

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Ako Tõnissoo

R-i põhised targa kodu analüüsilahendused

Bakalaureusetöö (9 EAP)

Juhendaja: Jakob Mass

Tartu 2018

R-i põhised targa kodu analüüsilahendused

Lühikokkuvõte:

Käesolevas bakalaureusetöös loodi baaslahendus asjade interneti andmete visualiseerimiseks ning analüüsimiseks. Lahendus pärib ühest või mitmest andmebaasist andmed, viib need sobivale kujule ning visualiseerib need. Lahenduse loomiseks on kasutatud OpenHAB-i nimelist targa kodu haldamise keskkonda ning programmeerimiskeelt R ja tema teeki Shiny. Töö käigus arendas autor toimiva prototüübi. Lahendus koosneb kolmest näidisrakendusest, millest esimene on juhtpaneel, kust on kasutajal võimalik näha enda seadmete informatsiooni. Juhtpaneel koosneb neljast näidikust ning kahest graafikust. Teine lahenduse osa on näidiskrakendus, kus ARIMA mudeli põhjal prognoositakse tulevasi andmeid. Kolmandaks rakenduseks on kahe seadme väärtuste seoste leidmine Pearsoni korrelatsioonikordaja abil. Loodud lahendust testiti Tartu Ülikooli mobiili- ja pilvearvutuste labori targa kodu seadmete andmestikuga.

Võtmesõnad:

R, Shiny, OpenHAB, Asjade internet, Andmeanalüüs

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

R Based Smart Home Analysis Solutions

Abstract:

In the current bachelor thesis a sample solution for visualizing and analyzing data of internet of things was created. The solution gets the data from one or more OpenHAB databases, transforms and visualizes it. OpenHAB, an automation software for smart homes, and R with Shiny package are main tools used to create this solution. The solution consists of three main parts, firstly, a dashboard for visualizing the data of the internet of things solution in real time. The dashboard consists of four gauges and two graphs. Secondly, forecast based on ARIMA model. Third part of the solution is finding the correlation between values of two devices with Pearson's correlation coefficient. The solution was tested on the data of the smart home of Mobile and Cloud Computing Lab of the University of Tartu.

Keywords: R, Shiny, OpenHAB, Internet of Things, Data analysis

CERCS: P170 Computer science, numerical analysis, systems, control

Sisukord

1.	Sissejuhatus	5
2.	Ülevaade asjade internetist ja andmete analüüsist	7
2.1	Asjade internet.....	7
	Ajalugu	7
	Kasutusvaldkonnad	7
2.2	Andmete analüüsimine	8
	Kogumine	8
	Töötlemine	9
	Analüüsimine	9
2.3	Andmete visualiseerimine	9
3.	Ülevaade sarnastest lahendustest	12
4.	Kasutatud tehnoloogiate ülevaade.....	14
4.1	OpenHAB	14
	Arhitektuur ja Struktuur	14
	Komponendid.....	15
	Lisad.....	16
	Andmebaasid.....	16
4.2	R ja Shiny	17
	Programmeerimiskeel R.....	17
	Shiny	17
4.3	SQLite.....	17
4.4	Andmetöötlus	18
	Prognoosimine.....	18
	Korrelatsioon.....	18
5.	Loodud Lahendus.....	20
5.1	Eesmärk ja arhitektuur.....	20

5.2	OpenHAB-i seadistamine	21
5.3	Süsteemi komponendid	22
	Andmebaasid	23
	Andmetöötlus	23
	Kasutajaliides	23
6.	Kokkuvõte	26
7.	Viidatud kirjandus	27
Lisad	30
I.	Litsents	30

1. Sissejuhatus

Asjade internet ehk vārkvōrk on omavahel suhtlevate ning andmeid jagavate seadmete vōrgustik, mis on viimastel aastatel muutunud järjest populaarsemaks. Vārkvōrgu kasutusvaldkonnad järjest laienevad ning see on juba jõudnud ka tavakasutajateni [1]. Selleks, et asjade interneti potentsiaali täielikult ära kasutada, on aga vaja ülevaadet seadmete töö käigus tekivatest andmetest. Näiteks on kasutaja kodus kliimaseade, mis hoiab kodu temperatuuri alati muutumatuna. Andmeid analüüsidest oleks kasutajal võimalik seadme tööd hōlbustada, ennustades tulevasi andmeid. Selle abil saaks seade hakata varem valmistuma tulevasteks temperatuurimuutusteks, vähendades seadme töötamise intensiivsust. Andmeid analüüsimata piirdub vārkvōrgu kasulikkus lihtsamate automatiseerimislahendustega, kuid paljud potentsiaalsed kasulikud rakendused eeldavad mingil määral andmete analüüsimist.

Kuna elame ajastul, kus andmetel on tähtis osakaal, siis on visualiseerimine praegu aktuaalsem kui kunagi varem. See võimaldab kasutajal lihtsasti mõista infot, mille analüüsimine muidu võtaks palju aega ning teadmisi. [2]

Autor kasutab enda töös vārkvōrgu haldamiseks OpenHAB-i nimelist keskkonda. OpenHAB on avatud lähtekoodiga tarkvara, mis võimaldab kasutajal asjade interneti lahendust üles seada ning muu hulgas seadmete poolt tekkivad andmed andmebaasi salvestada. Autor valis tööks just selle platvormi, kuna seda on kasutajal lihtne üles seada. Lisaks on OpenHAB-il suur valik laiendusi turul olevate erinevate tootjate seadmetele, nagu näiteks Philips HUE valgustid [3].

Kuigi andmete visualiseerimist võimaldab ka OpenHAB, kasutab andmete töötluks ja visualiseerimiseks autor programmeerimiskeelt R, mis on üks populaarsemaid andmetöötluks vahendeid [4]. R-ile leidub tänu sellele hulganisti õpetusi ja lisateeke, mis jällegi hōlbustavad kasutaja tööd. Lisaks sellele on kogenumatel kasutajatel R-i kasutades võimalik tarkvara enda nägemise järgi modifitseerida lihtsam, kui mõnda teist tööriista kasutades.

Töö eesmärk on luua asjade interneti andmete analüüsimist hōlbustav baaslahendus näidiskonstruktsiooniga. Lahenduse abil oleks võimalik lihtsasti arendada OpenHAB-i ja R-i põhiseid lahendusi, mis võimaldavad erinevate andmebaaside andmete visualiseerimist analüüsimist. Selle eesmärgi saavutamiseks luuakse töö käigus kolmest osast koosnev näidiskonstruktsioon. Näidiskonstruktsioon peaks hōlmama endas andmeid visualiseerivat juhtpaneeli ning nende keerkamaks analüüsimiseks tuleviku prognoosi ja vārtuste korrelatsiooni.

Töö koosneb neljast peatükist, millest esimeses tutvustatakse lugejale asjade interneti ning andmete analüüsimist ja visualiseerimist. Teises peatükis tutvutakse varem tehtud sarnaste töödega. Kolmas peatükk annab ülevaate kasutatud tehnoloogiatest. Viimases peatükis räägib autor töö käigus valminud lahendusest.

2. Ülevaade asjade internetist ja andmete analüüsist

2.1 Asjade internet

Asjade internet on omavahel seotud seadmete võrgustik, kus seadmed üksteisega infot vahetavad ning neile määratud ülesandeid täidavad. See võimaldab seadmetel töötada efektiivselt, ilma pidevat inimese juhtimist vajamata [1].

Ajalugu

Teadaolevalt esimene näide asjade internetist pärineb aastast 1982. Nimelt ühendasid töötajad Carnegie Mellon Ülikoolis Coca-Cola masina serveriga ning lisasid sinna väiksed lülid. Siis oli võimalik internetist kontrollida, kas masinast üldse praegu juua saab ning kas joogid olid külmad [5].

Asjade internet sai oma nime aastal 1999 Kevin Ashtoni, tolaegse MIT Auto-ID labori tegevdirektori poolt, kes mainis seda terminit enda kõnes Procter&Gamble-le. Ta uskus, et RFID (*radio frequency identification*) on asjade internet eelduseks. Kui kõik seadmed on märgistatud, suudaksid arvutid neid hallata [6].

Tänapäeval on asjade internet arenenud sinnani, et praktiliselt kõiki elektroonilisi seadmeid on võimalik omavahel ühendada. Ennustatakse, et asjade internetist tulenev majanduslik kasu ulatub aastaks 2025 kuni 6.2 miljardi dollarini [7].

Kasutusvaldkonnad

Asjade internet on tänapäeval kasutuses juba väga paljudes erinevates valdkondades. Järgnevalt antakse ülevaade enim levinud kasutusvaldkondadele.

Kodu

Värkvõrku kuuluvad seadmed aitavad lihtsustada kodude automatiseerimise probleemi. Paljude seadmetega värkvõrgu lahendustes on tavaliselt kasutusel juhtpaneel, kust saab kõiki seadmeid hallata. Tihti võib see asuda kas seinale monteerituna või siis kasutaja nutitelefonis. Selliste lahendustega kodusid on küll algselt kallis üles seada, kuid edaspidi muudab see kasutajate elu lihtsamaks ning ka säästlikumaks, hoides kontrolli all elektrikulu [8].

