

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika  
õppekava

**Lenard Piir**  
**AOM4STS graafilise tööriista analüüs  
ja arendus**  
**Bakalaureusetöö (9 EAP)**

Juhendaja(d): Kuldar Taveter

# **AOM4STS graafilise tööriista analüüs ja arendus**

## **Lühikokkuvõte:**

Bakalaureusetöö eesmärgiks on analüüsida ja arendada AOM4STS rakendust. Töös tutvustatakse ja analüüsitakse vana AOM4STS rakendust. Lisaks luuakse ülevaade uuest AOM4STS rakendusest ning tehnoloogiateks, mida kasutatakse rakenduse arendamiseks.

## **Võtmesõnad:**

Analüüs, Javascript, React, veebirakendus

**CERCS:** P175 informaatika

## **Title in English**

### **Abstract:**

The aim of this Bachelor's thesis is to analyse and develop AOM4STS software tool. This Bachelor's thesis will give an overview of old AOM4STS software tool including systems functional requirements. There will also be overview, requirements and technical analysis for creatable system. This Bachelor's thesis will also include software development for new AOM4STS system.

### **Keywords:**

Analysis, Javascript, React, web application

**CERCS:** P175 Informatics

## Sisukord

1.	Sissejuhatus	4
2.	Vana AOM4STS rakenduse tutvustus	6
2.1.	Eesmärgimudel	6
2.2.	Rollimudel	7
2.3.	Organisatsioonimudel	9
2.4.	Valdkonna teadmiste mudel	10
3.	Vana AOM4STS funktsionaalsed nõuded	11
3.1.	Eesmärgi moodul	11
3.2.	Rollimudeli moodul	12
3.3.	Organisatsioonimudeli moodul	12
3.4.	Valdkonna teadmiste mudel	12
3.5.	Tagaprogrammi moodul	13
4.	Funktsionaalsed nõuded, mis lisanduvad uue projekti raames	14
5.	Seotud lahendused	15
6.	Kasutatud tehnoloogiad	16
7.	Arendusprotsessi kirjeldus	18
8.1.	Arendusprotsessi esimene etapp	18
8.2.	Arendusprotsessi teine etapp	21
8.3.	Arendusprotsessi kolmas etapp	22
8.3.1.	Tagaprogrammi süsteemianalüüs	22
8.	Kokkuvõte	23
9.	Viidatud kirjandus	24
I.	Litsents	25
	Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks	25

## 1. Sissejuhatus

AOM4STS on graafiline tööriist, mis on loodud sotsiotehniliste süsteemide agentorienteeritud modelleerimise jaoks. Termin sotsiotehniline süsteem on vahel kasutatud selleks, et viidata süsteemidele mis sisaldavad nii sotsiaalset aspekti (mis võib olla ka alamsüsteem) kui ka tehnilist aspekti. Peamiselt kasutatakse sotsiotehniliste süsteemide puhul terminit mitme agendi süsteem, sest tahetakse rõhutada huvi agentorienteeritud mudeli vastu [1]. Terminis agentorienteeritud on agent defineeritud üksusena, mis tunnetab ja mõjutab keskkonda ning rakendab loogikat.

AOM4STS kasutatakse graafiliste mudelite loomiseks ja uute mudelite genereerimiseks. Valdonna ekspertidel on vaja agentorienteeritud modelleerimiseks töökeskkonda, millega saaks luua graafilisi mudeleid ning genereerida uusi mudeleid mudelitest mis on juba olemas [2]. Töökeskkond peaks ka genereerima sotsiotehnilist süsteemi koodi. Lisaks valdkonna ekspertidele saavad süsteemi kasutada ka tudengid agentorienteeritud modelleerimise õppimiseks.

Bakalaureusetööl on mitu eesmärki. Esimene eesmärk on analüüsida AOM4STS tööriista ja sellega seostuvaid lahendusi. Analüüsis luuakse ülevaade graafilise tööriista AOM4STS funktsionaalsustest, mis on tööriistal olemas ja funktsionaalsustest mida on tööriistale juurde vaja. Lisaks kirjeldatakse tehnoloogiaid, mis toetaks uue graafilise tööriista loomise arendusprotsessi. Teine eesmärk on alustada uue AOM4STS tööriista arendusega. Vanale AOM4STS rakendusele on vaja luua uusi funktsionaalsusi, kuid vana tööriista edasi arendamine ei ole enam mõistlik. Põhjus miks luuakse uus projekt seisneb selles, et vana süsteem ei ole kooskõlas modernsete tarkvaraarendus standarditega. See tähendab, et vana süsteemi arendus ei ole enam jätkusuutlik ja vaja on luua uus süsteem. Uue projekti süsteemi arhitektuur, kasutatavad teegid ning raamistikud saavad olema kooskõlas modernsete tarkvaraarendus standarditega. Loodaval süsteemil saavad olema funktsionaalsused, mis on vanal süsteemil ning ka uued funktsionaalsused. Nii vanad kui ka uued funktsionaalsused on kirjeldatud järgnevates peatükkides.

Bakalaureusetöö on jaotatud kümneks peatükiks. Teises peatükis on vana AOM4STS rakenduse tutvustus. Kolmandas peatükis on kirjeldatud vana AOM4STS funktsionaalsed nõuded. Neljandas peatükis on kirjeldatud funktsionaalsused, mis puuduvad AOM4STS tööriistal ning

