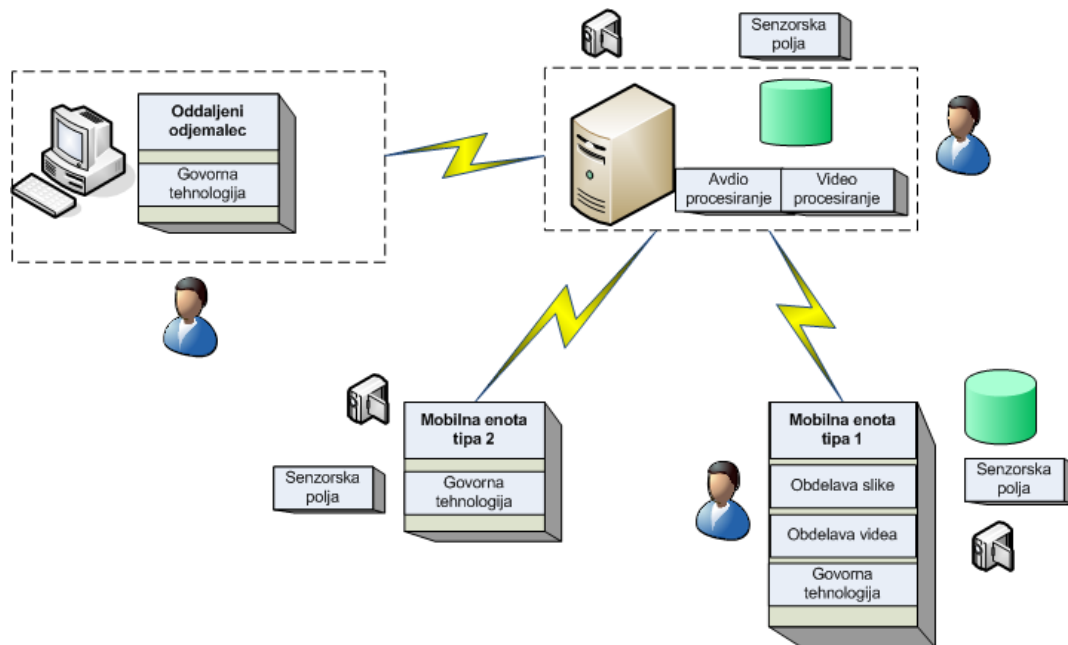
```

----- Glava XML paketa-----
<XML: db_manipulation>
<Data>
----- Telo XML paketa-----
  <Receivers></Receivers>
  <Content>
    <Action do="GET " objects="sensor_1, sensor_5"/>
  </Content>
----- Rep XML paketa-----
</Data>

```

9 DISTRIBUIRANO PROGRAMSKO OGRODJE »DATA«

V poglavju 9 se bomo posvetili razlagi programskega ogrodja »DATA«, ki smo ga razvili v okviru diplomske naloge.



Slika 9-1: Inteligentno okolje z uporabo ogrodja »DATA«.

9.1 Specifikacije za distribuirano programsko ogrodje »DATA«

V kompleksnih inteligentnih okoljih lahko komunicira s sistemom »DATA« več uporabnikov hkrati. V splošnem vsak izmed uporabnikov želi ali potrebuje drugačno obnašanje sistema, ki nadzoruje in upravlja z napravami v inteligentnem okolju. Zato je pomembno učinkovito in fleksibilno specificirane obnašanja sistema in njegovih modulov, ki pa ne zahteva preveč konfiguracijskega napora ali celo razvoja novih verzij sistema. Povečevanje števila naprav, modulov in števila raznih tipov mobilnih enot rezultira v eksponentno naraščajoči kompleksnosti sistema in vedno težjem nadzoru posameznih modulov in prilagajanju delovanja sistema posameznim uporabnikom. Dogodki proženi s strani uporabnika ali sistema se navadno pojavljajo v asinhronih časovnih trenutkih, morajo pa vedno biti v realnem času detektirani in procesirani. Samo tako se sistem lahko odzove uporabniku na primeren način. Vse te zahteve odražajo pomembnost pravilne izbire rešitve v smislu arhitekture in izvedbe

distribuiranega sistema, ki ga želimo uporabljati znotraj inteligentnega okolja. Zelo pomembni karakteristiki inteligentnega okolja sta še robustnost in učinkovito detektiranje, obravnavanje in odpravljanje napak. V okviru diplomske naloge smo razvili programsko ogrodje »DATA«, ki temelji na principu delovanja končnih strojev. Ogrodje »DATA« predstavlja inovativen alternativni pristop k snovanju arhitekture distribuiranih sistemov in odpravlja pomanjkljivosti arhitekturo črne table (predstavljene v poglavju 2.1 *Arhitekture inteligentnih okolij*). Ogrodje »DATA« omogoča fleksibilen tok podatkov med komunikacijskimi moduli sistema (ASCII, avdio/video podatki, itn.), avtomatsko določanje obnašanja sistema (za vsakega uporabnika posebej) z uporabo XML opisov in sočasno uporabo inteligentnega okolja več uporabnikom hkrati. Inteligentno okolje predstavljeno na sliki 9-1 gradijo: glavni strežniški modul, več tipov mobilnih enot in en ali več oddaljenih odjemalcev. Strežniški modul skrbi za procesiranje in izmenjavo podatkov, katerih vir so mobilne enote ali oddaljeni odjemalci, upravlja s podatkovnimi viri in nudi različne storitve avdio/video procesiranja. Uporabniška interakcija s sistemom lahko poteka na lokalnem ali oddaljenem nivoju. Vhodni uporabniški vmesniki so lahko govor, tipkovnica, igralna palica, zaslon na pritisk, mobilni telefoni, itn. Ogrodje »DATA« ne omejuje uporabljenih tipov mobilnih enot, saj se tipi mobilnih enot specificirajo glede na potrebo in namembnost sistema. Tako se funkcionalnosti mobilnih enot omejujejo v bistvu glede na razpoložljivo napajanje in procesorsko moč. Razpoložljivo napajanje neposredno vpliva na procesorsko moč enote (močnejši procesor, več procesorjev → večja poraba energije), zato na nekaterih enotah omogočamo recimo samo zajemanje podatkov, na zmogljivejših enotah pa lahko tudi procesiranje podatkov (slike, govora, zajetih senzorskih podatkov, itn.). S funkcionalnega stališča DATA specificira naslednje oblike gradnikov sistema:

Glavni strežniški modul je modul, ki podpira storitve zahtevnega procesiranja, storitve prikaza in upravljanja s podatkovnimi skladišči (podatkovne baze). Modul zajema avdio/video podatke, procesira sliko in signale, prikazuje rezultate in posreduje avdio/video tok ostalim modulom. Interakcija človek-stroj se izvaja z uporabo miške, tipkovnice, mobilnih naprav ali govora, dodatno pa je vgrajena podpora za implementacijo naprednejše multimodalne interakcije človeka in sistema. Model gradnika predstavlja porazdeljen sistem, ki je fizično sestavljen iz enega ali več

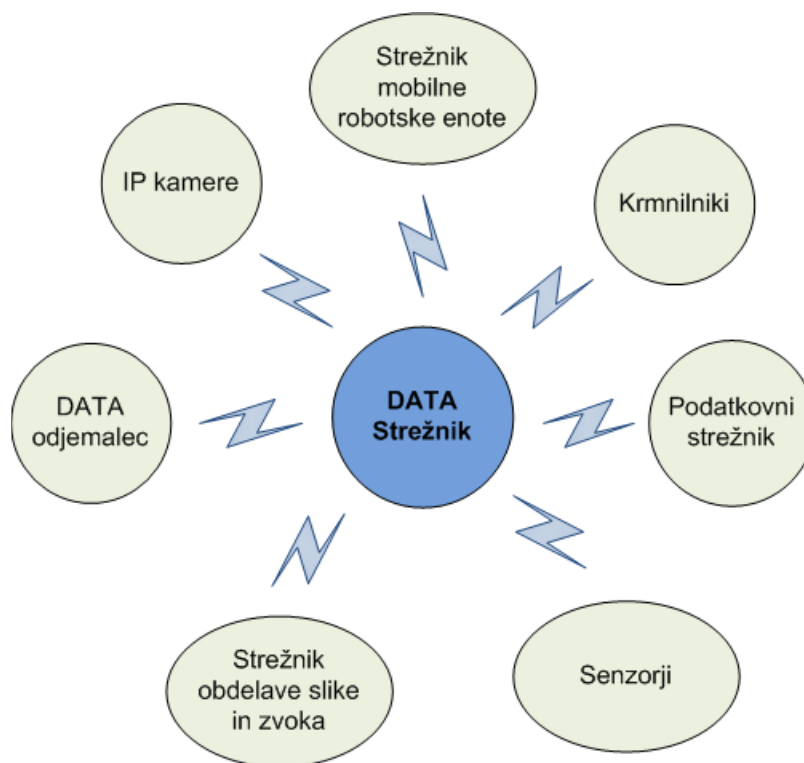
strežniških komponent (komunikacijski strežnik, strežnik za procesiranje, strežnik, ki gosti podatkovno skladišče, itn.). Z učinkovito povezavo strežniških postaj in naprav ter modalnosti okolja, DATA programsko ogrodje glavni strežniški modul pretvori v sposoben virtualni strežnik.

Modul oddaljenega klienta – gradnik je namenjen oddaljeni komunikaciji uporabnika in inteligentnega okolja. Uporablja se za nadzor in uporabljanje inteligentnega okolja in posameznih naprav in enot v takšnem okolju. Interakcija človek-stroj se izvaja z uporabo miške, tipkovnice, mobilnih naprav ali govora, dodatno pa je vgrajena podpora za implementacijo naprednejše multimodalne interakcije človeka in sistema.

Mobilna enota tipa 1 – predstavlja napredno mobilno robotsko enoto z srednjimi napajalnimi kapacitetami. Gradnik podpira zajemanje, predvajanje in pošiljanje avdio/video toka podatkov, procesorske zmogljivosti gradnika pa so v splošnem omejene (npr. samo razpoznavanje govora in procesiranje slike). Interakcija človek-stroj se izvaja z uporabo miške, tipkovnice, igralnih palic, mobilnih naprav ali govora, dodatno pa je vgrajena podpora za implementacijo naprednejše multimodalne interakcije človeka in sistema. Z enotami tipa 2 je možno upravljati tudi oddaljeno preko modulov oddaljenega klienta ali uporabniških vmesnikov postavljenih v inteligentnem okolju.

Mobilna enota tipa 2 - predstavlja enostavno mobilno enoto z nizkimi napajalnimi kapacitetami. V splošnem gradnik ne podpira zahtevnejšega procesiranja (npr. omogoča e le razpoznavanje govora). Enote tipa 1 podpirajo zajemanje in razpošiljanje avdio/video toka ostalim gradnikom, ki za enote opravljajo tudi procesiranje informacije. Interakcija človek-stroj se izvaja z uporabo miške, tipkovnice, igralnih palic, mobilnih naprav ali govora, dodatno pa je vgrajena podpora za implementacijo naprednejše multimodalne interakcije človeka in sistema. Z enotami tipa 2 je možno upravljati tudi oddaljeno preko modulov oddaljenega klienta ali uporabniških vmesnikov postavljenih v inteligentnem okolju. Splošni sistem upravljanja z informacijami/nadzorom je predstavljen na spodnji shemi. Inteligentno okolje v splošnem vsebuje različne naprave kot so IP kamere, senzorje, krmilnike za upravljanje z drugimi napravami glavni strežnik, enega ali več modularnih strežnikov, ki so

namenjeni pogonu različnih tipov mobilnih enot (opravljajo vlogo oddaljenih avdio, video in podatkovnih procesorskih enot), strežnikov procesiranja slike in signalov, podatkovnih strežnikov in enega ali več klientov, ki omogočajo naravno interakcijo človek-stroj.



Slika 9-2: Sistem inteligentnega okolja za upravljanje z informacijo/nadzorom.

V okviru takšnega sistema se lahko dogodki tvorijo sinhrono ali asinhrono. Scenariji, ki definirajo obnašanje sistemskih modulov in jih poganja strežniški modul se lahko zelo razlikujejo od scenarijev, ki jih poganjajo modularni strežniki in klienti. Različni uporabniki lahko z eno enoto komunicirajo simultano na povsem drugačen način. Inteligentno okolje poseduje tudi kognitivne zmožnosti in se je sposobno učiti o uporabi sistema različnih uporabnikov in informaciji, ki jo ti uporabniki potrebujejo. Sistem mora zato biti sposoben simultane poganjanja več uporabniških profilov in avtomatskega prilagajanja svojega obnašanja (različnih scenarijev) na vseh gradnikih sistema trenutnim potrebam posameznega uporabnika. Dogodki so v inteligentnih okoljih lahko poženi s strani uporabnika, sistema ali pa jih prožijo nepravilnosti v obnašanju sistema. Da se lahko zagotovi robustnost in neodvisnost sistema mora sistem

vsak tip dogodkov obravnavati na svoj način in jih učinkovito in fleksibilno procesirati. Dodatno robustnost sistema izboljšamo z učinkovitim mehanizmom povrnitve v primeru napake. Sistem se mora v primeru napake povrniti v osnovno stabilno stanje in nadaljevati z implementacijo predpisanega scenarija. Programsko ogrodje DATA je zasnovano v programskem okolju Java in predstavlja platformo neodvisno več uporabniško in več-storitveno naravnano programsko ogrodje. Scenariji delovanja se lahko avtomatsko prilagajajo trenutnim zahtevam in potrebam posameznega uporabnika. Programsko ogrodje DATA ne podpira direktne implementacije scenarijev v obliki programske kode. Takšen način implementacije se lahko kaj hitro sprevrže v nepremostljivo oviro v smislu nadzora, odpravljanja napak sistema in implementacije novih funkcionalnosti.

9.1.1 Sestava kompleksnega distribuiranega sistema z uporabo DATA ogrodja

Kot je prikazano na sliki *Sistemska arhitektura DATA* mora sistem znati obravnavati mnogo asinhronih in sinhronih dogodkov. Akcije in reakcije DATA sistema lahko prožijo uporabniki, sistem sam ali pa posamezni moduli sistema. Znotraj takšnega sistema morajo biti vsi dogodki, akcije in reakcije obravnavane na učinkovit in fleksibilen način. Dodatna prednost implementacije DATA sistema z uporabo Java tehnologij se kaže tudi v podpori številnih knjižnic. SAX predstavlja integrirane knjižnice za analizo in razčlenjevanje XML datotek, JMF je ključnega pomena za nadaljnjo procesiranje avdio/video in druge multimodalne informacije z uporabo JAI ("Java Advanced Imaging" – Java 2d in Java 3d tehnologije) in implementacijo RTP protokola za izmenjavo avdio/video informacije med posameznimi moduli DATA sistema. Dodatno pa Java okolje ponuja že dobre pogoje za odkrivanje in odpravljanje programskih napak, uporabljanje niti, upravljanje s sistemskimi viri, obravnavanje izjemnih stanj, itn. Za namene uporabe v kompleksnih okoljih smo implementirali sistem odvisnosti tipa Odjemalec/Strežnik. Dodatno pa smo morali pri implementaciji takšnega sistema biti pozorni na scenariško specifikiranje obnašanje sistema, ki bo omogočalo hkratno poganjanje različnih scenarijev in vgraditi osnovo za implementacijo modulov umetne inteligence. Robustnost in zmožnost povrnitve iz stanja anomalije je še ena izmed kvalitiet DATA programskega ogrodja. DATA sistem opravlja uporabniško specificiran nabor akcij in procesira specifičen nabor dogodkov. Z

večanjem števila uporabnikov sistema se povečuje nabor akcij in dogodkov, ki pa je omejen le z trenutnimi funkcionalnostmi DATA sistema. DATA sistem omogoča enostavno in fleksibilno vgradnjo novih modalnosti (brez direktnih odvisnosti med posameznimi moduli) in hkrati podpira učinkovit in fleksibilen način konfiguracije obnašanja posameznih modulov. Za specifičen scenarij uporabe DATA sistema, DATA klient, DATA strežnik in DATA modularni strežnik uporabljajo specifične podscenarije, ki upoštevajo pravila podprtih protokolov in predstavljeno arhitekturo odjemalec/strežnik.

Vsi moduli DATA sistema so na osnovi Java UniMod programskega ogrodja implementirani kot končni avtomati.

9.1.2 Java UniMod programsko ogrodje

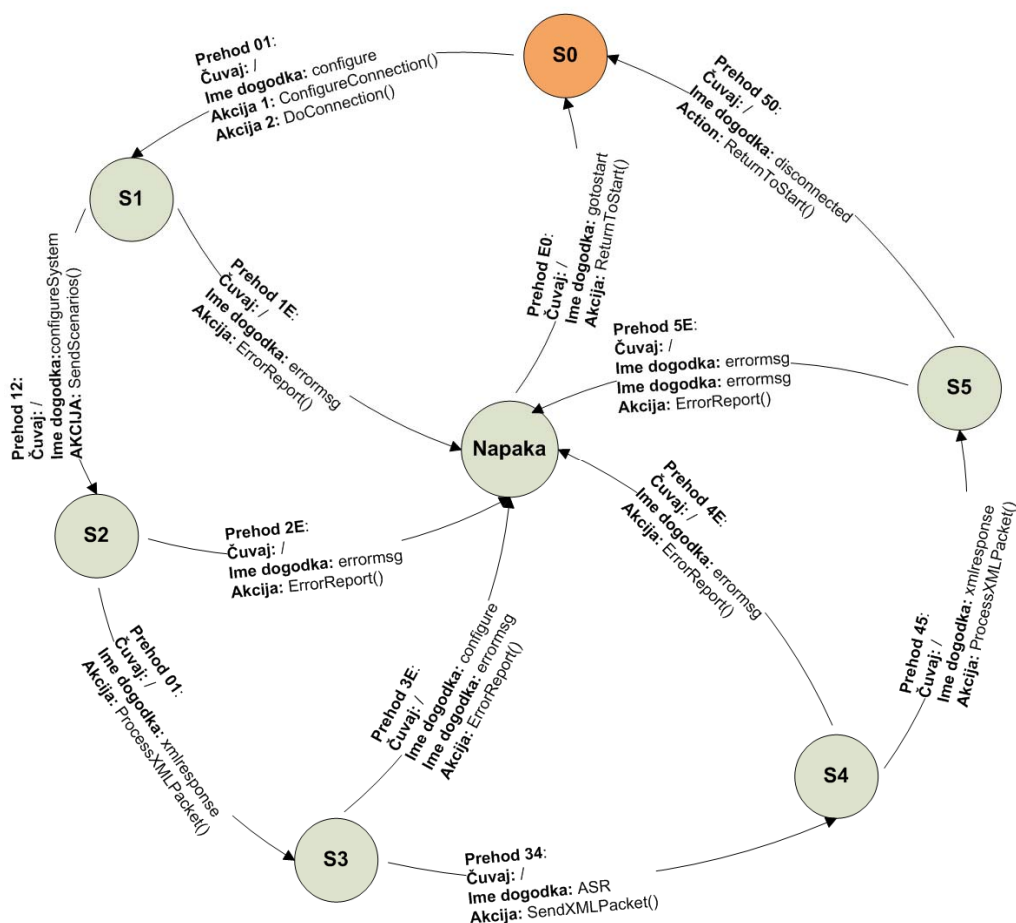
UniMod pomeni uniformno modeliranje. Dolgoročni cilj razvoja je ustvariti uniformno metodologijo za proces razvoja aplikacij, ki bo zmanjšala razmak med fazami zasnove in razvoja. Trenutno je UniMod fokusiran na zasnovo in implementacijo aplikacijskega obnašanja. Že implementirani pristop prilagaja SWITCH tehnologijo za UML notacije. SWITCH tehnologija je poznana tudi kot programiranje na osnovi avtomatov in predlaga modeliranje obnašanja sistema z uporabo FSM (strukturnega končnega avtomata). Strukturni FSM je definiran kot nabor abstraktnih končnih avtomatov in sheme povezovanja teh abstraktnih končnih avtomatov. Razvitih je veliko pripomočkov, ki uporabljajo FSM za opise programske logike, vendar pa so ali sistemsko orientirani (glede na sistem v katerega so postavljeni) ali pa ne podpirajo UML notacij. Kot kontrast se UniMod fokusira na MAD ("Model Driven Architecture" – modelsko usmerjena arhitektura), na UML in univerzalno procesiranje. UniMod za opisovanje systemskega obnašanja definira svojo metodologijo in ponuja nabor orodij, ki jih sestavljata tehnologija javanskih končnih avtomatov in dodatek za Eclipse. Orodja omogočajo zasnovo aplikacijske logike na podlagi razrednih diagramov in diagramov stanj, in generirajo XML opis končnega avtomata. XML opis nato zaženejo z uporabo ogrodja za zagon ("Runtime framework"), ki je del programskega paketa javanskih končnih avtomatov. Ogrodje za zagon omogoča tvorjenje interpreterjev modelov končnih avtomatov in definira naslednji princip interpretacije: interpretacija ob zagonu interpreterja, XML opis končnega avtomata se enkrat in popolnoma pretvori v model

končnega avtomata, ki se shrani v spomin; rezultirajoč sistem sestavljata okolje za zagon in prezentacija končnega avtomata shranjena v spominu; za obravnavo in streženje dogodkom sistem analizira naslednje stanje in vhodno akcijo in izbere prehod, ki ga bo sprožil; izbrani prehod sproži in izvede izhodne akcije predpisane v končnem avtomatu. UniMod omogoča uporabo principov predlagani v J2E samostojnih aplikacijah in J2EE spletnih aplikacijah, kjer vlogo interpreteja/končnega avtomata odigra JBoss strežnik. UniMod podpira tudi pristop generiranje programske kode v ciljnem jeziku. Stanski avtomat se pretvori v programsko kodo, ki nato ga implementira. Trenutno podprt programski jezik je samo Symbian C++. UniMod je primeren za izdelavo samostojnih Java GUI in konzolnih aplikacij, razvoju in izdelavi Java aplikacij tipa klient-strežnik, razvoju spletnih aplikacij, reševanju algoritmičnih problemov in izdelavo aplikacij na platformi Symbian. Programsko ogrodje javanskih končnih avtomatov vsebuje razrede, ki definirajo FSM meta modele in omogočajo manipulacijo modela, kompiliranje modela, validacijo in poganjanje modela. Programsko ogrodje ločuje trden FSM model – statično strukturo modela – in pravila interpretacije te strukture. Proces uporabe programskega ogrodja poteka v naslednjih korakih:

1. Izdelava FSM modela – faza zasnove
2. Izdelava razredov za ponudnike dogodkov in nadzorovane objekte – faza razvoja
3. Izdelava novega (ali re-konfiguracija obstoječega) FSM interpreterja – faza razvoja ali "deployment"
4. Z uporabo ustvarjenega interpreterja poženi FSM – faza življenja aplikacije

9.1.3 Izvedba scenarijev

Programsko ogrodje UniMod definira objekte za konstrukcijo in zaganja nje končnih avtomatov, vendar pa kot vhod potrebuje tudi opis končnega avtomata v specifičnem formatu. Za podajanje opisov končnih avtomatov se uporablja XML podatkovni format. Razvijalec lahko zunaj delovanja sistema napiše veliko različnih XML opisov končnih avtomatov ali pa jih generira kar v času delovanja DATA sistema. Ti opisi so nato, brez dodatne potrebe po vnovični kompilaciji kode ali spremembah v kodi, uporabljeni za poganjanje posameznih modulov DATA sistema. S takšnim pristopom eliminiramo nenehno kompilacijo in spreminjanje kode.



Slika 9-3: Enostaven primer scenarija uporabljenega za pogon DATA odjemalca.

Na zgornji shemi se nahaja sistemski opis modula DATA sistema, predstavljen s pomočjo diagrama prehajanja stanj. Z uporabo takšnih diagramov lahko popolnoma opišemo delovanje kateregakoli modula DATA sistema. Iz sheme je razvidno da takšen diagram sestavljajo stanska vozlišča povezana z enim ali več prehodnimi loki. Vsak prehod je opisan z unikatnim imenom, funkcijo paznika (pazi da se prehod izvede samo pod določenimi pogoji), imenom dogodka, ki stimulira prehod, vir in cilj prehoda, in eno ali več akcij, ki naj se izvedejo ob prehodu v naslednje stanje. Dogodki, ki lahko stimulirajo prehode v modulih DATA sistema so npr. prejeti in poslani paketi različnih protokolov, dogodki grafičnega vmesnika, dogodki krmilnih naprav, alarmna sporočila, ki jih generira DATA sistem itn. Moduli DATA sistem so torej stroji končnih stanj, ki jih poganjajo dogodki v sistemu. Takšna implementacije je zelo primerna za uporabo v komunikacijski arhitekturi klient/strežnik in implementacijo učinkovite in fleksibilne

komunikacije znotraj inteligentnega okolja. Diagram iz zgornje sheme moramo nato pretvoriti v zapis XML formata. Direktni zapis obnašanja sistema v XML formatu, je tudi z uporabo diagrama prehajanja stanj težak, zato raje najprej obnašanje opišemo v osnovnem XML formatu, ki ga nato pretvorimo v UniMod XML format, ki se uporablja za opise obnašanja modulov znotraj DATA sistema.

