

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Juri Garanin 211734IABM

# **Tervishoiu andmeformaadi teisenduskeelte võrdlus ja analüüs**

Magistritöö

Juhendajad: Igor Bossenko  
MSc (tehnikateaduste  
magister informaatika  
erialal)

Gunnar Piho  
PhD

Tallinn 2024

## **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: Juri Garanin

03.01.2024

## Annotatsioon

Käesolevas magistritöös süvenes autor andmete konverteerimise probleemi erinevate vormingute ja standardite vahel, mis on oluline eelkõige tervishoiu valdkonnas. Autor otsustas võrrelda ja analüüsida kolme andmeformaadi teisendamise keelt ning nende keelte loogiliste mudelite konverteerimist FHIR-standardisse. Autor töötas välja konkreetse kriteeriumide süsteemi nende keelte hindamiseks. Seejärel määras autor igale keelele hindeid vastavalt konkreetsele kriteeriumile. Lõpuks analüüsis autor saadud tulemusi ja kirjeldas üksikasju, millele tuleb tähelepanu pöörata andmeformaadi teisendamise keele valimisel uurimistöö või projekti jaoks.

Kolme andmeformaadi teisendamise keele valimiseks uuris autor varasemaid meditsiiniliste andmete teisendamist käsitlevaid teadusartikleid ja uuringuid. Andmeformaadi teisendamise keeled, mida käesolevas töös võrreldakse, on FHIR Mapping Language, Whistle Data Transformation Language ja FHIR Utilized Mapping Engine.

Seejärel kirjeldas autor kriteeriumide ja punktide süsteemi nende keelte hindamiseks. Süsteem hõlmab keele funktsionaalsuse, keele äratundmise taseme ja keele kasutamise maksumuse hindamist.

Autor kirjeldas teatavate andmeformaadi teisendamise keelte peamisi puudusi, sealhulgas piiratud kogukonna toetust, mis mõjutab probleemide lahendamist ja innovatsiooni, ning madalat äratundmise taset, mis tuleneb ebapiisavast dokumentatsioonist ja juhendmaterjalidest, mis muudab need uutele kasutajatele keeruliseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 56 leheküljel, 7 peatükki, 6 joonist, 7 tabelit.

# **Abstract**

## **Comparison and analysis of mapping languages in healthcare**

In this master's thesis, the author delved into the problem of converting data between different formats and standards in healthcare. The author chose to compare and analyze three mapping languages, in the context of converting logic models to FHIR standard. The author developed a specific criteria system to evaluate these languages. Next, the author assigned scores to each language according to a specific criterion. Finally, the author analyzed the obtained results and described the details to be paid attention to when choosing a mapping language for a research or project.

To select the three mapping languages, the author studied previous research articles and papers on medical data transformation. The mapping languages that are compared in this paper are FHIR Mapping Language, Whistle Data Transformation Language and FHIR Utilized Mapping Engine.

Next, the author described a system of criteria and scores for evaluating these languages. The system includes the evaluation of language functionality, level of language familiarity, and the cost of using the language.

The author described the main weaknesses in certain mapping languages, including limited community support, which affects problem-solving and innovation, and a low familiarity level due to inadequate documentation and tutorials, making them challenging for new users.

The thesis is in Estonian and contains 56 pages of text, 7 chapters, 6 figures, 7 tables.

## Lühendite ja mõistete sõnastik

EHR	Electronic health record
HL7	Health Level Seven International
OSI	Open Systems Interconnection
FHIR	Fast Healthcare Interoperability Resources
API	Application Programming Interface
XML	Extensible Markup Language
JSON	JavaScript Object Notation
XSLT	eXtensible Stylesheet Language Transformations
QVT	Query/View/Transformation
OMG	Object Management Group
MDA	Model-Driven Architecture
MDMI	Model Driven Message Interoperability
Whistle	Whistle Data Transformation Language
FUME	FHIR Utilized Mapping Engine
LOINC	Logical Observation Identifiers Names and Codes
SNOMED CT	Systematized Nomenclature of Medicine Clinical Terms
UCUM	The Unified Code for Units of Measure
KKK	Korduma kippuvad küsimused

# Sisukord

Autorideklaratsioon .....	2
Annotatsioon.....	3
Abstract.....	4
Lühendite ja mõistete sõnastik .....	5
Sisukord.....	6
Jooniste loetelu .....	8
Tabelite loetelu .....	9
1 Sissejuhatus .....	10
1.1 Üldist .....	10
1.1.1 Koostalitlusvõime tervishoiusektoris .....	10
1.1.2 Standardid ja versioonid .....	10
1.1.3 HL7.....	10
1.1.4 Loogiline mudel.....	11
1.2 Probleem ja selle aktuaalsus .....	11
1.3 Motivatsioon.....	12
1.4 Eesmärk .....	12
1.5 Uurimisküsimused .....	12
1.6 Töös kasutatavad andmed.....	12
1.7 Töös kasutatavad tööriistad .....	12
1.8 Teoreetiline ja praktiline uudsus.....	13
1.9 Kellele on antud tööst kasu? .....	13
2 Varasema kirjanduse ülevaade .....	14
3 Metoodika.....	16
3.1 Uurimisobjekt .....	16
3.1.1 Andmeformaadi teisendamise keeled andmete teisendamiseks ühest standardist teisele.....	16
3.2 Tööriistad.....	17
3.2.1 FHIR Mapping Language.....	17
3.2.2 Whistle Data Transformation Language.....	19
3.2.3 FUME .....	19
3.2.4 Loogilise mudeli valik konverteerimiseks.....	20

4	Protsess .....	23
4.1	Loogilise mudeli andmete teisendamine valitud andmeformaadi teisendamise keelte abil.....	23
4.1.1	FML .....	23
4.1.2	Whistle.....	25
4.1.3	FUME .....	25
4.1.4	Esimene pilk .....	26
4.2	Andmeformaadi teisendamise keelte võrdlus.....	26
4.2.1	Hindamiskriteeriumid .....	26
4.2.2	Hindamiskriteeriumide tabel .....	32
4.2.3	Võrdlus .....	33
4.2.4	Kohandamine.....	41
4.2.5	Teadusuuringute olemasolu.....	42
4.2.6	Litsentsitasud.....	42
5	Analüüs ja järeldused.....	43
5.1	Tulemused .....	43
5.2	Parima keele valimine .....	43
5.3	Puudused.....	45
6	Kokkuvõte .....	46
7	Bibliograafia .....	47
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	50
	Lisa 2 – Vital Signs - Blood Pressure2 instansi näide.....	51
	Lisa 3 – Structure Map süstoolse vererõhu komponent .....	54

## Jooniste loetelu

Joonis 1. FML teisendamise protsess .....	18
Joonis 2. Whistle andmete teisendamise protsess .....	19
Joonis 3. FUME andmete teisendamise protsess.....	20
Joonis 4. Termx FML Visual Editor.....	41
Joonis 5. Mõõdetud Effektiivsus punktide jagunemine.....	44
Joonis 6. Äratundmise tase punktide jagunemine .....	44



## Tabelite loetelu

Tabel 1. MyBloodPressure loogilise mudeli ülevaade .....	20
Tabel 2. Tähemärkide arv andmeformaadi teisendamise keele järgi .....	26
Tabel 3. Kriteeriumide tabel .....	32
Tabel 4. FML veateated .....	35
Tabel 5. Whistle veateated .....	36
Tabel 6. FUME veateated .....	37
Tabel 7. Tulemuste tabel .....	43

# 1 Sissejuhatus

## 1.1 Üldist

### 1.1.1 Koostalitlusvõime tervishoiusektoris

Koostalitlusvõime tähendab erinevate süsteemide, seadmete või tarkvararakenduste võimet vahetada ja tõlgendada andmeid täpselt, tõhusalt ja turvaliselt. Tervishoiu kontekstis on koostalitlusvõime probleemid - olulised ülesanded, mis takistavad patsiendiandmete sujuvat jagamist ja kasutamist erinevate tervishoiuteenuse osutajate ja süsteemide vahel [1].

Ravivigade peamiseks põhjuseks on patsiendiinfo kättesaamatus, mis on suur tervishoiuprobleem, kuna ravivead on haiglates kuues peamine surma põhjus [27].

Selliste süsteemide loomine, mis võimaldavad tervishoius hõlpsasti teavet jagada, on oluline mõne valdkonna keerulise probleemi lahendamiseks ja võib kaasa tuua märkimisväärse kokkuhoiu tervishoiukuludes [26].

Tervishoiuandmeid säilitatakse sageli erinevates süsteemides ja formaatides, mis raskendab patsiendi haigusloo kokkuvõtmist ja sellest tervikliku ülevaate saamist [24]. Elektroonilised tervisekaardid [25] ei pruugi omavahel ühilduda, mistõttu patsiendi info on katkendlik.

### 1.1.2 Standardid ja versioonid

#### 1.1.3 HL7

HL7 (Health Level Seven International) on rahvusvaheline mittetulundusühing, mis töötab välja standardeid elektroonilise tervishoiuinfo vahetamiseks, integreerimiseks, jagamiseks ja taastamiseks. Nimi "Health Level Seven" kajastab sidesüsteemide OSI (Open Systems Interconnection) mudeli seitset taset [2].

HL7 peamine toode on sõnumite standardid, mis määratlevad meditsiinisüsteemide vahelise teabevahetuse formaadi ja järjestuse. Need standardid võimaldavad koostalitlusvõimet erinevate süsteemide vahel, nii et näiteks patsiendikaartide süsteem

saab "rääkida" laboratoorsete testide süsteemiga või elektroonilise meditsiiniandmestikuga.

HL7 Fast Healthcare Interoperability Resources (FHIR) [2] on uuem standard, mis on andmemudel ja API elektroonilise tervishoiuinfo vahetamiseks. FHIR kasutab kaasaegseid veebistandardeid ja saab käsitleda nii XML- kui ka JSON-vorminguid.

#### **1.1.4 Loogiline mudel**

Loogiline mudel viitab andmestruktuuride esitusviisile, mis võib põhineda või mitte põhineda konkreetselt FHIR raamistikul või olla sellesse rakendatud [3]. Need mudelid hõlmavad andmelemente, sealhulgas olemeid, atribuute, piiranguid ja seoseid. Need on abstraktsed andmestruktuurid, mille esitamiseks saab kasutada FHIRi. Siiski on oluline märkida, et loogilisi mudeleid saab rakendada ka muude standardite abil peale FHIRi. Selline paindlikkus võimaldab rakendada loogilisi mudeleid erinevates tervishoiukontekstides, võttes arvesse erinevaid koostalitlusvõime nõudeid ja standardeid.

## **1.2 Probleem ja selle aktuaalsus**

Erinevad meditsiiniastutused võivad kasutada erinevaid andmestandardeid. Samuti ilmuvad uued standardid ja nende uued versioonid, mida tuleb muuta. Üks peamisi nõudeid standardile on selle ühilduvus teiste standarditega, samuti standardi ühilduvus standardi eri versioonide vahel. Kui on vaja standardit muuta, siis see tähendab, et on vaja andmeid konverteerida.

Tervishoiuvaldkonna andmevahetuse käigus võib eristada järgmisi probleeme:

- Tervishoiuandmete üleminek ühest standardist teisele
- Üleminek sama standardi teisele versioonile
- Loogiliste mudelite teisendamine valitud standardile

Käesolevas töös käsitletakse andmete teisendamise probleemi lahendamist tervishoius, kasutades andmeformaadi teisendamise keeli.

### **1.3 Motivatsioon**

Praegu on Eesti tervishoiusektoris toimumas üleminek dokumendipõhiselt lähenemisviisilt ressursipõhisele ning see üleminek hõlmab ka HL7 FHIR-standardi [4] kohandamist.

### **1.4 Eesmärk**

Töö eesmärk on välja selgitada kõige sobivam vahend meditsiiniliste andmete konverteerimiseks FHIR-standardisse, analüüsides konkreetset andmeformaadi teisendamise keelt vastavalt kriteeriumide loetelule, mida on kirjeldatud punktis 4.2.2 (Hindamiskriteeriumid).

