TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Infosüsteemide õppetool

IDU40LT

Eduard Mazurtšak 104224IABB

TARGA MAJA UNIVERSAALNE PLATVORMI HÄÄLESTUSSÜSTEEMI ARENDAMINE

bakalaurusetöö

Juhendaja: Enn Õunapuu
PhD
dotsent

Tallinn 2016

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Eduard Mazurtšak

08.05.2016

Annotatsioon

Käesoleva bakalaurusetöö eesmärgiks on targa maja erinevatega komponentide ja andurite kokku panemise võimalus, mis võib olla juhinud ühest kohast. Targa maja juhtimise võimaluse mugavus kasutajatele keskkonnas ning targa maja automatiseerimine universaalsete käskudega.

Töö põhifookus on leida vajalikud kriteeriumid targa maja juhtimis keskkonnale et see oleks võimalik mugavalt kasutada kõikidele, kes oskab kasutada arvutit algtasemel.

Lõputöö tulemusena on pakutud lahendus juhtimispunkti targale majale, et ühendada kõik „targad“ seadmed. Samuti ka pakutud mugav keskkond tavalise kasutajatele, et automatiseerimine toimus.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 27 leheküljel, 6 peatükki, 7 joonist.

Abstract

The Development of Interface for Smart Home Universal Platform

The main purpose of this bachelor is to find out opportunities to unite smart home control system, if components and sensors released by different manufacturer. Smart home control by convinient interface for users and smart home automation by universal commands.

The main focus of this thesis is to find out necessary criteria for smart home control interface in order to be convenient for everyone who manage to use computer at least at base level.

As the result of the thesis solution for uniting all intelligent devices of smart home into unite control system is proposed. Also, solution for comfortable interface of automation smart home was suggested.

The thesis is in Estonian and contains 27 pages of text, 6 chapters, 7 figures.

Lühendite ja mõistete sõnastik

JVM	Java Virtual Machine –käivitab Java lähtekoodi sõltumatu platvormist.[15]
Z-Wave	Juhtmevaba võrgu tehnoloogia, mis on välja arendatud Zensys Incorporationi poolt. See on juhtmevaba kommukatsiooni protokoll, mida kasutatakse ennekõike intelligentsete kodulahenduste juhtimiseks.Pakub kahepoolsete kommunikatsiooni ehk pealse teate saatmist, saab tagasi ka teadet, kas see jõudis sihtmärki.[16]
KNX	Andeedastusprotokoll, mis on välja töötatud ühtse andmedastussini ning ei sõltu riistvarast. KNX-tähisega seadmed järgivad omavahel samu reegleid.[17]
Bluetooth	Juhtmevaba võrgu tehnoloogia, mille abil suudavad erinevad seadmed nagu mobiiltelefonid, arvutid omavahel suhelda.Põhieesmärk on andmeedastuste ja juhtmeühenduste asendamine.[18]
OSGI	Open Services Gateway Initiative – dünaamilisi mooduli süsteemi ja Java-põhiste rakenduste teenuste platvormi spetsifikatsioon[19]
IDE	Integrated development environment-Integreeritud programmeererimiskeskkond - on kogum tööristu tarkvara arendamisel.[20]
Java Servlet	Servlet on Java kasutajaliides, mille realiseerumine laieneb funktsionaalsuse serveri võimeid. Servlet suhtleb kasutajatega taotluse-vastuse kaudu.Servletid on levinud veebiserveri laiendamiseks.[21]
Xtend	On üldotstarbiline kõrgtasemeline programmeerimiskeel JVMle. Süntaksilt ja semantiliselt ta on java programmeerimiskeel, vaid keskendub rohkem täiendava funktsioone, nagu pikendamise meetodeid ja operaatori ülekoormus.[10]
Philips Hue	Patenteeritud oma disain Philips, mis on ZigBee kontrollitud LED pirn.[22]
Insteon	Koduautomaatika tehnoloogia, mis võimaldab valgustuslülitid, tuled, termostaadid, andurid ja muud elektriseadmed omavahel suhelna läbi elektriliinid raadiosagedusliku side.[23]
MVC	Model-View-Controller, tarkvara disainimuster[24]

ZigBee	Traadita kohtvõrgu tehnoloogia, mis on mõeldud kasutamiseks kodudes, büroohoonetes ja tööstuses. Vastab väikese andmekiirusega võrkude standardile IEEE 802.15.4.[25]

Sisukord

1 Sissejuhatus 9
1.1 Taust ja probleem 9
1.2 Ülesande püstsitus 10
1.3 Metoodika 10
1.4 Ülevaade tööst 10
2 Targa maja kokku panemine 11
2.1 OpenHAB süsteem 11
2.2 OpenHAB struktuur 12
3 Programmeerimine ja automatiseerimine 14
3.1 Programmeerimiskeel 14
3.2 Reeglite süsteem 14
3.3 Programmeerimiskeskkond 16
4 Keskkond automatiseerimiseks 17
4.1 Loogika arendamiskeskkonnas 17
4.2 Keskkonna prototüüp 18
4.3 Reeglite genereeremine 22
5 Kokkuvõte 23
6 Summary 24
Kasutatud kirjandus 25