Ettevõtlus

Kui kodudes piirdub asjade interneti kasu pigem mugavuse ning säästlikkusega, siis ettevõtluses on sellel märksa rohkem kasutusvaldkondi. Nendest üks tähtsamaid on kindlasti

tootmine. Pannes masinad omavahel suhtlema, on võimalik muuta tootmist efektiivsemaks ning selle abil hoida kokku kulusid. Lisaks sellele on värgvõrk ettevõtluses kasutuses näiteks põllumajanduses, kus sensorite abil on võimalik saada reaajas infot põldude kohta. See muudab põllumehed teadlikumaks, kuna info on neile kaugelt kätte saadav ning ei pea alati saaduste seisukorra hindamiseks kohapeal käima [8].

Muud kasutusvaldkonnad

Mida küpsemaks asjade interneti tehnoloogia saab, seda enam julgetakse seda kasutusele võtta ka valdkondades, kus eksimisruumi sisuliselt pole. Näiteks tervishoius on järjest populaarsemaks muutumas erinevad jälgimisseadmed. Alustades tavaliste aktiivsusemonitoridega ning lõpetades patsiendi kehasse siirdatud seadmete monitoridega. Ka haiglates katsetatakse järjest enam asjade interneti tehnoloogiat, et vähendada suure pingel all töötavate arstide koormust [8].

Järjest enam proovitakse integreerida asjade interneti ka linnapilti. On hakatud disainima lausa terveid linnu, milles leidub palju omavahel suhtlevaid seadmeid. Sellega üritatakse muuta meie elukeskkondi puhtamaks, efektiivsemaks ning turvalisemaks [8]. Üheks selliseks projektiks näiteks on SmartEnCity, milles osaleb ka Tartu linn. [9]

2.2 Andmete analüüsimine

Andmete analüüsimine on protsess, mis koosneb suuremas perspektiivis kolmest faasist: andmete kogumine, töötlemine ning analüüsimine. Selle protsessi eesmärk peaks olema toorandmetest väärtusliku informatsiooni leidmine, mille abil saaks teha järeldusi ning otsuseid näiteks teaduses või ärimaailmas [10].

Kogumine

Andmete kogumine on protsess, mis võib suuresti hõlbustada edasist analüüsi protsessi. Enne andmete kogumise hakkamist on vaja paika panna nõuded. Nende järgi otsustatakse, milliseid andmeid on vajalik koguda ning mis on vähem oluline. Heade nõuete seadmine tähendab seda, et analüüsitavates andmetes on potentsiaali millegi kasuliku leidmiseks. Kehvasti sätestatud nõuete kohal võib andmetest olla võimatu midagi välja lugeda. Pärast nõuete kinnitamist algab andmete kogumise protsess [10]. Andmete kogumiseks on väga palju erinevaid mooduseid, alustades intervjuudega ning lõpetades sensorite andmete automaatse salvestamisega andmebaasi. Kindlasti on mõned neist variantidest usaldusväärsemad kui teised. Näiteks on suurem tõenäosus, et intervjuu meetodil kogutud andmetest läheb

sisestamisel midagi kaotsi, kui et sensorite poolt andmebaasi kogutud andmed kaduma lähevad. Samas võib intervjuudest saadud informatsioon olla palju kasulikum, seega ei saa eelistada üht meetodit teisele [11].

Töötlemine

Carl French on oma 2006. aasta raamatus “Data Processing and Information Technology” defineerinud andmetöötluse järgmiselt: Andmetöötlus on andmete kogumine ja manipuleerimine tähendusrikka informatsiooni saamiseks. Ta mainib ka, et juba aastaid on arvutid olnud andmetöötluse põhiliseks tööriistaks ning nimetab seda elektrooniliseks andmetöötluseks [12]. Üheks oluliseks andmetöötluse osaks on andmete puhastamine. Kui eelnevalt on andmed organiseeritud ning struktureeritud, siis on vaja kontrollida andmetest puudujääke, duplikaate või vigu. Tihti võib nende probleemide teket ennustada uurides andmete päritolu ning saamisviisi. Üks moodus vigade leidmiseks on võrrelda erinevatelt allikatelt saadud andmeid sama sündmuse kohta [11].

Analüüsimine

Pärast andmete puhastamist on võimalik hakata andmeid analüüsima. Uurimisandmete analüüs (ing. *Exploratory data analysis*) seisneb puhastatud andmetest kasuliku informatsiooni leidmises. Selleks on mitmeid võimalusi, näiteks andmete visualiseerimine või kirjeldavate statistiliste näitajate, nagu keskmine ja mediaan leidmine. Ei ole ebatavaline, et protsessi käigus tuleb andmeid uuesti puhastada. Analüüsi üheks osaks on ka algoritmide rakendamine andmetele, et tuvastada seoseid muutujate vahel [13].

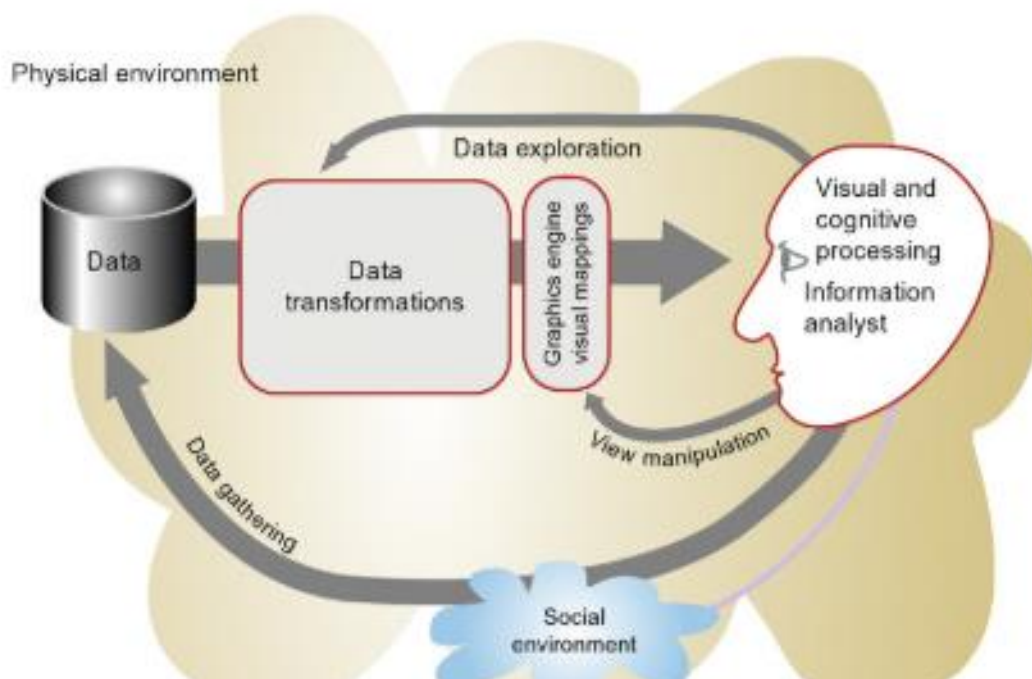
2.3 Andmete visualiseerimine

Visualiseerimine on meetod, mille abil saab teavet esitada diagrammidena, graafikutena või muul visuaalsel moel. Läbi aegade on kasutuses olnud mitmeid visualiseerimise tüüpe, näiteks võimaldasid kunstiteosed sündmustest kiiremini aru saada, kui kirja lugedes. Tänapäeval on andmete visualiseerimist mingil määral kasutatud pea igas eluvaldkonnas, kuid mõned tähelepanuväärsemad rakendusvaldkonnad neist on teadus, meditsiin ja haridus. Tüüpiliselt toimub visualiseerimine arvutigraafika abil, just arvutite tekkel on olnud suur panus visualiseerimise kiirele arengule [14].

Üks levinumaid visualiseerimise valdkondi on massilise andmete hulga suurenemise tõttu järjest aktuaalsemaks muutuv teaduslik visualiseerimine ehk visuaalne andmeanalüüs. See

kujutab endas graafilise kuvamise abiga andmetest paremini aru saamist. Seda meetodit kasutatakse tihti valdkondades, kus leidub massiivses koguses informatsiooni, millest muul viisil on võimatu aru saada. Nendeks valdkondadeks on näiteks füüsika, astronoomia ning meteoroloogia [2].

Nagu varem mainitud, elame ajastul, kus on antakse pidevalt kasutajale edasi infot, mille muul kujul kinnistamine võtaks kaua aega. Näiteks õpetatakse koolis juba küllalt varakult aatomi struktuuri, mida reaalselt oleks küllaltki keeruline mõista, kuid graafilise esituse korral on see hoomatav suurele osale inimestele. Nagu eespool mainitud, on tihti analüüsitavate andmete hulk nii suur, et nende töötlemiseks ning kuvamiseks kulub palju aega ning ressurssi. See on ka põhjus, miks enamus visualiseerimist tänapäeval arvutigraafika abil tehakse, kuna tihti on vaja visualiseerimise algoritme korduvalt muuta, et saada kõige optimaalsem tulemus [2]. Teiseks põhjuseks, miks andmeid visuaalsel kujul hoida on, et suurt hulka andmeid on võimalik salvestada vähem ruumi võtvasse graafikusse. Lisaks sellele on tihti graafikutelt parem näha, mida uuritava sündmuse juures muuta.



