

**VAASAN YLIOPISTO**

**TEKNILLINEN TIEDEKUNTA**

**TIETOTEKNIikka**

Ville Salomäki

**TUOTEKODIN VALIDOINTIJÄRJESTELMÄN KEHITTÄMINEN**

Diplomityö, joka on jätetty tarkastettavaksi diplomi-insinöörin tutkintoa varten

Vaasassa 7.4.2011

Työn valvoja

Jouni Lampinen

Työn ohjaaja

Petri Helo

## ALKULAUSE

Tämä diplomityö tehtiin Wapice Oy:n asiakkaan toimeksiantamaan ohjelmistokehitysprojektiin. Toimeksiantajayritys on ABB Oy. Haluaisin kiittää työn aiheesta järjestelmäarkkitehti Mari Mannilia sekä tuotteistuskoordinaattoria Jan Granlundia. Lisäksi tahdon kiittää Wapice Oy:n henkilökuntaa avunannosta työnteossa. Vaasan yliopistolta haluan kiittää Jouni Lampista neuvoista ja avusta työn rakenteen kanssa. Lopuksi kiitos ystäväilleni, jotka ovat auttaneet tekstin oikolukemisessa ja muotoilussa.

Vaasassa 24.3.2011

Ville Salomäki

| <b>SISÄLLYSLUETTELO</b>                       | <b>sivu</b> |
|---|-------------|
| ALKULAUSE                                     | 1           |
| LYHENNE- JA TERMILUETTELO                     | 4           |
| TIIVISTELMÄ                                   | 8           |
| ABSTRACT                                      | 9           |
| 1. JOHDANTO                                   | 10          |
| 1.1. Tutkielman kohde ja taustat              | 10          |
| 1.2. Työn rajaus                              | 11          |
| 1.3. Tutkielman rakenne                       | 12          |
| 2. KATSAUS TUOTEKOODEIHIN JA VALIDOINTIIN     | 13          |
| 2.1. Kiinteä tuotekoodi                       | 13          |
| 2.2. Modulaarinen tuotekoodi                  | 14          |
| 2.3. Lajimerkkiavain                          | 16          |
| 2.4. Validoinnin periaate                     | 18          |
| 2.5. Konfigurointimallit                      | 18          |
| 3. AIEMMAN JÄRJESTELMÄN KUVAUS                | 19          |
| 3.1. Tuotteen määrittely                      | 20          |
| 3.2. Tuotekonfiguraattori                     | 20          |
| 3.3. Toiminnanohjausjärjestelmä               | 21          |
| 4. UUDEN JÄRJESTELMÄN VAATIMUKSET             | 22          |
| 4.1. Tuotteen määrittelyn hallinta            | 22          |
| 4.2. Rajapinta muihin järjestelmiin           | 22          |
| 4.3. Tuotekonfiguraattorin luomat rajoitukset | 26          |
| 4.4. Tuotteiden näkyvyys                      | 27          |
| 4.5. Ylläpitoliittymä                         | 28          |
| 4.6. Ylläpitoliittymän käyttötapaukset        | 28          |

|        |  |    |
|--------|--|----|
| 5.     | TUOTETIETOJEN MÄÄRITTELY                     | 38 |
| 5.1.   | Tietomallin hierarkia                        | 38 |
| 5.2.   | Tuotekoodin merkkien sisältö                 | 39 |
| 5.3.   | Materiaalikoodin tunnistaminen tuotekoodista | 41 |
| 5.4.   | Validointisäännöt                            | 41 |
| 5.5.   | Tuotekoodin varianttien määrän tarkastelua   | 44 |
| 5.6.   | Hinnoittelukoodin muodostaminen              | 46 |
| 6.     | JÄRJESTELMÄN TOTEUTUS                        | 48 |
| 6.1.   | Yleiskatsaus järjestelmään                   | 48 |
| 6.2.   | Tietokantamalli                              | 49 |
| 6.3.   | Tuotekoodin validointi                       | 52 |
| 6.4.   | .NET moduuli                                 | 55 |
| 6.5.   | Rajapintasovellus                            | 57 |
| 6.6.   | Ylläpitoliittymä                             | 62 |
| 6.6.1. | Käytetyt tekniikat                           | 63 |
| 6.6.2. | Tuotetietojen siirtotiedosto                 | 64 |
| 6.6.3. | Käyttöliittymä                               | 65 |
| 7.     | JOHTOPÄÄTÖKSET                               | 76 |
|        | LÄHDELUETTELO                                | 78 |
|        | LIITTEET                                     | 81 |