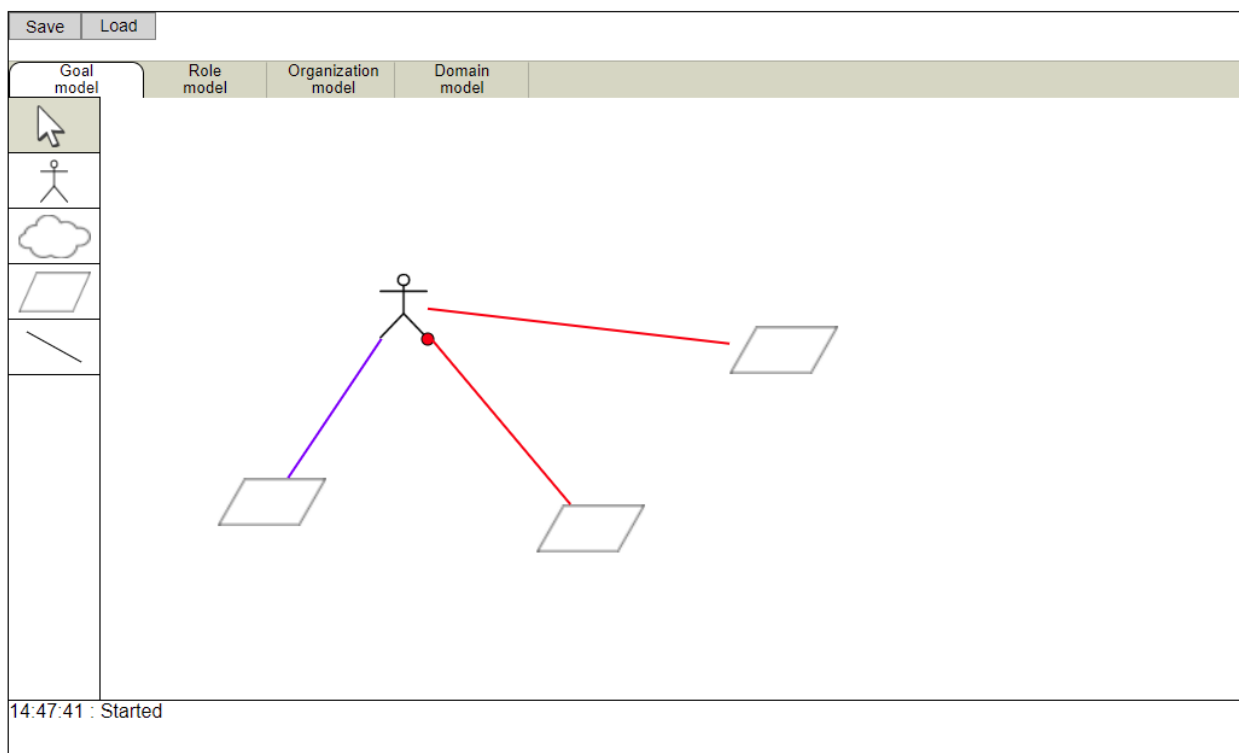
mis luuakse lisaks vana AOM4STS funktsionaalsusele uue projekti raames. Viiendas peatükis on toodud välja seostuvad lahendused. Kuuendas peatükis on analüüs uues projektis kasutatavate tehnoloogiate kohta. Seitsmendas peatükis on kirjeldatud arendusprotsessi. Kaheksandas peatükis on tehtud töö analüüs. Üheksandas peatükis on tehtud bakalaureusetöö kokkuvõte.

## 2. Vana AOM4STS rakenduse tutvustus

AOM4STS on loodud Aleksandr Sapožnikov ja Msury Rogasian Mahunnah poolt. Rakenduses on neli erinevat mudelit. Nendeks on eesmärgimudel (*Goal model*), rollimudel (*Role model*), organisatsioonimudel (*Organization model*) ja valdkonna teadmiste mudel (*Domain model*) [2]. Nii nagu ka Aleksandr Sapožnikov poolt 2015 kaitsitud magistritöös “Agentorienteeritud modelleerimise töövahendi graafilise kasutajaliidese kavandamine ja realisatsioon” mainitud, on komponendid, mudelid ja reeglid võetud 2012 aastal kaitsitud Andrei Rõbinski magistritööst “Valdkonnaontoloogia agentorienteeritud modelleerimise tööriista jaoks” [2] [6].

### 2.1. Eesmärgimudel

Eesmärgimudel kirjeldab loodava süsteemi eesmärki [2]. Antud mudelis kuvatakse seoseid eesmärkide ja rollide vahel ning ka eesmärkide omavahelisi seoseid [2] [6]. Joonisel 1 on näha eesmärgimudeli vaade. Eesmärgimudeli vaates on loodud roll, mis on seotud kolme funktsionaalse eesmärgiga.



Joonis 1. Eesmärgimudel, kus on kujutatud roll kolme seotud funktsionaalse eesmärgiga

## 2.2. Rollimudel

Rollimudelis kuvatakse kõiki rolle ning nende detailsed kirjeldusi. Rollimudeli ja eesmärgimudeli vahel tehtud muudatused kajastuvad mõlemas mudelis. See tähendab, et kui eesmärgimudelis luuakse eesmärk siis see kajastub ka rollimudelis ja vastupidi. Joonisel 2 on näha rollimudel ja joonisel 3 eesmärgimudel, kus on need samad rollid näha.