Joonis 1: Andmete visualiseerimise protsess [15]

Autor Colin Ware kirjutas enda 2012. aasta raamatus “Information Visualization: Perception for Design”, et andmete visualiseerimise protsess jaguneb nelja faasi, mis on näidatud ka joonisel 2. Esimeseks faasiks olgu andmete kogumine ning salvestamine. Teises faasis töödeldakse andmed kujule, mis on algandmetest lihtsamini töödeldav. See faas toob tavaliselt kaasa andmete mahu vähenemise, kuna ebasobival kujul andmed heidetakse välja.

Kolmandaks faasiks on Ware arvates andmete visuaalne kuvamine, millele lisandub kasutaja võimalus valida, millist alamhulka kuvada. Viimane faas on andmete kognitiivne töötlemine ning analüüs kasutaja poolt [15].

Visualiseerimise olulisemaid kriteeriume on selle arusaadavus. Mugavaim viis suuri andmehulkasid esitada on tabelite ning graafikute abil. Hästi loodud graafikult on keeruliste protsesside kohta järeldusi teha võimalik ka sellest valdkonnast kaugel inimesel. On palju mooduseid andmehulkade visualiseerimiseks. Õige meetodi valimisel on suur kaal lõpptulemuse kvaliteedis. Halvasti valitud algoritmi või graafiku tüübi korral võivad andmed jääda arusaamatuks või veel hullem, anda sootuks vale arusaama. Lisaks sellele on väga oluline graafiku skaala, tavaliselt valitakse telgede vahemikud vastavalt andmete väärtustele. Samuti omab suurt rolli graafiku kujundus, millega tuleks kindlustada, et inimene keskenduks juba intuiitiivselt graafiku kõige olulisematele osadele. See eeldab, et mida tähtsam on informatsioon, seda pilkupüüdvam on ta ka graafikul.

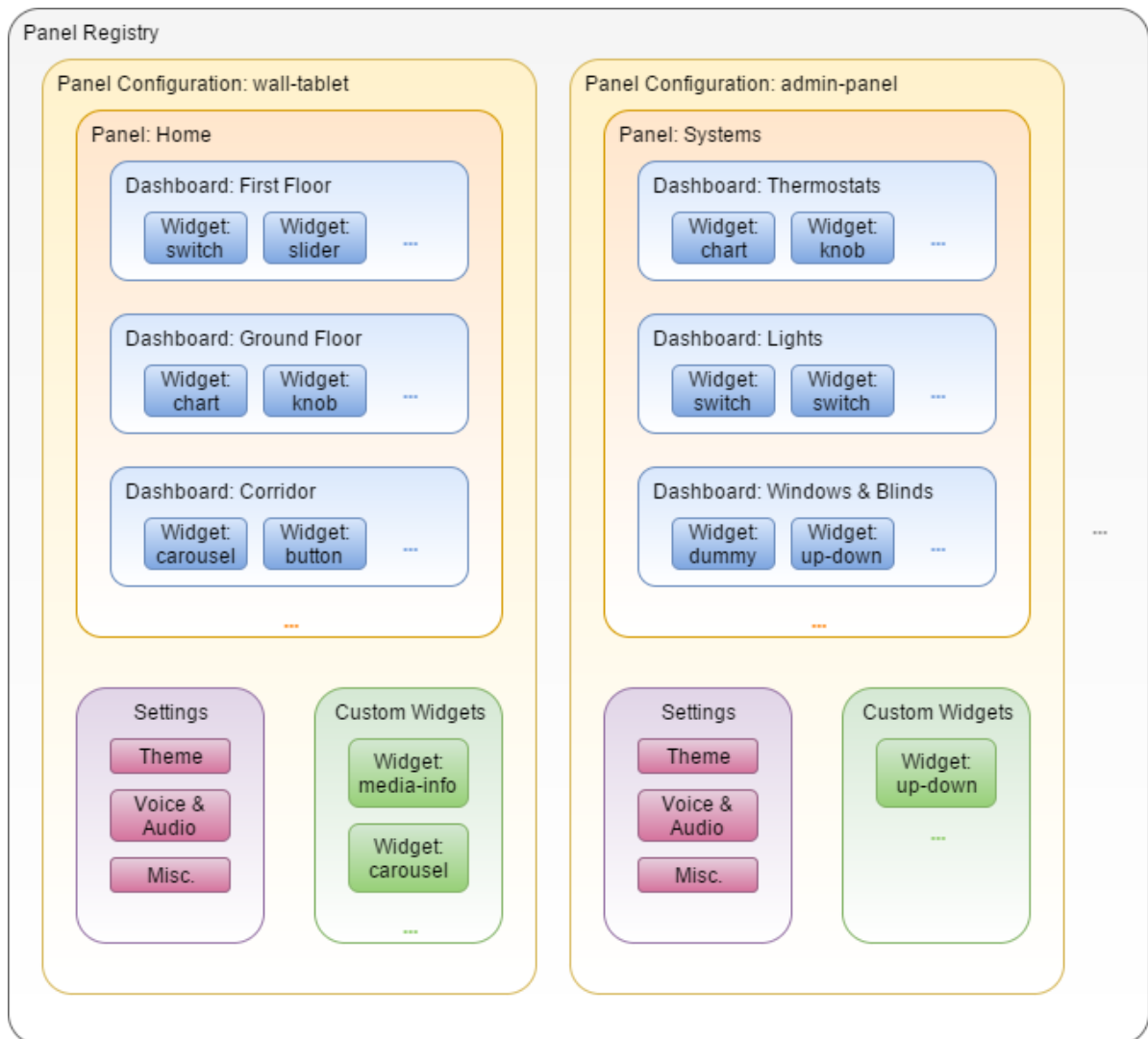
3. Ülevaade sarnastest lahendustest

Internetis on saadaval mitmeid lahendusi asjade interneti andmete visualiseerimiseks. Enamasti on need tehtud eelnevalt mainitud juhtpaneelile sarnasel põhimõttel. Järgnevalt on kirjeldatud olemasolevaid töid, mis kasutavad OpenHAB-i tehnoloogiat.

Smirek et al. on töös “Towards Universally Usable Smart Homes – How Can MyUI, URC and OpenHAB Contribute to an Adaptive User Interface Platform” uurinud MyUI, Universal Remote Control (URC) ja OpenHAB-i abil kohanemisvõimelise kasutajaliidese loomist. Vaatluse all on eelnimetatud tehnoloogiate võimalus ühenduda erinevate seadmetega, pakkuva kasutajale võimalikult sobivat kasutajaliidest ning kasutajaliidese adaptiivsust. Selgus, et MyUI suudab pakkuva kasutajale kohanemisvõimelist kasutajaliidest, URC toimib kui platvorm kasutajaliidestele ning on seega eelduseks MyUI kasutamisele. OpenHAB pakub suurepärasest võimalust erinevate tootjate poolt loodud seadmete sidumiseks ning toimib ka hea vahevara lahendusena. Käesolevas lõputöös loob autor ühe osana kasutajaliidese, sarnaselt eelnevalt mainitud lahendusele. Autor kasutab antud töös ainult openHAB-i raamistikku, mis muudab süsteemi arusaadavamaks. Ühe miinusena kaob tööst URC-st loobumise tagajärjel kaugelt kontrollimise võimalus [16].

OpenHAB pakub kasutajatele süsteemi haldamiseks mitmeid lahendusi, millest antud tööga sarnaneb HABPaneli lahendus. Tegemist on keskkonnaga, milles on lihtne luua juhtpaneel, mis sobivad eriti hästi (seinale monteeritud) tahvelarvutitele. Joonisel 1 on näha HABPaneli arhitektuur. HABPanel koosneb paneeli registrist (ing k. *Panel Registry*), mis haldab erinevate paneelide seadistusi. Iga paneeli seadistus on ümbris, kuhu kuuluvad paneel ja tema seaded ning kohandatud vidinad. Iga HABPaneliga ühendatud seade omab omaette seadistust. Paneel on HABPaneli kontekstis hulk juhtpaneel, mis kuvavad lõppkasutajale vidinate läbi informatsiooni. Üks populaarne HABPaneli lahendus on luua kodu jaoks kaks paneeli: tavakasutaja ning administraator. Mõlemad koosnevad juhtpaneelidest, milleks on tavakasutaja jaoks toad majas ning administraatori jaoks eri liiki seadmed. HABPanel on üleüldiselt väga hea süsteem asjade interneti haldamiseks, seal on suur valik vidinaid ning arusaadav kasutajaliidest. Siiski jätavad HABPaneli võimalused natuke soovida, näiteks pole võimalik teha ennustusi andmete tulevaste väärtuste kohta ning ka kohandamisvõimalused võiksid veidi paremad olla. Seal on küll võimalik luua kohandatud vidinaid, kuid antud töö

peaks pakkuma kasutajale võimalust kohandada terve juhtpaneel enda soovide järgi, eeldusks teadmised R-i keelest [17].