## LYHENNE- JA TERMILUETTELO

|                    |   |
|--------------------|---|
| Aktiviteettikaavio | UML-standardin mukainen tiedon kulkukaavio  |
| C#                 | Microsoftin kehittämä oliopohjainen ohjelmointikieli, joka julkaistiin .NET-ohjelmistokomponenttikirjaston yhteydessä. Muistuttaa syntaksiltaan C++- ja Java-ohjelmointikieliä. |
| DLL                | Dynamic Link Library. Microsoft Windows – käyttöjärjestelmässä käytettävä jaettu koodikirjasto.   |
| ERP                | Enterprise Resource Planning. Toiminnanohjausjärjestelmä.   |
| GAC                | Global Assembly Cache. Välimuistitekniikka, jonka tarkoitus on estää DLL-kirjastojen konfliktitilanteita  |
| IIS                | Internet Information Service. Microsoftin palvelinympäristö.  |
| Kontrolli          | Käyttöliittymäkomponentti, jolla voi vaikuttaa ohjelman toimintaan.   |
| Käyttötapaus       | Toiminto, jonka käyttäjä voi sovelluksessa toteuttaa.   |
| Käyttötapauskaavio | Kaaviokuva, josta nähdään käyttäjät ja käyttäjien mahdolliset käyttötapaukset.  |
| Luokka             | Olion malli ja tietotyyppi, joka koostuu ryhmitellyistä attribuuteista ja metodeista.   |
| Metodi             | Luokan jäsenfunktio, jonkin tietyn operaation toteutus.   |
| Moduuli            | Tietokoneohjelman itsenäinen osa, jolla voi olla attribuutteja ja metodeja.   |
| .NET               | Microsoftin kehittämä ohjelmistokomponenttikirjasto, jota Microsoftin Visual Studio .NET -ympäristössä kehitetyt ohjelmistot käyttävät.   |
| .NET moduuli       | Microsoft .NET -ympäristössä tehty koodikirjasto, jossa on yleisimmin käytetyt metodit kuten tuotekoodin validointi ja hinnanhaku.  |
| Näkymä             | Käyttäjän näkemä käyttöliittymä, joka koostuu kontrolleista.  |
| Rakenne            | Kevyempi versio luokka-tietotyypistä.   |
| SAP                | Systeme, Anwendungen und Produkte. Ohjelmistotalo, jonka valmistamasta toiminnanohjausjärjestelmästä käytetään yleisesti nimitystä SAP.   |
| SOAP               | Simple Object Access Protocol. Protokolla XML-pohjaisten viestien vaihtamiseen tietokoneverkossa, useimmiten HTTP-protokollaa käyttäen.   |

|                      |  |
|----------------------|--|
| SQL                  | Structured Query Language. Standardoitu kyselykieli, jolla relaatiotietokantaan voi tehdä erilaisia hakuja, muutoksia ja lisäyksiä.                            |
| SQL funktio          | Relaatiotietokantaan liittyvä ohjelma, joka on toiminnoiltaan rajoittuneempi kuin SQL Stored Procedure.  |
| SQL Stored Procedure | Sovellusten käytössä oleva aliohjelma, jolla on pääsy relaatiotietokantaan.  |
| Tuotekonfiguraattori | Ohjelmisto, jolla pystyy lisäämään tai muuttamaan perustuotteen ominaisuuksia, kunnes se vastaa käyttäjän vaatimuksia.   |
| UML                  | Unified Modeling Language. Graafinen mallinnuskieli.   |
| Web Service          | Ohjelmistojärjestelmä, joka on koneelta koneelle yhteensopivaa vuorovaikutusta tietoverkon yli.  |
| XML                  | Extensible Markup Language. World Wide Web Consortiumin (W3C) laatima laitteisto- ja alustariippumaton tekstipohjainen kieli minkä tahansa tiedon kuvaamiseen. |
| XML-skeema           | Tapa kuvata XML-tiedoston rakennetta.  |
| XSD                  | XML Schema Definition. XSD-kielen voi mieltää XML-skeeman sanastoksi.  |

