



Vaasan yliopisto  
UNIVERSITY OF VAASA

Pia West

## **Käyttöohjeiden hallinnan haasteet**

Ratkaisuna rakenteisuus, modulaarisuus  
ja yksilähteistäminen?

Markkinoinnin ja viestinnän akateeminen yksikkö  
Teknisen viestinnän pro gradu -tutkielma  
Teknisen viestinnän maisteriohjelma

Vaasa 2023

---

**VAASAN YLIOPISTO****Markkinoinnin ja viestinnän akateeminen yksikkö**

<b>Tekijä:</b>	Pia West		
<b>Tutkielman nimi:</b>	Käyttöohjeiden hallinnan haasteet: ratkaisuna rakenteisuus, modulaarisuus ja yksilähteistäminen?		
<b>Tutkinto:</b>	Filosofian maisteri		
<b>Oppiaine:</b>	Tekninen viestintä		
<b>Työn ohjaaja:</b>	Niina Nissilä		
<b>Valmistumisvuosi:</b>	2023	<b>Sivumäärä:</b>	79

---

**ABSTRAKT:**

Processerna i arbetslivet har digitaliserats, och mängden programvaror som används ökar kontinuerligt, liksom deras komplexitet. Som stöd för användningen och utnyttjandet av tillämpningarna behövs oftast användarmanualer. Att producera manualerna som linjär dokumentation med hjälp av textbehandlingsprogram motsvarar inte de krav som ställs på effektiv hantering av användarmanualerna för en omfattande produktportfölj. Strukturerad dokumentation, modularitet och återanvändning av innehåll erbjuder i teorin lösningar för hantering av dokumentationen, men hur bra kan lösningen förverkligas i de praktiska verktygen, det vill säga i innehållshanteringssystemen?

Målet med denna undersökning var att ta reda på hur strukturerad dokumentation, modularitet och återanvändning av innehåll löser de utmaningar som finns med att producera och upprätthålla användarmanualer för en komplex programvaruportfölj. Undersökningen var tvådelad. Den första delen utfördes som en expertbedömning, där egenskaperna hos tre innehållshanteringssystem granskades för att ta reda på om de verktyg som systemen tillhandahåller stöder principerna för strukturerad dokumentation, modularitet och återanvändning av innehåll. Undersökningen visade att de system som jämfördes stöder dessa principer väl. I undersökningens andra del testades ett av de tre jämförda komponentinnehållshanteringssystemen i praktiken, med syfte att hitta en effektiv hanteringsmodell för manualerna. Lösningar söktes specifikt för fem identifierade huvudfaktorer som orsakar utmaningar: olika målgruppers informationsbehov, situationsspecifika informationsbehov, korrelation mellan tillämpningarna, olika versioner av tillämpningarna samt manualernas språkversioner. Testen utfördes genom att först omvandla de linjära användarmanualerna för två tillämpningar till modulär form, och sedan förverkliga dem i innehållshanteringssystemet, genom att utnyttja dess funktionaliteter och verktyg. På basen av denna test kunde man konstatera, att komponentinnehållshanteringssystemet på samtliga fem delområden underlättade, men gav inte fullkomliga lösningar för hanteringen av manualerna.

På basen av undersökningen kan konstateras, att innehållshanteringssystem som stöder strukturerad dokumentation, modularitet och återanvändning av innehåll kan minska utmaningarna med hanteringen av komplex dokumentation. Ett optimalt utnyttjande av systemet förutsätter också planering och helhetsorganisering av innehållet.

---

**AVAINSANAT:** ohjelmistotuotanto, tekninen dokumentaatio, käyttöohje, rakenteisuus, modulaarisuus, yksilähteistäminen, komponenttisisällönhallintajärjestelmä

---

**VAASAN YLIOPISTO****Markkinoinnin ja viestinnän akateeminen yksikkö**

<b>Tekijä:</b>	Pia West		
<b>Tutkielman nimi:</b>	Käyttöohjeiden hallinnan haasteet: ratkaisuna rakenteisuus, modulaarisuus ja yksilähteistäminen?		
<b>Tutkinto:</b>	Filosofian maisteri		
<b>Oppiaine:</b>	Tekninen viestintä		
<b>Työn ohjaaja:</b>	Niina Nissilä		
<b>Valmistumisvuosi:</b>	2023	<b>Sivumäärä:</b>	79

---

**TIIVISTELMÄ:**

Työelämän prosessit ovat digitalisoituneet ja käytettävien ohjelmistojen määrä sekä kompleksisuus kasvavat jatkuvasti. Ohjelmistojen käytön tueksi ja niiden hyödyntämisen optimoimiseksi tarvitaan useimmiten käyttöohjeita. Niiden tuottaminen lineaarisena dokumentaationa tekstinkäsittelyohjelmia käyttäen ei vastaa laajan ohjelmistoportfolion käyttöohjeiden tehokkaan hallinnoinnin asettamia vaatimuksia. Rakenteisuus, modulaarisuus ja yksilähteistäminen tarjoavat teoriassa ratkaisuja dokumentaation hallinnointiin, mutta kuinka hyvin ratkaisu on toteutettavissa käytännön työkaluissa eli sisällönhallintajärjestelmissä?

Tämän tutkielman tavoitteena oli selvittää, miten dokumentaation rakenteistaminen, modulaarisuus ja yksilähteistäminen ratkaisevat kompleksisen ohjelmistotuoteportfolion tuomia haasteita käyttöohjeiden tuottamisessa ja ylläpidossa. Analyysi oli kaksivaiheinen. Ensimmäinen osio suoritettiin asiantuntija-arviointina, jossa tarkasteltiin kolmen komponenttisisällönhallintajärjestelmän ominaisuuksia, jotta saataisiin selville, tukevatko järjestelmien tarjoamat työkalut rakenteistamisen, modulaarisuuden ja yksilähteistämisen periaatteita. Analyysi osoitti, että vertailussa olleet järjestelmät tukevat hyvin näitä periaatteita. Analyysin toisessa osiossa testattiin yhtä ensimmäisessä tutkimusosiossa tutkittua komponenttisisällönhallintajärjestelmää käytännössä, tavoitellen tehokasta käyttöohjekokonaisuuden hallintamallia. Ratkaisuja haettiin erityisesti viiteen tunnistettuun haasteita aiheuttavaan päätekijään: eri kohderyhmien tiedontarpeet, tilannekohtaiset erilaajuiset tiedontarpeet, sovellusten väliset riippuvuudet, sovellusten eri tuoteversiot sekä käyttöohjeiden eri kieliversiot. Testi suoritettiin muuntamalla ensin kahden tiedonhallintaan liittyvän sovelluksen lineaariset käyttöohjeet modulaariseen muotoon, ja toteuttamalla ne sen jälkeen sisällönhallintajärjestelmään, sen toiminnallisuuksia ja työkaluja hyödyntäen. Testin perusteella voitiin todeta, että komponenttisisällönhallintajärjestelmä tarjosi käyttöohjeiden hallintaan helpotusta, joskin ei täydellisiä ratkaisuja kaikilla viidellä osa-alueella.

Tutkimuksen perusteella voidaan todeta, että rakenteisuutta, modulaarisuutta ja yksilähteistämistä tukeva sisällönhallintajärjestelmä voi vähentää kompleksisen dokumentaation hallinnoinnin haasteita. Järjestelmän optimaalinen hyödyntäminen edellyttää myös sisällön suunnittelua ja kokonaisvaltaista organisointia.

---

**AVAINSANAT:** ohjelmistotuotanto, tekninen dokumentaatio, käyttöohje, rakenteisuus, modulaarisuus, yksilähteistäminen, komponenttisisällönhallintajärjestelmä

## Sisällys

1	Johdanto	7
1.1	Tavoite	8
1.2	Aineistonkeruu	10
1.2.1	Vertailtavien sisällönhallintajärjestelmien valinta	10
1.2.2	Käyttöohjeaineiston valinta	13
1.3	Aineiston analyysimenetelmät	16
2	Teknisen dokumentoinnin toimintaympäristön ydintekijät	20
2.1	Ohjelmistotuotannon kompleksisuus	20
2.2	Tekninen viestintä ja dokumentointi	21
2.3	Dokumentaatioon liittyvät standardit	22
2.4	Käyttöohjeiden hallinta	23
2.5	Sisällönhallintajärjestelmä	25
3	Rakenteisen dokumentaation edellytykset ja mahdollisuudet	29
3.1	Rakenteinen dokumentaatio	30
3.1.1	Informaatiomalli	32
3.1.2	Merkintäkieli	34
3.2	Modulaarisuus	35
3.3	Yksilähteistäminen	37
4	Sisällönhallintajärjestelmä käyttöohjeiden hallinnassa	39
4.1	Sisällönhallintajärjestelmien vertailu	40
4.2	Käyttöohjeiden toteuttaminen sisällönhallintajärjestelmään	46
4.2.1	Sisällön analysointi	47
4.2.2	Sisällön organisointi	51
4.2.3	Sisällön esittely	54
4.2.4	Dokumentaation versionhallinta	55
5	Sisällönhallintajärjestelmien vertailun ja testaamisen tulokset	58
5.1	Sisällönhallintajärjestelmien vertailun tulokset	58
5.2	Sisällönhallintajärjestelmän tarjoamat hyödyt	60

6	Loppupohdinnat	65
	Lähteet	69
	Liitteet	75
	Liite 1. DITA-informaatiomallin kategoriat ja niiden edellyttämät ominaisuudet	75
	Liite 2. Käytettävyydestauksen kriteeristö, Triplan Oy (2020)	77
	Liite 3. Käyttöohje eri yksityiskohtaisuustasoisena	79

## Kuvat

<b>Kuva 1.</b> Esimerkki komponenttisisällönhallintajärjestelmän käyttöliittymästä.	28
<b>Kuva 2.</b> Testiprojekti ja aiheet MadCap Flare -järjestelmässä.	44
<b>Kuva 3.</b> Testiprojekti ja aiheet ClickHelp-järjestelmässä.	44
<b>Kuva 4.</b> Testiprojekti ja aiheet Oxygen XML Editor -järjestelmässä.	45
<b>Kuva 5.</b> Yksityiskohtaisuustason merkinnät DOCX-muotoisessa käyttöohjeessa.	51
<b>Kuva 6.</b> Aihehierarkia ClickHelp-järjestelmän editorissa ja julkaisuversiossa.	53
<b>Kuva 7.</b> ClickHelp-järjestelmän lokalisointieditori.	56

## Kuviot

<b>Kuvio 1.</b> Esimerkki keskikokoisen organisaation ratkaisukokonaisuudesta.	15
<b>Kuvio 2.</b> Rakenteisen dokumentaation edellytykset ja mahdollisuudet.	30
<b>Kuvio 3.</b> Tutkimusprosessi.	39
<b>Kuvio 4.</b> Moduulityypit Amentin mukaan.	48

## Taulukot

<b>Taulukko 1.</b> Kvantitatiivisen ja kvalitatiivisen tutkimuksen erot.	17
<b>Taulukko 2.</b> Modulaarinen vs. lineaarinen dokumentaatio.	36
<b>Taulukko 3.</b> Dokumentaation yksilähteistämisen prosessin askeleet.	46
<b>Taulukko 4.</b> Sisällön yksityiskohtaisuustasot.	50
<b>Taulukko 5.</b> Sisällönhallintajärjestelmien vertailun tulokset.	59
<b>Taulukko 6.</b> Ominaisuuksien toteuttamiseen hyödynnetyt työkalut ClickHelp-järjestelmässä.	61

# 1 Johdanto

Ohjelmistotuotanto eli tietokoneohjelmien suunnittelusta, valmistuksesta ja kehityksestä koostuva liiketoiminta on jatkuvasti kasvava ala, jonka lopputuotokset harvoin, jos koskaan, ovat staattisia tuotteita. Uusia tarpeita, uusia käyttökohteita ja uusia käyttäjäkuntia syntyy koko ajan, ja pysyäkseen kilpailukykyisinä ohjelmistotuottajien täytyy niihin vastata, joko kehittämällä olemassa olevia tuotteita tai lanseeraamalla täysin uusia tuotteita (Redish, 2010, s. 191).

Työelämässä valtaosa prosesseista on siirtynyt digitaaliseen maailmaan. Tänä päivänä suuri osa käytettävistä sovelluksista ei kuitenkaan vielä ole niin intuitiivisia, että käyttäjät pystyisivät niitä käyttämään täysin ilman erillisiä ohjeistuksia (Blackler ja muut, 2018, s. 26). Mikäli sovelluksen tehokkaan käytön edellytyksiä ei jostakin syystä ole, osa sovelluksen kapasiteetista jää hyödyntämättä. Asiakasorganisaatiossa tämä voi aiheuttaa virhetilanteita ja prosessien tehottomuutta, josta seuraa resurssien turhaa kuluttamista. Ohjelmistotuottajalle se voi tarkoittaa asiakastuen ruuhkautumista ja pidemmän päälle se voi vaikuttaa negatiivisesti tuotteen maineeseen. Niin asiakasorganisaatio kuin ohjelmistotuottaja hyötyy siis siitä, että loppukäyttäjille suunnattu dokumentaatio, eli ohjeistus sovelluksen käytön tueksi on saatavilla ja ajan tasalla.

Ohjelmistojen toimitukseen sisältyy useimmiten käyttöohjeita, joiden tarkoitus on tukea käyttäjää sovelluksen käytössä. Monet ohjelmistoyritykset tuottavat käyttöohjeita lineaarisena dokumentaationa tekstinkäsittelyohjelmia käyttäen. Tällöin syntyy staattinen sisältö, joka muodostaa järkevän kokonaisuuden vain kyseisessä dokumentissa olevassa muodossa ja kokoonpanossa (Ament, 2003, s. 5). Tämä ei ole optimaalinen tapa valtavan tietomassan hallinnassa. Nykypäivän dokumentointityökalut, eli sisällönhallintajärjestelmät tarjoavat monipuolisia ominaisuuksia monitahoisen sisällön tehokkaaseen hallintaan. Sisällönhallintajärjestelmä on digitaalisen sisällön luontiin, organisointiin ja ylläpitoon

kehitetty työkalu (Shivakumar, 2017). Näihin sisällönhallintajärjestelmiin liittyy käytännön oppaita ja niiden toimittajat tarjoavat usein myös konsultointi- ja koulutuspalveluita, joiden avulla käyttöönotto helpottuu.

Viestintätieteissä dokumentointiin liittyviä menetelmiä ja teorioita käsitellään lukuisissa tutkimuksissa ja muissa teoksissa monesta eri näkökulmasta. Tutkimusaiheina ovat muun muassa standardoidut informaatiomallit (esimerkiksi Hackos, 2016), rakenteisen ja modulaarisen sisällön tuottaminen sekä yksilähteistäminen (Ament, 2003; Andersen & Batova, 2015; Bailie & Huset, 2015; Hashimoto ja muut, 2020).

Koen, että käytännön, eli ratkaisuja tarjoavien sisällönhallintajärjestelmien ja teorian, eli ratkaisuja tarjoavien menetelmien välillä on aukko. Esimerkiksi Anderson ja Batova analysoivat sisällönhallintajärjestelmiin liittyvää tutkimusta muun muassa teknisen viestinnän ja tiedonhallinnan aloilta. Artikkelissa he toteavat, että tieteellisillä julkaisuilla on ollut suhteellisen vähän vaikutusta sisällönhallintajärjestelmien käyttöön, koska vain muutama tieteellinen julkaisu on lähestynyt aihetta siitä näkökulmasta, eikä dialogia tutkijoiden ja alan toimijoiden välillä juurikaan ole (Andersen & Batova, 2015). Dokumentaationsa uudistamista harkitsevan organisaation on vaikea tietää, mitä kaikkea tulee huomioida ja vaatia, päästääkseen dokumentaationsa tehokkaaseen hallintaan. Tutkielmallani haluan täyttää tämän tiedonaukon ja selvittää, voiko käyttöohjeiden hallinnan haasteita vähentää.

## **1.1 Tavoite**

Tutkielmassani selvitän käyttöohjeissa olevan dokumentaation rakenteistamisen, modulaarisuuden ja yksilähteistämisen hyötyjä kompleksisen ohjelmistotuoteportfolion käyttöohjeiden hallinnassa. Kompleksinen ohjelmistotuoteportfolio tarkoittaa toisistaan riippuvaisista laajoista sovelluksista koostuvaa ohjelmistokokonaisuutta. Tutkielman tavoitteena on löytää ratkaisuja, joilla kompleksisen tuoteportfolion käyttöohjeiden hallinnointi helpottuu.



Tutkimus koostuu kahdesta osiosta, joista kukin osio käsittelee yhtä tutkimuskysymystä. Tutkimuskysymykset, joiden avulla pyrin tavoitteeseen ovat seuraavat:

- 1) Miten hyvin rakenteisuus, modulaarisuus ja yksilähteistäminen toteutuvat sisällönhallintajärjestelmissä a, b ja c?

Dokumentaation rakenteistaminen tarkoittaa dokumentaation tuottamista ennalta määritellyn rakenteen mukaisesti (Horn, 1998). Amentin (2003) klassikkoteos tutustuttaa modulaarisen sisällön ja yksilähteistämisen periaatteisiin. Yksilähteistämällä tarkoitetaan informaation uudelleenkäyttöä, toisaalta eri teknisissä formaateissa ja toisaalta eri kokoonpanoina (Ament, 2003, s. 15). Modulaarisuus tarkoittaa informaation jakamista pieniin osiin, jotka ovat ymmärrettäviä kontekstista irrotettuina ja toisiinsa eri tavoin yhdistettyinä (Kratky, 2017). Sisällönhallintajärjestelmä on työkalu, jolla voidaan luoda, organisoida ja ylläpitää digitaalista sisältöä (Syncro Soft, n.d.a).

- 2) Miten kompleksisen ohjelmistotuoteportfolion tuomia haasteita käyttöohjeiden hallinnassa voidaan ratkaista dokumentaation rakenteisuudella, modulaarisuudella ja yksilähteistämällä?

Ohjelmistotuotanto itsessään on hyvin kompleksista toimintaa ja sen tuotokset ovat useimmiten jatkuvasti kehitettäviä. Tällöin myös niihin liittyvät käyttöohjeet vaativat jatkuvaa päivitystä. Haasteita käyttöohjeiden hallinnassa aiheuttaa niiden monitasoisuuden tarve; laajat ohjelmistotuoteportfoliot, heterogeeniset käyttäjäkunnat, sovellusten väliset riippuvuudet, vanhojen ja uusien sovellusversioiden rinnakkaiskäyttö sekä tarve monikielisille käyttöohjeille.

Tutkimus rajataan koskemaan työelämän tiedonhallintaan liittyvien tietokoneohjelmistojen kirjallisia käyttöohjeita. Tutkimuksessa fokusoin teknisessä

dokumentoinnissa käytettäviin menetelmiin ja työkaluihin viestinnällisestä, en teknisestä näkökulmasta. Tutkimuksessa en tarkastele dokumentointityön prosesseja.

## 1.2 Aineistonkeruu

Tämä tutkimus koostui kahdesta peräkkäisestä vaiheesta ja kutakin osiota varten oli omat aineistot. Ensimmäisen osion aineisto koostui kolmesta sisällönhallintajärjestelmästä, joita on suunniteltu käytettäväksi tekniseen dokumentaatioon. Sisällönhallintajärjestelmien tarkemmat valintakriteerit ja muut aineiston valikoitumiseen vaikuttaneet tekijät on kerrottu luvussa 1.2.1.

Toisen osion aineistona oli kahden työelämässä käytettävän, tiedonhallintaan liittyvän sovelluksen DOCX-muotoiset käyttöohjeet, joiden sisältöä muunsin modulaariseen muotoon ja vein yhteen edellisessä vaiheessa tutkittuun sisällönhallintajärjestelmään. Valitsin käyttöohjeet työnantajani Triplan Oy:n tuoteportfoliosta. Valintaan vaikuttavat seikat on kuvattu tämän tutkielman luvussa 1.2.2.

### 1.2.1 Vertailtavien sisällönhallintajärjestelmien valinta

Ensimmäisessä tutkimusosiossa vertailtavien järjestelmien valinta perustui seuraaviin kriteereihin:

1. Järjestelmän on oltava tyypiltään komponenttisisällönhallintajärjestelmä eli uudelleenkäytettävään tekniseen dokumentaatioon tarkoitettu.
2. Järjestelmän on oltava tyypiltään ns. *Help Authoring Tool* eli järjestelmällä tuotettu dokumentaatio pitää pystyä julkaisemaan verkossa olevaan ohjesivustoon.
3. Järjestelmän kuvauksesta täytyy ilmetä, että se ainakin joltain osin tukee dokumentaation rakenteisuutta.

4. Järjestelmän kuvauksesta täytyy ilmetä, että se ainakin joltain osin tukee modulaarisuutta ja yksilähteistämistä.
5. Järjestelmän kuvauksesta/suosituksista täytyy ilmetä, että se soveltuu suurien sisältömassojen hallintaan.
6. Järjestelmän kuvauksesta täytyy ilmetä, että se tukee monikielisen sisällön tuottamisen prosesseja.
7. Järjestelmän kuvauksesta täytyy ilmetä, että se tarjoaa muitakin monipuolisia toiminnallisuuksia sisällön hallintaan ja muotoiluun.
8. Järjestelmän käyttöliittymä ja tuki on oltava saatavilla englannin kielellä.
9. Järjestelmälle on oltava luotettavia suosituksia.

Sisällönhallintajärjestelmiä on olemassa todella suuri määrä. Tämän tutkimuksen tavoite ei ollut soveltuvimman sisällönhallintajärjestelmän löytäminen, vaan tarkoitus oli tutkia yleisemmällä tasolla, onko sisällönhallintajärjestelmät suunniteltu rakenteisuutta, modulaarisuutta ja yksilähteistämistä silmällä pitäen. Täten ei olisi ollut tarkoituksenmukaista pyrkiä löytämään mahdollisimman kattavaa valikoimaa, vaan otanta tunnetuimmista sisällönhallintajärjestelmistä oli riittävä. Internetistä löytyy vertailusivustoja, jotka tarjoavat koostetusti tietoja sisällönhallintajärjestelmätarjonnasta ja järjestelmien ominaisuuksista. Kuten järjestelmiäkin, myös vertailusivuja löytyy runsain määrin. Eri sivustoilla vertailtavat järjestelmät ovat luonnollisesti osittain samoja, erityisesti mitä tulee tunnettuihin ja laajasti käytössä oleviin sisällönhallintajärjestelmiin. Näin ollen, melkein mikä tahansa suhteellisen laaja vertailusivusto antaa hyvin tietoja markkinoilla olevasta tarjonnasta. Yksi laaja vertailusivusto on Indoition:n sivusto (Achtelig, 2022b). Se sisältää kattavasti tietoja markkinoilla olevista sisällönhallintajärjestelmistä ja niiden ominaisuuksista, sekä myös sisällönhallintajärjestelmän valinnassa huomioitavista seikoista, joten se soveltuu hyvin suunnittelemani kartoitukseen. Indoition-sivustolla mainitaan, että sisällönhallintajärjestelmän valinnassa kannattaa huomioida, että niistä on olemassa kolme perusryhmää: sisäistä editoria käyttävät järjestelmät, ulkoista editoria käyttävät järjestelmät sekä pelkkää konversiota varten käytettävät järjestelmät (Achtelig, 2022a).