Za pospeševanje generiranja vseh opisov obnašanja modulov znotraj DATA sistema, se je razvilo javansko orodje DATAProtocol. Sestavljeno je iz JavaCC in Java kompilatorja. JavaCC kompilator se uporablja za implementacijo Java razčlenjevalnika, ki avtomatsko pretvori zapis v osnovnem XML formatu v opis scenarija v UniMod XML formatu. Razvijalec mora opisati scenarij v osnovnem XML formatu, nato pa ga orodje DATAProtocol avtomatsko pretvori v UniMod XML format. Med pretvorbo orodje še avtomatsko preveri UniMod XML opise systemskega obnašanja in razvijalca obvesti o napakah ali nekonsistentnosti v XML opisih. Razvijalec mora XML opis tako dolgo popravljati, dokler mu orodje ne javlja napak. Uporabniki lahko nato izbirajo med ponujenimi scenariji in DATA sistem uporabljajo po svojih specifikacijah. Na novo generirani XML opisi se v DATA sistem vgradijo tako, da se XML datoteke prenesejo v točno določeno mapo. Tako lahko sistem avtomatsko ali na izbiro uporabnika vnese in pretvori nov XML opis in izvaja v opisu specificirano obnašanje. Dodatno lahko vsak uporabnik prilagodi delovanje celotnega DATA sistema svojim potrebam. Med delovanjem lahko uporabnik pošlje XML opise obnašanja DATA strežniku ali DATA modularnim strežnikom in tako predpiše želeno obnašanje. Dodatno lahko več uporabnikov, ki uporablja različne DATA kliente DATA strežniku hkrati pošlje več različnih XML opisov, DATA strežnik pa bo simultano pognal te opise in prilagodil svoje obnašanje vsakemu uporabniku.

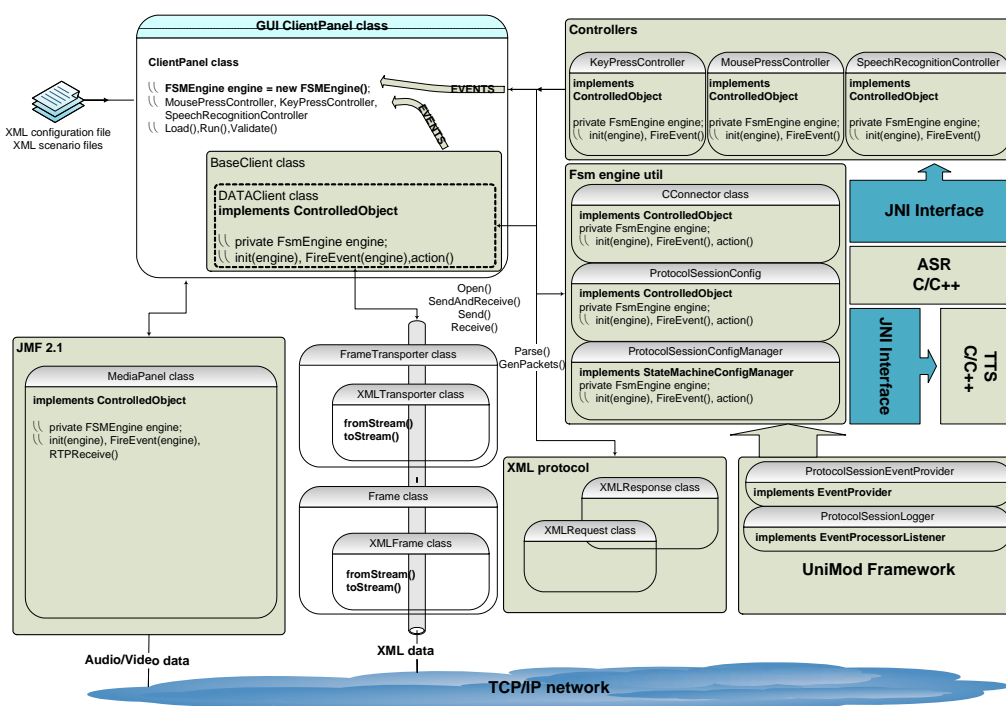
9.1.4 Gradniki sistema »DATA«

Kot je bilo že omenjeno v opisu arhitekture DATA sistema, so gradniki sistema naslednji:

- DATA odjemalec,
- DATA strežnik,
- DATA modularni strežnik.

DATA odjemalec

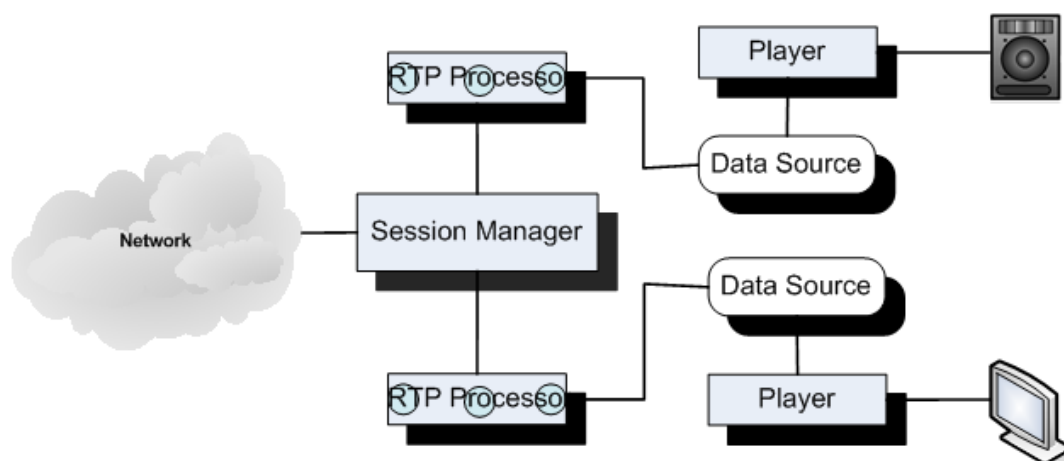
DATA odjemalec sestavlja več Java paketov, ki implementirajo različne naloge sistema. DATA odjemalec za pogon končnega avtomata in implementacijo XML opisa ustvari nit. Protokol uporabljen za komunikacijo med DATA odjemalcem in DATA strežnikom je baziran na osnovnem XML protokolu. XML paketi se med posameznimi moduli izmenjujejo preko TCP/IP protokola. Za kreacijo in razčlenitev teh paketov se ustvari XML paket. Za komunikacijo s strežniškim delom se implementira poseben razred t.i. *BaseClient*.



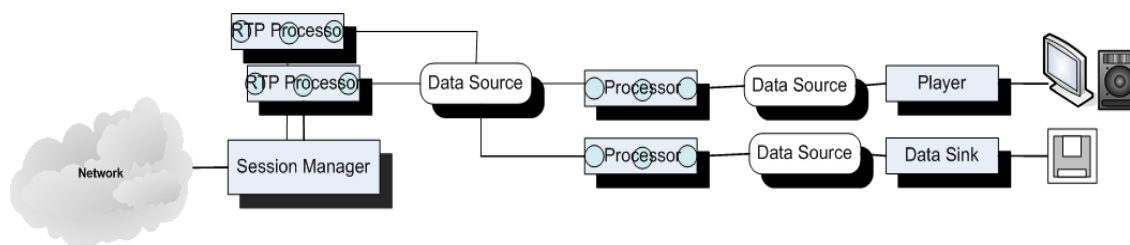
Slika 9-4: Funkcionalna in logična shema DATA odjemalca.

Razred implementira generalne metode za komunikacijo tipa odjemalec/ponudnik na osnovi vtičnic. Razširjeni razred t.i. *DATAClient* (razširja razred *BaseClient*) uporablja pri komunikaciji drugačne transportne mehanizme kot "splošni" klient. Vmesniški razred *FrameTransporter* le definira metode za izmenjavo paketov med odjemalcem in ponudnikom (klient in strežnik). Vmesniški razred uporablja razred *XMLFrame*, ki se uporablja transport paketov med ponudnikom in odjemalcem. Tudi XML paketi so specifični, zato razred *XMLFrame* predstavlja razširjeno obliko *Frame* razreda. Razred

DATAClient poskrbi za vzpostavitev povezave s strežnikom takoj ko uporabnik zažene modul DATA odjemalec. Komunikacija med DATA odjemalcem in DATA strežnikom ostane aktivna dokler končni avtomat ne doseže končnega stanja. Vsak DATA odjemalec zaključi s svojo operacijo, ko pridobi rezultate s strani DATA sistema. Na spodnji shemi je predstavljena logična in funkcionalna zasnova modula DATA odjemalec. V okviru RTP izmenjave podatkov DATA odjemalec predstavlja prejemnika RTP tokov (sprejema avdio/video informacijo v obliki RTP tokov). DATA klient zato vsebuje *rtp* paket, ki vsebuje objekte in metode potrebne za implementacijo RTP predvajalnika. Mehanizem sprejemanja avdio/video RTP toka je predstavljen na spodnji shemi.



Slika 9-5: DATA odjemalec: sprejemanje RTP tokov in ločeno predvajanje.



Slika 9-6: DATA odjemalec: sprejemanje RTP tokov in združeno predvajanje.

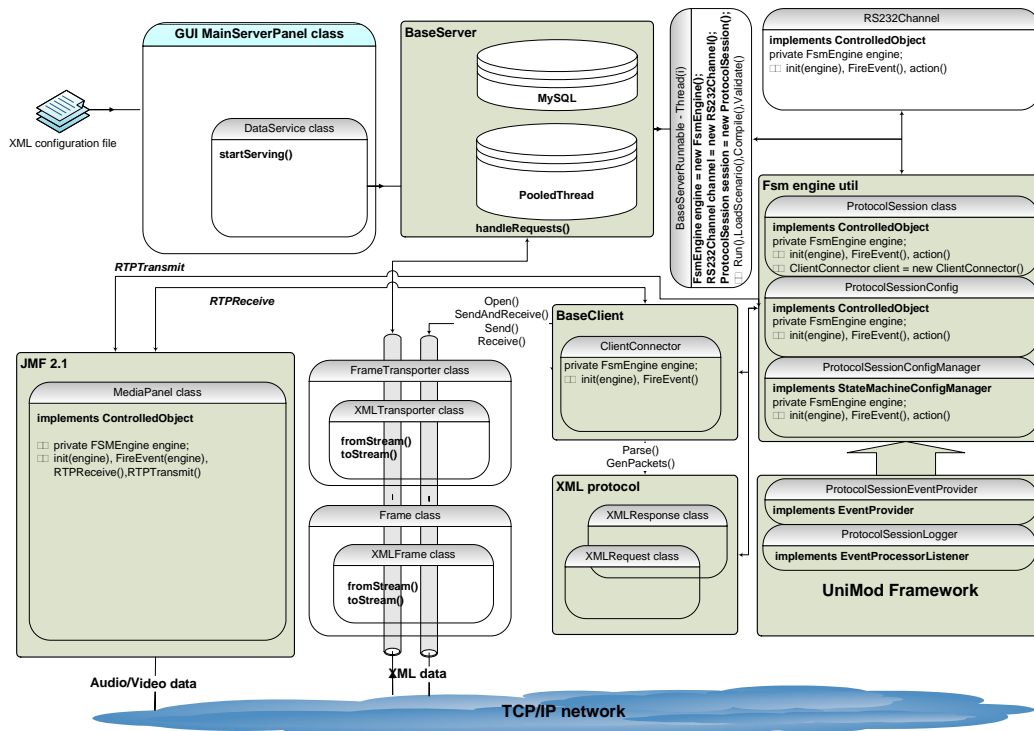
Kot prikazujeta gornji shemi, obstajata dva različna načina predvajanja prejetih rtp tokov. Prvi način predvideva ločeno predvajanje posameznih tokov. Takšna

implementacija rezultira v implementaciji dveh različnih podatkovnih virov (ločena video in avdio podatkovni vir) in dveh ločenih RTP predvajalnikov. RTP predvajalnik lahko do posameznega podatkovnega vira dostopa direktno, vendar pa se navadno uporablja metoda podvojevanja vira. Tako osnovni vir ostane nedotaknjen in se lahko nadalje uporablja, saj vsak podatkovni vir v primeru dostopanja rtp procesorja ali rtp predvajalnika postane nedostopen. Drugi način predvajanja predvideva združevanje tokov v enojni vir. Takšna implementacija za predvajanje zahteva le en RTP predvajalnik. Na shemi združenega predvajanja lahko opazimo nadaljnje deljenje vira. Če želimo podatkovni vir kodirati to opravimo z RTP procesorjem. Ki si ustvari podvojen podatkovni vir in za namene predvajanja oziroma shranjevanja v datoteko ustvari novo kodiran podatkovni vir. Do njega nato dostopamo z uporabo podatkovnega vira ali "Data Sink" objekta, odvisno od operacije ki jo želimo nad virom opravljati. "Data Sink" objekt se uporablja izključno kadar želimo rtp tok shraniti v datoteko (snemanje).

DATA strežnik

Modul DATA strežnik gradi več Java paketov, ki implementirajo različne operacije. Strežniška aplikacija mora ponujati funkcionalnosti kot so: končni avtomat, prehodi med stanji avtomata in upravljanje s spominom. Vse naloge znotraj modula poganja FSM stroj in pripadajoči komunikacijski protokol. Organizacija strežnika se razlikuje od organizacije klienta, saj strežnik vsebuje t.i. "PooledThread" objekt (objekt z naborom niti). Objekt se instancira ob inicializaciji, ko se inicializira vnaprej določeno število niti, ki se uporabljajo za streženje odjemalcem (klientom). Ko DATA klient proži klic se iz nabora niti izbere ni, ki je trenutno na voljo. Nit se za DATA klienta rezervira za celoten čas komunikacije kot rezervirana XML seja. V primeru da v naboru niti ni več proste niti, strežnik novo nastalo zahtevo po povezavi zavrne, DATA klient pa mora počakati da kakšna izmed niti postane prosta (izgubi status rezerviranosti). Protokol, ki se uporablja za izmenjavo podatkov preko rezerviranih sej je osnovni XML protokol za prenos XML paketov pa se uporablja TCP/IP protokol. Tudi DATA strežniški modul vsebuje XML paket, ki omogoča generiranje in razčlenjevanje XML paketov. Trenutno nastavljeno število niti v naboru niti je 200, nastavitev pa se lahko spremeni preko konfiguracijskih nastavitev. Za namene komunikacije DATA strežnik \leftrightarrow DATA

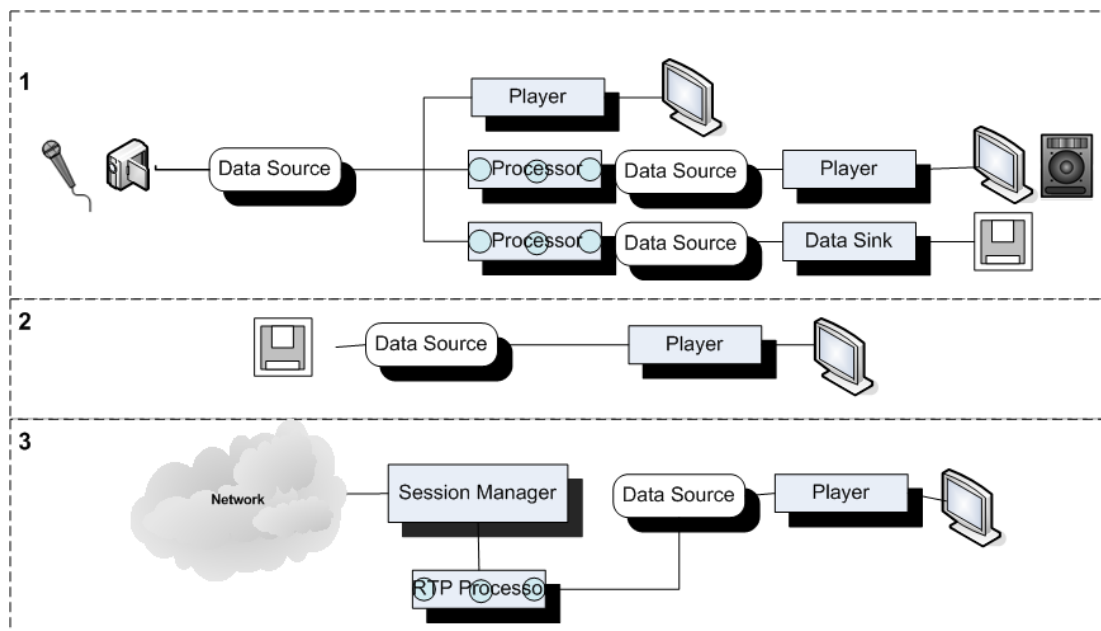
klient se implementira razred *BaseServer*, ki vsebuje vse splošne metode za obravnavanje povezav preko vtičnic. Dodaten razred *DATA Server* (razširitev razreda *BaseServer*) se implementira saj DATA strežniki uporabljajo drugačne transportne mehanizme kot "splošni" strežniki. Vmesniški razred uporablja razred *XMLFrame*, ki se uporablja transport paketov med ponudnikom in odjemalcem. Tudi XML paketi so specifični, zato razred *XMLFrame* predstavlja razširjeno obliko *Frame* razreda. Podrobna funkcionalna struktura DATA strežnika je predstavljena na spodnji shemi.



Slika 9-7: Funkcionalna in logična shema DATA strežnika.

Ko se povezava med DATA strežnikom in DATA odjemalcem vzpostavi, strežnik začne sprejemati vhodno informacijo poslano s strani DATA klienta in se po potrebi lahko poveže na enega ali več DATA modularnih strežnikov. Po vzpostavljeni povezavi se prenašajo tudi XML opisi scenarijev. Ti opisi se uporabljajo za namestitev in prilagoditev DATA sistema uporabniškim željam in zahtevam. Komunikacija med DATA strežnikom in DATA klientom poteka dokler končni avtomat ne preide v zaključno stanje. V tem stanju se komunikacija prekine, nit rezervirane XML seje pa se ubije (nit postane prosta). DATA strežniški modul v DATA sistemu igra vlogo RTP prejemnika in oddajnika ter modula ki je zmožen RTP procesiranja lokalnih ali oddaljenih RTP tokov. Vse te vloge se lahko uporabljajo vsaka zase ali združeno v eni

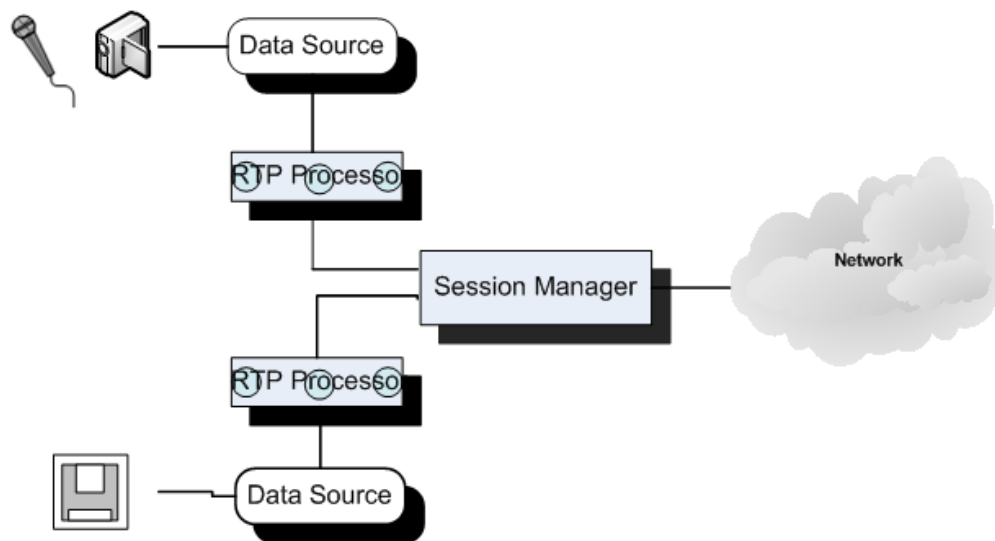
funkcionalnosti (npr. zajemanje → oddajanje → sprejemanje → procesiranje → ponovno oddajanje).



Slika 9-8: Načini predvajanja RTP toka na DATA strežniku.

Zgornja shema prikazuje tri načine prikazovanja medijskega vira. Načini predstavljanja podatkovnega toka se razlikujejo predvsem v načinu zajemanja informacije. Prvi način predstavlja prikazovanje lokalno zajete multimedijske informacije. Enota za zajemanje informacije je lahko hibrid (kot na zgornji sliki) ali pa zna zajemati posamezen tip informacije. Iz zajetega toka ustvarimo primarni podatkovni vir. Vsak RTP procesor in predvajalnik primarni podatkovni vir duplicira in procesira/predvaja. Z uporabo "Data Sink" objekta pa lahko zajeto informacijo tudi shranjujemo v datoteko (snemanje). Prvi način prikaza informacije prikazuje pomembno lastnost JMF RTP modula in sicer možnost sočasnega operiranja nad primarnim podatkovnim virom na različne načine. Drugi način predstavljanja zajemanje medijske informacije iz datoteke (.avi, .mov, .mp3, itn.). V tem načinu tvorimo primarni podatkovni vir s katerim lahko poljubno manipuliramo. Na shemi je prikazan direktno manipulacijo podatkovnega vira (brez dupliciranja), kar posledično pomeni da se podatkovni vir zaklene za določen RTP predvajalnik. Pomembno pri zajemanju multimedijskega vira iz datoteke je dejstvo, da ni potrebno obravnavati celotne datoteke naenkrat, marveč se datoteka obravnava segmentirano (po video okvirjih), s čimer se

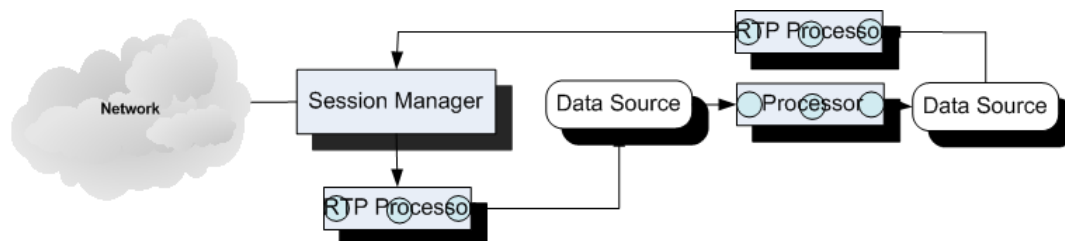
zmanjšajo zahteve po sistemskih virih. Tretji način predvajanja avdio/video tokov pa predstavlja predvajanje tokov zajetih iz omrežja. V primeru da neka modularnost DATA sistema DATA strežniku pošilja avdio/video tok lahko strežnik ta tok procesira in vrne rezultat oziroma rezultirajoč nov podatkovni tok in/ali prejeti tok predvaja na svojem vmesniku. Objekt sejnega upravljanja ("Session Manager") skrbi za pravilno in učinkovito razvrščanje posameznih multimedijskih sej in skupaj z RTP procesorjem implementira proces de-multipleksiranja toka. Objekt sejnega upravljanja v osnovi ločuje avdio in video tokove, ki se pošiljajo na različna vrata. DATA strežnik v okviru RTP programskega paketa implementira tudi razpošiljanje multimedijskih tokov ("streaming"). Primer razpošiljanja multimedijske informacije je prikazan na spodnji shemi.



Slika 9-9:Razpošiljanje multimedijskega toka z uporabo RTP protokola.

Shema prikazuje možnost združevanja in hkratnega pošiljanja več tipov multimedijskih tokov. RTP procesor in sejni upravljelec skupaj tvorita RTP multiplekser in zajeti podatkovni vir v obliki multimedijskih tokov s pomočjo RTP multimedijskih tokov proti točno določenem cilju ali vsem poslušalcem v omrežju DATA sistema. V primeru "broadband" načina razpošiljanja morajo posamezni moduli DATA sistema prej vzpostaviti RTP sejo. Za upravljanje s sejami skrbi objekt "Session Manager", ki s svojo implementacijo omogoča vzpostavitev več neodvisnih RTP sej med posameznimi moduli DATA sistema, ki vsebujejo RTP programski paket. RTP procesor pa ne služi le procesom multipleksiranja, de-multipleksiranja in kodiranja

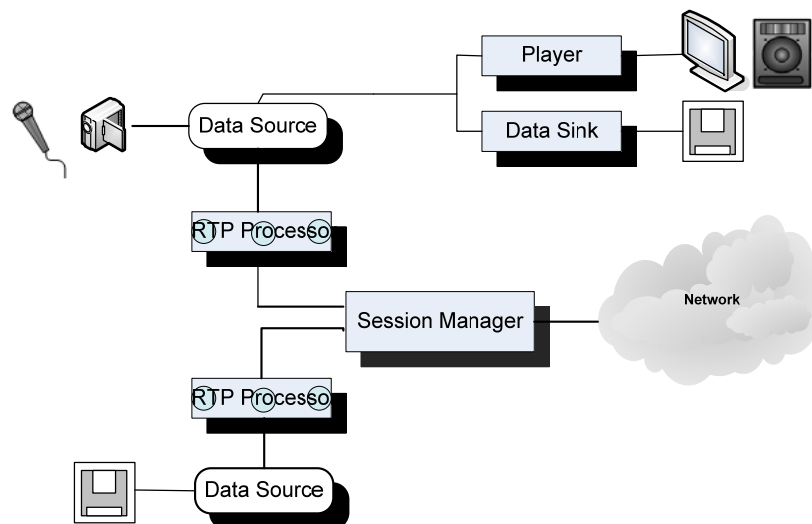
multimedijskih tokov marveč lahko opravljala tudi naloge procesiranja avdio in video podatkov v obliki razpoznavanja in procesiranja govora (ASR), procesiranja slike, procesiranja tekočega video toka itn. DATA strežnik lahko poleg lokalnih zajetij procesira tudi avdio/video informacije posredovane s strani modulov RTP sistema. Takšno procesiranje v okviru DATA sistema imenujemo post-procesiranje. Primer post-procesiranja je predstavljen na spodnji shemi.



Slika 9-10: Post-procesiranje oddaljenega RTP toka.