| <b>TAULUKKOLUETTELO</b> |   | <b>sivu</b> |
|-------------------------|---|-------------|
| <b>Taulukko 1.</b>      | Konfigurointimallin esimerkki   | 13          |
| <b>Taulukko 2.</b>      | Syötä tuotekoodi -käyttötapaus  | 24          |
| <b>Taulukko 3.</b>      | Hae materiaalikoodi -käyttötapaus                                     | 24          |
| <b>Taulukko 4.</b>      | Validoi tuotekoodi -käyttötapaus                                      | 25          |
| <b>Taulukko 5.</b>      | Hae hinnoittelukoodi -käyttötapaus                                    | 25          |
| <b>Taulukko 6.</b>      | Hae hinta -käyttötapaus   | 26          |
| <b>Taulukko 7.</b>      | Käyttäjän identifiointi -käyttötapaus                                 | 29          |
| <b>Taulukko 8.</b>      | Lähetä siirtotiedosto -käyttötapaus                                   | 30          |
| <b>Taulukko 9.</b>      | Poista materiaalikoodin tiedot -käyttötapaus                          | 30          |
| <b>Taulukko 10.</b>     | Vaihda järjestelmäkohtainen näkyvyys -käyttötapaus                    | 31          |
| <b>Taulukko 11.</b>     | Vaihda maakohtainen näkyvyys -käyttötapaus                            | 31          |
| <b>Taulukko 12.</b>     | Lisää käyttäjä -käyttötapaus  | 32          |
| <b>Taulukko 13.</b>     | Poista käyttäjä -käyttötapaus   | 33          |
| <b>Taulukko 14.</b>     | Vaihda käyttäjän rooli -käyttötapaus                                  | 33          |
| <b>Taulukko 15.</b>     | Hae tuotteen tiedot -käyttötapaus                                     | 34          |
| <b>Taulukko 16.</b>     | Hae hinta -käyttötapaus   | 35          |
| <b>Taulukko 17.</b>     | Massatestaa tuotekoodit -käyttötapaus                                 | 35          |
| <b>Taulukko 18.</b>     | Valitse hintalistojen nimet, jotka ovat exportoitavissa -käyttötapaus | 36          |
| <b>Taulukko 19.</b>     | Tulosta hintalista ruudulle -käyttötapaus                             | 37          |
| <b>Taulukko 20.</b>     | Exportoi valitut hintalistat CSV-tiedostoon -käyttötapaus             | 37          |
| <b>Taulukko 21.</b>     | Tuotekoodin merkkien sisältö  | 40          |
| <b>Taulukko 22.</b>     | Materiaalikoodin tunnistaminen tuotekoodista                          | 41          |
| <b>Taulukko 23.</b>     | Validointisäännöt   | 42          |

|                     |   |    |
|---------------------|---|----|
| <b>Taulukko 24.</b> | Ensimmäinen sääntö aukikerrottuna                   | 43 |
| <b>Taulukko 25.</b> | Varianttien määrä ilman validointisääntöjä          | 44 |
| <b>Taulukko 26.</b> | Validointisäännöt yhdistettynä yhdeksi säännöksi    | 45 |
| <b>Taulukko 27.</b> | Varianttien määrä huomioon ottaen validointisäännöt | 45 |
| <b>Taulukko 28.</b> | Hinnoittelukoodin muodostaminen                     | 47 |
| <b>Taulukko 29.</b> | Tietokantataulut                                    | 51 |
| <b>Taulukko 30.</b> | Validointifunktion parametrit                       | 52 |
| <b>Taulukko 31.</b> | Validointisäännöt auki kerrottuna                   | 53 |
| <b>Taulukko 32.</b> | Sääntöjen vastaavuudet suhteessa tuotekoodiin       | 54 |
| <b>Taulukko 33.</b> | Sääntöjen määrän laskeminen                         | 55 |
| <b>Taulukko 34.</b> | Rajapintasovelluksen palauttavat virheilmoitukset   | 62 |



