

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Liisa Kalling 206299IABB

**Toote arhetüüp mustri realiseerimine, testimine  
ja vastavuse hindamine ISO 13940:2015  
(ContSys) põhjal mõistete deklaratiivseks  
spetsifitseerimiseks**

Bakalaureusetöö

Juhendaja: Toomas Klementi  
MSc  
Gunnar Piho  
PhD

Tallinn 2023

## **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: Liisa Kalling

17.05.2023

## **Annotatsioon**

Meditsiin muutub järjest enam patsiendikeskseks ning seetõttu pööratakse suuremat tähelepanu ravi järjepidevuse tagamisele. Selleks on vaja meditsiinis heterogeensete infosüsteemide ehk koostalitlusvõimeliste ja standardiseeritud IT lahenduste kasutamist. Käesoleva töö eesmärk on Toote arhetüübi käsitledes terviseandmete edastamise tarkvara ABC4HEDA sidusust ja vastavust ISO 13940:2015 meditsiinistandardiga. ABC4HEDA on semantiline alusmudel, mis loodi selleks, et muuta erinevate raviasutuste meditsiiniandmed formaliseerituks ja koostalitlusvõimeliseks. ISO 13940:2015 on mõistesüsteem tervishoiu ja arstiabi järjepidevuse toetamiseks ning tervishoiuorganisatsiooni kõikide tasandite koostalitlusvõime saavutamiseks. Töö eesmärk on panna kirja meditsiinistandard niimoodi, et IT pool suudaks seda infosüsteemis realiseerida. Selle tulemusel on võimalik ühtlustada erinevate raviasutuste infosüsteemid, et aidata tagada ravi järjepidevus ja inimeste pikaajaline tervis.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 22 leheküljel, 5 peatükki, 8 joonist, 6 tabelit.

## **Abstract**

### **Realization, testing, and suitability assessment of the Product archetype pattern based on declarative specification of concepts according to ISO 13940:2015 (ContSys)**

Medicine is becoming increasingly patient-centered, and therefore greater attention is paid to ensuring the continuity of treatment. This requires the use of heterogeneous information systems, or interoperable and standardized IT solutions in medicine. The aim of this work is to address the coherence and compliance of the ABC4HEDA health data transmission software with the Product archetype and the ISO 13940:2015 medical standard. ABC4HEDA is a semantic foundation model created to formalize and enable interoperability of medical data across different healthcare facilities. ISO 13940:2015 is a terminology system aimed at supporting continuity of care and achieving interoperability at all levels of healthcare organizations. The main goal of this thesis is to evaluate the suitability of the ABC4HEDA model for specifying products defined in the ContSys standard. Since the ABC4HEDA product archetype was previously implemented incompletely and according to the MVC (model-view-controller) architecture, another accompanying goal of this work was to fully implement and test these models and to add SPA (single page application) architectural support to ABC4HEDA archetypes. The aim of this thesis is to document the medical standard in a way that the IT side can implement it in information systems. This will help to standardize different healthcare information systems and ensure treatment continuity and long-term health for people.

The thesis is in Estonian and contains 22 pages of text, 5 chapters, 8 figures, 6 tables.

## Lühendite ja mõistete sõnastik

ABC4HEDA	<i>Archetypes Based Computing for Health Data</i> ehk arhetüüpidel ja arhetüüpmuustritel põhineva isiklike terviseandmete edastamise tarkvara
Arhetüüp	Süsteemi mõne olulise komponendi mudel (näiteks osapool, roll, protsess, tellimus, reegel)
ISO	<i>International Organization for Standardization</i> ehk Rahvusvaheline Standardiorganisatsioon
UML	<i>Unified Modeling Language</i> ehk ühtne visualiseerimiskeel
ContSys	ISO 13940:2015
MVC	<i>Model-View-Controller</i>
SPA	<i>Single Page Application</i>

## Sisukord

1 Sissejuhatus .....	9
1.1 Üldine .....	9
1.2 Probleem .....	9
1.3 Eesmärk .....	9
1.4 Töö struktuur .....	10
2 Metoodika .....	11
2.1 ABC4HEDA .....	11
2.2 Toote arhetüüp muster .....	11
2.3 ISO 13940:2015 .....	14
2.4 Tööprotsess ja kasutatud tööriistad .....	14
3 Peamised tulemused .....	16
3.1 ABC4HEDA Toote arhetüüpmustri valideerimine .....	16
3.1.1 Tervishoid teenuste protsessi kirjeldusena .....	17
3.2 Testimine ja testimise mustrid .....	20
4 Analüüs ja järeldused .....	22
4.1 Toote SPA raamistiku analüüs .....	22
4.2 Toote disainimustrid .....	23
4.3 Toote sobivus valdkonnapõhiseks spetsifitseerimiseks .....	25
4.4 Toote kajastamine erinevates publikatsioonides .....	27
4.5 Kitsendused ja piirangud töös .....	28
4.6 Edasised tööd .....	29
5 Kokkuvõte .....	29
6 Kasutatud kirjandus .....	31
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	34

## Jooniste loetelu

Joonis 1. Erinevad toote tüüpid autori joonis. ....	12
Joonis 2. Toote arhetüüpumustri joonis autori joonis. ....	13
Joonis 3. ABC4HEDA Git commit väljavõte.....	15
Joonis 4. Näide editoride testklassist EditCatalogEntries näitel.....	21
Joonis 5. Näide editoride testimise sisse võetavatest meetoditest. ....	21
Joonis 6. Üleminek HTML ja C# vahel.....	23
Joonis 7. Klasside nimetuste näidis. ....	24
Joonis 8. Kitsendus MasterDetail näite põhjal. ....	29

## **Tabelite loetelu**

Tabel 1. Tootega seotud terminid. ....	16
Tabel 2. Teenusega seotud terminid. ....	17
Tabel 3. Tervishoiuteenustega seotud mõisted. ....	18
Tabel 4. Seisunditega seotud mõisted. ....	20
Tabel 5. Võrdlus T.Sõerdi lõputööga. ....	26
Tabel 6. Võrdlus J.Buhvestova lõputööga. ....	26



# 1 Sissejuhatus

## 1.1 Üldine

Meditsiin muutub ajapikku patsiendikesksemaks, mille tõttu pööratakse rohkem tähelepanu ravi järjepidevusele. Selle saavutamiseks on vaja heterogeenseid infosüsteeme ehk koostalitlusvõimelisi ja standardiseeritud IT lahendusi meditsiinis. Käesolevas töös kasutatakse ISO 13940:2015 standardit, et spetsifitseerida meditsiinis kasutusel olevad terminid. Standardi eesmärk on kindlaks määrata nõuded tervishoiuteabele hõlmates kõiki spetsifikatsioonitasemeid. Töö autor valideerib neid termineid ABC4HEDA alusmudelis, et pakkuda arstidele ühte võimalikku mudelit. Lõputööd kaitstakse 2023 aasta juunis.

## 1.2 Probleem

ISO 13040:2015 (ContSys) [1] on 2015. aastast pärit rahvusvaheline meditsiiniinformaatika standard tervishoiu ja arstiabi järjepidevuse toetamiseks. Standardi eestikeelne versioon [2] avaldati 2023. aasta veebruaris. Toote arhetüüpimuster on ABC4HEDA (Archetypes Based Computing for Health Data) valdkonnamudeli [3] toodete deklaratiivseks spetsifitseerimiseks infosüsteemide käitluse ajal (runtime). 2022 aastal Äriinfotehnoloogia bakalaureuse õppekaval kaitstud lõputöodes hinnati ABC4HEDA valdkonnamudeli sobivust lähtuvalt LOINC [4], HL7 FHIR [5] ja ContSys protsessidest [6], kuid seni pole hinnatud ABC4HEDA mudeli sobivust ContSys standardis defineeritud toodete deklaratiivseks spetsifitseerimiseks.