Joonis 2: HABPaneli ülesehitus [17]

4. Kasutatud tehnoloogiate ülevaade

4.1 OpenHAB

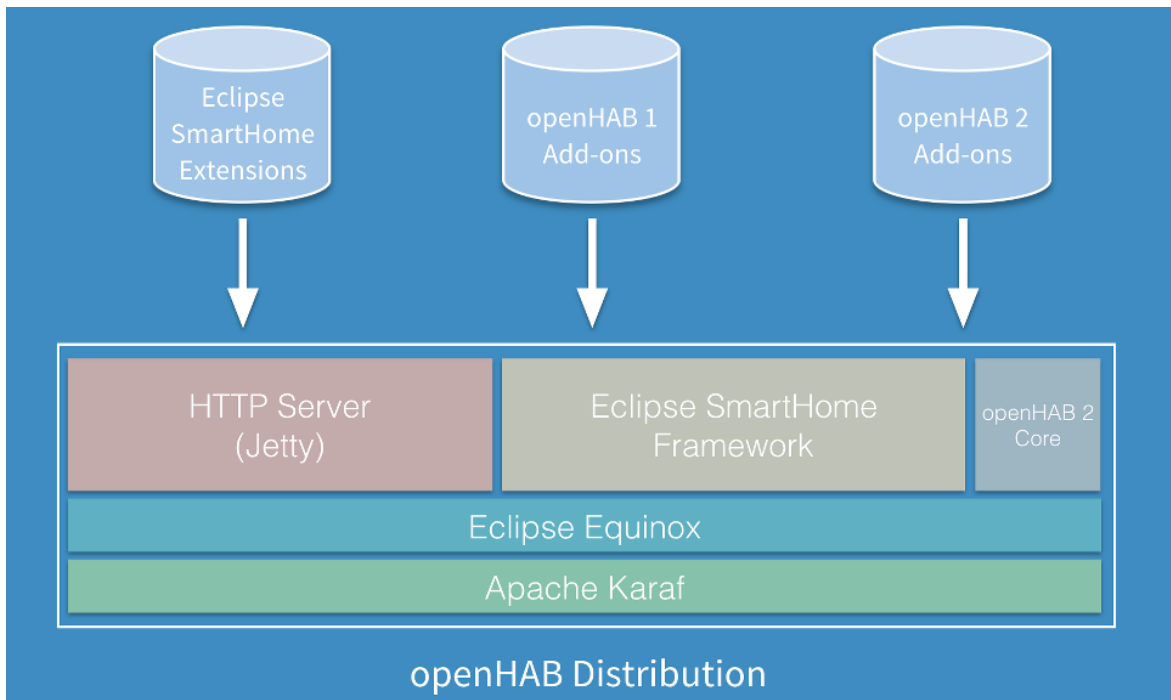
OpenHAB on avatud lähtekoodiga automatiseerimisplatvorm, mis on loodud erinevate seadmete integreerimiseks ühte lahendusse. OpenHAB on kasutatav seadmetes, mis on võimalised jooksutama Java virtuaalmasinat ning on loodud tootjatest sõltumatuna [3]. Seega on võimalik integreerida väga suur hulk erinevaid seadmeid erinevatelt tootjatelt [18].

Arhitektuur ja Struktuur

OpenHAB on alamsüsteemidest koosnev ning üheks OpenHAB-i kasutamise eelduseks on tema alamsüsteemide sõltumatu üles seadmine ning konfigureerimine. Joonisel 3 on tutvustatud OpenHAB-i arhitektuuri. Selle mõistmiseks tuleb arhitektuurile mõelda kui kahest vaatest koosnevale tervikule.

Esimesene, füüsiline vaade hõlmab endas süsteemi lisatud seadmeid, nende vahelisi ühendusi ning teisi süsteemi füüsilisi aspekte. Teine, funktsionaalne vaade keskendub sellele, kuidas informatsioon seadmete ning ühenduste vahel kasutajaliideses esitatud on, kuidas kasutaja poolt seatud reeglid mõjutavad seadmete kujutamist tarkvaras. Lisaks sellele on funktsionaalse osa ülesanne ka kasutajaliideses läbi viidud tegevustele vastavad muudatused sisse viia ka valitud seadme tarkvaras [3].

OpenHAB on arendatud Javat kasutades ning suures osas Eclipse Smarthome [19] raamistikul põhinev. Tegemist on modulaarse tarkvaraga, mis on lisade kaudu laiendatav. Lisade abil on OpenHAB-il lai valik funktsionaalsusi, alustades kasutajaliidestega ning lõpetades suure hulga ühendatavate seadmetega [20].



Joonis 3: OpenHAB-i ülesehitus [3]

Komponendid

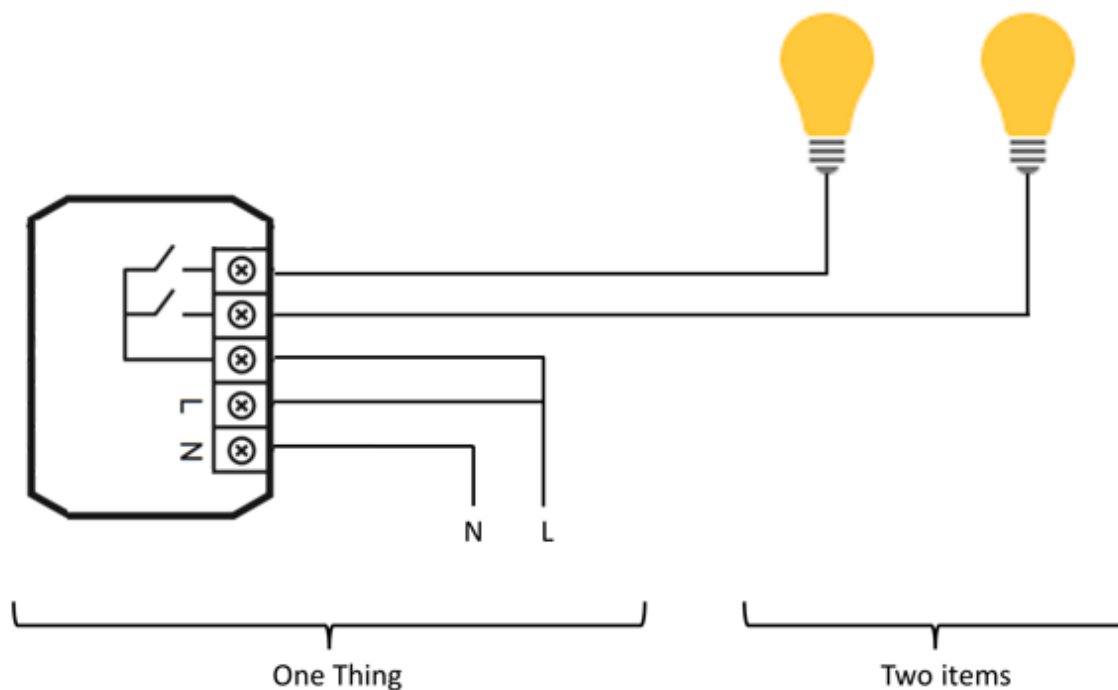
Asjad (ing. *things*) on OpenHAB-i kontekstis üksused, mida saab süsteemile lisada. Need võivad olla füüsilised seadmed, kuid võivad olla ka mingi muu hallatava informatsiooni või funktsionaalsuse allikad. Asjadel võib olla rohkem kui üks funktsioon.

Asjad täidavad oma funktsionaalsust läbi **kanalite** (ing. *channels*). Asjadel on iga rolli jaoks kanal ning ainult juhul, kui asi on seadistatud vastavat rolli kasutama, on selle rolli kohta käiv kanal aktiivne. Kasutaja ei ole kohustatud kasutama asja kõiki funktsionaalsusi.

Seosed (ing. *bindings*) on lisad, mis ühendavad **kirjed** (ing. *items*) füüsiliste seadmetega. Joonisel 4 on näidatud asja kahe kirjega, mille vahel on seosed.

Kirjed on rakenduse võimalused, mida on võimalik kasutada kasutajaliideses või automatiiseerimises. Igal kirjel on määratud seisund ning neile on võimalik käsklusi anda.

Lülid (ing. *links*) on seotud täpselt ühe kanali ning ühe kirjega. Kui kanal on kirjega seotud, siis on võimalik kasutada vastava kirje võimalusi läbi kanali asjaga suhtlemiseks. Igal kirjel võib olla mitu kanalit ning igal kanalil mitu kirjet [21].



Joonis 4: OpenHAB-i ühendamine seadmetega [21]

Lisad

OpenHAB-i süsteemi muudavad täielikuks mitmed erinevat liiki lisad. On olemas palju erinevad kasutajaliideseid, näiteks nutitelefonide või siiski arvuti jaoks. Lisaks sellele on kasutajatele saadaval veel ka palju spetsiifilisi teenuseid nagu hääljuhtimine ning suur hulk erinevaid reegleid ning skripte, mida on seadmete kontrollimiseks võimalik kasutada. OpenHAB-i on ka integreeritud võimalus andmeid salvestada. Selleks on võimalik kasutada erinevaid püsivusteenuseid [22].