### **1.5 Uurimisküsimused**

1. Kuidas on võrreldavad erinevate andmeformaadi teisendamise keelte omadused ja võimalused tervishoius ning millised neist on kõige paremini kooskõlas FHIR-standardiga?
2. Milline andmeformaadi teisendamise keel on hindamiskriteeriumide seisukohast kõige tõhusam?
3. Milliseid probleeme ja piiranguid olemasolevates andmeformaadi teisendamise keeltes on võimalik lahendada tulevaste uuringute ja arendustööde käigus?

### **1.6 Töös kasutatavad andmed**

Mudel, mida on kirjeldatud artiklis „Resource Profile: Vital Signs - Blood Pressure2“ [5], on teisendatud FHIRi ressursiks (Observation), mida on kirjeldatud HealthSense Taltechi uurimisprojekti [9].

### **1.7 Töös kasutatavad tööriistad**

Autor koostas kriteeriumide ja hinnete süsteemi, mis aitaks valitud andmeformaadi teisendamise keeli omavahel võrrelda. Selleks, et võrrelda andmeformaadi teisendamise keelt ja teostada andmete teisendamist loogilisest mudelist FHIR-ressursiks, kasutas autor käesolevas töös Visual Studio Code rakendust.

## **1.8 Teoreetiline ja praktiline uudsus**

Nagu eespool mainitud, liigub Eesti praegu dokumendipõhistelt terviseandmetelt ressursipõhistele. Täielik üleminek FHIR-standardile nõuab suure hulga teistes standardites kasutuses olevate andmete teisendamist.

Käesoleva magistritöö teema on eriti aktuaalne seoses tervishoiu andmeformaadi teisendamise keelte ja nende keelte võrdluse peaaegu täieliku puudumisega.

## **1.9 Kellele on antud tööst kasu?**

Selle töö tulemused võivad aidata teadlastel ja arendajatel otsustada, milline andmeformaadi teisendamise keel vastab kõige paremini konkreetse projekti või organisatsiooni nõuetele ja eesmärkidele. See võib aidata maksimeerida andmete konverteerimise tõhusust, vähendada kulusid ja parandada tulemuse kvaliteeti.

Käesoleva töö tulemusi saab kasutada edasiste uuringute teoreetilise alusena ning need võivad olla kasulikud arendajatele ja ettevõtetele, kes seisavad oma projekti jaoks andmeformaadi teisendamise keele valiku ees.

## 2 Varasema kirjanduse ülevaade

Kuna HL7 FHIR standard võeti suhteliselt hiljuti kasutusele, on ühest meditsiinilisest formaadist teise üleminekut käsitlevate teadusartiklite arv piiratud. Kuna FHIR on uus lähenemisviis tervishoiualase teabe vahetamisele, on selle valdkonna teadusuuringud ja väljaanded alles kujunemas. Mida laialdasemalt standard vastu võetakse ja praktikas rakendatakse, seda rohkem on oodata selle kasutamisele ja üleminekuprotsessile pühendatud uuringute mahu suurenemist.

Enne käesoleva töö metoodika ja töövahendite valimist leidis ja uuris autor hiljutisi teadlaste kirjutatud töid.

Magistritöö koostamise alguses alustas autor FHIR-kirjanduse ülevaate lugemisega [6]. Valiti välja ja analüüsiti 80 FHIRi käsitleva artikli autorit. Oma töö tulemusena määrasid nad kindlaks avatud küsimused, väljakutsed, rakendusmudelid, kasutatavad ressursid, andmete ülemineku lähenemisviisid ja FHIRi eesmärgid.

Rishi Kanth Saripalle [7] uuris FHIR-standardi praegust olukorda tervishoiusüsteemis. Tulemused näitasid, et FHIR standard ja selle ressursid on suhteliselt hiljutised arengud biomeditsiinilise informaatika ja tervishoiu valdkonnas. Väljakujunenud standardid, nagu HL7 CCD, CDA, V2 ja V3 sõnumite edastamise standardid, on traditsiooniliselt domineerinud tööstusharus ja leidnud laialdast heakskiitu tervishoiuorganisatsioonide seas kogu maailmas.

FHIR-standardil on siiski teatavad eelised, mis eristavad seda. See kasutab ära oma tehnoloogilisi edusamme, eelkõige REST-veebiteenuste kasutamist, mis suurendavad koostalitlusvõimet ja hõlbustavad integreerimist kaasaegsete süsteemidega. Lisaks sellele on FHIR standard tuntud oma kasutajasõbralikkuse poolest, mis muudab selle kättesaadavamaks ja lihtsamini kasutatavaks kui teised HL7 standardid.

Neid funktsioone kasutades on FHIR-standardi eesmärk ületada lõhe tervishoiu infosüsteemide vahel, hõlbustada andmevahetust ja edendada tervishoiuorganisatsioonide vahelist sujuvat koostalitlusvõimet.

Igor Bossenko ja teised [8] on oma töös näidanud, et FHIR-spetsifikatsioon näeb ette meetmed edasi- ja tagasiulatava ühilduvuse tagamiseks, kuid iga arendaja peaks võtma

versioonimist tõsiselt. Autorid soovivad, et standardi arendajad koostaksid kasutusjuhendid, võttes arvesse tagasi- ja edasi-tagasi koostalitlusvõime põhimõtteid.

Viimasel ajal on FHIR Mapping Language (FML) [14] üha enam populaarsust kogunud. Kohtumisel CDA to FHIR - HL7 Europe Workshop, mis toimus 28. Aprillil 2023, esitati 3 aruannet eri riikide esindajatelt teemal "National experience in transforming CDA into FHIR". Esinejad Austriast teatasid, et nad genereerisid edukalt FHIR-standardil põhineva rahvusvahelise patsiendi kokkuvõtte, kasutades ELGA andmeid Austria riiklikust elektroonilisest tervisekaardisüsteemist [17]. Oma ülesande täitmiseks kasutasid nad FML, millega nad muutsid Austria elektroonilise tervisekaardi andmed FHIR-standardile vastavaks.

## **3 Metoodika**

### **3.1 Uurimisobjekt**

#### **3.1.1 Andmeformaadi teisendamise keeled andmete teisendamiseks ühest standardist teisele**

2020 HL7 analüüsis [10] vaatlevad autorid mõningaid andmeformaadi teisendamise keeli meditsiiniliste andmete teisendamiseks. Selles analüüsis oli igal andmeformaadi teisendamise keelel nimekiri eelistest ja puudustest.

##### **3.1.1.1 Andmeformaadi teisendamise keeled, mida oli varem kasutatud**

Ülaltoodud artiklis analüüsitud andmeformaadi teisendamise keeltest on kirjeldatud XSLT keelt. XSLT on keel, mida kasutatakse XML-dokumentide teisendamiseks teistesse dokumendivormingutesse või teistesse XML-versioonidesse [11]. XSLT on võimas vahend, kuid see on loodud igat tüüpi andmete jaoks, see ei ole meditsiiniliste andmete jaoks spetsialiseerunud keel. Üks selle puudusi on XML-keele kohustuslik kasutamine, mis seab selle kasutamisel piiranguid. Samuti on see halvasti loetav, mistõttu on seda raske õppida ja siluda.

QVT ehk "Query/View/Transformation" on Object Management Group (OMG) poolt välja töötatud spetsifikatsioon, mis kirjeldab mudelipõhise arhitektuuri (MDA) erinevate andmemudelite vahelisi transformatsioonireegleid [12]. FML-i loomiseks kasutati QVT keelt, mis on valitud üheks käesolevas töös analüüsitavaks andmeformaadi teisendamise keeleks. Nagu XSLT, on ka QVT keelt üsna raske mõista ja õppida.

Model Driven Message Interoperability (MDMI) standard ja lähenemisviis on juba keskendunud meditsiinivaldkonnale [13]. MDMI on eriti kasulik keskkondades, kus on palju heterogeenseid tervishoiu infosüsteeme, kuid selle kasutamine on keeruline.

##### **3.1.1.2 Andmeformaadi teisendamise keelte probleemid**

Andmeformaadi teisendamise keelte peamised probleemid on kehv loetavus, raskused õppimisel ja piirangud andmeformaatide osas, millega nad töötavad.



### **3.1.1.3 Käesolevas dokumendis võrdluseks valitud andmeformaadi teisendamise keeled**

Käesolevas töös valiti võrdleva analüüsi jaoks 3 andmeformaadi teisendamise keelt: FML [14], Whistle Data Transformation Language (Whistle) [15] ja FHIR Utilized Mapping Engine (FUME) [16].

Edasi kirjeldab autor põhjusi, miks just need andmeformaadi teisendamise keeled võrdluseks valiti.

FML on osa FHIRi spetsifikatsioonist, seda arutatakse FHIRi kogukonnas ja on töid, mille autorid on seda keelt edukalt kasutanud.

Artiklis [17], kus FML-i kasutatakse transformatsiooniks, mainivad autorid, et alternatiivina võiks kasutada ka Whistle.

FUME on noor andmeformaadi teisendamise keel, mille kohta autor ei ole leidnud ühtegi mainimist artiklites ja ei ole leidnud ühtegi töötavat projekti, mis seda keelt kasutaks. Siiski on selle süntaks kiirel vaatlusel selge ja loetav, funktsionaalsus on mahukas, kuid arusaadav.

Üks selle töö juhendajatest, Igor Bossenko, soovitas vaadata seda hiljutist andmeformaadi teisendamise keelt, mille süntaks on esmapilgul hästi loetav.

Vaatamata sellele, et seda keelt ei kasutata praegu kusagil (selle nooruse tõttu), usub autor, et FUME-l võib olla potentsiaali, mis realiseerub alles mõne aasta pärast.

Erinevate andmeformaadi teisendamise keelte kaasamine aitab tagada analüüsi laiahaardelisuse ja selle, et uuring hõlmab suurt hulka keeleparadigmasid.

Järgnevalt on esitatud üldine teave igaühe kohta ja nende töö põhiprintsiibid.

## **3.2 Tööriistad**

### **3.2.1 FHIR Mapping Language**

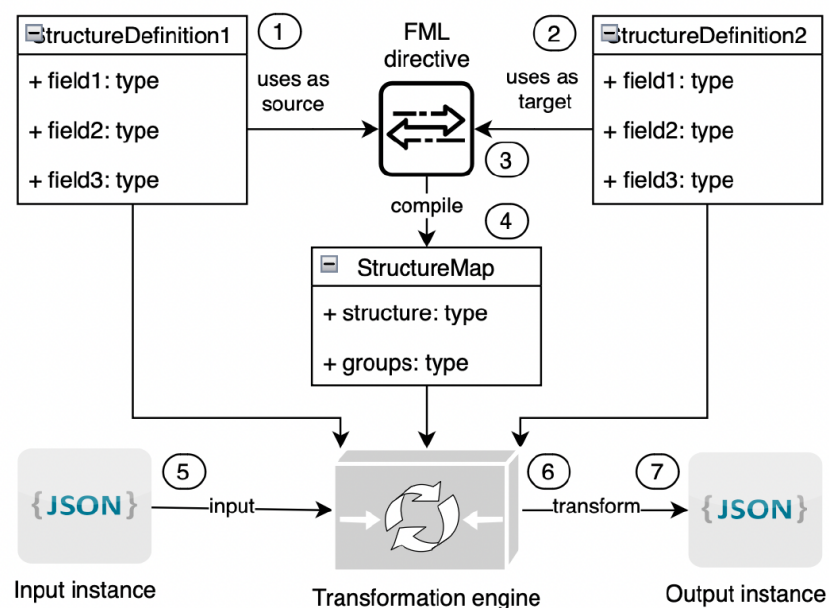
FML on FHIR-spetsifikatsiooni lahutamatu osa, mis võimaldab luua vastavusi erinevate andmemudelite vahel [14].