## 1.3 Eesmärk

Töö peamine eesmärk on hinnata ABC4HEDA mudeli sobivust ContSys standardis defineeritud toodete spetsifitseerimiseks. Kuna ABC4HEDA toote arhetüüpimuster oli realiseeritud seni mittetäielikult ning vastavalt MVC (*model-view-controller*) [7] arhitektuurile, oli töö kaasnevaks eesmärgiks antud mudelite täielik realiseerimine ja testimine ning ABC4HEDA muustritele SPA (*single page application*) [8] arhitektuurse

toe lisamine. Töö eesmärk on panna kirja meditsiinistandard niimoodi, et infotehnoloogiline pool suudaks seda infosüsteemis realiseerida. Tänu sellele on võimalik ühtlustada erinevate raviasutuste infosüsteemid, et aidata tagada ravi järjepidevus ja inimeste pikaajaline tervis.

#### **1.4 Töö struktuur**

Lõputöö on jagatud neljaks suuremaks peatükiks. Esimeses kirjeldatakse käesoleva töö eesmärke, lähtudes arstiravi igapäevastest probleemidest. Teises peatükis kirjutatakse lahti meetodika, ISO 13940:2015 standard, ABC4HEDA alusmudel ja ABC4HEDA toote arhetüübi muster. Kolmandas peatükis ehk peamistes tulemustes selgitatakse põhjalikult toote mustrit järgides ISO 13940:2015 standardit ja sidudes seda ABC4HEDA toote arhetüübi mustriga ehk teisisõnu leitakse lahendus esimeses peatükis tõstatatud probleemile. Samuti selgitatakse toodete testimist ABC4HEDA rakenduses. Viimases osas analüüsitakse põhjalikult valminud toote arhetüübi valdkonna põhist spetsifitseerimist ning selgitatakse pikemalt lahti toote arhetüübi SPA muster ja disainimuster. Lisaks analüüsitakse toote arhetüübi käsitlust muudes allikates ja selgitatakse projekti järgmised sammud.

## 2 Metoodika

Käesoleva töö koostamisel kasutatakse äriinfotehnoloogia erialal omandatud teadmisi full-stack tarkvaraarenduse kohta, tuginedes puhta koodi [9], ekstreemprogrammeerimise [10] ja testidel tugineva arenduse [11] nõuetele. Lisaks ContSys standardile kasutatakse ContSys terminite identifitseerimisel ka spetsiaalset veebipõhist abivahendit [12]. Töös kasutatakse Visual Studio 2022 vabavaralist versiooni [13], kõige uuemat C# programmeerimiskeele versiooni 11.0 [14] ning ASP.NET Core 7.0 [15], Blazor SPA [16] ja Radzen Blazor [17] raamistikke.

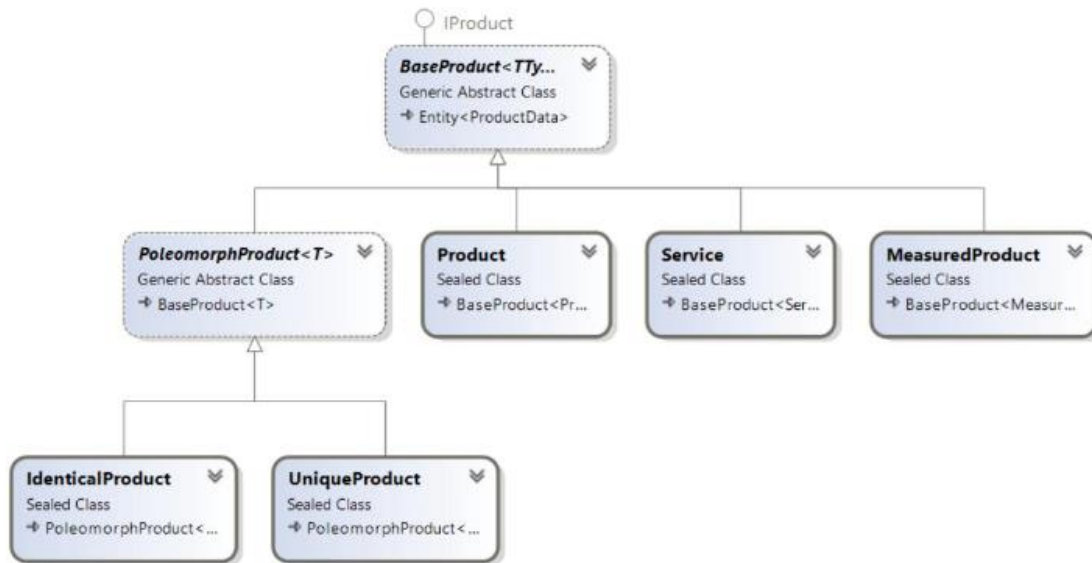
### 2.1 ABC4HEDA

ABC4HEDA [3] on semantiline alusmudel, mis loodi, et muuta erinevate raviasutuste meditsiiniandmed formaliseerituks ja koostalitlusvõimeliseks. ABC4HEDA mudeli eesmärgiks on toetada meditsiiniinformaatikas kasutusel olevate terminoloogiate ja nõuete deklaratiivset esitust ja selle kaudu tagada ravi järjepidevus ja inimeste pikaajaline tervis. See toimub nii, et üks meditsiiniiasutus annab meditsiiniinformatsiooni enda standarditele vastavas formaadis (näiteks HL7 CDA [18]) ABC4HEDA kaudu, kus kättesaadud informatsiooni töödeldakse ja edastatakse teisele raviasutusele neile vajalikus formaadis (näiteks HL7 FHIR [5]). Mudel on üles ehitatud kasutades Arlow ja Newstadt arhetüüpmustreid [19] ja seda edasi arendades. Terviseandmed on esitatud vastavalt Zachmani raamistikule [20].

### 2.2 Toote arhetüüpmuster

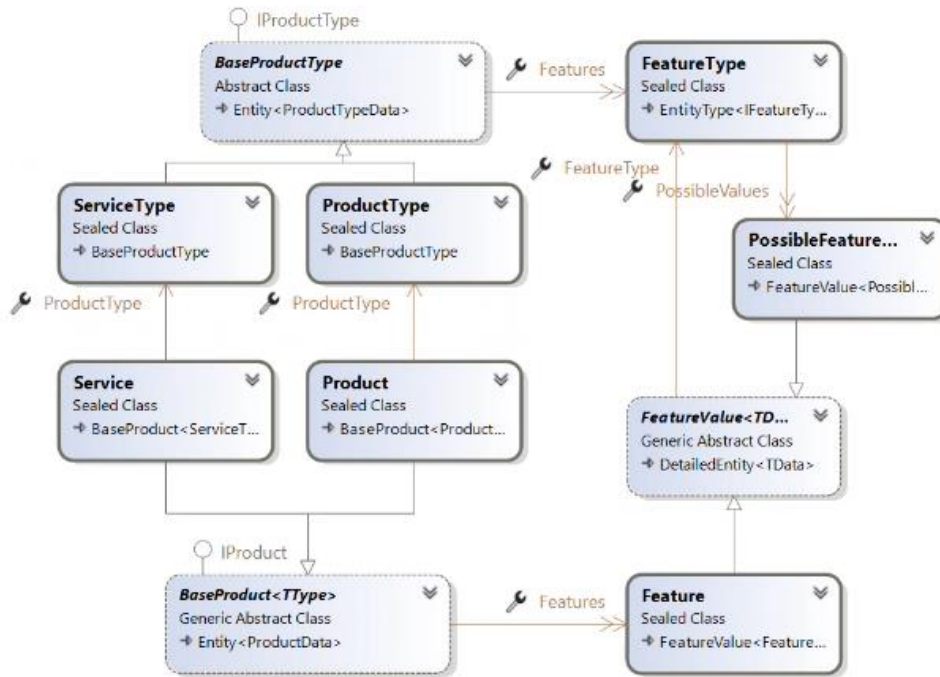
Kõik ärid tegelevad toodete pakkumisega, olgu selleks siis tooted või teenused. Arlow ja Newstadt [19] vaatlevad toodet rohkem müümise punktist kui selle tootmise punktist. Toode jaguneb pleomorfseks tooteks, tooteks, teenuseks ja mõõdetavaks tooteks (Joonis 1). Pleomorfsed tooted jagunevad identsed tooted ja unikaalsed tooted. Unikaalsed tooted on tooted, mis on täielikult ainulaadsed. Sobivaks näiteks on kunstiteosed. Identsed tooted on tooted, mille omadused on detailideni ühesugused ja, mida ei ole võimalik üksteisest

eristada ehk tegemist on masstoodanguga. Identsete toodete puhul võib tegemist olla erinevate partiide (*Batch*) väljalaskega, et tagada kvaliteedi kontroll.