Andmebaasid

Andmebaasidesse salvestamine toimub OpenHAB-is püsivusteenuste (ing. *persistence*) abil. Püsivusteenused salvestavad ühendatud seadmete andmed kasutaja soovitud andmebaasi kindlate reeglite järgi, millest on täpsemalt räägitud OpenHAB-i seadistamise peatükis. Antud töös on kasutatud SQLite andmebaasi, kus iga seadme jaoks luuakse andmebaasi tabel, kus on kaks veergu: aeg ja väärtus. Näiteks esimese seadme tabeli nimi on *item0001*, teise *item0002* jne. Lisaks sellele luuakse tabel nimega *items*, kuhu salvestatakse kõigi tabelite seadmete nimed ning tabel nimega *sqlite_sequence*, kuhu salvestatakse andmebaasis olevate seadmete arv. Tabelitele vaikimisi määratud nimesid on võimalik ka muuta, kuid autori arvates on antud struktuur piisavalt loogiline. Lisaks on pakutud populaarsematest andmebaasidest ka näiteks MySQL ja PostgreSQL andmebaasid [23].

4.2 R ja Shiny

Programmeerimiskeel R

R on vabavaraline süsteem statistiliste arvutuste ning graafikute tarbeks. See koosneb programmeerimiskeelest ning kasutaja keskkonnast. R on arendatud suuresti keeli S ja Scheme aluseks võttes. Süntaksilt on R väga sarnane S-iga, kuid põhjaks olev semantika tuleneb Scheme keelest.

R-i tuumaks on interpreteeritud programmeerimiskeel. Suur osa R-i funktsioonidest on ka samas keeles kirjutatud, kuid vajadusel on efektiivsuse nimel võimalik kasutada keeli C ja C++. R-is leidub funktsioone suure hulga statistiliste protseduuride jaoks. Lisaks sellele on saadaval suur hulk funktsioone, mis varustavad R-i graafilise keskkonnaga andmete visuaalseks kuvamiseks. R-is on saadaval ka täiendavad teegid kindlate ülesannete, näiteks on neid palju andmetöötluse ja analüüsi jaoks, mis on ka üheks põhjuseks, miks antud töös R-i kasutati [24].

Shiny

Shiny on R-i pakett, mis on mõeldud veebirakenduste loomiseks. See võimaldab R-i statistiliste arvutuste võimalusi hõlpsasti esile tuua, ilma et oleks vaja kasutusele võtta mingit uut programmeerimiskeelt. Shiny-l on sisse ehitatud mitmeid vidinaid kiireks interaktiivsete kasutajaliideste loomiseks. Nagu mainitud, on Shiny-t võimalik kasutada ilma klassikaliste veebiarenduse keelteta nagu HTML, CSS ja JavaScript, kuid vajadusel on neid väga lihtsasti võimalik rakendusse integreerida, mis muudab Shiny keskkonna hästi kohandatavaks [25].

4.3 SQLite

Kuigi andmebaase on tänapäeval valikus palju erinevaid, näiteks MySQL ja MariaDB, otsustas autor antud töös kasutada SQLite andmebaasi. SQLite on avatud lähtekoodiga serverivaba andmebaas, mille autor valis kasutamiseks tema vähese ressursinõudlikkuse tõttu. Sellest hoolimata omab ta kõiki SQL keele funktsioone, ei vaja konfigureerimist ning hoiab terve andmebaasi ühes failis [26].

4.4 Andmetöötlus

Prognoosimine

Prognoosimine (ing. *forecasting*) on protsess, mis seisneb tulevaste sündmuste ennustamises eelnevate sündmuste andmete põhjal. Selleks on saadaval mitmeid mudeleid, näiteks deflatsiooni või libiseva keskmise mudel. On väga oluline prognoosimiseks valida sobiv mudel, kuna mõned mudelid sobivad kindlateks ülesanneteks paremini kui teised [27].

Tulevaste väärtuste prognoosimiseks kasutatakse antud töös ARIMA ehk Autoregressiivse integreeritud libiseva keskmise (ing k. *Auto-Regressive Integrated Moving Average*) mudelit. ARIMA mudel on peamiselt kasutuses, nagu ka antud töös, tulevaste väärtuste ennustamiseks. Eriti hästi sobib ARIMA mudel perioodilistele andmetele, näiteks temperatuurile. See on ka põhjus, miks ta antud töös kasutusel on [28].

ARIMA jaguneb kolmeks komponendiks: autoregressiivseks, integreeritud ning libiseva keskmise komponendiks. Esimesena toimub töö integreeritud komponendis, kus algne aegrida (ing. *time series*) lahutatakse temast maas olevast aegreast (näiteks novembri väärtuse lahutatakse oktoobri väärtustest). Sellega kaotatakse aegreast tendents. Teine osa mudelist on autoregressiivne osa, mis leiab eelmiste perioodide andmete mõju nendele järgnevate perioodide andmetele. See toimub eelmises faasis statsionaarseks muudetud andmetele vastava regressioonimudeli loomise abil. Viimases, libiseva keskmise faasis leitakse eelmiste perioodide veamäärade mõju tulevaste perioodide veamääradele [29].

Korrelatsioon

Korrelatsioon on kahe-suunaline analüüs, mis mõõdab seose tugevust ning suunda kahe muutuja vahel. Korrelatsiooni uurimine on oluline paljudes valdkondades, väga suur tähtsus on sellel näiteks majanduses, kus muuhulgas saab korrelatsiooniga jälgida mingite sündmuste mõju hindadele. Korrelatsiooni tugevust saab mõõta korrelatsioonikordaja abil. Selle väärtused asuvad vahemikus -1 kuni 1 ning mida suurem on kordaja absoluutväärtus, seda tugevam on muutujate vaheline seos. Negatiivse kordaja puhul on ka seos negatiivne. Levinumad korrelatsioonimeetodid on näiteks Pearsoni, Kendalli, Spearmani või Point-Biseriali korrelatsioon.

Nendest kõige laialdasemalt on kasutuses Pearsoni korrelatsioon, mille puhul on tegemist lineaarse korrelatsiooniga. Pearsoni korrelatsiooni kasutatakse ka antud töös. Üldiselt loetakse tugevaks seoseks juhtusid, kui Pearsoni korrelatsioonikordaja absoluutväärtus on üle

0,5. Antud mooduse kasutamiseks korrelatsiooni leidmisel tuleb kindel olla, et mõni kasutatud väärtus ei ole tekkinud teise tagajärjel. Lisaks on Pearsoni korrelatsioonikordaja kasutamisel vaja jälgida, et andmestikus puuduksid erandid, kuna need võivad antud korrelatsioonikordaja väärtust väga oluliselt mõjutada [30].

5. Loodud Lahendus

5.1 Eesmärk ja arhitektuur

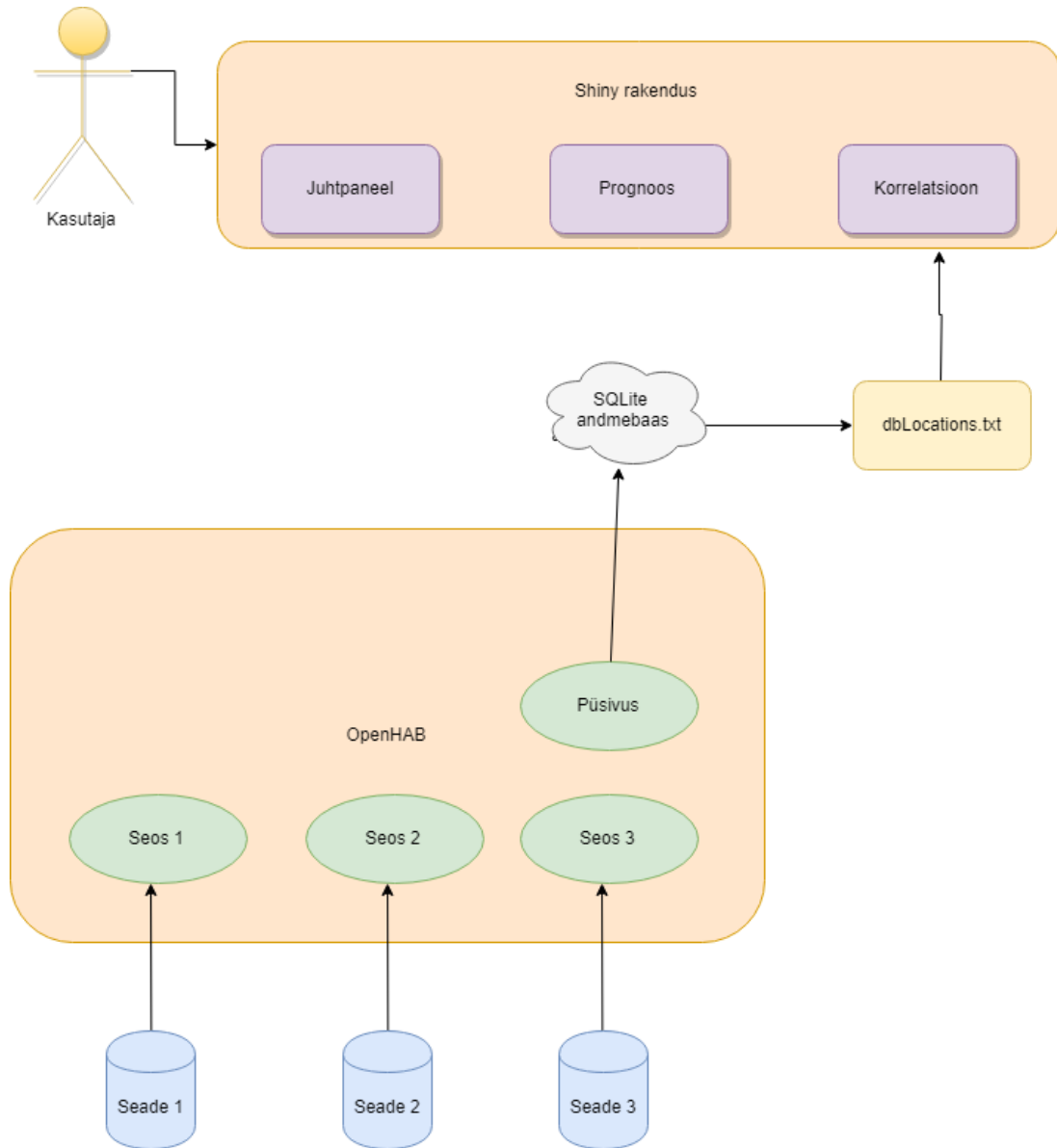
Eesmärk on luua baaslahendus näidisrakendusega, mis suudab ühenduda OpenHAB-i süsteemi poolt hallatavate SQLite andmebaasidega, sealt saadud andmed töödelda ning kuvada kolmes vaates. Lahendus ei ole mõeldud igapäeva elus kasutamiseks, vaid pigem arendajatele malliks, millest on võimalik lihtsasti OpenHAB-i ja R-i põhiseid rakendusi luua.