FML täidab andmete transformeerimise valdkonnas mitmeid eesmärke. Seda kasutatakse mitmesuguste FHIRi ressursse hõlmavate konverteerimiste korral. Esiteks hõlbustab FML FHIR-ressursside konverteerimist FHIR-standardi eri versioonide vahel, tagades ühilduvuse ja koostalitlusvõime. Teiseks võimaldab see teisendada andmeid teistest standarditest FHIR-standardisse, edendades integratsiooni ja andmevahetust erinevate tervishoiusüsteemide vahel. Lisaks toetab FML loogiliste mudelite ümberkujundamist mis tahes struktureeritud andmevormingusse, sealhulgas FHIR-standardisse.

Ühe andmemudeli teisendamiseks teise andmemudelisse on vaja esitada FML andmeformaadi teisenduste eeskiri andmeformaadi teisendamise mootorile. Sel põhjusel kasutatakse RESTful API kaudu töötamiseks HAPI FHIR valideerijat [40] või HTTP FHIR serverit [42].

FML-keeles kirjutatud juhised teisendatakse FHIR Structure Map ressursiks, kasutades operatsiooni **convert**. Structure Map on FHIRi ressurss, mis pakub kahe struktuuri vaheliste ühenduste või assotsiatsioonide teisendamist, võimaldades andmete teisendamist või ümberkujundamist nende vahel [18].

Diagramm illustreerib ühe mudeli andmete teisendamist teise mudeli andmeteks FML-i abil. (Joonis 1) [37].



Joonis 1. FML teisendamise protsess

Structure Map on FHIRi ressurss, mis illustreerib kahe struktuuri vahelisi seoseid ja võimaldab andmete ümberkujundamist. See on ressurss, mis sisaldab üksikasjalikku reeglistikku, mis määrab kindlaks, kuidas üks struktuur on seotud teise struktuuriga. Need reeglid reguleerivad lähtestruktuurist sihtstruktuuri teisendamise protsessi.

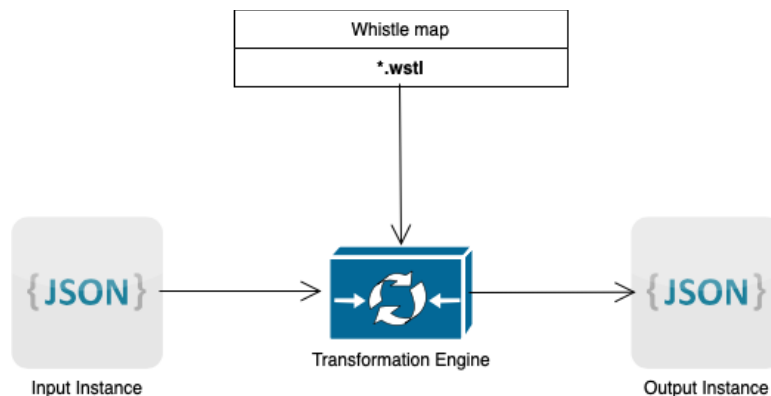
Oluline on märkida, et FHIR Structure Map ei sisalda iseenesest vastupidise teisendamise võimekust, mis tähendab, et nad keskenduvad andmete teisendamisele lähtestruktuurist sihtstruktuuri, andmata juhiseid vastupidiseks teisendamiseks.

### 3.2.2 Whistle Data Transformation Language

Whistle pakub vahendeid erinevate skeemide vaheliste seoste väljendamiseks, võimaldades kasutajatel teisendada keerulisi, üksteisega seotud andmeformaate teistesse sama keerulistesse ja üksteisega seotud andmeformaatidesse [15].

Erinevalt FMList ei nõua Whistle konverteeritavate andmete loogiliste mudelite kirjeldust. Konverteerimiseks on vaja ainult lähteandmeid JSON-vormingus ja eeskirja, mis kirjeldab konverteerimisreegleid. Transformeerimise tulemuseks on väljundandmed JSON- vormingus.

Joonis (Joonis 2) näitab andmete teisendamise protsessi Whistle'i abil.



Joonis 2. Whistle andmete teisendamise protsess

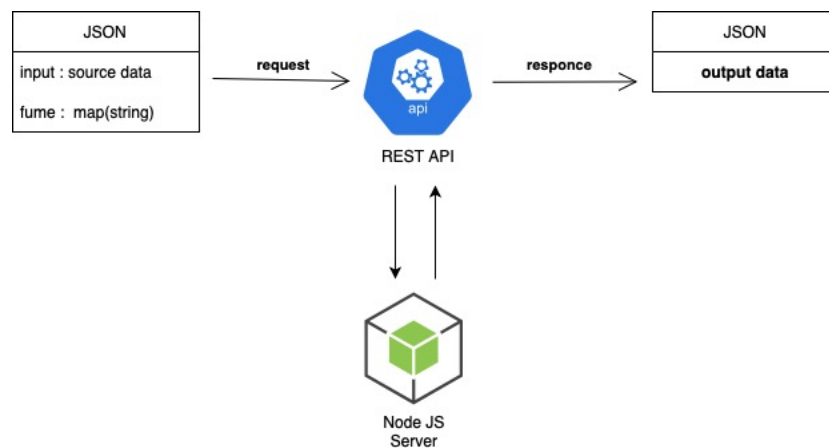
### 3.2.3 FUME

FUME on 2022. aastal Outborn'i poolt välja töötatud andmeformaadi teisendamise keel, et teisendada mis tahes andmeformaat FHIR-ressurssideks [16]. FUME on TypeScriptis kirjutatud lahendus, mis sisaldab FLASH-skripti, mis loob eeskirjad andmete konverteerimiseks. See on andmeformaadi teisendamise mootor, mis konverteerib

andmed FHIR-i ressurssideks, nõudes seejuures Restful API-d konverteerimise käivitamiseks.

Andmete teisendamiseks on vaja saata POST päring serveri aadressile, mis sisaldab JSON-objekti teisendatavate andmetega ja eeskirja. Server tagastab vastuse, mis sisaldab FHIR ressurssi.

Joonisel (Joonis 3) on näidatud andmete teisendamise protsess FUME'i abil.



Joonis 3. FUME andmete teisendamise protsess

FUME-l on veebipõhine interaktiivne kasutajaliides [31], mis võimaldab FHIR-transformatsioonide loomist, testimist ja haldamist. See uuendab tulemusi kohe. See rakendus pakub ka funktsioone, et salvestada teisendamisi ja hiljem neid edasiseks redigeerimiseks ja testimiseks üles otsida.

### 3.2.4 Loogilise mudeli valik konverteerimiseks

Enne valitud keelte võrdlemist oli ülesanne teostada andmete teisendamine iga keele abil. Selle ülesande jaoks valiti andmeallikaks loogiline mudel, mis vastutab patsiendi vererõhu andmete säilitamise eest.

Tabelis (Tabel 1) on näha MyBloodPressure mudeli kardinaalsust, andmetüüpe ja komponentide kirjeldusi.

Tabel 1. MyBloodPressure loogilise mudeli ülevaade

MyBloodPressure			
Nimetus	Kardinaalsus	Tüüp	Kirjeldus
patientName	1..1	string	Patsiendi nimi

dateOfBirth	1..1	dateTime	Patsiendi sünniaeg
systolic	1..1	integer	Süstoolse vererõhu tegelik numbriline tulemus
diastolic	1..1	integer	Diastoolse vererõhu tegelik numbriline tulemus
date	1..1	dateTime	Millal protseduur läbi viidi

See mudel koosneb kuuest komponendist: ressursi tüüp, patsiendi nimi, patsiendi sünnikuupäev, süstoolne ja diastoolne vererõhk ning mõõtmise aeg.

```
{
  "resourceType": "MyBloodPressure",
  "patientName": "Juri Garanin",
  "dateOfBirth": "03.12.1983",
  "systolic": 135,
  "diastolic": 80,
  "date": "10.01.2023"
}
```

HealthSense Implementation Guide'i (Taltech) [19] sihtmudeliks valiti FHIR-ressursi profiil: Vital Signs - Blood Pressure2. See andmemudel kujutab FHIRi vererõhuprofiili koos positsiooniga. Näide selle profiili instantsi kohta on esitatud lisas.

Komponendid, millest see koosneb:

1. Süstoolne vererõhk
2. Diastoolne vererõhk
3. Keha asend vererõhu mõõtmiseks

Igale komponendile määratakse konkreetne kood ühes kodeerimissüsteemis, LOINC või SNOMED CT (või mõlemas).

LOINC [20] ja SNOMED CT [21] on standardkodeerimissüsteemid, mida kasutatakse meditsiinis terminite ja mõistete ühtlustamiseks ja standardiseerimiseks.

UCUM [41] on kõiki mõõtühikuid hõlmav kodeerimissüsteem, mille eesmärk on võimaldada selget ja täpset elektroonilist andmevahetust suuruste ja nende vastavate mõõtühikute vahel.

Kõik need süsteemid on olulised erinevate meditsiiniliste infosüsteemide koostalitlusvõime seisukohalt.

```
"code" : {
  "coding" : [
    {
      "system" : "http://loinc.org",
      "code" : "8480-6",
      "display" : "Systolic blood pressure"
    },
    {
      "system" : "http://snomed.info/sct",
      "code" : "271649006",
      "display" : "Systolic blood pressure"
    }
  ]
},
"valueQuantity" : {
  "value" : 120,
  "unit" : "mmHg",
  "system" : "http://unitsofmeasure.org",
  "code" : "mm[Hg]"
}
```

Need süsteemid tagavad täpse ja standardiseeritud teabevahetuse eri asutuste ja riikide meditsiinitöötajate, teadlaste ja süsteemide vahel.

## **4 Protsess**

### **4.1 Loogilise mudeli andmete teisendamine valitud andmeformaadi teisendamise keelte abil**

Järgnevas segmendis on esitatud võrdlevad lähtekoodisegmendid kolmes erinevas andmeformaadi teisendamise keeltes, millest igaüks täidab analoogseid ülesandeid. Need ülesanded on:

1. Kuupäevaandmete teisendamine lähteobjektist "date" FHIR-komponendiks `effectiveDateTime`
2. Süstoolse vererõhu komponendi loomine, LOINC-koodi lisamine ja lähtemudeli süstoolse väärtuse kasutamine

#### **4.1.1 FML**

Nagu varem mainitud, on andmete teisendamisel FMLis kaks etappi. Esiteks muudetakse eeskiri FHIR-ressursiks Structure Map. Seejärel muudetakse sisendandmed (instants) FHIR-profiiliks.

Siin on näide koodi osast, mis vastutab süstoolse vererõhu komponendi teisendamise eest.

```

src.date as dt -> tgt.effective = dt "rule_datetime";

src -> tgt.component as t_component then {
  src -> t_component.code = create('CodeableConcept') as newCodeCC then
  {
    src -> newCodeCC.coding = create('Coding') as newCoding then {
      src -> newCoding.system = 'http://loinc.org' "rule-1";
      src -> newCoding.code = '8480-6' "rule-2";
    } "rule_code_1";
  } "rule_code";

  src.systolic as ss -> t_component.valueQuantity as t_vq then {
    ss -> t_vq.value = ss,
    t_vq.unit = "mmHg",
    t_vq.system = "http://unitsofmeasure.org",
    t_vq.code = "mm[Hg]" "rule_value";
  } "rule_vq";

} "rule_component";

```

Selleks, et see kood käivituks, imporditi eespool lähte- ja sihtmudeli Structure Definition ning anti lähtemudeli jaoks nimed **src** ja mudeli jaoks, millesse andmed tuleks konverteerida, nimed **tgt**.

```

uses "https://healthsense.taltech.ee/StructureDefinition/hs-observation-bp2"
alias HsObservationBloodPressure2 as target

```

```

uses "http://example.org/StructureDefinition/MyBloodPressure" alias
MyBloodPressure as source

```

```

group main(source src : MyBloodPressure, target tgt :
HsObservationBloodPressure2)

```

Esialgne ülesanne oli konverteerida .map fail Structure Map ressursiks, kasutades Mapping Engine'i funktsiooni **-convert**.

Lisas on võimalik näha, kuidas sama osa koodist, mis vastutab süstoolse vererõhu komponendi teisendamise eest, näeb välja samamoodi Structure Map ressursis.