Joonis 1. Erinevad toote tüüpid autori joonis.

Toode (*Product*) on baasklass erinevatele toote tüüpidele (Joonis 2). Toote tüüp (*ProductType*) defineerib informatsiooni kõikide sarnaste toodete kohta. Igal erineval toote tüübil peab olema oma nimi (*Name*), näiteks „30ml suurune veenivere proov“, ja unikaalne identifikaator (*ProductTypeId*) tootetüüpide eristamiseks. Tooted võivad olla jagatud partiidesse (*Batch*). Sellisel juhul sisaldavad need partiid identifitseerivat numbrit (*BatchId*). Tunnuste tüüpi (*FeatureType*) arhetüüp esindab toodete ja teenuste teatud omaduste tüüpi (näiteks kolesterooli sisaldus veres), millele konkreetne väärtus (näiteks 189 mg/dl) spetsifitseeritakse vastava toote (näiteks veeniveri) omaduse (*Feature*) kaudu ja mis on ära määratletud võimalike tunnuste väärtuste (*PossibleFeatureValue*) järgi. Näiteks saab *PossibleFeatureValue* abil määrata ära keskmise kolesterooli väärtuste piirid normaalse, piiripealse ja kõrge kolesteroolitaseme jaoks erinevate sugude- ja vanusekategoriate järgi.



Joonis 2. Toote arhetüüpmustri joonis autori joonis.

Teenuse on kirjeldatud kui protsessi või tegevust, mida müüakse. Arlow ja Newstadt [19] kohaselt seisneb peamine semantiline erinevus kaupade ja teenuste vahel selles, et kaup esindab tavaliselt mingit objekti (näiteks arvutisüsteemi), mis on enam-vähem konkreetne, samas kui teenus esindab protsessi või tegevust (näiteks juukselõikust või hotellis viibimist), mis pole konkreetne. Teenuse arhetüüp on moduleeritud väga sarnaselt toote arhetüübile. Teenuse tüüp (*ServiceType*) defineerib ära üldise informatsiooni kõikide sarnaste teenuste kohta. Igal teenuse tüübil peab olema oma unikaalne nimi (*Name*) ja identifikaator (*ServiceTypeId*). Erinevalt toote tüübist lisandub teenusetüübile juurde veel teenuse algus (*Start*) ja lõpp (*End*), et paremini teenust monitoorida.

Mõõdetavad tooted on tooted, mille omadused on kvantifitseeritavad ja mõõdetavad mingisuguste standartsete mõõtmisvahendite abil. Näiteks riideid müüakse suuruste ja mõõtude järgi, elektrit kilovattidega tunnis ja juurvilju müüakse poes kaalu järgi. Mõõdetavate toodete puhul on oluline, et nende omadused oleks selgesti määratletud ja standardiseeritud, et neid oleks võimalik objektiivselt hinnata ja võrrelda. See võimaldab ettevõtetel tagada toodete kvaliteeti ja tõhusust ja klientidel teha paremaid ostuotsuseid

## 2.3 ISO 13940:2015

ISO 13940:2015 [1] on mõistesüsteem tervishoiu ja arstiabi järjepidevuse toetamiseks ning tervishoiuorganisatsiooni kõikide tasandite koostalitlusvõime saavutamiseks. Standard loodi 2016 aastal ning 2023 alguses tõlgiti ISO standardi eesti keelde Tervise Arengu Instituudi soovil. Selle rahvusvahelise standardi eesmärk on praktikas määratleda nõuded tervishoiuteabele ning hõlmata kõiki spetsifikatsioonitasemeid, et luua ärimudeleid, mis toimiksid rahvusvahelise, riikliku või kohaliku tasandi ühise alusena koostalitlusvõimeliste infosüsteemide ja struktureeritud teabe arendamiseks kindlaksmääratud kliiniliste protsesside jaoks.

Standard määratleb üldised terminid, mis on kasutuses arstiabi andmisel ja inimeste tervise järjepidevuse saavutamiseks ning need mõisted hõlmavad peamiselt arstide ja patsientide vahelist suhtlust. Mõisteid on kokku üle 150, mille omavahelised seosed on illustreeritud UML-diagrammidega. Mõisted jagunevad kaheksasse valdkonda: tervishoiutöötajad, tervishoiuküsimused, tegevused, protsessid, tervishoiu planeerimine, aeg, kohustused ja teabehaldus.

ISO standardi kasutajaid hõlmab kõiki osapooli, kes on huvitatud tervishoiu koostalitlusvõime probleemidest, muu hulgas tervishoiutöötajaid ja nende meeskondi, patsiente, tervishoiu valdkonna juhte, tervishoidu rahastavaid organisatsioone ning igat tüüpi tervishoiuteenuse osutajaid ning tervishoius töötavaid meeskondi.

## 2.4 Tööprotsess ja kasutatud tööriistad

Tarkvara ja meta-mudeli arendamiseks kasutatakse tööriistu nagu C# [14], mis hõlmab Blazori [17] komponente, Asp.Net Core 7.0 [15], Entity Framework Core [21] ja Visual Studio 2022 Community [13]. Andmebaasina on kasutusele võetud Microsoft SQL Server [22]. Projekti versioonide haldamiseks kasutatakse Microsoft DevOps. Lisaks on sinna seadistatud *pipeline*.

Tööprotsess kujunes välja nii, et aasta alguses jagati meeskonnasiseselt ära lõputöö teemad ja tegeleti programmeerimisega kuni aprillini (Joonis 3). Aprilli alguses toimus üleminek programmeerimiselt bakalaureusetöö jaoks vajaliku analüüsi ja kirjutamise peale. Iganädalaselt tehti koosolekuid esmaspäeviti ja neljapäeviti kõikide

meeskonnaliikmete ja juhendaja vahel, et vaadata seni tehtud töid, tagada meeskonnaliikmete ühine arusaam ja saada juhendajapoolset tagasisidet.

13092/0/149 - Errors in PriceViewFactoryTests is fixed by making changes into the Tests4Vi...	Gunnar Pi...	10/03/202...	bef987d8
Merge branch 'dev' of https://dev.azure.com/Nunnu/Abc/_git/Abc into dev	Jane-Ly B...	10/03/202...	a2ae3a80
Merge branch 'dev' of https://dev.azure.com/Nunnu/Abc/_git/Abc into dev	Liisa Kalling	10/03/202...	e6ff7d09
13089/2/147 - Editors tested in EditProductCategoriesTests	Liisa Kalling	10/03/202...	7bd10e3e
PartyRoleConstraintType domain modified	Jane-Ly B...	10/03/202...	5ddf8405
13089/2/149 - RoleConstraintTypes pages test added	Jane-Ly B...	10/03/202...	34c28fc5
Changes to CatalogEntryView	Liisa Kalling	10/03/202...	900e7272
EditCatalogEntries done, CatalogEntries.razor in progress	Liisa Kalling	10/03/202...	5523fea9
13088/2/147 - CatalogEntriesController done	Liisa Kalling	10/03/202...	19d58244
Changes to Batches (Client)	Liisa Kalling	10/03/202...	9f31ab46
13088/2/147 - Batches fixed	Liisa Kalling	09/03/202...	f47394d7
13091/2/145 - ProductCategoryViewFactoryTests done	Liisa Kalling	09/03/202...	7ac1ab67
Changes again	Liisa Kalling	09/03/202...	9c7c654c
13090/2/145 - Changes to ProductCategoryViewFactory	Liisa Kalling	09/03/202...	56933a27
ProductPrice (in progress, not tested) - controller [done], client [done], razor [done]	Viljam Pu...	09/03/202...	87a23601
fatal error	Jane-Ly B...	09/03/202...	981d1509
13063/2/146 Editors are tested in EditBatchesTests, tests in PriceViewFactoryTests are failing	Liisa Kalling	09/03/202...	d9f9fc60
Changes to Batches (Client)	Liisa Kalling	09/03/202...	75e9b2b7
Dropdown list added to EditBatches	Liisa Kalling	09/03/202...	21a5fe7e
Started working on BatchesControllerTest but it fails so i commented everything out	Liisa Kalling	09/03/202...	a6a3ea28
Merge branch 'dev' of https://dev.azure.com/Nunnu/Abc/_git/Abc into dev	Gunnar Pi...	09/03/202...	2e842d65
ProductPrice (in progress, not tested) - controller [done], client [done], razor [done?]	Viljam Pu...	09/03/202...	60c9ebae
Merge branch 'dev' of https://dev.azure.com/Nunnu/Abc/_git/Abc into dev	Viljam Pu...	09/03/202...	9e0d955a
Started working on Batches.client and EditBatches	Liisa Kalling	08/03/202...	d596fd03
BatchesController done	Liisa Kalling	08/03/202...	4009f50e
13041/0/145 - ProductCategoriesControllerTests done	Liisa Kalling	08/03/202...	1e7a05ea
13020/0/145 - EditProductCategoriesTests not entirely done	Liisa Kalling	08/03/202...	9861aa75
Changes to ProductCategoryViewFactory and ProductCategoryView	Liisa Kalling	08/03/202...	3d3d1371
ProductPrice (in progress, not tested) - controller [done], client [done], razor [done?]	Viljam Pu...	09/03/202...	c14bddbe