Näidislahendus koosneb OpenHAB-i keskkonnast, kuhu on seoste abil ühendatud seadmete asjad. Asjad sümboliseerivad antud struktuuris füüsilisi seadmeid. Seosed suhtlevad OpenHAB-i keskkonnaga andes sinna pidevalt infot enda seisust ning saades sealt vastu käsklusi. Seadmetelt saadud info salvestab OpenHAB antud näitel SQLite, kuid tegelikkuses ühte paljudest saadaval olevatest andmebaasidest. Andmebaasist tehtud päringute abil jõuavad andmed Shiny rakendusse, milles need vastavalt kasutaja valikule graafiliselt kuvatakse.

Esimeses vaates kuvatakse kasutaja poolt valitud andmebaasist andmed traditsioonilise juhtpaneeli lahendusena. Juhtpaneel koosneb kahest graafikust ning neljast graafilisest näidikust, mis ennast uute andmete saabumisel uuendaks. Kasutajal peab olema võimalus juhtpaneeli enda soovi järgi valida graafikute või näidikute poolt kuvatavad väärtused. Mõlema graafiku jaoks on selleks loodud rippmenüü ning näidikute jaoks märkeruudud, kus saab aktiivseks teha kuni neli andmebaasi tabelit.

Teises vaates on eesmärk luua näiterakendus prognoosist mingile väärtusele. Prognoosis peaks kasutama ARIMA mudelit ning see peaks kajastuma joondiagrammina. Lisaks võiks olla graafikul kuvatud ka prognoositava väärtuse eeldatav viga.

Kolmandas vaates on eesmärk luua näiterakendus kahe seadme andmete vahelise korrelatsiooni kuvamiseks. Korrelatsiooni leidmiseks peaks kasutama Pearsoni korrelatsioonikordajat.



Joonis 5: Arhitektuur

5.2 OpenHAB-i seadistamine

Järgnevalt on toodud juhendid Windowsi operatsioonisüsteemil OpenHAB-i seadistamiseks. Kasutajal peab olema installitud Java platvorm ning defineeritud JAVA_HOME süsteemi keskkonnamuutuja. OpenHAB-i loojad soovivad selleks kasutada Zulu nimelist platvormi [31], kuid suurema osa funktsionaalsuse jaoks sobib ka Oracle Java. Soovitatakse hoiduda OpenJDK platvormist, kuna kasutajatel on sellega tekkinud probleeme, näiteks 20

minutiline käivitamise aeg. Lisaks soovitatakse kasutada 32-bitilist Java virtuaalmasinat juhul, kui OpenHAB jookseb ARM (ing k. Advanced RISC Machine) struktuuriga protsessoril. Töö kirjutamise seisuga 25ndal aprillil 2018 ei ole toetatud Java 9-s version ning minimaalselt soovitatakse 8-nda versiooni korral kasutada väljalaset 101 [32].

OpenHAB-i paigaldamiseks peab kasutaja alla laadima OpenHAB-i zip-faili ning pärast selle lahtipakkimist käivitama start.bat nimelise skripti. Seejärel on võimalik aadressil <http://localhost:8080> siseneda OpenHAB-i veebipõhistesse kasutajaliidesesse, kust on juba lihtne süsteemi seadmeid liita.

Seadmete liitmisest natuke keerulisem on sealt saadud andmete salvestamine andmebaasi. Selleks pakub OpenHAB n-ö püsivusteenuseid (ing k. *persistence services*). Autor kasutab antud töös JDBC (*Java Database Connectivity*) püsivusteenuse poolt pakutud SQLite andmebaasi lahendust. Antud juhend on SQLite andmebaasi näitel, kuid ka teist tüüpi andmebaaside liitmine on küllaltki analoogne.

Esiteks tuleb *services/jdbc.cfg* failis lisada rida `url=jdbc:sqlite:./testSqlite.db`, kus juures faili nime (antud näitel *testSqlite.db*) võib kasutaja valida oma soovi järgi. Pärast seda on vaja defineerida vaikimisi püsivusteenus. OpenHAB võimaldab andmebaaside jaoks valida ka salvestusstrateegia ehk üldise loogika, mille järgi andmeid salvestatakse. Antud lahenduses kasutusel olev strateegia salvestab kõikide seadmete andmeid 10 minutilise intervalliga. Selleks on vaja lisada faili *persistence/jdbc.persist* järgnevad read:

```
org.eclipse.smarthome.persistence:default=jdbc
Strategies {
    everyHour : "0 0 * * * ?"
    everyDay : "0 0 0 * * ?"
    everyMinute : "0 * * * * ?"
    default = everyChange
}
Items {
    * : strategy = everyMinute
}
```

5.3 Süsteemi komponendid

Shiny rakendusele iseloomulik ülesehitus näeb ette, et rakendus koosneb kahest failist: *ui.r* ja *server.r*. Esimene neist sisaldab endas ainult kasutajaliidesesse kuuluvaid komponente nagu graafikud jms. *Server.r* failis toimub kogu tegelik rakenduse loogika [33].

Andmebaasid

Rakenduse töö algab alati andmete pärimisest dbconf.txt failis kasutaja poolt eelnevalt defineeritud andmebaaside aadressidelt. Aadressid võivad olla üle veebi kättesaadavad või paikneda kasutaja arvutis. Arvutis asuvate andmebaasi andmetega on võimalik koheselt ühendust saada. Internetis asuvate andmetega see võimalik pole, kuna SQLite andmebaas seda erinevalt näiteks MySQL-ist ei võimalda. Seega tuleb esmalt andmebaas ajutisse faili alla laadida, mis pole suur probleem väiksemate andmebaaside korral, kuid suuremate andmebaaside korral paljastab lahenduses puudujäägi. Andmebaasi operatsioonide läbi viimiseks kasutati DBI, dplyr ja RSQLite teeke.