---

**VAASAN YLIOPISTO****Teknillinen tiedekunta**

|                                      |   |                      |
|--------------------------------------|---|----------------------|
| <b>Tekijä:</b>                       | Ville Salomäki                                  |                      |
| <b>Diplomityön nimi:</b>             | Tuotekoodin validointijärjestelmän kehittäminen |                      |
| <b>Valvojan nimi:</b>                | Jouni Lampinen                                  |                      |
| <b>Ohjaajan nimi:</b>                | Petri Helo                                      |                      |
| <b>Tutkinto:</b>                     | Diplomi-insinööri                               |                      |
| <b>Koulutusohjelma:</b>              | Tietotekniikan koulutusohjelma                  |                      |
| <b>Suunta:</b>                       | Ohjelmistotekniikka                             |                      |
| <b>Opintojen aloitusvuosi:</b>       | 2007  |                      |
| <b>Diplomityön valmistumisvuosi:</b> | 2011  | <b>Sivumäärä: 84</b> |

---

**TIIVISTELMÄ**

Modulaaristen tuotteiden dynaamisuus asettaa vaatimuksia tuotekoodien määrittelylle ja erityisesti validoinnille, eli oikeellisuuden tarkistamiselle. Tuotteiden modulaarisuuden vuoksi samasta tuotteesta voi eri komponenttivalinnoilla muodostaa lukuisia eri variantteja, eli kombinaatioita. Tällöin ei ole järkevää käsitellä näitä tuotekoodeja kiinteinä listauksina, sillä se tekee niiden hallinnasta lähdes mahdotonta.

Tuotetietojen tulee näkyä samalla tavalla useampaan eri järjestelmään ja näiden tietojen hallintaa varten tulee olla helppokäyttöinen ylläpitoliittymä. Tuotetiedot, joita järjestelmän tulee palauttaa toiminnanohjausjärjestelmään, ovat validointitulos, hinnoittelukoodi ja hinta. Lisäksi tulee tarjota virheilmoitus, joka kuvaa tarkemmin, mikä syötearvoissa oli vikana.

Tutkielma on enemmän käytäntö- kuin teoriapainotteinen. Teoriaan liittyen tehtiin havainto, että tuotekoodin varianttien määrä pystytään laskemaan tuloperiaatteen avulla. Lisäksi käytiin läpi tuotekoodin määrittelyyn liittyvää kirjallisuutta.

Lopputuloksena saatiin järjestelmä, joka ei ainoastaan vastannut asiakkaan vaatimuksiin, vaan myös tarjosi helposti laajennettavan alustan muita toimintoja varten. Järjestelmä on aktiivisessa käytössä ja sitä kehitetään edelleen. Tuotetiedot tarjotaan Web Service -rajapinnan yli toiminnanohjausjärjestelmään. Tuotekonfiguraattori ei tue Web Service -rajapintaa, joten tuotekonfiguraattori hakee tiedot suoraan tietokannasta.

---

**AVAINSANAT:** tuotekoodi, validointi, ERP, .NET, C#, tuotekonfiguraattori

---

**UNIVERSITY OF VAASA****Faculty of technology**

|   |   |                  |
|---|---|------------------|
| <b>Author:</b>                          | Ville Salomäki                            |                  |
| <b>Topic of the Thesis:</b>             | Developing product code validation system |                  |
| <b>Supervisor:</b>                      | Jouni Lampinen                            |                  |
| <b>Instructor:</b>                      | Petri Helo                                |                  |
| <b>Degree:</b>                          | Master of Science in Technology           |                  |
| <b>Degree Programme:</b>                | Degree Programme in Computer Science      |                  |
| <b>Major of Subject:</b>                | Software Engineering                      |                  |
| <b>Year of Entering the University:</b> | 2007                                      |                  |
| <b>Year of Completing the Thesis:</b>   | 2011                                      | <b>Pages: 84</b> |

---

**ABSTRACT**

The dynamicity of modular products sets requirements for product code definition and especially for validation. Because of products' modularity there can be numerous variations within the same product with different component choices. Because of this it is not reasonable to handle product codes as static lists, because it makes their handling almost impossible.

Product information should be visible in the same way to several different systems, and for the maintenance of information there should be an easy-to-use interface. Product information that the system should be able to return to an enterprise resource planning system is the validation result, pricing code and price. In addition, an error message should be provided describing more specifically what was wrong in the input parameters.

The thesis is more practical than theory based. With respect to theory, it stated that the number of product code variants can be calculated with multiplication principle. In addition it reviewed literature about defining the product.

The result was a system which didn't just meet the requirements of client, but also offered basis for ease of extension of other functionalities. The system is in active use and the developing process has not stopped. Information is provided through Web Service -interface for the enterprise resource planning system. Product configurator does not support this kind of interface, so the product configurator loads data directly from database.