Joonis 3. ABC4HEDA Git commit väljavõte.

## 3 Peamised tulemused

### 3.1 ABC4HEDA Toote arhetüüpumustri valideerimine

Siin peatükis on formaliseeritud toote arhetüüpumustri ISO standardi järgi valideerides seda ABC4HEDA meta-mudelis. ABC4HEDA mudeli valideeringud on iga mõiste taha sulgudesse lisatud kui just ei ole muud moodi ära mainitud.

ISO standardis on toode defineeritud kui organisatsiooni tegevuse väljund, mida saab toota ilma organisatsiooni ja kliendi vahelise suhtluseta. Toode on protsessi tulemusel saadud lõppväljund, mis jaguneb toodeteks ja teenusteks. Toote peamine tunnus on, et see on tavaliselt käegakatsutav, kuid kui tulemused pole käegakatsutavad, on toodet mõeldud teenusena. Pakutavaid tooteid ja teenuseid saavad selle organisatsiooni peamised kliendid. ABC4HEDA mudelis on toode määratletud kui *Product* (Tabel 1).

Tabel 1. Tootega seotud terminid.

ISO 13940 termin (ingl)	Lõputöös kasutatud ISO 13940 termini vaste (ee)	ABC4HEDA arhetüüp
Product	Toode	Product
Medicinal product	Ravim	Product
Medical device	Meditsiiniseade	Product
Automated medical device	Automaatne meditsiiniseade	Product

ISO standardi kohaselt on teenus organisatsiooni tegevuse väljund, mille puhul vähemalt üks tegevus peab toimuma organisatsiooni ja kliendi vahel. Teenus hõlmab sageli kliendiga suhtlemist nii kliendi nõuete tuvastamiseks, aga ka teenuse osutamise vältel ning see võib sisaldada jätkuvat suhet. Teenuse osutamine jaguneb ISO standardi järgi neljaks: tegevus toimub kliendi tarnitud materjaalse tootega, tegevus toimub kliendi



tarnitud mittematerjaalse tootega, mittematerjaalse toote kohaletoimetamine, kliendile õhkkonna loomine. ABC4HEDA mudelis on teenus määratletud kui *Service* (Tabel 2).

Tabel 2. Teenusega seotud terminid.

ISO 13940 termin (ingl)	Lõputöös kasutatud ISO 13940 termini vaste (ee)	ABC4HEDA arhetüüp
Service	Teenus	Service
Healthcare service	Tervishoiuteenus	Service
Healthcare treatment	Ravi	Service
Appointment	Vastuvõtt	Service
Healthcare investigation	Terviseuuring	Service
Self-Care activity	Määratud enesehooldus	Service
Healthcare provider activity	Tervishoiuteenuse osutaja tegevus	Service

### 3.1.1 Tervishoid teenuste protsessi kirjeldusena

Tervishoiuteenust (*Service*) osutatakse tervishoiu ja kliiniliste protsesside tegevuste kaudu. Kliinilised protsessid (*Process*) tagavad patsiendi seisukohalt järjepidevuse. Tervishoiu peamised tulemused on tervishoiuteenused (*Service*), mis on ISO standardis defineeritud kui kliiniliste protsesside koondtulemused. Patsient on nende tervishoiuteenuste saaja, kuna need parandavad või aitavad säilitada tema tervist ja heaolu. Kliinilised protsessid on patsiendi seisukohalt terviklikud ja hõlmavad kogu ravi (*Service*), mis on seotud terviseprobleemi lahendamisega. Näiteks võib insuldijärgsete patsientide täieliku ravikuuri määratleda kliinilise protsessina, mis hõlmab esmatasandi arstiabi, haiglaravi, taastusravi ja ravimite (*Product*) manustamist. Järgmisena kirjeldatakse üldisi raviteenuseid ja seotud tooteid ravi etappidena. Kuigi toote arhetüüp ei ole protsesside arhetüüp, on need omavahel seotud ja ilma üheta ei ole võimalik teist kirjeldada. Protsesside arhetüübid valideeris 2022 aasta juunis T. Sõerd [20]. Osapoolte, mille alla kuulub ka juba mainitud patsient, ja nende vaheliste rollide arhetüübi selgitab lahti J. Buhvestova enda bakalaureuse lõputöös [24].

Inimene täheldab endal haiguse sümptomid ja pöördub terviseasure (BodyMetrics) ehk kaebusega tervishoiuteenuse osutaja (*PartyRoleType*) poole. Selline pöördumine on ISO standardis defineeritud ravitaotlusena (*ServiceType*) ehk täpsemalt kirjeldatud olukorras esmase tervishoiukontakti taotlus (*ServiceType*). Taotluste ja volituste terminid on välja toodud Tabel 3.

Tabel 3. Tervishoiuteenustega seotud mõisted.

ISO 13940 termin (ingl)	Lõputöös kasutatud ISO 13940 termini vaste (ee)	ABC4HEDA arhetüüp
Demand for care	Ravitaotlus	ServiceType
Demand for initial contact	Esmase tervishoiukontakti taotlus	ServiceType
Demand mandate	Nõudemandaat	ServiceType
Care period mandate	Ravivolitus	ServiceType

Tuleb ära mainida, et kõikidel inimestel on nõudemandaat (*ServiceType*), mis on seaduslikult [25] tulenev õigus ja kohustus nõuda tervishoiu tegevust. Seejärel toimub arsti vastuvõtt (*Service*). Vastuvõtt pakub tervishoiuteenust (*Service*), mis on Eesti seadusandluses [25] määratletud kui tervishoiutöötaja tegevus haiguse, vigastuse või mürgituse ennetamiseks, diagnoosimiseks ja ravimiseks eesmärgiga leevendada inimese vaevusi, hoida ära tema tervise seisundi halvenemist või haiguse ägenemist ning taastada tervist.

Arsti vastuvõtu (*Service*) ajal hinnatakse tervishoiutegevuste vajadust, mille eesmärgiks on patsiendi terviseseisundite (*BodyMetrics*) väljaselgitamine. Vastuvõtu ajal hinnatakse patsiendi terviseolekut, tervisevajadust ning terviseprobleeme. See kõik tehakse terviseuuringute (*Service*) kaudu. Vajaminevate tervishoiutegevuste hindamine, mille kohta teevad järeldusi tervishoiu töötajad, põhineb ravi vajava inimese tervishoiuteenuse pakkujate poolt täheldatud (*BodyMetrics*) ja võimalikul seisundil (*BodyMetrics*). Selleks, et raviprotsessi alustada on vaja ravivolitust (*ServiceType*). Ravivolitus on ISO standardi definitsiooni järgi patsiendi ja tervishoiuteenuse osutaja vaheline kokkulepe või leping, mille järgi osutatakse spetsiifilisi tervishoiuteenuseid volitustega kaetud raviperioodi ajal.

