



ARHITEKTURA VEB-BAZIRANOG SISTEMA ZA NADZOR I UPRAVLJANJE UREĐAJIMA

ARCHITECTURE OF A WEB-BASED SYSTEM FOR DEVICE MONITOR AND CONTROL

Nikola Vukašinić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratka sadržaj – Ovaj rad se bavi problematikom kreiranja platforme veb-baziranog sistema čija je primarna uloga udaljen nadzor i kontrola uređaja. Sistemi za upravljanje i kontrolu karakteriše veliki broj članova, svaki sa zasebnim zadacima koji međusobnom komunikacijom dovode do realizovanja namenske funkcionalnosti rešenja. Analiza se prevashodno svodi na opisivanje arhitekture sistema koja uključuje opisivanje svakog pojedinačnog podsistema. Posebna pažnja poklonjena je komunikacionom aspektu podsistema sa akcentom na izbor protokola mreže, formatom i načinom prenosa poruka. Drugi aspekt analize podrazumeva opis implementacije sistema, gde je predstavljena realizacija svakog pojedinačnog člana u okviru korištene programske paradigme.

Ključne reči: *Sistemi za kontrolu i upravljanje, pametni domovi, veb sistemi kontrole*

Abstract – *The goal of this paper is to describe a system of a web-based platform for device monitoring and control. Systems of this nature contain a large number of components each with its own task who communicate with each other in other for the solution to function properly. The primary goal is to fully describe the system layout and architecture by analysing each component individually. Special attention is given to cross component communication by analysing different network protocols, specifying a message format and the functionality of sending those messages. The second part of the analysis is focused on describing the way the system as a whole is implemented through its components and the programming languages used for their individual implementation.*

Keywords: *Surveillance and control systems, smart home, control web systems*

1. UVOD

Koncept kontrole i automatizacije kućnih uređaja je relativno star pojam u odnosu na životni vek informacionog doba. Godine 1975-te javlja se X10 kao prvi standardizovani protokol čija je namena udaljenog upravljanja kućnih aparata realizovana slanjem kratkih radio-signalâ kroz postojeću električnu arhitekturu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branko Milosavljević, red. prof.

Mane ovog protokola manifestuju se u njegovoj nepouzdanosti i nemogućnosti dvosmerne komunikacije. Sama arhitektura postojećih električnih instalacija nije savršeno kompatibilna za prenos radio-signalâ. Pojavljivanjem povremenih šumova uzrokovana je nepouzdanost primljenog signalâ. Jednosmerna komunikacija onemogućila je bilo kakvu mogućnost slanja povratnih informacija od uređaja ka korisniku čime se uveliko ograničava sfera korišćenja ovakvih sistema.

Razvojem interneta pojavljuje se potreba takozvanog udaljenog (*remote access*) pristupa. Koncept po kome se uređaji u prostorijama mogu nadgledati i upravljati sa relativnom sigurnošću iz bilo koje tačke na planeti u realnom vremenu. Uporedno sa razvojem interneta, razvijaju se novi, i unapređuju postojeći komunikacioni protokoli koji omogućavaju obostranu trenutnu komunikaciju između uređaja u relativnoj blizini.

Svi prevashodno definisani koncepti se grupišu u jedan sistem za kontrolu i upravljanje uređajima. Svrha ovih sistema je omogućavanje krajnjim korisnicima da na lak, brz i intuitivan način kontrolišu, upravljaju i automatizuju svoje prostorije svrsishodno ličnim preferencama.

Ciljevi analize se prvenstveno odnose na dva glavna aspekta. Prvi, predstavlja opis arhitekture, analizu funkcionalnih operacija članova sistema i načine njihove međusobne komunikacije, kao i načine dodavanja i brisanja uređaja. Drugi se odnosi na predstavljanje lične implementacije jednog ovakvog sistema i opisivanjem svakog pojedinačnog člana u okviru programske platforme njegove implementacije. Takođe, potrebno je analizirati kretanje informacija od uređaja do krajnjeg korisnika u oba smera.