Viimeisen ryhmän sisällönhallintajärjestelmät eivät voineet tulla tässä tutkielmassa kysymykseen, koska niissä ei toteudu kaikki yllä listatut valintakriteerit. Myös ulkoista editoria käyttävät järjestelmät päätin jättää pois, koska niitä käytettäessä sisällönhallintaprosessi monimutkaistuu, jolloin sisällön tuonti sisällönhallintajärjestelmään aiheuttaa helposti virhetilanteita (Achtelig, 2022a). Päädyin siis keskittymään sisäistä editoria käytettäviin sisällönhallintajärjestelmiin. Indoition-sivuston sisällönhallintajärjestelmät on jaettu kahteentoista eri kategoriaan. Yllä lueteltujen valintakriteerien perusteella yhdeksän kategoriaa sulkeutui pois. Jäljelle jäi kolme kategoriaa:

1. Klassiset järjestelmät ohjeistusten luontiin
2. Pilvipohjaiset teknisen dokumentoinnin järjestelmät (SaaS)
3. XML-editorit teknisen dokumentaation luontiin

Maksuttoman kokeiluversion saatavuus oli tutkielmassani käytännön syistä rajoittava tekijä, joten ensimmäisenä tarkistin jokaisen osalta tämän. Tämän kriteerin perusteella osa rajoittui jo pois. Osa palveluntarjoajista tarjoaa mahdollisuuden tilata tuote-esittelyä, mutta, koska ei ollut tietoa olisiko tuote-esittelynkään jälkeen ollut mahdollisuus maksuttomaan kokeiluun, rajoitin nämäkin pois.

Tutustuin näiden kolmen ryhmän sisällönhallintajärjestelmien ominaisuuksiin niiden palveluntuottajan verkkosivuilla olevien tuotekuvausten avulla. Tieto, onko sisällönhallintajärjestelmä tyypiltään komponenttisisällönhallintajärjestelmä vai muu tyyppi, ei kaikissa ilmennyt yksiselitteisesti. Rajasin pois ne, joiden osalta en saanut täyttä varmuutta, onko kyseessä komponenttisisällönhallintajärjestelmä vai ei.

Saadakseni mahdollisimman kattavasti erityyppisiä sisällönhallintajärjestelmiä tutkittavaksi, valitsin jokaisesta ryhmästä yhden. Ensimmäisessä ryhmässä oli kaksi sisällönhallintajärjestelmää, jotka täyttivät kaikki yllä listatut kriteerit. Niistä valitsin MadCap Flare -järjestelmän (Madcap Software, 2023c) sillä perusteella, että se on monipuolisempi, koska siitä on työpöytäversion lisäksi saatavilla myös pilviversio,

tuotenimellä MadCap Central (Madcap Software, 2023b). Toisesta ryhmästä valikoitui ClickHelp-järjestelmä (ClickHelp, 2023a), sillä se oli ryhmässä ainut, josta selvästi ilmeni, että se on tyypiltään komponenttisisällönhallintajärjestelmä. Kolmannessa ryhmässä valintani osui Oxygen XML Editor -järjestelmään (Syncro Soft, 2023), koska se perustuu DITA:n tietoarkkitehtuuristandardiin (Oasis Open, 2018). DITA-standardista ja periaatteista kerron tarkemmin tämän tutkielman luvussa 3.1.1.

### **1.2.2 Käyttöohjeaineiston valinta**

Tutkimuksen toinen osio koostui testistä, jossa toteutin osaa kahden tiedonhallintaan liittyvän sovelluksen käyttöohjeita yhteen niistä sisällönhallintajärjestelmistä, joita olin vertaillut tutkimuksen ensimmäisessä osiossa. Koska oma työpaikkani Triplan Oy edustaa kompleksisen tuoteportfolion ohjelmistotuottajia, tuotteet ovat minulle entuudestaan tuttuja, ja näin oli luontevaa valita nämä kaksi sovellusta työnantajani tuoteportfoliosta. Triplanin sovellukset sisältävät laajasti toiminnallisuuksia, ja sovellusten välillä on riippuvuuksia. Tuotekehitys on jatkuvaa, ja käyttäjäkunta hyvin heterogeeninen. Kaikissa sovelluksissa on suomen- ja ruotsin kielen tuki, koska niitä käytetään sekä täysin suomenkielisisissä että täysin ruotsinkielisissä organisaatioissa ja myös kaksikielisisissä organisaatioissa. Nämä kaikki seikat vaikuttavat käyttöohjeiden suunnitteluun.

Triplan Oy on ohjelmistotalo, joka toimittaa asian-, dokumentin- ja arkistohallinnan ratkaisuja muun muassa julkishallinnon organisaatioille. Päätuotteet, jotka ovat SÄHKE2-sertifioituja, ovat Tweb asian- ja asianhallintajärjestelmä, WebArkki tiedonohjausjärjestelmä sekä Triplan Arkisto sähköinen säilytysjärjestelmä. Lisäksi tuotevalikoimaan kuuluu ratkaisuja, joilla voidaan hallinnoida muun muassa sähköisiä kokouksia, allekirjoitusratkaisuja, luottamushenkilö- ja sidonnaisuustietoja, sopimusportfolioita ja muita sähköisiä ratkaisuja ja integraatioita. Tuotteisiin liittyvät toimitusprojektien lisäksi laaja palveluverkko, joka koostuu muun muassa asiakastuki-, koulutus- ja konsultointipalveluista. Asiakaskunta on laaja ja koostuu esimerkiksi

kunnista, kaupungeista, kuntayhtymistä, ammattikorkeakouluista, yliopistoista, valtionhallinnon virastoista ja laitoksista sekä ministeriöistä (Triplan, 2023).

Käyttöohjeiden hallinnan haasteet aiheutuvat kompleksisuudesta, joka syntyy useasta eri tekijästä. Saadakseni mahdollisimman luotettavan vastauksen tutkimuskysymykseen kaksi, oli tärkeää, että testattavissa sovelluksissa ja käyttöohjeissa täytyivät ne ominaisuudet, jotka tyypillisesti aiheuttavat käyttöohjeiden hallinnoinnissa haasteita. Näitä ominaisuuksia ovat laajat ohjelmistotuoteportfoliot, sovellusten väliset riippuvuudet, toiminnallisuuksiltaan laajat sovellukset, vanhojen ja uusien sovellusversioiden rinnakkaiskäyttö, heterogeeniset käyttäjäkunnat sekä tarve monikielisille käyttöohjeille.

Yksi käyttöohjeiden hallinnointiin vaikuttava tekijä oli siis sovellusten väliset riippuvuudet. Tällä tarkoitan sitä, että vaikka tuotteet ovat itsenäisiä sovelluksia, niiden välillä on tiedonsiirtoja tai integraatioita, jotka muodostavat niiden kesken vuorovaikutusta. Tällöin toisen sovelluksen toiminto voi vaikuttaa toisen sovelluksen käyttöön, jolloin osa näiden kahden sovelluksen käyttöohjeista koostuu samoista tiedoista. Yksi tutkimukseen valittuihin sovelluksiin kohdistuva kriteeri oli siis, että niiden välillä piti olla riippuvuuksia. Myös yksittäisten sovellusten sisäinen kompleksisuus oli yksi kriteeri; jos sovellus on hyvin suppea ja suoraviivainen, sen käyttöohjeen hallintaan ei juuri liity haasteita.

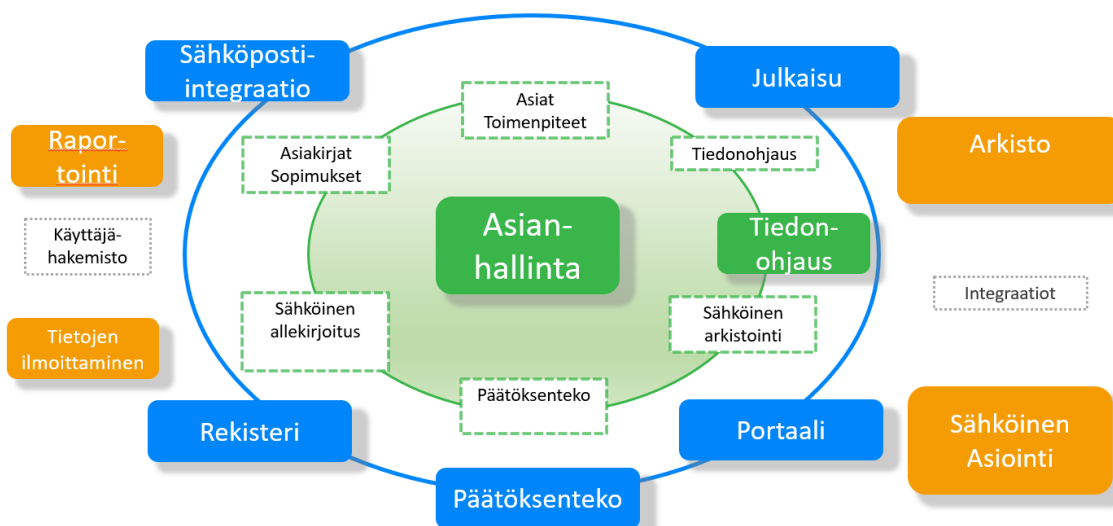
Toinen kompleksisuustekijä käyttöohjeiden hallinnassa on sovellustuotteiden eri versiot. Tuotekehityksen myötä sovelluksesta saattaa olla käyttäjäkunnassa rinnakkain aktiivikäytössä uudempia ja vanhempia versioita, joiden käyttö, ja sen myötä käyttöohjeet, eroavat joltakin osin toisistaan. Näin ollen, valittujen sovellusten piti kuulua jatkuvan tuotekehityksen piiriin.

Kolmas käyttöohjeissa huomioon otettava seikka on niiden kohderyhmät. Samaa sovellusta käyttävät erityyppiset käyttäjät, joiden tiedontarpeet eroavat toisistaan

paljon. Tiedontarpeiden eroavaisuudet voivat liittyä siihen, kuinka laajoja tai yksityiskohtaisia ohjeistuksia tarvitaan. Eroavaisuus voi myös olla siinä, että käyttäjät käyttävät sovellusta eri rooleissa, joiden tiedontarve kohdistuu vain tiettyihin roolikohtaisiin toimintoihin. Sovelluksiin liittyy useimmiten myös sellaisia tietoja, joita vain pääkäyttäjä tarvitsee. Tutkimukseen valikoituihin sovelluksiin piti siis liittyä selkeästi eri käyttäjärooleja.

Valintakriteerinä oli myös, että käyttöohjeiden piti olla olemassa esimerkiksi DOCX-muotoisina niin, että suomenkielinen ja ruotsinkielinen käyttöohje olivat substanssiltaan keskenään identtisiä.

Kuvio 1 alla näyttää skemaattisen esityksen ratkaisukokonaisuudesta, kun käytötapaus on keskikokoinen organisaatio. Kuviosta ilmenee ohjelmistokokonaisuus ja järjestelmäosien riippuvuus toisistaan.



**Kuvio 1.** Esimerkki keskikokoisen organisaation ratkaisukokonaisuudesta.

Testini kahden käyttöohjeen valintakriteerit täyttyivät suurimmaksi osaksi melkein kaikissa tuotteissa. Täysin identtisiä käyttöohjeita kahdella kielellä ei kuitenkaan kaikkiin

tuotteisiin ollut saatavilla, joten tämä rajoitti vaihtoehtoja. Päädyin tuotepakettiin, joka koostuu rekisteristä ja siihen liitetystä sähköisestä tietojen ilmoittamistyökalusta. Nämä kaksi tuotetta muodostavat suhteellisen rajatun kokonaisuuden, jolla ei juuri ole riippuvaisuuksia muihin tuotteisiin, mikä helpotti mahdollisimman autenttisen käyttöohjekokonaisuuden rakentamisen sisällönhallintajärjestelmään.

### **1.3 Aineiston analyysimenetelmät**

Analyysini koostui kahdesta osiosta. Ensimmäisessä vaiheessa vertailin kolmen sisällönhallintajärjestelmän rakenteistamista, modulaarisuutta ja yksilähteistämistä tukevia ominaisuuksia. Tutkin ominaisuuksia pääasiallisesti DITA-informaatiomallin periaatteiden näkökulmasta (Syncro Soft, n.d.d) sekä hyödynsin työpaikkani aiemmin suoritetuissa testauksissa laadittua kriteeristöä. DITA-informaatiomallista löytyy lisätietoja tutkielman luvusta 3.1.1, ja liitteessä 1 on yhteenveto DITA:n osa-alueista ja niiden huomioimisesta omassa tutkimuksessani. Työpaikkani testauskriteeristö on tutkielman liitteenä 2.

Tutkimusmenetelmät jaetaan kahteen pääkategoriaan: kvantitatiiviset eli määrälliset menetelmät, ja kvalitatiiviset eli laadulliset menetelmät. Niiden erot on kuvattu alla olevassa taulukossa. Kvantitatiiviseen menetelmään liittyy perusjoukon ja siihen liittyvien ominaisuuksien määrittäminen ja muuntaminen tilastollisesti käsiteltävään muotoon (Kyrö, 2014). Aineiston valinnassa hyödynnetään perusjoukkoa ja analyysi tapahtuu usein koejärjestelyjen avulla. Kvalitatiivisessa menetelmässä lähtökohtana on todellisen elämän kuvaaminen. Laadullisten menetelmien aineistonkeruumenetelmiä ovat esimerkiksi haastattelut ja havainnoinnit, ja siinä huomioidaan tutkittavien näkökulma (Isohella, 2019).



**Taulukko 1.** Kvantitatiivisen ja kvalitatiivisen tutkimuksen erot (Isohella, 2019, RajatOn, 2015).

Kvantitatiivinen tutkimus	Kvalitatiivinen tutkimus
Strukturoitu tutkimusstrategia	Strukturoimaton tutkimusstrategia
Aineiston luonne kova, luotettava	Aineiston luonne rikas, syvä
Teoriaa varmistava	Teoriaa luova
Tutkija aineistostaan ulkopuolinen	Tutkija aineiston tulkitsijana
Laajat määrälliset aineistot	Pienempi aineisto, aineiston laadun merkitys
Mittaaminen ja testaaminen	Havainnointi ja tulkitseminen
Vastaa kysymykseen ”kuinka suuri”, ”montako”	Vastaa kysymykseen ”miksi”, ”millainen”
Yleistettävyyys	Joustavuus

Näin ollen, ensimmäinen osuus tutkimuksestani oli kvantitatiivinen tutkimus, sillä vaikka vertailtavia järjestelmiä oli vain kolme, itse tutkimus tapahtui testaamalla ja mittaamalla. Tutkittava aineisto oli tarkoin rajattu ja tulokset olivat täsmällisiä, eivätkä jättäneet tulkinnanvaraa.

Suoritin analyysin asiantuntija-arviointina. Se on menetelmä, jota käytetään pääasiallisesti käytettävyyсарviointiin. Arviointi perustuu erilaisiin ohjeistuslistoihin, joiden tarkoitus on auttaa arvioijaa arvioimaan käyttöliittymää tai tuotetta. Arviointiin voi sisältyä myös skenaarioita, jolloin arvioijalla on ennalta määritellyt tehtävät suoritettavana (Ovaska ja muut, 2005, s. 111). Tässä tutkimusosiossa en käyttänyt skenaarioita, sillä tarkoitus ei ollut testata eri käyttötapauksia, vaan ainoastaan tarkistaa, löytyvätkö vaaditut toiminnallisuudet järjestelmistä.

Tutkimus eteni niin, että yhdistin ensiksi DITA-informaatiomallin periaatteiden (Syncro Soft, n.d.d) edellyttämät ominaisuudet ja työpaikkani testauskriteeristön ominaisuudet yhdeksi kriteeristökokonaisuudeksi. Tällöin muodostui kuudentoista vaatimuksen lista, jonka pohjalta tutkin sisällönhallintajärjestelmien ominaisuuksia. Vertailun tulokset, jotka kirjasin taulukkoon, olivat tässä vaiheessa melko teoreettisia. Ne olivat odotetusti ”Kyllä” (ominaisuus on tuettu täysin), ”Osittain” (ominaisuus on tuettu

osittain) tai ”Ei” (ominaisuutta ei ole tuettu). Ensimmäisen tutkimusosion analyysi on tarkemmin kuvattu luvussa 4.1.

Tutkimuksen toisessa osiossa toteutin osaa kahden tiedonhallintaan liittyvän sovelluksen käyttöohjeita yhteen niistä sisällönhallintajärjestelmistä, joita olin vertaillut tutkimuksen ensimmäisessä osiossa. Järjestelmätestaus on Kasurisen (2013) mukaan yleisnimitys kaikelle testaukselle, joka tehdään kokonaiselle järjestelmälle, eikä se tarkoita itsessään mitään tiettyä testaustapaa. Tavoitteena on varmistaa, että järjestelmä toteuttaa kaikki sille asetetut tavoitteet ja toimii kokonaisuutena. Ohjelmistotuotannossa järjestelmätestaus suoritetaan testiympäristössä ja siitä voi vielä seurata muutoksia toteutettavaan järjestelmään. Ganesh, (2014, s. 123) määrittelee hyväksymistestausta vaiheeksi, jossa asiakas validoi ohjelmistoa saadakseen selville vastaako se organisaation vaatimuksia. Tässä tutkimusosiossa pyrin kuudentoista vaatimuksen listan avulla selvittämään, miten hyvin rakenteistamista, modulaarisuutta ja yksilähteistämistä käytännön tasolla pystyy toteuttamaan sisällönhallintajärjestelmän tarjoamilla ominaisuuksilla ja työkaluilla. Vaikka tutkielmassani ei ollut kyse ohjelmistotoimitukseen liittyvästä testauksesta, sovellettu järjestelmätestauksen ja hyväksymistestauksen yhdistelmä vastaa testausteknisesti sitä, mitä testaamisella tavoittelin.

Suoritin testaamisen itse viemällä osia valitsemieni kahta käyttöohjetta kahdella kielellä sisällönhallintajärjestelmään. Tämän aloitin testimateriaalin esivalmistelulla, eli muutin käyttöohjeiden lineaarisen sisällön modulaariseen muotoon. Tämän tein soveltamalla Amentin (2003, s. 24) kymmenen askeleen prosessikuvausta. Seuraavaksi vein muunnetun sisällön testiin valittuun sisällönhallintajärjestelmään rakenteiseen muotoon ja yksilähteistämisen periaatteita soveltaen. Kuten tutkimuksen edellisessä vaiheessa, vertailin myös tässä järjestelmän ominaisuuksia DITA-informaatiomallin (Syncro Soft, n.d.) periaatteisiin ja työpaikkani testauskriteeristöön pohjautuvaan kuudentoista vaatimuksen listaan. Dokumentoin havaintoni sekä kirjallisesti taulukkomuotoon, että videotallenteena mahdollista myöhempää käyttöä varten. Käyttöohjeiden hallintaan liittyen olin tunnistanut viisi haastetta aiheuttavaa päätekijää: eri kohderyhmien

tiedontarpeet, tilannekohtaiset erilaajuiset tiedontarpeet, sovellusten väliset riippuvuudet, sovellusten eri tuoteversiot sekä käyttöohjeiden eri kieliversiot. Näihin pyrin löytämään ratkaisuja vaatimuksissa kuvattujen ominaisuuksien avulla. Tällä tavoin hain vastausta kysymykseen, tuoko rakenteistamista, modulaarisuutta ja yksilähteistämistä tukeva sisällönhallintajärjestelmä ratkaisuja laajan käyttöohjeportfolion hallintaan. Toisen tutkimusosion analyysi on tarkemmin kuvattu luvussa 4.2.

## 2 Teknisen dokumentoinnin toimintaympäristön ydintekijät

Tutkimuksellani tavoittelen kokonaisvaltaista katsausta ohjelmistoalan loppukäyttäjille tarkoitettun dokumentaation hallintaan. Tässä luvussa avaan teknisen dokumentoinnin toimintaympäristöön liittyvien osa-alueiden sisältöä ja selostan, millä tavalla ne kytkeytyvät tutkimukseen.

### 2.1 Ohjelmistotuotannon kompleksisuus

Ohjelmistotuotanto on yhteisnimitys niille työnteon vaiheille, jotka sisältyvät tietokoneohjelmien suunnitteluun, valmistukseen ja kehitykseen. Tyypillisiä työvaiheita ovat määrittely, suunnittelu, toteutus, testaus, käyttöönotto ja ylläpito (Kasurinen, 2013). Ohjelmistotuotannon toimijoita ovat tietokoneohjelmistoja valmistavat ohjelmistoyritykset. Pääsääntöisesti ne ovat erikoistuneet tietyn toimialan tai aihealueen ohjelmistoihin.

*Software is everywhere*, eli "Ohjelmistoja on kaikkialla" kirjoittaa Rajlich (2011, s. 3) osuvasti. Lähes jokaisen ihmisen taskusta löytyy älypuhelin kaikenlaisine sovelluksineen. Kun käymme ruokakaupassa, menemme bussilla, ajamme autoa, pesemme vaatteita ja niin edelleen. Kaikessa tekemisessämme ohjelmistoja pyörii koko ajan taustalla. Jokapäiväisissä askareissa sitä ei tule juuri ajatelleeksi. Työelämässä ohjelmistot ja digitalisaatio ovat konkreettisemmin läsnä suuressa osassa työtehtävistä.

Rajlich (2011, s. 4) kuvailee viisi olennaista ominaisuutta, jotka ovat ohjelmistoille tyypillisiä, ja, jotka tekevät ohjelmistotuotannosta haastavaa. Nämä viisi ominaisuutta ovat kompleksisuus, näkymättömyys, vaihdettavuus, mukautuvuus ja epäjatkuvuus. Kompleksisuuteen liittyy sovellusten laajuus, ja sen hallittavuuden parantamiseksi on kehitetty erilaisia strategioita ja menetelmiä, kuten ohjelmiston jakaminen pienempiin osakokonaisuuksiin. Vaihdettavuus ja epäjatkuvuus liittyvät ohjelmistojen muokkaamiseen. Ohjelmistoihin on helppo tehdä muutoksia, mutta juuri tähän

helppoutteen liittyy myös haittapuoli: jos kyseessä on laaja ja kompleksinen sovellus, muutoksia tehdessä on monta huomioitavaa seikkaa, ja jos joku niistä jää huomioimatta, siitä saattaa kumuloitua laajasti vaikuttavia seurauksia. Mukautuvuudella tarkoitetaan sopeutumista ympäröivään maailmaan. Esimerkiksi lakimuutos saattaa edellyttää ohjelmistolta uusia ominaisuuksia.

Ohjelmistotuotanto on siis hyvin kompleksinen kokonaisuus. Rajlich (2011, s. 4) pitää ohjelmistoja jopa yhtenä kompleksisimpana ihmiskunnan kehittämistä systeemeistä.

## **2.2 Tekninen viestintä ja dokumentointi**

Teknistä dokumentaatiota ja teknistä viestintää on tieteellisesti alettu laajemmin tutkia vasta 1980-luvulla (Suomivuori ja muut, 2020, s. 225). Teknologian lisäksi lähestymiskulmia tutkimuksissa ovat muun muassa käytettävyys ja visuaalisuus. Saavutettavuusdirektiivien (Euroopan Parlamentti ja Neuvosto (EU), 2016; W3, 2018) myötä myös saavutettavuus on, muun muassa osana yleistä käytettävyyttä noussut tutkimuksissa keskiöön.