Jooniste loetelu

Joonis 1. OpenHAB tööskeem 13
Joonis 2. Reegel „Fire Alarm“ 15
Joonis 3. Lehte plaan 18
Joonis 4. Reeglite leht 19
Joonis 5. Tingimuste leht 20
Joonis 6. Tegevuste leht 21
Joonis 7. Genereeritud reeglid 22

1 Sissejuhatus

Tehnoloogia ja teaduslikud avastused muudavad meie elu erinevates valdkondades. Kui vaadata autosid või mobiiltelefone, siis viimase 20 aasta jooksul tehnoloogiad sellistes valdkondades nagu turvalisus, keskkonna sõbralikus, multimeediasüsteem muutusid igal pool. Sellese nimekirja võib lisada ka majad. Terviklikud targa maja lahendused võivad kontrollida ja automatiseerida palju asju majas, et teha omanikule elu mugavaks ja kergemaks. Selles majas peaks kindlasti olema võimalus seadistada süsteem oma soovi järgi.

Töö eesmärk on välja arendada lahendus selleks, et tavakasutaja suudaks ise automatiseerida oma tarka maja ühest kohast, ka sellisel juhul, kui tark maja komponendid on erinevatelt tootjatelt ja töötavad erinevate protokollidega. See võib olla vaja ka sellistele inimestele, kes ei taha terve maja automatiseerida, vaid mõned osad või mõned ruumid.

1.1 Taust ja probleem

Targa maja tervislik süsteem baasfunktsioonidega, nagu valgustus ja kütus maksab umbes 10000 euro. Kui inimene tahab, et majas oleks kõik automaatne, nagu kardinad ja uksed siis selline targa maja süsteem nimetatakse mitte baas, vaid arendatud või isegi lux ja selline süsteem maksab vähemalt 15000 euro [1].

Nendel inimestel, kellel ei ole võimalust kohe kogu tarkmaja süsteem osta või nendel kellel ei ole vaja kõik automatiseerida, näiteks, vaid lihtsalt ühes toas valgustus automatiseerida, siis need inimesed võivad osta ka eraldi komponendid erinevatelt tootjatelt. Sellisel juhul nad saavad hakkama palju väiksema eelarvega. Kulusid võib vähendada umbes 70-80% [2].

Siis tekkib selline probleem, et kõik komponente ei saa koos juhtida ühest kohast, selle pärast, et nendel on erinevad protokollid. Selleks hetkeks on olemas

universaalne platvorm nimega OpenHAB mis ühendab kõik seadmed targa maja ühte juhtimissüsteemi.

1.2 Ülesande püstsitus

Käesoleva töö põhieesmärk on välja pakkuda lahendus, millega kasutaja võiks häälestada oma tark maja. Et see lahendus oleks arusaadav kõigile, kes oskab lihtsalt arvutit kasutada, ja ei oska programmeerida. Veel see lahendus peaks olema universaalne, et kui targa maja süsteem koosneb erinevatest süsteemidest ja erinevatest protokollidest, et seda oleks võimalik juhtida ühest punktist.

1.3 Metoodika

Selleks, et targa maja universaalse platvormi häälestusüsteemi arendada on vaja leida sobiv platvorm, analüüsida seda ning samuti leida tähtsad nõuded mugava häälestussüsteemi jaoks. Nõuete analüüsi puhul kasutan oma isiklikku kogemust ning täiendavat kirjandust.

1.4 Ülevaade tööst

Esimeses peatükis vaadetakse läbi milleks open-source projekt OpenHAB sobib. Kuidas toimub see protsess, et erinevad andurid ja seadmed saaksid signaale vahetada ja kuidas kogu süsteem võib olla kontrolli all ühest kohast.

Teises peatükis vaadetakse kuidas programmeerida ja seadistada OpenHAB. Milliseid programme on vaja ja mida peab tegema selleks, et oma targa kodu automatiseerimine ei vajaks eri programmeerimisoskusi.

Kolmandas peatükis pakutakse lahendust, kuidas teha nii, et automatiseerimise käsu andmiseks ei oleks vaja kasutajal alati programmeerida. Selleks piisab üks kord panna kood süsteemi, ja siis seda võib alati kasutada.

2 Targa maja kokku panemine

Selle töö kirjutamise ajal turul pakutakse palju variante oma tavalisest majast teha tark maja[3]. See võib olla terviklik lahendus ühelt tootjalt, näiteks Telia lahendus või eraldi komponentide ja andurite kokku panemise lahendus. Teisel juhul tekkib küsimus selles, et kuidas kõik need targa maja funktsioone kasutada. Erinevad andurid ja komponendid töötavad erinevate protokollidega ja ei saa töötada ühes keskkonnas. Sellisel juhul targa maja kasutaja peab seadistama kõik erinevad komponendid eraldi pultidega. Näiteks, kui inimene tahab, et kui ta avab ukse ja astub tuppa siis seda märkab Fibaro andur, kardinad Z-wave pannakse automaatsed kinni ja Philips Hue Led lampid pannakse põlema siis ta peab kohe kasutama vähemalt kahte pulti. Üks kardinate ja teine valgustuse jaoks. Aga on arenduses OpenHAB projekt, millega kõike seda saab teha ühest kohast. Selleks et andurid erinevatelt tootjatelt edastaks üksteistele signaale, selleks on võimeline OpenHAB.