2. ARHITEKTRA SISTEMA

Zadaci sistema ogledaju se u:

- Definisaniu veb-bazirane platforme preko koje svaki korisnik može nadgledati i upravljati svoje uređaje.
- Omogućiti korisnicima kreiranje, brisanje, dodavanje i modifikaciju uređaja za prikupljanje podataka jednog kućnog sistema, kao i njihovo grupisanje u jednu jedinstvenu tačku, u daljem nastavku čvorište.
- Mogućnosti opsluživanja što većeg broja korisnika u datom momentu.
- Prihvatanju numeričkih podataka poslatih od strane mernih senzora ka čvorištu nezavisno od frekvencije slanja. Omogućiti uporedno slanje povratnih informacije ka senzorima.

- Definisane logike za prosleđivanje podataka aplikativnom serveru.
- Prihvatanje podataka od strane servera i prosleđivanje istih aplikaciji korisničkog interfejsa.
- Definisane valjanog i intuitivnog grafičkog korisničkog interfejsa koji omogućava nadgledanje i slanje komandi definisanim uređajima preko ostvarenih komunikacionih veza.

Shodno predstavljenim zadacima arhitektura je izgrađena sa ciljem da u što efikasnijoj meri omogući ispunjenje funkcionalnih zahteva.

Takođe, imajući u vidu da je analizirani sistem prvenstveno veb-baziran, skalabilnost predstavlja još jedan ključni zahtev koji treba imati na umu. Ključne karakteristike projektovanja arhitekture su:

1. *Nezavisnost komponenti sistema* – koja govori o osobini po kojoj glavne funkcionalnosti jedne komponente minimalno moraju da zavise od ostalih u sistemu.
2. *Oslobađanje resursa na serveru* – koji se odnosi na oslobađanje aplikativnog servera od potrebe poznavanja korisničke konfiguracije mernih i kontrolnih uređaja.
3. *Definisane jasne komunikacione veze kao i univerzalnih poruka za komunikaciju* – koja je produkt nezavisnosti komponenti
4. *Razdvajanje logike prikupljanja podataka i kontrole od logike predstavljanja istih* – po svojoj prirodi arhitektura razlikuje korisničku i serversku konfiguraciju.
5. *Mogućnost sistema da podrži što veći broj merno-upravljačkih uređaja* – čime se povećavaju konfiguracione mogućnosti.
6. *Jednostavno proširivanje funkcionalnosti sistema* – gde je platforma izgrađena sa ciljem da lako i jednostavno obezbedi mogućnost proširivanja funkcionalnosti sa akcentom na štednji resursa i novca.

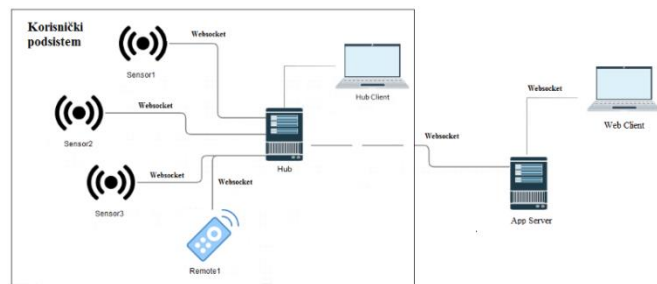
Imajući u vidu zadatke i karakteristike projektovanja arhitekture ceo sistem za upravljanje i kontrolu može se podeliti na dva strogo izolovana podsistema: *korisnički i serverski podsistem*.

Korisnički podsistem bavi se ispunjenjem konfiguraciono-operacionih funkcionalnosti vezanih za jednog korisnika. Definisane konfiguracije merno-upravljačkih senzora, prikupljanje mernih podataka i prosleđivanje istih aplikativnom serveru predstavljaju glavne funkcionalne zahteve koji podsistem treba da ispuni. Bitno je istaći da svaki korisnik sadrži svoju instancu podsistema, odnosno jedan podsistem vezan je za tačno jednog korisnika platforme.