Save Load

Goal model Role model Organization model Domain model

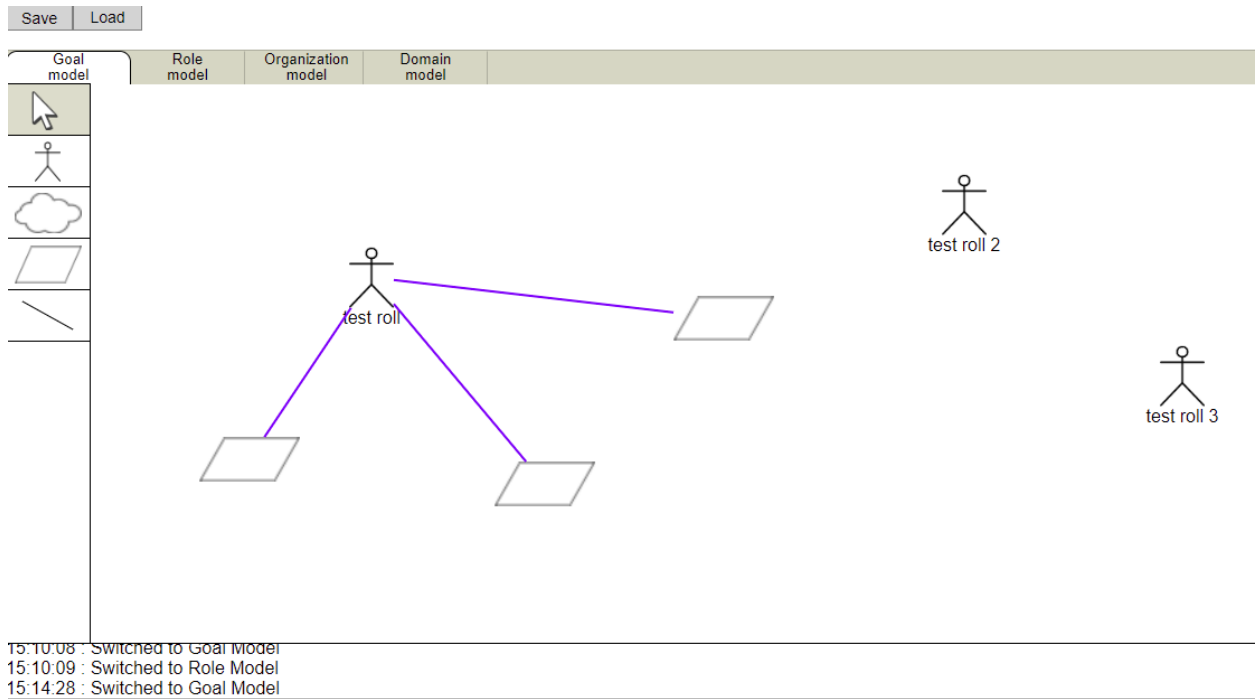
Create new  
test roll  
test roll ...  
test roll ...

Edit Delete

Role name	test roll 3
Description	
Responsibilities	
Constraints	

15:15:39 : Switched to Goal model  
15:15:58 : Switched to Role Model  
15:16:01 : Role "" deleted from RoleModel

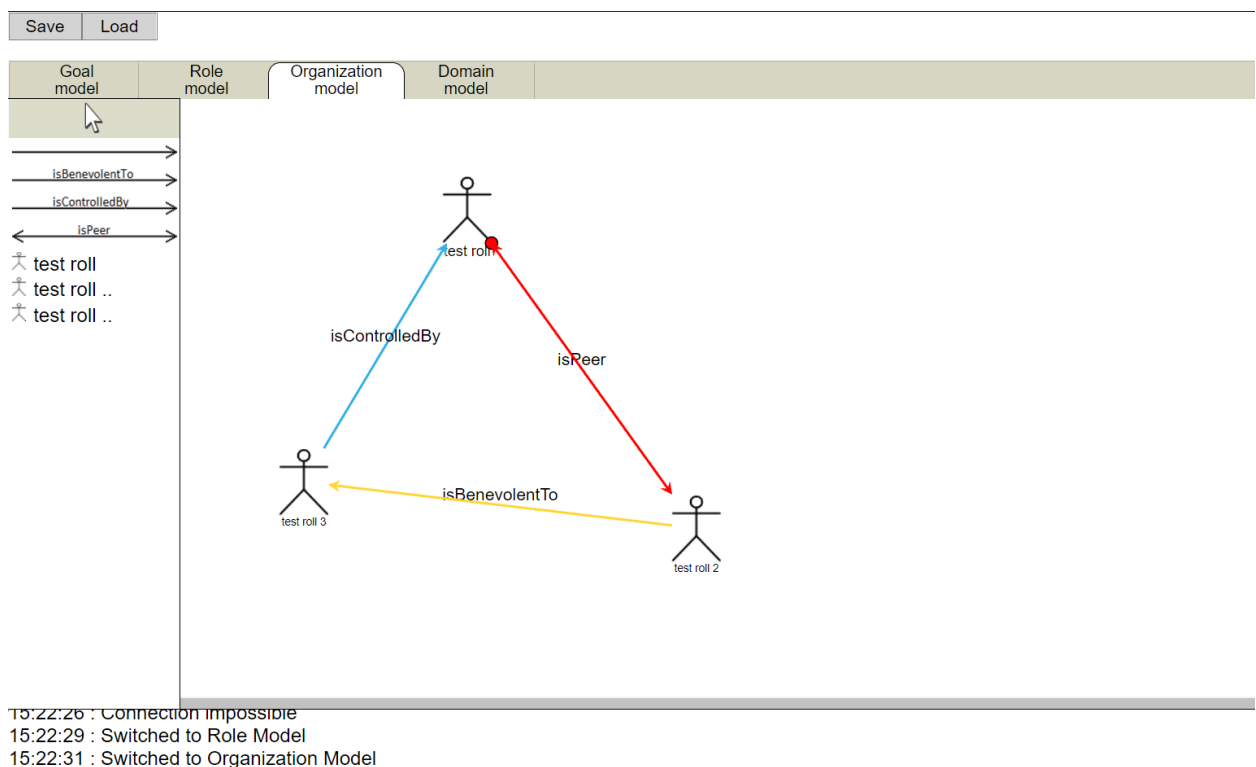
Joonis 2. Rollimudel



Joonis 3. Eesmärgimudel koos rollimudelis loodud rollidega

## 2.3. Organisatsioonimudel

Organisatsioonimudel on kujutatud rollide vahelised seosed. Rollide vahel on kolm erinevat seost ning nendeks on kontroll (*isControlledBy*), heatahtlikus (*isBenevolent*) ja võrdsus (*isPeer*). Kontroll seoses üks roll juhib teist, heatahtlikus seoses üks roll pakub teisele teenust ja teine roll võib selle teenuse vastu võtta või sellest keelduda ja võrdsus seoses on rollid võrdsed [2] [6]. Joonisel 4 on kujutatud organisatsioonimodeli vaade eelnevalt loodud rollidega.