Pärast kaardi konversiooni on vaja kutsuda funktsioon **transform** ja anda parameetritena üle Structure Mapiga saadud lähteandmed ning 2 FHIR Structure Definition profiili lähte- ja sihtmudelite jaoks.

Pärast transformatsiooni funktsiooni kutsumist loob FML-mootor sihtandmed XML- või JSON-vormingus.



### 4.1.2 Whistle

Valitud mudeli teisendamiseks vererõhu FHIR-profiiliks tuleb luua andmeformaadi teisenduste eeskiri.

Whistle süntaks kasutab funktsionaalset lähenemist. Kasutaja loob andmete teisendamiseks vajalikud funktsioonid ja seejärel kutsub neid kas üksikult või funktsioonide ahelate abil [22].

Näiteks: `Funktsioon1(Funktsioon2(Funktsioon3()))`

```
effectiveDateTime : $root.date
```

```
component[] : Component(Code(Coding("http://loinc.org", "8480-6")),  
Value_Unit_System_Code($root.systolic, "mmHg", "http://unitsofmeasure.org",  
"mm[Hg]"))
```

```
def Code(coding) {  
  coding[]: coding  
}
```

```
def Component(code, value) {  
  code: code  
  valueQuantity :value  
}
```

```
def Value_Unit_System_Code(value, unit, system, code) {  
  value: value  
  unit: unit  
  system: system  
  type: code  
}
```

```
def Coding(system, code) {  
  system: system  
  code: code  
}
```

Selles koodis saab muutuja **\$root** abil juurdepääsu lähteandmete objektidele JSONis.

Näiteks **\$root.systolic** sisaldab lähteandmetes olevat süstoolse vererõhu väärtust.

### 4.1.3 FUME

Flash-skriptide süntaks, mida FUME-s kasutatakse andmeformaadi teisenduste eeskirja loomiseks, on hästi loetav. Sellel on kõige selgem struktuur ja lihtsus. Siin on näide samast eeskirja osast, mis vastutab süstoolse vererõhu muutmise eest.

```

* effectiveDateTime = dateOfBirth
* component
  * code
    * coding
      * system = "http://loinc.org"
      * code = "8480-6"
  * valueQuantity
    * value = systolic
    * unit = "mmHg"
    * system = "http://unitsofmeasure.org"
    * code = "mm[Hg]"

```

FUME'is saab kasutada lähteandmetest saadud väärtusi lihtsalt nende nimede abil, näiteks süstoolse vererõhu väärtuse määramiseks sihtmudeli komponendile on vaja ainult koodi:

**\* value = systolic**

#### 4.1.4 Esimene pilk

Autor usub, et FUME paistab teiste keelte seast silma oma loetavuse poolest. Tabelis (Tabel 2) on esitatud samu funktsioone täitvate tähemärkide arv (ilma tühikuteta) lähtekoodis:

Tabel 2. Tähemärkide arv andmeformaadi teisendamise keele järgi

Andmeformaadi teisendamise keel	Tähemärkide arv
FML	502
Whistle	400
FUME	182

## 4.2 Andmeformaadi teisendamise keelte võrdlus

Leila Samimi-Dehkordi ja teised [23] sõnastasid ja kirjeldasid oma töös kriteeriumid andmemudelite teisendamise keele hindamiseks. Kriteeriumid, mida autor käesolevas töös kasutab, on suures osas inspireeritud eespool nimetatud artiklist, kuid neisse on tehtud muudatusi ja kohandusi.

### 4.2.1 Hindamiskriteeriumid

Loogilise mudeli teisendamise protsess tugineb lähte- ja sihtmudelitele ning andmeformaadi teisendamise keelele, mis sisaldab direktiive teisendamisprotseduuri juhtimiseks.

Nende kriteeriumide arvessevõtmine aitab samuti vähendada integratsiooniprobleemidega seotud riske ning võib aidata kaasa projekti sujuvamale ja edukamale läbiviimisele.

#### **4.2.1.1 Funktsionaalsus**

##### **Sisseehitatud funktsioonid**

Sisseehitatud funktsioonid mudeliandmete teisendamiseks.

Võimalikud variandid:

- 0 punkti - Sisseehitatud funktsioonide puudumine: Keel ei paku sisseehitatud funktsioone, mis nõuab kasutajatelt isegi põhiliste toimingute sooritamiseks rohkem koodi kirjutamist.
- 1 punkt - Piiratud hulk sisseehitatud funktsioone: Keel pakub sisseehitatud funktsioonide põhikomplekti, mis katab ainult osa standardsetest andmetöötlusoperatsioonidest.
- 2 punkti - Ulatuslik ja täielik sisseehitatud funktsioonide komplekt: Keel sisaldab laia valikut sisseehitatud funktsioone, mis hõlmavad kõiki põhilisi ja paljusid spetsialiseeritud andmetöötlusoperatsioone. Kasutajad saavad standardfunktsioonide abil hõlpsasti teostada keerulisi teisendusi, suurendades tootlikkust ja vähendades arendusele kulutatud aega.

##### **Kohandatud funktsioonid**

Kohandatud funktsioon tähendab võimalust luua ja kutsuda oma funktsioone. Selle kriteeriumi hindamine sõltub sellest, kas funktsioonile on võimalik anda argumente, kas funktsioon saab tagastada väärtusi, kas funktsiooni sees saab kutsuda teist funktsiooni, kas rekursioon on võimalik, kas on võimalik luua anonüümseid funktsioone ja anda funktsioone argumentidena.

Võimalikud variandid:

- 0 punkti - Võimatu luua kohandatud funktsioone: Keel ei võimalda luua kohandatud funktsioone, mis piirab oluliselt paindlikkust ja skaleeritavust andmetöötles.

- 1 punkt - Piiratud toetus kohandatud funktsioonidele: Keel pakub võimalust luua kohandatud funktsioone, kuid nende keerukus või võimekus on piiratud.
- 2 punkti - Täielik toetus ja paindlikkus kohandatud funktsioonide loomisel: Keel pakub täielikku vabadust ja paindlikkust kohandatud funktsioonide loomiseks ilma piiranguteta. Arendajad saavad hõlpsasti luua ja integreerida keerulisi kohandatud funktsioone, mida saab erinevates projektides uuesti kasutada.

#### 4.2.1.2 Erandite käsitlemine

Mudeli teisendamise käigus võivad tekkida erandlikud olukorrad. Turvalise teisendamise tagamine nõuab nende olukordade hoolikat käsitlemist.

Leila Samimi-Dehkordi ja teiste [23] töös oli kriteerium "Erandite käsitlemine" osa kriteeriumide rühmast "Reliability". Käesoleva töö autor otsustas panna selle kriteeriumi eraldi gruppi ja jagada selle järgmistesse kolme kategooriasse:

#### Kohandatud erandite käitlejad

Võimalus määratleda kohandatud erandikäitlejaid, et sooritada teatud vigade korral konkreetseid toiminguid. Kas on olemas **try-catch-finally** funktsioon või selle ekvivalent?

Võimalikud variandid:

- 0 punkti - Kohandatud erandite käitlejate puudumine: Keel ei paku mehhanisme kohandatud erandikäitlejate loomiseks, mistõttu arendajad on sunnitud toetuma standardsetele veakäitlusvahenditele.
- 1 punkt - Piiratud toetus kohandatud erandite käitlejatele: Keel toetab kohandatud erandikäitlejate loomist, kuid piiratud funktsionaalsuse või võimalustega. Kohandatud käitlejaid saab rakendada ainult teatud stsenaariumides või piiratud hulga eranditüüpide puhul.
- 2 punkti - Täielik ja paindlik tugi kohandatud erandite käitlejatele: Keel pakub täielikku toetust kohandatud erandikäitlejate arendamiseks ja integreerimiseks ilma piiranguteta.

#### Üksikasjalikud veateated

Võimalikud variandid:

Keel peaks andma selgeid ja informatiivseid veateateid, mis osutavad probleemi põhjusele ja võimalusel ka sellele, kuidas seda parandada.

- 0 punkti - Üksikasjalike veateadete puudumine: Keel annab väga piiratud või ebainformatiivset tagasisidet vigade kohta, mis raskendab probleemide diagnoosimist ja parandamist. Veateated on üldised või mitmetähenduslikud, viitamata probleemi konkreetsele asukohale või põhjusele koodis.
- 1 punkt - Veateated on veidi üksikasjalikumad: Keel pakub veateateid teatava detailsuse tasemega, mis võib sisaldada teavet vea tüübi ja asukoha kohta, kuid mitte alati piisavalt, et viga kiiresti lahendada.
- 2 punkti - Täielik ja üksikasjalik teave veateadetes: Keel pakub põhjalikke ja üksikasjalikke veateateid, mis sisaldavad probleemi, selle asukoha ja võimalike põhjuste täpset kirjeldust.

## **Logimine**

Võimalus logida vigu ja hoiatusi hilisemaks analüüsiks. See peaks hõlmama erinevaid logimise tasemeid, nagu DEBUG, INFO, WARN ja ERROR.

Võimalikud variandid:

- 0 punkti - Puudub sisseehitatud toetus logimisele: Keel ei paku sisseehitatud mehhanisme tegevuste või sündmuste logimiseks programmi täitmise ajal.
- 1 punkt - Piiratud logimisvõimalused: Keel pakub põhilisi logimisvõimalusi, näiteks lihtsalt sõnumite kirjutamine konsooli või faili, ilma täiendavate kohandamisvõimalusteta.
- 2 punkti - Täiustatud ja kohandatav logimine: Keel pakub võimsaid ja paindlikke logimisvahendeid, mis võimaldavad kohandada logimistasemeid, väljundformaate ja logi sihtmälu (nt failid, andmebaasid, välised jälgimissüsteemid).

### **4.2.1.3 Äratundmise tase**

Käesolevas dokumendis [23] kuulus kriteerium "Familiarity level" kriteeriumide rühma "Cost". Käesoleva töö autor otsustas panna selle kriteeriumi eraldi gruppi ja jagada selle kolme kategooriasse järgnevalt:

## **Dokumentatsioon**

Selle kriteeriumiga hinnatakse ametlike käsiraamatute, viitematerjalide, koodinäidiste ja muude teabeallikate kvaliteeti, täielikkust, juurdepääsetavust ja arusaadavust, mis aitavad kasutajatel keelt õppida ja rakendada.

Võimalikud variandid:

- 0 punkti - Dokumentatsioon puudub või on puudulik: Puuduvad ametlikud dokumendid või väga piiratud ja puudulikud materjalid. Dokumentatsiooni puudumine, milles kirjeldatakse uuendusi, veaparandusi ja uusi funktsioone.
- 1 punkt - Olemasolev, kuid puudulik või vananenud dokumentatsioon: Dokumentatsioon on olemas, kuid see võib olla vananenud või ei kajasta täielikult keele hetkeseisu. Dokumentatsioon sisaldab mõnede funktsioonide kirjeldusi, kuid ei ole piisavalt üksikasjalik või ajakohastatud.
- 2 punkti - Ulatuslik, ajakohastatud ja üksikasjalik dokumentatsioon: Täielik dokumentatsioon, sealhulgas üksikasjalik kasutusjuhend ja KKK. Dokumentatsiooni ajakohastatakse regulaarselt, et kajastada keele viimaseid muudatusi ja täiendusi. Koodinäited erinevate kasutusstsenariumide ja parimate tavade kohta, mis on uutele kasutajatele kättesaadavad ja kergesti mõistetavad.

## **Juhend**

See kriteerium viitab uute kasutajate koolitamiseks mõeldud õppevahendite kättesaadavusele, kvaliteedile ja tõhususele. Juhendid peaksid olema kättesaadavad sõltuvalt erinevatest teadmiste tasemetest, alates algajatest kuni edasijõudnud kasutajateni.

Võimalikud variandid:

- 0 punkti - Koolitusmaterjalide puudumine: Puuduvad ametlikud või mitteametlikud koolitusmaterjalid või juhendid. Puuduvad samm-sammulised juhised või näited, mis aitaksid kaasa keele omandamisel.
- 1 punkt - Piiratud või põhilised koolitusmaterjalid: On olemas mõned õppematerjalid, kuid need pakuvad ainult põhilist sissejuhatust keelde. Õppematerjalid võivad olla katkendlikud ega paku järjepidevat õpetust alates algajast kuni edasijõudnute tasemeni.