Serverski podsistem služi kao mesto pristupa krajnjeg korisnika sistemu. Funkcionalni zadaci serverskog podsistema odnose se na prikupljanje i prenos podataka od različitih korisničkih podsistema, grafičko prikazivanje izmerenih podataka i prenos korisničkog unosa do adekvatnog podsistema.

Članovi korisničkog sistema predstavljaju:

- Upravljački i merni uređaji
- Čvorište podsistema
- Aplikacija korisničkog upravljanja



Slika 1. Celokupna arhitektura sistema

Upravljački i merni uređaji predstavljaju glavnu operativnu jedinicu celokupnog sistema. Primarna uloga ogleda se u prikupljanju podataka o trenutnom stanju prostorija različite prirode. Sam broj, fizička lokacija i položaj ovih uređaja u potpunosti su prepušteni željama krajnjih korisnika. Uređaji prikupljaju podatke čija je priroda ograničena pravilima sistema dok, u konkretnom rešenju, dozvoljeni su podaci o: temperaturi, nivou vode, dimu, količini potrošene energije, gasovima, kretanju i stanju aparata za kontrolu temperature. Potrebno je napomenuti da po šestoj stavci karakteristika arhitekture, ovu listu je moguće proširiti. Da bi se uređaj karakterisao kao deo podsistema on mora uspostaviti vezu sa čvorištem i imati mogućnost slanja podataka dozvoljene prirode preko uspostavljanje veze u numeričkom formatu. Uređaj se smatra operativno-funkcionalnim u koliko je registrovan na čvorište korisničkog podsistema. Svaki uređaj sadrži svoj identifikacioni broj čime se osigurava njegovo jedinstveno prepoznavanje od strane aplikacije čvorišta, unutar podsistema.

Čvorište predstavlja glavnu logičko-funkcionalnu jedinicu korisničkog podsistema. Definisano kao centralni deo podsistema, prvenstveno služi kao tačka uspostavljanja veza sa svim uređajima kao i aplikativnim serverom. Svaki korisnički podsistem sadrži tačno jedno čvorište. Zadatak ostvarivanja veze čvorišta i uređaja je ključan preduslov funkcionisanja korisničkog podsistema. S obzirom na prirodu samog podsistema potrebno je uspostaviti veze između potencijalno velikog broja članova koji su fizički u relativno neposrednoj blizini. Adekvatno rešenje zahteva korišćenje bežične komunikacije, dok je sam tip protokola mrežne komunikacije i dalje predmet diskusije i razlikuje se od sistema do sistema. Konačni izbor mrežnog protokola korisničkog podsistema u potpunosti zavisi od namene, okruženja i bitnih karakteristika projektovanja.

Protokoli čije prednosti i mane predstavljaju domen analize su: *Wifi*, *Zigbee*, *Z-wave* i *Bluetooth*. Prednosti *Wifi* mreže prvenstveno se ogledaju u njenoj dostupnosti i širokoj primeni. Oslanjanjem na postojeću korisničku arhitekturu oslobađamo se od potrebe uvođenja nove u cilju realizovanja komunikacije.

Mane ovog pristupa prvenstveno se ogledaju u njenoj ceni, korišćenju frekvencijskog opsega oko 2.4 GHz (opseg se mnogo koristi i mogu da se pojave smetnje u komunikaciji) kao i mogućom potrebom ulaganja u kvalitetnije mrežne uređaje s obzirom da komercijalni kućni uređaji često nemaju dovoljno snage i dometa da se nose mrežnim potrebama koje korisnički podsistem zahteva.

Zigbee i *Z-wave* predstavljaju komunikacione protokole razvijene sa ciljem da zadovolje potrebe sistema za upravljanje i kontrolu [1]. *Zigbee* odlikuje mrežna moć sa mogućnošću uvezivanja velikog broja uređaja u jednom momentu.