2.1 OpenHAB süsteem

OpenHAB tähistatakse nagu Open Home Automation Bus. Selle targa maja automatiseerimise platvorm oli välja lastud 2010.aastal ja sellest ajast tulevad kogu aeg uuendused, mille abil see platvorm on võimeline töötama erinevate protokollidega. Klassikaaliste targa maja süsteemid KNX, Z-Wave, Insteon ja isegi uued värkvõrkud nagu Sonos ja Phillips Hue ka toetatakse OpenHabiga. Kokku see platvorm võimaldab töötada rohkem kui 100 protokolliga [4]. See süsteem on kirjutatud Java keeles kasutades OSGI standardid. Samuti, OpenHab on saanud ka üks väärtusliku auhinna Java maailmas „Duke's choice award 2013" [5]. See auhind antakse ainult kõige suurematele, innovatiliste ja arenenud projektidele mis on tehtud java keeles. Veel üks tugev külg OpenHUBis on see, et selle projektiga võib töötada spetsiaalselt tehtud tarkade majade jaoks framework nimega „Eclipse Smarthome" [6]. Kõik need asjad näitavad, et selline süsteem on kindlasti vajalik, ka sellel juhul, kui inimene tahab moderniseerida või uuendada oma olevat tarka maja.

2.2 OpenHAB struktuur

OpenHab on server, mis võib töötada igasugusel arvutil, igasugusel operatsioon süsteemil, see võib olla raspberryPi, või isegi modem. Peamine tingimus on, et selles seadmes peab olema JVM. Selleks, et panna OpenHAB süsteem käima, on vaja ainult alla laadida distribution OpenHABiga ja kindlasti peab olema installitud JVM. Tänu sellele, et OpenHAB on kirjutanud OSGI tehnoloogiaga siis see on väga kiiresti reageeriv uuendusele programm. See tähendab, et kui uuendada üks osa tarkvarast ei ole vaja teha restarti kogu süsteemile vaid ainult selle osale, kus oli tehtud uuendus ja kogu server töötab korralikult edasi [7].

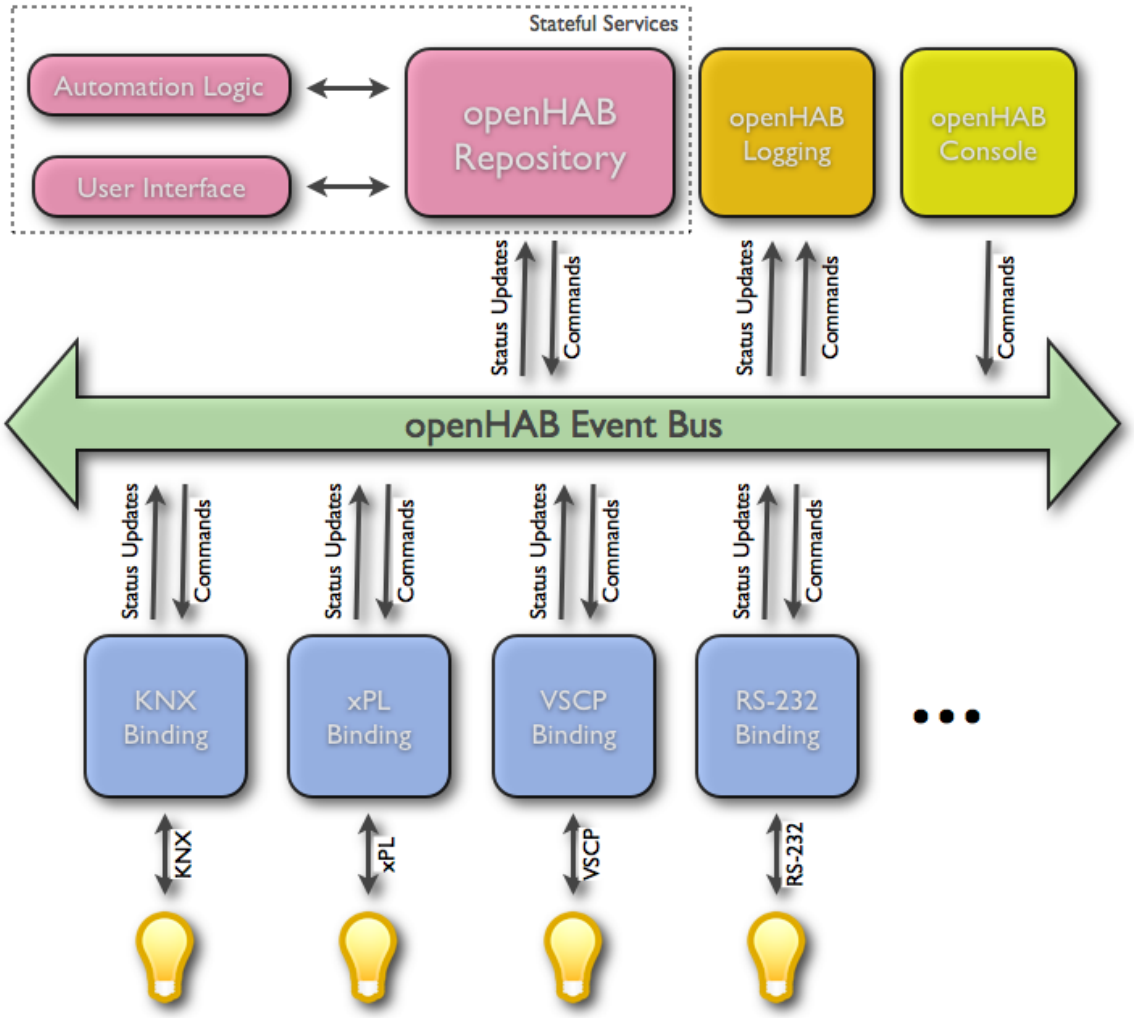
Esimesel joonisel on olemas skeem, kuidas toimub tööprotsess OpenHABis.