V primeru da prejeti RTP tok potrebno le post procesirati (npr.: procesiranja slike ni moč opraviti na mobilnih enotah tipa 1, nekateri DATA modularni strežniki ne morejo opraviti filtracije, ...) se za namene procesiranja uporabi objekt RTP procesor. DATA sistem rezervira določene številke vrat in jih nameni za vhodne in izhodne podatkovne toke post-procesiranja (2000-5000 vrata vhodnih RTP tokov, ki jih je potrebno post-procesirati; 5000-6000 vrata RTP tokov nad katerimi je potrebno izvesti obdelavo slike, itn.). Ker odziv post-procesiranja vedno ne rezultira v izhodnem RTP toku (razpoznavanje oseb, kjer se odziv poda v obliki imena ali profila uporabnika), lahko ob zajetju, ki ne zahteva odziva v obliki RTP toka, vrata, ki jih je uporabljal vhodni RTP tok sprostimo. Kot prikazano na gornji shemi, DATA strežnik preko vrat rezerviranih za post procesiranje vhodne toke prejema, de-multipleksira na identičen način kot je implementirano predvajanje multimedijskih tokov prejetih iz omrežja. Namesto predvajanja vsebine pa se nad multimedijsko vsebino opravi želeno procesiranje, specificirano v XML opisu obnašanja. V primeru da ima odziv obliko multimedijskega toka podatkov, se ga ciljnemu modulu posreduje na identičen način kot pri posredovanju zajetih avdio/video tokov.

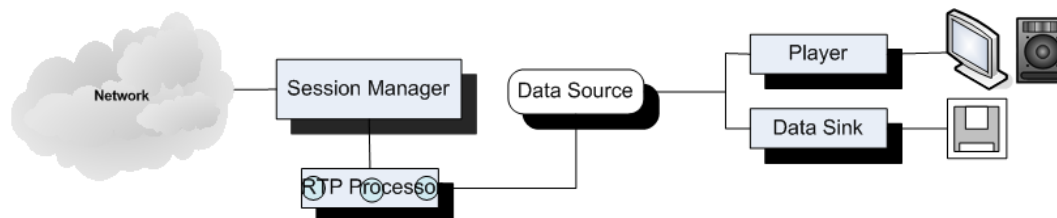
nastavitev pa se lahko spremeni preko konfiguracijskih nastavitev. Za namene komunikacije DATA strežnik \leftrightarrow DATA modularni strežnik se implementira razred *BaseServer*, ki vsebuje vse splošne metode za obravnavanje povezav preko vtičnic. Dodaten razred *DATAService* (razširitev razreda *BaseServer*) se implementira saj DATA strežniki uporabljajo drugačne transportne mehanizme kot "splošni" strežniki. Vmesniški razred uporablja razred *XMLFrame*, ki se uporablja transport paketov med ponudnikom in odjemalcem. Tudi XML paketi so specifični, zato razred *XMLFrame* predstavlja razširjeno obliko *Frame* razreda. Funkcionalna struktura DATA modularnega strežnika je predstavljena na spodnji shemi. Tudi DATA modularni strežnik podpira RTP protokol in zato vsebuje RTP paket. Implementacija RTP modula DATA modularnega strežnika je podobna Implementacijam rtp paketa DATA strežnika in DATA klienta. Glavna razlika med RTP modulom DATA modularnega strežnika in DATA strežnika je omejenost zmožnosti obdelave avdio/video informacije DATA modularnega strežnika. DATA modularni strežnik opravlja je osnovna procesiranja.



Slika 9-12: Shema procesa zajemi \rightarrow predvajaj/razpošlji.

Kot prikazuje zgornja shema, so nekatere RTP operacije DATA modularnega strežnika direktno dedovane iz implementacije RTP operacij na DATA strežniku. Zajemanje avdio/video informacije lahko opravljamo s pomočjo kamere in mikrofona (lahko hibridna naprava) ali pa avdio/video informacijo zajemamo iz datoteke. Iz zajeta tvorimo podatkovni vir (objekt "Data Source"), ki ga lahko predvajamo lokalno s pomočjo RTP predvajalnika ali pa s pomočjo objekta "Session Manager" informacijo

razpošljemo v obliki RTP tokov. V primeru sočasnega opravljanja operacije predvajanja in razpošiljanja, vsaka izmed operacij klonira osnovni podatkovni vir. Tako se osnovni podatkovni vir ohranja nedotaknjen in zmožen uporabe v morebitnih nadaljnji procesih. Duplicirani in originalni "Data Source" objekt skupaj tvorijo virtualni podatkovni vir, ki je odvisen od stanja originalnega podatkovnega vira. V primeru da uničimo originalni podatkovni vir uničimo tudi vse klonirane (duplicirane) podatkovne vire.



Slika 9-13: Shema procesa prejmi → predvajaj/shrani.

Podroben opis procesov predvajanja in shranjevanja je podan v poglavjih o DATA strežniku in DATA klientu. DATA modularni strežnik deduje procese predvajanja. Podobno se zaradi ohranjanja nedotaknjenosti osnovnega podatkovnega vira, le ta klonira. Shema prikazuje predvajanje in shranjevanje avdio/video informacije v datoteko. Tudi na modularnem strežniku je mogoče ločeno predvajati/procesirati avdio in video tokove, ki se prejemajo iz omrežja in de-multipleksirajo ter primerno kodirajo (avdio, video kodeki) s pomočjo objektov "Session Manager" in "RTP procesor".

9.2 Konfiguriranje in profiliranje kognitivne mobilne platforme »DATA«

Kadar uporabljamo sistem DATA, lahko pričakujemo kakršnokoli število uporabnikov, ki želijo upravljati z napravami v inteligentnem okolju preko avdio, video in drugih komunikacijskih kanalov. Posledično lahko pričakujemo tudi veliko število scenarijev uporabe sistema, ki tudi definirajo zahtevano obnašanje posameznih modulov sistema DATA, za katere lahko pričakujemo tudi njihovo sočasno izvajanje. Uporabniki za definiranje zelenega obnašanja sistemskih modulov, običajno izkoriščajo oddaljeno upravljanje sistema in sicer s pomočjo prenosa XML opisov, po osnovnem XML protokolu. Hkrati lahko pričakujemo tudi uporabo avdio/video prenosa podatkov in procesiranja le teh. Iz zgoraj naštetih razlogov, smo v okviru diplomskega dela nekatere

teoretske študije in predvidevanja, preizkusili tudi na praktičnih primerih in implementirali naslednje:

- Implementacija osnovnih funkcionalnosti mobilne robotske enote,
- implementacija porazdeljenega programskega ogrodja DATA,
- Implementacija osnovnih funkcionalnosti RTP sistema.

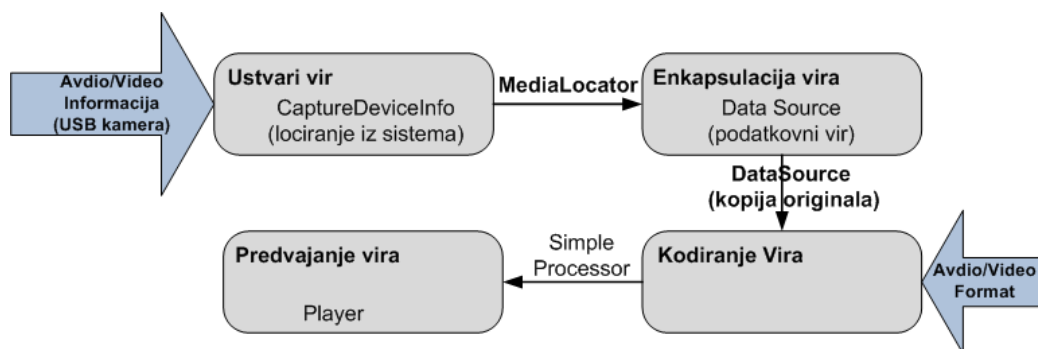
9.2.1 *Specifikacije za distribuirano programsko ogrodje »DATA«*

Kot smo že omenili, programsko ogrodje DATA, predstavlja modularen in enostavno nadgradljiv programski paket, ki omogoča distribuirano, porazdeljeno delovanje sistema kot celote in ponuja učinkovito in zanesljivo izvajanje kompleksnih kombinacij protokolov, na katerih gradimo komunikacijo med posameznimi moduli sistema DATA. Za lažje razumevanje bomo v nadaljevanju izpostavili tri glavne programske gradnike programskega ogrodja DATA. *DATA strežnik* – konfigurabilen programski paket, ki ga poganja FSM stroj. V sistemu DATA se predstavlja kot centralni gradnik, ki lahko upravlja s katerikoli DATA modularnim strežnikom in hkrati predstavlja komunikacijsko točko med ostalimi programskimi gradniki v sistemu DATA (vzpostavitev povezav med posameznimi gradniki vedno poteka preko DATA strežnika). DATA strežnik vsebuje sistem za upravljanje z viri in je v tem kontekstu primarni modul, ki odloča o uspešnosti nadaljne povezave. V primeru, da dva uporabnika želita hkrati krmiliti isto mobilno robotsko enoto, se DATA strežnik na podlagi prioritete odloči, kateremu uporabniku bo dovolil takšno poseganje v obnašanje mobilne robotske enote in tudi kako bo uskladi upravljanje oz. komunikacijo z obema uporabnikoma hkrati. Delovanje DATA strežnika lahko implementiramo v obliki porazdeljenega delovanja na več strežnikih. V takšni konfiguraciji lahko DATA strežnik služi kot močno orodje za procesiranje multimodalne informacije v realnem času. *DATA modularni strežnik* – predstavlja programski paket namenjen za implementacijo na različnih mobilnih enotah (ali drugih podsistemih) inteligentnega okolja. Njegova zasnova je podobna zasnovi DATA strežnika, vendar pa se DATA strežnik sam ne more povezovati na druge komponente v okolju. Njegovo obnašanje je konfigurabilno z uporabo XML opisov obnašanja, ki jih prejme od DATA odjemalca oz. posredno od DATA strežnika. DATA strežnik implementira vse potrebne metode za implementacijo

komunikacije med programskim in strojnim delom sistema DATA (serijska/USB povezava in BlueTooth povezava). Lokalna ali oddaljena komunikacija s posameznimi komponentami inteligentnega okolja (ki ne podpirajo TCP/IP povezave), poteka preko uporabe DATA modularnega strežnika. *DATA odjemalec* – predstavlja programski paket namenjen multimodalni interakciji človek-stroj. Podpira naravne načine interakcije (govorna, vizualna, ali z uporabo igralne palice ali ploščka, tipkovnice ali s pritiski na zaslonu – "Touch screen"). Implementacija interakcije je odvisna od zmožnosti, ki jih poseduje sama enota in pa tudi od uporabniških zahtev/potreb. Vsak DATA odjemalec lahko sočasno podpira tudi več različnih načinov interakcije človek-stroj.

10 PROGRAMSKO OGRODJE ZA ZAJEMANJE IN PRENOS AVDIO/VIDEO INFORMACIJE

Kot smo že omenili, DATA sistem na vsakem izmed posameznih programskih gradnikov implementira RTP paket. Implementacija je vedno odvisna od tipa gradnika, funkcionalnosti RTP paketa pa od uporabniškega scenarija. Zajemanje avdio/video informacije poteka v realnem modelu okolja na štiri načine: z uporabo RTP kamere, z uporabo IP kamere, iz datoteke ali s sprejemanjem RTP tokov iz omrežja DATA sistema (LAN ali WAN oz. internet).

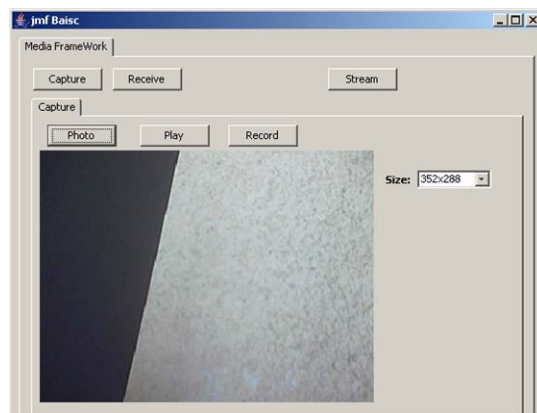


Slika 10-1: Funkcionalna shema procesa zajemanja informacije iz USB kamere.

Kot prikazuje zgornja shema lahko z uporabo RTP paketa implementiranega v okviru DATA sistema informacijo zajemamo iz USB kamere. Ob inicializaciji procesa zajemanja moramo najprej locirati napravo s katere želimo zajemati. JMF trenutno podpira le eno napravo za zajemanje (ob priklopu na USB). Lociranje storimo z uporabo klica metode *CaptureDeviceManager.getDeviceList*, ki nam v obliki objekta *Vector* vrne vse naprave, ki jih imamo v registru priklopljenih naprav in podpirajo določen video format (RGB ali YUV). Ker JMF podpira priklop le ene naprave objekt *MediaLocator* tvorimo s pomočjo prve naprave v polju priključenih naprav (polje hkrati vsebuje le eno napravo). S pomočjo *MediaLocatorja* tvorimo *DataSource* objekt. *DataSource* objekt predstavlja avdio/video informacijo ali kombinacijo obeh. Ko določimo lokacijo ali protokol vira, metode objekta *DataSource* enkapsulirajo vse podane vire in jih združijo. Obstajata dva tipa *DataSource* objekta:

- "Push DataSource" – razpošiljanje avdio/video informacije vsem ali video na zahtevo. Strežnik inicializira prenos podatkov in tok podatkov vira nadzoruje.
- "Pull DataSource" – razpošiljanje avdio/video informacije v obliki http ali FILE.

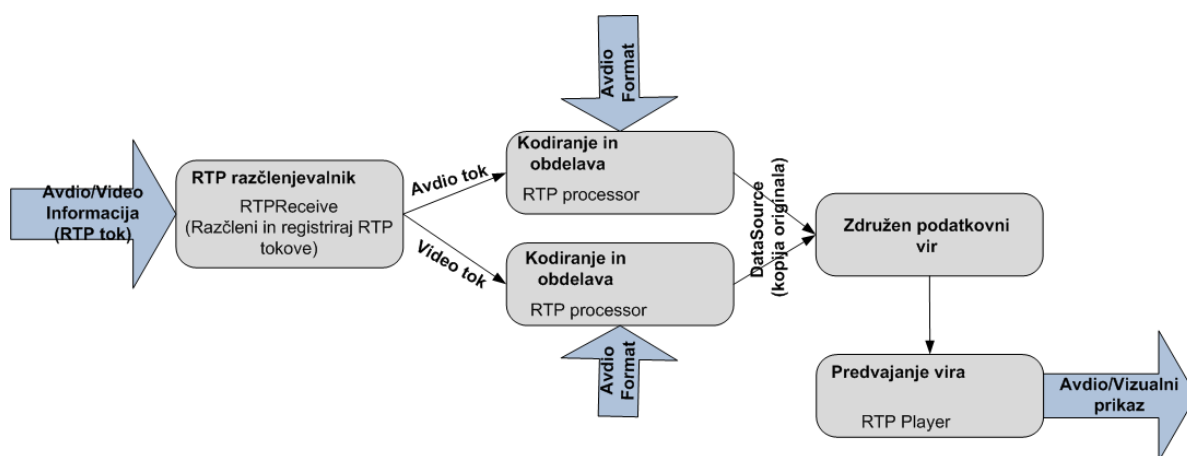
Odjemalec inicializira prenos podatkov in tok podatkov vira nadzoruje *DataSource* objekt tvorimo s pomočjo objekta *Manager*. Ustvarjen vir podatkov pred direktno uporabo še kloniramo (kopiramo), tako da pri manipulaciji vira puščamo originalni podatkovni vir nedotaknjen. Za pravilno predvajanje vira je potrebno podatkovni vir še zakodirati. V obliki *Format* objekta RTP procesorju (*SimpleProcessor*) podamo informacijo o želeni avdio in/ali video kodiranja predvajane informacije ob njegovi inicializaciji. Inicializiranemu procesorju ukažemo poslušanje podanega *DataSource* objekta nastavimo opis vsebine ("*Content Descriptor*") in ga zaženemo. S pomočjo enostavnega procesorja, ki ga ustvarimo v procesu kodiranja vira ustvarimo predvajalnik (objekt "*Player*"). Klic metode *Manager.createRealizedPlayer* ustvarimo nosilni predvajalnik, ki nosi avdio/video vsebino. Za samo vizualizacijo pa se moramo lotiti še problema lokalizacije. Avdio video informacijo zato preusmerimo na točno določeno mesto in končni predvajalnik inicializiramo. Spodnja slika prikazuje rezultat zgoraj opisanega postopka zajemanja avdio/video informacije, tvorjenja predvajalnika in prikaza zajete informacije z uporabo uporabniškega vmesnika. Vmesnik je namenjen le prikazu delovanja in sistema in ne predstavlja končne oblike.



Slika 10-2: Samostojni grafični vmesnik RTP programskega paketa v sistemu DATA.

Uporaba grafičnega vmesnika prikazanega na sliki 10-2, je preprosta. S klikom na gumb "Capture", začnemo z zajemanjem videa iz USB naprave za zajemanje. Ko se

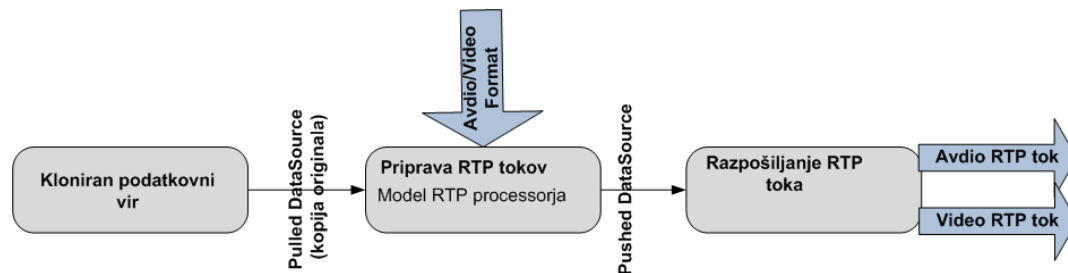
zaključi proces inicializacije in imamo pripravljeni tako podatkovni vir, procesor in predvajalnik, se odpre zavihek "Capture". Zavihek nam ponuja možnosti, s katerimi lahko upravljamo s trenutnim podatkovnim virom. Iz gornje slike je razvidno, da lahko spreminjamo razločljivost podatkovnega vira (meni "Size"), zajemamo sliko (gumb "Photo"), predvajamo video signal zajet s kamero (gumb "Play") in shranjujemo video signal v datoteko (gumb "Record"). V primeru izbire slikanja ali shranjevanja video podatkov v datoteko, se nam odpre novi meni, ki nas povpraša po imenu datoteke, obliki kodiranja in velikosti slike. Shranjevanje avdio/video podatkov ustvari nov kloniran podatkovni vir, nastavi izbrano kodiranje in velikost, namesto objekta pa predvajalnik odpre objekt *DataSink*. Objekt *DataSink* pripravi glavo datoteke izbranega tipa in datoteko ustvari, odpre, zaklene za svojo instanco in v izbrano datoteko prične zapisovati informacijo v obliki podatkovnega toka (tok zlogov z informacijo o sliki). Ko s procesom zajemanja informacije zaključimo, se kloniran podatkovni vir uniči in tudi sama instanca zajemanja. S tem uspešno sprostimo sistemske vire in preprečimo nepotrebno zasedanje procesorja in spomina. Na grafičnem vmesniku je predstavljen še gumb za prejemanja RTP informacije v obliki RTP toka (gumb "Receive"). S pritiskom na gumb začnemo prejemati avdio/ video podatke iz LAN ali WAN omrežja v obliki RTP tokov podatkov. Podrobnejši prikaz procesa zajemanja avdio/video informacije je predstavljen na spodnji sliki.



Slika 10-3: Funkcionalna shema procesa zajemanja informacije RTP toka.

Ob inicializaciji instance RTP zajemanja, ustvarimo dve UDP vtičnici, preko katerih sprejemamo avdio in video informacijo iz omrežja. Vsak RTP tok, ki je instanciran na

modulu, pridobi dve vtičnici (soda in liha vrata so namenjena prejemanju avdio in video informacije). Prejeta informacijo posredujemo RTP razčlenjevalniku, ki predstavlja demultiplekser avdio in video tokov in hkrati dekodirnik RTP vsebine. RTP razčlenjevalnik ne samo loči avdio in video informacijo, marveč tudi dekodira posamezne avdio/video RTP pakete in tvori po en bitni tok podatkov za posamezno RTP sejo, detektirano na paru vtičnic. Poslušalca RTP sej ustvarimo z *RTPReceive* objektom, ki z javanskim objektom *RTPmanager*, metodo *SessionAddress.addFormat* ter metodo *SessionAddress.addSessionListener*, implementira dekodiranje in poslušanje seje na vratih RTP poslušalca. V predstavljenem primeru avdio in video tok procesiramo ločeno. Pri takšni implementaciji lahko ločeno kodiramo in obdelujemo avdio in video tok podatkov nad ustvarjenimi podatkovnimi viri. Rezultat procesiranja z objektom *RTPprocessor*, predstavlja kloniran podatkovni vir z izbrano razločljivostjo in kodiranjem. Namesto dodatnega procesiranja dalje izvedemo združevanje obeh kloniranih podatkovnih virov, da tako združimo avdio in video informacijo. Z uporabo objekta *RTPplayer*, ki deduje metode in lastnosti predvajalnika, pa združen podatkovni vir direktno predvajamo uporabniku. Dodatno kloniranje v tem primeru ni več potrebno, saj original hranimo v obliki ločenega podatkovnega vira za video in avdio informacijo, ki smo ju ustvarili v procesu kodiranja in obdelave. Grafični vmesnik za predvajanje sprejete informacije v obliki RTP toka podatkov, je identičen grafičnemu vmesniku prikazanem na sliki 10-15. S pritiskom na gumb "Receive", instanciramo RTP sprejemnik. Pri tem se odpre nov zavihek, na katerem lahko izvajamo podobne operacije, kot pri predvajanju avdio/video informacije zajete iz USB kamere. Implementacija RTP paketa omogoča tudi sočasno predvajanje zajete in prejete avdio/video informacije. V okviru procesiranja avdio/video informacije, je pomembno omeniti še proces razpošiljanja informacije, ki ga izvedemo nad zajetim podatkovnim virom (ali zajetim lokalno s pomočjo video kamere ali pa prejetim iz medmrežja v obliki RTP toka). Proces razpošiljanja avdio/video informacije je predstavljen na sliki 10-4



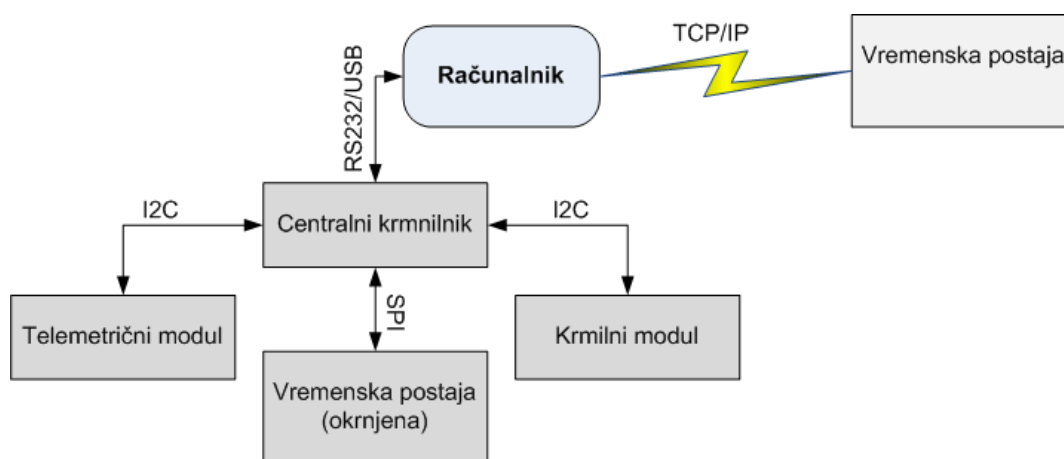
Slika 10-4: Funkcionalna shema razpošiljanja RTP toka.

Ko imamo avdio/video informacijo zajeto in pripravljeno za razpošiljanje v obliki podatkovnega vira, lahko s pomočjo modela RTP procesorja in izvedbe RTP procesorja na podlagi tega modela, zajeto avdio/video informacijo primerno zakodiramo. V pripravi RTP tokov hkrati zagotovimo, da kodiranje ustreza kodiranju, ki ga je moč prenesti z uporabo RTP protokola. Rezultat priprave RTP tokov je podatkovni vir tipa "Pushed", ki ga posredujemo procesu za razpošiljanja RTP toka. Proces dejansko predstavlja inverzni proces sprejemanja RTP toka in tako loči avdio in video informacijo, jo primerno zakodira in za vsak tip informacije tvori novo RTP sejo. S pomočjo *RTPManager.newInstance()* ustvarimo oddajnik RTP informacije na vsaki RTP seji. RTP sejo inicializiramo s pomočjo *initialize(localAddr)* metode, ki kot vhodni parameter zahteva naš IP naslov, dodati pa moramo še ciljni naslov s pomočjo metode *addTarget(destAddr)*. V primeru podajanja naslova za več prejemnikov (multicast), za ciljni naslov vpišemo *224.0.0.1* oz. broadcast **.*.255.255*.

11 RAZVOJ KOGNITIVNE MOBILNE PLATFORME »DATA«

11.1 Strojni nivo

Na strojnem nivoju, komponente inteligentnega okolja združujemo v funkcionalne skupine (module). Module lahko med seboj s pomočjo principa centralizacije združujemo ali pa uporabljamo možnost neposredne povezave modulov na računalnik. Implementirani moduli so namenjeni zajemanju podatkov o okolju in krmiljenju mobilnih enot. Spodnja shema predstavlja funkcionalnost in logično povezanost posameznih modulov na strojnem nivoju.