Kui vajalikud terviseuuringud (*Service*) on tuvastatud ja planeeritud, sisestatakse need raviplaani (*Process*). Pärast terviseuuringute määramist ning patsiendi ja tervishoiutöötaja vahelist dialoogi, kinnitatakse raviplaani. Raviplaani sisaldab kavandatud terviseuuringuid. Vastutus nende järjepidevuse aspektide kooskõlastamise eest tuleks dokumenteerida (*ServiceType*), et tagada ravi järjepidevus.

Järgmine samm on planeeritud uuringute (*ServiceType*) läbiviimine. Seejärel tervishoiutöötajad analüüsivad terviseuuringute käigus saadud tulemusi ning teevad järeldusi saavutatud seisundite (*BodyMetrics*) kohta. Nii uuringute läbiviimine kui ka analüüsimine toimub meditsiiniseadmetega (*Product*) või automaatsete meditsiiniseadmetega (*Product*). Tervishoiu teenuse pakkujad otsustavad analüüsi tulemuste põhjal, kas tervise seisund vajab ravi (*Service*) või mitte ehk ISO standardi definitsiooni kohaselt kas need vastavad teadmispõhiste meditsiiniliste näidustuste kriteeriumidele.

Pärast uuringuid tuvastatud seisundite põhjal hinnatakse laiemalt vajalikke tervishoiutegevusi ravitoimingute teostamiseks ja vajadusel toimub ka raviplaani ümberhindamine. Edasiste ravitoimingute hindamiseks (*Process*) tuleb üle vaadata ja arvestada kõiki patsiendiga seotud asjaolusid. Selle kaudu saab tervikliku ülevaate vajalikest ravinäidustustest ja edasistest ravi protseduuridest. Edasiste tervishoiutoimingute hinnang valmib konsulteerides patsiendi ja teiste tervishoiutöötajatega. Ravivajaduste hindamisel arvestatakse tervise seisunditega ja terviseprobleemidega (*BodyMetrics*) ning seda esitatakse terviseuuringute lõpus diagnoosina (*BodyMetrics*). Tuleb arvestada ka muid patsiendi tervisevajadustega seotud asjaolusid, näiteks varasemate terviseuuringute või raviprotsesside käigus avastatud allergiaid. Ravitoimingute hindamistulemusena võib juhtuda, et tuvastatud tervise seisund (*BodyMetrics*) määratakse ravi mittevajavaks. Seejärel konsulteeritakse patsiendiga ning hinnatakse juba tehtud tervishoiutegevusi edasiste tervisetöimingute sooritamiseks.

Sellele järgneb plaaniline ravi (*Process*). Mõningaid raviprotseduure viib läbi patsient, selleks on näiteks enesehooldus või tervishoiutöötaja poolt määratud enesehooldus, kuid enamiku tegevustest sooritab tervishoiutöötaja. Ravi järjepidevuse korraldamise volitus vaadatakse üle iga kord, kui raviplaani muudetakse.

Pärast kõikide ravi tegevuste lõpetamist hinnatakse kliinilise protsessi tulemust ja selle käigus avastatud probleeme, mis võiv mõjutada patsiendi tervist. Seejärel analüüsitakse tulemusi. Tulemuste analüüsimiseks võrreldakse saavutatud seisundeid sihtseisundiga (Tabel 4). Kui sihtseisund (*BodyMetrics*) on saavutatud, loetakse tervisevajadus rahuldatuks. Kui kõik raviprotsessides osalevad tervishoiutöötajad ja patsient on nõus, võib kliinilise protsessi lõpetada ning edasisi tervishoiutegevusi pole vaja. Kui sihtseisund ei ole saavutatud, suunatakse patsient uutele terviseuuringutele ja mõeldakse esialgne diagnoos ümber ning vajadusel alustatakse raviprotsessi uuesti.

Tabel 4. Seisunditega seotud mõisted.

ISO 13940 termin (ingl)	Toote arhetüübiga seotud ISO 13940 termini vaste (ee)	ABC4HEDA arhetüüp
Diagnosis	Diagnoos	BodyMetrics
Health problem	Terviseprobleem	BodyMetrics
Healthcare matter	Terviseküsimus	BodyMetrics
Health issue	Tervisemure	BodyMetrics
Healthcare condition	Terviseseisund	BodyMetrics
Observed condition	Täheldatud seisund	BodyMetrics
Possible condition	Võimalik seisund	BodyMetrics
Resultant condition	Saavutatud seisund	BodyMetrics

### 3.2 Testimine ja testimise muustrid

Siin töös kirjeldatakse detailselt rakenduse editoride testimist ja selle mustreid. Joonis on näidatud, kuidas toimub faili *EditCatalogEntries* testimine. Testklass (Joonis 4) võtab sisse vajaliku faili (*EditCatalogEntries*) ja selle faili vaate (*CatalogEntryView*), kus on kirjas kõik faili alla kuuluvad muutujad. Seejärel igas meetodis testitakse ühte muutujat.

```

[TestClass] public class EditCatalogEntriesTests :
    Tests4EditNamed<EditCatalogEntries, CatalogEntryView> {

    [TestMethod] public void EditCatalogIdTest() => edSelectTest(x => x.CatalogId);

    [TestMethod] public void EditCategoryIdTest() => edSelectTest(x => x.CategoryId);

    [TestMethod] public void EditCatalogNameTest() => labelEdStrTest(x => x.CatalogName, true, false);

    [TestMethod] public void EditCategoryNameTest() => labelEdStrTest(x => x.CategoryName, true, false);

}

```

Joonis 4. Näide editoride testklassist EditCatalogEntries näitel.

Koodi korduse vähendamiseks on tehtud eraldi meetodid (Joonis 5), et testida erinevaid tüüpi muutujaid. Näiteks *EditCatalogIdTest* testib *edSelectTest* kaudu, kas rakenduses on muutuja muutmine võimalik rippmenüüga või mitte.

```

protected void edSelectTest(Expression<Func<TItem, string>> ex, bool exists = true, string label = null)
    => labelEdSelectTest(ex, exists, exists, label);

2 references
protected void labelEdSelectTest(Expression<Func<TItem, string>> ex, bool labelExists = true, bool editorExists = true, string label = null) {
    hasLabel(ex, labelExists, label);
    var s = ex.Compile().Invoke(item);
    isEditForeignKey(s, editorExists);
}

```

Joonis 5. Näide editoride testimise sisse võetavatest meetoditest.

Testimise puhul ei ole kasutatud Test Driven Development (TDD) [11] põhimõtet. See on tarkvaraarendamise meetod, mille puhul kirjutatakse test enne koodi, et aidata tagada koodi funktsionaalsuse nõuetele vastavus ja aitab vältida võimalike vigade esinemist. Projekti vältel kirjutati kood enne ja kohe selle järel kirjutati valmis test. Küll aga on jälgitud testide kirjutamise puhul R.C.Martini [9] puhta koodi printsiipe, on vähendatud kordavat koodi ja kõik testid on kirjutatud KISS (Keep it Simple, Stupid) printsiibi järgi ehk võimalikult lihtsalt ja lühidalt. Kokku on rakenduses 14269 testi. Testitud on ka kõik rakenduses olevad kontrollid (*Controller*), millest täpsemalt kirjutab J. Buhvestova enda bakalaureuse töös [24].