Tekninen viestintä voidaan määritellä eri tavoin, mutta yhteisenä lähtökohtana kaikissa määritelmissä on se, että kyseessä on tiedon välittämistä henkilöltä, joka tietää, henkilölle, joka tiedon tarvitsee (Barnum & Carliner, 1993, s. 3). Markel (2012, s. 2) määrittelee teknisen viestinnän prosessiksi, jossa teknistä tietoa luodaan, suunnitellaan ja siirretään niin, että tieto on helposti ymmärrettävissä, turvallisella ja tehokkaalla tavalla. Tekninen dokumentointi on teknisen viestinnän osa-alue. Sen tuotokset eroavat muusta dokumentaatiosta muun muassa formaatin, rakenteen ja käytetyn kielen perusteella.

Digitalisaatio ja monipuoliset sisällönhallintajärjestelmät ovat tuoneet tekniseen dokumentointiin uusia ulottuvuuksia. Kun dokumentaatio aikaisemmin koostui lineaarisista dokumenteista, nykypäivän työkalut mahdollistavat sisällön

yksilähteistämisen, uudelleenkäytön, julkaisemisen eri formaatteihin, keskitetyn muokkaamisen ja tiedonsiirron eri tietojärjestelmien välillä. Ford ja Mott (2007, s. 333) totesivat jo 2007, että digitalisaation tuomat muutokset muuttavat myös teknisen dokumentoinnin tehtävää ja vertailevat sitä informaatioarkkitehdin tehtävään; teknisessä dokumentoinnissa on monitasoisesti huomioitava informaation rakennetta monesta eri näkökulmasta.

### 2.3 Dokumentaatioon liittyvät standardit

ISO:n (International Organization for Standardization, n.d.) hieman humoristinen kuvaus, *"Think of them as a formula that describes the best way of doing something"*, (suomeksi vapaasti käännettynä "Ajattele niitä kaavana, joka kuvaa parasta tapaa tehdä jotakin") valaisee hyvin sitä, mitä standardilla yleisesti tarkoitetaan. Puhuttaessa informaation kehittämisessä, Hackos (2016, s. 26) käyttää *The British Standards Institute*:n standardin määritelmää (oma vapaa käänös):

*Pohjimmiltaan standardi on sovittu tapa tehdä jotain. Kyse voi olla tuotteen valmistamisesta, prosessin hallinnasta, palvelun tai materiaalien toimittamisesta – standardit voivat kattaa valtavan valikoiman organisaatioiden toteuttamia ja asiakkaidensa käyttämiä toimintoja.*

*Standardit ovat viisautta, joka syntyy sellaisten ihmisten työstä, joilla on asiantuntemusta ja jotka tuntevat edustamiensa organisaatioiden tarpeet – ihmiset kuten valmistajat, myyjät, ostajat, asiakkaat, toimialajärjestöt tai käyttäjät. (Hackos, 2016, s. 26)*

International Organization for Standardization (ISO), tarjoaa standardeja hyvin laajalle skaalalle aihealueita, mukaan lukien tietotekniikka. Organization for the Advancement of Structured Information Standards (OASIS) on tärkeä kansainvälinen toimija informaatioalan standardien tuottamisessa ja kehittämisessä. Muun muassa DITA-standardi (Oasis Open, 2018), josta on kerrottu tämän tutkielman luvussa 3.1, on OASIS:n teknisen komitean ylläpitämä standardi (Hackos, 2016, s. 26).

Teknisen dokumentaation standardien tarkoitus on luoda yleisiä periaatteita ja yksityiskohtaisia vaatimuksia dokumentaation muotoon, muotoiluun, sisältöön ja rakenteeseen. Niitä voidaan soveltaa kaikenlaisiin dokumentaatioon, kuten tuotekuvauksiin, tuotekehityksen suunnitelmiin, myynti- ja markkinointimateriaaliin, asennusohjeisiin, käyttöohjeisiin, huolto-ohjeisiin ja turvallisuusohjeisiin (Palola, 2022).

Esimerkkejä tekniseen dokumentaatioon liittyvistä standardeista on EN 62079: 2001, *Preparation of instructions - Structuring, content and presentation* (iTeh, Inc, 2014) sekä BSI - BS 4899-2 *Guide to User's Requirements for Technical Manuals*. EN 62079 on kansainvälinen standardi, joka sisältää yleiset periaatteet ja yksityiskohtaiset vaatimukset, joita on noudatettava erilaisten tuotteita koskevien ohjeiden suunnittelussa ja laatimisessa (SFS, n.d.). BSI-BS 4899-2 on *British Standards Institutionin* ylläpitämä standardi. Se tarjoaa sääntöjä sellaisen teknisen sisällön esittämiseen, joka liittyy kompleksisiin tietojärjestelmiin tai rakennelmiin, jotka on tuotettu laajasti erityyppisiä käyttäjiä varten (British Standards Institution, 2023).

## 2.4 Käyttöohjeiden hallinta

Käyttöohjeiden hallinnan haasteiden voi sanoa olevan yksi seuraus kompleksisuuden kumuloitumisesta. Yksittäiset sovellukset voivat olla kompleksisia. Sovellus sisältää silloin laajasti erilaisia toiminnallisuuksia ja ominaisuuksia, jolloin sovelluksen tehokas hyödyntäminen useimmiten edellyttää jonkinasteista ohjeistusta. Useammasta toisistaan riippuvaisesta sovelluksesta muodostuu kompleksinen tuoteportfolio. Tällaisen kokonaisuuden ohjeistus ja dokumentaatio aiheuttavat väistämättä kompleksiset käyttöohjeet.

Mitä laajempi ja kompleksisempi tuoteportfolio, sitä haastavampaa on käyttöohjeiden ajan tasalla pitäminen (Redish, 2010, s. 199). Laaja tuoteportfolio tarkoittaa myös laajaa ja heterogeenistä käyttäjäkuntaa, jolloin käyttöohjeessa pitää huomioida sen eri kohderyhmiä. Esimerkiksi sovelluksen peruskäyttäjän tiedontarve on eri kuin saman

tuotteen ylläpitäjäroolillisen käyttäjän tiedontarve (Redish, 2010, s. 193). Toisaalta tiedontarve ei välttämättä ole sidoksissa käyttäjän rooliin, vaan voi olla, että esimerkiksi pääkäyttäjä tarvitsee ylläpitäjän ohjeesta vain pikaohjeversion, tai, että satunnaiskäyttäjä tarvitsee jostakin yksittäisestä kohdasta tarkempia tietoja. Tällöin voidaan puhua tilannekohtaisesta tiedontarpeen laajuudesta.

Käyttäjät kokevat usein käyttöohjeet hidasteina ja kääntyvät niiden puoleen pääsääntöisesti yksittäisten ongelmatilanteiden selvittämiseksi (Blackler ja muut, 2018, s. 26, Redish, 2010, s. 194). Käyttöohjeen rakenne tulisi siis olla selkeä, jotta yksittäinen tieto on helposti löydettävissä ja turhaa tekstiä jää mahdollisimman vähän luettavaksi. Tämä on osittain ratkaistavissa minimalistisella dokumentaatiolla, jossa käyttäjäkeskeisyys on fokuksessa, ja jossa pääpainona on toimintakeskeisyys ja tehtävälähtöisyys. Minimalistisen dokumentaation periaatteiden mukaan käyttäjälle tulee kirjoittaa lyhyesti, eikä kaikkea tarvitse selittää tai kuvailla. Se ei kuitenkaan tarkoita pelkkää tekstimäärän vähentämistä, vaan täsmädokumentaatiota käyttäjän tarpeisiin (Isohella ja muut, 2018, s. 189). Monipuolisten ja kompleksisten sovellusten käyttöohjeissa tämä on vaikea yhtälö. Kuten yllä todettu, käyttöohjeet ovat tärkeä apuväline ongelmatilanteissa, jolloin ongelman ratkaisemiseksi tarvitaan usein yksityiskohtaisia tietoja. Dokumentaatiosta ei voi siis jättää tietoa pois sillä perusteella, että sitä tarvitaan harvoin, vaan ainoastaan silloin, jos sitä ei tarvita koskaan.

Saman ohjelmistotuottajan laajan tuoteportfolion yksittäiset tuotteet ovat harvoin omia eristettyjä "saaria", vaan ne muodostavat usein keskenään modulaarisia, toiminnallisia kokonaisuuksia. Esimerkiksi sovelluksessa A suoritetuista tietyistä toimenpiteistä voi olla vaikutusta sovelluksen B:n tiettyihin toimintoihin. Tämä vaikuttaa myös käyttöohjeisiin (Redish, 2010, s. 195).

Käyttöohjeissa aiheuttaa haasteita myös se, että samasta tuotteesta saattaa olla eri tuoteversioita käytössä eri asiakkailta (Wingkvist ja muut, 2011, s. 150). Tällöin vanhaa käyttöohjetta ei voi korvata uudella, vaan se pitää versioda niin, että myös vanhempien



tuoteversioiden ohjeistukset ovat edelleen saatavilla. Tällainen versiointi on helppo tehdä luomalla uusi asiakirja vanhan pohjalta (esimerkiksi MS Word-ohjelmassa ”Tallenna nimellä” -toiminto). Silloin on kuitenkin heti olemassa kaksi asiakirjaa, ja kun johonkin vanhaan tietoon on tarve tehdä muutoksia, samat muutokset pitää tehdä kahteen paikkaan. Kun ajan kuluessa kopioita ja niistä mahdollisesti eri kieliversioita on syntynyt monta, tietojen ajan tasalla pitäminen on melkein mahdoton tehtävä.

Monesti käyttöohjeiden täytyy olla saatavilla useammalla kielellä. Kaksi- tai monikielisissä organisaatioissa on välttämätöntä, että ohjeistusten eri kieliversioiden sisällöt vastaavat toisiaan täysin. Myös ohjelmistotoimittajan asiakastukitehtävät tehostuvat, kun tukihenkilö voi tarvittaessa ohjata kysyjää suoraan käyttöohjeen oikeaan kieliversioon (Wingkvist ja muut, 2011, s. 154).

Yllä kuvattujen seikkojen valossa voi todeta, että kompleksisen tuoteportfolion käyttöohjeiden tuottaminen, ylläpito ja hyödyntäminen on vaikea toteuttaa pelkästään hyviä ohjetekstejä kirjoittamalla. Monet ohjelmistotuottajat ovat kuitenkin käyttöohjeiden tuottamisen osalta edelleen tekstinkäsittelyohjelmien varassa.

## **2.5 Sisällönhallintajärjestelmä**

Sisällönhallintajärjestelmä (eng. *Content Management System*, lyh. *CMS*) on digitaalisen sisällön luontiin, organisointiin ja ylläpitoon kehitetty työkalu. Se on ikään kuin sisällönhallinnan ekosysteemi, joka tarjoaa muun muassa mallipohjia, hallintaliittymiä työnkulkujen hallintaan, sisällöntuotantotyökaluja, julkaisutoimintoja sekä sisäänrakennettuja kollaboraatiotoiminnallisuuksia (Shivakumar, 2017). Syncro Soft:n (n.d.a) mukaan sisällönhallintajärjestelmä sisältää tyypillisesti seuraavat ominaisuudet:

- Asiakirjojen ja multimediamateriaalin tuonti ja luominen
- Käyttäjien ja heidän rooliensa hallinta
- Mahdollisuus määrittää rooleja ja vastuita eri sisältökategorioille tai -tyypeille

- Mahdollisuus ilmoittaa kiinnostuneille käyttäjille sisällön muutoksista
- Mahdollisuus seurata ja hallita yhden sisältösiintymän useita versioita
- Mahdollisuus hakea kaikki sisältö tekstistä tai metatiedoista
- Mahdollisuus julkaista sisältöä

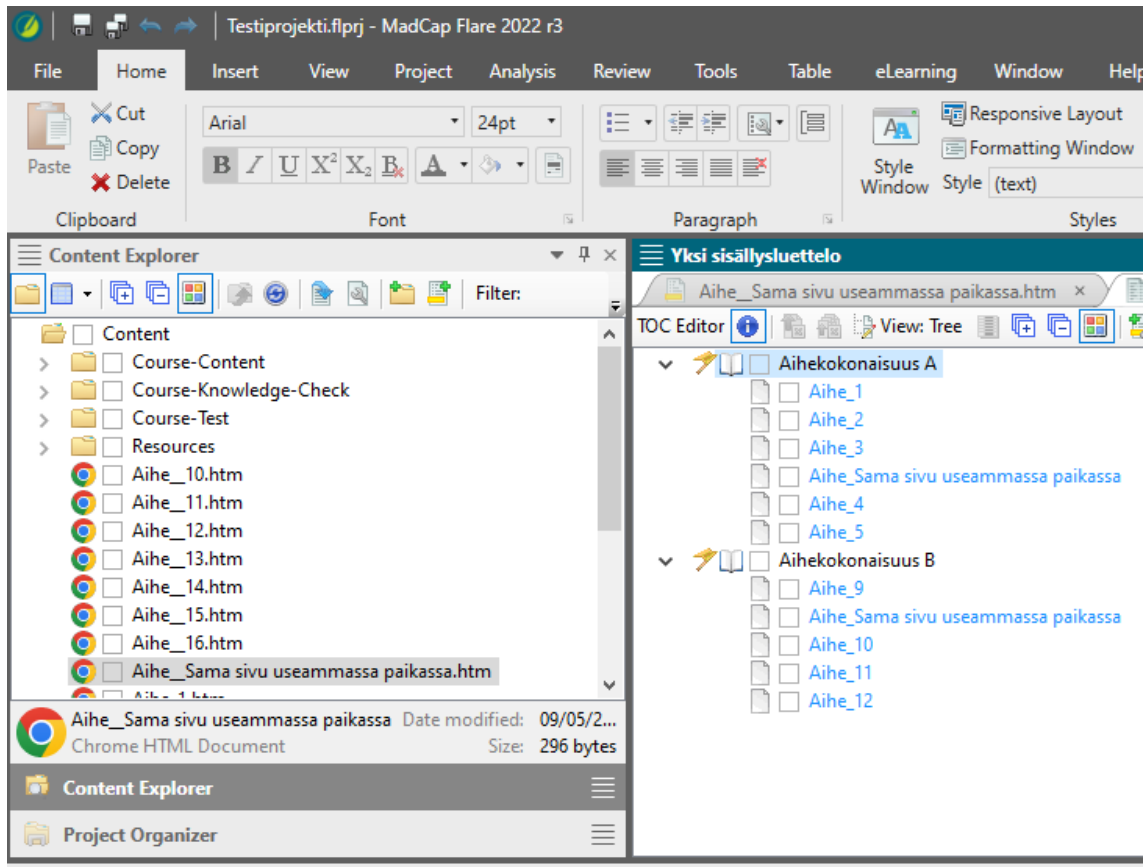
Sisällönhallintajärjestelmiä on olemassa erityyppisiä, eri sisältötyyppeihin ja käyttötarkoituksiin tarkoitettuja. Jones (2018) kuvailee kirjoituksessaan viisi perustyyppiä: dokumentinhallintajärjestelmä (eng. *Document Management System*, lyh. *DMS*), yrityksen sisällönhallintajärjestelmä (eng. *Enterprise Content Management System*, lyh. *ECM*), digitaalinen aineistonhallintajärjestelmä (eng. *Digital Asset Management System*, lyh. *DAM*), verkkosisällönhallintajärjestelmä (eng. *Web Content Management System*, lyh. *WCMS*) ja komponenttisisällönhallintajärjestelmä (eng. *Component Content Management System*, lyh. *CCMS*).

Dokumentinhallintajärjestelmä on ratkaisu, jolla tallennetaan, hallinnoidaan ja säilytetään asiakirjoja sähköisesti. Dokumentinhallintajärjestelmän avulla voidaan säilyttää, prosessoida ja jakaa dokumentteja täysin paperittomasti. Yrityksen sisällönhallintajärjestelmä kerää, järjestää ja toimittaa organisaation dokumentaatiota. Se antaa kaikille organisaation jäsenille helpon pääsyn sisältöön, jota he tarvitsevat työtehtävien suorittamiseksi ja tärkeiden päätösten tekemiseksi. Yrityksen sisällönhallintajärjestelmä poistaa tiedot tietyn säilytysajan jälkeen, mikä varmistaa, ettei tarpeeton sisältö vie tilaa. Digitaalinen aineistonhallintajärjestelmä on järjestelmä, johon käyttäjät voivat tallentaa digitaalista sisältöä, sekä järjestää ja jakaa sitä. Se on keskitetty kirjasto, jossa työntekijät, asiakkaat ja muut sidosryhmät voivat käyttää digitaalista sisältöä kuten tiedostoja, videoita, asiakirjoja ja esityksiä. Aineistonhallintajärjestelmä on pilvipohjainen, joten käyttäjät voivat käyttää sisältöä mistä tahansa.

Verkkosisällönhallintajärjestelmä on ratkaisu, jolla käyttäjät voivat hallita verkkosivuston sisältöä käyttöliittymän kautta, ilman merkintäkielten tai verkko-ohjelmoinnin

tuntemusta. Se sisältää yhteistyö-, sisällönlouontij- ja hallintatyökaluja digitaalisen sisällön hallintaan. Komponenttisisällönhallintajärjestelmä eroaa verkkosisällönhallintajärjestelmästä siten, että se järjestää sisällön granulaarisella tasolla. Se ei hallinnoi sisältöä sivu sivulta, vaan ottaa sanoja, lauseita, kappaleita ja muita elementtejä ja tallentaa ne keskitettyyn säiliöön. Kun sisältö on optimaalisesti suunniteltua komponenttisisällönhallintajärjestelmän toimintalogiikkaa hyödynnettäväksi, jokainen aihe tai muu komponentti on tallennettuna vain kerran, ja sitä voi hyödyntää eri julkaisuissa.

Kuvassa 1 näkyy esimerkki yhden komponenttisisällönhallintajärjestelmän käyttöliittymästä. Vasemmassa laidassa näkyy komponentit, ja oikealla puolella niiden eri julkaisuversioyhdistelmät. Kuvasta ilmenee, että komponentti nimeltään ”Sama sivu useammassa paikassa” esiintyy kahdessa eri julkaisuissa.



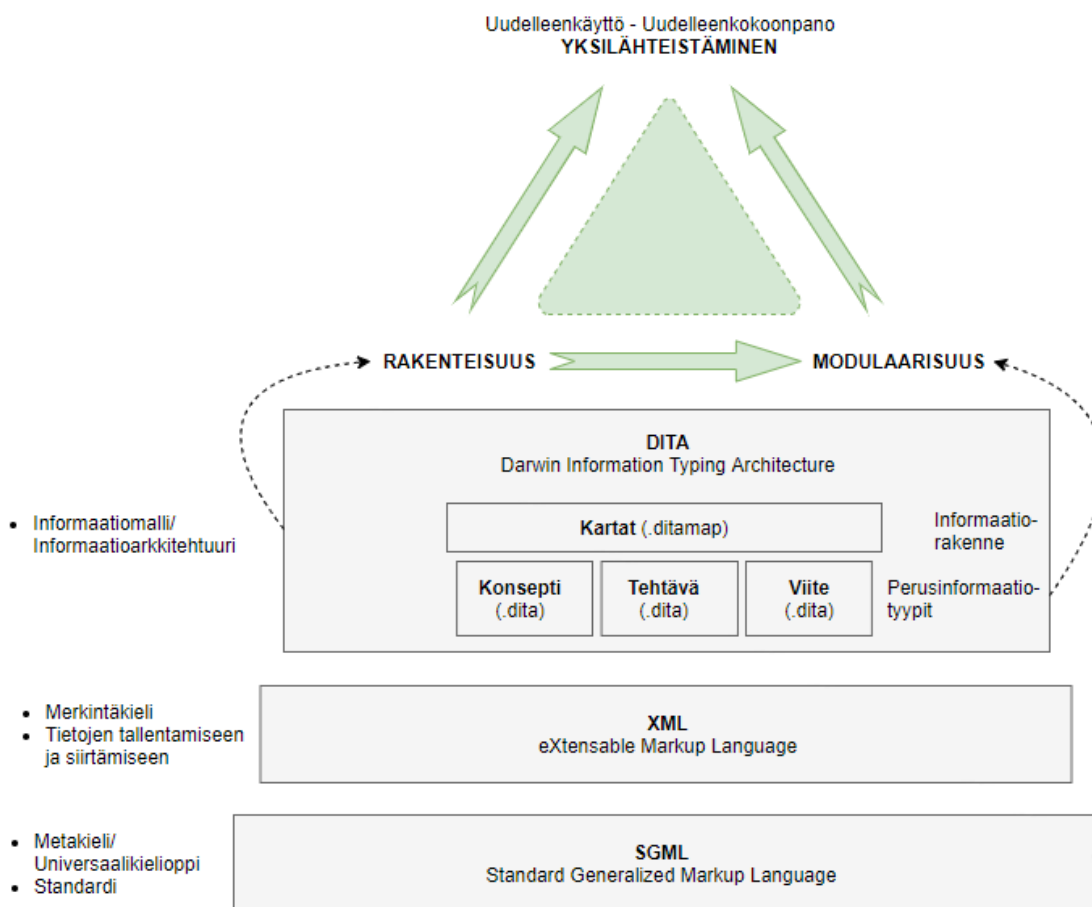
**Kuva 1.** Esimerkki komponenttisisällönhallintajärjestelmän käyttöliittymästä.

Andersen ja Batova (2015) mainitsevat kirjallisuustutkimuksessaan, että uudelleenkäyttö ja monikanavajulkaiseminen ovat komponenttisisällönhallinnan päämääriä. He toteavat lisäksi, että komponenttisisällönhallintaa pidetään yhä enemmän osana organisaation integroitua sisältöstrategiaa, jossa pyritään yhdistämään sisällönhallinnan eri tyyppisiä.

### **3 Rakenteisen dokumentaation edellytykset ja mahdollisuudet**

Dokumentaation rakenteistaminen, modulaarisuus ja yksilähteistäminen eivät ole uusia ilmiöitä. Jo 1980-luvulla, kun tietokoneiden käyttö alkoi yleistyä, oli selvää, että digitaalinen maailma toi mukanaan uusia ulottuvuuksia myös tekstisisällön tuottamisen osalta. Goldfarb (1992) kehitti ensimmäisiä standardeja jo 1970-luvulla. Suomessa esimerkiksi Nikunen totesi (1996, s. 108), että tietokoneiden tuomat mahdollisuudet tekstinkäsittelyssä edellyttävät uusia mallintamistapoja, ja esitti kirjoituksessaan Goldfarbin kehittämän standardin ratkaisuja.

Tässä luvussa tutustutaan rakenteiseen dokumentaatioon ilmiönä sekä rakenteisen dokumentaation edellytyksiin ja mahdollisuuksiin. Kun aiheeseen syventyy, huomaa, että dokumentaation rakenteistamisesta ei ole juuri mitään hyötyä ilman modulaarisuutta ja yksilähteistämistä. Nämä kolme muodostavat dokumentaation kolmiomallin, jossa on vaikeaa vetää rajaa, missä toinen osa päättyy ja toinen alkaa. Kuviossa 2 olen skemaattisesti esittänyt tämän kolmiomallin ja sen taustalla toimivat tekijät.