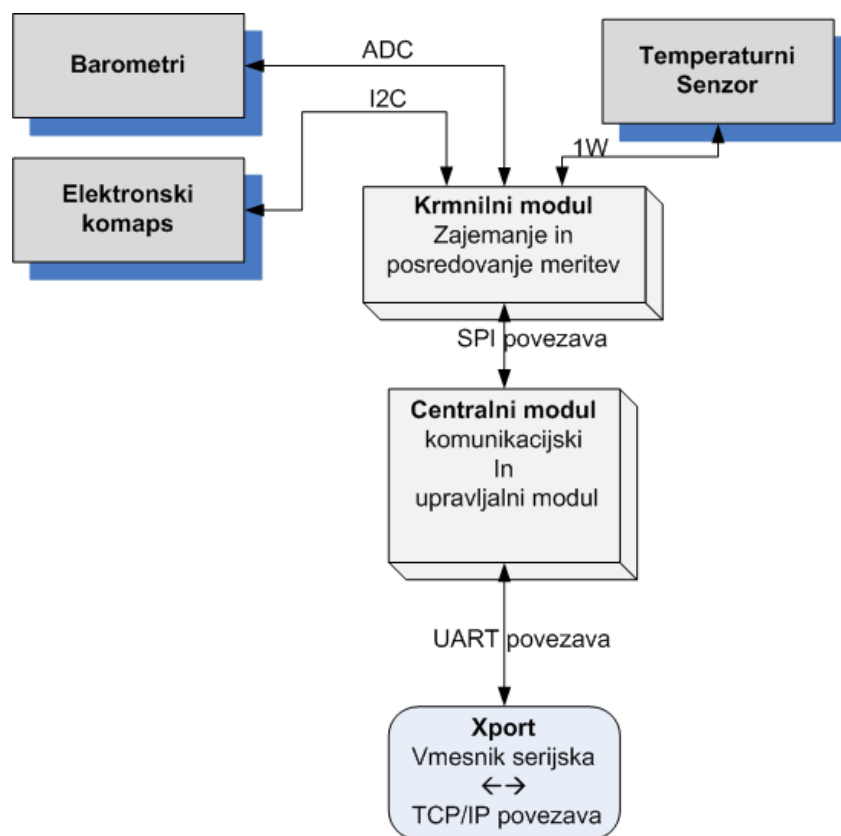


Slika 11-1: Funkcionalna shema sistema DATA na strojnem nivoju.

Iz slike 11-1 je razvidno, da za povezavo med posameznimi strojnimi moduli in računalnikom, v osnovi izkoriščamo RS232/USB povezavo, naprednejše in zmogljivejše komponente sistema DATA, pa podpirajo tudi TCP/IP povezave. Na strojnem nivoju smo implementirali več modularnih sistemov, ki lahko z manjšimi konfiguracijskimi spremembami delujejo neodvisno ali pa tako kot je prikazano na sliki, koristijo pristop centralizacije. Po pristopu centralizacije, enega izmed krmilnikov uporabimo za gospodarja sistema, in le ta lahko tvori povezavo z računalnikom. Za takšno realizacijo, potrebujemo le eno skupno povezavo do računalnika in ne več ene povezave za vsak modul. Ob ustrezni implementaciji centralnega krmilnika, lahko zmanjšamo tudi napajalne zahteve sistema, saj posamezni moduli takšnega centraliziranega sistema, delujejo le v primeru zahteve po njihovi uporabi.

11.1.1 Telemetrični modul (Vremenska postaja)

V okviru diplomskega dela smo razvili telemetrični modul, ki ga lahko uporabimo kot eno od komponent inteligentnega okolja. Sam modul je del kognitivne mobilne platforme »DATA«, je pa zaradi modularnega pristopa razvoja, takšen modul oz. več modulov možno postaviti kjerkoli v inteligentnem okolju. Modul omogoča merjenje zračnega pritiska, temperature okolice in usmerjenost uporabnika oz. mobilne enote (lokalizacija). S pomočjo grafičnega vmesnika, implementiranega v programskem okolju MATLAB,, modul omogoča interaktivno uporabo z možnostjo izvajanja meritev, izpisa in izrisa statističnih rezultatov in meritev zajetih v določenem časovnem obdobju. Spodnja slika predstavlja funkcionalno zgradbo celotnega modula.



Slika 11-2: Funkcionalna shema modula »vremenska postaja«.

Modul vremenske postaje sestavljajo naslednje komponente:

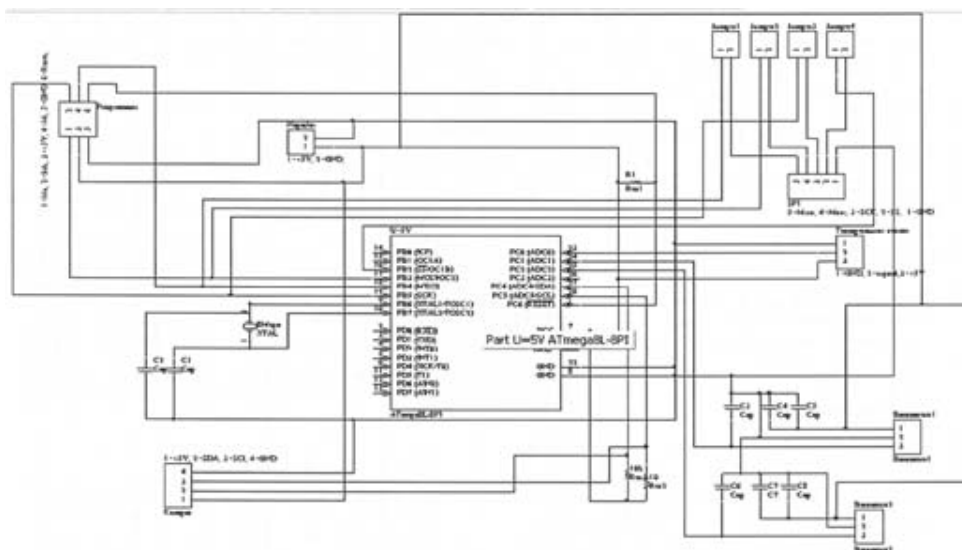
- mikrokrmilnik 2 x Atmel ATmega8,
- barometer MPX4100A in barometer MPX4115A,

- temperaturni senzor DS18B20,
- elektronski kompas CMPS03.

Osrednji gradnik modula je centralni modul, ki predstavlja komunikacijsko komponento strojnega dela z medmrežjem in posredno implementira podporo TCP/IP komunikaciji. Krmilni modul predstavlja zajemalno – posredovalno komponento sistema, ki skrbi za pravilno delovanje senzorjev. Krmilni modul implementira enožično (1W), TWI in ADC in SPI komunikacijske povezave. Komponenta Xport predstavlja direktni vmesnik med medmrežjem in strojnim delom modula vremenske postaja.

11.1.2 Krmilni modul

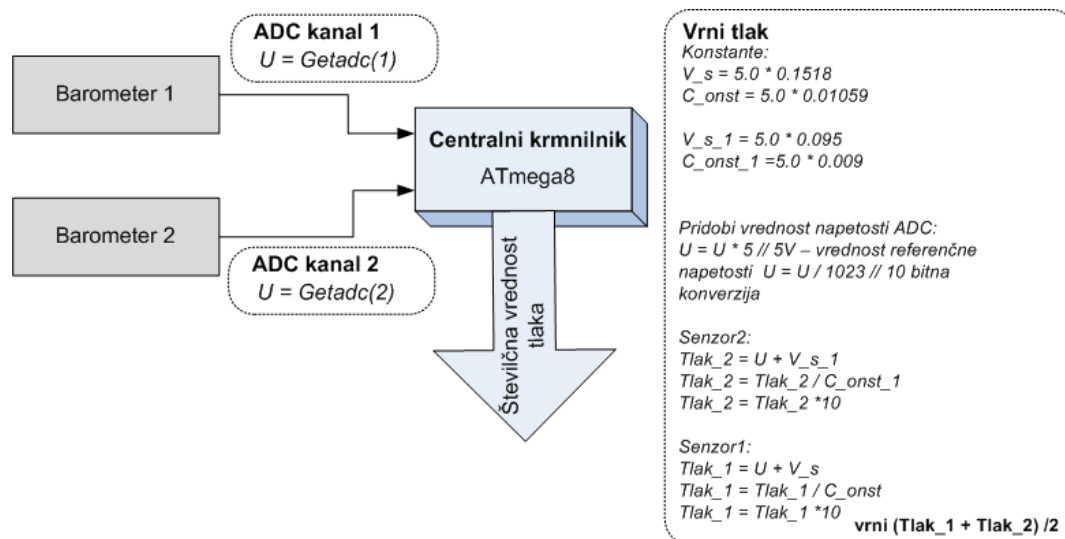
Mikrokrmilnik ATmega8, predstavlja krmilno komponento celotnega modula. Nanj smo priklopili dva barometra, elektronski kompas in temperaturni senzor. Naloga modula je zajemanje senzorskih meritev stanja okolja in posredovanja teh meritev centralnemu modulu. Pripadajoče vezje je predstavljeno na sliki 11-3.



Slika 11-3: Vezje krmilnega modula.

Barometra sta na centralni mikrokrmilnik povezana z uporabo ADC priključkov krmilnika. Na krmilnik je mogoče priključiti največ štiri naprave, ki uporabljajo ADC pretvorbo, saj ima krmilnik strojno podprte samo štiri ADC kanale. Vrednosti in postavitev samih kondenzatorjev ADC povezave, so predpisane s strani proizvajalca barometrov in so zapisane v priloženi dokumentaciji barometrov. V modulu smo

uporabili dva različna barometra zaradi doseganja večje natančnosti meritve, saj končni rezultat predstavlja povprečna vrednost obeh zajetih meritev. V ADC povezavi barometra na določenem priključku (ki je povezan na enega izmed ADC kanalov mikrokrmilnika) nastavljata višino napetosti, mikrokrmilnik pa prebrano vrednost pretvori v številsko vrednost. Povezava ADC je povsem enosmerna komunikacijska povezava, saj mikrokrmilnik v vnaprej predpisanih intervalih meri napetost na posameznih ADC kanalih. Slika 11-4 in prikazani izseki programske kode prikazujejo implementacijo ADC pretvorbe.



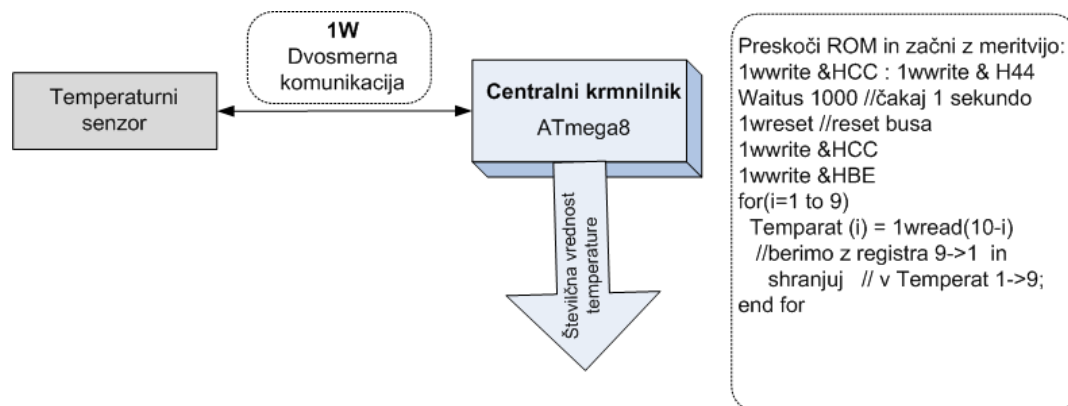
Slika 11-4: Funkcionalna shema merjenja tlaka okolja.

Centralni krmilnik implementira ADC povezavo na način, kot je prikazano na sliki 11-4. ADC pretvorbo konfiguriramo z ukazom »*Config Adc = Single, Prescaler = Auto, Reference = Avcc*«. S tem dosežemo pravilno nastavitve registra ADC na enojno konverzijo in 10 bitno pred-skaliranje. Metoda »*Start Adc*« postavi bit 7 ADEN kontrolnega registra. Referenčno vrednost odčitamo na priključku mikrokrmilnika Avcc. Napetost na ADC priključkih 1 in 2, preberemo s pomočjo metode *Getadc (kanal)*. V primeru, da se branje izvaja prvič, se v bit ADSC registra ADSCRA vpiše ena. Pretvorbe se tako izvajajo ne glede na stanje ADIF (prekinitvene zastavice). Ker smo nastavili enojno pretvorbo, se vrednost registra ADSC po opravljeni pretvorbi takoj izbrše. Metoda *Getadc* s pomočjo 10 bitnega pred-skaliranja in pretvorbe, analogno napetost na priključku pretvori v digitalno vrednost (digitalizira). V dokumentaciji

barometrov je predstavljena preslikava prebrane napetosti in formula po kateri izračunamo vrednost tlaka. Ker programsko okolje omogoča opravljanje le ene matematične operacije naenkrat, definiramo najprej potrebne konstante. Vrednosti teh konstant nato uporabimo v formuli, ki kot končni rezultat pretvori napetost na priključku v izmerjeni tlak. Postopek ponovimo za oba barometra, kot končni tlak pa vzamemo srednjo vrednost obeh meritev. Metodi za pretvorbo napetosti v vrednost tlaka sta:

- MPX4100A: $V_{out} = V_s(P * 0.01059 - 0.1518)$; $V_s = 5.1V \pm 0.25 * V_{dc}$
- MPX4115A: $V_{out} = V_s(P * 0.009 - 0.095)$; $V_s = 5.1V \pm 0.25 * V_{dc}$

Mikrokrmilnik opravlja ADC pretvorbo v določenem časovnem intervalu, ki je pogojen z ukazi centralnega krmilnika. Spremembo ADC kanala pretvorbe moramo vedno izvesti pred začetkom pretvorbe, zato v metodi *Getadc* podamo številko kanala kot vhodni parameter. ADMUX register se med izvajanjem pretvorbe zaklene in njegovo ažuriranje v tem času ni mogoče. Pretvorba vrednosti napetosti v vrednost tlaka, predstavlja množenje vrednosti podatkovnega ADC registra z referenčno vrednostjo (5V) in deljenje s 1023 (resolucija konverzije).



Slika 11-5: Funkcionalna shema merjenja temperature okolja.

Temperaturni senzor je na mikrokrmilnik povezan preko enožične povezave (1W). Za priklop smo izbrali ADC kanal 0, ki ga programiramo kot generalni vhodni priključek. Za implementacijo 1W povezave, bi lahko izbrali katerikoli priključek (razen rezerviranih) mikrokrmilnika. Za merjenje temperature smo uporabili Dallasov DS18B20 senzor. Funkcionalna shema meritve temperature je prikazana na sliki 11-5. Temperaturni senzor povežemo z mikrokrmilnikom preko enožične 1W povezave.

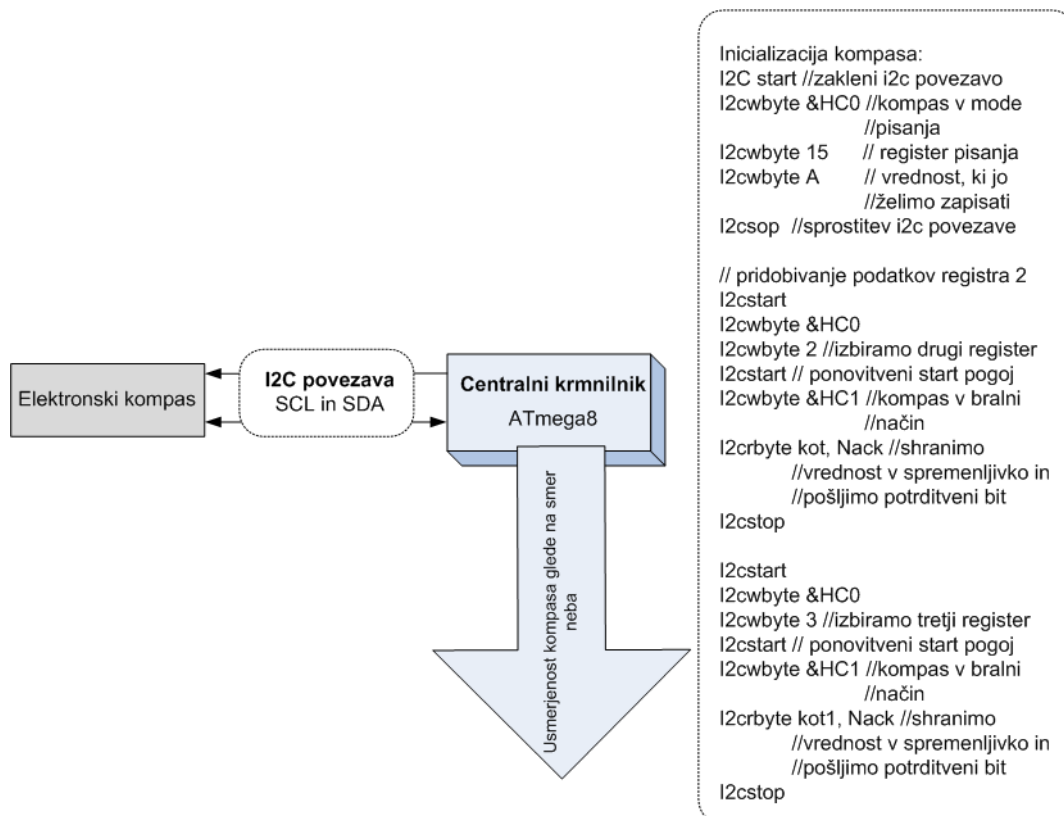
Protokol takšne povezave podpira izmenično pisanje in branje vrednosti registrov temperaturnega senzorja. I2C povezavo definiramo z uporabo *Config I2C = port X*. S tem povemo, da smo port X registrirali kot vhodno/izhodni priključek, ki je namenjen za enožično povezavo. Ob inicializaciji pošljemo senzorju ukaz, da želimo preskočiti branje/pisanje na interni ROM senzorja. To storimo z ukazom *I2Cwrite &HCC*, ki mu sledi ukaz za začetek meritev *I2Cwrite &H44*. Po koncu inicializacijskega postopka enožično povezavo sprostimo z ukazom za resetiranje. Meritev na senzorjih se izvaja nemoteno. Pred vsakim ponovnim pisanjem na senzor, moramo enožično povezavo vedno ponovno sprostiti. Po inicializaciji, s temperaturnega senzorja beremo vrednosti z uporabo ukaza *I2Cwrite &HB*. Senzor nam vrača rezultat v obliki 9-bitnih paketov. Branje zato implementiramo s pomočjo zanke, kjer števec teče od 1 do 9, v polje pa zapisujemo vrednosti sledeče:

- Vrednost(1) = I2Cread(9),
- Vrednost(9) = I2Cread(1).

Vrednost temperature podajata le prvi in drugi paket, ki ju tudi uporabimo pri izračunu temperature. Temperaturo izračunamo s pomočjo naslednje enačbe:

$$\text{Temperatura} = (\text{Makeint}(\text{Vrednost}(1), \text{Vrednost}(2))/16)$$

Za določanje usmerjenosti uporabnika oz. mobilne enote, smo na mikrokrmilnik priključili tudi elektronski kompas CMPS03. Kompas že podpira I2C komunikacijo, zato ga povežemo z mikrokrmilnikom preko priključkov, ki jih mikrokrmilnik uporablja pri I2C povezavi. V kontekstu I2C povezave, predstavlja mikrokrmilnik »gospodarja«, elektronski kompas pa sužnja. Komunikacija med njima poteka v obliki pisanja in branja na I2C registre kompasa. Podroben prikaz implementacije je prikazan na sliki 11-6.



Slika 11-6: Funkcionalna shema elektronskega kompasu.

Programsko orodje Bascom, omogoča implementacijo povezave I2C, brez potrebe po direktnem naslavljanju I2C registrov mikrokrmilnika. I2C komunikacija poteka po naslednjem postopku:

- Zagon povezave: *i2cstart*,
- Pošiljanje zahteve po pisanju: *i2cwbyte &HC0*,
- Pošiljanje registra na katerega bomo pisali: *i2cwbyte 15*,
- Zapis vrednosti v izbrani register: *i2cwbyte A*,
- Sprostitev povezave *i2cstop*.

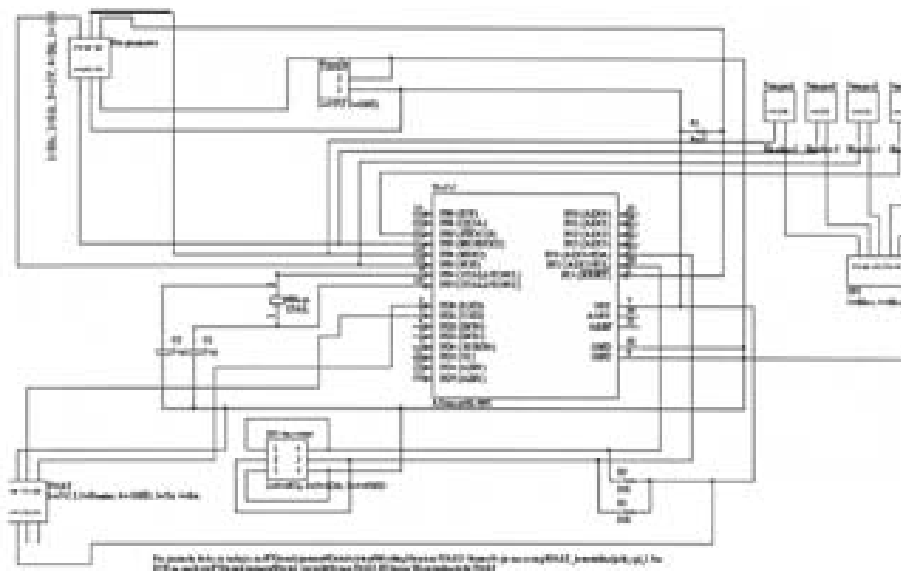
V primeru takojšnjega branja z določenega registra, namesto pogoja STOP, uporabimo ponovitveni pogoj START, ki mu sledi zapisovanje registra, iz katerega želimo brati vrednosti registra. Ob koncu upravljanja določenega registra, moramo I2C povezavo nujno sprostiti. V primeru elektronskega kompasu, uporabljamo za pisanje v register kompasu heksa-decimalno oznako &HC0, trenutne meritve kompasu pa

pridobimo z zapisom heksa-decimalne vrednosti &HC1 v podatkovna registra kompas 2 in 3. &HC0 označuje naslov za pisanje v registre kompas, &HC1 pa naslov za branje registrov kompas. Na sliki 11-6 je predstavljen način branja trenutne meritve kompas, kjer se uporablja ponovitveni pogoj START, brez uporabe pogoja STOP. Najprej pošljemo na naslov za pisanje vrednost 2, ki pove, da želimo brati iz registra številka 2, nato pošljemo ponovitveni pogoj START in iz naslova za branje preberemo vrednost registra 2. I2C povezavo nato sprostimo. Postopek ponovimo še za drugi register. Po končanem branju informacije, izvedemo abstrakcijo informacije, in jo pretvorimo v kotne stopinje. Za abstrakcijo uporabimo naslednjo formulo, ki je zapisana v dokumentaciji kompas:

$$\text{Zasuk} = (\text{Kot} * 256 + \text{Kot}_1) / 10$$

11.1.3 Centralni modul

Centralni modul predstavlja vmesnik med mrežnim in strojnim nivojem. Za povezovanje s strojnimi nivojem, implementira SPI povezavo s krmilnim modulom, z uporabo UART povezave in Xport vmesnika pa je implementirana podpora za TCP/IP povezavo. Implementacija centralnega modula je prikazana na sliki 11-7.



Slika 11-7: Centralni modul.

Centralni modul omogoča komunikacijo med strojnimi komponentami vremenske postaje in medmrežjem. Za komunikacijo s strojnim delom, smo implementirali SPI povezavo. SPI komunikacija predstavlja alternativni način povezave dveh mikrokrmilnikov. Iz vezja na sliki 11-7 je razvidno, da se v našem primeru uporablja SPI komunikacija tudi za programiranje krmilnikov. V primeru programiranja mikrokrmilnika s pomočjo preklopnih priključkov (jumpers), onemogočimo SPI komunikacijo z ostalimi komponentami v povezavi. Ta korak je nujen, saj bi v nasprotnem primeru programirali vse mikrokrmilnike v povezavi. Atmega8 podpira implementacijo SPI povezave s priključki MISO, MOSI, SCK in SS. Z ukazom *Config Spi = Hard, Master = Yes, Clockrate = 4* definiramo mikrokrmilnik kot »gospodar«, z ukazom *Config Spi = Hard, Master = No, Clockrate = 4* pa definiramo vlogo mikrokrmilnika kot »suženj«. Kadar mikrokrmilnik nastopa kot »gospodar«, generira urin impulz na povezavi SCK, s ciklom $4 \cdot \text{hitrost_zunanjega_priključenega_kristala}$. Preko SPI povezave prenašamo podatke v obliki 8-bitnih znakov. Samo SPI povezavo inicializiramo z ukazom *spiinit*. Za pošiljanje znakov uporabimo metodo *spiout "karakter", 1*, sprejemamo pa s pomočjo metode *chr(spiin("karakter", 1))*.