Kõige keskmine objekt on OpenHAB Event Bus, mis suhlteb nii erinevate komponentidega, kui ka juhtimiskeskusest tulevate käskudega ning kuhu lähevad andmed anduritest ja komponentidest.

Skeemil on näha ka add-on, ehk siis binding'ud. See tähendab eraldi seadet, mis töötab oma protokolliga ja suhtleb OpenHAB Event Bus'iga ja vajava seadmega. Standard OpenHABis distribution paketis ei ole ühtegi add-on ning kui inimene laeb endale alla, siis ta ise valib mis bindingud temale on vaja, vastavalt sellele mis seadmed temal on.

Siis kui binding ja seade on paigas, võib hakkata juhtima seda seadet universaalsete käskudega. Näiteks, selleks et lülitada midagi sisse või välja on vaja on/off käsud. Kõige tähtsam on see, et kõik seadmed ja andurid, mis vajab automatiseerist saaksid olla tehtud ühel lihtsat ja selgel viisil. Kes on tuttav programmeerimis koodiga saab aru kuidas programmeerida oma tarka maja OpenHABiga. Sellistest lihtsastest lausetest võib koostada nii väikseid skripte, kui ka suuri ja mahukaid automatiseerimise loogikait ja protsesse.

Kui bindingul on suhe OpenHABi ja seadme vahel, siis selleks et oleks võimalik kirjutada funktioone, mille signaale võib võtta anduritest, siis selleks on tehtud item hoidla. See on seotud OpenHAB Event Busiga ja jälgib hetke staatust anduritedes ja komponentides [8].



Joonis 1. OpenHAB tööskeem

3 Programmeerimine ja automatiseerimine

Programmeerimine OpenHABis hakkab toimuma alles siis, kui binding ja item on paigas ja töökorras. Programmeerimine toimub scriptitega ja reeglitega.

Skript OpenHABist on koodi lõik, mis on kirjutatud kasutajaga ja mis võib olla kutsutud mitte üks kord ja mitte ühest kohast [9].

Reeglid OpenHABis on selleks, et kasutada neid automatiseerimise protsessis. Reeglid on hästi integreeritud, samuti nendel on väike suurus, kuid nende võimalused on võimsad. Iga reegel võib olla eraldi käivitatud. Selleks, et käivitada reegel on olemas trigger, mis käivitab skripti ja see toetab ja täidab kõik vajalikud ülesadned. Näiteks, on vaja teha nii, et vahemikus kella 18 - 8 kui keegi astub kööki valgustus pannakse sisse. Kui on sobiv kell ja keegi astub kööki reageerib trigger ja skript lülitab valgustuse köögis.

3.1 Programmeerimiskeel

OpenHABi programmeerimiskeeleks on Xtend. See on java basil asutatud keel. Tegu on tavalise java keelega, kuhu on lisatud paar lisa võimalust, näiteks operaatori ülekoormus või laienemise meetodid [10]. Eclipce'ist on tehtud Xtendi jaoks eraldi oma IDE, kuid seda koodi võib koostada ja käivitada ilma Eclipse platvormita. Selle keele üks tugev külg on selles, et seda koodi võib koostada tavalise java projektiga ja kõik kompeleerib õigesti. Samuti, Xtendil on ka tavalise java raamatukogu toetus, ehk siis tavalised java raamatukogud võib töötada Xtendiga keelega.