**Kuvio 2.** Rakenteisen dokumentaation edellytykset ja mahdollisuudet.

Teknisiä taustatekijöitä ovat SGML metakieli ja XML merkintäkieli. Niihin pohjautuu informaatiomallin rakenne, joka on edellytys rakenteisuuden, modulaarisuuden ja yksilähteistämisen toteuttamiselle. Kuviossa informaatiomallina on DITA. Kuviossa näkyvät osat on tarkemmin kuvattu luvuissa 3.1-3.3.

### 3.1 Rakenteinen dokumentaatio

Dokumentaation rakenteistamisella tarkoitetaan dokumentaation tuottamista ennalta määritellyn rakenteen mukaisesti (Horn, 1998). Teksti jaetaan erityyppisiin elementteihin, kuten esimerkiksi otsikko-, kappale- ja lista-elementteihin. Elementteille voidaan määritellä eri tyyliä, jolloin sama sisältö voi näyttää eri esitysmuodoissa

erilaiselta. Rakenteisen dokumentaation taustalla on informaatiomalli sekä metakielen avulla luodut määritteet (ks. kuvio 2). Esimerkiksi rakenteisen tekstin elementtien jäsentely on määritelty metakielen avulla (W3schools, n.d.b).

Rakenteisen dokumentaation tuottamista ja siihen liittyviä standardeja ja informaatiomalleja on tutkinut muun muassa sisällönhallinnan ja informaatio suunnittelun asiantuntijana tunnettu Hackos (2016). Yhtenä alan uranuurtajana pidetään Kurt Amentia, joka on kehittänyt lukuisia tyylioppaita ja itsekin toiminut teknisenä kirjoittajana. Hän käsittelee perusteellisesti modulaarisuuden ja yksilähteistämisen periaatteita opaskirjassaan *Single Sourcing: Building Modular Documentation* (2003).

Käytettävä merkkauskieli ja siihen liittyvät säännöt ja skeemat ovat rakenteisen dokumentaation keskeisiä teknisiä taustatekijöitä. Internetistä löytyy aihetta teknisestä näkökulmasta lähestyvää materiaalia, kuten käytännön tason oppaita. Tunnettuna esimerkkinä mainittakoon W3schools:n (n.d.) interaktiiviset oppaat, johon sisältyy muun muassa XML-merkkauksen kurssi. Myös käytettävä informaatiomalli, eli tietoarkkitehtuuri on ohjaava tekijä rakenteisessa dokumentaatiossa. Esimerkiksi Airlines for America (2019) antaa hyvin yksityiskohtaisesti tietoja S1000D dokumentointimenetelmien standardeista. DITA-informaatiomalli (*Darwin Information Typing Architecture*) on Oasis Openin (2018) ylläpitämä, ohjelmistodokumentaation hallinnointiin kehitetty standardi ja se on kattavasti kuvattu myös *The DITA Style Guide*:ssa (Syncro Soft, n.d.d).

Siirtymä lineaarisesta dokumentaatiosta rakenteiseen dokumentaatiomalliin vaikuttaa organisaatiossa suuresti myös dokumentointiprosessiin. Näiden prosessien tarkastelu ei kuitenkaan sisälly tähän tutkimukseen.

### 3.1.1 Informaatiomalli

Informaatiomalli on etukäteen määritelty tietoarkkitehtuuri. Informaatiomallien peruseriaatteet ovat sisällön uudelleenkäyttö ja yhdestä lähteestä julkaiseminen. Ne erottelevat sisällön tyyppin mukaan, ottavat siitä yksittäisiä osia ja jäsentävät ne ulkoiseen niin sanottuun karttaan julkaisemista varten. Tällainen ennalta määritelty malli tukee sisällön yhtenäisyyttä. Se ohjaa sisällöntuottajaa johdonmukaisen ja oikeanlaisen rakenteen ja sisällön muodostamiseen. Informaatiomalli voi olla standardoitu tai organisaatiolle räätälöity. Kaksi maailmanlaajuisesti tunnettua tekniseen dokumentaatioon liittyvää standardoitua informaatiomallia ovat DITA ja S1000D (Hackos, 2016, s. 24). Niistä DITA on kehitetty ohjelmistodokumentaation hallinnointiin ja S1000D laitetuotteiden dokumentaatiota varten.

DITA eli *Darwin Information Typing Architecture* (Oasis Open, 2018) on XML-pohjainen tietoarkkitehtuuristandardi, jota käytetään laajasti ohjelmistodokumentaation tuottamiseen, hallinnointiin ja julkaisemiseen. Se on alun perin IBM:n omiin tarpeisiinsa kehittämä arkkitehtuuri. Vuonna 2004 DITA-standardi lahjoitettiin OASIS Open:lle (*Organization for the Advancement of Structured Information Standards*), joka on vastannut sen kehittämisestä sitä siitä lukien. DITA-standardi koostuu joukosta suunnitteluperiaatteita, jotka tukevat sisällön kategorisointia aiheitasolla tyyppin mukaan (Evia ja muut, 2015, s. 330).

DITA-standardit ohjaavat asiakirjojen suunnittelua ja kirjoittamista. Ne määrittelevät parhaat käytännöt kirjoitustyyliin, nimeämiseen ja rakenteeseen sekä opastaa sisältöelementtien semanttiseen tunnistamiseen, mikä on DITA:n ydin. Johdonmukaisuus on DITA:ssa keskeistä, ja DITA:n periaatteet tähtäävät teknisen dokumentaation tuottamisen virtaviivaistamiseen (Syncro Soft, n.d.d).

Uudelleenkäyttöä tukevat DITA:n kaksi informaatorakennetyyppiä, ks. kuvio 2. Nämä ovat kartat (eng. *Maps*), joita käytetään sisällön hierarkkiseen strukturoimiseen sekä perusinformaatiotyyppi eli aiheet (eng. *Topics*). Käytännössä nämä ovat



käsittelyohjelmistossa tiedostoja (*DITA Files*); karttatiedostot ovat .ditamap-päätteisiä tiedostoja ja aihetiedostot ovat .dita-päätteisiä tiedostoja (Hamilton, 2022).

Modulaarisuutta tukevat DITA:n kolme perusinformaatiotyyppiä (eng. *Base DITA information types* tai *Core information types*), jotka ovat konsepti (eng. *Concept*), tehtävä (eng. *Task*) ja viite (eng. *Reference*), ks. kuvio 2. Konsepti on yleiskuavaava informaatiota, joka vastaa kysymykseen ”Mitä se on?”. Tehtävä esittää askel askeleelta eteneviä ohjeita, ja vastaa kysymykseen ”Kuinka teen?”. Viite sisältää yksityiskohtaisia lisätietoja esimerkiksi taulukko- tai luettelomuodossa (Syncro Soft, n.d.e).

Kuten edellä todettu, DITA ei ole itsessään ohjelmisto, vaan ohjelmistojen käyttämä tietoarkkitehtuuria ohjaava standardi. Ensimmäisiä kehitettyjä DITA-standardia noudattavia ohjelmistoja ovat editorityökalu Oxygen XML Editor (Syncro Soft, n.d.b) sekä julkaisutyökalu DITA Open Toolkit (DITA Open Toolkit, n.d.). Nykyään useimmat kaupalliset sisällönhallintajärjestelmät tukevat hyvin DITA:aa (Kimber, 2017).

S1000D on teknisten julkaisujen tuottamiseen kehitetty XML-pohjainen tietoarkkitehtuuristandardi. Sitä ylläpitää ja kehittää S1000D-neuvosto, joka koostuu eri organisaatioiden, muun muassa ASD (*AeroSpace and Defense Industries Association of Europe*) edustajista. S1000D huomioi teknisen dokumentoinnin prosesseja DITA:aa laajemmin (Airlines for America, 2019). Käyttökohteena on pääasiassa laitetuotteiden ja ilmailu- ja puolustusalojen liittyvä dokumentaatio, johon se alun perin on kehitettykin, joskin se soveltuu myös muuhun tekniseen dokumentaatioon. Kuten DITA, myös S1000D tukee uudelleenkäytön periaatteita ja se tunnistaa sisältömoduulit tietotyypeittäin. S1000D:ssä tämä pitää sisällään esimerkiksi kuvaukset, menettelyt ja huoltoaikataulut. Toisin kuin DITA, S1000D on täysin määritelty, ilman räätälöintimahdollisuuksia erikoisalojen tai käyttötarkoituksen mukaan (Hackos, 2016, s. 33).

### 3.1.2 Merkintäkieli

Merkintäkieli on dokumenttiin lisättyä tietoa, jonka avulla se tunnistaa osat ja miten ne liittyvät toisiinsa. Sen voi sanoa olevan joukko symboleja, jotka voidaan käyttää asiakirjan osien rajaamiseksi ja merkitsemiseksi (Tutorialspoint, n.d.).

SGML (*Standard Generalized Markup Language*) on metakieli, jota käytetään merkintäkielten määrittelyyn. Sitä sanotaan myös kuvauskieleksi tai universaalikieliopiksi. SGML on rakenteisen dokumentaation yksi ohjaava tekninen taustatekijä. Kaksi yleisintä SGML:ään pohjautuvaa merkintäkieltä on XML ja HTML, joista XML on itsessään myös metakieli, jonka pohjalta on kehitetty muita merkintäkieliä (W3, n.d.).

XML (*eXtensible Markup Language*) on SGML-merkintäkielen pohjalta kehitetty ja SGML:ä yksinkertaisempi merkintäkieli, joka on luotu tukemaan rakenteisen dokumentaation vaatimuksia. XML on muun muassa DITA- ja S1000D-standardien perusta (Hackos, 2016, s. 32), ja sitä voidaan käyttää tietojen tallentamiseen ja siirtämiseen. XML on itsensä kuvailevaa kieltä, jolloin sekä ihmiset että tietokoneet ymmärtävät sitä. Se on myös sekä ohjelmisto- että laiteriippumaton. Kielen nimen termi *extensible* tarkoittaa suomeksi laajennettavaa, mikä kuvaa hyvin sitä, että XML:ssä ei ole ennalta määriteltyjä tunnisteita, vaan sille voidaan määritellä omat tunnisteet. Myös niiden esiintymisjärjestys, käsittelytapa ja näyttötapa on määriteltävissä. Tämän ansiosta XML:llä voidaan tehdä vaativampia asioita kuin HTML:llä. XML myös erottelee tiedon ja esityksen toisistaan (W3schools, n.d.b).

HTML (*Hyper Text Markup Language*) on verkkosivujen tuottamiseen kehitetty merkintäkieli. Sillä on ennalta määritellyt tunnisteet jotka kertovat selaimelle millä tavalla sisältö pitää esittää; otsikkona, kappaleena, kuvana ja niin edelleen (W3schools, n.d.a).

## 3.2 Modulaarisuus

Modulaarisuudella tarkoitetaan dokumenteissa olevan informaation pilkkomista moduuleihin eli pieniin toisistaan riippumattomiin rakenneosiin. Moduulien avulla informaatiota voidaan käyttää uudelleen ja sen hallintaa voidaan tehostaa. Hallinnan tehostuminen näkyy muun muassa käänös-, julkaisu- ja tuotantoprosessien tehostumisena (Isohella, 2021).

Standardoitujen moduulien käyttö tarjoaa useita etuja, kuten johdonmukaisuutta sekä sisällön osakokonaisuuksien uudelleenkäyttöä. Uusien työntekijöiden on myös helpompi lähestyä dokumentaatiota, jossa on käytetty standardoituja malleja ja perussisältötyyppejä. Tallentamalla dokumentaatiolähteet modulaarisella tavalla optimoidaan keinoja, joilla dokumentaatio voidaan havainnollistaa kohderyhmäkohtaisesti (Kratky, 2017).

Modulaarinen muoto on yksi yksilähteistämisen edellytyksiä (ks. kuvio 2). Ament toteaa (2003, s. 1), että yksilähteistäminen ei ole teknologia, vaan metodologia. Näin ollen yksilähteistämiseen kehitetyistä sisällönhallintajärjestelmistä on hyötyä vain, jos sisältö on luotu modulaarisesti.

Tiedon kartoitus (eng. *Information Mapping*) on rekisteröity metodi, jonka pohjalta voidaan päästä modulaarisen sisällön tuottamiseen. Ensimmäisiä tiedon kartoituksen metodeja on kehittänyt Robert E. Horn (1978, s. 39), ja periaatteet pätevät yhä. Tiedon kartoitus koostuu kolmesta päävaiheesta: analysointi, organisointi ja esittely (TCBok, n.d.).

Amentin (2003, s. 23) mukaan modularisointiin ja yksilähteistämiseen tähtäävä dokumentointiprosessi ei aluksi eroa kovin paljon perinteisestä dokumentointiprosessista, josta syntyy lineaarinen dokumentaatio. Ennen kuin informaation tuottaminen voi alkaa, pitää tunnistaa käyttäjät, minkä tyyppistä informaatiota he tarvitsevat, sekä minkä tyyppisillä dokumenteilla informaatio parhaiten

käyttäjille välittyä. Kun informaation tuottaminen alkaa, on ero yksilähteistämiprojektin ja perinteisen dokumentointiprojektin välillä kuitenkin dramaattinen. Alla olevaan taulukkoon olen koostanut Amentin (2003) teoksen luvuissa 1 ja 2 esitetyt oleellimmat modulaarisen ja lineaarisen dokumentaation väliset erot.

**Taulukko 2.** Modulaarinen vs. lineaarinen dokumentaatio.

Modulaarinen dokumentaatio	Lineaarinen dokumentaatio
Luodaan niin sanottuina <i>stand-alone</i> -, eli itsenäisinä moduuleina	Luodaan kertomuksina, joita on tarkoitus lukea tietyssä järjestyksessä alusta loppuun
Ei-hierarkkinen	Hierarkkinen
Keskittyy sisältöön, ei formaattiin	Luodaan tiettyä formaattia varten
Luodaan yhteen master-lähteeseen	Luodaan dokumenttikohtaisesti
Uudelleenkäyttö helppoa	Uudelleenkäyttö vaikeaa, vaatisi muokkaamista
Tukee johdonmukaisuutta	

Siirtymä lineaarisesta dokumentaatiosta modulaariseen, yksilähteistämiskelpoiseen dokumentaatioon on Amentin (2003, s. 24) mukaan prosessi, joka voidaan jakaa kymmeneen osa-askeleeseen. Ensimmäinen askel on moduulien tunnistaminen. Tällöin tunnistetaan niitä erilaisia osia, joista dokumentti koostuu. Näitä osia ovat esimerkiksi aiheet, prosessit, kuvat ja huomautukset. Osat voivat Amentin mukaan olla ensisijaisia tai toissijaisia. Seuraavana askeleena tulee tunnistetuille moduuleille antaa etiketit. Kolmas askel on ensisijaisten ja toissijaisten moduulien organisoiminen. Neljäntenä ja viidentenä vaiheena moduuleja kootaan ja muokataan tulevia julkaisutarpeita huomioon ottaen. Kuudentena vaiheena tulee dokumenttien organisoiminen. Tässä yhteydessä tarkoitetaan dokumenteilla sekä asiakokonaisuuksia että erillisiä julkaisuja. Modulaarisen sisällön ansiosta voidaan toteuttaa julkaisuja esimerkiksi tietylle kohderyhmälle tai tiettyä käyttötarkoitusta varten. Seitsemäs vaihe on dokumenttien ristiinviittaukset. Saman julkaisun sisällä voidaan linkittää osioita toisiinsa. Tämän vaiheen yksi tärkeä osa on sisällysluetteloiden muodostaminen. Loppukäyttäjällä tiedon löytäminen helpottuu huomattavasti, kun sekä koko osion että yksittäisen sivun sisällysluettelot ovat käytettävissä. Kahdeksannessa vaiheessa dokumentit

konvertoidaan eri esitysmuotoihin, kuten online-helpiin tai PDF:ksi. Yhdeksäs vaihe koostuu dokumenttien testaamisesta. Amentin opastus, joka on tehty kokonaisia yksilähteistämisprojekteja varten, sisältää vielä kymmenentenä askeleena oppaiden kehittäminen organisaation dokumentointityön tueksi.

### 3.3 Yksilähteistäminen

Yksilähteistäminen (eng. *Single Sourcing*) tarkoittaa informaation uudelleenkäyttöä eri formaateissa. Yksinkertaisimmillaan yksilähteistäminen tarkoittaa saman sisällön esittämistä eri teknisissä formaateissa, esimerkiksi verkkosivulla, mobiiliversiona ja PDF-asiakirjana. Tällöin kaikissa versioissa on sama substanssi, mutta esitystapa ja esimerkiksi ulkonäkö ovat eri esitysmuodoissa erilaiset. Ament (2003, s. 15) käyttää tälle yksilähteistämistyyppille termiä *repurposing* (suomeksi vapaasti käännettynä "uudelleenkäyttö").

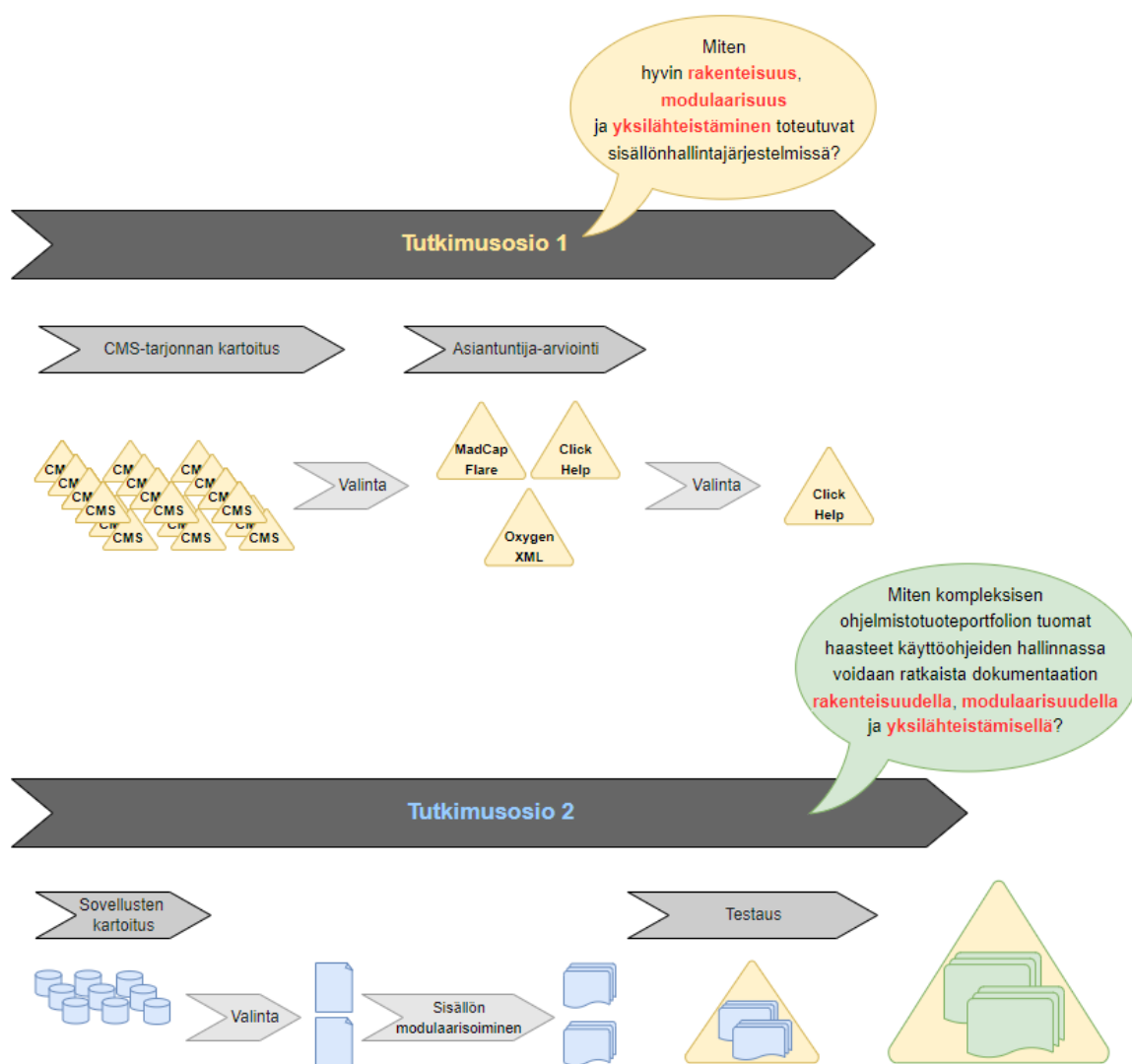
Laajemmassa perspektiivissä yksilähteistäminen on kuitenkin moniulotteisempaa ja ulottuu substanssin tilannekohtaiseen räätälöintiin. Tällöin kaikki sisältö luodaan vain kerran ja tästä niin sanotusta master-lähteestä julkaistaan valittuja moduuleja sellaisessa yhdistelmässä ja sellaisessa muodossa, joka parhaiten palvelee kyseisen julkaisun kohderyhmää. Amentin (2003, s. 15) käyttämä termi tälle on *re-assembly* (suomeksi vapaasti käännettynä "uudelleenkokoonpano"). Käyttöohjeessa se voi tarkoittaa, että esimerkiksi satunnaiskäyttäjä, peruskäyttäjä ja ylläpitäjäkäyttäjä näkevät kukin vain tarvitsemansa osan informaatiosta, vaikka tiedot heijastuvat samasta lähteestä. Toinen esimerkki on eri tuoteversioiden käyttöohjeet; siltä osin kun tuoteversioiden ominaisuudet ovat samat, sisältö heijastuu master-lähteestä identtisenä kummankin tuoteversion käyttöohjeeseen ja eroavaisuuksista näkyy kunkin tuoteversion käyttöohjeessa kyseisen version tiedot.

Yksilähteistäminen edellyttää sisällön tuottamista modulaariseen muotoon (ks. kuvio 2). Sisältö pitää siis suunnitella moduuleittain niin, että moduulien käyttö on mahdollista

yhdistettynä toisiinsa eri tavoin, ilman että sisällön eheys menetetään (Ament, 2003, s. 3). Tätä voi hyödyntää esimerkiksi tilanteessa, että erirooliset käyttäjät tarvitsevat samasta tuotteesta osittain samoja tietoja ja osittain roolikohtaisia tietoja.

## 4 Sisällönhallintajärjestelmä käyttöohjeiden hallinnassa

Tässä luvussa kuvailen analyysin etenemistä, sekä eri osa-alueiden kytköstä edellä kuvattuun teoreettiseen viitekehykseen. Tutkimusprosessi kokonaisuudessaan, mukaan lukien tutkimusaineiston keruuvaihe, on visuaalisesti esitetty kuviossa 3.



**Kuvio 3.** Tutkimusprosessi.

Kuten kuviosta ilmenee, tutkimusprosessi jakaantui kahteen osioon, jossa kunkin osion analyysi kohdistuu yhteen tutkimuskysymykseen. Nämä analyysit on kuvattu seuraavissa luvuissa 4.1 ja 4.2.