Za komunikacijo z medmrežjem uporabimo na centralnem modulu mrežni vmesnik Xport, na katerega se centralni modul lahko poveže preko UART povezave. UART povezavo konfiguriramo z naslednjima ukazoma:

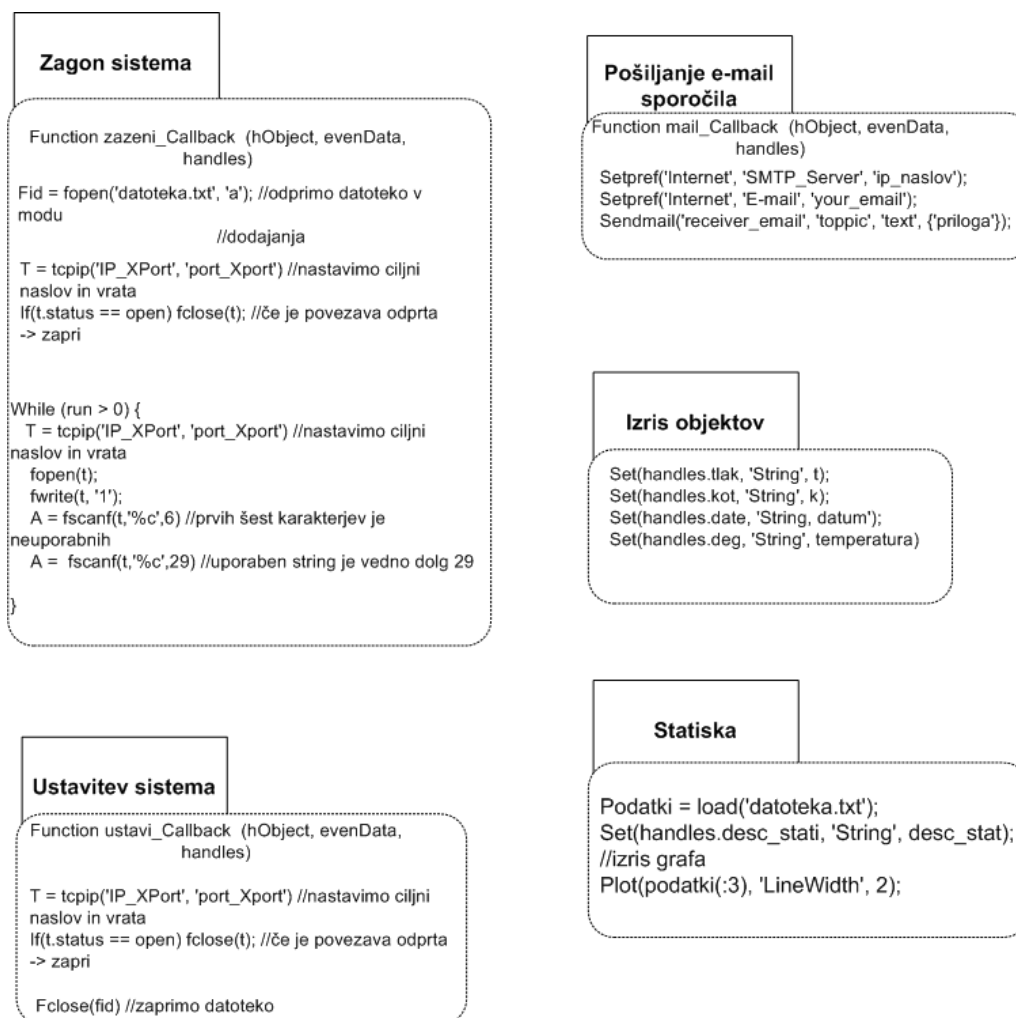
- *Config Serialin = Buffered, Size = 20,*
- *Config Serialout = Buffered, Size = 20.*

Ob inicializaciji tako rezerviramo prostor pomnilnika krmilnika velikosti 20 znakov, v katerega lahko začasno shranjujemo podatke za pošiljanje in prejemanje. Metoda *Inputbin znak* bere vhodne podatke v *buferju Serialin*. Če na vhodu modula ni zahtevanega znaka za zagon modula, se centralni krmilnik postavi v fazo spanja. V nasprotnem primeru, zahtevamo trenutne podatke krmilnega modula, in jih v obliki niza znakov posredujemo Xport modulu, ki jih posreduje v obliki TCP/IP paketov odjemalcu, ki je zahteval dano operacijo. Zaradi razumevanja vrednosti, ki se pošiljajo, izvedemo pošiljanje v obliki paketa na naslednji način: »<k1>vrednost<t1>vrednost<p1>vrednost«. Pri tem **k1** označuje, da gre za vrednost kompasa, **t1** označuje, da gre za vrednost temperature, **p1** pa označuje, da gre za srednjo

vrednost tlaka. Takšna implementacija paketov omogoča relativno enostavno razčlenjevanje sprejetega TCP/IP paketa in razumevanje prejete informacije v vmesniku na oddaljenem računalniku.

11.1.4 Vmesnik na oddaljenem računalniku

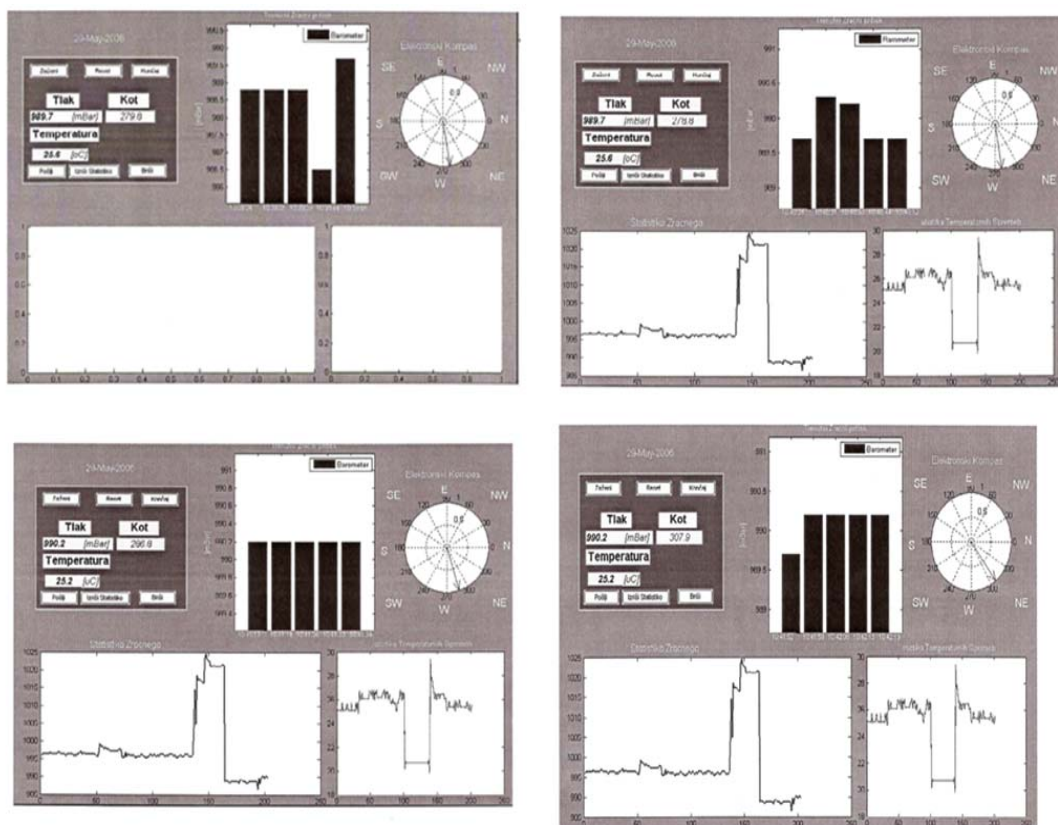
Grafični vmesnik modula vremenska postaja smo implementirali v programskem okolju MATLAB [45]. Vmesnik omogoča predstavitev podatkov zajetih s pomočjo senzorjev, zagon posameznih meritev in njihovo ustavitve, ter pregled statistike za določeno obdobje. Funkcionalna shema vmesnika je predstavljena na sliki 11-8.



Slika 11-8: Funkcionalna shema vmesnika v MATLAB orodju.

Slika 11-8 predstavlja funkcionalni opis zaključenih funkcionalnosti vmesnika. Za zagon celotnega sistema pritisnemo na gumb »zaženi«. Ta nato proži *callBack* metodo,

ki do pritiska na gumb »ustavi«, pošilja na strojni del ukaze za pridobivanje podatkov. Podatki se na vmesniku prikazujejo tako v obliki številk kot v obliki grafov. Za izris grafov uporabljamo v MATLABU omogočene metode izrisa podatkov tipa *plot* in manipulacije nad njimi. Dodatno implementiramo tudi elektronski kompas, ki v obliki puščice pokaže usmerjenost kompasa glede na smer neba. Stolpčni diagram je dinamičen in sproti posodablja svoje vrednosti. Naenkrat prikazuje trenutno in nekaj preteklih meritev. S pritiskom na gumb »pošlji«, se izvrši metoda za pošiljanje statične datoteke. Najprej moramo nastaviti IP naslov SMTP strežnika, nato naslov naslovnika. Sistem za enkrat ne podpira pošiljanja e-mail sporočil preko strežnikov, ki zahtevajo avtentikacijo in avtorizacijo. Trenutno implementiramo pošiljanje sporočil enkrat dnevno.



Slika 11-9: Grafični vmesnik telemetričnega sistema.

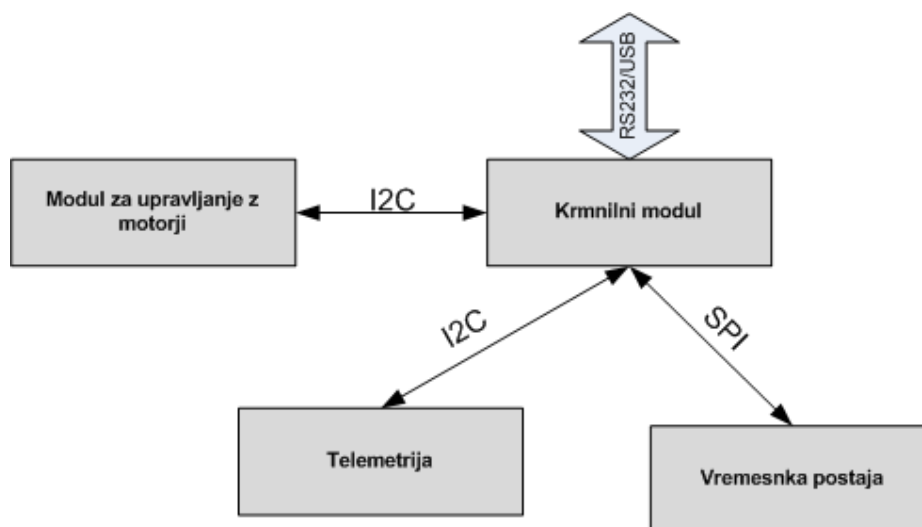
Za prikaz trenutnih vrednosti implementiramo še mehanizme preverjanja smiselnosti vrednosti. V primeru, da je vrednost tlaka izven meja 900 in 1120 oz. kot izven meja 0 in 359, ali pa trenutna sprememba tlaka varira za več kot 50 oz. trenutni kot za več kot

1, vrednosti enostavno ignoriramo in jih ne prikažemo. V takšnih primerih predvidevamo, da je prišlo do napačne informacije pri sprejemu. Sledeče slike prikazujejo uporabo in delovanje modula vremenske postaje. Zgornje slike predstavljajo grafični vmesnik modula Vremenske postaje. Prva slika prikazuje delovanje v realnem času brez prikaza statistike, vse nadaljnje slike pa prikazujejo še statistični graf tlaka in temperature za časovno obdobje 200 vnosov v podatkovno datoteko (časovno obdobje 30 minut).

11.2 Krmilni sistem mobilne robotske enote

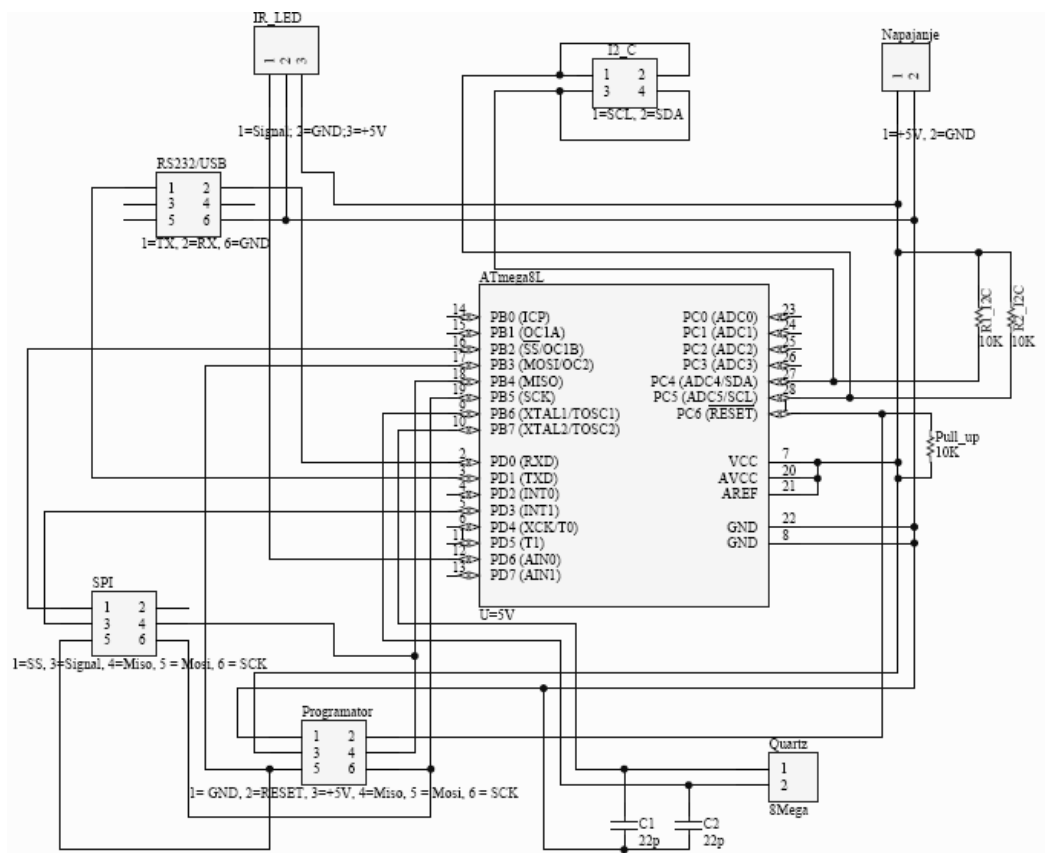
V prejšnjem poglavju predstavljen modul »vremenska postaja«, je predstavljal izhodišče razvoja vseh nadaljnjih strojnih modulov sistema DATA. V okviru diplomske naloge smo tako v nadaljevanju razvili merilno-krmilni sistem, ki smo ga preko RS232/USB povezali z računalniško aplikacijo, oz. gradnikom sistema DATA, imenovan tudi DATA modularni strežnik. V tem podpoglavju se bomo osredotočili le na strojni del krmilnih modulov, aplikativni del pa bomo predstavili v poglavju 12.

Na spodnji sliki je predstavljena funkcionalna shema krmilnih modulov mobilne robotske enote.



Slika 11-10: Funkcionalna shema modula upravljanja mobilne robotske enote .

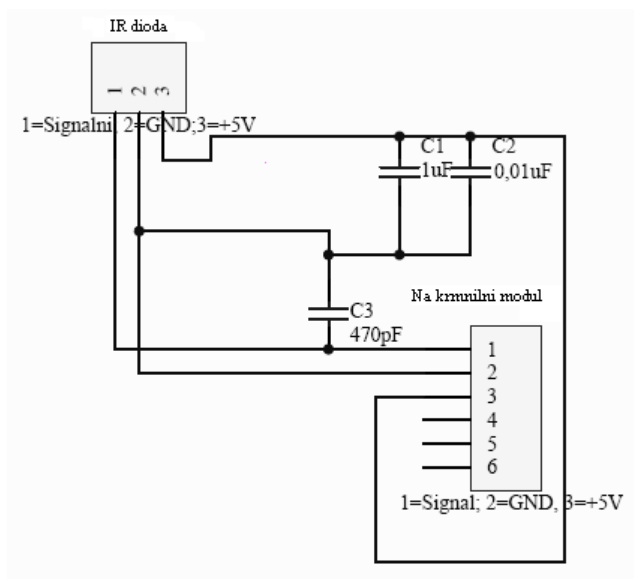
Krmilni modul je implementiran centralizirano in ga sestavljajo glavni krmilnik, t.i. krmilni modul, ki gosti RS232/USB povezavo do lokalnega računalnika, in naslednji suženjski moduli: za telemetrijo, za, upravljanje z motorji in vremensko postajo. Komunikacijske povezave so implementirane ob uporabi I2C in SPI tehnologije. S pomočjo krmilnega modula lahko upravljamo z enoto in prebiramo podatke o statusu okolja v katerem se enota nahaja. V nadaljevanju sledi opis posameznih zaključenih pod modulov sistema krmiljenja. Programska logika strojnega nivoja, je zapisana v programskem orodju Bascom [39], ki z implementacijo posameznih metod poenostavi potrebo po ročnem naslavljanju določenih registrov. Realizacija komunikacijskih povezav v kontekstu naslavljanja registrov, je predstavljena v poglavju 8. Na vsakem vezju posameznega modula, smo implementirali možnost reprogramiranja (z uporabo SPI priključkov mikrokrmilnika) in vsakemu mikrokrmilniku dodelili zunanji generator takta s frekvenco 8Mhz. Enaka ura zagotavlja sinhroniziran prenos podatkov med moduli, ni pa nujen pogoj za implementacijo.



Slika 11-11: Funkcionalna shema krmilnega modula.

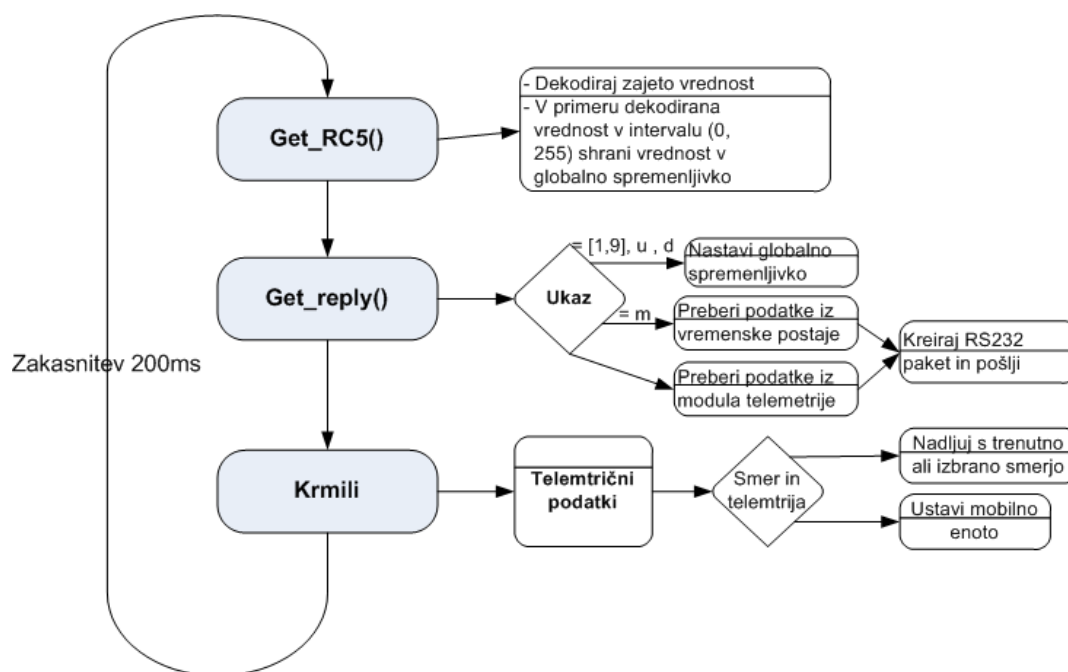
11.2.1 Krmilni modul

Krmilni modul predstavlja srce modula upravljanja. Je edini modul, ki gosti povezavo z lokalnim računalnikom in lahko tako sprejema ukaze uporabnikov ter posledično upravlja z ostalimi moduli sistema. Elektronska shema vezja je predstavljena na sliki 11-11. Iz slike je razvidno, kako so implementirane komunikacijske povezave krmilnega modula in ostalih suženjskih modulov sistema krmiljenja mobilne robotske enote. RS232/USB povezava je implementirana v obliki dvožične povezave, ki uporablja RxD in TxD rezervirana priključka za strojno implementacijo serijske povezave. Zaradi pravilnega prenosa podatkov, takšna implementacija zahteva še uporabo skupne mase (na ta način prisilimo sistem v enotno označevanje visokega in nizkega nivoja napetosti in posledično enoznačno označevanje +5V kot 1 in 0V kot 0) . SPI povezava se podobno kot v modulu vremenske postaje, implementira tako za programiranje mikrokrmilnika, kot za komunikacijo med dvema mikrokrmilnikoma. V primeru programiranja mikrokrmilnika moramo zato SPI povezavo na drug mikrokrmilnik odklopiti (programsko orodje Bascom nam ne dopušča programiranja v primeru aktivnosti SS priključka). Povezava s štirimi priključki MISO, MOSI, SCK in SS se na mikrokrmilniku implementira z uporabo strojno rezerviranih priključkov.



Slika 11-12: RC5 sprejemno vezje.

Za implementacijo I2C povezave uporabimo zadnja priključka ADC pretvorbe, ki sta rezervirana za strojno implementacijo I2C podpore. Vsak izmed linij SDA in SDL, potrebujeta za pravilno delovanja 10kΩ upor (pull-up). Vrednost upora je približna, saj moramo v izračunu vrednosti upoštevati celotno dolžino I2C komunikacijske povezave in prevodnost materiala s katerim implementiramo to povezavo. Za boljšo stabilnost delovanja povezave, lahko v primeru napak pri prenosu na vir napetosti povežemo še kondenzator, ki napetost dodatno stabilizira. Direktno krmiljenje enote je podprto z uporabo IR sprejemne diode in RC5 protokola. Krmiljenje na ta način lahko opravljamo z navadnim daljinskim upravljalcem za televizijo. RC5 sprejemno vezje smo izvedli kot ločeno komponento. Vezje je predstavljeno na sliki 11-12. Vezje z IR diodo, ima dve priključni spojki. V prvo spojko vstavimo IR diodo, z drugo pa povežemo sprejemno in krmilno vezje. Napajanje zagotovimo iz krmilnega vezja, s čimer hkrati zagotovimo tudi skupno maso. Kondenzatorji v vezju služijo za stabilizacijo signalov in napetosti na povezavah. Njihove vrednosti so podane s strani proizvajalca IR diode. Na sliki 11-13 je predstavljena programska logika delovanja modula.



Slika 11-13: Programska logika krmilnega modula.

Ob vsaki iteraciji "do-while" zanke, najprej preverimo ali kdo želi upravljati direktno z enoto. To izvedemo z branjem vrednosti sprejemne IR diode, ki podatke oddaja v obliki RC5 protokola. RC5 komunikacijo v vsaki iteraciji zanke inicializiramo in iz priključka beremo vrednost z naslednjim nizom ukazov:

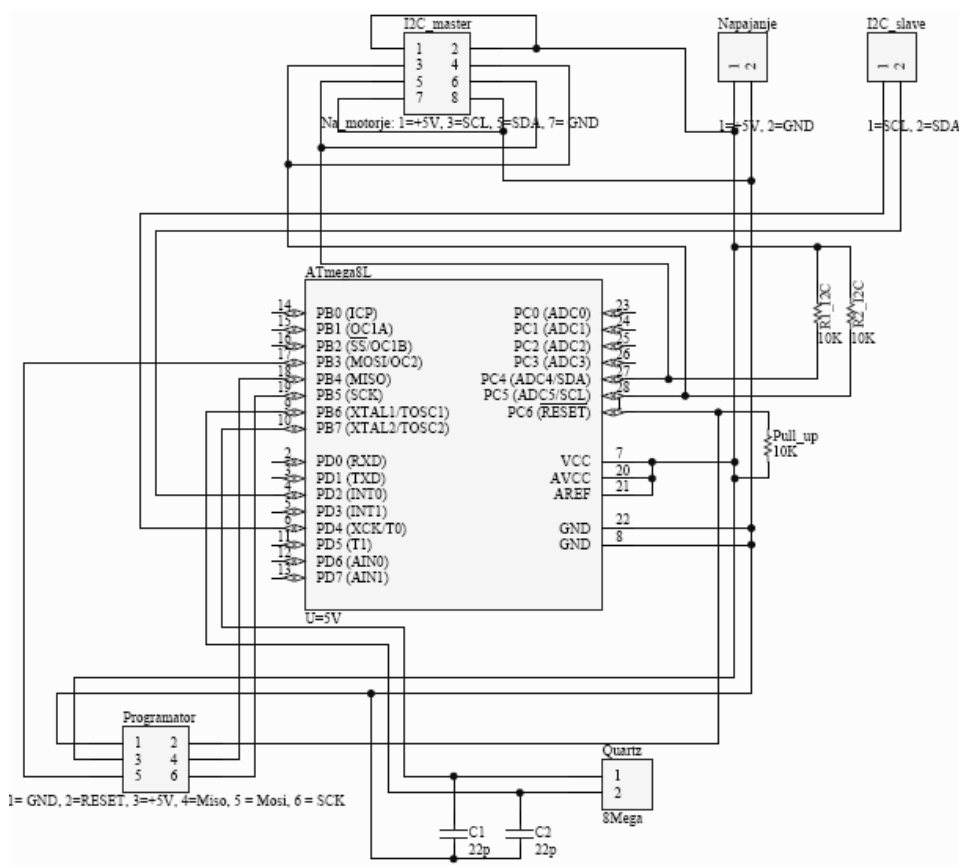
```
Config Pind.6 = Input
Getrc5(address , Command)
```

Getrc5 je metoda, ki jo že podpira Bascom in nam omogoči branje RC5 vrednosti z določenega naslova, ki jo shranimo v spremenljivko *Command*. Kodirano vrednost spremenljivke nato pretvorimo v desetiško obliko z uporabo *Command = Command And &B01111111*. Sledi samo še preverjanje, ali je vrednost na intervalu (0, 255). V primeru da se pretvorjena vrednost nahaja v tem intervalu, nastavimo globalno vrednost spremenljivke, ki jo pošiljamo modulu za upravljanje z motorji. V primeru, da direktni krmilni ukaz ni bil sprožen, poizkusimo še z branjem vhoda na serijski povezavi. Vhodni predpomnilnik povezave preberemo z uporabo metode *Inkey()*. Za razliko od prej omenjene metode *Inputbin*, metode branja, metoda *Inkey()* vsebuje časovnik, ki v primeru, da predpomnilnik ne vsebuje povezave, preneha poslušati in programska logika lahko nadaljuje z samim izvajanjem. Metoda *Inputbin()* onemogoči nadaljevanje izvajanja logike, vse dokler predpomnilnik ne vsebuje neke vsebine. Glede na sprejeto vrednost ukaza, se nato programska logika odloči, kakšno akcijo je primerno implementirati. V primeru, da je ukaz na intervalu [1, 9] ali *d* (upočasnjevanje) oz. *u* (pospeševanje), se ponastavi vrednost globalne spremenljivke za določanje smeri. V primeru, da je ukaz enak znaku *m* (meritve), se zažene metoda, ki izvede pridobivanje vrednosti meritev iz modula vremenske postaje. Ta zajem izvedemo s pomočjo SPI komunikacije, ki se obravnava identično, kot smo že opisali v poglavju 8. Najprej nastavimo povezavo "pogoj" (ki se dodatno implementira), s čimer vremenskemu modulu »povemo«, da naj izvede operacijo merjenja. Ker sta modula sinhronizirana, vremenski modul najprej pošlje vrednosti temperature in nato še vrednosti tlaka. Krmilni modul jih nato pretvori v besede (podatkovni tip "String") in v obliki RS232 paketa pošlje na glavno računalniško enoto. RS232 paket formiramo na način, kot smo opisali v poglavju 8. V primeru, da vrednost, ki je prebrana iz predpomnilnika serijske povezave, ne zavzema nobene izmed prej naštetih vrednosti, se izvede pošiljanje

trenutnega stanja telemetričnih meritev, ki so se izvedle v prejšnji iteraciji. Vrednosti se vračajo glede na trenutno smer gibanja mobilne enote. V kodi mikrokrmilnika implementiramo RS232 povezavo podobno kot UART povezavo, saj obe povezavi predstavljata serijsko povezavo.