Andmetöötlus

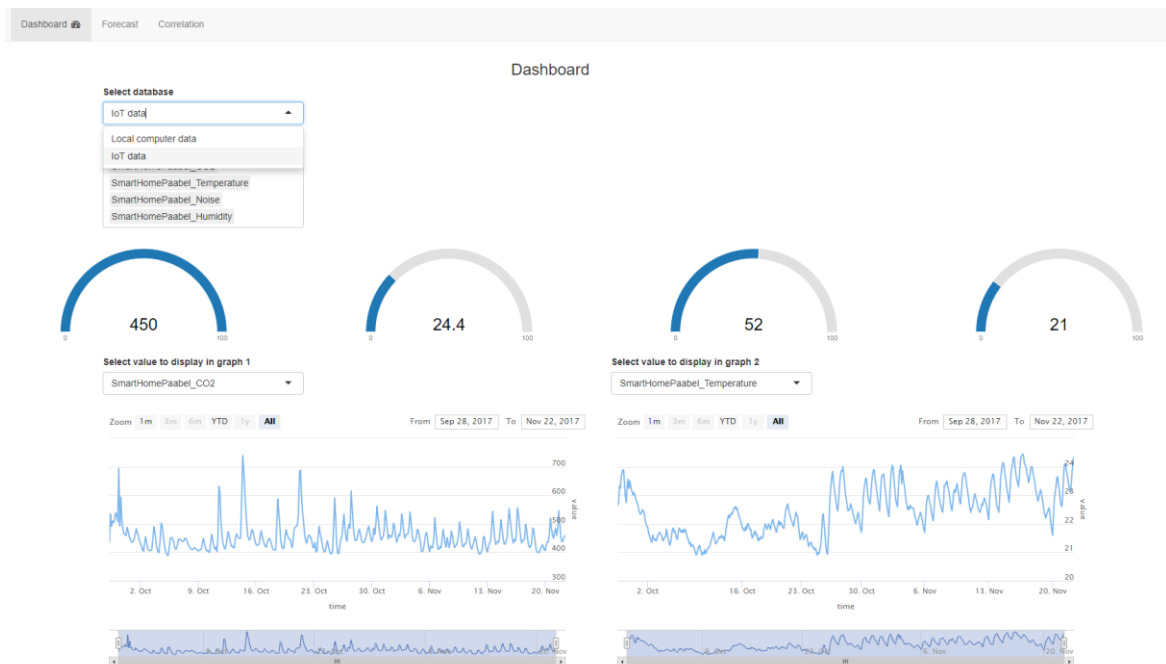
Pärast andmete pärimist andmebaasist töödeldakse andmed vajalikule kujule. Kuigi andmeid salvestatakse küll seadistuse järgi iga 10 minuti tagant, ei saa selles siiski alati kindel olla. Võib juhtuda, et mõni seade lakkab paariks minutiks töötamast ning pärast tagasi võrku ühendades ei sobi saadud andmed enam graafikute loomiseks, kuna pole püsivalt sama ajatempliga. Selle pärast on vaja andmeid interpoleerida. See tähendab vahepealsete puuduvate väärtuste lisamist andmehulka. Autor loob andmete interpoleerimiseks listi andmete esimesest ajatemplist viimaseni, kus on väärtused iga 10 minuti tagant. Pärast seda, kasutades `full_join()` funktsiooni, ühendatakse list algse andmehulgaga ning eemaldatakse tühjad väärtused kohtades, kus alguses andmehulgas väärtus leidis. Puuduvate väärtustega kohtades jäävad seega alles tühjad väärtused, mis `na.approx()` funktsiooniga asendatakse. Asendamisel kasutatakse lineaarset interpoleerimist, mis tähendab, et väärtuseks saab antud ajatemplile eelneva ning järgneva väärtuse aritmeetiline keskmine. Pärast andmete interpoleerimist leiab autor kasutusel oleva andmebaasi tabelitest need, kus on arvulised väärtused ning mis on seega graafikutes kasutatavad.

Kasutajaliides

Juhtpaneel

Ekraanivaatel 1 on kuvatud kasutajaliidese esimene vahekaart, juhtpaneel. Pärast töötlemist saadetakse andmed juhtpaneeli kasutajaliidese elementideni. 10 minuti möödumisel tehakse andmebaasi uus päring kõige viimase väärtuse saamiseks ning see lisatakse juba olemasolevale andmehulgale ning kasutajaliidese elemendid uuendatakse. Autori poolt loodud paneelis on kasutajal võimalik navigeerida erinevate andmebaaside vahel, võimaldades hõlpsasti

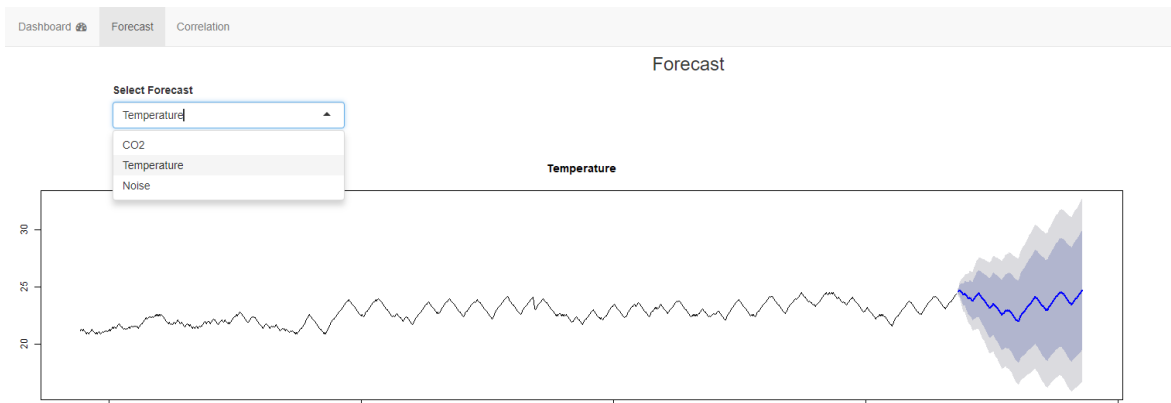
andmeid eraldi hoida. Lisaks sellele on kasutajal võimalik valida, millised väärtused peaks juhtpaneelis kajastuma. Kasutajaliidese elementideks on C3 ja htmlwidgets-i teekide abil loodud näidikud. C3 on teek, mis põhineb JavaScripti teegi Data-Driven Documents põhjal, võimaldades R-i rakendustes kasutada JavaScripti funktsioone [34]. Lisaks sellele on kasutajaliidese näidikute all 2 graafikut, mille abiks on kasutatud highcharter teeki. Kasutajaliidese elemendid muudavad enda väärtusi iga 10 minuti tagant.



Ekraanivaade 1: juhtpaneel

Prognosis

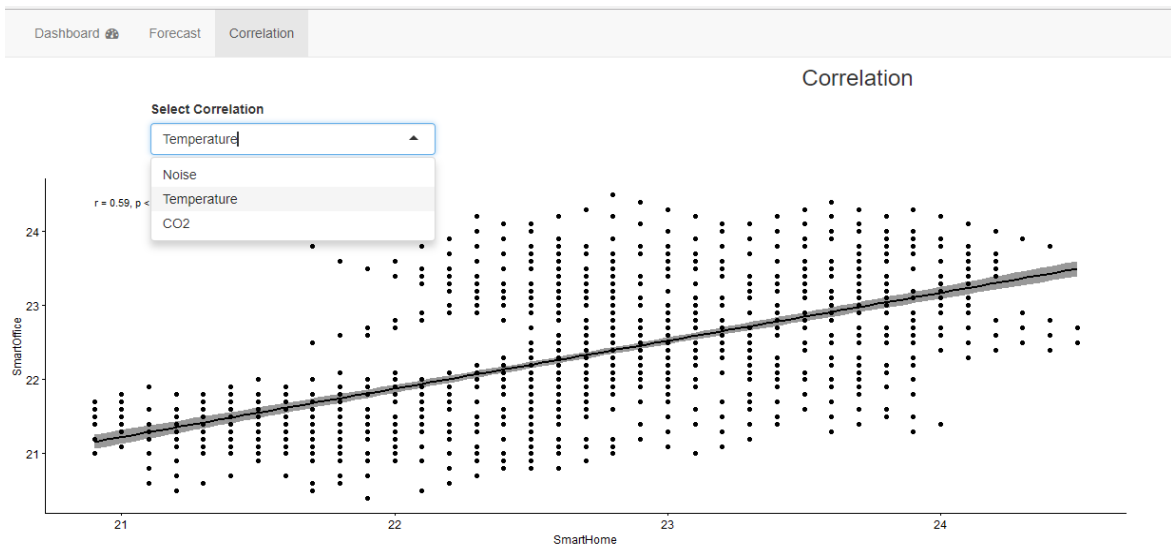
Ekraanivaatel 2 on kuvatud näidislahenduse teine osa, ARIMA mudeli abil tulevaste väärtuste prognoosimine, kus kuvatakse prognoos kolme väärtuse: süsihappegaasi, temperatuuri ning heli taseme kohta. Kuna prognoosimisel on parema tulemuse nimel soovitatav kasutada perioodiliselt muutuvaid andmeid, siis kasutatakse selleks algset andmebaasi, mida kasutajaliidese muuta ei saa. Ka ARIMA prognoosi programmeerimisel oli oluline kasutada interpoleeritud väärtusi. Kuna mudeli treenimine on küllaltki aega nõudev protsess ning enustuse tegemiseks on sellest täitsa piisavalt, siis on kasutuses ainult tabeli viimased tuhat väärtust.



Ekraanivaade 2: ARIMA prognoos

Korrelatsioon

Ekraanivaatel 3 on kuvatud näidislahenduse viimane vahekaart, kus paiknevad Pearsoni korrelatsioonigraafikud. Sarnaselt ARIMA mudeliga on ka korrelatsiooni leidmisel kasutatud kindla andmebaasi väärtusi, nimelt on võrreldud Tartu Ülikooli pilvelabori targa kodu ning targa kontori temperatuuri, heli taset ning süsihappegaasi tasemeid. Selgus, et ainult kahe toa temperatuuride vahel oli tugev seos, kordajaga 0,59.



Ekraanivaade 3: Korrelatsioon

Kasutamine

Rakenduse kasutamiseks tuleb esmalt jooksutada fail *install.r*, mis laeb alla ning salvestab kõik antud töös kasutatud teigid. Seejärel tuleb lisada faili *databases.txt* kõik soovitud andmebaaside veebi- või kohalikud aadressid ning rakendus käivitada.