## 4.1 Sisällönhallintajärjestelmien vertailu

Tutkimuksen ensimmäisessä osiossa tutustuin kolmen komponenttisisällönhallintajärjestelmän ominaisuuksiin. Vertailuun valitsemani järjestelmät olivat MadCap Flare (Madcap Software, 2023c), ClickHelp (ClickHelp, 2023a) ja Oxygen XML Editor (Syncro Soft, 2023). Valintaprosessi ja siihen liittyvät kriteerit on kuvattu tämän tutkielman luvussa 1.2.1. Tutkin, miten sisällönhallintajärjestelmien ominaisuudet tukevat rakenteisuuden, modulaarisuuden ja yksilähteistämisen periaatteita.

Saadakseni vertailukelpoiset tulokset, oli nojattava yhteen informaatiomalliin ja sovellettava samat kriteerit kaikissa. Tässä käytin apuna DITA-informaatiomallin periaatteita. Periaatteet koostuvat hyvin laajasta kokonaisuudesta, joka on kuvattu Oasis Openin standardissa (2018). Koska informaatiomallin tarve tutkimuksessani oli tulosten vertailukelpoisuuden takaaminen, eikä tavoitteena ollut itse informaatiomallin syvällinen tutkiminen, hyödynsin tyylioppaan *The DITA Style Guide*:n (Syncro Soft, n.d.) yhteenvetona standardista. Oppaassa opastetaan myös DITA:n tekniseen toteuttamiseen, mutta, koska tutkimukseni keskittyi viestinnälliseen näkökulmaan, hyödynsin siitä vain niitä periaatteita, jotka ohjaavat asiakirjojen sisällöllistä suunnittelua ja kirjoittamista. Tyylioppaassa periaatteet on jaoteltu kahdeksaan kategoriaan, jotka olen kuvannut tämän tutkielman liitteessä 1.

DITA-informaatiomallin lisäksi hyödynsin tässä tutkimusosiossa työpaikallani aikaisemmin suoritetuissa sisällönhallintajärjestelmien testauksissa laadittua kriteeristöä, joka tarkastelee sisällönhallintajärjestelmien ominaisuuksia käytännönläheisemmästä näkökulmasta. Kriteeristön avulla mitattiin silloin muitakin ominaisuuksia kuin rakenteisuuteen, modulaarisuuteen ja yksilähteistämiseen pohjautuvia toiminnallisuuksia. Työpaikkani testeissä punnittiin ja pisteytettiin sisällönhallintajärjestelmiä sekä ylläpitäjän että loppukäyttäjän näkökulmasta. Sisällönhallintaan liittyvät kriteerit perustuivat työpaikallani todettuihin tarpeisiin ja ne olivat karkeasti jaettuina kuuteen pääkategoriaan: navigointi ja tiedon nopea



löytyminen, yksilähteistäminen, kuvien käsittely, versionhallinta, kaksikielisyyden huomioiminen sekä muut käytettävyyteen liittyvät ominaisuudet. Tämä kategoriajaottelu on eri kuin DITA-informaatiomallin oppaan jaottelu, mutta testattavat ominaisuudet vastaavat DITA:n periaatteita. Työpaikallani laadittu kriteeristö löytyy tämän tutkielman liitteenä 2. Kriteeristön pääkategorioista liittyivät tähän tutkielmaan eniten navigointi, yksilähteistäminen, versionhallinta ja kaksikielisyyden huomioiminen.

Alla näkyy laatimani yhteenveto ominaisuuksista, joita lähdin sisällönhallintajärjestelmistä hakemaan. Yhteenvetoon olen yhdistänyt DITA-oppaan periaatteiden täyttymisen edellyttämät ominaisuudet ja työpaikkani testauskriteeristön sisältämät ominaisuudet. Olen ryhmitellyt niitä sen mukaan, mihin kolmesta osa-alueesta ne eniten liittyvät: rakenteisuuteen, modulaarisuuteen tai yksilähteistämiseen. Tätä ryhmittelyä ei pidä katsoa mustavalkoisesti. Käytännössä melkein kaikki ominaisuudet vaikuttavat jollakin tavalla kaikkiin kolmeen osa-alueeseen. Ryhmittelyn avulla näytän, mihin ne oman tulkintani perusteella kaikista eniten liittyvät. Lisäksi neljäs ryhmä muodostuu niistä ominaisuuksista, jotka eivät suoraan liity mihinkään mainituista kolmesta osa-alueesta.

#### **Eniten rakenteisuuteen liittyvät ominaisuudet**

Muotoilun ja sisällön erottelu toisistaan

Informaatiotyyppien mallien määrittely ja käyttö

#### **Eniten modulaarisuuteen liittyvät ominaisuudet**

Sisällön jakaminen omiksi tiedostoiksi tallentuviksi moduuleiksi

Sisällön johdonmukainen muotoilu erityyppisiksi elementeiksi

Sisällön näyttäminen valitun laajuuden mukaan

Sisällysluettelo – koko sivusto

Sisällysluettelo – yksittäinen sivu

**Eniten yksilähteistämiseen liittyvät ominaisuudet**

- Sisällön näyttäminen eri kokonaisuuksien välillä ristiin
- Kirjanmerkit
- Kuvien uudelleenkäyttö
- Sisällön organisoiminen räätälöidyksi julkaistaviksi kokonaisuuksiksi
- Sisällön ehdollinen julkaiseminen

**Muut ominaisuudet**

- SVG- ja PNG-muotoisten kuvien käyttö
- Kuvien sijainnin ja koon säätäminen
- Multimediasisällön käyttö (esim. Flash-animaatiot ja videot)
- Kieliattribuuttien käyttö

Tässä tutkimusosiossa kyseessä ei ollut käytettävyysestaus, vaan itsenäisesti suorittamani asiantuntija-arviointi, jonka lopputulosten mittaamisen vastauksien oletin olevan melko yksiselitteisesti kyllä tai ei. Valli (2018) suosittelee semanttisen differentiaalimittauksen ja muiden vastaavien tyyppisten mitta-asteikkojen avulla kerätyn aineiston koodaamista asteikolla, joka sisältää negatiivisen ja positiivisen ulottuvuuden, esimerkiksi -2... +2. Näin ollen hyödynsin tulosten vertailemisessa asteikkoa -1... +1, jossa -1 tarkoittaa "Ei", eli ominaisuutta ei ole järjestelmässä, +1 tarkoittaa "Kyllä", eli ominaisuus on järjestelmässä olemassa, ja 0 tarkoittaa "Osittain", eli ominaisuus on järjestelmässä osittain olemassa.

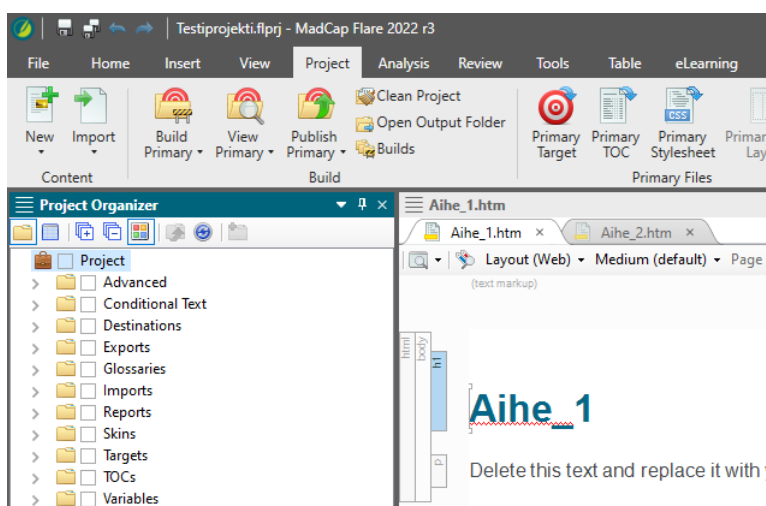
Suunnitelmana oli tämän tutkimusosion päätteeksi valita tutkituista kolmesta komponenttisisällönhallintajärjestelmästä se, joka parhaiten täytti rakenteistamisen, modulaarisuuden ja yksilähteistämisen vaatimuksia. Koska vertailtavien sisällönhallintajärjestelmien väliset erot tämän suhteen osoittautuivat hyvin pieniksi, myös sisällönhallintajärjestelmien haltuunotettavuus vaikutti tähän valintaan. Toimintojen käytettävyys, käyttöohjeiden selkeys ja käyttötuen saatavuus sujuvoitti

tutkimusprosessia, jolloin pystyin minimoimaan tutkimuksen kannalta epärelevantteja selvityksiä.

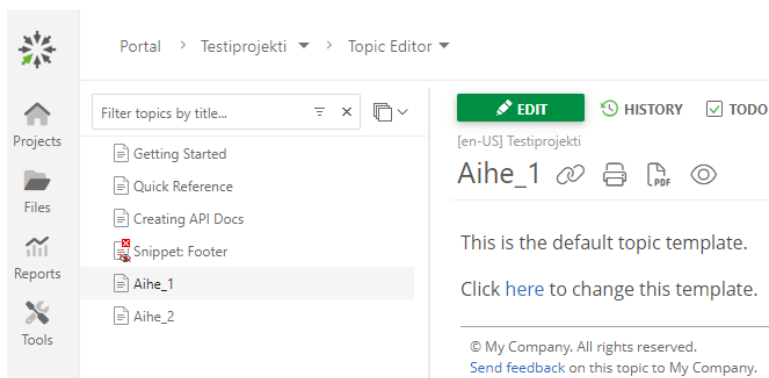
Hyödynsin tutkimuksessa järjestelmien maksuttomia kokeiluversioita, joissa toiminnallisuuksia pystyi täyslaajuisesti kokeilemaan. Käytin kunkin järjestelmän viimeisintä versiota, joka MadCap Flare:ssa oli 2022\_r3, ClickHelp:ssä Wave ja Oxygen XML Editor:ssa 25.0. Kokeiluversion väärinkäytön estämiseksi MadCap Flare:n maksuttomassa kokeiluversiossa tuotettu teksti on sekoitettu, eli kirjaimia on säännöllisin välein korvattu muilla kirjaimilla. Tämä ei kuitenkaan rajoittanut toiminnallisuuksien tutkimista, koska tekstistä vain noin joka kolmaskymmenes merkki oli korvattu.

Koska tämä osio tutkimuksesta kohdistui puhtaasti järjestelmäominaisuuksiin, tähän ei liittynyt materiaalin sisällöllistä käsittelyä tai muuntamista modulaariseen muotoon. Loin ensiksi testattaviin järjestelmiin pienimuotoiset testiprojektit. ”Projekti” on sisällönhallintajärjestelmissä yksi kokonaisuus, joka vastaa esimerkiksi yhtä käyttöohjetta. Se on ikään kuin säiliö tai kansio, johon tallentuu kyseisen kokonaisuuden kaikki aiheet, jokainen omana dokumenttina. Tutustuin sisällönhallintajärjestelmien käyttöohjeisiin ja etsin sieltä tietoja kunkin testattavan ominaisuuden käytöstä. Kaikkiin kolmeen järjestelmään oli saatavilla kattavat ohjeistukset, joita hyödyntämällä pystyin suureksi osaksi alustavasti päättämään, löytyykö vaadittu ominaisuus järjestelmästä vai ei. Käyttöohjeet on kuitenkin laadittu järjestelmien käytön tueksi, eikä varsinaisesti ominaisuuksien kuvausta varten. Näin ollen, niissä on yksityiskohtaisesti selostettu vaihe vaiheelta, miten tiettyä toimintoa käytetään, mutta ei suoranaisesti sitä, mikä on toiminnan lopputulos, joka soveltamassani tutkimusasetelmassa olisi ollut olennaisempi tieto. Sisällönhallintajärjestelmien tuottajien verkkosivuilta löytyy myös tuotekuvauksia, jotka kuitenkin ovat lähinnä markkinointia ja myyntiä varten, eivätkä näin ollen anna yksityiskohtaisia tietoja järjestelmien konkreettisista ominaisuuksista.

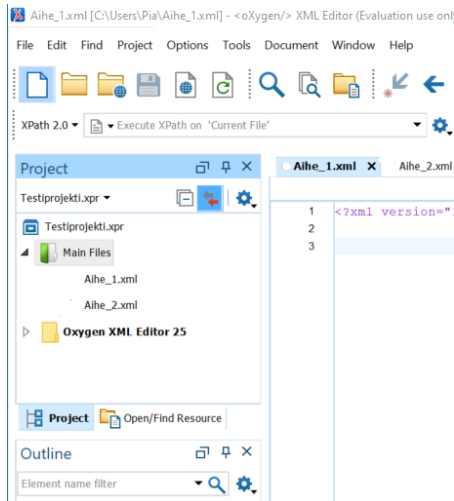
Mikäli hakemani ominaisuus käyttöohjeiden tietojen perusteella vaikutti sisältyvän, varmistin tämän vielä rakentamalla siitä mahdollisimman yksinkertaisen version testiprojektiin. Alla olevista kuvista 2-4 näkyy kuhunkin vertailtavaan järjestelmään luomani testiprojektit ja niiden alaisuuteen luodut aiheet, eli moduulit. Kuvista näkyy, että peruslogiikka on kaikissa kolmessa järjestelmässä sama, mutta käyttöliittymien rakenteissa ja kyseisen vaiheen toiminnoissa on eroavaisuuksia.



**Kuva 2.** Testiprojekti ja aiheet MadCap Flare -järjestelmässä.



**Kuva 3.** Testiprojekti ja aiheet ClickHelp-järjestelmässä.



**Kuva 4.** Testiprojekti ja aiheet Oxygen XML Editor -järjestelmässä.

Kaikki valitsemani järjestelmät olivat ominaisuuksiltaan laajat. Ominaisuusriikkaus ja monipuolisuus ovatkin ehdottomia vaatimuksia silloin, kun järjestelmällä hallittava sisältö on laaja ja kompleksinen, mutta samalla se myös edellyttää dokumentaation suunnittelijoilta sisällönhallintajärjestelmän syvällistä tuntemusta. Samoin järjestelmien vahvemmassa tuntemuksesta olisi ollut hyötyä vertailussani. Tällainen tietotaito kuitenkin voi tulla vain pidemmän kokemuksen myötä, joten vertailussani turvauduin saatavilla oleviin käyttöohjeisiin.

Komponenttisisällönhallintajärjestelmien vertailuprosessi eteni kaikissa kolmessa vertailtavassa järjestelmässä rinnakkain. Testaamalla ominaisuuksia tällä tavalla systemaattisesti yhtä kerrallaan pyrin sujuvoittamaan vertailua. Testaamisen tein siiten, että ensiksi etsin kaikkien kolmen sisällönhallintajärjestelmän käyttöohjeista tietoa hakemastani ominaisuudesta, esimerkiksi ”Sisällön ehdollinen julkaiseminen”. MadCap Flaresta kyseinen ohjeistus löytyi aiheesta ”Ehtojen liittäminen kohteisiin” (eng. *Associating Conditions With Targets*) (Madcap Software, 2023a), ClickHelpistä aiheesta ”Julkasumerkinnät” (eng. *Output Tags*) (ClickHelp, 2023b) ja Oxygen XML Editorista aiheesta ”Profilointi ja ehdollinen teksti” (eng. *Profiling and Conditional Text*) (Syncro Soft, n.d.c). Ohjeistusten tietojen perusteella päättelin, että kyseinen ominaisuus on

järjestelmissä tuettu, jolloin vielä varmistin asiaa toteuttamalla ominaisuutta järjestelmiin, kyseisten käyttöohjeiden avulla.

## 4.2 Käyttöohjeiden toteuttaminen sisällönhallintajärjestelmään

Tutkimuksen toisessa osiossa testasin, miten hyvin rakenteistamisen, modulaarisuuden ja yksilähteistämisen periaatteita pystyy sisällönhallintajärjestelmässä käytännön tasolla hyödyntämään. Tämän tein rakentamalla osan kahden sovelluksen käyttöohjeista edellisessä tutkimusvaiheessa valittuun komponenttisisällönhallintajärjestelmään. Testaamiseen olin valinnut työpaikkani Triplan Oy:n tuoteportfoliosta kaksi tiedonhallintaan liittyvän sovelluksen käyttöohjeet. Perustelut näiden kahden sovelluksen valintaan on kerrottu tämän tutkielman luvussa 1.2.2.

Tutkimuksessani käytin apuna Amentin (2003, s. 24) prosessikuvausta, jossa TCBoK:n (n.d.) kuvaamat tiedon kartoituksen kolme päävaihetta eli analysointi, organisointi ja esittely, on purettu kymmeneen osa-askeleeseen. Amentin prosessikuvauksessa on kuvattu vaihe vaiheelta miten siirtymä lineaarisesta dokumentaatiosta modulaariseen, yksilähteistämiskelpoiseen dokumentaatioon voidaan tehdä. Yhteenveto Amentin prosessikuvauksesta löytyy tämän tutkielman luvusta 3.2. Prosessin askeleet ilmenevät alla olevasta taulukosta 3, ryhmiteltynä TCBoK:n kolmen päävaiheen mukaan. Harmaalla taustavärillä merkityt askeleet 9 ja 10 eivät sisällyneet tähän analyysiin.

**Taulukko 3.** Dokumentaation yksilähteistämisen prosessin askeleet.

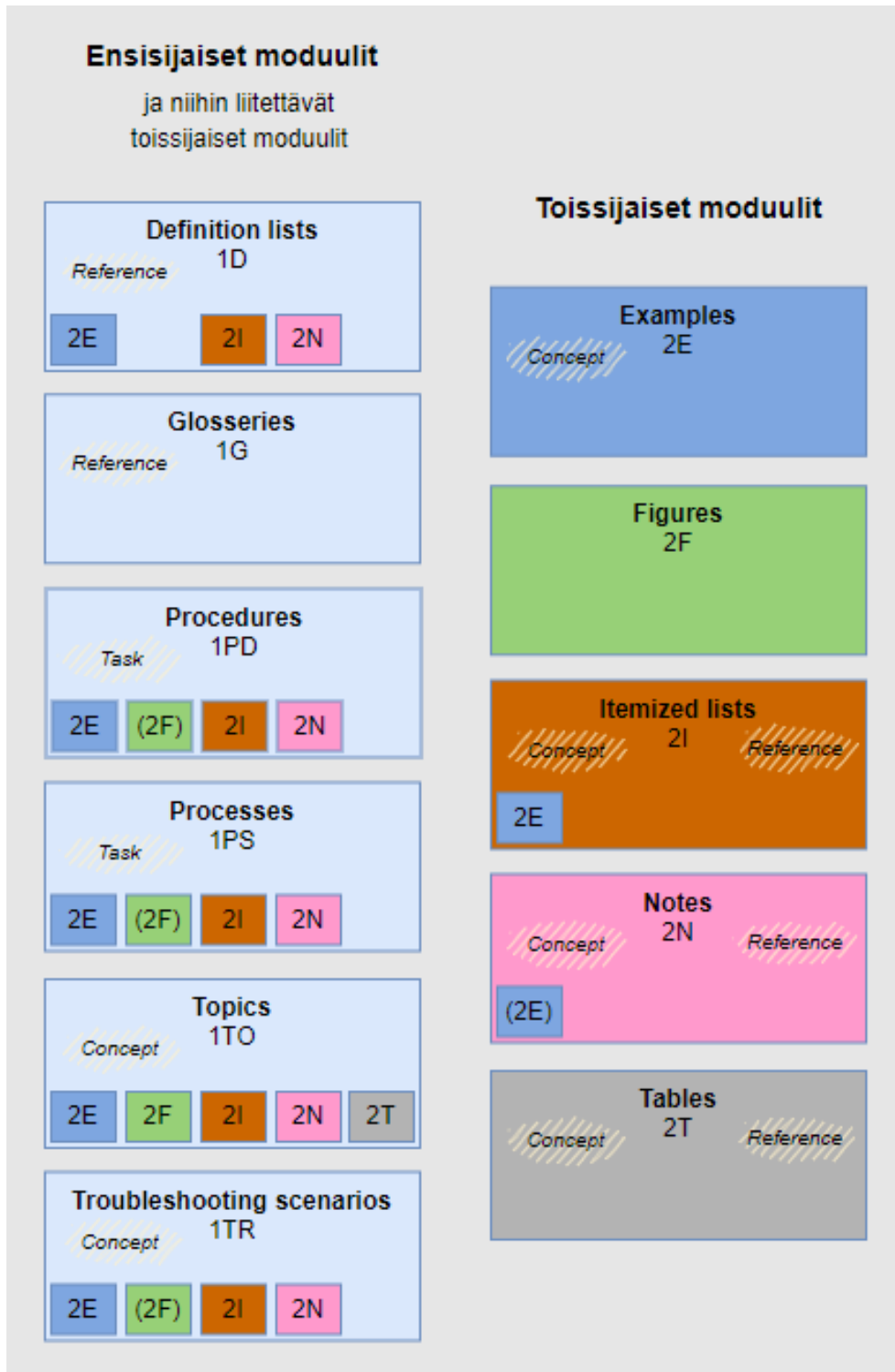
Päävaihe (TCBoK:n mukaan)	Askel (Amentin mukaan)
Sisällön analysointi	1. Moduulien tunnistaminen
	2. Etikettien antaminen moduuleille
	3. Moduulien organisoiminen
	4. Moduulien kokoaminen
	5. Moduulien muokkaaminen
Sisällön organisointi	6. Dokumenttien organisoiminen
	7. Dokumenttien ristiinviittaukset

Päävaihe (TCBok:n mukaan)	Askel (Amentin mukaan)	
Sisällön esittely	8.	Dokumenttien konvertointi
---	9.	Dokumenttien testaaminen
	10.	Dokumentointioppaiden kehittäminen

Seuraavaksi kuvaan, miten käyttöohjeiden toteuttaminen sisällönhallintajärjestelmään Amentin prosessikuvauksen kahdeksan ensimmäisen askeleen avulla tapahtui omassa tutkimuksessani.

#### 4.2.1 Sisällön analysointi

Amentin (2003, s. 26) prosessikuvauksen viisi ensimmäistä askelta, eli moduulien tunnistaminen, etiketointi, organisoiminen, kokoaminen ja muokkaaminen muodostavat analysointivaiheen. Tämän tutkielman luvussa kolme kuvaamani dokumentaation kolmiomallin kolmesta osa-alueista, analysointivaihe kytketty eniten modulaarisuuteen. Analysointivaihe on sisällön esivalmisteluvaihe, joten en vielä hyödyntänyt sisällönhallintajärjestelmää, vaan suoritin sen kokonaisuudessaan käyttöohjeiden DOCX-tiedostoissa. Se tarkoitti olemassa olevien DOCX-muotoisten käyttöohjeiden purkamista ja muuntamista modulaariseen muotoon. Tämä ei ole ollenkaan suoraviivaista, vaan vaatii substanssitason suunnittelua ja sisällön punnitsemista eri näkökulmista. Amentin (2003, s. 26) prosessikuvauksessa ensimmäinen askel on moduulien tunnistaminen. Hän jakaa moduulit kahteen tasoon: ensisijaiset (eng. *primary*) ja toissijaiset (eng. *secondary*). Ensisijaiset moduulit ovat päämoduuleja, jotka muodostavat varsinaisen sisällön. Niitä ovat esimerkiksi aiheet, prosessit ja määrittelyluettelot. Toissijaisia moduuleja ovat esimerkiksi kuvat, huomautukset ja taulukot, jotka täydentävät ensisijaisten moduulien sisältöä. Kuviossa 4 alla esittelen moduulityypit visuaalisesti, sekä miten Amentin mukaan toissijaiset moduulit liittyvät ensisijaisiin moduuleihin. Ament ei ota kantaa moduulien DITA-informaatiotyyppien vastaavuuksiin, vaan nämä merkinnät kuviossa ovat omat tulkintani.