- 2 punkti - Ulatuslikud ja põhjalikud juhendid: Põhjalikud õpetused, mis hõlmavad nii keele põhi- kui ka edasijõudnute aspekte. Praktilised näited, interaktiivsed õpetused, videoõpikud ja samm-sammult juhendid, mis sobivad iseõppimiseks. Erinevate õpistiilide ja kasutajate tasemete jaoks on saadaval mitmesuguseid vahendeid.

### **Kogukond**

See kriteerium viitab arendajate ja kasutajate kogukonna aktiivsusele, suurusele ja osalemise tasemele antud keele raames.

Võimalikud variandid:

- 0 punkti - Kogukonna puudumine: Puudub aktiivne veebipõhine kohalolek või foorumid keelearutelu jaoks. Puuduvad ametlikud kanalid kasutajate tagasiside või toetuse andmiseks.
- 1 punkt - Olemasolev, kuid nõrk kogukond: Kogukond on olemas, kuid osalus on väike. Olemas on väike arv foorumeid või arutelusid, kuid uuenduste ja suhtluse sagedus on väike. Toetus on olemas, kuid vastused võivad olla aeglased või mittetäielikud.
- 2 punkti - Aktiivne ja toetatud kogukond: Regulaarne ja sagedane osalemine veebipõhistes aruteludes ja foorumites. Kiire ja tõhus tugi uutele kasutajatele ja vastutulelik kasutajatugi.

### **Kohandamine**

Selle kriteeriumiga hinnatakse, kui sageli kasutatakse konkreetset keelt reaalsetes projektides.

Võimalikud variandid:

- 0 punkti - Vähene või puuduv kohandamine: Programmeerimiskeelt kasutatakse vaevalt kommertsprojektides.
- 1 punkt - Mõõdukas kohandamine: Keel on kasutusel piiratud arvul projektides või kogub teatavates valdkondades üha suuremat populaarsust.
- 2 punkti - Kõrge kohandamine: Keel on laialdaselt kasutusel paljudes äriprojektides.

## **Teadusuuringute olemasolu**

Selle kriteeriumiga hinnatakse, kui sageli kasutatakse antud keelt teadusuuringutes.

Võimalikud variandid:

- 0 punkti - Vähene või puuduv olemasolu: Programmeerimiskeelt kasutatakse akadeemilistes uuringutes vähe. Vähene mainimine akadeemilistes artiklites ja teadustöodes.
- 1 punkt - Mõõdukas olemasolu: Akadeemilistes artiklites ja teadustöodes on mainimisi, kuid need ei ole laialt levinud.
- 2 punkti - Kõrge olemasolu: Seda keelt kasutatakse laialdaselt paljudes akadeemilistes uuringutes. Tihti mainitakse teaduspublikatsioonides ja tsiteeritakse akadeemilistes ringkondades, mis näitab selle tähtsust ja mõju.

### **4.2.1.4 Kulud**

#### **Litsentsitasud**

Keele või sellega seotud vahendite kasutamise litsentsi ostmise kulud.

Võimalikud variandid:

- 0 punkti - Kõrged litsentsitasud: Keele või platvormi kasutamine nõuab märkimisväärseid litsentsitasusid.
- 1 punkt - Mõõdukad litsentsitasud: Litsentsitasud on olemas, kuid need on mõõdukad ja neid saab õigustada pakutavate lisafunktsioonide ja toetusega.
- 2 punkti - Litsentsitasud puuduvad: Keel või platvorm on saadaval tasuta, ilma litsentsitasuta. Avatud lähtekoodiga litsents või tasuta toode, mis võimaldab seda kasutada ilma litsentsi soetamise ja hooldamisega seotud rahaliste kuludeta.

### **4.2.2 Hindamiskriteeriumide tabel**

Tabelis (Tabel 3) kirjeldatakse kõiki kriteeriume kategooriate kaupa, mida kasutatakse keelte hindamiseks, ning võimalikke hindamisvariante.

Tabel 3. Kriteeriumide tabel

<b>Kategooria</b>	<b>Kriteeriumid</b>	<b>Võimalikud variandid</b>
Funktsionaalsus	Sisesehitatud funktsioonid	0, 1, 2



	Kohandatud funktsioonid	0, 1, 2
Erandite käsitlemine	Kohandatud erandite käitlejad	0, 1, 2
	Üksikasjalikud veateated	0, 1, 2
	Logimine	0, 1, 2
Äratundmise tase	Dokumentatsioon	0, 1, 2
	Juhend	0, 1, 2
	Kogukond	0, 1, 2
	Kohendamine	0, 1, 2
	Teadusuuringute olemasolu	0, 1, 2
Kulud	Litsentsitasud	0, 1, 2

### 4.2.3 Võrdlus

#### 4.2.3.1 Sisseehitatud funktsioonid

- **FML**

FHIRi andmeformaadi teisendamise keele dokumentatsioon kirjeldab 18 sisseehitatud funktsiooni, millest ei piisanud käesolevas töös kasutamiseks valitud mudeli mõnede elementide teisendamiseks. Siiski saab funktsiooni **evaluate** kasutamisel kasutada FHIRPath koodi.

FHIRPath on keel, mis kasutab teepõhiseid meetodeid andmete navigeerimiseks ja väljavõtete tegemiseks. See toimib hierarhiliste andmemudelite kontekstis, kasutades loogilisi väljendeid, et võimaldada andmete läbimist, valimist ja filtreerimist.

FHIRPath sisaldab üle 60 sisseehitatud funktsiooni, mida saab kasutada FML-is mudelite teisendamiseks. Kuna FHIRPath funktsioone ei sisaldu FML-is, annab autor selle kriteeriumi hindamisel **1 punkti**.

- **Whistle**

Whistle sisaldab 94 funktsiooni. Impordi nõue puudub. Need on kõikjal saadaval. Lisaks sellele kirjutavad arendajad keele dokumentatsioonis, et Whistle saab kutsuda Java keeles kirjutatud funktsioone, mis on eelnevalt imporditud.

Eespool kirjeldatud teabe põhjal otsustati anda sellele kriteeriumile **2 punkti**.

- **FUME**

FUME dokumentatsioon kirjeldab 51 funktsiooni järgmistes kategooriates - String, Numeric, Date, Boolean, Array, FHIR, Object. Enamik neist on stringifunktsioonid. Autori sõnul on keelega kaasas palju kasutusvalmis funktsioone, mis hoolitsevad lihtsate ja mõningate keeruliste ülesannete eest. Selle põhjal saab see kriteerium **2 punkti**.

#### **4.2.3.2 Kohandatud funktsioonid**

- **FML**

FML tööriistad ei võimalda luua kohandatud funktsioone, kuid on võimalus taaskasutada kirjutatud koodi, importides seda teistesse andmeformaadi teisenduste eeskirjadesse. Sel viisil saab luua vajalikud teisendused vajalike elementide(ressursside) jaoks, mis sageli esinevad, ja seejärel importida neid vajaduse korral. Need on pigem protseduurid kui funktsioonid (argumentide üleandmine puudub), kuid need on kasutaja poolt loodud. Arvestades, et selle kriteeriumi 0 punkti kirjeldus ütleb, et "Võimatu luua kohandatud funktsioone", ei saa autor anda sellele kriteeriumile 0 punkti. See kriteerium saab **1 punkti**.

- **Whistle**

Selle kriteeriumi kohaselt saab Whistle **2 punkti**. Whistle'il on täielik funktsionaalsus kohandatud funktsioonide loomisel. Funktsioone saab deklareerida ja kutsuda samamoodi nagu sellistes programmeerimiskeeltes nagu C ja Python.

On võimalik luua funktsioonide ahelaid, edastades ühe funktsiooni tulemuse argumentina teisele funktsioonile.

- **FUME**

Autor ei ole leidnud keele dokumentatsioonist teavet või näiteid, mis kirjeldaksid kohandatud funktsioonide loomist.

Seetõttu saab FUME selle kriteeriumi eest **0 punkti**.

#### **4.2.3.3 Kohandatud erandite käitlejad**

- **FML**

FMLi dokumentatsioonis ei mainita kohandatud erandikäsitlejat, seega saab see kriteerium **0 punkti**.

- **Whistle**

Veakäitlus toimub withError-funktsiooni abil, mille eesmärk on sarnane try/catch-mehhanismile. See funktsioon pakub põhjalikke võimalusi kohandatud erandite käsitlemise mehhanismide loomiseks. Võttes arvesse selles kriteeriumis kirjeldatud tingimuste kirjeldust, saab ta **2 punkti**.

- **FUME**

Erandite käsitlemiseks FUME-s on ainult 2 piiratud funktsionaalsusega funktsiooni. Esimene **\$assert()** kontrollib, kas tingimus on tõene ja kui see ei ole, siis viskab vea. Teine funktsioon on **\$error()**, kui seda kutsutakse, viskab vea koos talle edastatud sõnumiga. FUME keeles ei ole midagi muud, mis aitaks erandeid käsitleda. Autor hindab seda kriteeriumit **1 punktiga**.

#### 4.2.3.4 Üksikasjalikud veateated

Selgete ja informatiivsete veateadete hindamiseks valis autor juhuslikult välja mitu juhtumit, mis põhjustavad erinevaid vigu.

- **FML**

Tabelis (Tabel 4) on kirjeldatud 5 erinevat FML juhtumit, kus autor teeb koodis tahtlikult vea, et saada viga ja selle kirjeldust.

Tabel 4. FML veateated

#	Õige lähtekood	Lähtekood koos veaga	Veateade
1	src -> tgt.id = uuid() "rule_uuid";	src -> tgt.id = uuid();	ignored due to error: Error @12, 7: Complex rules must have an explicit name
2	uses "https://healthsense.tal tech.ee/StructureDefinit ion/hs-observation-bp2" alias HsObservationBloodPressu re2 as target	uses "https://healthsense.tal tech.ee/StructureDefinit ion/hs-observation-bp2- Error" alias HsObservationBloodPressu re2 as target	Unable to find StructureDefinition for target type ( <a href="https://healthsense.taltech.ee/StructureDefiniti">https://healthsense.taltech.ee/StructureDefiniti</a>

			on/hs-observation-bp2-Error')
3	src.resourceType -> tgt.resourceType;	src.resourceType -> tgt.resourceType	Error @12, 3: Found "src" expecting ";"
4	sd -> t_vq.value = sd	sd -> t_vq.value = sd + 1	Error @102, 13: Found "1" expecting ";"
5	group main(source src : MyBloodPressure, target tgt : HsObservationBloodPressure2) {}	group main(source srcErr : MyBloodPressure, target tgt : HsObservationBloodPressure2) {}	Unknown input variable src in http://hl7.org/fhir/StructureMap/bp rule resourceType (vars = source variables [srcErr: (MyBloodPressure)], target variables [tgt: (Observation)], shared variables [])

Esimesel ja kolmandal juhul on veateates esitatud konkreetsed üksikasjad vea olemuse kohta. 1, 3 ja 4 juhul on teates selgelt märgitud konkreetne koht koodis. Neljanda juhtumi veateade ei anna piisavalt teavet probleemi mõistmiseks ja lahendamiseks. See osutab konkreetsele kohale koodis, kuid ei kirjelda vea olemust. Viiendal juhul ei viita teade konkreetsele kohale koodis ja ei anna ka piisavalt teavet probleemi mõistmiseks ja lahendamiseks.

Autori arvates on FML-i vigade kirjeldus kõige paremini kooskõlas selle kriteeriumi kirjeldusega, mis annab **1 punkti**.

- **Whistle**

Tabelis (Tabel 5) on kirjeldatud 5 erinevat Whistle juhtumit, kus autor teeb koodis tahtlikult vea, et saada viga ja selle kirjeldust.