---

**KEYWORDS:** product code, validation, ERP, .NET, C#, product configurator

## 1. JOHDANTO

Tuotekoodin validointi, eli oikeellisuuden tarkistaminen, on tärkeä osa-alue toiminnanohjausjärjestelmässä. Tuotekoodin perusteella määräytyy, onko tuotetta ylipäättänsä olemassa ja mitä komponentteja tuote sisältää. Lisäksi tuotekoodista pystytään päättämään monia asioita, kuten mikä tuotteen hinta on ja missä maassa sitä saa myydä.

Oman haasteensa validointiin tekee se, että yrityksellä on useita eri järjestelmiä joihin validointitiedot täytyy näkyä juuri samalla tavalla. Lisäksi tuotekoodit ovat modulaarisia, joka tarkoittaa sitä että eri komponenttien yhdistelmistä joista tuotekoodin voi muodostaa, voi muodostua massiivinen määrä eri tuotekoodeja. Tällöin tuotekoodin validointia tulee miettiä erityisen tarkkaan, sillä ei ole järkevää esimerkiksi luetella yhteen tietueeseen kaikkia valideja, eli oikeellisia, tuotekoodeja, koska tietueesta tulisi massiivinen ja validointiprosessista hidas (Jørgensen 2003: 3). Tuotesuojan vuoksi työssä ei paljasteta yksityiskohtaisia tuotetietoja, vaan tuotteena käytetään kuvitteellista tietokoneen keskusyksikköä.

### 1.1. Tutkielman kohde ja taustat

Pääkohde tutkielmassa on tuotekoodin validointi ja validointitietojen tarjoaminen eri järjestelmiin. Lisäksi tuotetietoja on pysyttävä hallinnoimaan omalla järjestelmällään. Järjestelmät, joihin validointitietoja tulee pystyä tarjoamaan, ovat toiminnanohjausjärjestelmä ja tuotekonfiguraattori. Tuotekonfiguraattori asettaa omat rajoituksensa sille, miten validointitietoja voidaan tarjota. Tämä rajaa teknisten ratkaisuvaihtoehtojen määrää.

Aikaisemmin tuotekoodien validointi on perustunut toiminnanohjausjärjestelmässä kiinteään listaukseen valideista tuotekoodeista, jotka ovat sijainneet järjestelmän paikallisessa tietokannassa. Modulaarisia tuotekoodeja ei pystytty validoimaan. Tuotekonfiguraattorissa on ollut kiinteä listaus tuotekoodeista ja rajoittunut tuki modulaarisille tuotekoodeille, mutta tuotekonfiguraattorin teknisten rajoitusten takia modulaaristen tuotekoodien validointitietojen vieminen muihin järjestelmiin on ollut hankalaa ja epäluotettavaa. Nämä validointitiedot on saatava vietyä molempiin järjestelmiin luotettavasti ja tehokkaasti.

Yksi järjestelmän tärkeä osa-alue on tuotteiden versioiden hallinta. Jokaisella tuotteen versiolla on määritelty omat komponentit ja omat validointisäännöt. Tällaisten tuotteiden versioiden näkyvyyksiä tulee pystyä hallinnoimaan siten, että esimerkiksi tuotteen versiota 1.0 ei saa enää myydä mutta uudempaa versiota 1.1 tulee pystyä myymään.

## 1.2. Työn rajaus

Tutkielman päätavoite on ratkaista ongelma, miten modulaaristen tuotekoodien validointi toteutetaan siten, että suorituskyky on riittävän hyvä ja tietokantataulujen koot eivät kasva liian suuriksi. Lisäksi tavoitteena on toteuttaa helppokäyttöinen ylläpitoliittymä tuotetietoja varten. Järjestelmä tulee rakentaa sellaiseksi, että sitä on helppo laajentaa myöhemmissä vaiheissa.

Tutkielmassa käsitellään itse tuotekoodin määrittelyn perusteita, validointia, rajapintaa jolla tiedot viedään eri järjestelmiin ja tuotekoodien ylläpitoliittymää. Vaikka asiakas on tehnyt tuotekoodin määrittelymallin, esitellään sen perusteet, koska se on olennaista järjestelmän toiminnan ymmärtämisen kannalta. Työhön eivät kuulu itse tuotekonfiguraattori, toiminnanohjausjärjestelmä ja hintojen tarkempi laskenta. Tuotekonfiguraattorista ja toiminnanohjausjärjestelmästä kerrotaan vain sellaiset seikat, jotka oleellisia järjestelmän toteutuksessa tehtyjen ratkaisujen ymmärtämisen kannalta.