**Kuvio 4.** Moduulityypit Amentin mukaan.

Lähdin liikkeelle olemassa olevista DOCX-muotoisista käyttöohjeista, ja merkkasin erityyppiset moduulit koodeilla niihin. Koodaus oli tähän spesifi tarkoitukseen itse



keksimäni. Koodit ovat muodostettu moduulityypin tason sekä nimen alkukirjaimen perusteella. Esimerkiksi 2E on toissijainen moduuli *Examples*. Myös nämä koodit ilmenevät yllä näkyvästä kuviosta 4. DOCX-ohjeiden kieliversiot olivat keskenään identtiset, joten merkinnät tein vain suomenkielisiin versioihin. Ohjeiden rakenne oli ennestään systemaattinen ja johdonmukainen, mikä helpotti sisällön modularisointia. Amentin mukaan yksilähteistämismahdollisuudet ovat sitä monipuolisempia mitä pienemmiksi moduuleiksi sisältö on purettu. Ajattelin kuitenkin myös, että liian suuri määrä pieniä moduuleja vaikeuttaisi sisällön organisointia julkaistaviksi kokonaisuuksiksi. Näin ollen, pyrin tietynlaiseen tasapainoon moduulien kokojen ja määrän suhteen.

Seuraavana askeleena tulee Amentin (2003, s. 28) prosessikuvauksen mukaan antaa tunnistetuille moduuleille etiketit (eng. *labels*). Etiketeissä pitää käyttää johdonmukaista syntaksia, joka ilmaisee moduulin tyypin sekä tarkalleen, mitä informaatiota moduuli sisältää. Lähdemateriaalissa oli käytetty yhdenmukaista otsikointilogiikkaa, joten etikettien luomiseen pystyin pitkälle hyödyntämään olemassa olevia otsikoita. Lisäksi merkkasin otsikoiden alkuun sen sovelluksen nimen, jota ohjeistus koski. Tällä tavoin moduulien ristiinkäyttö eri projektien, eli eri sovellusten käyttöohjeiden välillä pitäisi olla selkeämpää. Sellaiset moduulit, jotka koskivat kaikkia sovelluksia, merkitsin koodilla "Kaikki".

Kolmas askel on moduulien organisoiminen. Tämä vaihe alkaa toissijaisten moduulien yhdistämisellä ensisijaisiin moduuleihin, jonka jälkeen ensisijaiset moduulit erotetaan erillisiksi osioiksi. Amentin (2003, s. 30) opas sisältää tarkat ohjeistukset siitä, minkätyyppisiin ensisijaisiin moduuleihin eri toissijaiset moduulityypit voi liittää. Tämä on merkitty kuvioon 4 yllä. Myös moduulien organisoimisessa oli hyötyä käsittelemäni lähdemateriaalin johdonmukaisesta rakenteesta, jossa oli huomioitu eri informaatiotyyppien keskenäiset suhteet.

Neljäntenä ja viidentenä askeleena moduuleja rekonstruoidaan ja muokataan tulevia julkaisuvaihtoehtoja silmällä pitäen. Yksilähteistämisen edellytys ja modulaarisen

sisällön ominaisuus on, että moduulit toimivat itsenäisinä kokonaisuuksina, jotka ovat ymmärrettäviä missä kontekstissa tahansa (Ament, 2003, s. 6). Käsittelemäni käyttöohjeet oli alun perin luotu lineaarisina dokumentteina, eli sisältö noudatti tiettyä järjestystä. Näin ollen, tunnistamieni moduulien sisältöjä piti muokata modulaarisesti toimivaan muotoon. Tämän tein sekä suomenkielisiin että ruotsinkielisiin käyttöohjeisiin, jotta ne pysyisivät keskenään identtisinä.

Osana neljättä ja viidennettä askelta suunnittelin sisällön esittämistä eri laajuisena. Tässä ajatuksenani oli, että verkkojulkaisun loppukäyttäjä voisi valita minkä detaljitason ohjeistuksen hän näkee. Tätä varten jaoin moduulit kolmeen yksityiskohtaisuustasoon: *Large (L)*, *Medium (M)* ja *Small (S)*. Ylempi taso sisältää alemman tason / alemmat tasot, eli L:ään sisältyy M ja S, ja M:ään sisältyy S. Taulukossa 4 olen kuvaillut tämän suunnitelman skemaattiseen muotoon ja siitä ilmenee myös moduulityyppitasot sekä DITA:n perusinformaatiotyytit, jotka pääasiallisesti vastaisivat kyseistä yksityiskohtaisuustasoa.

**Taulukko 4.** Sisällön yksityiskohtaisuustasot.

	Large (+ M + S)	Medium (+ S)	Small
Käyttötaso	Hallitsee sovellusta täysin	Hallitsee sovelluksen käyttöä	Pikaohje
Tiedontarve	<i>Detailed knowledge</i>	<i>Good to know</i>	<i>Need to know</i>
Moduulin sisällöstä näytettävät tiedot	Kaikki tiedot tehtävän läpiviemiseksi + lisätiedot + liittyvät vaikuttajat	Kaikki tiedot tehtävän läpiviemiseksi + lisätiedot	Välttämättömät tiedot tehtävän läpiviemiseksi.
Moduulityyppitasot	Ensisijaiset, toissijaiset	Ensisijaiset, toissijaiset	Ensisijaiset
DITA:n perusinformaatiotyytit	Tehtävä Konsepti Viite	Tehtävä Konsepti	Tehtävä

Yksityiskohtaisuustasot merkkasin DOCX-ohjeeseen alla olevan kuvan 2 mukaisilla värikorostuksilla.

### Toimielimen tietojen julkaiseminen

Sekä organisaation omien päättävien toimielimien (päänäkymän ensimmäisellä välilehdellä), että muiden toimielimien (päänäkymän viimeisellä välilehdellä) tietoja voidaan julkaista.

### Julkaise toimielimen kokoonpano

1. Valitse päänäköymästä tilanteen mukaan **Toimielimet-** tai **Muut toimielimet -välilehti** (välilehtien nimet voivat poiketa asennuskohtaisesti).
2. Näkymän vasemman puolen puurakenteesta, valitse **toimielin**. Toimielimen perustiedot näkyvät näkymän oikean puolen yläosassa.
3. **Toimielintiedot julkaistaan -kenttään**, valitse **Kyllä**.
4. Näkymän oikean puolen alaosassa näkyy **Toimielimen julkaistavat tehtävät**. Rasti **Ei julkaista -** kenttää niiden roolien kohdalla, joita ei haluta näkyvän julkaisussa.
5. **Tallenna muutokset**.

**Kuva 2.** Yksityiskohtaisuustason merkinnät DOCX-muotoisessa käyttöohjeessa.

Kuvassa näkyvä keltainen korostus on ohjeen suppein versio eli *Small*. Tummanharmaa korostus on *Medium* ja vaaleanharmaa on laajin versio (*Large*), joka näyttää kyseisen ohjeistuksen kokonaisuudessaan. Tutkielman liite 3 näyttää, miltä kyseinen teksti näyttäisi eri yksityiskohtaisuustasoilla.

#### 4.2.2 Sisällön organisointi

Sisällön organisointivaihe pitää sisällään Amentin prosessikuvauksen askeleet kuusi ja seitsemän (Ament, 2003, s. 36) eli dokumenttien organisoinnin ja ristiviittaukset. Dokumentaation kolmiomallin osa-alueista, tämän vaiheen läpivienti liittyy eniten yksilähteistämiseen, mutta edellyttää niin sisällön modulaarisuutta kuin taustalla toimivaa rakenteisuutta tukevaa tekniikkaa. Tässä vaiheessa otin sisällönhallintajärjestelmän käyttöön eli vein DOCX-muotoisten käyttöohjeiden sisältöä ClickHelp-järjestelmään (ClickHelp, 2023a). Järjestelmässä on tiedonsiirtotoiminnallisuus, jonka avulla pystyy tuomaan sisältöä muista järjestelmistä,

muun muassa DOCX-tiedostoista. Testasin tätä ja tulos oli hyvä, joskin haluttu lopputulos esimerkiksi visualisuurien ja sisältöhierarkian osalta olisi vaatinut vielä melko laajasti tuodun sisällön muokkaamista. Varmistaakseni, ettei testimateriaalissa olisi tiedonsiirron mukana tulleita asetuksia tai muita vaikuttajia, en ClickHelp-järjestelmän testaamisessani käyttänyt tiedonsiirtoa, vaan vein testimateriaalin manuaalisesti järjestelmään.

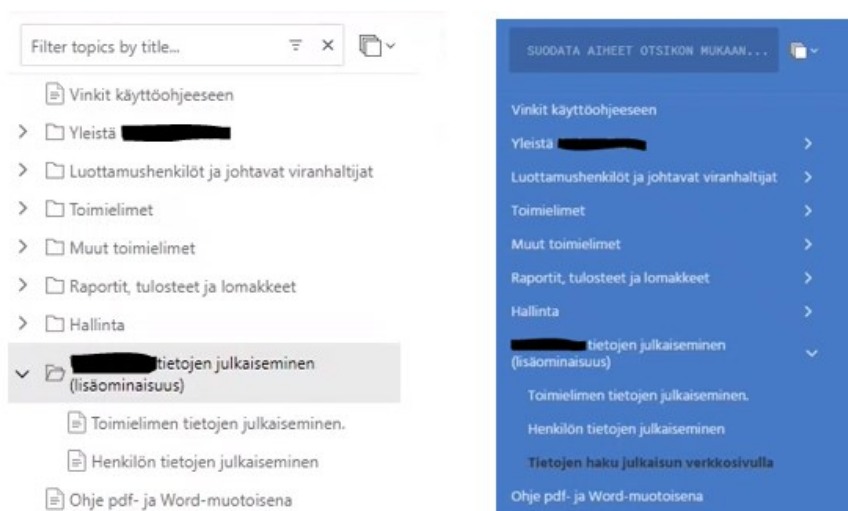
DITA:n periaatteiden mukaan muotoilu ja sisältö pitää erotella toisistaan (Syncro Soft, n.d.d). ClickHelp-järjestelmässä muotoilu määräytyy käytetyistä sivupohjista (eng. *templates*), joita pystyy jonkin verran muokkaamaan. Edistysellisempää räätälöintiä voi tehdä CSS -tiedostoja muokkaamalla ja sitä kautta tein jonkin verran muutoksia testissä käyttämäni perusmalliin.

Loin ensiksi kumpaakin käyttöohjetta varten projektin ja sitten projekteihin aiheet. ClickHelp-järjestelmässä sisällön käsittely tapahtuu editorissa, joka sisältää perinteisiä tekstinkäsittelytyökaluja. Niiden avulla loin tarvittavat elementit, kuten otsikot ja luettelot. Elementtien muotoilut kuitenkin tulivat sivupohjan määrityksistä ja olivat näin ollen johdonmukaisia koko projektin läpi. Editorissa lisäsin sisältöön myös kuvia, hyperlinkkejä sekä sisällön laajennus-/pienennys -työkaluja ja muita sisällön rakennetta selkeyttäviä elementtejä.

Sisällön organisointi koskee Amentin mukaan (2003, s. 36) ensivaiheessa moduulien organisointia asiakokonaisuuksiksi ja seuraavaksi asiakokonaisuuksien organisointia julkaistaviksi yhdistelmiksi. Huomionarvoista on, että julkaiseminen on tässä yhteydessä laajempi käsite kuin pelkästään konkreettinen julkaisutoiminto esimerkiksi verkkosivustolla; käsite pitää sisällään kaikkia niitä menetelmiä, joilla dokumentaatio voidaan saattaa lukijoille käyttöön. Organisoimisessa voidaan Amentin prosessikuvauksen mukaan noudattaa eri sisällön järjestämisstrategioita. Kuvauksessa (Ament, 2003, s. 36) mainitut strategiat ovat aakkosjärjestys, järjestys kohderyhmän mukaan, järjestys detaljitason mukaan, tärkeysjärjestys, järjestys sijainnin mukaan,

suoritusjärjestys sekä järjestys moduulityypin mukaan. Myös useamman strategian yhdistäminen on mahdollista. Käyttöohjeiden sisällön organisoimisessa oli luontevaa käyttää suoritusjärjestystä, johon yhdistin myös järjestystä kohderyhmän mukaan sekä sijainnin mukaan. Sijainti viittaa sovelluksen käyttöliittymän elementteihin. Tällä tavoin pyrin sisällön mahdollisimman toimivaan hierarkiaan. Amentin (2003, s. 38) mukaan tietojen hierarkia on ensiarvoisen tärkeä. Looginen hierarkia auttaa loppukäyttäjiä hahmottamaan osioiden keskenäisiä suhteita ja se pitää havainnollistaa käyttäjille sisällysluetteloiden avulla. Verkkosivustossa sisällysluetteloita tulee olla sekä koko sivuston sisällön että yksittäisten sivujen osalta.

Asiakokonaisuuksien organisoimisen toteutin ClickHelp-järjestelmään projektikohtaisena aihehierarkiana. Hierarkiatasoja loin kolme, joista ensimmäinen ja toinen taso olivat moduuleja (ClickHelp-järjestelmässä *Topic*). Nämä näkyivät projektin sisällysluettelona eli verkkosivustolle julkaistussa versiossa vasemman laidan päävalikkona. Tämä hierarkia näkyy kuvassa 3 alla. Kuvan vasemmalla puolella on editorinäkymä ja kuvan oikealla puolella näkyy sinitaustainen verkkosivulle julkaistu näkymä. Sovelluksen nimi on kuvassa peitetty.



**Kuva 3.** Aihehierarkia ClickHelp-järjestelmän editorissa ja julkaisuversiossa.

Kolmantena hierarkiatasona oli moduulien sisältämiä alalukuja, jotka erottelin otsikkotasojen avulla. Ne näkyivät verkkojulkaisussa erillisenä sivukohtaisena sisällysluettelona näkymän oikeassa reunassa (ClickHelp-järjestelmässä *Mini-TOC*). Nämä sisällysluettelot ovat osa Amentin prosessikuvauksen seitsemättä vaihetta, eli muun muassa niiden avulla toteutuivat ristiviittaukset.

Koska moduulien ja niiden alalukujen muodostamat asiakokonaisuudet oli luotu loogiseen hierarkiaan projektien eli kunkin sovelluksen käyttöohjeen sisään, projekti oli sellaisenaan yksi julkaistava yhdistelmä. Samasta projektista pystyi kuitenkin muodostamaan muitakin julkaisuyhdistelmiä, esimerkiksi eri kohderyhmiä varten. Tämä onnistui ehdollisella julkaisemisella eli määrittelemällä julkaistaville aiheille ehtoja (ClickHelp-järjestelmässä *Output Tags*) ja hyödyntämällä näitä ehtoja julkaistaessa. Myös manuaalinen, kertaluonteinen aiheiden valinta julkaisun yhteydessä oli mahdollista. Kolmas tapa organisoida sisältöä uudelleenkäytettäväksi, oli moduulien käyttö projektien välillä. Samaa uudelleenkäyttömahdollisuutta hyödynsin myös yksityiskohtaisempien tietojen osalta käyttämällä ClickHelp-järjestelmän *Snippet*-nimistä työkalua. Työkalun avulla pystyy tallentamaan valitun sisältöosion omaksi moduuliksi, ja uudelleenkäyttämään tätä moduulia muualla. Sillä tavalla pystyin luomaan pienemmänkin sisältöosan kerran ja näyttämään sen samasta lähteestä tarvittavissa dokumenteissa rajattomasti koko käyttöohjeportaalissa. *Snippet*-työkalua käytettäessä voi lähdetiedossa tarkistaa, mihin kaikkialle se on laitettu näkyviin sekä myös toisin päin; heijastetun tiedon osalta voi tarkistaa mistä lähteestä se näkyy.

### **4.2.3 Sisällön esittely**

Esittelyvaihe on dokumentaation kolmiomallin yksilähteistämiseen liittyvä, mutta perustuu modulaariseen sisältöön ja vaatii niin ikään rakenteisuutta tukevaa tekniikkaa. Se on yhtä kuin Amentin prosessikuvauksen kahdeksas askel, jossa dokumentit konvertoidaan eri esitysmuotoihin, kuten online-helpiin tai PDF-dokumenteiksi. Tällöin

se tarkoittaa sitä yksilähteistämistyyppiä, josta Ament (2003, s. 15) puhuu termillä *repurposing* (suomeksi vapaasti käännettynä "uudelleenkäyttö").

Tämän tein ClickHelp-järjestelmässä hyödyntämällä tarkoitukseen käytettävää julkaisuutoimintoa ja julkaisin sisällön verkkojulkaisuna, DOCX-muotoisena ja PDF-muotoisena. Järjestelmä tarjoaa muitakin esitysmuotoja, joita en kuitenkaan testannut. ClickHelp-järjestelmästä käsin pystyy luomaan formaattikohtaisia ulkoasu- ja muita määrittelyjä sekä julkaisuehtoja. Esimerkiksi jos sisältönä on video, sen voi julkaisuehdon avulla jättää pois DOCX-muotoisesta julkaisusta.

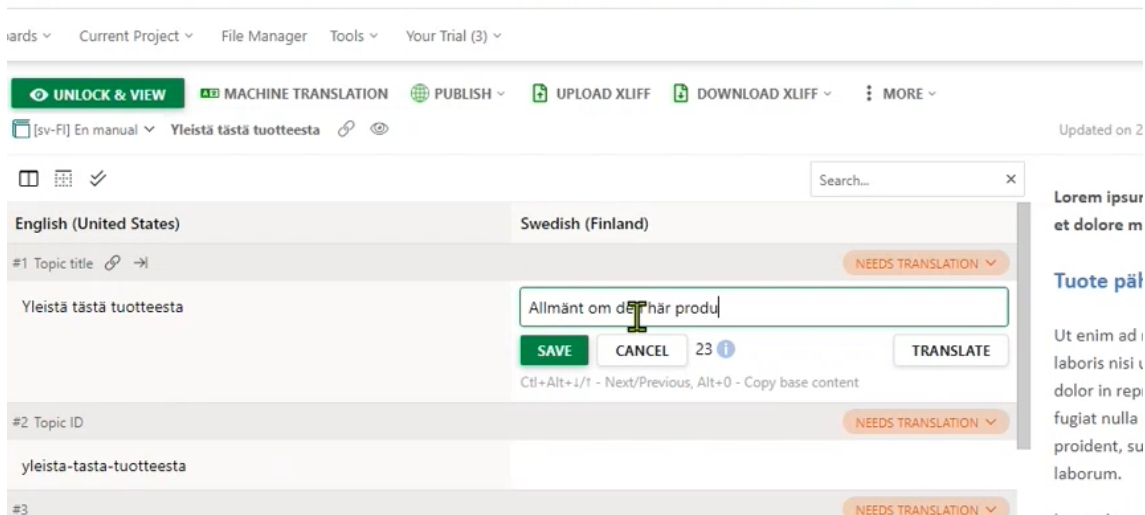
Kuten edelleä totesin, jo sisällön organisointivaiheessa pystyi huomioimaan tulevia aiheyhdistelmätarpeita ja hyöty tästä tuli konkreettisesti esille sisällön esittely- eli julkaisuvaiheessa. Tässä vaiheessa toteutui siis myös yksilähteistämisen toinen tyyppi, josta Ament (2003, s. 15) käyttää termiä *re-assembly* (suomeksi vapaasti käännettynä "uudelleenkoonpano").

#### **4.2.4 Dokumentaation versionhallinta**

Sisällönhallintajärjestelmän testaamiseen sisältyi myös käyttöohjeiden versionhallinnan testaaminen. Versionhallintatermiin sisällytän tässä toisaalta käyttöohjeiden kieliversioiden hallinnan ja toisaalta sovellusten eri tuoteversioiden käyttöohjeiden hallinnan.

Käyttöohjeista, joita käytin testaamisessa, oli olemassa suomenkieliset ja ruotsinkieliset versiot, jotka olivat substanssiltaan keskenään identtiset. ClickHelp-järjestelmässä on olemassa lokalisoituminto kieliversioiden hallintaan. Loin ensiksi suomenkielisen käyttöohjeversion ja lokalisoitumintoa käyttäen loin sen rinnalle ruotsinkielisen vastaavuuden omana projektina, jonka aiheet toiminnon ansiosta olivat kytkettyinä suomenkielisiin vastaavuuksiinsa. Aiheet muodostuivat kopioina alkuperäiskielen

aiheista eli sisältö oli suomen kielellä. Käännöstyö oli siis tehtävä sen jälkeen, ja sen tein manuaalisesti ClickHelp-järjestelmän omassa lokalisointieditorissa, ks. kuva 4 alla.



**Kuva 4.** ClickHelp-järjestelmän lokalisointieditori.

Järjestelmä tarjoaa myös mahdollisuuden käännöstyön suorittamiseen järjestelmän ulkopuolella, viemällä sisällön tiedonsiirtona ulos ClickHelpistä, tehdä käännöstyön halutulla tavalla, ja tuoda käännetyn sisällön takaisin. Tätä toimintoa en testannut.

Eri tuoteversioiden käyttöohjeiden hallintaan löytyi useampia toimintatapoja. Yksi tapa on vanhan käyttöohjeen eli projektin kopiointi ja kopion muokkaaminen uuden tuoteversion mukaiseksi. Tämä on siis ihan vastaava kuin Office-ohjelmien, esimerkiksi DOCX-muotoisten tiedostojen "Tallenna nimellä" -toiminto.

Toinen tapa hallinnoida versioita on ehdollisen julkaisemisen hyödyntäminen. Tällöin uuden ja vanhan tuoteversion käyttöohjeet julkaistaan samasta lähteestä. Eroavaisuudet merkitään ehdoilla, joiden avulla ohjataan, mikä tieto tulostuu mihinkäkin käyttöohjeeseen.

Kolmas toimintatapa on sovelluskohtaisten master-projektien käyttö. Tällöin luodaan tuotteen ensimmäinen käyttöohje tavallisena projektina. Tämä toimii master-projektina.



Seuraavan version käyttöohjetta varten luodaan uusi projekti, jossa uudelleenkäytetään niitä ensimmäisen käyttöohjeen aiheita, joihin ei ole tullut muutoksia uuden sovellusversion myötä. Aiheet, joiden sisältöön tulee uudessa versiossa uutta tietoa, luodaan uuteen käyttöohjeeseen uusina aiheina.

## **5 Sisällönhallintajärjestelmien vertailun ja testaamisen tulokset**

Tällä tutkimuksella tavoittelin kokonaisvaltaista näkemystä ohjelmistoalan loppukäyttäjille tarkoitetun dokumentaation hallintaan. Tässä luvussa kuvailen tekemieni analyysien tuloksia ja peilaan niitä alkuperäiseen tutkimusasetelmaan.