3.2 Reeglite süsteem

Reeglid on üks tähtsaim osa targa maja programmeerimises, sest just reeglite loogika ja selle automatiseerimine on targa maja põhi eesmärk. Selleks et töötaks OpenHAB süsteemi reeglid on vaja panna neid õigesse kohta ja õigesse formaati. Ühes failis võib olla mitu reeglit. Reeglite fail on teksti fail, mille peaksid olema impordid, ehk siis vajalikud raamatukogud, ja muutjate deklaratsioonid. Samuti peavad ka olema reeglid ise [12]. Importide kirjutamine ja töötalemine toimub nagu tavalises java keeles. Muutjate deklaratsiooniga peab meeles pidama, et nad on saadaval terves failis.

Veel üks tähtis osa reeglites on trigerid. Selleks, et reegel hakkaks jooksma, peab olema triggeri tingimus täidetud. Reeglis võib olla ka mitu trigerit, et kontrollida mitut tingimust aga vähemalt üks peab reeglis alati olema.

On olemas kolm liiki trigerid:

- Item(Event) – see trigger töötab siis, kui tuleb event busist käsk, või staatuse uuendamine. Näiteks, kui tuli signaal liikumisanduritest, või omanik muutis oma kodus staatust.

- Time-based – see trigger reageerib sellele, kui jõuab teatud aeg. Näiteks, siis kui on kell 12, või kui on talv.

- System-based – see trigger töötab kui terve süsteemi staatus muutub. Näiteks, siis kui süsteem on välja lülitatud, või kui puudub interneti ühendus.

Kõik need asjad kokku moodustavad reeglite süsteem OpenHABis, millega võib kirjutada erinevate seadmete automatiseerimise ja loogika targas majas.

Joonisel 2 koodi lõigus on näide kuidas reegel välja näeb.

Reegel nimega „Fire alarm“. Siis kui Event trigger sai tingimuse kinnituse siis pannakse skript käima, kus on kirjutatud, et pannakse valgustus sisse. Samal ajal nii läbi telefoni kui ka läbi e-maili tuleb häire.

```
rule "Fire alarm"
when
    Item Alarm_Fire received update ON
then
    // turn on all lights
    callScript("lights_on")

    // send notifications
    notifyMyAndroid("Security", "FIRE alarm has been tripped!!!", 2)
    sendTweet("*** FIRE alarm - someone call the fire dept!!! ***")

    // send an email
    sendMail("ben@home.com", "FIRE ALARM!!", "The fire alarm has been activated!!!
end
```

Joonis 2. Reegel „Fire Alarm“

3.3 Programmeerimiskeskkond

Selleks et programmeerimine ja automatiseerimie OpenHABis oleks mugavam on tehtud eraldi IDE nimega OpenHAB Designer. See programm on kättesaadaval erinevatele operatsioon süsteemidele. Selle programmiga on mugav kirjutada kood Xtendis, ta kohe kontrollib programmeerimise loogikat ning kui on midagi valesti, siis see vale koht rõhutab teist värvi ja kohe on näha kus on viga. Vaatamata sellele, et nimes on sõna Designer pole mingit tegemist disainiga vaid on vaja ainult kood kirjutada.

4 Keskkond automatiseerimiseks

Kui inimene oskab programmeerida, siis temale seadistada oma tark maja OpenHABiga ja OpenHABi Designeriga ei ole keeruline ülesanne. Aga kõik ei oska programmeerida ja mitte kõik kes oskavad tahavad programmeerida oma tarka maja. Programmi kirjutamine automatiseerimiseks võib võtta palju aega ja selleks, et see tegevus võttaks vähem aega oleks mõistlik teha keskkond, kust süsteemi võiks seadistada kiiremini.

Üks variant on teha valikvastusega keskkond, kus inimestele pakutakse ette kõik võimalikud variandid automatiseerimiseks ja kasutaja lihtsalt valib endale sobivad seaded. Samuti, kui inimene tahab juba olevat reeglid muuta oleks see protsess kergem. Tihti juhtub nii, et kasutaja ei saa kohe välja mõelda need reeglid, mida temale vaja on. Aga kui kõik võimalikud reeglid on juba valmis ja pakutud siis saab kiiremini valida sobivad seaded.

4.1 Loogika arendamiskeskkonnas

Selleks, et välja pakkuda lahendus on vaja teada, millised kriteeriumid on tähtsad, et teha automatiseerimise keskkond OpenHABiga arusaadavaks ning mugavamaks kasutajatele.

Tähtis on:

- Arusaadavus – et keskkonna loogika oleks kohe arusaadav, mida on vaja teha selleks, et seadistada oma tark maja [12].
- Universaalsus - et see töötaks mitte ainult lauaarvutis, vaid ka mobiiltelefonis, tahvelarvutis [13].
- Kiirus - et targa maja seadistamine oleks võimalikult kiire ja lihtne protsess. See sobib nii uute reeglite koostamiseks, kui ka vanade reeglite modifitseerimiseks või eemaldamiseks [13].