6. Kokkuvõte

Käesoleva bakalaureusetöös loodi OpenHAB-il ja R-il põhinev asjade interneti andmete analüüsimist hõlbustav näidislahendus. Lahenduse abil on võimalik erinevate andmebaaside andmete visualiseerimine, lihtsamate analüüsimeetodite teostamine ning see on tulevikus arendajatele lihtsasti ülesseatav ja kohandatav.

Lahendus kasutab OpenHAB-i süsteemi asjade interneti seadmete halduseks ja andmete salvestamiseks ning R-i ja selle paketti Shiny salvestatud andmete visualiseerimiseks. OpenHAB salvestab kasutaja seadmete informatsiooni iga 10 minuti tagant SQLite andmebaasi, sealt saab rakendus päringuga vajalikud andmed, töötleb vajalikule kujule ning visualiseerib kolmel vahekaardil. Esimene vahekaart sisaldab endas juhtpaneeli, kus asuvad 2 graafikut ning 4 vidinat, millega kuvatakse kasutaja seadmete viimane andmebaasi salvestatud väärtus. Teine vahekaart sisaldab endas ARIMA mudeli põhjal tehtud ennustust kasutaja andmebaasist saadud seadme väärtustele. Kolmas vahekaart sisaldab endas Pearsoni korrelatsiooni kahe kasutaja seadme väärtuste vahel.

Käesoleva töö käigus loodud lahenduse põhjal on võimalik luua kodudes kasutatavaid rakendusi ning seega seda mitmes suunas edasi arendada. Näiteks on võimalik andmebaasi salvestusstrateegiat muutes saada seadmete kohta infot väiksema ajavahemiku tagant. Samuti on võimalik vastavalt vajadusele kasutajaliidese elemente lisada või muuta ning korrigeerida analüüsimeetodite mudeleid vastavalt oma soovile.

7. Viidatud kirjandus

- [1] E. Brown, „Who Needs Internet of Things,“ 11 5 2018. [Võrgumaterjal]. Available: <https://www.linux.com/news/who-needs-internet-things>.
- [2] F. M. J. S. a. H. Z. Daniel A. Keim, „Challenges in Visual Data Analysis,“ *Tenth International Conference on Information Visualisation (IV'06)*.
- [3] „Introduction,“ 11 5 2018. [Võrgumaterjal]. Available: <https://www.openhab.org/introduction.html>.
- [4] D. Smith, „R moves up to 5th place in IEEE language rankings,“ 11 5 2018. [Võrgumaterjal]. Available: <https://www.r-bloggers.com/r-moves-up-to-5th-place-in-ieee-language-rankings/>.
- [5] „Coke,“ 11 5 2018. [Võrgumaterjal]. Available: https://www.cs.cmu.edu/~coke/history_long.txt.
- [6] S. V. P. Tanjim T. Mulani, „Internet of Things,“ *International Research Journal of Multidisciplinary Studies*, p. 1, 2016.
- [7] M. G. M. M. M. A. M. A. Ala al-Fuqaha, „Internet of Things: A Survey on Enabling,“ *IEEE Communication Surveys and Tutorials, Vol 17, No 4, Fourth Quarter*, p. 2, 2015.
- [8] K. Raja, „Mida tähendab asjade internet ja kuidas see meie elu paremaks teeb?,“ *Õhtuleht*, 17 2 2016.
- [9] „Tark Tartu,“ [Võrgumaterjal]. Available: www.tarktartu.ee. [Kasutatud 12 5 2018].
- [10] „Data analysis,“ 11 5 2018. [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Data_analysis.
- [11] J. G. Koomey, „Best Practices for Understanding Quantitative Data,“ p. 2, 2006.
- [12] C. French, „Data Processing and Information Technology,“ *Data Processing and Information Technology*, Oliver and Chapman's, 1972, p. 80.
- [13] „Exploratory data analysis,“ 11 5 2018. [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Exploratory_data_analysis.
- [14] M. Friendly, „Milestones in the history of thematic cartography, statistical,“ 2009.
- [15] C. Ware, *Information Visualization: Perception for Design*, Morgan Kaufmann, 2012.
- [16] G. Z. D. Z. Lukas Smirek, „Towards Universally Usable Smart Homes – How Can MyUI, URC and openHAB,“ *CENTRIC 2014 : The Seventh International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services*, 2014.
- [17] „HABPanel,“ 11 5 2018. [Võrgumaterjal]. Available: <https://docs.openhab.org/configuration/habpanel.html>.
- [18] R. Holloway, „Opensource,“ 11 5 2018. [Võrgumaterjal]. Available: <https://opensource.com/life/16/4/automating-your-home-openhab>.
- [19] „Smarthome,“ Eclipse, [Võrgumaterjal]. Available: <https://www.eclipse.org/smarthome/>. [Kasutatud 12 5 2018].
- [20] „Documentation,“ 11 5 2018. [Võrgumaterjal]. Available: <https://docs.openhab.org/introduction.html>.

- [21] „Concepts,“ 11 5 2018. [Võrgumaterjal]. Available: <https://docs.openhab.org/concepts/index.html>.
- [22] „Add-ons,“ 11 5 2018. [Võrgumaterjal]. Available: <https://docs.openhab.org/addons/index.html>.
- [23] „JDBC Persistence,“ 11 5 2018. [Võrgumaterjal]. Available: <https://docs.openhab.org/addons/persistence/jdbc/readme.html>.
- [24] „About R,“ 11 5 2018. [Võrgumaterjal]. Available: <https://www.r-project.org/about.html>.
- [25] „Shiny,“ 11 5 2018. [Võrgumaterjal]. Available: <http://shiny.rstudio.com/>.
- [26] „About SQLite,“ 11 5 2018. [Võrgumaterjal]. Available: <https://www.sqlite.org/about.html>.
- [27] „Steps in choosing a forecasting model,“ 11 5 2018. [Võrgumaterjal]. Available: <https://people.duke.edu/~rnau/411fcst.htm>.
- [28] „Data transformations and forecasting models: what to use and when,“ 11 5 2018. [Võrgumaterjal]. Available: <https://people.duke.edu/~rnau/whatuse.htm>.
- [29] R. Upadhyay, „ARIMA Models,“ 11 5 2018. [Võrgumaterjal]. Available: <http://ucanalytics.com/blogs/arima-models-manufacturing-case-study-example-part-3/>.
- [30] „Correlation (Pearson, Kendall, Spearman),“ 11 5 2018. [Võrgumaterjal]. Available: <https://www.statisticssolutions.com/correlation-pearson-kendall-spearman/>.
- [31] „Zulu,“ [Võrgumaterjal]. Available: <https://www.azul.com/downloads/zulu/zulu-windows/>. [Kasutatud 12 5 2018].
- [32] „Installation,“ 11 5 2018. [Võrgumaterjal]. Available: <https://docs.openhab.org/installation/index.html>.
- [33] „Basics,“ 11 5 2018. [Võrgumaterjal]. Available: <https://shiny.rstudio.com/articles/basics.html>.
- [34] „C3,“ 11 5 2018. [Võrgumaterjal]. Available: <https://github.com/c3js/c3>.
- [35] S. A. Bernhardt, „Seeing the Text,“ *ACM SIGDOC Asterisk Journal of Computer Documentation*, kd. 16, nr 3, pp. 3-16, September 1992.
- [36] A. Dillon, „Reading from paper versus screens: a critical review of the empirical literature,“ *Ergonomics*, kd. 35, nr 10, pp. 1297-1326, 1992.
- [37] J. Foster ja P. Coles, „An Experimental Study of Typographic Cueing in Printed Text,“ *Ergonomics*, kd. 20, nr 1, pp. 57-66, 1977.
- [38] K. Martin, „Common Mistakes Made When Writing a Book in Microsoft Word,“ 01 November 2012. [Võrgumaterjal]. Available: <http://www.self-pub.net/blog/common-mistakes-made-when-writing-a-book-in-microsoft-word/>.
- [39] Nathan, „Bibliography & Citations 102 – Building Custom styles,“ April 2009. [Võrgumaterjal]. Available: <http://blogs.office.com/2009/04/29/bibliography-citations-102-building-custom-styles/>.
- [40] A. J. Wilkins ja M. I. Nimmo-Smith, „The clarity and comfort of printed text,“ *Ergonomics*, kd. 30, nr 12, pp. 1705-1720, 1987.
- [41] Loodus- ja täppisteaduste valdkonna nõukogu, „Lõputöödele esitatavad nõuded ja kaitsmise kord,“ 28 jaanuar 2016. [Võrgumaterjal]. Available: <http://reaalteadused.ut.ee/et/oppetoo/loputoodele-esitatavad-nouuded-kaitsmise-kord>. [Kasutatud 20 veebruar 2016].

- [42] Council of Faculty of Science and Technology, “Requirements for Graduation Thesis and Procedure for Defence in the Faculty of Science and Technology,” 28 January 2016. [Online]. Available: <http://realteadused.ut.ee/en/tuition/requirements-graduation-thesis-and-procedure-defence-faculty-science-and-technology>. [Accessed 20 February 2016].

Lisad

I. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, **Ako Tõnissoo**,

(autori nimi)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

R-i põhised targa kodu analüüsilahendused,

(lõputöö pealkiri)

mille juhendaja on Jakob Mass,

(juhendaja nimi)

1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **15.05.2018**