### **5.1 Sisällönhallintajärjestelmien vertailun tulokset**

Tämän tutkimuksen ensimmäisessä osiossa olen tutustunut kolmen komponenttisisällönhallintajärjestelmän ominaisuuksiin: ClickHelp (ClickHelp, 2023a), MadCap Flare (Madcap Software, 2023c) ja Oxygen XML Editor (Syncro Soft, 2023). Tutkimuksen tämän vaiheen tarkoitus oli selvittää, miten hyvin sisällönhallintajärjestelmien ominaisuudet tukevat rakenteisuuden, modulaarisuuden ja yksilähteistämisen periaatteita ja valita yksi näistä kolmesta sisällönhallintajärjestelmästä seuraavan tutkimusvaiheen testiin.

Tutkimuksen olen tehnyt vertailemalla järjestelmien tarjoamia toiminnallisuuksia DITA-informaatiomallin (Oasis Open, 2018; Syncro Soft, n.d.d) periaatteisiin (liite 1) sekä työpaikallani aikaisemmin suoritetuissa sisällönhallintajärjestelmien testauksissa laadittuun kriteeristöön (liite 2). Tutkimustulosteni yhteenveto näkyy taulukossa 5 alla.

**Taulukko 5.** Sisällönhallintajärjestelmien vertailun tulokset.

Ominaisuus	ClickHelp	MadCap Flare	Oxygen XML Editor
Muotoilun ja sisällön erotteleminen toisistaan	+1	+1	+1
Informaatiotyyppien mallien määrittelemine ja käyttö	+1	+1	+1
Sisällön jakaminen omiksi tiedostoiksi tallentuviksi moduuleiksi	+1	+1	+1
Sisällön johdonmukainen muotoilu erityyppisiksi elementeiksi	+1	+1	+1
Sisällön näyttäminen valitun laajuuden mukaan	0	0	0
Sisällysluettelo – koko sivusto	+1	+1	+1
Sisällysluettelo – yksittäinen sivu	+1	+1	+1
Sisällön näyttäminen eri kokonaisuuksien välillä ristiin	+1	+1	+1
Kirjanmerkit	+1	+1	+1
Kuvien uudelleenkäyttö	+1	+1	+1
Sisällön organisoiminen räätälöidyiksi julkaistaviksi kokonaisuuksiksi	+1	+1	+1
Sisällön ehdollinen julkaiseminen	+1	+1	+1
SVG- ja PNG-muotoisten kuvien käyttö	+1	+1	+1
Kuvien sijainnin ja koon säätäminen	+1	+1	+1
Multimediasisällön käyttö (esim. Flash-animaatiot ja videot)	+1	+1	+1
Kieliattribuuttien käyttö	+1	+1	+1

+1: ominaisuus on järjestelmässä olemassa

0: ominaisuus on järjestelmässä osittain olemassa

-1: ominaisuutta ei ole järjestelmässä

Vertailussa olivat kehittyneet ja monipuoliset komponenttisisällönhallintajärjestelmät, joten ei ole yllättävää, että kriteerit täyttyivät miltei sataprosenttisesti kaikissa kolmessa järjestelmässä. Ainoastaan yhdestä ominaisuudesta löysin huomionarvoisen eroavaisuuden; ominaisuus löytyi kaikista järjestelmistä, mutta ClickHelp:ssä se oli monipuolisempi kuin muissa. Kyseessä oli työkalut, jotka mahdollistavat sisällön näyttämisen valitun laajuuden mukaan. ClickHelpissä näitä työkaluja oli kolme erilaista, muissa järjestelmissä niitä oli kaksi. Vastauksena ensimmäiseen tutkimuskysymykseen voin vertailuni perusteella todeta, että sisällönhallintajärjestelmät tyypiltään komponenttisisällönhallintajärjestelmä tukevat hyvin rakenteisuutta, modulaarisuutta ja yksilähteistämistä.

Koska tutkitut sisällönhallintajärjestelmät olivat ominaisuuksiltaan hyvin tasavertaisia, toisen tutkimusosion testissä käytettävän sisällönhallintajärjestelmän valinta perustui myös sisällönhallintajärjestelmien hallintakäyttöliittymän haltuunotettavuuteen. Tähän vaikutti muun muassa toimintojen käytettävyyys, käyttöohjeiden selkeys ja käyttötuen saatavuus. Tämän perusteella lopullinen valintani oli ClickHelp-järjestelmä.

## **5.2 Sisällönhallintajärjestelmän tarjoamat hyödyt**

Tutkimuksen toisessa osiossa olen testannut tutkimuksen ensimmäisessä osiossa valittua komponenttisisällönhallintajärjestelmää käytännössä, rakentamalla siihen osan kahden sovelluksen käyttöohjeita. Testaamiseen valitsemani sisällönhallintajärjestelmä oli ClickHelp ja sen työkaluja ja toiminnallisuuksia hyödyntämällä olen pyrkinyt totetuttamaan rakenteisuuden, modulaarisuuden ja yksilähteistämisen periaatteiden mukaiset käyttöohjeet. Periaatteiden täyttymisen edellyttämät ominaisuudet ja niiden toteuttamiseen hyödyntämäni työkalut ClickHelp-järjestelmässä on kuvattu taulukossa 6 alla. Taulukossa käytän järjestelmän omia toimintojen nimiä, jotka ovat englanninkielisiä. Testimateriaalina olen käyttänyt Triplan Oy:n tuoteportfolion kahden sovelluksen käyttöohjeita.

**Taulukko 6.** Ominaisuuksien toteuttamiseen hyödynnetyt työkalut ClickHelp-järjestelmässä.

Ominaisuus	ClickHelp-järjestelmässä hyödynnetty työkalu
Muotoilun ja sisällön erotteleminen toisistaan	<i>Templates, CSS-tiedostojen räätälöinti</i>
Informaatiotyyppien mallien määrittäminen ja käyttö	<i>Templates, CSS-tiedostojen räätälöinti</i>
Sisällön jakaminen omiksi tiedostoiksi tallentuviksi moduuleiksi	<i>Topics, Snippets</i>
Sisällön johdonmukainen muotoilu erityyppisiksi elementeiksi	<i>Templates, CSS-tiedostojen räätälöinti, tekstinmuotoilutyökalut</i>
Sisällön näyttäminen valitun laajuuden mukaan	<i>Show More Box, Drop-down Text, Text Popup</i>
Sisällysluettelo – koko sivusto	<i>Table of Content</i>
Sisällysluettelo – yksittäinen sivu	<i>Mini-TOC</i>
Sisällön näyttäminen eri kokonaisuuksien välillä ristiin	<i>Reuse Topic, Reuse Snippet</i>
Kirjanmerkit	<i>Insert link, Anchors</i>
Kuvien uudelleenkäyttö	<i>Storage</i>
Sisällön organisoiminen räätälöidyiksi julkaistaviksi kokonaisuuksiksi	<i>Output Tags, julkaisukohtainen aiheiden valinta</i>
Sisällön ehdollinen julkaiseminen	<i>Conditional Content, Output Tags</i>
SVG- ja PNG-muotoisten kuvien käyttö	<i>Images</i>
Kuvien sijainnin ja koon säätäminen	<i>Image properties</i>
Multimediasisällön käyttö (esim. Flash-animaatiot ja videot)	<i>Insert media</i>
Kieliattribuuttien käyttö	<i>Translation Projects, Localization functionality</i>

Tämän tutkielman teoriaosassa luvussa 2.4 kuvailin kompleksisen tuoteportfolion käyttöohjeiden hallintaan liittyviä ongelmakohtia. Niistä voi tunnistaa viisi dokumentaation hallinnan haasteita aiheuttavaa päätekijää: eri kohderyhmien tiedontarpeet, tilannekohtaiset erilaajuiset tiedontarpeet, sovellusten väliset

riippuvuudet, sovellusten eri tuoteversiot sekä käyttöohjeiden eri kieliversiot. Seuraavaksi kuvailen, millä tavoin pystyin hyödyntämään sisällönhallintajärjestelmää käyttöohjeiden hallinnassa, kun tavoitteena oli näiden viiden osa-alueen haasteiden vähentäminen.

Sisällönhallintajärjestelmässä pystyi huomioimaan eri kohderyhmiä, julkaisemalla sisältöä yhdistettynä eri tavoin, yksilähteistämisen periaatteiden mukaisesti. Tämä onnistui useammalla eri tavalla; ehdollisella julkaisemisella sekä julkaistavien aiheiden manuaalisella valinnalla. Verkkojulkaisuun pystyi liittämään myös esimerkiksi DOCX- ja PDF-tiedostoja. Julkaisemalla käyttöohjeen ensiksi DOCX-tiedostoon ja liittämällä tiedoston vastaavaan verkkojulkaisuun käyttäjille ladattavaksi, olisi siis myös mahdollista antaa käyttäjien itse muokata ohjeesta omia organisaatiokohtaisia versioita.

Tilannekohtainen tiedontarpeen laajuus ei ole yksi yhteen käyttäjän roolin kanssa, vaan jokainen rooli tarvitsee toisinaan pikaohjeen ja toisinaan mahdollisimman yksityiskohtaiset tiedot. Sisällönhallintajärjestelmästä löytyi useampia niin sanottuja laajenna-/pienennä -työkaluja, joiden avulla sisältöä oli mahdollista esittää verkkojulkaisussa eri laajuisena. Sellaista sisäänrakennettua työkalua ei kuitenkaan ollut, joka olisi mahdollistanut sellaista tavoittelevaani Small-Medium-Large -mallia, jonka olen kuvannut luvussa 4.2.1 ja liitteessä 3. Tavoitteena oli, että käyttäjä roolistaan riippumatta pystyisi sujuvasti vaihtamaan ohjeen yksityiskohtaisuustasoa samassa näkymässä, esimerkiksi S-, M- ja L-painikkeita napauttamalla.

Sovellusten väliset riippuvuudet vaikuttavat käyttöohjeiden hallintaan siten, että osaa yhden sovelluksen ohjeistuksia saatetaan tarvita toisen sovelluksen käyttöön liittyen. Sisällönhallintajärjestelmässä sekä kokonaisia moduuleja että pienempiä osioita pystyi näyttämään myös eri projektien, eli eri käyttöohjeiden välillä ristiin. Tällä tavoin aiheen pystyi kirjoittamaan kerran, osana esimerkiksi projektia A, ja sisällyttää osaksi myös projektia B. Tällöin kyseisen aiheen sisällön päivittäminen tapahtuu molempien

projektien osalta yhdessä ja samassa paikassa eli projektissa A, mutta tehdyt muutokset näkyvät yhtä lailla projektissa B.

Tuotekehityksen myötä syntyy sovelluksista uusia versioita, jolloin käyttöohjeet pitää päivittää vastaamaan uusia versioita. Elleivät käyttäjät siirry kaikki samaan aikaan käyttämään sovelluksen uusinta versiota, pitää kuitenkin myös vanha käyttöohje olla saatavilla. Useimmiten muutoksia tulee vain osaan sovellusta eli suuri osa vanhasta käyttöohjeesta pätee myös uudessa sovellusversiossa.

Sisällönhallintajärjestelmässä käyttöohjeiden versiointi onnistui useammalla tavalla. Vanhan käyttöohjeen kopiointi uuden ohjeen pohjaksi oli yksi tapa, jossa kuitenkin syntyy tilanne, että samoja tietoja on ylläpidettävä kahdessa paikassa ja ajan mittaan hallitsemattoman monessa paikassa. Tämä tapa voisi tulla kysymykseen, jos uusi sovellusversio eroaa niin paljon edellisistä, että se vaikuttaa koko käyttötapaan, minkä seurauksena käyttöohje muuttuu radikaalisesti.

Ehdollinen julkaiseminen oli toinen versionhallintamahdollisuus. Tässä toimintatavassa ohjataan julkaisuehtojen avulla sitä, mikä tieto tulostuu mihinkäkin julkaistavaan käyttöohjeeseen. Julkaisuehtoja voi asettaa kokonaisuun aiheisiin tai yksittäisiin tietoihin, kuten sanoihin, lauseisiin ja kuviin. Kokonaisuutta, eli monitasoiset muut projektien väliset vuorovaikutukset ja esimerkiksi eri kohderyhmiä sekä sovellusten välisiä riippuvuuksia ajatellen, tämä tapa saattaisi pitkän päällä aiheuttaa melko vaikeasti hallittavan palapelin.

Testini perusteella kallistuisin versionhallinnan osalta kolmanteen toimintatapaan, jossa uusi käyttöohje luodaan uutena projektina. Projektissa kuitenkin käytetään vanhan käyttöohjeen aiheita siltä osin, kun sama tieto pätee uudessakin versiossa. Tällä tavoin uuden ja vanhan käyttöohjeversion yhteiset tiedot ylläpidetään vain yhdessä paikassa. Tässä uudelleenkäyttötavassa sisältö heijastuu sellaisenaan master-projektista muihin projekteihin. Toimintatavassa on siis myös se etu, että sisällöntuottaja näkee tiedot siinä

kontekstissa, jota hän on päivittämässä. Toisesta projektista heijastettua tietoa hän ei voi kuitenkaan muuttaa vahingossa, koska se näkyy vain lukutilassa.

Käyttöohjeiden eri kieliversioiden hallintaan sisällönhallintajärjestelmä tarjosi kielilokalisointitoiminnon. Sen ansiosta kunkin kielen sisällölle tuli oma kieliattribuutti. Erikielisten aiheiden väliin muodostui kytkentä, joka mahdollisti verkkojulkaisussa niin sanotun ”kielenvaihdon lennossa”, eli kun dokumentaation loppukäyttäjä vaihtaa sivun kieltä, se vie toisen kielen vastaavaan aiheeseen eikä esimerkiksi aloitussivulle. Lokalisointitoiminnon erillinen editori helpotti käännösprosessin hallinnointia.



## 6 Loppupohdinnat

Tutkielmassani tavoitteena oli selvittää, onko rakenteisuuden, modulaarisuuden ja yksilähteistämisen periaatteita tukevasta sisällönhallintajärjestelmässä apua kompleksisen sovelluskäyttöohjekokonaisuuden hallinnassa.

Tutkimusta varten kartoitin ensiksi markkinoilla olevaa sisällönhallintajärjestelmätarjontaa ja valitsin kolme komponenttisisällönhallintajärjestelmää, joiden ominaisuuksia tutkin tarkemmin. Tämän suoritin kvantitatiivisena tutkimuksena asiantuntija-arvioinnin muodossa. Vertailemani kolme järjestelmää olivat MadCap Flare (Madcap Software, 2023c), ClickHelp (ClickHelp, 2023a) ja Oxygen XML Editor (Syncro Soft, 2023). Ensimmäisen tutkimusosion päätteeksi valitsin niistä Click Help -järjestelmän seuraavassa tutkimusosiossa testattavaksi komponenttisisällönhallintajärjestelmäksi.

Tutkimuksen toisessa osiossa rakensin osan kahden sovelluksen käyttöohjeita kahdella kielellä Click Help -järjestelmään, mahdollisimman tehokasta käyttöohjekokonaisuuden hallintamallia tavoitellen. Tämän tein järjestelmätestauksen ja hyväksymistestauksen menetelmiä soveltaen.

Tutkimukseni kysymykset olivat:

- 1) Miten hyvin rakenteisuus, modulaarisuus ja yksilähteistäminen toteutuvat sisällönhallintajärjestelmissä a, b ja c?
- 2) Miten kompleksisen ohjelmistotuoteportfolion tuomia haasteita käyttöohjeiden hallinnassa voidaan ratkaista dokumentaation rakenteisuudella, modulaarisuudella ja yksilähteistämällä?

Vastausta ensimmäiseen tutkimuskysymyksen hain vertailemalla sisällönhallintajärjestelmien ominaisuuksia DITA-informaatiomallin (Oasis Open, 2018; Syncro Soft, n.d.d) periaatteisiin sekä työpaikallani aikaisemmin laadittuun kriteeristöön

pohjautuvaan kuudentoista kriteerin luetteloon. Vertailun tein ensivaiheessa järjestelmien tuotekuvausten ja käyttöohjeiden perusteella ja jatkeena siitä testaamalla toiminnallisuuksia järjestelmissä. Kaikki kolme tutkittavaa järjestelmää oli tyypiltään komponenttisisällönhallintajärjestelmiä eli juuri uudelleenkäytettävään tekniseen dokumentaatioon suunniteltuja, joten niissä täyttyi odotetusti kaikki asettamani kuusitoista kriteeriä. Tutkimukseni toiseen vaiheeseen valitsin niistä sen, jonka hallintakäyttöliittymä ja käyttöohjeet olivat selkeimmät.

Toiseen tutkimuskysymykseen pyrin saamaan vastauksia rakentamalla osan kahden sovelluksen käyttöohjeita Click Help-komponenttisisällönhallintajärjestelmään. Testaus perustui samaan kuudentoista kriteerin luetteloon kuin tutkimuksen ensimmäisen osion arviointi. Tämän vaiheen aloitin muuntamalla sovellusten olemassa olevat lineaariset käyttöohjeet modulaariseen muotoon, Amentin kymmenen askeleen prosessikuvausta soveltaen. Tämän jälkeen vein käyttöohjeet Click Help -järjestelmään sen rakenteisuutta, modulaarisuutta ja yksilähteistämistä tukevia toiminnallisuuksia hyödyntäen. Tämä tutkimuskysymys oli moniulotteisempi ja näin ollen vastaus ei ole suoraviivainen, vaan perustuu useamman eri tekijän summaan. Tutkimuksen tämän osion tavoitteena oli selvittää komponenttisisällönhallintajärjestelmän tuomaa helpotusta niillä osa-alueilla, joista aiheutuu dokumentaation hallinnassa haasteita. Näillä osa-alueilla olin tunnistanut viisi haasteita aiheuttavaa päätekijää: eri kohderyhmien tiedontarpeet, tilannekohtaiset erilaajuiset tiedontarpeet, sovellusten väliset riippuvuudet, sovellusten eri tuoteversiot sekä käyttöohjeiden eri kieliversiot. Tutkimukseni osoitti, että komponenttisisällönhallintajärjestelmä tarjoaa käyttöohjeiden hallintaan helpotusta kaikilla näillä osa-alueilla.

Eri kohderyhmille räätälöityjä käyttöohjeita pystyi luomaan ja julkaisemaan samasta lähteestä useamman eri toiminnon avulla. Myös tilannekohtaisia, erilaajuisia tiedontarpeita pystyi huomioimaan jakamalla sisältöä pienempiin, eri työkalujen avulla avattaviin ja suljettaviin osiin. Sellaista valmista toiminnallisuutta ei kuitenkaan ollut, joka olisi mahdollistanut koko ohjeen yksityiskohtaisuustason vuorottelemista yhdellä

napautuksella samassa näkymässä. Sovellusten väliset riippuvuudet aiheuttavat tarpeen heijastaa käyttöohjeiden aiheita muidenkin sovellusten käyttöohjeisiin. Tähänkin löytyi sisällönhallintajärjestelmästä hyvin toimivia yksilähteistämisperiaatteiden mukaisia ratkaisuja. Samoja toiminnallisuuksia pystyi hyödyntämään käyttöohjeiden versioinnissa. Versiointiin löytyi useampia toimintatapoja. Käytettävä tapa määräytyy muun muassa muutosten laajuudesta ja siitä, onko tarve uuden ja vanhan käyttöohjeen rinnakkaiskäytölle. Käyttöohjeiden kieliversioita pystyi järjestelmässä hallinnoimaan tarkoitukseen kehitetyllä kielilokalisointitoiminnolla, joka sisälsi myös käänösprosessin seurantaan helpottavia ominaisuuksia.

Yhteenvetona voin siis todeta, että tutkimukseni perusteella vastaus ensimmäiseen tutkimuskysymykseen on, että rakenteisuus, modulaarisuus ja yksilähteistäminen toteutuvat sisällönhallintajärjestelmissä hyvin, kun ne ovat tyypiltään komponenttisisällönhallintajärjestelmiä. Vastaus toiseen tutkimuskysymykseen on tutkimuksen perusteella, että kompleksisen ohjelmistotuoteportfolion tuomia haasteita käyttöohjeiden hallinnassa ovat osittain ratkaistavissa dokumentaation rakenteisuudella, modulaarisuudella ja yksilähteistämällä. Pelkkä tarkoitukseen kehitetyn sisällönhallintajärjestelmän käyttö ei kuitenkaan tuo ratkaisua, vaan kuten varsinkin toisen tutkimusosion prosessista voi päätellä, sisällön suunnittelu ja kokonaisvaltainen organisointi ovat vähintään yhtä tärkeitä.

Pidän tutkimuksessa käytettyjä menetelmiä siinä mielessä onnistuneina, että ne muodostivat polun, jonka avulla pystyin saamaan vastausta tutkimuksen ydinkysymykseen. Tutkimukseen olisi mahdollisesti voinut valita tieteellisempiä analyysimenetelmiä, mutta aiheen konkreettisen luonteen takia konkreettisten analyysimenetelmien käyttö oli perusteltua. Tutkimustuloksista voi olla hyötyä käyttöohjeiden haasteiden kanssa kamppailevissa organisaatioissa. Tieteellisempien analyysimenetelmien käyttö olisi luultavasti tarkoittanut tutkimuksessa eri lähestymiskulmaa. Tällöin myös tutkimusasetelma ja tutkimuskysymykset olisivat muuttuneet.

Komponenttisisällönhallintajärjestelmissä on eroavaisuuksia ja tässä tutkittiin niitä kolme, joista yksi testattiin perusteellisemmin. Tutkimuksesta ei voi näin ollen vetää johtopäätöstä, että tulos pätee kaikkiin komponenttisisällönhallintajärjestelmiin. Tutkimuksen rajoitteena voi myös pitää sen, että suoritin sen kokonaisuudessaan yksin. Tutkimusta olisi siis hyvä jatkaa tai valituin osin toistaa useamman henkilön toimesta.

Tutkielmassani tarkastelin käyttöohjeiden **hallinnan** haasteisiin liittyviä ratkaisuja. Luontevaa olisi seuraavaksi tutkia, miten hallinnan haasteiden väheneminen vaikuttaa käyttöohjeiden **käyttöön**. Dokumentaation testaaminen sisältyykin Amentin prosessikuvaukseen ja mielestäni esimerkiksi komponenttisisällönhallintajärjestelmällä tuotetun käyttöohjeen käytettävyys- tai käyttäjäkokemustestaus voisi antaa arvokasta tietoa käyttöohjeiden kehittämiseen.