Arusaadavuse saavutamiseks peab automatiseerimise keskkond olema minimalistlik, et kõik mis ei puuduta reegleid või käske ei pea seal süsteemis olema. Kõik vajalik peab juba keskkonnas kättesaadav olema. Võimalikud tingimused, võimalikud tegevsed, võimalikud seaded ja andurid peab süsteemis olema ja kõik mida kasutaja peab tegema on lihtsalt kogu loogika kokku panna.

Universaalsuse saavutamiseks keskond peab olema internetis. OpenHAB süsteem pakub internetist oma targa maja seadistamise võimaluse. Kui OpenHAB töötab javaga siis on mõistlik kasutada servlette javast. Java Servletid on effektiivsed ja stabiilsed [14].

Kiiruse saavutamiks peab programmi kasutamine olema valikvastustega. Kõik variandid, kuidas on võimalik automatiseerida OpenHABi tarka maja peab olema juba programmi sees ja kasutaja teeb valiku vastavalt oma soovile. Siis, kui reeglid on valmis, kasutaja kontrollib neid ja otsustab teha uuenduse süsteemis. Ta vajutab spetsiaalsele nupule ja kõik uuendused kirjutatakse õigesse faili või tehakse uus fail ja see fail läheb sihtkohta, kus ta on juba kättesaadaval OpenHABile.

4.2 Keskkonna prototüüp

Prototüübi plaan koosneb kolmest veebilehest. Joonisel 3 on plaan.

Keskmine objekt on „Reeglite“ leht. Sellelt lehelt võib minna kas „Tegevuse“ lehele või „Tingimuse“ lehele. “Tegevuse“ lehelt võib minna ainult tagasi „Reeglite“ lehele.„Tingimuse“ lehelt võib minna ainult tagasi “Reeglite“ lehele.

Joonis 3. Lehte plaan

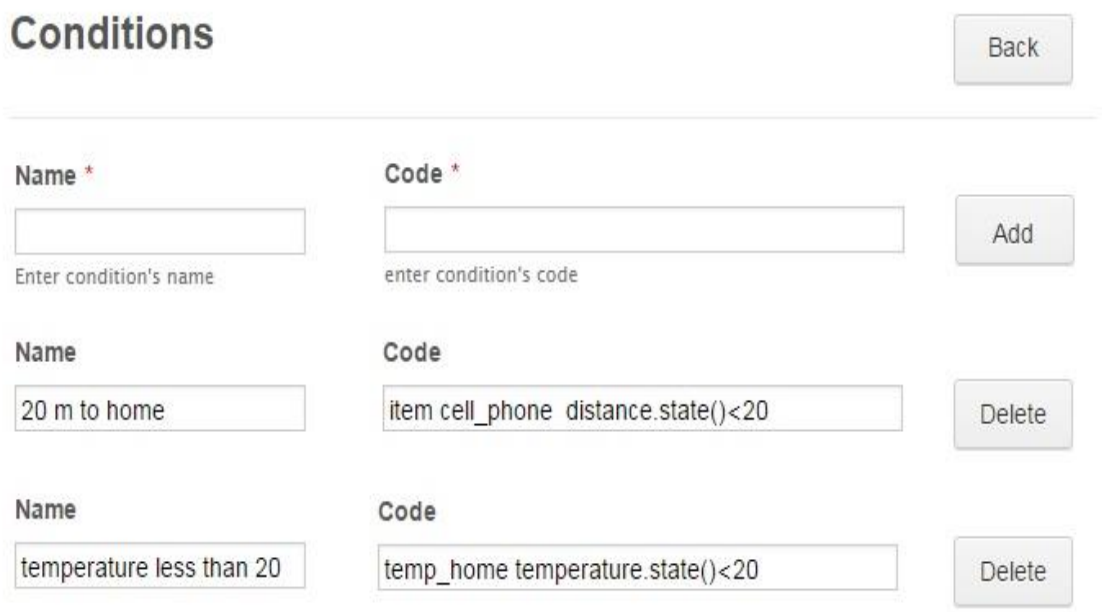
Joonisel 4 on reeglite lehe prototüüp.

Kasutaja, kes soovib teha uut reeglit peab panema reegli nime, ja peale seda valib võimalikest variantidest tingimuse millal see reegel hakkab toimima. Peale seda kasutaja valib sobiva tegevuse, mis tuleb siis, kui tingimus on täidetud. Kui tingimus ja tegevus on valitud, siis kasutaja lisab reegli vajutades „Add rule“. Kasutaja võib vaadata juba olemas olevaid reegleid ning kui mõnda reegelit rohkem ei ole vaja, siis ta eemaldab selle vajutades nuppu „Detele rule“. Kui kasutaja soovib minna teisele lehele, siis ta valib „Conditions“ või „Actions“ vahel. Vajutades nuppu „Generate“ kasutaja genereerib reeglid õigesse faili, et need oleksid kättesaadaval süsteemile.

Joonis 4. Reeglite leht

Joonisel 5 on tingimuste lehe prototüüp.