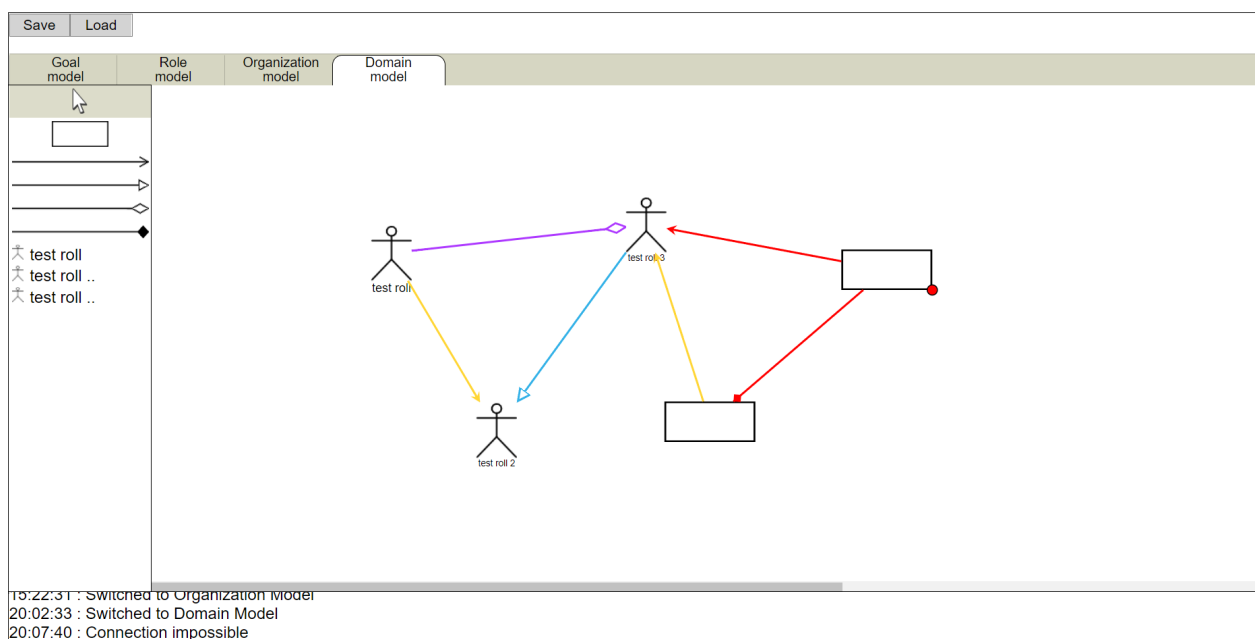


Joonis 4. Organisatsioonimudel koos seostega rollide vahel

## 2.4. Valdkonna teadmiste mudel

Valdkonna teadmiste mudel kirjeldab süsteemis teadmisi, mis tuleb valdkonna kohta esitada. Selles mudelis kasutatakse rolle ning rolle ei saa selles mudelis luua ega kustutada. Valdkonna teadmiste mudelis saab kirjeldada valdkonnaolemeid, mis on seotud omavahel ning mis on ka seotud rollidega. [2] [6]

Joonisel 5 on näha eelnevalt loodud rollid, nende vahelised seosed, valdkonna teadmiste üksused ning nende omavahelised seosed ja seosed rollidega.



Joonis 5. Valdkonna teadmiste mudel rollide ja valdkonna teadmiste üksustega

### 3. Vana AOM4STS funktsionaalsed nõuded

Vana rakenduse funktsionaalsete nõuete loomise eesmärk on toetada uue AOM4STS tarkvaraarendust. Nii nagu ka eelnevalt mainitud peab uus AOM4STS sisaldama ka vana AOM4STS funktsionaalsust. Rakenduse funktsionaalsed nõuded on jaotatud moodulite vahel. Rakendusel on viis moodulit. Järgnevad funktsionaalsed nõuded on loodud autori enda poolt.

#### 3.1. Eesmärgi moodul

Eesmärgi mooduli funktsionaalsed nõuded:

- Eesmärgi moodulis peab süsteem võimaldama kasutajal rollimudeli objekti lisamist.
- Kasutaja peab saama lisada funktsionaalse eesmärgi objekte.
- Kasutaja peab saama lisada kvaliteedieesmärgi objekte.
- Kui kasutaja on lisanud rolli, funktsionaalse eesmärgi või kvaliteedieesmärgi peab ta saama luua seoseid nende vahel.
- Kui kasutaja on lisanud rolli, funktsionaalse eesmärgi või kvaliteedieesmärgi peab ta saama neid objekte liigutada tööpinnal.
- Rolli objekti nimi, kirjeldus, vastutus ja piirangud peavad olema muudetavad kasutaja poolt.
- Rolli, funktsionaalse eesmärgi ja kvaliteedieesmärgi objekte peab olema võimalik kasutaja poolt kustutada.
- Süsteem peab võimaldama kasutajal rolli objektile muutmise vaates lisada funktsionaalne eesmärk, millel on seos selle rolli objektiga.
- Funktsionaalse eesmärgi objekti nime ja kirjeldus peavad olema kasutaja poolt muudetavad.
- Süsteem peab võimaldama kasutajal rolli ja funktsionaalse eesmärgi muutmise vaates lisada seostuvat positiivset või negatiivset emotsiooni.
- Kasutaja peab saama lisada funktsionaalse eesmärgi külge seostuvat rolli, kvaliteedieesmärki või funktsionaalset eesmärgi.
- Kasutajal peab olema võimalik kõikide muutmise vaadete vorme lukustada.
- Kvaliteedieesmärgi nimi ja kirjeldus peavad olema muudetavad kasutaja poolt.
- Süsteem peab võimaldama kasutajal lisada kvaliteedieesmärgi muutmise vaates seostuvat funktsionaalse eesmärgi objekti.
- Kasutaja peab saama seoste nimesid muuta.