Takođe, uređaji koji komuniciraju ovim protokolima odlikuju se relativno malom potrošnjom energije. Prednost *Z-wave* protokola proizilazi iz upotrebe frekvencijskog opsega (800-900Mhz) što ga čini imunim na potencijalne smetnje prouzrokovane ostalim signalima i samim tim, najpouzdanijim rešenjem. Korišćenje *Bluetooth* protokola za potrebe korisničkog podsistema ograničava se na sisteme sa relativno malim brojem uređaja i malim dometom. Odlikuju ga mala potrošnja energije kao i mogućnost svakog uređaja da ima primopredajne karakteristike koje proizilaze iz same definicije mreže.

Nepouzdanost signala i mali domet predstavljaju glavne mane korišćenja ovog rešenja čime se zahteva dalje razvijanje tehnologije kako bi ona pronašla širu primenu u domenu sistema za upravljanje i kontrolu.

Kada je reč o komunikaciji između čvorišta i aplikativnog servera glavni aspekt predstavlja faktor trenutnog (*real time*) prenosa poruka. Imajući na umu da se komunikacija obavlja preko interneta, standardni *HTTP* protokol nije adekvatan za ostvarivanje ovakve komunikacije. Limitirajući faktor predstavlja zahtev-odgovor karakteristike *HTTP* protokola gde se pri svakom slanju poruke mora prvobitno poslati zahtev i čekati odgovor na isti čime se veza zatvara i svako novo slanje zahteva kreiranje novog zahteva što vremenski nije prihvatljivo.

Takođe, sam broj transakcija koji se otvaraju, zatvaraju i izvršavaju u jednoj jedinici vremena od strane velikog broja korisnika prouzrokovala bi smetnje u funkcionisanju aplikativnog servera. Rešenje je realizovano uvođenjem *Websocket* konekcije između čvorišta i aplikativnog servera.

Za razliku od *HTTP-a*, *Websocket* se zasniva na uspostavljanju stalne veze između dve partije omogućavajući slobodan i trenutni protok informacija. Nakon ostvarivanja, veza dozvoljava klijentu i serveru slanje podataka kroz otvorenu konekciju u vidu poruka bez prethodne potrebe druge strane da te informacije zatraži i bez dodatnih informacija kao što su zaglavlja akcentujući pojam trenutne komunikacije. *Websocket* protokol izgrađen je preko *HTTP-a* obzirom da se inicijalni zahtev za uspostavljanje veze (*handshake*) šalje preko *HTTP* zahteva. Ova veza takođe doprinosi većoj skalabilnosti aplikativnog servera koji otvaranjem samo jedne veze sa čvorištem oslobađa resurse čime se omogućava istovremeno opsluživanje većeg broja korisnika.

Za uspešno opisivanje logike prenosa podataka potrebno je obratiti pažnju na vremenski faktor. Merni uređaji šalju izmerene podatke čvorištu korišćenjem definisane mrežne arhitekture različitim frekvencijama.

Takođe, podaci od čvorišta do aplikativnog servera šalju se unapred definisanim intervalima slanja. Pravilno funkcionisanje čvorišta zahteva definisanje logike za prikupljanje, skladištenje, računanje proseka i slanje datih merenja. Problematika slanja podataka aplikativnom serveru može se svesti na problem „jedan senzor, jedan broj“, kojom se definiše potreba da svaka vrednost

senzora bude predstavljanja tačno jednim mernim rezultatom nasuprot mogućnosti senzora da u vremenskom periodu od prethodnog do novog slanja na čvorište pošalje jedno, više ili ni jedno merenje. Slučaj gde u periodu od prethodnog do sledećeg slanja nije pristiglo ni jedno merenje realizuje se prosleđivanjem prethodno izmerene vrednosti ponovo, sa određivanjem praga tolerancije.

Slanje nove vrednosti podrazumeva računanje centralne tendencije skupa merenja pristiglih od trenutka prethodnog slanja. Statističkim metodama zaključeno je da računanje medijane skupa podataka predstavlja najbolje rešenje računanja centralne tendencije s obzirom da je ova metoda imuna na retke nepravilnosti u merenju što je neretka pojava u funkcionisanju svih senzora merenja [2].