## 4 Analüüs ja järeldused

### 4.1 Toote SPA raamistiku analüüs

SPA (*Single Page Application*) [8] ja MVC (*Model-View-Controller*) [7] on mõlemad veebirakenduste disainimustrid, kuid neil on omavahel palju erinevusi. SPA on veebirakendus, mis kirjutab veebilehe dünaamiliselt ümber veebiserverist pärinevate uute andmetega ehk kogu funktsionaalsus on koondatud ühte lehte. SPA eelised on parem kasutajakogemus, kuna lehe uuestilaadimine ei ole vajalik. Ka serveril on väiksem koormus, kuna server peab sisse laadima ainult algse lehe ja API päringute tulemused erinevate andmete kuvamiseks. SPA puudusteks on väiksem skaleeritavus kuna kogu rakenduse loogika peab olema ühel lehel ning SPA võib olla raskemini hooldatav kuna kogu rakenduse loogika on ühes kohas.

MVC on tarkvara disainimuster. See jagab rakenduse kolmeks peamiseks kihiks, milleks on mudel, vaade ja kontrollor. Mudel esindab rakenduse andmeid ja äri loogikat, vaade kuvab andmeid kasutajale ja kontrollor suunab kasutaja sisendi mudeli ja vaate vahel. MVC eeliseks on kihtide eraldatus, mille tulemusel on MVC lihtsamini hooldatav ning annab arendajale võimaluse keskenduda konkreetsele rakenduse osale. Samuti on tänu kihtide eraldatusele kergem teostada üksikute osade testimist. MVC puudusteks on potentsiaalselt suurem serveri koormus, kuna server peab töötlemas rohkem päringuid. Samuti on MVC lehekülgede sisselaadimine aeglasem kuna iga lehekülje jaoks on vaja laadida erinevad komponendid.

Rakenduse puhul eelistati SPA kasutamist, kuna see võimaldab kasutajasõbralikumat kasutajaliidest. Samuti võimaldab SPA kiiremat sisselaadimist ja väiksemat koormust serverile, kuna leht laetakse ainult ühe korra sisse. ABC4HEDA on suuremahuline ning MVC kasutusele võtmine teeks seda veelgi aeglasemaks.

## 4.2 Toote disainimustrid

Blazor [16] on Microsofti poolt välja töötatud raamistik, mille abil saab luua interaktiivseid veebirakendusi, kasutades .NET-i. Blazori eelised on tõhusus, kuna Blazor vähendab serveri koormust, sest enamus töötlemist toimub veebibrauseris, mis teeb veebirakenduse kiiremaks ja sujuvamaks. Samuti on Blazoris sisse ehitatud turvalisuse funktsioonid, mis aitab rakendust kaitsta võimalike turvariskide eest. Blazor sisaldab autentimis- ja autoriseerimisfunktsioone, mis võimaldavad arendajatel hõlpsasti rakendada turvameetmeid, nagu näiteks paroolide haldamine. Samuti võimaldab Blazor arendajatel kasutada serveripoolseid turvameetmeid, mis aitavad kaitsta rakendust mitmesuguste turvariskide eest. Serveripoolsed turvameetmed tagavad, et kogu loogika ja andmetöötlus toimuvad serveris, mitte kliendi poolel.

Suur pluss Blazori puhul on HTML keele kombineerimine C# programmeerimiskeelega. See võimaldab vahetada samas failis HTML ja C# vahel. Sobiv näide (Joonis 6) on ABC4HEDA rakenduses *CatalogedProducts.Client*, kus on selgelt näha, et faili algus on HTML versioonis ning poole pealt läheb üle C# programmeerimiskeelele.

```
@page "/catalogedProducts"

@using Abc.Facade.Products
@using Abc.Razor.Products

@inherits MasterDetail<CatalogedProductView>

@attribute [Authorize]

@{
    base.BuildRenderTree(__builder);
}

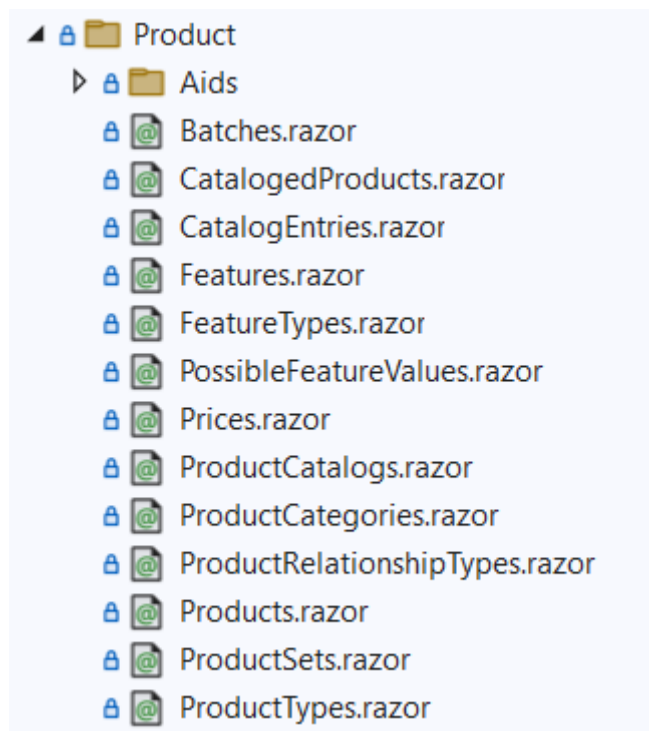
@code {
    protected override string caption => "Cataloged products";
    protected override Type editDlg => typeof(EditCatalogedProducts);

    protected override string[] columns => new string[] {
        nameof(CatalogedProductView.Name),
        nameof(CatalogedProductView.Code),
        nameof(CatalogedProductView.CatalogEntryName),
        nameof(CatalogedProductView.ProductTypeName),
    };
};
```

Joonis 6. Üleminek HTML ja C# vahel.

Blazori puudusteks on suure rakenduse puhul aeglane laadimiskiirus. Seda on näha ka ABC4HEDA rakenduse puhul, kuna esmakordne sisselaadimine on väga aeglane. Blazor on ka uus tehnoloogia, mis tähendab, et sellel on piiratud dokumentatsioon.

Koodi kirjutamisel on lähtutud puhta koodi põhimõttest, on vähendatud koodi kordusi. Klassidele ja meetoditele on pandud selgesti ja üheselt mõistetavad nimed (Joonis 7). Meetodid on hoitud lühikesed ja konkreetsed, tehes ainult seda, milleks nad mõeldud on.



Joonis 7. Klasside nimetuste näidis.

On järgitud ka Liskovi [9] printsiipi. Liskovi printsiip on tarkvaratehnika põhimõte, mille kohaselt ülemklassidelt päritud alamklassid ei tohi muuta ülemklasside käitumist. Kõige tüüpilisemaks näiteks infotehnoloogias tuuakse Liskovi printsiibi kohta risküliku ja ruudu näidet. Kõik „Ruudud“ kuuluvad alamklassi „Riskülik“. kui "Ruudu" klassile lisatakse piirang, et kõik selle küljed peavad olema võrdsed, siis see rikub Liskovi printsiipi, kuna "Riskülik" klass ei saa olla asendatud "Ruudu" klassiga ilma programmi korrektsust mõjutamata. ABC4HEDA rakenduses kasutatakse seda näiteks toote programmeerimisel. Programmeeritakse üldine klass „Toode“, mille alamklassid pärivad toote klassis kirjeldatud omadusi. Alamklassid ei tohi neid omadusi mitte mingil juhul üle kirjutada, kuna muidu tekib rakendusse viga.



Samuti on järgitud ka avatud-suletud (*open-close*) [9]printsipi. Põhimõtte sätestab, et tarkvarakomponendid peaksid olema avatud laiendamiseks ja suletud muutmiseks. See tähendab, et kui tarkvarakomponent on loodud, peaks seda olema võimalik laiendada, lisades uusi funktsioone või võimalusi, ilma et see mõjutaks olemasolevat koodi. Samuti peaks tarkvarakomponent olema suletud muutmiseks, mis tähendab, et kui komponendi käitumise muutmine on vajalik, siis peaks see toimuma nii, et olemasolev kood ei muutuks.

### **4.3 Toote sobivus valdkonnapõhiseks spetsifitseerimiseks**

Toote arhetüüp muster on tugevalt seotud nii protsesside kui ka osapoolte arhetüüpidega. Sellest tulenevalt analüüsitakse järgnevalt Tabel Sõerdi 2022. aasta bakalaureusetööd [6] ja Jane-Ly Buhvestova 2023. aasta bakalaureuse tööd [24].