- Kasutaja peab saama seoseid kustutada.

### **3.2. Rollimudeli moodul**

Rollimudeli mooduli funktsionaalsed nõuded:

- Rollimudeli moodulis peavad kõik rolli objektid olema kasutajale kuvatud.
- Kasutaja peab saama uut rolli lisada.
- Süsteem peab võimaldama kasutajal rolli objekti nime, kirjelduse, kohustuste ja piirangute lisamist ning muutmist.
- Kasutaja peab saama rolle lisada.

### **3.3. Organisatsioonimudeli moodul**

Organisatsioonimudeli mooduli funktsionaalsed nõuded:

- Organisatsioonimudeli moodulis peab kasutaja saama luua erinevaid seoseid rolli objektide vahel (tava, heatahtlikus, kontroll, võrdsus).
- Seosed ja rolli objektid peavad olema kustutatavad kasutaja poolt.
- Kasutaja peab saama vaadata rolli nime, kirjeldust, kohustusi ja piiranguid.
- Kõik rolli objektid peavad olema kasutajale kuvatud.

### **3.4. Valdkonna teadmiste mudel**

Valdkonna teadmiste mudeli mooduli funktsionaalsed nõuded:

- Valdkonna teadmiste mudeli moodulis peab kasutaja saama lisada valdkonna teadmiste objekti
- Valdkonna teadmiste objektid peavad olema kasutaja poolt kustutatavad.
- Kasutajal peab olema võimalik valdkonna teadmiste objekti nime ja kirjelduse muutmine.
- Süsteem peab kuvama kasutajale kõiki rolli objekte.
- Kasutaja peab saama luua erinevaid seoseid rolli objektide vahel ning samuti valdkonna teadmiste objektide vahel.

### 3.5. Tagaprogrammi moodul

Import / export moodul:

- Süsteem peab võimaldama kasutajal salvestada enda töö ning exportida see kas XML või JSON failina.
- Süsteem peab võimaldama XML või JSON failide sisselugemist, mis on genereeritud läbi AOM4STS süsteemi.

Kõik kasutaja poolt tehtud liigutused peavad olema süsteemi poolt logitud (välja arvatud objektide liigutamine tööpinnal).

Süsteemil peab olema valideerimine selleks, et kasutaja saaks teha liigutusi ainult vastavalt ettenähtud loogikale.

#### 4. Funktsionaalsed nõuded, mis lisanduvad uue projekti raames

Lisanduvad funktsionaalsed nõuded on koostatud Aleksandr Sapožnikov poolt 2015 kaitsitud magistritöö põhjal [2]. Antud töös on kirjeldatud detailset millised funktsionaalsused peaksid olema ühel agentorienteeritud modelleerimis tööriistal.

Projektile lisanduvad uued moodulid:

1. Agendimudel (*Agent model*)
2. Agentide tutvuste mudel (*Acquaintance model*)
3. Agentide teadmiste mudel (*Knowledge model*)
4. Agentide suhtlusmudel (*Interaction model*)
5. Agentide käitumismudel (*Behaviour model*)

Uued moodulid on oma olemuselt samasugused nagu vanas süsteemis olevad moodulid. See tähendab, et kõikidel moodulitel on oma vaade ning kõikide moodulite objekte on võimalik tööpinnasele tekitada ja neid tööpinnasel muuta. Samuti on võimalik nende objektide vahel seoseid luua. Lisaks on võimalik nii objekte kui ka seoseid kustutada. Nii nagu ka vanas süsteemis, on võimalik tekitada vastavaid mooduli objekte teistes moodulites. Moodulite vahelised seosed on kirjeldatud Aleksandr Sapožnikov poolt järgnevalt [2]:

- Rollimudelis peab saama kasutada eesmärgimudelis loodud rolle ja vastupidi.
- Organisatsioonimudelis peab saama kasutada eesmärgimudelis ja/või rollimudelis loodud rolle.
- Valdkonna teadmiste mudelis peab saama kasutada eesmärgimudelis ja/või rollimudelis loodud rolle.
- Agendimudelis peab agentide loomisel saama kasutada eesmärgimudelis ja/või rollimudelis loodud rolle. Ühte agenti peab olema võimalik siduda mitme rolliga ja vastupidi.
- Agentide tutvuste mudelis peab saama kasutada agendimudelis loodud agente ja eesmärgimudelis ja/või rollimudelis loodud rolle.
- Agentide teadmiste mudelis peab saama kasutada agendimudelis loodud agente ja valdkonna teadmiste mudelis loodud valdkonnaolemeid.
- Agentide suhtlusmudelis peab saama kasutada agendimudelis loodud agente.
- Agentide käitumismudelis peab saama kasutada agendimudelis loodud agente.

## 5. Seotud lahendused

Seotud lahenduste puhul on valitud graafilised modelleerimis tööriistad.

Üks seotud lahendustest on AToMPM (A Tool for Multi-Paradigm Modelling). AToMPM saab liigitada samuti graafiliseks tööriistaks nagu on AOM4STS. Kuigi on tegemist erinevate tööriistadega, on nendes kahes tööriistas teatud sarnasusi.

Kõigepealt proovisin seda rakendust läbi nende õpetuse dockeriga tööle saada, kuid see ei õnnestunud. Midagi on neil kas juhendis uuendamata jäänud või Docker fail ei ole kooskõlas praeguse süsteemi seisuga. Kuna Docker variant ei töötanud rakenduse püsti saamisel, siis tuli läbida päris mitmeid samme selleks, et rakendus tööle saada.

AToMPM on võrdlemisi suure õppimiskõveraga rakendus. Kui seda võrrelda AOM4STS rakendusega, siis AToMPM on märkimisväärselt mahukam ja keerulisem süsteem. Kuna selle töö käigus on analüüsi poolel üks olulisemaid punkte leida sarnaseid lahendusi või tööriistu mille abil arendada uus AOM4STS, siis antud lahendust me selleks kasutada ei saa. Põhjus seisneb selles, et AOM4STS ja AToMPM võivad küll olla mõlemad graafilised modelleerimis tööriistad, kuid meil on vaja tööriista mis oleks konkreetselt agentorienteeritud modelleerimise jaoks. AToMPM sihtrühm on rohkem abstraktsem modelleerimine.