Tabel 5. Whistle veateated

#	Õige lähtekood	Lähtekood koos veaga	Veateade
1	id : \$UUID()	id \$UUID()	Failed to load mapping config: [line 3 col 4] parser error at "\$UUID()":

			extraneous input '\$UUID' expecting {'(', ':'}
2	resourceType : "Observation"	resourceType : Observation	Failed to load mapping config: [line 1 col 15] unable to find input "Observation"
3	value: value	value: valueErr	Failed to load mapping config: [line 28 col 9] unable to find input "valueErr"
4	performer : Performer ("Organization/org1")	performer : PerformerError ("Organization/org1")	error finding projector: projector not found: PerformerError
5	status : "final"	status : "final" + 1	error extracting variadic argument 0: variadic argument error: got jsonutil.JSONStr, expected jsonutil.JSONNum Native Function Preamble "\$Sum"

Kõikidel juhtudel annab veateade konkreetsed üksikasjad vea olemuse kohta ja need üksikasjad vastavad vea iseloomule. Juhtudel 1, 2 ja 3 viitab teade selgelt konkreetsele kohale koodis. Kuna viiest juhtumist ei olnud ühtegi juhtumit, mis ei andnud teavet probleemi mõistmiseks, saab see kriteerium **2 punkti**.

- **FUME**

Tabelis (Tabel 6) on kirjeldatud 5 erinevat FUME juhtumit, kus autor teeb koodis tahtlikult vea, et saada viga ja selle kirjeldust.

Tabel 6. FUME veateated

#	Õige lähtekood	Lähtekood koos veaga	Veateade
1	InstanceOf: Observation	InstanceOf: ObservationErr	FLASH Parse error: can't find definition of ObservationErr!
2	* id = \$uuid()	* id = \$uuid()	FLASH Parse error: expected indentation of max 0 spaces, got 1 instead. Rule: * id = \$uuid()

3	* status = "final"	* status = "final"+1	The left side of the "+" operator must evaluate to a number
4	div = \$length("Text")	div = \$length(12345)	Argument 1 of function "length" does not match function signature
5	* value = systolic	* value = "systolic"	Transformation error: value 'systolic' failed RegEx defined for type decimal (Observation.component.valueQuantity.value)

Kõigil juhtudel ei ole veateadetes märgitud koodi täpset asukohta. Juhtudel 2, 3 on esitatud üksikasjad, mis vastavad vea iseloomule, muudel juhtudel on kirjeldus mitmetähenduslik ja ebaselge. Selle põhjal on selle kriteeriumi hinne **1 punkt**.

#### 4.2.3.5 Logimine

- **FML**

FML sisaldab lihtsaid logimisvõimalusi. On olemas funktsioon log(), mis võib sisaldada nii lihtsat stringi kui ka FHIRPathi väljendit. Kuna puuduvad erinevad logimise tasemed (DEBUG, INFO, WARN ja muud), saab see kriteerium **1 punkti**.

- **Whistle**

Sellel keelel on logimisvõimalus ja isegi 3 erinevat taset - INFO, SEVERE ja WARNING. Siiski puudub võimalus valida erinevaid väljundformaate ja puudub võimalus valida logi sihtmälu. Selle põhjal võib öelda, et Whistle'i logimisvõimalused on piiratud ja seda hinnatakse selle kriteeriumi alusel **1 punktiga**.

- **FUME**

Autor ei ole leidnud keele dokumentatsioonist teavet või näiteid, mis kirjeldaksid logimist.

Seetõttu saab FUME selle kriteeriumi eest 0 punkti.

#### 4.2.3.6 Dokumentatsioon

- **FML**

FMLi dokumentatsioon [14] kirjeldab autori arvates küll keele peamisi omadusi ja funktsioone, kuid sisaldab ebapiisaval hulgal koodinäiteid. See kehtib nii sisseehitatud

funktsioonide kasutamise kui ka keele põhikonstruktsioonide kohta. Keele iga versiooniga täiendatakse dokumentatsiooni, kuid näidete arv on endiselt ebapiisav. Seetõttu saab see kriteerium **1 punkti**.

- **Whistle**

Whistle'i repositoorium [15] sisaldab keele spetsifikatsioonide üksikasjalikku kirjeldust ja kasutusjuhendit. Juhend sisaldab palju näiteid ja kirjeldab üksikasjalikult keele sisseehitatud funktsioone. Whistle saab dokumentatsiooni kriteeriumi alusel **2 punkti**.

- **FUME**

Dokumentatsioon on olemas, kuid see piirdub sisseehitatud funktsioonide, operaatorite ja päringute kirjeldustega. Näiteid on ainult mõnede funktsioonide kohta. Seetõttu saab see kriteerium **1 punkti**.

#### **4.2.3.7 Juhend**

- **FML**

FMLi jaoks on olemas ametlik juhendmaterjal [28] üksikasjaliku kirjeldusega, mis koosneb 14 harjutusest ja sammust, mis kirjeldavad põhilisi olukordi andmete teisendamisel ühest loogilisest mudelist teise. Neist 14 etapist on rakendatud 12 (käesoleva töö autori poolt 11 ja 12). Lisaks õpetuskoodile sisaldab see repositoorium [29] muid näiteid FML-i kasutamise kohta. Õpik sisaldab põhilist sissejuhatust keelde ja on seetõttu hinnatud **1 punktiga**.

- **Whistle**

Whistle'il on oma repositooriumis ka dokument "Getting Started", mis õpetab lihtsat teisendamist, funktsioonide loomist ja kutsumist, tööd massiividega, tingimusi jne. See ei ole interaktiivne ja selles puuduvad samm-sammult juhised, seega saab Whistle selle kriteeriumi eest **1 punkti**.

- **FUME**

FUME-l on arendajate videoõpetus, mis koosneb 9 õppevideost, mis on pühendatud põhilisele süntaksile, andmete transformeerimisele sisseehitatud funktsioonide abil. Muid FUME-le pühendatud õpetusi autor ei leidnud. See kriteerium saab **1 punkti**.

#### 4.2.3.8 Kogukond

- **FML**

FML on osa FHIR-standardi [14] spetsifikatsioonist. FHIR-standardile pühendatud ametlikus kogukonnas [30] on olemas FMLile pühendatud tegevusvoog. Selles teemas suhtlevad FMLi kasutavad spetsialistid ja selle arendajad. Selles teemas on juba 21 alamteemat loodud.

HL7 FHIR DevDays konverentsidel peetakse FMLi teemalisi ettekandeid, nt HL7 FHIR DevDays 2020 konverentsil näitas Oliver Egger [32] (koos näidetega), kuidas teha transformeerimisi CDA standardist FHIR standardisse, kasutades FMLi. Selle ettekande esitus on avalik ja on saadaval tutvumiseks. DevDays 2020 Virtual June konverentsil näitas Vadim Peretokin [33], kuidas konverteerida loogilise mudeli andmeid FHIR-i, kasutades FML-i. Selle ettekande videot saab vaadata Youtube'i kanalil DevDays. Samuti leiab Youtube'ist ka teisi salvestusi nendelt konverentsidelt, mis on pühendatud FMLile. Amsterdamis toimunud konverentsi DevDays 2023 [34] programmis on võimalik näha FMLile pühendatud ettekandeid.

Selle teabe põhjal järeldeb autor, et huvi FMLi vastu kasvab ning asjaolu, et see on osa FHIRi spetsifikatsioonist, suurendab selle huvi tõenäosust kogu FHIRi kogukonnas. Selle kriteeriumi eest saab FML **2 punkti**.

- **Whistle**

Autoril ei ole õnnestunud leida ühtegi foorumit või gruppi sotsiaalvõrgustikes, mis oleks pühendatud keele arutelule. Whistle repositooriumis rubriigis Issues on 21 teemast ainult 10 saanud vähemalt 1 vastuse. Autor ei suutnud leida ühtegi Whistle'i keelele pühendatud kogukonda ja seetõttu saab Whistle selle kriteeriumi eest **0 punkti**.

- **FUME**

Uuringu autor ei leidnud ühtegi foorumit, kus arutataks FUME Engine'i üle. Selle keele õppimise käigus tekkis autoril raskusi, mille tõttu ta lõi küsimust repositooriumis [35]. Samal päeval andis arendaja autorile üksikasjalikud juhised koos kirjelduse ja piltidega selle probleemi lahendamiseks. See probleem oli esimene ja ainus repositooriumis. Vaatamata arendajate kiirele abile ei suutnud autor leida FUME-le pühendunud kogukonda ja seetõttu andis FUME-le selle kriteeriumi eest **0 punkti**.

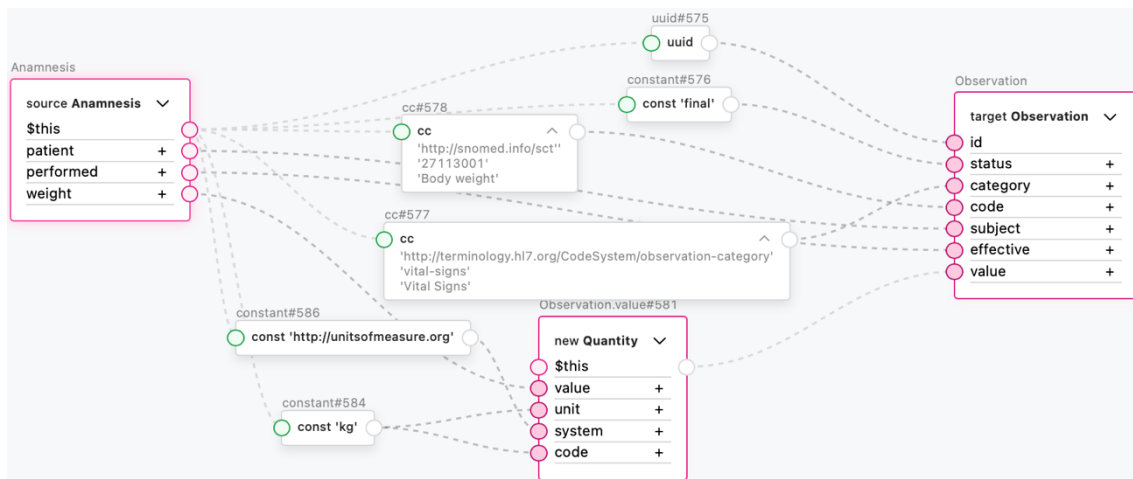


## 4.2.4 Kohandamine

- **FML**

Bossenکو ja kolleegid [36] kirjeldavad üksikasjalikult visuaalse FML-redaktori väljatöötamist oma uurimuse raames, mis on kohandatud spetsiaalselt ärianalüütikutele. Selle FML-redaktori eesmärk on eelkõige esitada visuaalselt transformatsioone, lihtsustada FML-keele keerukust ja hõlbustada analüütikute kiiret kohandamist. See visuaalne toimetaja on juba rakendatud [37] ja on kasutajatele kättesaadav.

Joonis (joonis 4) [38] illustreerib, kuidas konverteerida loogiline mudel Termx FML redaktori abil Observation ressursiks.



Joonis 4. Termx FML Visual Editor

Vaatamata sellele, et FMLile pühendatud töid või selle abil teostatud projekte on olemas, on nende arv siiski väike. Autor hindab FMLi selle kriteeriumi alusel **1 punktiga**.

- **Whistle**

Autor ei suutnud leida ühtegi projekti, mis oleks rakendatud Whistle'i abil. Seetõttu saab see kriteerium **0 punkti**.

- **FUME**

Autor ei suutnud leida ühtegi projekti, mis oleks rakendatud FUME'i abil. Seetõttu saab see kriteerium **0 punkti**.

#### 4.2.5 Teadusuuringute olemasolu

- **FML**

Alexander Dimitrov, Georg Duftschmid kirjeldavad oma artiklis [17] lahendust International Patient Summaries (IPS) automaatseks genereerimiseks FMLi abil ja näitavad genereeritud dokumendi edukat valideerimist IPS-profiilide suhtes. See kriteerium saab **1 punkti**.

- **Whistle**

Autor leidis ühe preprint'i (2023) paberist [39], kus autorid kasutasid Whistle'i andmete konverteerimiseks. Kriteeriumi kirjelduse põhjal oleks vale anda sellele kriteeriumile 0 punkti ja see saab **1 punkti**.