## Lähteet

- Achtelig, M. (2022a). *Help authoring tools and content management systems for technical documentation*. Indoition. Noudettu 29.10.2022 osoitteesta <https://www.indoition.com/online-help-authoring-tools-survey.htm>
- Achtelig, M. (2022b). *Top 40: Best Technical Documentation Software of 2022*. Indoition. Noudettu 29.10.2022 osoitteesta <https://www.indoition.com/best-technical-documentation-software.htm>
- Airlines for America. (28.6.2019). *S1000D: International specification for technical publications*. Noudettu 20.10.2022 osoitteesta <https://publications.airlines.org/CommerceProductDetail.aspx?Product=158>
- Ament, K. (2002). *Single Sourcing: Building Modular Documentation*. William Andrew Publishing.
- Andersen, R., & Batova, T. (2015). The Current State of Component Content Management: An Integrative Literature Review. *IEEE transactions on professional communication*, 58(3), 247–270. <https://doi.org/10.1109/TPC.2016.2516619>
- Bailie, R. A., & Huset, J. (2015). The Effect of CMS Technology on Writing Styles and Processes: Two Case Studies. *IEEE Transactions on Professional Communication*, 58(3), 309–327. <https://doi.org/10.1109/TPC.2016.2516642>
- Barnum, C., & Carliner, S. (1993). *Techniques for Technical Communicators*. Macmillan.
- Blackler, A., Gomez, R., Popovic, V., & Thompson, M. H. (2018). Life is too short to RTFM: how users relate to documentation and excess features in consumer products. *Interacting with Computers*, 30(6), 27–46. <https://doi.org/10.1093/iwc/iwy021>
- British Standards Institution. (2023). *Guide to user's requirements for technical manuals*. BSI.knowledge. Noudettu 2.1.2023 osoitteesta <https://knowledge.bsigroup.com/products/guide-to-users-requirements-for-technical-manuals-based-on-the-principles-of-bs-4884-presentation/standard>
- ClickHelp. (2023a). *ClickHelp Online Documentation Tool*. Noudettu 17.1.2023 osoitteesta <https://clickhelp.com/>

- ClickHelp. (2023b). *Output Tags*. Noudettu 21.3.2023 osoitteesta <https://clickhelp.com/software-documentation-tool/user-manual/output-tags.html>
- DITA Open Toolkit. (n.d.). *DITA Open Toolkit*. Noudettu 5.12.2022 osoitteesta <https://www.dita-ot.org/>
- Euroopan Parlamentti ja Neuvosto (EU). (2016). *EU:n direktiivi julkisen sektorin elinten verkkosivustojen ja mobiilisovellusten saavutettavuudesta*. Noudettu 1.11.2022 osoitteesta <https://eur-lex.europa.eu/legal-content/FI/TXT/HTML/?uri=CELEX:32016L2102&from=FI>
- Evia, C., Sharp, M. R., & Perez-Quinones, M. A. (2015). Teaching Structured Authoring and DITA Through Rhetorical and Computational Thinking. *IEEE Transactions on Professional Communication*, 58(3), 328–343. <https://doi.org/10.1109/TPC.2016.2516639>
- Ford, J. D., & Mott, R. K. (2007). The Convergence of Technical Communication and Information Architecture: Creating Single-source Objects for Contemporary Media. *Technical Communication*, 54(3), 333–342.
- Ganesh, K. (2014). *Enterprise resource planning: Fundamentals of design and implementation*. Cham: Springer.
- Goldfarb, C. F. (1992). *The SGML Handbook*. Oxford University Press.
- Hackos, J. T. (2016). International Standards for Information Development and Content Management. *IEEE transactions on professional communication*, 59(1), 24–36.
- Hamilton, M. (13.4.2022). *What is DITA and How Does it Differ From MadCap Flare?* MadCap Software. Noudettu 13.12.2022 osoitteesta <https://www.madcapsoftware.com/blog/what-is-dita-and-how-does-it-differ-from-madcap-flare/>
- Hashimoto, K., Buschiazzo, R., Bradbury, J., Marshall, T., Socher, R., & Xiong, C. (2020). A High-Quality Multilingual Dataset for Structured Documentation Translation. *arxiv.org*, DOI: <https://doi.org/10.48550/arXiv.2006.13425>
- Horn, R. E. (1978). How to write information mapping. *IEEE Transactions on Professional Communication*, PC-21(1), 39. <https://doi.org/10.1109/TPC.1978.6592441>

- Horn, R. E. (1998). *Structured Writing as a Paradigm*. *Educational Technology Publications*. Noudettu 11.12.2022 osoitteesta <https://faculty.washington.edu/farkas/TC510-Fall2011/Horn-StructuredWritingParadigm.pdf>
- International Organization for Standardization. (n.d.). *Standards*. Noudettu 26.12.2022 osoitteesta <https://www.iso.org/standards.html>
- Isohella, S. (2019). *Metodologiset ja teoreettiset lähtökohdat* [video] YouTube. Noudettu 2.4.2023 osoitteesta <https://www.youtube.com/watch?v=my1qJY2mTxI>
- Isohella, S. (2021). *Tekninen viestintä käytännössä ja tekninen dokumentti käytössä* [luentomateriaali]. Moodle [Rajattu pääsy].
- Isohella, S., Suojanen, T., & Virtaluoto, J. (2018). Minimalismiin pohjautuvan dokumentointiprosessimallin kehittäminen. *VAKKI Publications*, 9, 187–200.
- iTeh, Inc. (2014). *Preparation of instructions—Structuring, content and presentation*. Noudettu 26.12.2022 osoitteesta <https://standards.iteh.ai/catalog/standards/clc/fd6d923f-0532-46db-9a4b-ea2e28f12e3a/en-62079-2001>
- Jones, S. (26.11.2018). *5 Types of Content Management Systems (CMS)*. Ixiasoft. Noudettu 18.12.2022 osoitteesta <https://www.ixiasoft.com/types-of-content-management-systems/>
- Kasurinen, J. P. (2013). *Ohjelmistotestauksen käsikirja* (1. p.). Docendo.
- Kimber, E. (2017). *What is DITA?*. XML.com. Noudettu 29.11.2022 osoitteesta <https://www.xml.com/articles/2017/01/19/what-dita/>
- Kratky, R. (27.9.2017). *Modular documentation: How to make both writers and users happy*. Opensource.com. Noudettu 2.12.2022 osoitteesta <https://opensource.com/article/17/9/modular-documentation>
- Kyrö, P. (2014). *Tieteellinen tutkimusprosessi*. Metodix. Noudettu 2.4.2023 osoitteesta <https://metodix.fi/2014/05/17/kyro-paula-tieteellinen-tutkimusprosessi/>
- Madcap Software. (2023a). *Associating Conditions With Targets*. Noudettu 21.3.2023 osoitteesta

- <https://help.madcapsoftware.com/flare2022r3/Content/Flare/Conditions/Process/Associating-Conditions-Targets.htm>
- Madcap Software. (2023b). *Madcap Central*. Noudettu 17.1.2023 osoitteesta <https://www.madcapsoftware.com/products/central/>
- Madcap Software. (2023c). *Madcap Flare*. Noudettu 17.1.2023 osoitteesta <https://www.madcapsoftware.com/products/flare/>
- Markel, M. (2012). *Technical communication* (10. p.). Bedford.
- Nikunen, E. (1996). Rakenteinen teksti—Tekstin käsittelyä tietokoneella. *AFinLAN vuosikirja 1996*, 54, 107–118.
- Oasis Open. (2018). *Darwin Information Typing Architecture (DITA) Version 1.3 Part 0: Overview*. Noudettu 21.10.2022 osoitteesta <http://docs.oasis-open.org/dita/dita/v1.3/dita-v1.3-part0-overview.html>
- Ovaska, S., Aula, A., & Majaranta, P. (2005). *Käytettävyytutkimuksen menetelmät* (B-2005-1). Tampereen yliopisto, tietojenkäsittelytieteiden laitos.
- Palola, I. (2022). *Rakenteinen teksti ja sen sovellukset* [luentomateriaali]. Moodle [Rajattu pääsy]. Noudettu 1.12.2022 osoitteesta [https://moodle.uwasa.fi/pluginfile.php/595719/mod\\_resource/content/6/Rakenteinen%20teksti%202022%2C%20johdanto.pdf](https://moodle.uwasa.fi/pluginfile.php/595719/mod_resource/content/6/Rakenteinen%20teksti%202022%2C%20johdanto.pdf)
- RajatOn. (2015). *Tutkijan ABC*. Tee tutkimus. Noudettu 15.11.2022 osoitteesta <https://rajatontatiedekasvatusta.wordpress.com/tutkijan-abc/>
- Rajlich, V. (2011). *Software Engineering*. CRC Press LLC.
- Redish, J. G. (2010). Technical Communication and Usability: Intertwined Strands and Mutual Influences Commentary. *IEEE Transactions on Professional Communication*, 53(3). <https://doi.org/10.1109/TPC.2010.2052861>
- SFS. (n.d.). *SFS-EN 62079*. Noudettu 12.12.2022 osoitteesta <https://sales.sfs.fi/fi/index/tuotteet/SFSsahko/CENELEC/ID6/6/10570.html.stx>
- Shivakumar, S. K. (2017). *Enterprise Content and Search Management for Building Digital Platforms* (1. p.). Wiley-IEEE Press.
- Suomivuori, T., Virtualuoto, J., & Suojanen, T. (2020). Applying Minimalism in the Real World: Results From a Workshop. *VAKKI Publications*, 12, 225–238.



- Syncro Soft. (2023). *Oxygen XML Editor*. Noudettu 28.12.2022 osoitteesta [https://www.oxygenxml.com/xml\\_editor.html](https://www.oxygenxml.com/xml_editor.html)
- Syncro Soft. (n.d.a). *Content Management System*. Noudettu 28.12.2022 osoitteesta [https://www.oxygenxml.com/dita/styleguide/Documentation\\_Process/c\\_Content\\_Management\\_Systems.html](https://www.oxygenxml.com/dita/styleguide/Documentation_Process/c_Content_Management_Systems.html)
- Syncro Soft. (n.d.b). *Oxygen XML Editor*. Oxygen XML. Noudettu 13.1.2023 osoitteesta <https://www.oxygenxml.com/>
- Syncro Soft. (n.d.c). *Profiling and Conditional Text*. Oxygen XML. Noudettu 21.3.2023 osoitteesta <https://www.oxygenxml.com/doc/versions/25.1/ug-editor/topics/profiling-conditional-text.html?hl=profiling%2Cconditional%2Ctext>
- Syncro Soft. (n.d.d). *The DITA Style Guide*. Oxygen XML. Noudettu 17.12.2022 osoitteesta <https://www.oxygenxml.com/dita/styleguide/>
- Syncro Soft. (n.d.e). *The DITA Style Guide / Information types*. Oxygen XML. Noudettu 17.12.2022 osoitteesta [https://www.oxygenxml.com/dita/styleguide/Topics\\_and\\_Information\\_Types/c\\_Information\\_Types\\_Explained.html](https://www.oxygenxml.com/dita/styleguide/Topics_and_Information_Types/c_Information_Types_Explained.html)
- TCBok. (n.d.). *Information Mapping*. Society for Technical Communication. Noudettu 1.12.2022 osoitteesta <https://www.tcbok.org/principles-and-practices/information-mapping/>
- Triplan. (2023). *Triplan Oy*. Noudettu 7.4.2023 osoitteesta <https://triplan.fi/>
- Tutorialspoint. (n.d.). *Learn XML*. W3. Noudettu 5.12.2022 osoitteesta [https://www.tutorialspoint.com/xml/xml\\_overview.htm](https://www.tutorialspoint.com/xml/xml_overview.htm)
- Valli, R. (2018). *Ikkunoita tutkimusmetodeihin 2* (5. p.). PS-kustannus.
- W3. (2018). *Web Content Accessibility Guidelines (WCAG) 2.1*. Noudettu 18.10.2022 osoitteesta <https://www.w3.org/TR/WCAG21/>
- W3. (n.d.). *A brief SGML tutorial*. Noudettu 4.11.2022 osoitteesta <https://www.w3.org/TR/WD-html40-970708/intro/sgmltut.html>
- W3schools. (n.d.a). *HTML Tutorial*. Noudettu 12.10.2022 osoitteesta [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)

W3schools. (n.d.b). *XML Tutorial*. Noudettu 12.10.2022 osoitteesta  
<https://www.w3schools.com/xml/default.asp>

Wingkvist, A., Ericsson, M., & Löwe, W. (2011). A Visualization-based Approach to Present and Assess Technical Documentation Quality. *Electronic Journal of Information Systems Evaluation*, 14(1), 150–159.

## Liitteet

### Liite 1. DITA-informaatiomallin kategoriat ja niiden edellyttämät ominaisuudet

Kategorioiden otsikot on vapaasti käännetty suomeksi, ja *The DITA Style Guide*:n alkuperäinen englanninkielinen termi näkyy perässä suluissa (Syncro Soft, n.d.d).

Kategoria (Syncro Soft, n.d.d)	Kuvaus	Sisällönhallintajärjestelmän ominaisuus, joka kategorian periaatteiden toteuttaminen edellyttää
<b>Informaatiotyytit ja aiheet</b> (eng. <i>Informationtypes and topics</i> )	Kuvataan informaatiotyyppien, eli muun muassa konseptin, tehtävän ja viitteen käyttöperiaatteita, sekä huomioitavat seikat jaettaessa informaatiota itsenäisiin aiheisiin, eli moduuleihin.	Järjestelmässä pitää pystyä käyttämään tai määrittelemään eri informaatiotyyppien malleja /templateja. Sisältö pitää pystyä jakamaan omiksi tiedostoiksi tallentuviksi moduuleiksi.
<b>DITA karttatiedostot</b> (eng. <i>DITA map files</i> )	Kuvataan DITA:n karttatiedostojen käyttö julkaistavien aiheiden hierarkian ja suhteiden määrittelyissä sekä aiheiden välisten linkkien kontrolloinnissa.	Järjestelmässä pitää pystyä organisoimaan sisältöä räätälöidyiksi julkaistaviksi kokonaisuuksiksi.
<b>Syntaksi ja mark-up</b> (eng. <i>Syntax and mark-up</i> )	Kuvataan, mitä DITA-elementeissä tulee huomioida, ja miten niitä organisoidaan. Näitä elementtejä ovat esimerkiksi lyhytkuvaukset, luettelot, kappaleet, prosessit, taulukot ja fraasit.	Järjestelmässä pitää olla työkalut, millä sisältöä pystyy johdonmukaisesti muotoilemaan erityyppisiin elementteihin
<b>Kieli ja välimerkit</b> (eng. <i>Language and punctuation</i> )	Kuvataan periaatteita koskien muotoilua, välimerkkejä, otsikointia ja sisällön kielimäärityksiä.	Järjestelmässä pitää olla tuki sisällön kieliattribuutin käyttöön. Järjestelmässä pitää pystyä erottelemaan muotoilu ja sisältö toisistaan.
<b>Grafiikka ja kuviot</b> (eng. <i>Graphics and figures</i> )	Kuvataan periaatteita koskien kuvien ja	Järjestelmässä pitää olla tuki SVG ja PNG-muotoisten kuvien käyttöön, sekä niiden

<b>Kategoria</b> (Syncro Soft, n.d.d)	<b>Kuvaus</b>	<b>Sisällönhallintajärjestelmän ominaisuus, joka kategorian periaatteiden toteuttaminen edellyttää</b>
	multimediasisällön formaattia ja käyttöä.	sijoittamiseen ja koon säätämiseen liittyvät työkalut. Järjestelmän pitää myös tukea multimediasisältöä, kuten Flash animaatioita ja videoita.
<b>Ristiviittaukset</b> (eng. <i>Cross-referencing</i> )	Kuvataan erityyppisiä ristiviittauksia, eli linkityksiä, sekä niiden soveltamista eri tilanteissa.	Järjestelmässä pitää pystyä linkittämään sisällön osioita toisiinsa niin, että jos lähdetieto (otsikko) muuttuu, myös kohdetieto (linkkiotsikko) muuttuu.
<b>Sisällön uudelleenkäyttö</b> (eng. <i>Content re-use</i> )	Kuvataan periaatteita koskien sisällön uudelleenkäyttöä eri konteksteissa.	Järjestelmästä pitää pystyä käsittelemään moduuleja niin, että ne ovat julkaistavissa eri kokoonpanoina.
<b>Metadata, käsittely ehtojen avulla, sekä indeksointi</b> (eng. <i>Metadata, conditional processing, and indexing</i> )	Kuvataan, miten elementtien metadataa pystyy hyödyntämään ehdoissa ja suodatuksissa niin, että sama sisältö voidaan räätälöidä esimerkiksi kohderyhmän tai julkaisualustan mukaan.	Järjestelmässä pitää pystyä määrittelemään elementeille ehtoja, joiden avulla sisältö julkaistuu tietyllä tavalla, esimerkiksi tietyllä terminologialla.

## Liite 2. Käytettävyydestauksen kriteeristö, Triplan Oy (2020)

Tähän tutkielmaan eniten liittyvät pääkategoriat on korostettu tummemmalla taustavärillä. Triplanin testattavien sisällönhallintajärjestelmien nimet ja pisteytykset on tässä poistettu.

Kategoria	Ominaisuus	Käytettävyys ylläpitäjälle	Käytettävyys loppukäytäjälle
Navigointi ja tiedon nopea löytäminen	Sisällysluettelo koko ajan näkyvässä - koko sivusto		
	Sisällysluettelo koko ajan näkyvässä - yksittäinen sivu		
	Ankkurit (kirjainmerkit)		
	Cross-Reference. Hyperlinkki, jonka linkkiteksti päivittyy, kun source-linkkitekstiä muutetaan.		
	Navigointipolku näkyvässä (murupolku)		
	<b>Expandable blocks 1.</b> Laajennettu teksti aukeaa omaksi kappaleeksi +merkkiä tms. klikkaamalla.		
	<b>Expandable blocks 2.</b> Laajennettu teksti aukeaa omaksi kappaleeksi hotspottia klikkaamalla.		
	<b>Expandable blocks 3.</b> Laajennettu teksti aukeaa tooltippinä.		
	Hyvät hakutoiminnot		
Yksilähteistäminen	<b>Yksilähteistäminen 1.</b> Sisältömoduulille määriteltävä ehto, jonka avulla samasta topicista voi heijastua eri tietoja eri outputteihin.		
	<b>Yksilähteistäminen 2.</b> Topicin tekstinpätkä voidaan näyttää toisessa topicissa.		
	<b>Yksilähteistäminen 3.</b> Toistuva lyhyt teksti, joka näkyy useammassa paikassa, mutta on vain kerran ylläpidettävänä, esim. softaversion nimi/nro.		
	<b>Yksilähteistäminen 4.</b> Sama ohjesivu voi olla näkyvässä useammassa eri menussa		
	<b>Single Source 5.</b> Single Source Image Editing		
	<b>Yksilähteistäminen 6.</b> Pääsy suoraan käyttöohjeen oikeaan kohtaan Triplanin järjestelmistä (online-help)		
	Tuki mobiililaitteille		
	Word-export mahdollisuus valituista osioista		

Kategoria	Ominaisuus	Käytettävyys ylläpitäjälle	Käytettävyys loppukäytäjälle
	Pdf-export mahdollisuus valituista osioista		
Kuvat	Kuvan kiinnittäminen tekstin tietyn kohdan viereen (Huom! Saavutettavuus)		
	Kuva suurennettavissa sitä napauttamalla (thumbnail)		
Versioiden hallinta	Ohje kopioitavissa kokonaisuudessaan uuden ohjelmaversio-ohjeen pohjaksi		
	Päivitettyjen tietojen (luonnosten) julkiseminen valittuna ajankohtana		
Kaksikielisyys ohjeissa	Sisällön käännökset ulkopuolisen tahon toimesta		
	Fi- ja sv -päivitys rinnakkain (päivitetyt osiot highlightattuina tms.)		
	Sivun kielenvaihto "lennossa" (sivun "Luo asiakirja" kielenvaihto vie "Skapa dokument" -sivulle, eikä esim. aloitussivulle)		
Muut käytettävyyden asiat	Sivutemplates		
	WYSIWYG		
	Ylläpitäjän käyttöliittymä yleisesti		
	Toimittajan kirjalliset ohjeet/videot		
	Asiakaspalvelu		
	Luotettava toimittaja (saavutettavissa, vakaa)		
	Loppukäyttäjän käyttöliittymä yleisesti		

### Liite 3. Käyttöohje eri yksityiskohtaisuustasoina

**Toimielimen tietojen julkaiseminen**

Sekä organisaation omien päättävien toimielimien (päänäkömön ensimmäisellä välilehdellä), että muiden toimielimien (päänäkömön viimeisellä välilehdellä) tietoja voidaan julkaista.

**Julkaise toimielimen kokoonpano**

1. Valitse päänäkömön tilanteen mukaan **Toimielimet-** tai **Muut toimielimet** -välilehti (välilehtien nimet voivat poiketa asennuskohtaisesti).
2. Näkömön vasemman puolen puurakenteesta, valitse **toimielin**. Toimielimen perustiedot näkyvät näkömön oikean puolen yläosassa.
3. **Toimielintiedot julkaistaan** -kenttään, valitse **Kyllä**.
4. Näkömön oikean puolen alaosassa näkyy **Toimielimen julkaistavat tehtävät**. Rasti **Ei julkaista** -kenttää niiden roolien kohdalla, joita ei haluta näkyvän julkaisussa.
5. **Tallenna muutokset**.

Yksityiskohtaisuustason merkinnät DOCX-muotoisessa käyttöohjeessa.

**Julkaise toimielimen kokoonpano**

1. Valitse päänäkömön tilanteen mukaan **Toimielimet-** tai **Muut toimielimet** -välilehti
2. Näkömön vasemman puolen puurakenteesta, valitse **toimielin**.
3. **Toimielintiedot julkaistaan** -kenttään, valitse **Kyllä**.
4. Rasti **Ei julkaista** -kenttää niiden roolien kohdalla, joita ei haluta näkyvän julkaisussa.
5. **Tallenna muutokset**.

Tavoiteltu näkymä yksityiskohtaisuustasolle *Small*.

**Julkaise toimielimen kokoonpano**

1. Valitse päänäkömön tilanteen mukaan **Toimielimet-** tai **Muut toimielimet** -välilehti (välilehtien nimet voivat poiketa asennuskohtaisesti).
2. Näkömön vasemman puolen puurakenteesta, valitse **toimielin**. Toimielimen perustiedot näkyvät näkömön oikean puolen yläosassa.
3. **Toimielintiedot julkaistaan** -kenttään, valitse **Kyllä**.
4. Näkömön oikean puolen alaosassa näkyy **Toimielimen julkaistavat tehtävät**. Rasti **Ei julkaista** -kenttää niiden roolien kohdalla, joita ei haluta näkyvän julkaisussa.
5. **Tallenna muutokset**.

Tavoiteltu näkymä yksityiskohtaisuustasolle *Medium*.

**Toimielimen tietojen julkaiseminen**

Sekä organisaation omien päättävien toimielimien (päänäkömön ensimmäisellä välilehdellä), että muiden toimielimien (päänäkömön viimeisellä välilehdellä) tietoja voidaan julkaista.

**Julkaise toimielimen kokoonpano**

1. Valitse päänäkömön tilanteen mukaan **Toimielimet-** tai **Muut toimielimet** -välilehti (välilehtien nimet voivat poiketa asennuskohtaisesti).
2. Näkömön vasemman puolen puurakenteesta, valitse **toimielin**. Toimielimen perustiedot näkyvät näkömön oikean puolen yläosassa.
3. **Toimielintiedot julkaistaan** -kenttään, valitse **Kyllä**.
4. Näkömön oikean puolen alaosassa näkyy **Toimielimen julkaistavat tehtävät**. Rasti **Ei julkaista** -kenttää niiden roolien kohdalla, joita ei haluta näkyvän julkaisussa.
5. **Tallenna muutokset**.

Tavoiteltu näkymä yksityiskohtaisuustasolle *Large*.