Klijentska aplikacija korisničkog podsistema predstavlja tačku interakcije korisnika sa konfiguracijom podsistema i ogleda se u pružanju korisničkog interfejsa i omogućavanju funkcija za dodavanje i brisanje registrovanih uređaja korisničkog podsistema, kao i uspostavljanje i raskidanje konekcije sa aplikativnim serverom.

Model serverskog podsistema svodi se na klijent-server arhitekturu. Primarna uloga aplikativnog servera ogleda se u omogućavanju funkcionalnosti prijave na sistem, uspostavljanje veza sa korisničkim podsistemima i prenosom informacije istih do adekvatnih klijenata ulogovanog korisnika.

Veb - klijentska aplikacija služi kao mesto interakcije korisnika sa funkcionalnostima celokupnog sistema. S obzirom na prirodu sistema, glavni akcenat projektovanja veb-klijenta odnosi se na pružanje intuitivnog grafičkog interfejsa za prikazivanje pristiglih numeričkih podataka i davanjem mogućnosti korisničkog unosa definisanjem adekvatnih formi.

2. IMPLEMENTACIJA SISTEMA

Opis implementacije sistema za kontrolu i upravljanje uređajima odnosi se na način realizacije funkcionalnih karakteristika svakog od člana sistema u okviru određene programske paradigme prema definisanim pravilima arhitekture. *Websocket* veza povlači sa sobom definisanje protokola za organizaciju poruka čiju potrebu u analiziranom rešenju ispunjava *STOMP* protokol [3]. *STOMP* uvodi novi sloj organizacije destinacija poruka za slanje i primanje po putanjama, dok je funkcionalnost realizovana preko različitih komandi.

2.1 Merni i upravljački uređaji

Implementacija merno-upravljačkih uređaja realizovana je korišćenjem *Arduino* platforme. Platforma se odnosi na hardversku mikro-kontroler ploču sa analognim i digitalnim ulazima i izlazima koja služi za integrisanje različitih malih električnih uređaja. Konkretno, merni uređaj predstavlja *DS18B20* temperaturni senzor dok kontrolni uređaj je predstavlja *IR* lampica koja kontroliše klima uređaj prosleđivanjem adekvatnih komandi. Veza između ploče i računara ostvarena je preko komunikacionog porta na koji se šalju rezultati merenja i komande. Aplikacija senzora realizovana je korišćenjem *SPRING* framework-a preko *JAVA* programske platforme. Glavni zadaci ove aplikacije ogleda se u čitanju komunikacionog

porta i prosleđivanju izmerenih vrednosti na čvorište kao i prosleđivanju komandi primljenih od čvorišta na komunikacioni port koji kasnije dolaze do ploče.

2.2 Čvorište korisničkog podsistema

Analizirani sistem definiše čvorište kao zasebnu server aplikaciju razvijenu u *JAVA* programskoj paradigmi korišćenjem *SPRING* framework-a. Registracija uređaja na čvorište podsistema realizuje se upisivanjem informacija o istom u internu bazu podataka čvorišta. Svaki uređaj definisan je informacijama identifikatora, naziva i tipa koji opisuje prirodu poslatih podataka iz skupa mogućih vrednosti. Manipulisanje konfiguracijom korisničkog podsistema podrazumeva kreiranje zasebnog *API*-ja koji, zajedno sa klijentom korisničkog podsistema, realizuje date funkcionalnosti.

Povezivanje uređaja na čvorište realizuje se iniciranjem sesije između dve partije. Model sesije definisan je informacijama uređaja, nizom pristiglih merenja, informaciju o poslednje poslatoj izračunatoj vrednosti tendencije, trenutnom pragu tolerancije i logičkoj promenljivoj koja govori o aktivnosti samog uređaja.

```
public class ComponentSession {  
  
    private Component component;  
    private List<Double> stack = new  
ArrayList<Double>();  
    private double lastSent;  
    private boolean sent = false;  
    private int counter = 0;
```