- **FUME**

Autor püüdis leida viiteid FUME-le IEEE Xplore, Pubmed ja Google Scholar andmebaasidest, kuid see ei andnud tulemusi. See kriteerium saab **0 punkti**.

#### 4.2.6 Litsentsitasud

- **FML**

FHIR-standardi kasutamine ja FML on FHIR-spetsifikatsiooni osa, on täiesti tasuta ja seega saab see kriteerium **2 punkti**.

- **Whistle**

Kogu Whistle'i lähtekood on avalikus repositooriumis. Autor ei leidnud dokumentatsioonist ühtegi viidet selle keele maksumusele või tasulisele versioonile, mis lubab autoril eeldada, et see on täiesti tasuta ja anda selle kriteeriumi eest **2 punkti**.

- **FUME**

Vastavalt arendaja veebilehele [32] on olemas 2 versiooni transformatsioonikeelest: FUME Community ja FUME Enterprise. Esimene versioon on tasuta, kuid piiratud funktsionaalsusega. Enterprise versiooni hinda ei ole täpsustatud, kuid on võimalik taotleda juurdepääsu demoversioonile. Arvestades, et tasuta versioon on piiratud, saab see kriteerium **1 punkti**.

## 5 Analüüs ja järeldused

### 5.1 Tulemused

See tabel (Tabel 7) sisaldab kaardistavate keelte kogupunkte ning iga kriteeriumi hindeid.

Tabel 7. Tulemuste tabel

Criteria	FML	Whistle	FUME
Sisseehitatud funktsioonid	1	2	2
Kohandatud funktsioonid	1	2	0
Kohandatud erandite käitlejad	0	2	1
Üksikasjalikud veateated	1	2	1
Logimine	1	1	0
Dokumentatsioon	1	2	1
Tutorial	1	1	1
Kogukong	2	0	0
Kohandamine	1	0	0
Teadusuuringute olemasolu	1	1	0
Litsentsitasud	2	2	1
<b>Kokku</b>	<b>12</b>	<b>15</b>	<b>7</b>

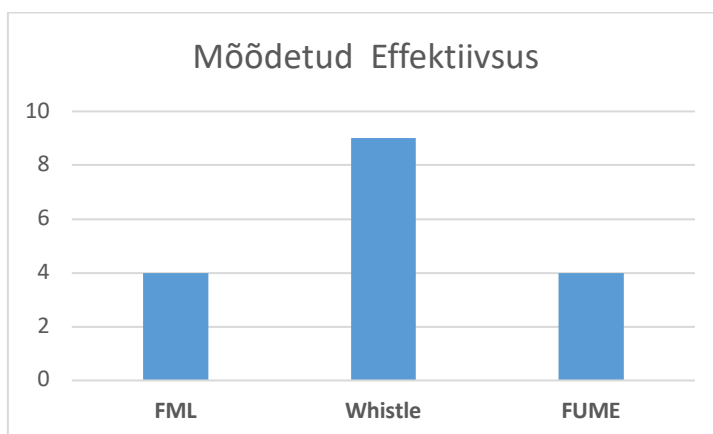
### 5.2 Parima keele valimine

Nagu nähtub saadud tabelist, jagunesid hinded järgmiselt: Kõige rohkem punkte sai Whistle (15), teisel kohal oli FML (12) ja kõige vähem punkte FUME (7).

Siiski ei piisa ühest üldhinnangust, et teha otsus andmeformaadi teisendamise keele valimisel projekti jaoks. Autor usub, et sõltuvalt olukorrast on keele valikul mõned kriteeriumid olulisemad kui teised.

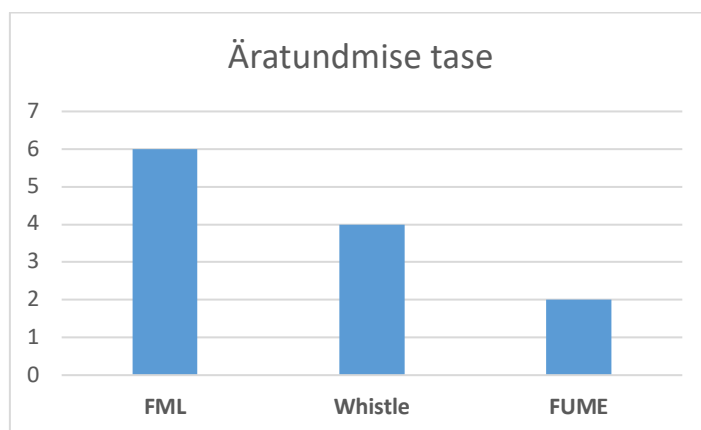
Käesolevas dokumendis kasutatud kriteeriumid on jaotatud 4 rühma: Funktsionaalsus, Erandite käsitlemine, Äratundmise tase ja Kulud. Tulemuste analüüsimisel otsustas autor ühendada funktsionaalsuse ja erandite käsitlemise ühte rühma (Mõõdetud Effektiivsus), sest mõlemad aspektid mõjutavad üheskoos keelekasutuse kasutatavust ja tõhusust.

Graafik (Joonis 5) tervikliku tõhususe kriteeriumide rühma punktide jagunemise kohta:



Joonis 5. Mõõdetud Effektiivsus punktide jagunemine

Graafik (joonis 6) punktide jagunemise kohta kriteeriumide rühma "Äratundmise tase" puhul:



Joonis 6. Äratundmise tase punktide jagunemine

**Whistle** näib olevat "parim" valik, sest selle funktsionaalsus on tugev ja erandite käsitlemine on erakordne, samuti on selle dokumentatsioon ja koolitusmaterjalid kõrge hinnanguga. Kuid selle nõrk kogukond võib olla takistuseks uutele kasutajatele.

**FML** tasakaalustab head kogukonna toetust ja rahuldavat funktsionaalsust, mistõttu on see hea valik neile, kes hindavad toetust ja jagatud kogemusi.

**FUME** saavutab paljudes kategooriates keskmised tulemused, kuid selle nõrkus kasutajate funktsioonide ja logimise osas piirab selle üldist tõhusust.

**Andmeformaadi teisendamise keele valimine projekti jaoks:**

Andmeformaadi teisendamise keele valik sõltub projekti erivajadustest. Kui funktsionaalsus ja täiustatud erandite käsitlemine on kõige olulisemad, on eelistatud valik Whistle. Kui aga kogukonna tugi ja koolituseks vajalike ressursside kättesaadavus on olulisemad, võib FML olla parem valik.

Seega sõltub "parima" keele valik projekti prioriteetidest ja kasutajate teadmiste tasemest.

### 5.3 Puudused

**Piiratud kogukonna toetus:** Iga keele puhul näib, et kogukond on erineval määral kaasatud, kuid ükski neist ei saa selles valdkonnas kõige kõrgemaid punkte. See viitab sellele, et üldiselt puuduvad tugevad ja aktiivsed kogukonnad, mis toetaksid, arutaksid ja teeksid koostööd, mis võib olla probleemide lahendamisel ja uuendustegevuses väga oluline.

**Kohandamine ja teadusuuringute olemasolu:** Mõne keele madalamad tulemused selle kriteeriumi puhul viitavad sellele, et neid ei pruugita laialdaselt kasutada tegelikes projektides või neid ei käsitleta ulatuslikult akadeemilistes uuringutes. See võib olla puuduseks kasutajatele, kes otsivad väljakujunenud ja tõestatud vahendeid.

**Madal äratundmise tase:** Dokumentatsiooni, juhendaja ja kogukonna punktisummad näitavad, et uute kasutajate jaoks võib olla erinev tase, kui kergesti nad suudavad neid keeli õppida ja nendega kohaneda. Ulatusliku dokumentatsiooni või juhendmaterjalide puudumine võib olla uute kasutajate jaoks sisenemise takistuseks.

Need üldised puudused toovad esile valdkonnad, kus arendajad ja andmeformaadi teisendamise keele kogukonnad võiksid oma jõupingutusi suurendada. Samuti on oluline, et kasutajad võtaksid neid tegureid arvesse, valides andmeformaadi teisendamise keele oma konkreetsete vajaduste ja projektide jaoks.

## 6 Kokkuvõte

Antud uuringus uuriti ja võrreldi kolme andmeformaadi teisendamise keelt: FML, Whistle ja FUME. Analüüsides erinevaid kriteeriume, sealhulgas funktsionaalsust, erandite käsitlemist, keelte äratundmise taset ja litsentsikuluid, jõuti järeldusele, et igal keelel on oma tugevused ja nõrkused, mida tuleks konkreetse projekti jaoks sobiva tööriista valimisel arvesse võtta.

Whistle paistab silma oma tugeva funktsionaalsuse ja erakordse erandite käsitlemise, samuti kõrge hindega dokumentatsiooni ja koolitusmaterjalide poolest. Siiski võib nõrk kogukond olla uute kasutajate jaoks takistuseks.

FML pakub head kogukonna tuge koos rahuldava funktsionaalsusega, mistõttu on see hea valik neile, kes hindavad toetust ja üldist kogemust.

FUME saavutab paljudes kategooriates keskmised tulemused, kuid selle üldine jõudlus kasutajafunktsioonide ja ajakirjade pidamise osas on piiratud.

Kogukonna piiratud toetus, vähene osalemine teadusuuringutes ja kohandamises ning dokumentatsiooni ja koolitusmaterjalide erinev kättesaadavus on ühised puudused, millele arendajad ja nende andmeformaadi teisendamise keelte kogukonnad peaksid tähelepanu pöörama. Neid tegureid on ka kasutajatel oluline arvesse võtta, kui nad valivad oma konkreetsete vajaduste ja projektide jaoks andmeformaadi teisendamise keelt.

Seega sõltub "parima" keele valik projekti prioriteetidest ja kasutajate teadmiste tasemest. Kõige sobivama andmetöötluskeele teadliku valiku tegemiseks tuleb arvesse võtta kõiki aspekte, sealhulgas funktsionaalsust, kogukonna toetust, koolitusressursside kättesaadavust ja kasutuskuluid.

## 7 Bibliograafia

- [1] “What is interoperability in healthcare?”. URL: <https://www.ibm.com/topics/interoperability-in-healthcare> [Kasutatud 18 september 2023]
- [2] “FHIR® - Fast Healthcare Interoperability Resources®”. URL: <https://ecqi.healthit.gov/fhir> [Kasutatud 18 september 2023]
- [3] “Logical models”. URL: <https://build.fhir.org/logical.html> [Kasutatud 20 september 2023]
- [4] Igor Bossenko, Kerli Linna, Gunnar Piho, Peeter Ross, “Migration from HL7 Clinical Document Architecture (CDA) to Fast Health Interoperability Resources (FHIR) in Infectious Disease Information System of Estonia”, 2023.
- [5] “HealthSense Implementation Guide (Taltech) - Vital Signs - Blood Pressure2”. URL: <https://ig.kodality.dev/healthsense/StructureDefinition-hs-observation-bp2.html> [Kasutatud 20 september 2023]
- [6] Muhammad Ayaz, Muhammad F Pasha, Mohammed Y Alzahrani, Rahmat Budiarto, Deris Stiawan, “The Fast Health Interoperability Resources (FHIR) Standard: Systematic Literature Review of Implementations, Applications, Challenges and Opportunities”, 2021.
- [7] Rishi Kanth Saripalle, “Fast Health Interoperability Resources (FHIR): Current Status in the Healthcare System”, 2019.
- [8] Igor Bossenko, Gunnar Piho and Peeter Ross, “Forward and backward compatibility design techniques applying the HL7 FHIR standard”, 2022.
- [9] Peeter Ross, Janek Metsallik, Kristian Juha Ismo Kankainen, Igor Bossenko, Carmen Mäe, Mall Maasik, “Health Sense: universaalse andmemudeli ja raviteekondade järjepidevuse standardi väljatöötamine lähtudes rahvusvahelistest uue põlvkonna terviseinfosüsteemide standarditest”, ISBN 978-9949-584-19-2, 2023
- [10] Lynn Laakso, “HL7 CIMI Logical Model for Analysis: Analysis Normal Form (ANF), Release 1”, 2020
- [11] “XSLT: Extensible Stylesheet Language Transformations”. URL: <https://developer.mozilla.org/en-US/docs/Web/XSLT> [Kasutatud 26 september 2023]
- [12] Edward Willink and contributors, “QVT Declarative Documentation”, URL: <https://download.eclipse.org/qvtd/doc/0.14.0/qvtd.pdf> [Kasutatud 26 september 2023]
- [13] “Model Driven Message Interoperability”. URL: <https://www.omg.org/spec/MDMI/1.0/About-MDMI> [Kasutatud 26 september 2023]
- [14] “FHIR Mapping Language”. URL: <https://build.fhir.org/mapping-language.html> [Kasutatud 1 detsember 2023]
- [15] “Whistle Data Transformation Language”. URL: <https://github.com/GoogleCloudPlatform/healthcare-data-harmonization> [Kasutatud 4 oktoober 2023]
- [16] “FUME – Introducing The Ultimate FHIR Converter!” <https://outburn.co.il/blog/fhir-utilized-mapping-engine-fhir-converter/> [Kasutatud 11 oktoober 2023]
- [17] Alexander DIMITROV, Georg DUFTSCHMID “Generation of FHIR-Based International Patient Summaries from ELGA Data” 2022
- [18] “Resource StructureMap”. URL: <http://build.fhir.org/structuremap.html> [Kasutatud 14 oktoober 2023]