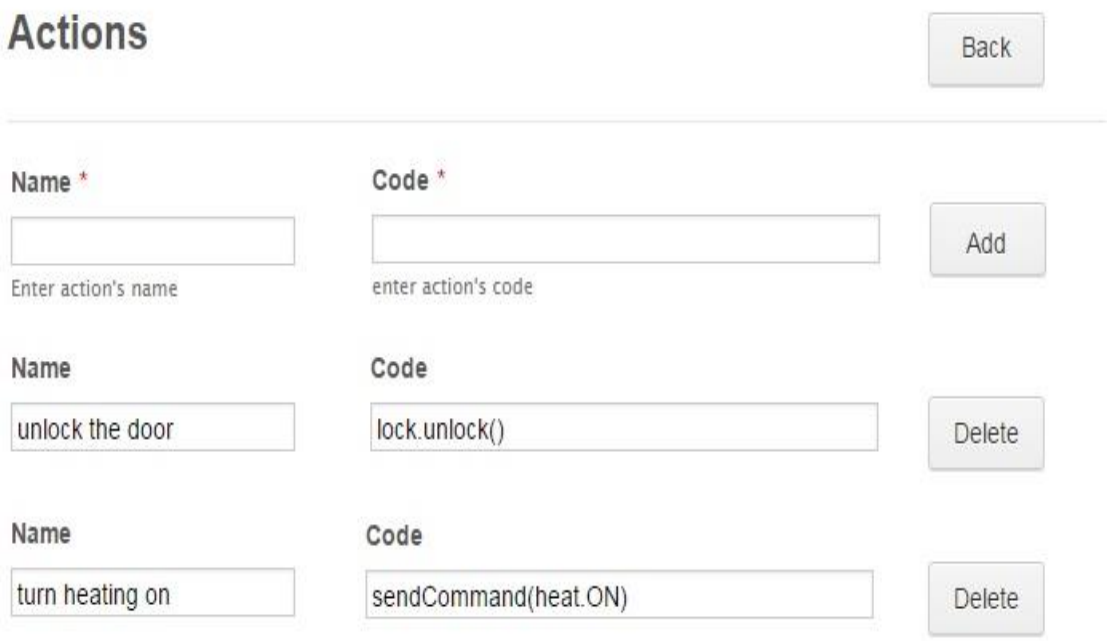
Tingimuse lehel paneb kasutaja koodi lõigud tingimusest ja annab sellele nime. Kui ta vajutab klahvi „Add“ siis koodi lõik on salvestatud ja nüüd võib seda kasutada reeglite koostamisel. Kui mingit tingimust rohkem ei ole vaja, siis kasutaja võib seda eemaldada klahviga „Delete“. Kui inimene soovib, et oleks mitte üks tingimus, vaid rohkem, siis peab ta selleks panema koodi, kus on seda kirjeldatud. Vajutades klahvi „Back“ süsteem tagastab reeglite lehele.



Joonis 5. Tingimuste leht

Joonisel 6 on tegevuste lehe prototüüp.

Tegevuse lehel paneb kasutaja koodi lõigud tegevusest ja annab sellele nime. Kui ta vajutab klahvi „Add“ siis koodi lõik on salvestatud ja nüüd võib seda kasutada reeglite koostamisel. Kui mingit tegevust rohkem ei ole vaja, siis kasutaja võib seda eemaldada klahviga „Delete“. Vajutades klahvi „Back“ süsteem tagastab reeglite lehele.



Joonis 6. Tegevuste leht

4.3 Reeglite genereeremine

Peale seda, kui kõik tingimused ja tegevused on pandud ja reeglid koostatud siis toimub kõige viimane protsess nagu reeglite genereeremine failile. Süsteem lisab reegli nime, “when“ ja „then“, lõpus ka „end“, ülejäänud kood võetakse tingimuste koodilõikudest ning tegevuste koodilõikudest. Kui koodilõigud on õigesti kirjutatud, ja automaatika loogika on valitud siis lausete genereerimisega võib juba seda koodi kasutada oma targas majas. Reeglid võib salvestada otse sinna, kuhu on vaja, et neid ei oleks vaja kuhugi transportida. Raamatukogud mida võib vaja minna peavad failis ette olema.

Joonisel 7 on gereneeritud tulemus sellelt, mis oli eespool näidetel. Kaks reegit nimega „Heat on the way home“ ja „unlock door on way home“on failis kirjutatud Xtendis mis on arusaadaval OpenHABi süsteemile.

```
rule "Heat on the way home"
when
    item cell_phone  distance.state()<20
then
    sendCommand(heat.ON)
end


rule "unlock door on way home"
when
    item cell_phone  distance.state()<20
then
    lock.unlock()
end
```

Joonis 7. Genereeritud reeglid

5 Kokkuvõte

Bakalaureusetöö eesmärgiks oli leida lahendus selleks, et ühendada erinevad komponendid ja andurid ühte süsteemi, mis võib olla juhitud ühest kohast ja universaalsete käskudega**.**

Samuti, eesmärk oli leida lahendus selleks, et targa maja seadistamine ja automatiseerimise protsess toimuks mugavalt kasutajatele ning sobilik ka inimestele, kes ei oska programmeerida.