T. Sõerdi bakalaureuse töö keskendub protsesside arhetüübi valideerimisele, et viia vastavusse ISO 13940:2015 standardi inglise keelne versioon ja ABC4HEDA. Kuigi protsesside ja toote arhetüübid on omavahel tugevalt seotud, on näha selgeid erinevusi terminite (Tabel 5) lahtimõtestamises. Esimene erinevus tuleb sisse mõistes ravitegevuse planeerimine, mille ISO standardi järgne tähendus on tervishoiutegevuse juhtimise osa, mil toimub raviplaani koostamine või muutmine. T. Sõerd on käsitlenud seda kui protsessi osa, aga siin töös kui toote arhetüübi alla kuuluvat teenuse osa, kuna ISO standardis on välja toodud teenuse definitsiooni kohaselt kuulub teenuse alla ka mittemateriaalse tootega seotud tegevused ning ka õhkkonna loomine. Sellel pärast võib väita, et ravitegevuse planeerimine on mittemateriaalse tootega ümberkäimine kliendi ehk patsiendi tervise parandamiseks ja edasiste raviteenuste teostamiseks.

Samuti on kirjeldatud T. Sõerdi töös perearsti vastuvõttu, terviseuuringuid ja ambulatoorset vastuvõttu ehk kõiki tervishoiutegevusi kui tervishoiu alamprotsesse. Kui viia need mõisted Toote arhetüübiga vastavusse, siis on kõikide eelmainitud terminite puhul tervishoiu osapoolte pakutud teenustega. Samuti ka tervishoiutegevuste kogum on protsesside arhetüübis märgitud kui *TaskType*, aga toote arhetüübis oleks selle vaste *ProductCatalog*.

Tabel 5. Võrdlus T.Sõerdi lõputööga.

ISO standardi mõiste	Arhetüüp valideeritud mudel järgi	T.Sõerdi protsessi	Arhetüüp valideeritud toote mudeli järgi
Ravitegevuse planeerimine	Process		Service
Vastuvõtt	ActionType		Service
Terviseuuring	ActionType		Service
Tervishoiutegevus	Action		Service
Tervishoiutegevuste kogum	TaskType		ProductCatalog

J.Buhvestova lõputöö [24] kirjeldab osapoolte ja osapoolte rollidega seotud arhetüüpe ja valideerib nende seost ISO 13490:2015 standardi ja ABC4HEDA meta-mudeli vahel. Osapoolte arhetüüp ja toodete arhetüüp peaks olema ABC4HEDA alusmudeli järgi küll seotud, kuid praegusel momendil pole seda veel tehtud. Kui osapool tarbib mingit teenust, siis nende osapoolte ja teenuse vaheliseks siduvuseks oleks näiteks arve. J. Buhvestova töö defineerib mõisteid (Tabel 6) teistmoodi kui käesolev bakalaureuse töö. Esimeseks erinevuseks on nõudemandaadi ja ravivolituse mõiste, mida J. Buhvestova on käsitlenud kui *PartyRelationshipType* ja siin töös on käsitletud kui *ServiceType*. Teiseks erinevuseks on vastuvõtt, mida siin töös on käsitletud kui *Service*, aga J. Buhvestova töös kui *PartyRoleType*. Ka automaatne meditsiiniseade on viinud mõlemad tööd lahkarvamusteni. ISO standard väljendab automaatset meditsiiniseadet kui toodet, mida kuni 2007 aastani käsitleti osapoolte alla kuuluvana.

Tabel 6. Võrdlus J.Buhvestova lõputööga.

ISO standardi mõiste	Arhetüüp J.Buhvestova valideeritud osapoolte ja osapoolte rollide mudeli järgi	Arhetüüp valideeritud toote mudeli järgi
Nõudemandaat	PartyRelationshipType	ServiceType
Automaatne meditsiiniseade	Party	Product
Ravivolitus	PartyRelationshipType	ServiceType

ISO standardi mõiste	Arhetüüp J.Buhvestova valideeritud osapoolte ja osapoole rollide mudeli järgi	Arhetüüp valideeritud toote mudeli järgi
Vastuvõtt	PartyRoleType	Service

Mõistete valideerimise erinevus tuleb sisse standardi käsitlusest erinevate mudelitega. Ükski kolmest mudelist ei ole õige ega vale. Järgmise sammuna esitatakse valideeritud mudelid raviteenuse osutajatele ning küsitakse nende tagasisidet, et teada saada, millist valideeritud mudelit nad eelistavad.

#### 4.4 Toote kajastamine erinevates publikatsioonides

Toote arhetüüpi on kajastatud ka teistes publikatsioonides. „*A requirements data model for product service systems*“ [26] artikkel käsitleb toote arhetüübis olevat toodet kui füüsilisi ja tehnoloogilisi elemente ning teenuse elemente, mis on omavahel seotud klientide probleemide lahendamiseks. Praktikas on enamuse toote ja teeninduskomponente välja töötatud üksteise suhtes iseseisvalt, mis põhjustab koordineerimisprobleeme arendustegevuses ja klientidele kohale toimetamises. Selle probleemi eemaldamiseks tuleb kasutada artikli kohaselt andmete nõuete vastavusmudelit (Requirements data model) toote arhetüübi jaoks, mis kirjeldab erinevat tüüpi nõudeid ja seoseid nende vahel. Mudel sisaldab nelja tasandit: kontekstitase, süsteemitase, teenuste tase ja komponentide tase. Kontekstitase kirjeldab toote-teenuse süsteemi üldist konteksti, süsteemitase kirjeldab süsteemi struktuuri ja funktsionaalsust, teenuste tase kirjeldab teenuste funktsionaalsust ja komponentide tase kirjeldab süsteemi komponentide omadusi ja suhteid. Andmemudel võimaldab erinevatel sidusrühmadel jagada ja hallata toote ja teenuse süsteeminõudeid, mis võib aidata kaasa eelmainitud süsteemi tõhusamale kavandamisele ja arendamisele.

Artikkel toob välja ka mõningaid väljakutseid, mis võivad kaasneda toote ja teenuse süsteemide nõuete analüüsiga, sealhulgas erinevate sidusrühmade huvide ja vajaduste tasakaalustamine ning erinevate tehniliste ja mitte-tehniliste nõuete integreerimine.

Teises allikas ehk artiklis „*Data model for quality in product lifecycle*“ [27] on kirjeldatud lähenemisviisi, mis kasutab andmemudeleid toote elutsükli kvaliteedi tagamiseks. See

muldel sisaldab kogu toote elutsüklit alustades toote kavandamisest kuni selle kasutamiseni. Mudel hõlmab mitmeid etappe, mille hulgas on nõuete kogumine, disaini loomine, testimine ja toote käitamine ning võimaldab andmete kogumist ning analüüsi, See aitab tagada toote kvaliteedi ja tulevikus parendada toodete kavandamist ning arendust. Andmemudeli kasutamine aitab kaasa toote eluea pikendamisele, kliendi rahulolule ja kulude vähendamisele.

#### **4.5 Kitsendused ja piirangud töös**

Praegusel momendil on kaks olulist kitsendust. Esimene on see, et meta-mudeli koode ei saa siduda teiste koodidega. See on vajalik, kuna meditsiini terminites on kasutusel mitmeid erinevaid süsteeme ja taksonoomiaid, üheks süsteemiks on näiteks LOINC [4]. Samuti oleks vaja kõikidele klassifikaatoritele ühendada rahvusvahelised koodid, kuna inimesed reisivad palju ning kui välismaal peaks õnnetus või terviserike juhtuma, siis on vaja Eestist inimeste terviseandmed lihtsalt ja kiiresti kätte saada.