- [19] “HealthSense Implementation Guide (Taltech) - Vital Signs - Blood Pressure2”. URL: <https://ig.kodality.dev/healthsense/StructureDefinition-hs-observation-bp2.html> [Kasutatud 14 oktoober 2023]
- [20] “What LOINC is”. URL: <https://loinc.org/get-started/what-loinc-is/> [Kasutatud 17 oktoober 2023]
- [21] “Overview of SNOMED CT”. URL: [https://www.nlm.nih.gov/healthit/snomedct/snomed\\_overview.html](https://www.nlm.nih.gov/healthit/snomedct/snomed_overview.html) [Kasutatud 20 oktoober 2023]
- [22] “Whistle - Getting Started”. URL: [https://github.com/GoogleCloudPlatform/healthcare-data-harmonization/blob/master/doc/getting\\_started.md](https://github.com/GoogleCloudPlatform/healthcare-data-harmonization/blob/master/doc/getting_started.md) [Kasutatud 23 oktoober 2023]
- [23] Leila Samimi-Dehkordi, Alireza Khalilian, Bahman Zamani, "Programming Language Criteria for Model Transformation Evaluation", 2014.
- [24] Randa Hammami, Hatem Bellaaj, Ahmed Hadj Kacem, "Interoperability of healthcare information systems", 2014.
- [25] “Electronic Health Records”. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2793588/> [Kasutatud 11 oktoober 2023]
- [26] Kagiso Ndlovu, Richard E. Scott, Maurice Mars "Interoperability opportunities and challenges in linking mhealth applications and eRecord systems: Botswana as an exemplar", 2021.
- [27] Amir Torab-Miandoab, Taha Samad-Soltani, Ahmadreza Jodati, Peyman Rezaei-Hachesu "Interoperability of heterogeneous health information systems: a systematic literature review", 2023.
- [28] “FHIR Mapping Language – Tutorial”. URL: <https://build.fhir.org/mapping-tutorial.html> [Kasutatud 3 november 2023]
- [29] “FHIR tutorial mapping example”. URL: <https://github.com/ahdis/fhir-mapping-tutorial> [Kasutatud 3 november 2023]
- [30] “FHIR Chat”. URL: <https://chat.fhir.org> [Kasutatud 12 november 2023]
- [31] “FUME Designer | Online FHIR Converter”. URL: <https://try.fume.health> [Kasutatud 3 november 2023]
- [32] “Oliver Egger MAP CDA to FHIR and Back”. URL: [https://www.devdays.com/wp-content/uploads/2020/12/vobgc\\_201118\\_OliverEgger\\_MapCDAtoFHIRandBack.pdf](https://www.devdays.com/wp-content/uploads/2020/12/vobgc_201118_OliverEgger_MapCDAtoFHIRandBack.pdf)
- [33] “Vadim Peretokin - FHIR Mapping Language | DevDays 2020 Virtual June”. URL: [https://www.youtube.com/watch?v=2\\_rg79JKsP8](https://www.youtube.com/watch?v=2_rg79JKsP8) [Kasutatud 5 november 2023]
- [34] “PROGRAM HL7 FHIR DevDays”. URL: [https://www.devdays.com/wp-content/uploads/2023/06/DevDays\\_schedule\\_2023\\_vs\\_02\\_06.pdf](https://www.devdays.com/wp-content/uploads/2023/06/DevDays_schedule_2023_vs_02_06.pdf) [Kasutatud 17 september 2023]
- [35] “fume-community repository”. URL: <https://github.com/Outburn-IL/fume-community/issues/2> [Kasutatud 20 oktoober 2023]
- [36] Igor Bossenko, Gunnar Piho, Roman Bondarev, Peeter Ross, "Visual FML Editor for data transformations by analysts using TermX", 2023.
- [37] “FML transformations”. URL: <https://termx.kodality.dev/modeler/transformation-definitions> [Kasutatud 24 november 2023]
- [38] “FML Editor”. URL: <https://termx.kodality.dev/wiki/termx-tutorial/fml-editor> [Kasutatud 24 november 2023]



- [39] Shahim Essaid, Jeff Andre, Ian M Brooks, Katherine H Hohman, Madelyne Hull, Sandra L Jackson, Michael G Kahn, Emily M Kraus, Neha Mandadi, Amanda K Martinez, Joyce Y Mui, Bob Zambarano, Andrey Soares, "MENDS-on-FHIR: Leveraging the OMOP common data model and FHIR standards for national chronic disease surveillance", 2023.
- [40] "HAPI FHIR". URL: validator <https://github.com/hapifhir/hapi-fhir> [Kasutatud 6 november 2023]
- [41] "What is UCUM?". URL: <http://unitsofmeasure.org> [Kasutatud 6 november 2023]
- [42] "RESTful API". URL: <https://build.fhir.org/http.html> [Kasutatud 8 november 2023]

# Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>

Mina, Juri Garanin

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Tervishoiu andmeformaadi teisenduskeelte võrdlus ja analüüs“, mille juhendajad on Igor Bossenko ja Gunnar Piho.
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

03.01.2024

---

<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 – Vital Signs - Blood Pressure2 instansi näide

```
{
  "resourceType" : "Observation",
  "id" : "BloodPressureExample2",
  "meta" : {
    "profile" : [
      "https://healthsense.taltech.ee/StructureDefinition/hs-observation-bp2"
    ]
  },
  "text" : {
    "status" : "generated",
    "div" : "text" },
  "status" : "final",
  "category" : [
    {
      "coding" : [
        {
          "system" : "http://terminology.hl7.org/CodeSystem/observation-
category",
          "code" : "vital-signs"
        }
      ]
    }
  ],
  "code" : {
    "coding" : [
      {
        "system" : "http://loinc.org",
        "code" : "85354-9"
      },
      {
        "system" : "http://snomed.info/sct",
        "code" : "75367002"
      }
    ]
  },
  "subject" : {
    "reference" : "Patient/pat1"
  },
  "effectiveDateTime" : "2021-11-23",
  "performer" : [
    {
      "reference" : "Organization/org1"
    }
  ],
  "bodySite" : {
    "coding" : [
      {
        "system" : "http://snomed.info/sct",
```

```

        "code" : "80768000",
        "display" : "Structure of left upper limb"
    }
]
},
"component" : [
    {
        "code" : {
            "coding" : [
                {
                    "system" : "http://loinc.org",
                    "code" : "8480-6"
                },
                {
                    "system" : "http://snomed.info/sct",
                    "code" : "271649006"
                }
            ]
        },
        "valueQuantity" : {
            "value" : 120,
            "unit" : "mmHg",
            "system" : "http://unitsofmeasure.org",
            "code" : "mm[Hg]"
        }
    },
    {
        "code" : {
            "coding" : [
                {
                    "system" : "http://loinc.org",
                    "code" : "8462-4"
                },
                {
                    "system" : "http://snomed.info/sct",
                    "code" : "271650006"
                }
            ]
        },
        "valueQuantity" : {
            "value" : 80,
            "unit" : "mmHg",
            "system" : "http://unitsofmeasure.org",
            "code" : "mm[Hg]"
        }
    },
    {
        "code" : {
            "coding" : [
                {
                    "system" : "http://snomed.info/sct",

```

```
        "code" : "424724000"
      }
    ]
  },
  "valueCodeableConcept" : {
    "coding" : [
      {
        "system" : "http://snomed.info/sct",
        "code" : "33586001",
        "display" : "Sitting position"
      }
    ]
  }
}
]
```

## Lisa 3 – Structure Map süstoelse vererõhu komponent

```
<rule>
  <name value="rule_component"/>
  <source>
    <context value="src"/>
  </source>
  <target>
    <context value="tgt"/>
    <element value="component"/>
    <variable value="t_component"/>
  </target>
</rule>
<rule>
  <name value="rule_code"/>
  <source>
    <context value="src"/>
  </source>
  <target>
    <context value="t_component"/>
    <element value="code"/>
    <variable value="t_code"/>
  </target>
</rule>
<rule>
  <name value="rule_coding"/>
  <source>
    <context value="src"/>
  </source>
  <target>
    <context value="t_code"/>
    <element value="coding"/>
    <variable value="t_code_coding"/>
  </target>
</rule>
<rule>
  <name value="rule_coding_code"/>
  <source>
    <context value="src"/>
  </source>
  <target>
    <context value="t_code_coding"/>
    <element value="system"/>
    <transform value="copy"/>
    <parameter>
      <valueString value="http://loinc.org"/>
    </parameter>
  </target>
  <target>
    <context value="t_code_coding"/>
    <element value="code"/>
    <transform value="copy"/>
  </target>
</rule>
```

```

        <parameter>
            <valueString value="8480-6"/>
        </parameter>
    </target>
</rule>
</rule>
<rule>
    <name value="rule_coding"/>
    <source>
        <context value="src"/>
    </source>
    <target>
        <context value="t_code"/>
        <element value="coding"/>
        <variable value="t_code_coding"/>
    </target>
</rule>
<rule>
    <name value="rule_coding_code"/>
    <source>
        <context value="src"/>
    </source>
    <target>
        <context value="t_code_coding"/>
        <element value="system"/>
        <transform value="copy"/>
        <parameter>
            <valueString value="http://snomed.info/sct"/>
        </parameter>
    </target>
    <target>
        <context value="t_code_coding"/>
        <element value="code"/>
        <transform value="copy"/>
        <parameter>
            <valueString value="271649006"/>
        </parameter>
    </target>
</rule>
</rule>
</rule>
<rule>
    <name value="rule_vq"/>
    <source>
        <context value="src"/>
        <element value="systolic"/>
        <variable value="ss"/>
    </source>
    <target>
        <context value="t_component"/>
        <element value="valueQuantity"/>
        <variable value="t_vq"/>
    </target>
</rule>

```

```

</target>
<rule>
  <name value="rule_value"/>
  <source>
    <context value="ss"/>
  </source>
  <target>
    <context value="t_vq"/>
    <element value="value"/>
    <transform value="copy"/>
    <parameter>
      <valueId value="ss"/>
    </parameter>
  </target>
  <target>
    <context value="t_vq"/>
    <element value="unit"/>
    <transform value="copy"/>
    <parameter>
      <valueString value="mmHg"/>
    </parameter>
  </target>
  <target>
    <context value="t_vq"/>
    <element value="system"/>
    <transform value="copy"/>
    <parameter>
      <valueString value="http://unitsofmeasure.org"/>
    </parameter>
  </target>
  <target>
    <context value="t_vq"/>
    <element value="code"/>
    <transform value="copy"/>
    <parameter>
      <valueString value="mm[Hg]"/>
    </parameter>
  </target>
</rule>
</rule>
</rule>

```