Najprej povezavi dodelimo predpomnilnik ("buffer"), v katerega shranjujemo dohodne in odhodne podatke. S klicem *Config Serialin = Buffered , Size = 200 in Config Serialout = Buffered , Size = 200* nato nastavimo predpomnjenje vhodnih in izhodnih znakov velikostnega razreda 200 znakov. Preverjanje telemetričnih iteracij, se odvije v vsaki iteraciji ne glede na prejeto zahtevo. To predstavlja varovalni mehanizem, da se enota v nobenem primeru ne zaleti. Telemetrični modul je s krmilnim modulom povezan preko I2C povezave v relaciji gospodar-suženj, kjer telemetrični modul igra vlogo »sužnja«. Za branje vrednosti trenutnih meritev, najprej senzor naslovimo z ukazom *I2csend Slave , Sensor_num*. Ta metoda predstavlja skrajšano implementacijo I2C komunikacije, predstavljene v poglavju 8. S tem ukazom modulu povemo, katero vrednost sensorja želimo brati. Gornjemu ukazu sledi še niz ukazov *I2csend Slave , 10i* in *I2creceive Slave , Sprj_chr*. S tem nizom sužnju sporočimo, kateri znak vrednosti naj vrne; *i* predstavlja vrednosti na intervalu [1,4]. Meritev na posameznem sensorju zajamemo v treh iteracijah (vsaka iteracija predstavlja meritev na drugi poziciji; podrobneje smo to razložili v opisu telemetričnega modula). Ko prejmemo vse zelene vrednosti telemetrije in jih shranimo v spremenljivke, se lotimo še preverjanja razdalj. Preverjanje razdalj se izvede glede na trenutno smer gibanja. V programski kodi točno določimo vrednosti katerih sensorjev lahko vplivajo na gibanje enote. V primeru gibanja, se naprej upoštevajo vse tri iteracije sensorja na poziciji spredaj; v primeru zavijanja levo se upošteva prva iteracija meritve sensorja na poziciji spredaj in vse tri iteracije meritve sensorja na poziciji levo, itn. V primeru, da so pridobljene vrednosti manjše od priporočenih, se smer premikanja nastavi na 0, kar pomeni, da se mora enota ustaviti. Trenutno prepovedana smer gibanja oz. zelena smer gibanja, se shranita v začasno spremenljivko in se lahko izvedeta v primeru, ko se ovira dovolj oddalji, zajete vrednosti so postanejo večje od mejnih vrednosti bližine ovire. Nazadnje krmilni modul izvede še krmiljenje. Ta enostavna metoda pošlje vrednost globalne spremenljivke nastavitve smeri modulu krmiljenja v obliki I2C paketa. Prenos

se izvede v obliki naslednjega ukaza *I2csend Slave_1*, *Tmp_smer*. Podobno kot telemetrični modul, tudi krmilni modul igra vlogo »sužnja« v tej I2C povezavi.



Slika 11-14: Modul za upravljanje z motorji.

11.2.2 Modul za upravljanje z motorji

Modul omogoča manipulacijo krmilnikov motorjev MD05 preko I2C povezave. V tej povezavi igra mikrokrmilnik modula za upravljanje z motorji vlogo »gospodarja«, »sužnja« pa sta krmilnika MD05. Naslova motorjev MD05 izberemo fizično z nastavitvijo naslovnih priključkov. Točen postopek določanja I2C naslova krmilnikov motorjev, je predstavljen v dokumentaciji motorjev. Kot smo že omenili v opisu krmilnega modula, pa modula za upravljanje implementirata tudi programsko I2C povezavo, kjer igra vlogo »sužnja«. Z uporabo *I2Cslave* knjižnice, ki je na voljo kot nadstandardna knjižnica programskega orodja Bascom, lahko na mikrokrmilnikih *ATmega8*, implementiramo sočasno delovanje v načinu »gospodarja« in »sužnja«. Vežje modula je predstavljeno na sliki 11-14. Iz slike je razvidno, da se ločeni I2C

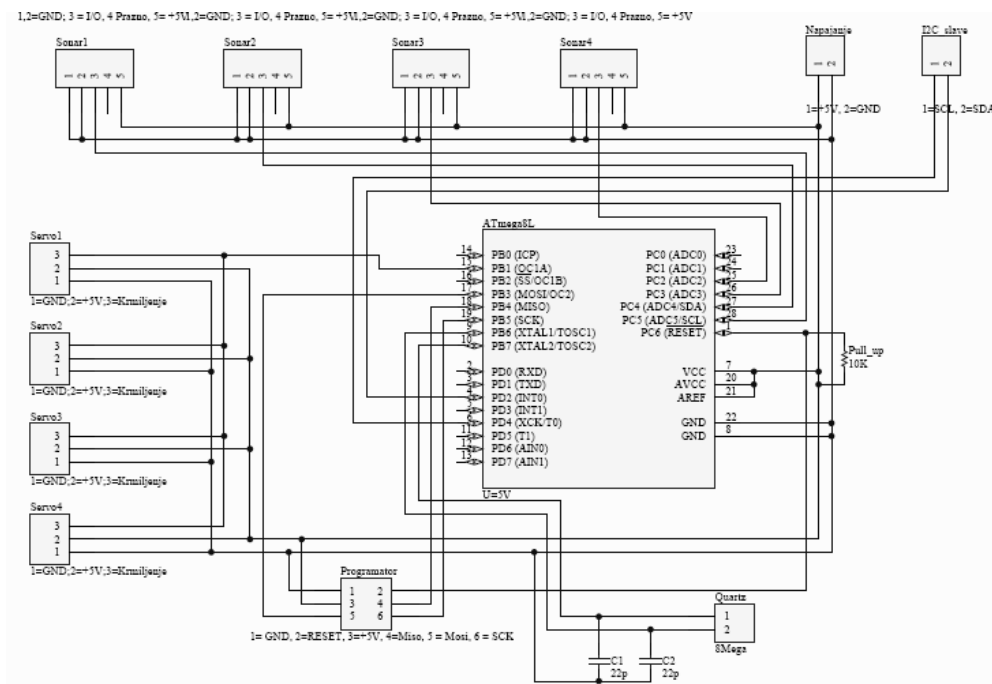
povezavi implementirata na ločenih priključkih mikrokrmilnika. Strojno podprta I2C povezava je implementirana na 4 in 5 kanalu ADC pretvorbe, ki sta rezervirana tudi za strojno implementacijo TWI povezave. V tem načinu povezave, predstavlja krmilnik modula »gospodarja«, krmilnika motorjev pa sta »sužnja«. Ker povezava še nima "pull-up" uporov, moramo dodati tudi te. V primeru programske implementacije I2C povezave, predstavlja krmilni modul »gospodarja« povezave, modul za upravljanje z motorji pa »sužnja«. Povezavo izvedemo nujno preko priključkov INT0 in T0 oz. INT1 in T1. Te povezave ni potrebno dodatno opremiti z upori, saj smo to napravili že na vezju, ki predstavlja »gospodarja« povezave. V smislu programske logike, izvedemo programsko I2C povezavo z uporabo naslednjega niza ukazov ob inicializaciji:

- *Enable Interrupts*; s čimer omogočimo prekinitve, ki so pogoj programske izvedbe I2C povezave,
- *Config I2cslave = &H50 , Int = Int0*; dodelimo naslov sužnja in SDA linijo povezave na priključku Int0,
- *Config I2cdelay = 0*; ne dovoljujemo zakasnitve pri prenosu .

V primeru programske implementacije I2C povezave, moramo nujno deklarirati metodi *I2c_master_needs_data* in *I2c_master_has_data*. Ti metodi ne potrebujeta nobene posebne logike. V primeru modula za upravljanje povezave, je metoda *I2c_master_needs_data* odveč, zato v njej ne implementiramo nobene programske logike. V metodi *I2c_master_has_data* pa implementiramo logiko komunikacije po strojno podprti I2C povezavi. V primeru, da gospodar želi pisati na modul, najprej preverimo ali je prejeta vrednost v intervalu (0,255), v nasprotnem primeru modul ne opravi akcije. Nato preverimo, ali ukaz predstavlja nastavitev smeri gibanja ali nastavitev pospeševanja v trenutni smeri gibanja. V ta namen pripravimo dve polji velikost 3. Vrednost *polje(1)* predstavlja hitrost gibanja, *polje(2)* predstavlja pospešek, *polje(3)* pa smer gibanja. Glede na vrednost ukaza polja primerno nastavimo in posredujemo krmilnikoma obeh motorjev. Posredovanje I2C paketa poteka po naslednjem zaporedju (sledеče zaporedje opravimo na obeh krmilnikih motorjev):

```
I2cstart                'generate start  
I2cwbyte &HB2          'write address, select control register  
I2cwbyte 2  
I2cwbyte Motor1(1) , Nack  
I2cstop
```

I2cstart
I2cwrite & HB2
I2cwrite 3
I2cwrite Motor1(2) , Nack
I2cstop



Slika 11-15: Vežje telemetričnega modula .

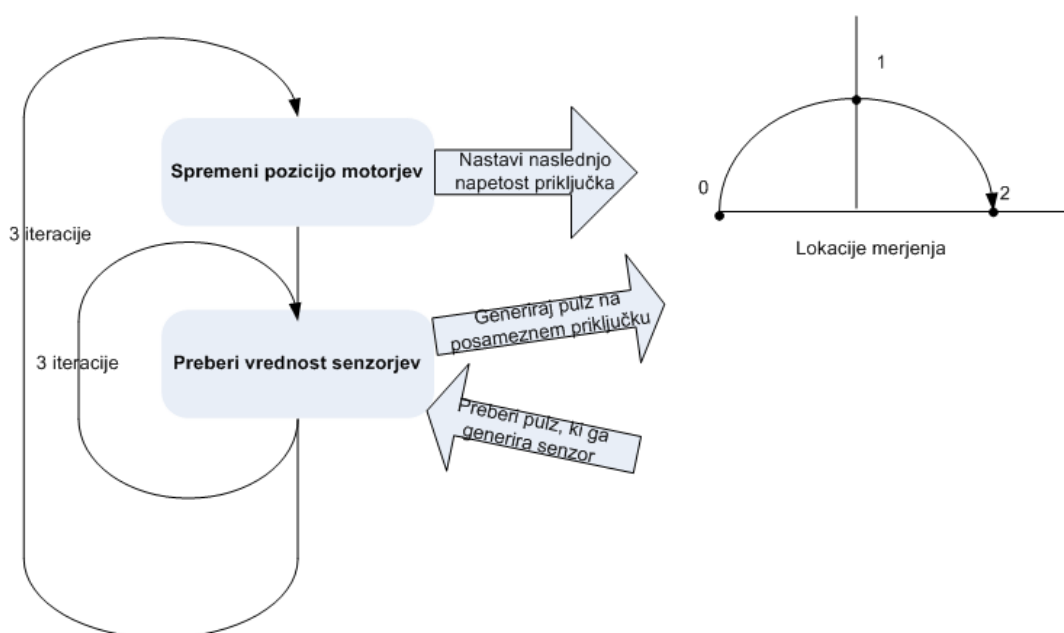
Vedno najprej nastavimo hitrost in pospešek obeh motorjev, šele nato pa motorju nastavimo smer in ga hkrati z naslednjim nizom ukazov poženemo v delovanje pod novimi pogoji:

I2cstart
I2cwrite & HB2
I2cwrite 0
I2cwrite Motor1(3) , Nack
I2cstop

11.2.3 Telemetrični modul

Telemetrični modul opravlja nalogo pridobivanja podatkov o oddaljenosti predmetov v bližnji okolici mobilne robotske enote. Sestavljajo ga *ATmega8* mikrokrmilnik, štiri SFP08 ultrazvočni senzorji in štiri servo motorji na katere pritrdimo ultrazvočne senzorje. Telemetrični modul preko programske izvedbe I2C povezave, upravlja krmilni

modul. Vežje modula je prikazano na sliki 11-15. Branje vrednosti senzorjev implementiramo z uporabo ADC pretvorbe, pri čemer se vsak izmed senzorjev poveže na enega izmed ADC kanalov. Pri tem direktno ne uporabljamo metod ADC pretvorbe, ki jih implementira Bascom. Vsak senzor potrebuje za zagon meritve napetostni pulz, kot rezultat pa vrne napetostni pulz. Zato moramo način pridobivanja senzorskih vrednosti prirediti njihovim zahtevam. Podobno kot modul za upravljanje z motorji, tudi telemetrični modul implementira programsko I2C povezavo, ki je ni potrebno dodatno opremiti s "pull-up" upori.



Slika 11-16: Programska logika telemetričnega modula.

Za upravljanje servo motorjev uporabimo priključek mikrokrmilnika OC1A, ki strojno podpira izvedbo PWM metode krmiljenja motorjev. Tudi Bascom ima že implementirane metode PWM krmiljenja, vendar pa zaradi nenatančnosti in težje manipulacije niso primerne. PWM krmiljenje zato izvedemo direktno z naslavljanjem registrov priključka OC1A. Prednost povezave vseh servo motorjev na eno upravljalno povezavo je onemogočanje pridobivanja napačnih meritev senzorja zaradi odmeva, ki ga povzroči senzor, ki meri v bližnji smeri. Funkcionalna shema programske logike je predstavljena na sliki 11-16. Informacija o ovirah v bližnji okolici mobilne enote, se izvede na treh postavitvah PWM motorjev. Same lokacije merjenja so bile izbrane tako,

da glede na karakteristiko zajemanja senzorja izpolnjujejo pogoj dovolj velike pokritosti površine kroga z radijem mejnih vrednosti dovoljene bližine predmeta. Ob inicializaciji modula, se registri OC1A priključka nastavijo na vnaprej nastavljene vrednosti, ki senzor postavijo v pozicijo 0 (0°C). Naslavljanje registrov v programskem okolju Bascom, izvedemo na naslednji način:

```
Tccr1a = &HA2
Tccr1b = &H1B
```

```
Tcnt1h = &H00
Tcnt1l = &H00
```

```
Icr1h = &H07
Icr1l = &HF0
```

```
Ocr1ah = &H00
Ocr1al = &H80
```

V vseh nadaljnjih iteracijah pa moramo za spremembo napetosti na priključku OC1A in s tem spremembo pozicije servo motorja, naslavljanje le še gornjih osem bitov (*Ocr1ah*) in spodnjih osem bitov (*Ocr1al*) registra priključka OC1A. Naslavljanje registra izberemo s pomočjo odločitvenega stavka, ki šteje na kateri poziciji smo bili in na katero pozicijo se moramo vrniti. Pozicije in hexa-decimalne vrednosti posameznih registrov 16bitnega OC1A registra, so predstavljene v spodnji tabeli.

Pozicija	Abstrakcija v °C	Vrednost Ocr1ah/Ocr1al
0	5	&H00/&H80
1	90	&H00/&HC0
2	175	&H01/&H00

Tabela 11-1: PWM vrednosti.

Na vsaki poziciji izvedemo tri meritve posameznega senzorja. Tri meritve predstavljajo dovolj veliko število iteracij za dovolj natančen izračun povprečne vrednosti meritve. Pridobivanje meritve z ultrazvočnega senzorja v okolju Bascom, izvedemo na naslednji način:

```
'prvi senzor'
```

```
Pulseout Portc , 5 , 10 //10 mikro sekund dolg pulz
```

```
Waitus 700
```

```
Pulsein Sensor , Pinc , 5 , 1 //pin na katerega je senzor priključen
```

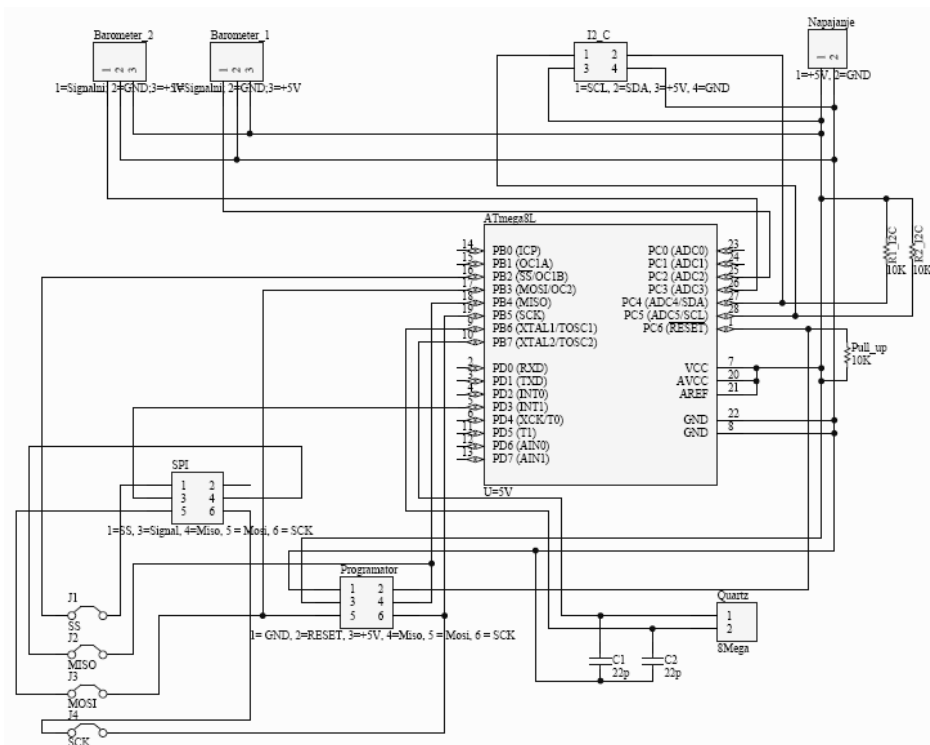
Med proženjem inicializacijskega pulza in pregledom vrnjene vrednosti, mora po specifikaciji senzorjev preteči vsaj 700 mikro sekund. Ta postopek izvedemo za vsak posamezen senzor trikrat in sicer po naslednjem postopku:

1. Izvedi prvo meritev na vseh senzorjih.
2. Izvedi drugo meritev na vseh senzorjih.
3. Izvedi tretjo meritev na vseh senzorjih.
4. Izračunaj povprečno vrednost meritve.

S takšnim pristopom onemogočimo motnje lastnega odboja senzorja, saj med dvema meritvama posameznega senzorja preteče dovolj časa (vsaj 1400 us), da se lahko odbiti signal porazgubi v okolici. Zaradi programske implementacije I2C povezave, zahteva tudi telemetrični modul deklaraciji metod *I2c_master_needs_data* in *I2c_master_has_data*. Glede na ukaz gospodarja, pretvorimo podatke izbranega senzorja najprej v besedo (podatkovni tip "String"). To napravimo s pomočjo metod *Fusing* (Bascom), kjer podamo najprej na koliko decimalnih mest želimo zaokrožiti zajeto vrednost in *Format*, kjer izberemo skupno dolžino podatkovnega tipa (vključno z decimalno vejico) in obliko znakov, ki jih bomo uporabili. Tako pripravljene podatke pošiljamo znak po znak »gospodarju«.

11.2.4 Nadgradnja modula Vremenska postaja

Implementacija modula vremenska postaja II je podobna implementaciji v poglavju 11.2 že opisanega modula telemetričnega modula. Komunikacijska povezava na centralni krmilnik, je ponovno deljena SPI povezava, kjer se SPI povezava hkrati uporablja za reprogramiranje in komunikacijo s krmilnim modulom, ki nastopa v vlogi »gospodarja« komunikacijske povezave. Vezje je podano na sliki 11-18.



Slika 11-17: Vremenska postaja II.

Vremenska postaja II predstavlja izboljšavo modula vremenske postaje, v smislu optimizacije električnih povezav in uporabe boljšega temperaturnega senzorja. Za merjenje temperature smo uporabili Dallasov temperaturni čip DS1631, ki ne le da omogoča opravljanje bolj natančnih meritev, marveč podpira tudi I2C povezavo. Kot novost izvedemo še kreiranje tiskovin, kot predlog za postavitev senzorjev na večji razdalji od mikrokrmilnika. Za implementacijo I2C, ponovno izkoristimo strojno podporo mikrokrmilnika, ki jo izvedemo preko priključkov SDA in SCL (kanalov 4 in 5 ADC pretvorbe). Mikrokrmilnik igra vlogo »gospodarja« povezave. Iz programskega stališča senzor zahteva za implementacijo I2C komunikacije, točno določen protokol komuniciranja:

//Inicializacijski postopek, naslov senzorja je &H92:

1. Ustavi kakršnokoli trajajočo konverzijo (*I2cwbyte &H22*)
2. Nastavi senzor na delovanje v neprekinjenem načinu
 - I2cwbyte &HAC 'komanda kaj naslavljam*
 - I2cwbyte &H02*

3. Nastavi maksimalno in minimalno temperaturo za alarm (maksimalna temperatura se nahaja v dvo-zlogovnem Th registru-komanda na naslovu A1 ; za 70 nastavimo zgornji zlog Th na 46 in spodnji na 00; minimalna temperatura, ki se nahaja na naslovu A2 na 10)

I2cwrite &HA1 *'Th komanda*

I2cwrite &H46 *'poseganje na zgornji byte dvobytnega Th registra*

I2cwrite &H00 *'poseganje na spodnji byte dvobytnega Th registra*

I2cwrite &HA2 *'*

I2cwrite &HB2 *'poseganje na zgornji byte dvobytnega Tl registra*

I2cwrite &H00

4. Zaženi konverzijo

I2cwrite &H92 *'*

I2cwrite &H51

Iz segmentov kode je razvidno, da opredeli temperaturni senzor zaporedno pisanje, kot zapis zgornjega zloga registra na naslov (prvi poseg) in kot zapis spodnjega zloga registra na istem naslovu (zaporedni poseg). Postopek branja izmerjenih čipa vrednosti v času delovanja poteka po naslednji sekvenci:

I2cstart

I2cwrite &H92 *'nastavimo kontrolni bit za pisanje*

I2cwrite &HAA *'naslovimo temperaturni register*

I2cstart

I2cwrite &H93 *'nastavimo kontrolni bit za branje*

I2crbyte T1_msb , Ack

I2crbyte T1_lsb , Nack

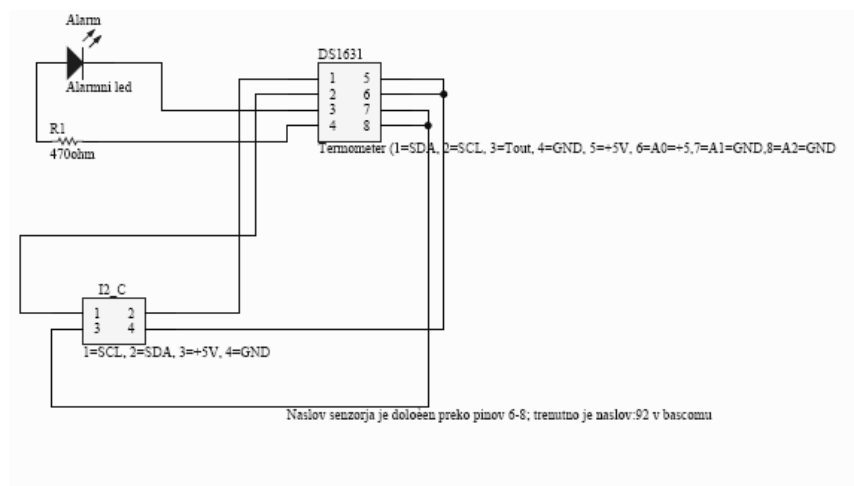
I2cstop

V dobro poznani kodi (glej poglavje 8), lahko poleg ponovitvenega pogoja »start«, spoznamo še ponovitveno branje registra. Z zaporednim podajanjem ukaza *I2crbyte*, najprej preberemo MSB dvo-zlogovnega registra, nato pa še LSB dvo-zlogovnega registra. Obe vrednosti nato združimo po naslednji formuli:

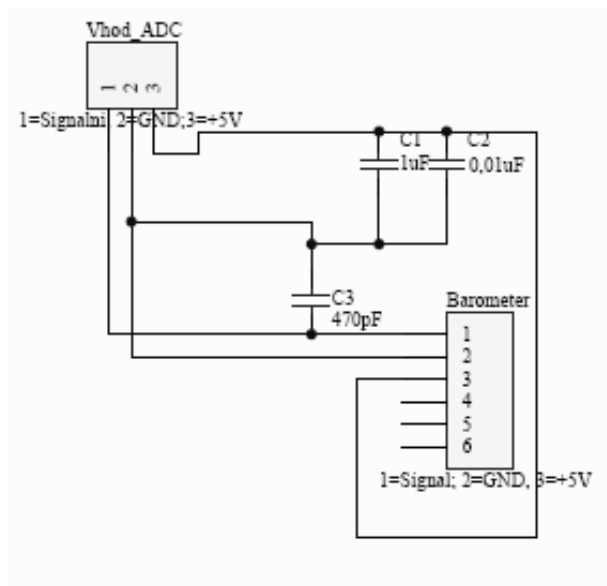
$T1_lsb_s = T1_lsb / 256$ *'decimalna mesta za decimalno vejico*

$T1_calc = T1_msb_s + T1_lsb_s$

Zajemanje meritev barometrov s pomočjo ADC pretvorbe, smo podrobno opisali že v poglavju 8. Podatke tudi ta modul »Vremenska postaja«, vrača v obliki niza znakov dolžine 1. Za prenos posameznega znaka naenkrat smo se odločili, ker je pri prenosu celotne vrednosti spremenljivke navadno prihajalo do napak v prenosu, katerih vzrok je nestabilnost žičnih povezav na razvojni ploščici. Kot dodatno novost smo omenili še predloge za priklop senzorjev. Predstavljene so na naslednjih slikah. Predloga omogoča implementacijo pasivnih elementov in kondenzatorjev na vezju samega sensorja in več direktno na vezju mikrokrmilnika. Izbrani način implementacije ima glavno prednost v možnostih različne uporabe samega vezja. V takšnem primeru nam vezje predstavlja nekakšno nadgradnjo razvojne ploščice, kjer lahko namesto izbranega sensorja priklopimo drugo komponento in ji že dovajamo napajanje. ADC pretvorbo na priključku, ki je spremenil uporabnost, pa s pomočjo programatorskega priklopa reprogramiramo in mu dodelimo želeno obnašanje.