Teine puudujääk on see, et ei saa vaadata programmi detailivaates erinevate toodete detaile. Näiteks ei saa vaadata vereproovi omadusi, mida on vaja, et kindlaks teha patsiendi veresuhkru sisaldus või verelaboris oleva kindla veretüübi kogust. Alloleval joonisel (Joonis 8) näidatakse, et on olemas universaalselt lahendatud *Master detail*, kuhu kuulub põhivaade ja detailivaade, aga vaja oleks ka vähemalt ühte lisatabelit. Detaili tabelis peaks eraldi olema veel lisaks *Master* ja *Detail*, et näha toote tüübi omadusi. Teisisõnu praegu ei ole näha rohkem kui ühte hierarhiat master detailis ehk näidatakse toodet ja tema omadusi, aga mitte omadusega kaasnevat võimalikke väärtuseid.

## Product types

Name	Code	Product type
Sample		Product
Colorectal	COLO	Package
Neuro-oncology	NONC	Package
Tumour in renal vein	TVEIN	Product
Other block	O-BLOCK	Product
PBMC	PB	Product
Test		Service
FFPE block	FFPE	Product
Culture block	CULTURE	Product
RNA block	RNA	Product
Pleural fluid(un)	PFu	Product
Urine-Pl(un)	U-Plu	Product

### ProductType details

Code:

Name:

Product type:

Is base type:

Alternative codes:

Base type:

Details:

Valid from:

Valid to:

Joonis 8. Kitsendus MasterDetail näite põhjal.

## 4.6 Edasised tööd

Järgmise etapp on esitada valideeritud mudel tervishoiutöötajatele ning küsitakse nende arvamust ja tagasisidet mudeli kohta. Samuti on rakenduses vaja kõikidele klassifikaatoritele ühendada rahvusvahelised koodid ja millise klassifikaatoritega milline kood kehtib. See on vajalik, kuna inimesed reisivad üha rohkem ning mõne õnnetuse või terviserikke juhtudes, on ka välismaal olles on vaja raviteenuseid. Selleks on kõikidel raviasutustel vaja inimeste terviseandmetele ligipääsu. Ligipääsu lihtsustamiseks peavad raviandmed vastama rahvusvahelistele standarditele. Samuti tuleks jätkata ABC4HEDA rakenduse edasiarendamist.

## **5 Kokkuvõte**

Meditsiin muutub järjest enam patsiendikeskemaks ning seetõttu pööratakse suuremat tähelepanu ravi järjepidevuse tagamisele. Selleks on vaja meditsiinis heterogeensete infosüsteemide ehk koostalitlusvõimeliste ja standardiseeritud IT lahenduste kasutamist. Töö tulemusena valideeriti toote meta-mudelit sidudes omavahel ISO 13940:2015 standardis olevaid üldiseid mõisteid, mida kasutatakse arstiabiandmisel ja raviteenuste puhul, ABC4HEDA alus-mudeliga. ABC4HEDA hoiab kõik oma andmed arhetüüpide ja arhetüüpustrite põhjal loodud meta-mudelis, mis hõlbustab järjepidevust ning eraldab meditsiiniteadmised andmetest, kus kõik üksused vastavad konkreetse üksust kirjeldavale tüübile. Tulemused näitavad, et meditsiinilisi tooteid ja teenuseid on võimalik arhetüüpustrites vastavalt ISO 13940 standardile spetsifitseerida.

## 6 Kasutatud kirjandus

- [1] ISO, Health informatics — System of concepts to support continuity of care, 2015.
- [2] TERVISEINFORMAATIKA Mõistesüsteem tervishoiu ja arstiabi järjepidevuse, Tallinn, 2023.
- [3] G. Piho, Archetypes based techniques for development of domains, Requirements and Software. Towards LIMS Software Factory, Tallinn: Tallinn University of Technology Press, 2011.
- [4] K. M. Raavel, LOINC terminoloogia kasutusele võtmine arhetüüpustritel põhineval ABC4HEDA alusmudelil, Tallinn: Tallinn University of Technology Press, 2022.
- [5] R. Randmaa, Äriarhetüüpidel põhinevate metamudelite ja FHIR ressursside semantilise koostalitluse hindamine, Tallinn: Tallinn University of Technology Press, 2022.
- [6] T.Sõerd, Meditsiiniliste protsesside spetsifitseerimine arhetüüpustrites vastavalt rahvusvahelistele standarditele ja kokkulepetele, Tallinn: Tallinn University of Technology Press, 2022.
- [7] Microsoft, „ASP.NET MVC,“ 19 2 2020. [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/aspnet/mvc/>. [Kasutatud 16 5 2023].
- [8] Microsoft, „Overview of Single Page Applications (SPA) in ASP.NET Core,“ 14 2 2023. [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/aspnet/core/client-side/spa/intro?view=aspnetcore-7.0>. [Kasutatud 16 5 2023].
- [9] R. C. Martin, Clean code.
- [10] K. Beck, Extreme Programming Explained: Embrace Change, Addison- Wesley, 1999.
- [11] K. Beck, Test Driven Development: By Example, Addison-Wesley Professional, 2002.

- [12] O. Ltd., „A system of concepts for the continuity of care,“ [Võrgumaterjal]. Available: <https://contsys.org/page/default>. [Kasutatud 16 5 2023].
- [13] Microsoft, „Visual Studio Community,“ [Võrgumaterjal]. Available: <https://visualstudio.microsoft.com/vs/community/>. [Kasutatud 15 5 2023].
- [14] Microsoft, „What's new in C# 11,“ [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-11>. [Kasutatud 15 5 2023].
- [15] Microsoft, „What's new in ASP.NET Core 7.0,“ [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/aspnet/core/release-notes/aspnetcore-7.0?view=aspnetcore-7.0>. [Kasutatud 15 5 2023].
- [16] Microsoft, „ASP.NET Core Blazor,“ 24 2 2023. [Võrgumaterjal]. Available: [https://learn.microsoft.com/en-gb/aspnet/core/blazor/?WT.mc\\_id=dotnet-35129-website&view=aspnetcore-7.0](https://learn.microsoft.com/en-gb/aspnet/core/blazor/?WT.mc_id=dotnet-35129-website&view=aspnetcore-7.0). [Kasutatud 16 5 2023].
- [17] Radzen, „Radzen Blazor Components,“ [Võrgumaterjal]. Available: <https://blazor.radzen.com/>. [Kasutatud 14 5 2023].
- [18] R. H. Dolin, „The HL7 Clinical Document Architecture,“ American Medical Informatics Association, 2001.
- [19] J. Arlow ja I. Neustadt, Enterprise Patterns and MDA: Building Better Software, 2003.
- [20] J. A. Zachman, „The Zachman Framework,“ 2008. [Võrgumaterjal]. Available: <https://www.zachman.com/about-the-zachman-framework>. [Kasutatud 15 5 2023].
- [21] Microsoft, „Entity Framework Core,“ 2021. [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/ef/core/>. [Kasutatud 15 5 2023].
- [22] Microsoft, „Microsoft SQL dokumentatsioon,“ 2022. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sql/?view=sql-server-ver15>. [Kasutatud 15 5 2023].
- [23] T. Sõerd, Meditsiiniliste protsesside spetsifitseerimine arhetüüpmustrites vastavalt ISO 13940, Tallinn, 2022.
- [24] J.-L. Buhvestova, Osapoole ja Osapoole Suhte arhetüüpmustrite realiseerimine, testimine ja sobivuse hindamine ISO 13940:2015 (ContSys) mõistete deklaratiivseks spetsifitseerimiseks, Tallinn: Unpublished, 2023.



- [25] Tervishoiuteenuste korraldamise seadus (lühend - TTKS), 2001.
- [26] M. Berkovich, J. M. Leimeister, A. Hoffmann ja H. Krcmar, A requirements data model for product service systems, 2012.
- [27] X. Tang ja Y. Hu, „Data model for quality in product lifecycle,“ *Computers in Industry*, kd. 59, nr 2-3, pp. 167-179, 2008.

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

Mina, Liisa Kalling

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Toote arhetüüp mustri realiseerimine, testimine ja vastavuse hindamine ISO 13940:2015 (ContSys) põhjal mõistete deklaratiivseks spetsifitseerimiseks“, mille juhendajad on Toomas Klementi 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.

17.05.2023

---

<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.