Analüüsisides AToMPM koodi on ka näha, et tegemist on samuti vanema süsteemiga. Kuna meie algne probleem on see, et AOM4STS on rakenduse mõistes vanaks jäänud, siis kasutades mõnda teist vana süsteemi ei lahendaks see meie probleemi. Samuti ei annaks amortiseerunud lahenduse puhul kasutada nende koodis olevaid funktsionaalsusi. AToMPM on ehitatud Pythoni põhjal, kuid rääkides tänapäeva tarkvaraarendus standarditega kaasas käivatest lahendustest võiks selline tööriist olla loodud mõne modernse Javascripti raamistiku peale.

## 6. Kasutatud tehnoloogiad

AOM4STS puhul on suures osas tegemist Javascripti projektiga. Uue AOM4STS arenduseks otsustasin valida Javascripti teegi React [5]. React on praegusel ajal üks parimatest Javascripti tehnoloogiatest. Samuti on React üks kõige populaarsematest Javascripti tehnoloogiatest praegusel ajal.

Projekti loomiseks võtsin kasutusele tehnoloogia nagu NPM [8]. NPM on vajalik ka teiste vabavaraliste teekide lisamiseks projekti.

Javascripti jooksutamiseks serveri poolel võtsin kasutusele Node [7]. Node teeb Javascripti arenduse märkimisväärselt kiiremaks. Kasutades Node saan ma enda arvutis serveri poolel Javascripti jooksutada.

Kuna tegemist on React-i projektiga otsustasin kasutusele võtta ka sellise tehnoloogia nagu Redux [9]. Redux võimaldab globaalselt olekute salvestamist. See on vajalik komponentide vahel andmete vahetamiseks. Kuna React on oma olemuselt olekute põhine, siis on tegemist väga kasuliku teegiga. Alternatiivselt saab kasutada andmete edastamist *Parent-Child* (edaspidi ülem ja alluv) loogikal, aga see töötab ainult siis kui üks komponent kutsub teist välja. Kui üks komponent ei kutsu teist välja, ei ole võimalik ilma Redux-i poe loogikata komponentide vahel andmeid vahetada *frontend* (edaspidi eessüsteem [10]) poolel. Kuna me näeme juba ette ära, et projektis saab olema vähemalt 9 eessüsteemi komponenti, siis ei oleks neid kõiki teha ülem ja alluv loogikal mõistlik. Antud juhul kõige mõistlikum on teha globaalne olekute hoidla.

Analüüsiga alustades leidsin tööriista nimega React-Raphael. Raphael on teek, mida on kasutatud AOM4STS algse tööriista arendamiseks. React-Raphael teek on loodud üle 5 aasta tagasi. Sellel ajal kasutati märkimisväärselt vanemaid Reacti versioone kui praegusel hetkel. Näited mis on toodud välja React-Raphaeli kohta ning ka näide nende kodulehel selle teegiga alustamise kohta on samuti loodud üle 5 aasta tagasi. Kuna jääb mulje, et React-Raphaeli peaks implementeerima vanema Reacti versiooniga, siis võtsin ühendust selle teegi loojaga. Bakalaureusetöö teostamise jooksul kahjuks React-Raphael teegi looja ei vastanud mulle. Kuna töö eesmärk on tarkvara arendusprotsessis kasutada modernseid tehnoloogiaid ja arendusstandardeid, siis React-Raphael teeki selle töö käigus ei kasutata. Põhjus seisneb selles, et React-Raphael teek ei tööta uuemate Reacti versioonidega.

Erinevate objektide liigutamise jaoks otsustasin kasutusele võtta sellise teegi nagu React DnD [3]. React DnD on teek, mis võimaldab objektide lohistamise ja paigutamise ühest kohast teise. Objektide lohistamine ja paigutamine on funktsionaalsus, mida suurem osa projekti moodulitest kasutab. Valisin React DnD sellepärast, et tegemist on teegiga mis on väga laialdaselt kasutusel ja mida pidevalt uuendatakse. Samuti olen ma enda tarkvaraarendaja karjääri jooksul seda teeki ühes erasektori projektis kasutanud.

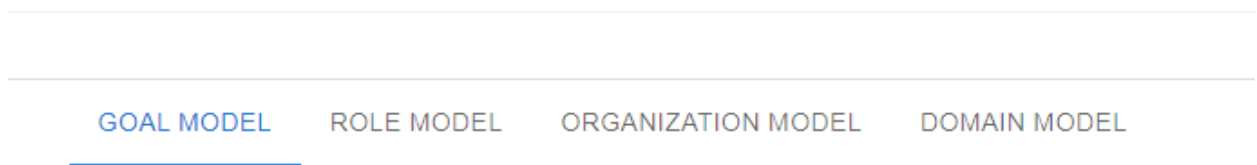
Kujunduse jaoks võtsin kasutusele teegi nimega MUI [9]. MUI annab väga suure mahus erinevaid React-i komponente, mida annab kasutada kasutajaliidese loomisel.

## 7. Arendusprotsessi kirjeldus

Eessüsteemi arenduse eesmärgiks on luua samad funktsionaalsused, mis on vanal AOM4STS süsteemil. Eesmärgi täide viimiseks kasutatakse eelnevalt kirjeldatud teeke. Lisaks võetakse eesmärk kasutada uut eessüsteemi vana rakenduse *backend* (edaspidi tagaprogramm [10]) funktsionaalsusega.