Kirjeldati ära OpenHAB süsteemi, struktuuri ning reeglite ja skriptide kirjutamise loogikat mis võimaldab ühendada süsteemi erinevaid seadmeid ja andureid. Samuti, käesolevas töös oli mainitud ka OpenHABiks tehtud programmeerimiskeskkond OpenHAB Designer kus saab reegleid kirjutada ja kohe kontrollida.

Lõputöös püstitatud eesmärgid on saavutanud täies ulatuses, mis on tehtud sügava analüüsiga, täiendavate materjalidega ning oma arvamusega.

6 Summary

This bachelor's thesis goal was to find out a solution in order to connect the various components and sensors into one system which can be controlled from one place and by universal orders.

Also, the goal was to find out solution for smart home automation user-friendly and convenient interface even for those people, who do not know how to write a code.

In this thesis OpenHAB system was described, the structure and logic of rules and scripts writing , which allows the system to connect variety of devices and sensors. Also, in this work was mentioned OpenHAB Designer, where users can write scripts and rules and check them immediately.

In the thesis objectives were fully achieved, which was made by analysis, by supplementary materials and own opinion.

Kasutatud kirjandus

1. SMARTech „How Much Does a Smart Home System Cost?“ [WWW] http://www.smarthome.eu/how-much-does-a-smart-home-system-cost [02.04.2016]

2. Joel Lee „How Much Does a Smart Home Really Cost?“ [WWW] http://www.makeuseof.com/tag/much-smart-home-really-cost/ [02.04.2016]

3. IControl Networks „State of the Smart Home Report.“ [WWW] https://www.icontrol.com/wp-content/uploads/2015/06/Smart_Home_Report_2015.pdf [10.04.2016]

4. heaterC „Overview of OpenHAB“ [WWW] https://github.com/openhab/openhab/wiki [15.04.2016]

5. Oracle Press Release „Oracle Announces Winners of the 2013 Duke’s Choice Awards“ [WWW] http://www.oracle.com/us/corporate/press/2020453 [15.04.2016]

6. Eclipse SmartHome „Eclipse SmartHome Documentation Overview“ [WWW] http://www.eclipse.org/smarthome/documentation/index.html [22.04.2016]

7. „OSGI Specification Process“ [WWW] https://en.wikipedia.org/wiki/OSGi [22.04.2016]

8. „OpenHAB Unterstützte Technologien“ [WWW] https://de.wikipedia.org/wiki/OpenHAB [27.04.2016]

9. Klaus Hildner „How to Use Scripts in OpenHAB“ [WWW] https://github.com/openhab/openhab/wiki/Scripts [27.04.2016]

10. Raoul-Gabriel Urma „Alternative Languages for the JVM“ [WWW] http://www.oracle.com/technetwork/articles/java/architect-languages-2266279.html [30.04.2016]

11. Sam Turner „How to work with automation rules“ [WWW] https://github.com/openhab/openhab/wiki/Rules [30.04.2016]

12. Marcela De Vivo „Essentials For Creating a User-Friendly Interface“ [WWW] http://www.instantshift.com/2013/08/14/creating-user-friendly-interface/ [30.04.2016]

13. Jack Wallen „10 Things that Make Software User-Friendly“ [WWW] http://www.techrepublic.com/blog/10-things/10-things-that-make-software-user-friendly/ [30.04.2016]

14. Jayson Falkner and Kevin W.Jones „What Servlets Are and Why You Would Want to Use Them“[WWW] http://www.informit.com/articles/article.aspx?p=170963 [30.04.2016]

15. Java Virtual Machine[WWW] https://en.wikipedia.org/wiki/Java_virtual_machine [02.05.2016]

16. „How does Z-Wave work?“[WWW] www.z-wave.com/faq/How_does_ZWave_work [01.05.2016]

17. „Misson and Objectivies“[WWW] https://www.knx.org/knx-en/knx/association/mission-objectives/index.php [30.04.2016]

18. „Bluetooth technology basis“[WWW] https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics [22.04.2016]

19. „Benefits of using OSGI technology“[WWW] https://www.osgi.org/developer/benefits-of-using-osgi/[15.04.2016]

20. „Integrated Development Environment“[WWW] https://en.wikipedia.org/wiki/Integrated_development_environment[26.04.2016]

21. „Java Servlet Technology“[WWW] https://docs.oracle.com/javaee/6/tutorial/doc/bnafd.html[15.04.2016]

22. Daniel Cooper „Philips Hue:the world’s smartest LED lightbulb that saves your time during Red Alert“[WWW] http://www.engadget.com/2012/10/29/philips-hue/[15.04.2016]

23. „Insteon:The Details“[WWW]
http://cache.insteon.com/documentation/insteon_details.pdf[18.04.2016]

24. „Model-View-Controller“[WWW]
https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller
[15.04.2016]

25. „ZigBee“[WWW]
http://www.vallaste.ee/index.asp [15.04.2016]