Kod 1. Model klase sesije uređaja i čvorišta

Standardizovana poruka za prenos mernih informacija predstavlja nužan preduslov valjanog rešenja. Bitan faktor postizanja ove standardizacije predstavlja usklađenost date poruke se mapom aktivnih sesija. Kreiranje modela poruke o stanju korisničkog podsistema ima za cilj da pruža realno stanje izmerenih podataka organizovanih na način na koji olakšava njihovu kasniju navigaciju i interpretiranje. Format poruke je definisan kao niz čija je dužina jednaka brojčanoj količini tipa senzora čija se merenja šalju za dati vremenski interval, dok svaki član niza sadrži podatke o merenjima senzora datog tipa. Pravila slanja podataka definišu pojavu da se u različitim vremenskim intervalima šalju podaci vezani za senzor različite prirode

Realizacija logike slanja poruka svodi se na definisanje internog brojača vremenskog intervala koji prema pravilu za slanje računa i šalje zahtevane vrednosti.

2.3 Aplikativni server

Aplikativni server funkcioniše po principu veb - servera čija se primitivna funkcionalnost ogleda u opsluživanju korisnika podacima potrebnim za nadzor i upravljanje njihovih podsistema, realizovan preko *SPRING* framework-a i *JAVA* programske platforme. Prvenstveno služi kao most koji prenosi informacije od čvorišta do odgovarajućeg klijenta preko otvorenih veza. Takođe, omogućava klijentima prijavu na sistem korišćenjem *JWT* šablona.

2.4 Veb-klijent

Veb-klijent razvijen je pomoću *Angular 7* framework-a koji se služi *Typescript* programskom paradigmom. Motivacija za korišćenje *Angular* framework-a proizilazi iz organizacije rešenja u komponente. Svaka komponenta predstavlja nezavisni deo koda sa svojom logikom funkcionisanja i zasebnim grafičkim prikazom. U analiziranom rešenju definisanje grafičkog prikaza za određeni tip merenja realizovan je kreiranjem zasebne komponente koja implementira dati prikaz.

Prednosti komponenti proizilaze iz njihove ponovne iskoristivosti, gde je pri svakom pozivu komponente data mogućnost prosleđivanja zasebnog ulaza, slično funkcijama. Predstavljenom organizacijom kao i standardizacijom poruke o stanju korisničkog podsistema osigurana je generičnost rezultata i omogućena je laka izmena logike grafičkog prikaza imajući na umu da je ona definisana samo na jednom mestu. Grafički prikaz komponenti omogućen je korišćenjem *chart.js* biblioteke.

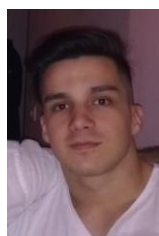
3. ZAKLJUČAK

Prevhodni cilj analize i implementacije bio je opisivanje arhitekture jedne platforme za efikasno ispunjenje funkcionalnih zahteva predstavljenog sistema. Glavni zadatak, osim same funkcionalnosti, bio je postavljanje sistema sa čvrstom osnovom čije bi dalje unapređenje i razvijanje bilo lako i intuitivno. Faktor daljeg napretka i razvitka može se odnositi na veliki dijapazon aspekata sistema kako sa funkcionalno-implementacione tako i sa hardversko-tehničke strane. Priroda sistema omogućava individualni razvitak i proširivanje funkcionalnosti pojedinačnih članova bez kompromitovanja samog toka informacija opisanom arhitekturom.

4. LITERATURA

- [1] Inovelli, <https://inovelli.com/z-wave-vs-zigbee-vs-bluetooth-vs-wifi-smart-home-technology/>
- [2] Laerd statistics, Measures of Central Tendency <https://statistics.laerd.com/statistical-guides/measures-central-tendency-mean-mode-median.php>
- [3] Stomp over Websocket, <http://jmesnil.net/stomp-websocket/doc/>
- [4] Components, <https://angular.io/api/core/Component>

Kratka biografija:



Nikola Vukašinić rođen je u Novom Sadu 1994. god. Završio je Gimnaziju "Svetozar Marković" u Novom Sadu. Osnovne četvorogodišnje studije na Fakultetu tehničkih nauka završio je 2018 godine i iste upisuje master studije na modulu elektronsko poslovanje. Ispunio je sve obaveze i položio sve ispite predviđene studentskim programom.