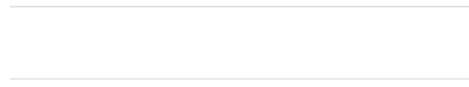
### 8.1. Arendusprotsessi esimene etapp

Arendusprotsessiga alustati uue React-i projekti genereerimisega. Arendusprotsessi esimene eesmärk on luua uus kasutajaliides. Kasutajaliidese puhul kasutasin sarnast loogikat, mida oli kasutatud vanas kasutajaliideses. See tähendab, et horisontaalses menüüs on võimalik valida mudelite vaateid. Joonisel 6 on kujutatud horisontaalne menüü.



Joonis 6. Horisontaalne menüü

Vertikaalses menüüs annab antud mudeli tegevuste vahel valida. Joonisel 6 on kujutatud vertikaalne menüü.



Role



Quality goal



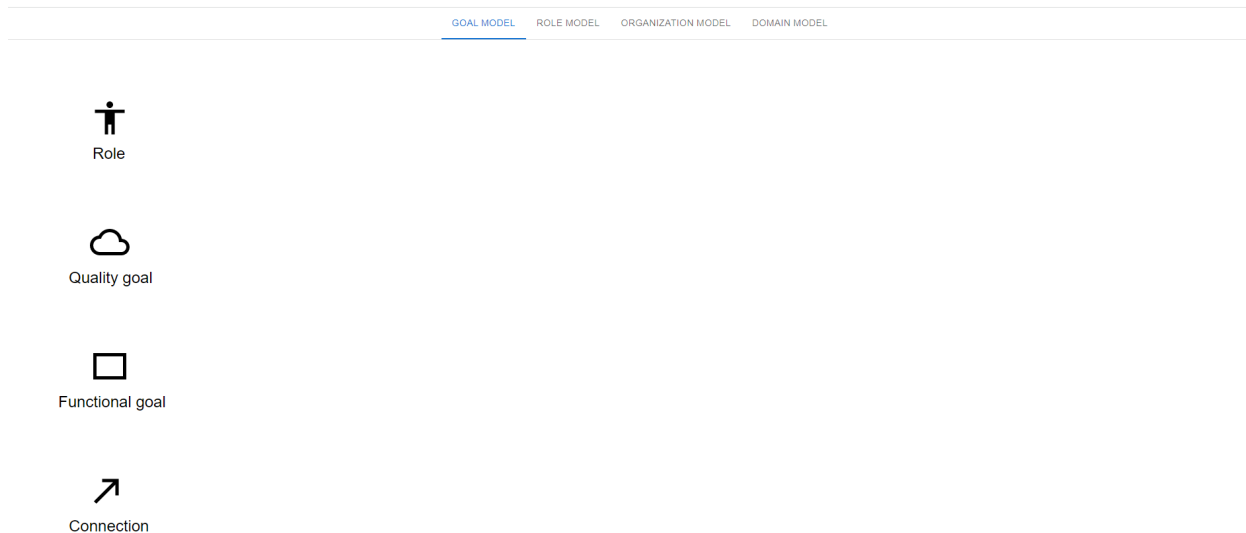
Functional goal



Connection

Joonis 7. AOM4STS vertikaalne menüü

Uue kasutajaliidese menüüde loomiseks kasutasin MUI [9] komponente. Joonisel 8 on kujutatud uus kasutajaliides täies mahus.



Joonis 8: Uus AOM4STS kasutajaliides eesmärgimudeli vaates

Võrreldes vana kasutajaliidesega on mudelite valikud tõstetud rohkem ekraani keskele. Samuti on tegevused, mida antud vaates teha annab paremini esile tõstetud.

## 8.2. Arendusprotsessi teine etapp

Arendusprotsessi teine eesmärk on luua funktsionaalsus, mis võimaldab vertikaalses menüüs olevaid objekte lohistada tööpinnasele ja luua seoseid objektide vahel. Kui objekt on lohistatud tööpinnasele, peab seda olema võimalik kasutada teistes vaadetes vastavalt projekti funktsionaalsetele nõuetele. Lisaks ei tohi objektide olekud muutuda, kui tehakse teisi tegevusi. Näiteks ei tohi ükski objekt kuhugi ära kaduda tööpinnaselt, kui vahetatakse vaadet. See tähendab, et iga tööpinnasel tehtud liigutus peab olema püsiv olenemata vaatest. Lisaks kui tehakse muudatusi ühes vaates, peavad muudatused kajastuma ka teistes vaadetes.

Objektide lohistamiseks tööpinnasele võtsin ma kasutusele React DnD [3]. React DnD võimaldab objekti lohistamist koos andmetega tööpinnasele. Koos andmetega tähendab, et kui meil on huvi panna kaasa lohistamise alguses andmeid, siis React DnD võimaldab seda. Arenduse käigus avastati, et vertikaalse menüü kõige viimane valik, milleks on seose loomine ei saa enam samal loogikal töötada. Algselt planeeritud versioon, kus seoseid saaks lohistada objektide vahel, on liiga keeruline et teostada. Põhjus miks seda on liiga keeruline teostada seisneb selles, et antud juhul on lohistamise alustamiseks andmete kätte saamine keeruline. Lohistamine toimub puhtalt andmete liigutamise põhimõttel ühest kohast teise. Kui meil on töölaud, kus objektid võivad olla mis iganes kohas, muutub andmete kätte saamine väga keeruliseks. Seose loomine võiks töötada nii nagu vanas süsteemis. See tähendab, et kui me valime seose loomise valiku siis me vajutame kahe objekti peal tööpinnasel. Sama probleem tekkis ka tööpinnasel olevate objektide liigutamisega. Funktsionaalsus mis seostub tööpinnasele objekti lohistamisega erineb väga funktsionaalsusest, mis seostub tööpinnase enda peal objektide lohistamisega.

### **8.3. Arendusprotsessi kolmas etapp**

Arenduse kolmas etapp keskendub andmete saatmisega tagaprogrammi. Kuna projekti käigus leiti, et uue tagaprogrammi tegemine on liiga mahukas töö ühe bakalaureusetöö raames, siis proovitakse kasutada vana AOM4STS rakenduse tagaprogrammi funktsionaalsust. Selleks tuleb analüüsida vana rakenduse tagaprogrammi.