Slika 11-18: Priklop temperaturnega sensorja .



Slika 11-19: Priklop barometra .

Implementacija predlog, omogoča optimizacijo samega vezja (manjša potreba po uporabi dvoslojnih vezij) in hkrati omogoča postavitev senzorja na lokacijo, drugačno od lokacije mikrokrmilnika, ki upravlja s senzorjem. Fizična postavitev predloge je ponovno odvisna predvsem od kvalitete povezovalnih priklopnih linij, saj le-te s svojimi električnimi karakteristikami vplivajo na kvaliteto vhodnega in izhodnega signala. Pri implementaciji omrežja I2Cpovezav, moramo biti še posebej pozorni na skupno dolžino vseh linij v posamezni I2C povezavi. Dolžina lahko preseže mejo primernosti uporabe $10\text{k}\Omega$ uporov in je njihove vrednosti potrebno preračunati. Kot sem že omenil, je izračun velikosti uporov zaradi velikega števila parametrov, ki imajo vpliv problematičen.

12 APLIKATIVNI SCENARIJI UPORABE SISTEMA »DATA«

V okviru razvoja sistema DATA, smo predvideli in izvedli že nekaj aplikativnih scenarijev uporabe, s katerimi smo želeli preveriti fleksibilnost in učinkovitost predstavljene kognitivne arhitekture »DATA« za inteligentna okolja. Kognitivni distribuiran sistem DATA, je zgrajen na osnovi predstavljenega porazdeljenega programskega ogrodja DATA, s katerim je moč komunicirati preko adaptivnega uporabniškega vmesnika. To pomeni, da se na vseh programskih gradnikih sistema DATA, uporabniški vmesnik implementira na identičen način in da se prilagoditev vmesnika izvede glede na scenarij, ki ga uporabnik želi izvajati. Implementacija uporabniškega vmesnika sistema DATA, predstavlja tako osnovo adaptivnemu multimodalnemu vmesniku (HMI) in hkrati omogoča nadaljnji razvoj na področju kontekstnega prilagajanja sistema trenutnim zahtevam tudi večih uporabnikov sočasno. Slika 12-1 prikazuje dani uporabniški vmesnik, ki zajema vmesnik za upravljanje z vsemi gradniki sistema DATA. V primeru uporabe DATA modularnega strežnika lociranega na mobilni robotski enoti, lahko z danim grafičnim vmesnikom takoj upravljamo s strojnim delom mobilne robotske enote in sicer govorno, preko tipkovnice, z uporabo miške, daljinskega upravljalca, igralnega ploščka ali igralne palice. Vmesnik hkrati prikazuje tako meritve kot tudi pošiljanje ukazov, ki jih razume mobilna robotska enota »DATA«.



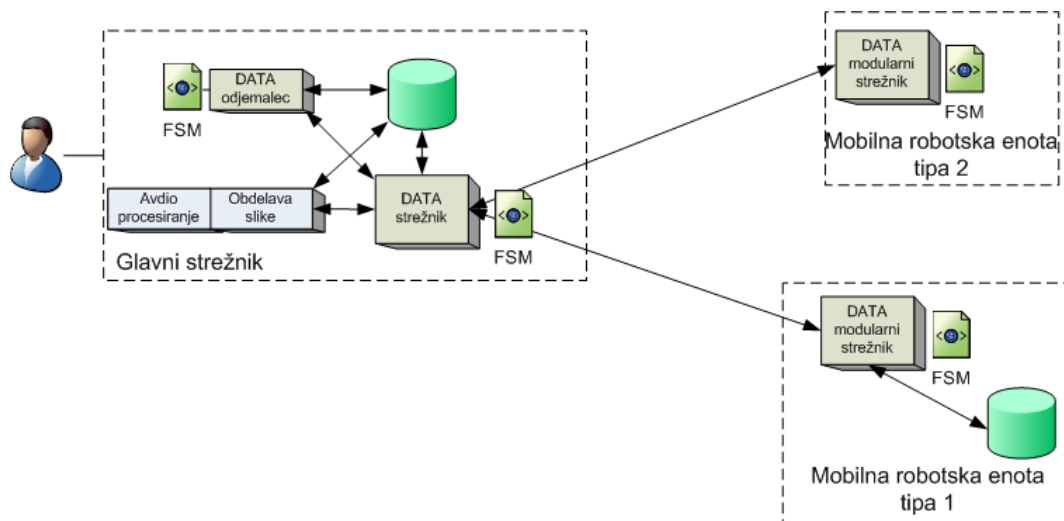
Slika 12-1: Grafični vmesnik sistema DATA.

Vmesnik na sliki 12-1, je razdeljen na tri logične enote. **Prva enota** (črni pravokotnik) je namenjena prikazovanju statusnih sporočil distribuiranega sistema DATA. Statusna sporočila so sporočila o napakah, sporočila o meritvah, zgradba RS232 ali TCP/IP paketa, ki se sprejema/oddaja itn. **Druga enota** grafičnega vmesnika, je namenjena samemu krmiljenju mobilne robotske enote. V primeru uporabe vmesnika na DATA modularnem strežniku, gre za direktno krmiljenje, v primeru inicializacija na DATA odjemalcu pa za oddaljeno krmiljenje naprav v inteligentnem okolju. Trenutno je možno mobilno enoto krmiliti s tipkovnico, z igralnim ploščkom ali igralno palico, z miško in tudi govorno (ASR). Tri podenote druge enote, so namenjene prikazu posameznih krmilnih sporočil (npr. razpoznana beseda pri govornem krmiljenju ali pritisnjena tipka). Na gornji sliki je prikazana oblika grafičnega vmesnika v primeru krmiljenja preko igralnega ploščka, pri čemer lahko hkrati izvajamo krmiljenje tudi z miško, pri čemer s klikom na ustrezno mesto v okviru vmesnika prožimo izbran ukaz. **Tretja enota** vmesnika je namenjena upravljanju z RTP protokolom oziroma prenosom avdio/video informacije. V poglavju 6, smo že spoznali RTP protokol in pripadajočo podporo z JMF in namenskim grafičnim vmesnikom. Le-ta je kot vtičnik (*plugin*), uporabljen tudi v predstavljenem grafičnem vmesniku sistema DATA na sliki 12-1. RTP vmesnik nam omogoča zajemanje, predvajanje in/ali razpošiljanje avdio/video informacije, zajemanje slik ali shranjevanje zajetega avdio/vidoa (prejetega v obliki RTP toka ali zajetega lokalno s pomočjo USB kamere) v datoteko.

12.1 Oddaljeno upravljanje z mobilno robotsko enoto

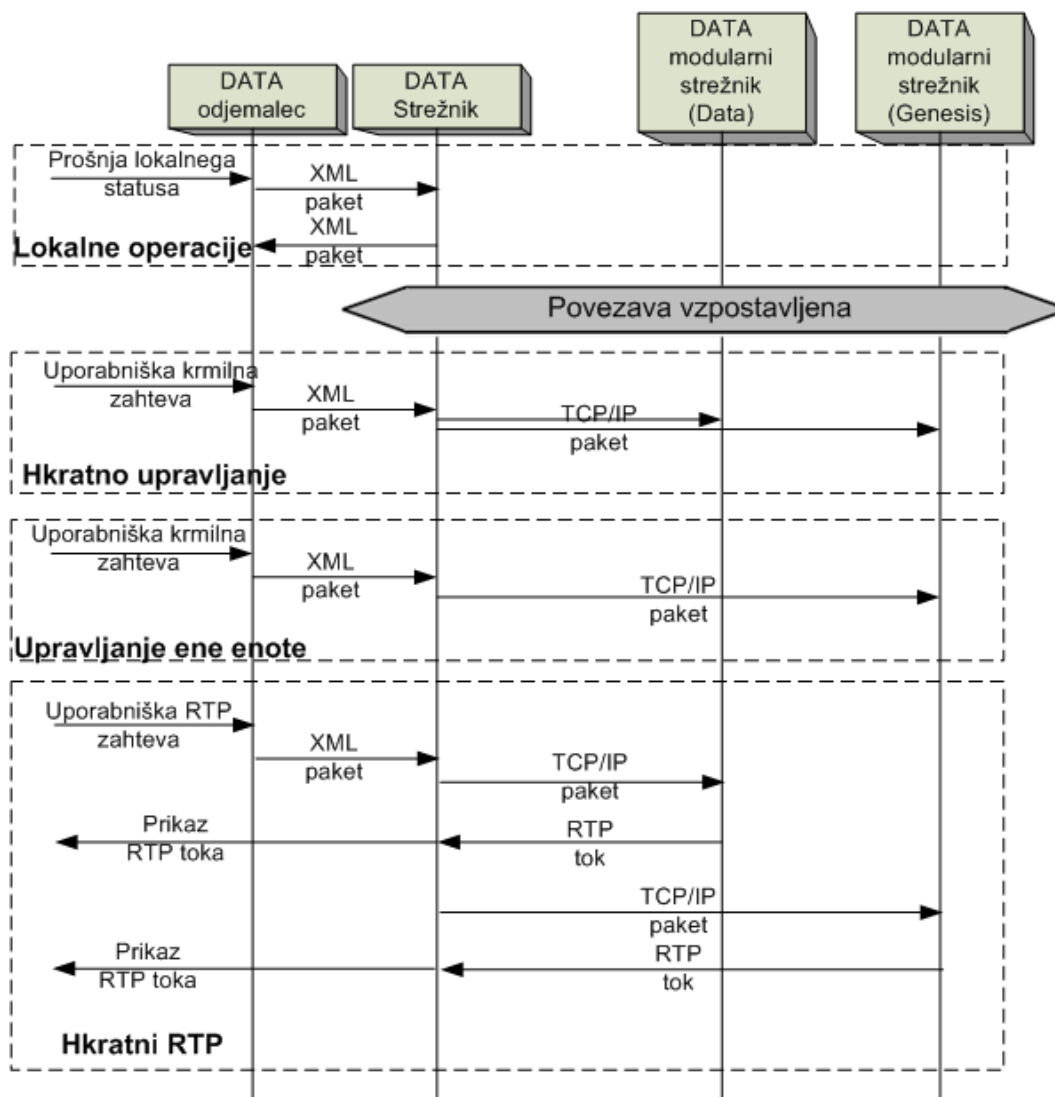
Predstavljeni grafični vmesnik na sliki 12-1, predstavlja vez med sistemom DATA in uporabnikom. Ena izmed funkcionalnosti sistema DATA je lokalno in oddaljeno upravljanje mobilnih robotskih enot oziroma naprav v inteligentnem okolju. Za ta scenarij uporabe, v osnovi potrebujemo DATA strežnik, DATA modularni strežnik in DATA odjemalec. Na vseh se moduli inicializirajo z vmesnikom, ki je predstavljen na sliki 12-1. Slika 12-2 predstavlja gradnike, ki se uporabljajo pri takšnem scenariju uporabe. Ob inicializaciji sistema, DATA odjemalec posreduje DATA strežniku, želeni način obnašanja sistema v obliki XML opisa. Ker XML opis vsebuje sodelovanje dveh mobilnih enot, DATA strežnik iz prejetega opisa obnašanja, formira svoj opis obnašanja

za vsak posamezen DATA modularni strežnik, ki je implementiran na mobilni robotski enoti. DATA strežnik vzpostavi rezervirano povezavo do obeh DATA modularnih strežnikov, ki se ohrani za čas trajanja dane komunikacijske seje.



Slika 12-2: Scenarij uporabe sistema: oddaljeno krmiljenje.

Kot je razvidno iz slike, scenarij predvideva uporabo dveh mobilnih enot: mobilne robotske enote DATA (mobilna enota tipa 1) in mobilne robotske enote GENESIS (mobilna enota tipa 2). Programski gradniki, ki usmerjajo delovanja sistema so: DATA odjemalec, s katerim upravlja uporabnik na glavnem strežniku; DATA strežnik, ki skrbi za vzpostavitev in komunikacijo med uporabnikom in mobilno robotsko enoto; DATA modularni strežnik, ki skrbi za upravljanje z dodeljeno mobilno robotsko enoto. Na sliki so predstavljeni še procesi, ki jih lahko posamezna enota v okviru danega scenarija oz. konfiguracije sistema, lahko opravlja. Tako lahko glavni strežnik (*Main frame*) opravlja porazdeljeno procesiranje slike in govora/zvoka in upravlja s skupno podatkovno bazo celotnega sistema. Mobilna enota tipa 1, lahko tudi ima svojo podatkovno bazo npr. za procese in meritve, ki se izvajajo na njej. Potek komunikacije predstavljamo na spodnjem diagramu. Med programskima gradnikoma, ki sta inicializirana fizično na isti mobilni robotski enoti, se komunikacijski paketi izmenjujejo v obliki XML paketov, komunikacijski paketi med oddaljenimi gradniki pa se pošiljajo v okviru TCP/IP sej v obliki TCP/IP paketov.



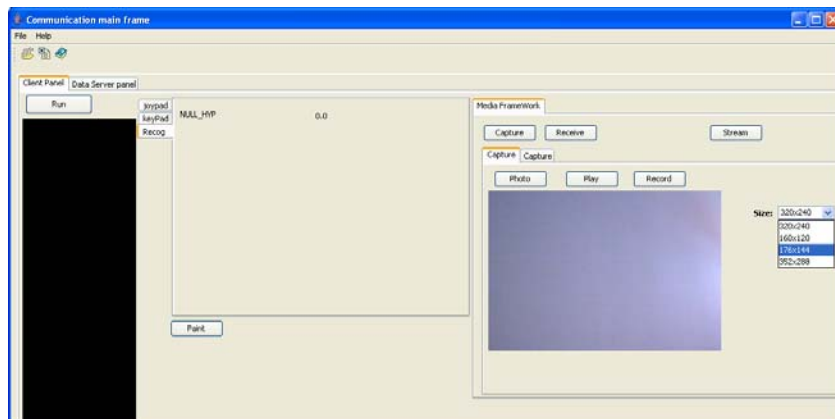
Slika 12-3: Komunikacijski diagram v primeru scenarija oddaljenega krmiljenja.

Lokalna komunikacija (Local operations) zajema vse akcije, ki izvajajo izmenjavo paketov med programskima gradnikoma DATA odjemalcev in DATA strežnikom. Med lokalno komunikacijo tako štejemo inicializacijski postopek, ko DATA odjemalec posreduje XML opis želenega opisa obnašanja sistema DATA strežniku, branje in zapisovanje v podatkovno bazo (MySQL), posredovanje ukazov, ki naj jih izbrani DATA modularni strežnik opravi, itn. DATA odjemalec ne more neposredno komunicirati s katerikoli oddaljenim DATA modularnim strežnikom, saj je vsa komunikacija izvedena preko DATA strežnika, ki gosti povezave do modularnih strežnikov. *Pogoj vzpostavljene povezave* pomeni, da se je rezervirana TCP/IP seja

uspešno vzpostavila in je DATA modularni strežnik prejel XML opis želenega obnašanja sistema, ter ga tudi začel izvajati. Šele po vzpostavljeni seji, lahko uporabnik preko vmesnika na DATA odjemalcu upravlja z ostalimi komponentami v sistemu, ki niso direktno povezane na mobilno robotsko enoto na kateri uporabljamo grafični vmesnik. Uporabnik lahko krmili eno ali več mobilnih robotskih enot hkrati. V primeru hkratnega krmiljenja več mobilnih robotskih enot hkrati, se v XML paketu v polju *Receiver*, dodajo identifikatorji enot, ki jih želimo krmiliti (identifikatorji so lahko imena ali IP naslovi). DATA strežnik prejeti XML paket razčleni in preko vzpostavljene rezervirane TCP/IP seje v obliki TCP/IP paketov posreduje nadaljnje akcije naslovnim DATA modularnim strežnikom. V našem primeru posreduje krmilni TCP/IP paket DATA modularnemu strežniku na mobilni enoti DATA in mobilni enoti Genesis. Dalje lahko uporabnik preko RTP protokola, *oddaljeno zajema* sliko iz naprav za zajemanje, ki so locirane na posameznih mobilnih robotskih enotah. RTP sejo z eno ali več mobilnih enot hkrati, DATA odjemalec odpre na naslednji način:

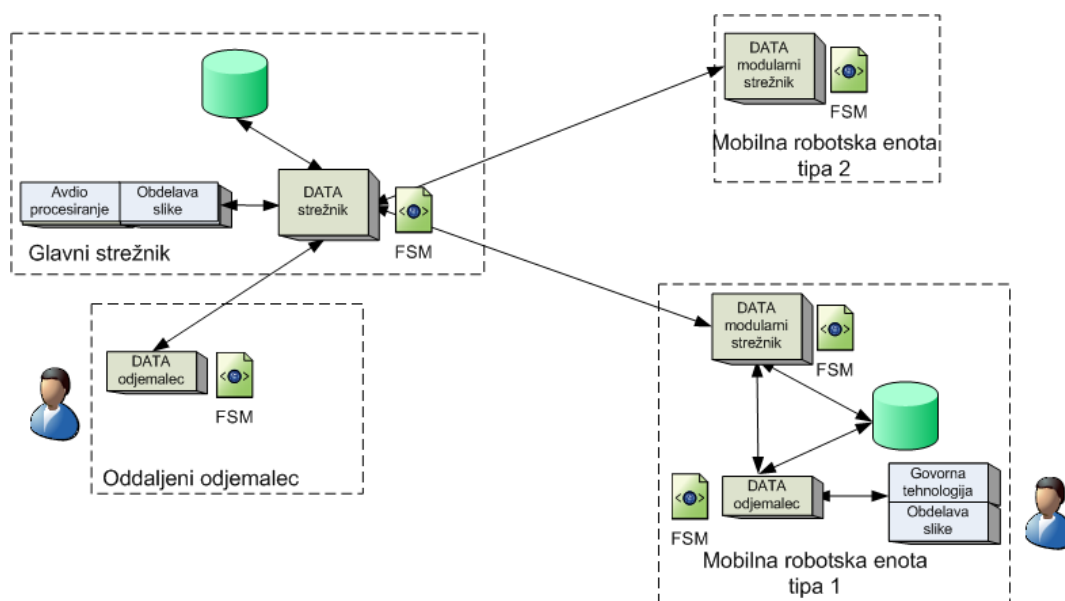
1. Pošlji XML paket, ki vsebuje ime akcije za sprejemanja oddaljenega RTP toka na DATA strežnik. Le-ta izvede akcijo, s katero odpre RTP sejo in začne poslušati na predpisanem IP naslovu in izbranih UDP vratih..
2. DATA strežnik posreduje oddaljenim DATA modularnim strežnikom TCP/IP paket, ki vsebuje naslov prejemnika RTP toka (to je DATA odjemalca).
3. Oddaljeni DATA modularni strežniki pričnejo z oddajanjem RTP toka na sprejeti naslov prejemnika RTP toka.

Na sliki 12-4, je prikazano stanje grafičnega vmesnika po prejetju RTP toka. Ker prejemamo RTP tok hkrati iz dveh naprav, se v vmesniku prikažeta dva zavihka. Na vsakem izmed zavihkov lahko z avdio/video informacijo manipuliramo ločeno in popolnoma neodvisno.



Slika 12-4: Sočasno prejetje dveh RTP tokov.

Izvedba predstavljenega scenarija ni pogojena z lokacijo uporabnika. Uporabnik lahko upravlja z mobilnimi robotskimi enotami preko DATA odjemalca lociranega tudi na isti enoti kot je DATA strežnik (primer navidezne TCP/IP povezave) ali pa tudi iz katerekoli druge enote, ki je sposobna poganjati DATA odjemalec in lahko gosti TCP/IP povezavo. DATA odjemalec se mora za pravilno delovanja opisanega scenarija nujno povezati na DATA strežnik.



Slika 12-5: Scenarij sočasne uporabe sistema DATA.

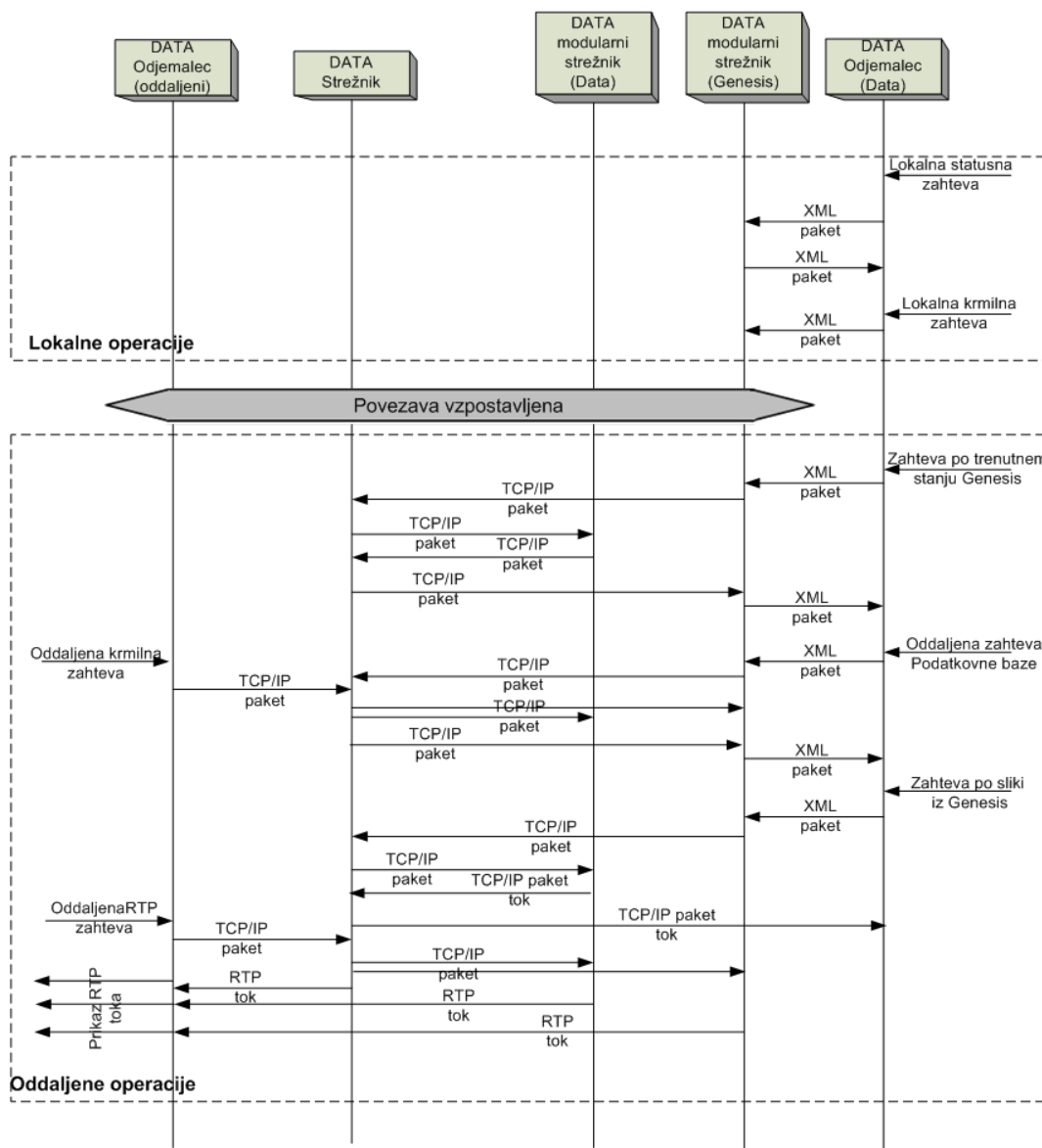
12.2 Sočasna komunikacija z mobilnimi robotskimi enotami

Sistem DATA omogoča sočasno krmiljenje naprav več uporabnikom. Za izvedbo sočasnega krmiljenja potrebujemo vsaj dva programska gradnika DATA odjemalec, ki se prav tako povezuje na DATA strežnik in posredno na katerekoli DATA modularne strežnike. Upravljanje lahko izvajamo direktno na mobilni robotski enoti ali pa oddaljeno na kateremkoli terminalu, ki omogoča poganjanje DATA odjemalca. Na sliki 12-5 je prikazan scenarij sočasne uporabe. Informacija, ki jo vmesnik lahko zajema je odvisna od implementacije enote, s katero uporabnik upravlja. Predvidena informacija se podaja v obliki avdio/vizualne informacije ali sporočil o stanju enote v obliki senzorskih meritev ali statusne informacije.

Za ta scenarija uporabimo naslednje programske gradnike sistema DATA:

- *DATA Strežnik*: podpira avdio procesiranje, procesiranje slike in upravlja s skupno podatkovno bazo.
- *DATA odjemalec*: razpolaga z grafičnim uporabniškim vmesnikom in podpira upravljanje preko tipkovnice, igralnega ploščka ali palice, zaslona občutljivega na dotik, miške ali govora.
- *DATA modularni strežnik*: neposredno upravlja s strojnimi komponentami na mobilni robotski enoti.

V predstavljenem scenariju sodelujeta sočasno dva uporabnika, ki ob inicializaciji DATA odjemalca, DATA strežniku oz. DATA modularnemu strežniku pošljeta preko dveh v splošnem različnih XML opisov želeno obnašanje sistema DATA, za vsakega od njiju. Za namene lokalnega upravljanja, se implementira navidezna TCP/IP povezava med DATA odjemalcev in DATA modularnim strežnikom. Oddaljeno upravljanje z mobilno robotsko enoto, poteka posredno preko DATA strežnika. Primer komunikacije med posameznimi gradniki je podan na sliki 12-6. Oba uporabnika lahko tako sočasno upravljata s sistemom DATA in izvajata različne akcije znotraj sistema inteligentnega okolja.



Slika 12-6: Komunikacijski diagram sočasnega upravljanja v sistemu DATA.

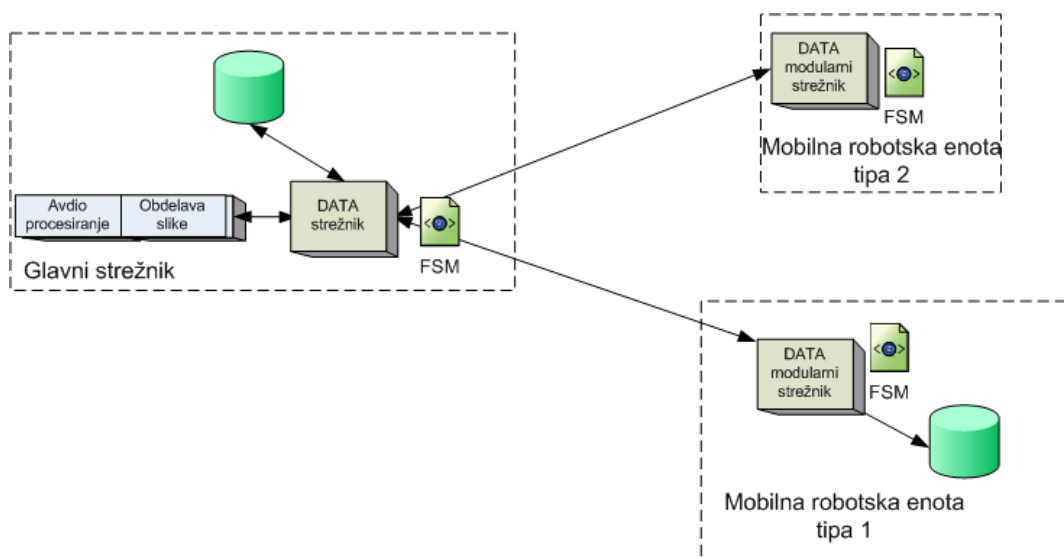
Podobno, kot smo omenili že v prejšnjem poglavju, poteka lokalna komunikacija med DATA odjemalcem in DATA modularnim strežnikom, v obliki XML paketov. Uporabnik lahko enoto krmili ali pa zahteva statusna sporočila. Ob implementaciji RTP paketa v okviru DATA modularnega strežnika, pa lahko uporabnik še zajema, predvaja in manipulira z zajeto avdio/video informacijo. Lokalne operacije so določene z izvedbo same mobilne robotske enote. V našem primeru en uporabnik uporablja vmesnik na mobilni enoti tipa 2 in zato lahko poleg krmiljenja, uporablja še funkcionalnosti v povezavi z RTP tokovi in izvaja operacije za delo z lokalno podatkovno bazo. RTP

tokove lahko zajema tudi oddaljeni uporabnik, ne more pa posredovati avdio/video informacije drugim enotam v DATA sistemu.

V predstavljeni izvedbi scenarija, lahko oba uporabnika opravljata oddaljeno krmiljenje sočasno. Uporabnik na oddaljenem odjemalcu uporablja programski gradnik DATA odjemalec, ki je preko TCP/IP povezave povezan na DATA strežnik. Na podoben način, kot opisano podpoglavju 12.1, nato upravlja s sistemom DATA. Uporabnik, ki uporablja DATA odjemalec na mobilni robotski enoti, lahko v okviru rezervirane dvosmerne povezave med DATA strežnikom in DATA modularnim strežnikom, opravlja omejeno upravljanje z napravami priklopljenimi na sistem DATA. Navadno te operacije zajemajo poizvedbe v skupni podatkovni bazi, pregledi statusov in zajetja meritev posameznih enot sistema DATA, itn. Komunikacija med DATA odjemalcem na mobilni enoti in DATA strežnikom, poteka na podoben način kot med DATA odjemalcem na glavnem strežniku in DATA strežnikom. Komunikacijski posrednik je v tem primeru DATA modularni strežnik, katerega naloga je XML pakete oblikovati kot TCP/IP pakete in jih posredovati DATA strežniku oz. prejete TCP/IP pakete pretvoriti v XML pakete in jih posredovati lokalnemu DATA odjemalcu. V primeru, ko želi uporabnik na mobilni enoti komunicirati ali upravljati z drugo mobilno robotsko enoto, mora vse ukaze posredovati DATA strežniku, ki informacijo posreduje DATA modularnem strežniku na ciljni mobilni robotski enoti. Posebnost sočasnega krmiljenja so varovalni mehanizmi, ki preprečujejo anomalije, ki jih takšen način upravljanja s sistemom prinaša. V primeru nasprotujočih si sočasnih ukazov, do katerih lahko potencialno prihaja pri krmiljenju določene mobilne robotske enote v DATA sistemu, primarni varovalni mehanizem implementira DATA modularni strežnik. Prioriteto krmiljenja ima vedno lokalni uporabnik. V primeru, da DATA modularni strežnik prejme ukaz krmiljenja lokalnega uporabnika in v določenem časovnem obdobju še ukaz oddaljenega uporabnika, DATA modularni strežnik vedno zavrže ukaz oddaljenega uporabnika. Podoben varovalni sistem je razvit tudi na DATA strežniku, ki deluje po principu izvajanja ukaza tistega, ki prvi ukaz poda. DATA strežnik pa vedno ne zavrže celotnega ukaza ampak le segment ukaza, ki lahko povzroči konfliktno situacijo (v primeru ko je ukaz namenjen več naslovnikom hkrati).

12.3 Nenadzorovano delovanje sistema DATA

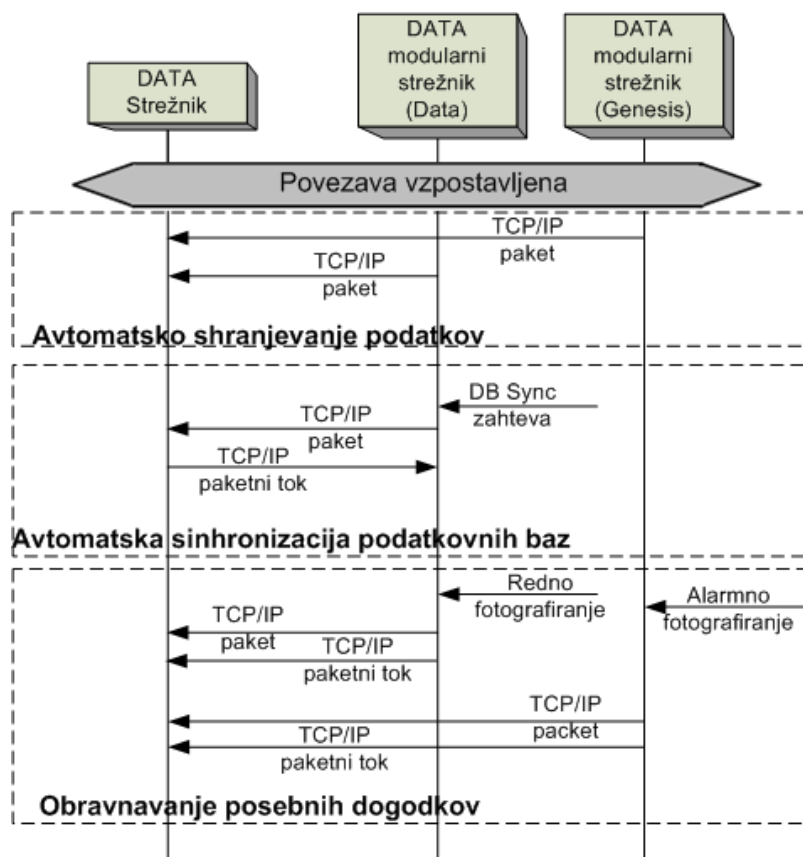
Sistem DATA omogoča delovanje brez posebnega nadzora oz. direktnega upravljanja. Uporabnik lahko DATA strežniku poda željen scenarij uporabe, po katerem DATA strežnik prevzame vlogo upravljalca celotnega sistema in samostojno izvaja tiste akcije, ki jih zahteva dani scenarij. Kot smo že omenili, se opis delovanja sistema poda v obliki XML zapisov. Diagram na spodnji sliki predstavlja primer implementacije takšnega sistema.



Slika 12-7: Scenarij avtonomnega delovanja sistema DATA.

Ob implementaciji takšnega scenarija sistem DATA opravlja svoje naloge povsem avtonomno in brez dodatne uporabniške interakcije. Kot je razvidno iz slike, smo za ta scenarij v sistem vključili dve mobilni robotski enoti, ki gostita vsaka po en DATA modularni strežnik in en glavni računalnik, ki gosti DATA strežnik. Komunikacijski diagram danega scenarija je prikazan na sliki 12-8. DATA strežnik kot upravljevec sistema, upravlja z vsemi komponentami DATA sistema ali opravlja nadzor nad njihovim delovanjem. Enega izmed procesov nadzora predstavlja tudi preverjanje pravilnosti delovanja komponent. V obliki TCP/IP paketov, DATA strežnik posreduje ukaze za preverjanje vsem DATA modularnim strežnikom in pričakuje njihov specifičen odziv (glede na tipe komponent, ki jih modularni strežnik upravlja in glede na ukaz, ki ga DATA strežnik posreduje). V primeru napake, DATA strežnik o napaki

in kritičnosti napake obvesti ustreznega uporabnika in dani modul označi za okvarjenega.



Slika 12-8: Komunikacijski diagram avtonomnega delovanja.

V komunikacijskem diagramu, smo posebej izpostavili tri akcije, ki jih sistem opravlja v avtonomnem načinu delovanja. *Avtonomno shranjevanje* v podatkovno bazo predstavlja proces zajemanja informacije na oddaljeni enoti (DATA ali Genesis), posredovanje te informacije iz DATA modularnega strežnika na DATA strežnik v obliki TCP/IP paketa in shranjevanje prejete informacije v skupno podatkovno bazo, s katero upravlja DATA strežnik. V primeru, da mobilna robotska enota uporablja tudi lokalno bazo, senzorske podatke sočasno vpisuje tudi v svojo podatkovno bazo. Kadar mobilna robotska enota uporablja lokalno podatkovno bazo, lahko v okviru avtonomnega delovanja zahteva ali prejema ukaze sinhronizacije podatkovnih baz. Sinhronizacija podatkovne baze pomeni izmenjavo podatkovnih vnosov v podatkovni bazi, ki se navezujejo na mobilno robotsko enoto, ki je sinhronizacijo zahtevala. Tako

mobilna robotska enota DATA prejme v postopku sinhronizacije le tiste podatke, ki zadevajo njeno delovanje. Če v podatkovni bazi enote vnosa ni, ga enota vnese, v primeru pa da je vnos v bazi podatkovne enote drugačen od prejetega sinhronizacijskega paketa, enota zahteva popravek vnosa v skupni bazi. Zahteva po popravku se izvede v obliki povratnega TCP/IP paketa, ki kot akcijo predvideva posodabljanje določenega vnosa. DATA Strežnik upravlja tudi s posebnimi dogodki sistema. Med posebne dogodke sistema lahko štejemo npr. zaznavanje gibanja, redno zajemanje slik posameznih enot in alarmna sporočila. Vsa komunikacija, ki se nanaša na posebne dogodke v sistemu, se DATA strežniku posreduje v obliki TCP/IP paketov. Glede na vsebino paketa, se DATA strežnik odloči, ali je potrebno izvesti.

13 SKLEP

V okviru predstavljenega diplomskega dela je bila izvedena analiza tehnologij in pristopov, ki omogočajo razvoj sistemov sposobnih kontekstnega procesiranja dogodkov v okolici in po drugi strani generiranje kognitivnih odzivov. Inteligentni sistemi s takšnimi lastnostmi postavljajo uporabnike v ospredje in se ne ukvarjajo zgolj z zajemanjem informacije, avtomatskim ali ukaznim krmiljenjem. Področje raziskav je bilo usmerjeno predvsem v tako imenovane multimodalne tehnologije (npr. zajemanje avdio/video informacije, procesiranje avdio/video informacije in generiranje multimodalnih odzivov), umetno inteligenco (npr. statistično modeliranje in tvorjenje uporabniških profilov) in robotiko (npr. mobilna robotska enota, v celoti samostojno gibanje enote v danem prostoru in multimodalnost robotskih enot). Na osnovi arhitektur inteligentnih okolij znanih EU projektov, smo postavili v izhodišču lastno zasnovo kognitivnega inteligentnega okolja in zasnovo multimodalnega vmesnika človek-stroj (HMI). Zadnjega lahko vključujemo ali na samih mobilnih robotskih enotah, fiksno na glavnem strežniku ali terminalih, ali celo distribuirano znotraj celotnega sistema DATA. V okviru diplomske naloge smo nato realizirali samo mobilno robotsko enoto, pri čemer smo natančno predstavili vse strojne dele, ki trenutno sestavljajo mobilno robotsko enoto DATA. Realizirali smo tudi distribuirano programsko ogrodje DATA, ki omogoči fleksibilno in učinkovito sestavo poljubne konfiguracije posameznih modulov (odjemalec, strežnik in modularni strežniki), ter vključevanje poljubnega števila naprav in mobilnih robotskih sistemov. Podobno kot inteligentni sistemi, ki smo jih predstavili kot izhodišče za naše raziskave DATA programsko ogrodje podpira modularno strukturo in enostavno vgradnjo novih sistemskih funkcionalnosti. Za razliko od arhitektur predstavljenih v okviru EU projektov AMI, Chil ter Amigo, arhitektura sistema DATA prinaša manjše soodvisnosti v delovanju posameznih modulov in zato manj kompleksno zasnovo arhitekture, ki pa kljub temu omogoča zaradi preglednosti, fleksibilnosti in uporabljenega formalizma s končnimi stroji, izvajanja še bolj kompleksnih nalog. Ena glavnih prednosti sistema DATA je tako praktično neomejena možnost uporabe predstavljenega ogrodja za različne namene in v različnih situacijah in kontekstih. Ker so glavni gradniki programskega ogrodja DATA (DATA odjemalec, DATA strežnik in DATA modularni strežnik) dogodkovno krmiljeni modularni sistemi, ki svoje delovanje implementirajo

na podlagi XML opisov, lahko ti gradniki podpirajo sočasno uporabo ter konfiguracijo obnašanja sočasno za poljubno število uporabnikov. Prav zmožnost sočasnega, včasih nasprotnojučega si obnašanja, ki ga uporabniki zahtevajo in pričakujejo od sistema (to lahko dosežemo brez potrebe po ponovnem zagonu sistema), odpira možnosti implementacije avtonomnega prilagajanja obnašanja in videza sistema DATA trenutnim zahtevam posameznih uporabnikov. Dodatno programsko ogrodje DATA uvaja v inteligentna okolja enoumno učinkovito komunikacijo med gradniki in enotami sistema DATA. Izvedba sistema na osnovi Java programskega jezika, prinaša DATA sistemu neodvisnost od uporabljene platforme (Windows, Linux) ter preprostejše nadaljnje nadgradnje sistema.

V okviru implementacije mobilnih robotskih enot, smo izdelali tehnični načrt izvedbe mobilne robotske enote DATA. Z uporabo mikrokrmilnikov smo izvedli krmilni sistem in zajemanje senzorskih podatkov. Strojne module mobilne robotske enote DATA, smo uspešno povezali z razvitimi programskimi moduli sistema DATA in izvedli: prikazovanje senzorskih podatkov na grafičnem vmesniku in shranjevanje v podatkovno bazo, krmiljenje mobilne robotske enote s pomočjo govora, miške in tipkovnice, igralnega ploščka in daljinskega upravljalnika. V kontekstu krmiljenja smo uspešno izvedli testiranje varnostnih mehanizmov enote, ki v primeru bližine ovire enoti ne dopustijo gibanja v smeri ovir. Če se ovira umakne, lahko enota samodejno nadaljuje po dani poti.

V okviru diplomske naloge smo predstavili tudi zasnovo multimodalnega vmesnika (HMI), pri katerem smo trenutno izvedli testiranja predvsem z uporabo razpoznavalnika govora izoliranih besed. Nalogo smo zaključili z nekaj primeri možnih konfiguracij predlaganega kognitivnega sistema DATA, ki smo jih tudi preizkusili v dani kombinaciji z mobilnimi robotskimi platformami DATA in Genesis. V bližnji prihodnosti bodo tekale aktivnosti predvsem v nadaljnjem razvoju predlagane zasnove multimodalnega vmesnika z analizo neverbalnih komunikacij na vhodni strani in na izhodu s sintezo verbalnih in neverbalnih komunikacij v kontekstu bolj naravnih inteligentnih avatarjev.

Sistem DATA, ki smo ga predstavili v diplomskem delu, predstavlja dobro osnovo za nadaljnji razvoj kognitivnih inteligentnih okolij za razne namene uporabe. Njegove

največje pomanjkljivosti v tej fazi razvoja, obstojijo predvsem zaradi ne implementiranih osnovnih funkcionalnosti, ki jih mora inteligentno okolje zagotoviti. Sistem DATA že podpira poleg tradicionalnih vmesnikov tudi govorno krmiljenje, ne podpira pa še procesiranje ostalih modalnosti (neverbalnih) in tudi pogovornih avatarjev. V okviru naloge smo tudi že uspelo implementirati osnovne lastnosti umetne inteligence. V razvoju sistema DATA nismo posvečali večje pozornosti zagotavljanju varnosti na omrežju (s stališča vdorov v sistem), niti posledic, ki bi jih morebitni vdori lahko povzročili. V povezavi s tem bo tako potreben še dodaten napor v zagotavljanje varovanja osebnih podatkov in možnosti hranjenja le-teh na nekem centraliziranem sistemu tudi v primeru več uporabnikov.

LITERATURA

- [1] http://en.wikipedia.org/wiki/Artificial_intelligence
 - [2] http://en.wikipedia.org/wiki/Cognitive_science
 - [3] <http://www.cogsci.ucla.edu/Background/>
 - [4] http://en.wikipedia.org/wiki/Multimodal_interaction
 - [5] Mozer, 1998; Mozer & Miller, 1998;
 - [6] Mozer, Vidmar, & Dodier, 1997 .
 - [7] (http://www.cse.unsw.edu.au/~claude/thesis_topics/intelligent_environment.html)
 - [8] <http://www.hitech-projects.com/euprojects/amigo/>
 - [9] <http://chil.server.de/servlet/is/101/>
 - [10] <http://www.amiproject.org/>
 - [11] DIVINES <http://www.divines-project.org/divines/public/html/home.html>
 - [12] <http://emotion-research.net/>
 - [13] NICE: <http://www.niceproject.com/>
 - [14] <http://www.nislab.no/>
 - [15] SmartKom, Rabiner and Juang, 1993; Young, et all., 2000; Young, 2000
 - [16] SmartKom, Hunt and Black, 1996
 - [17] SmartKom, Scherer, 2003,
 - [18] SmartKom, Weber et al., 2002
 - [19] SmartKom, Sagisaka, Campbell and Higuchi, 1998
 - [20] SmartKom, Hirose, Sato and Minematsu, 2004
 - [21] SmartKom, Rabiner, 1993
 - [22] SmartKom, Wessel et all, 2001
 - [23] SmartKom, Noguerras et al., 2001
 - [24] SmartKom, Noam, 2001
 - [25] SmartKom, Young and Ellis, 1989
 - [26] SmartKom, Kotsia et al., I. Kotsia, andI. Pitas, 2005
 - [27] *Krmiljenje naprav v inteligentnih okoljih ob uporabi govorne tehnologije*, Aleš Guzej, 2007
 - [28] http://www.smartkom.org/start_en.html
-

-
- [29] SmartKom, Wolfgang Wahlster, 2006
- [30] SmartKom, Allen et al. (2000)
- [31] SmartKom, Bordegoni et al. (1997)
- [32] SmartKom, Elting et al. (2003)
- [33] SmartKom, Feiner et al. (1991)
- [34] SmartKom, Hill et al. (1992)
- [35] SmartKom, Maybury in Wahlster (1998a)
- [36] SmartKom, Pfaff, 1985
- [37] SmartKom, Waktster et al., 2001
- [38] ATMEGA 8
http://www.atmel.com/dyn/products/product_card.asp?family_id=607&family_name=AVR%AE+8%2DBit+RISC+&part_id=2004
- [39] BASCOM <http://www.mcselec.com/>
- [40] MYSQL <http://www.mysql.com/>
- [41] SQL <http://en.wikipedia.org/wiki/SQL>
- [42] EUROP <http://www.robotics-platform.eu/>
- [43] MERINO <http://www98.griffith.edu.au/dspace/bitstream/10072/8930/1/22623.pdf>
- [44] AutoCad <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2704278>
- [45] MATLAB www.mathworks.com/products/matlab/
- [46] Zdravko Kačič, Zapiski predavanj Komunikacija človek-stroj, Univerza v Mariboru
- [47] COGNITION: MODELS AND PARADIGMS A preparatory workshop for EU Seventh Research Framework Programme Luxembourg, 20-21 March 2006
- [48] Nickel, K. and Stiefelhagen, R.: Realtime Person Tracking and Pointing Gesture Recognition for HumanRobotInteraction, International Workshop on HumanComputer Interaction, HCI 2004, May 2004, Prague (in conjunction with ECCV 2004). HumanRobot Interaction, International Workshop on HumanComputer Interaction, HCI 2004, May 2004, Prague (in conjunction with ECCV 2004)
- [49] Yang, M., H., Ahuja, N., Tabb, M., 2002, "Extraction of 2D Motion Trajectories and Its Application to Hand Gesture Recognition," IEEE Trans. Pattern Analysis Machine Intelligence, vol. 24, no. 8, pp. 10611074.
- [50] Donato, G., M.S. Bartlett, J.C. Hager, P. Ekman, T.J. Sejnowski, "Classifying Facial Actions", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 21, No. 10, pp. 974989, 1999.
-

-
- [51] Lien , J.J., T. Kanade, J.F. Cohn, CC. Li, "Automated Facial Expression Recognition Based on FACS Action Units", Proc. Third IEEE Int. Conf. Automatic Face and Gesture Recognition, pp. 390395, 1998
- Tian , YL., T. Kanade, J.F. Cohn, "Recognizing Action Units for Facial Expression Analysis", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 23, No. 2, pp. 97115, 2001
- [52] Imagawa, K., Lu, S., Igi, S., 1998, "ColorBased Hand Tracking System for Sign Language Recognition," Proc. Int'l Conf. Automatic Face and Gesture Recognition, pp. 462467.
- [53] Huang, C., L., Jeng, S., H., 2001, "A ModelBased Hand Gesture Recognition System," Machine Vision and Application, vol. 12, no. 5, pp. 243258.
- [54] Zieger C., Omologo M., "Acoustic Event Detection ITCirst AED database", ITC report, 2005
- [55] Padrell J., Macho D., Nadeu C., "Robust speech activity detection using LDA applied to FF parameters", in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.
- [56] Cui, Y., Weng, J., 1999, "A LearningBased PredictionandVerification Segmentation Scheme for Hand Sign Image Sequence," IEEE Trans. Pattern Analysis Machine Intelligence, vol. 21, no. 8, pp.798804.
- [57] Julier, S.J.; Uhlmann, J.K. (1997). "A new extension of the Kalman filter to nonlinear systems". Int. Symp. Aerospace/Defense Sensing, Simul. and Controls 3. Retrieved on 2008-05-03.
- [58] RTP http://en.wikipedia.org/wiki/Real-time_Transport_Protocol
- [59] JMF <http://java.sun.com/javase/technologies/desktop/media/jmf/>
- [60] Paul Hyde: *Java Thread Programming*, Sams
- [61] XML <http://www.w3.org/2000/xp/>
-

PRILOGE**a) Naslov študenta**

Izidor Mlakar
Ribniki 17 8290 Sevnica
Sevnica
E-mail: izidor.mlakar@amis.net

b) kratek življenjepis

Rojen: 24.05.1983
Šolanje: 1991 – 1998 Osnovna šola Savo Kladnik Sevnica
1998 – 2000 Gimnazija Lava Celje
2000 – 2002 II Gimnazija Maribor (mednarodna matura)
2002 – 2008 FERI smer telekomunikacije

c) Vsebina priložene zgoščenke

- Diplomaska naloga v PDF formatu
 - Diplomaska naloga v formatu Microsoft Word
 - Celotna koda sistema
 - PDF datasheet za ATmega8, XPort, MD05, MPX4100A, MPX4115A, DS18B20, CMPS03
 - Električne sheme (vključujoč PCB sheme) v pdf dokumentu in izvirniku: modulov Vremenska Postaja in DATA modulov
-