#### **8.3.1. Tagaprogrammi süsteemianalüüs**

Vana rakenduse tagaprogramm on ehitatud programmeerimiskeelega PHP. Rakenduse tagaprogrammi eesmärgiks on genereerida sotsiotehnilise süsteemi koodi. See tähendab, et rakenduses on realiseeritud lugeja ja kirjutaja kasutades programmeerimiskeelt PHP. See võimaldab andmete sisselugemise XML kujul ning samuti andmete väljastamise XML kujul. Andmete väljastamise ja sisselugemise puhul genereeritakse fail, milles on terve süsteemi andmete olek väljastamise või sisselugemise hetkel. Kasutaja vaatepunktist võimaldab selline funktsionaalsus mudeleid salvestada ja olemas olevaid mudeleid sisse lugeda.

Vana AOM4STS rakenduse tagaprogrammi süsteemianalüüsi puhul peame me lähtuma tervest süsteemist. Rakendus käivitatakse ühe tervikuna, see tähendab et tegemist on monoliit rakendusega. Antud monoliit rakenduse puhul ei ole eristatud teenuseid, milleks võiksid olla tagaprogramm ja eessüsteem. Monoliit rakenduse puhul, millel puuduvad suhtlemisvõimalused teiste rakendustega, on antud kujul võimatu luua suhtlus meie eraldiseisva eessüsteemiga. Selleks, et rakendused saaksid omavahel suhelda, peaksid olema rakendusel näiteks REST [11] lõpp-punktid. See tähendab, et rakendusel peab olema kiht mis võtab vastu HTTP päringuid (näiteks GET ja POST). Vana AOM4STS rakenduse arhitektuuri tõttu ei ole ka mõistlik ehitada vanale rakendusele kihti, mis võtaks vastu HTTP päringuid. Kuna vana rakenduse funktsionaalsus on omavahel niivõrd põimitud, ei oleks võimalik seal eraldada osasid, mis tegelevad tagaprogrammi loogikaga. Samuti ei ole seal raamistiku, mis võimaldaks meil luua lõpp-punkte HTTP päringute vastuvõtmise jaoks.

## 8. Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli analüüsida ja edasi arendada AOM4STS tööriista. Tulemusena tekkis ülevaade vanast AOM4STS rakendusest nii tehnilise kui ka mitte tehnilise poole pealt. Samuti tekkis nii tehniline kui ka mitte tehniline ülevaade uue rakenduse osas.

Bakalaureusetöös tehti sissejuhatus sotsiotehniliste süsteemide ja agentorienteeritud modelleerimise maailma. Koostati ülevaade ja analüüs vanast AOM4STS rakendusest. Samuti koostati ülevaade uutest loodavatest funktsionaalsustest. Bakalaureusetöös koostati ülevaade tehnoloogiatest millega võiks uut AOM4STS arendada. Töös analüüsiti milliseid tehnoloogiaid tuleks tänapäevases tarkvaraarendus maailmas kasutada antud projekti jaoks. Lisaks alustati uue AOM4STS tööriista arendust. Arenduse käigus tehti süsteemianalüüs võimalike tehniliste probleemide ennetamiseks. Töös toodi välja arenduse tehnilised probleemid nii teekide kui ka arhitektuuri poolelt.

Läbiviidud süsteemianalüüsi ja arenduse põhjal leiab autor, et uue AOM4STS arenduse jaoks on vaja graafilist teeki, mis on kooskõlas modernsete arendusstandarditega. Kasutades autori poolt valitud teeki, muutub selle projekti eessüsteemi arendus väga mahukaks. Lisaks väga suurele mahukusele oleks arendus suures osas funktsionaalsuses loomine, mida Raphael teek juba pakub. Kuna vana projekti tagaprogrammi kasutamine ei olnud võimalik, siis tuleks luua uus tagaprogramm AOM4STS rakendusele. Uus tagaprogramm peab olema eraldiseisev süsteem ja võimaldama suhtlust teiste süsteemidega läbi HTTP.

## 9. Viidatud kirjandus

- [1] L. Sterling ja K. Taveter, The Art of Agent-Oriented Modeling, MIT Press 2009.
- [2] Aleksandr Sapožnikov, Agentorienteeritud modelleerimise töövahendi graafilise kasutajaliidese kavandamine ja realisatsioon, Tallinn 2015.
- [3] Dan Abramov, React DnD dokumentatsioon. Kasutatud 23.04.2022, <https://react-dnd.github.io/react-dnd/about>
- [4] React-Raphael, React-Raphael dokumentatsioon. Kasutatud 23.04.2022, <https://www.npmjs.com/package/react-raphael>
- [5] React, React dokumentatsioon. Kasutatud 23.04.2022, <https://reactjs.org/>
- [6] A. Rõbinski, Valdkonnaontoloogia agentorienteeritud modelleerimise tööriista jaoks, Tallinn 2012.
- [7] Node, Node dokumentatsioon. Kasutatud 27.04.2022, <https://nodejs.org/en/>
- [8] Redux, Redux dokumentatsioon. Kasutatud 27.04.2022, <https://redux.js.org/>
- [9] MUI, MUI dokumentatsioon. Kasutatud 02.05.2022, <https://mui>
- [10] E-Teatmik, E-Teatmik sõnastik. Kasutatud 10.05.2022, <http://vallaste.ee/>
- [11] REST, REST dokumentatsioon. Kasutatud 10.05.2022, <https://restfulapi.net/>

## I. Litsents

### Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Lenard Piir,

*(autori nimi)*

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

AOM4STS graafilise tööriista analüüs ja arendus,  
*(lõputöö pealkiri)*

mille juhendaja on Kuldar Taveter,

*(juhendaja nimi)*

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Lenard Piir

10.05.2022