### 1.3. Tutkielman rakenne

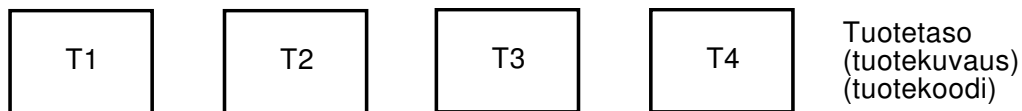
Tutkielma alkaa aiheeseen alustuksella ja työn rajauksella. Seuraava pääluku on teoria-painotteinen, käsitellen tuotekoodin määrittelyä ja validoinnin periaatetta. Näitä on käsitelty tarkemmin myös myöhemmissä luvuissa. Kolmannessa pääluvussa kerrotaan aiemmasta järjestelmästä ja kuvataan siihen liittyneitä ongelmia. Neljäs pääluku koostuu uuden järjestelmän vaatimusmäärittelystä. Viides pääluku kertoo tuotetietojen määrittelystä tarkemmin, joka on oleellista tutkielman myöhempien vaiheiden ymmärtämisen kannalta. Kuudes pääluku käy kattavasti läpi uuden järjestelmän toteutuksen ja seitsemännessä pääluvussa esitetään johtopäätökset.

## 2. KATSAUS TUOTEKOODEIHIN JA VALIDOINTIIN

Tuotekoodi on merkkijono, josta pystyy päättämään tuotteeseen liittyviä tietoja kuten komponentit, version ja standardin. Tämän tutkielman yhteydessä käsitellään kahdenlaisia tuotekoodeja; kiinteitä ja modulaarisia. Kiinteistä tuotekoodeista ei välttämättä pysty suoraan päättämään tuotteen komponentteja, mutta modulaarisesta tuotekoodista komponentit pystyy selvittämään (Tiihonen, Lehtonen, Soininen, Pulkkinen, Sulonen & Riitahuhta 1998: 7). Tuotekoodia ei tule sekoittaa sarjanumeroon. Tuotekoodi on sama kaikille samantyyppisille tuotteille, mutta sarjanumero on yksilöllinen jokaiselle yksittäiselle kappaleelle.

### 2.1. Kiinteä tuotekoodi

Kiinteitä tuotekoodeja käsitellään kokonaisuuksina. Tämä tarkoittaa, että tietomallissa tuotekuvaus on tuotekoodin kanssa samalla tasolla. Kuvauksessa on sanottu, mitä tuotekoodin mukainen kokoonpano sisältää. Kiinteän tuotekoodin malli on esitetty kuvassa 1.



**Kuva 1.** Kiinteän tuotekoodin malli

Kiinteän tuotekoodin heikkous on siinä, että tuotekoodeja ja tuotekuvauksia täytyy olla tietueessa yhtä monta kuin on tuotekoodin kombinaatioita eli variantteja. Esimerkkinä konfigurointimalli, jossa on neljä ”kyllä / ei” –tyyppistä parametria ja lisäksi viisi parametria neljällä eri vaihtoehdolla. Tämä on esitetty taulukossa 1 (Tiihonen ym. 1998: 7).

**Taulukko 1.** Konfigurointimallin esimerkki

| Parametri               | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 |
|-------------------------|----|----|----|----|----|----|----|----|----|
| Vaihtoehtojen lukumäärä | 2  | 2  | 2  | 2  | 4  | 4  | 4  | 4  | 4  |

Varianttien kokonaismäärä saadaan, kun kunkin parametrin vaihtoehtojen lukumäärä kerrotaan keskenään. Varianttien määrä on laskettu kaavalla 1.

$$\text{Varianttien määrä} = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 4 \cdot 4 \cdot 4 \cdot 4 \cdot 4 = 2^4 \cdot 4^5 = 16384 \quad (1)$$

Tästä huomataan, että varianttien määrä saadaan laskettua suoraan tuloperiaatteen mukaisesti, joka on esitetty kaavassa 2 (Viertola 2011).

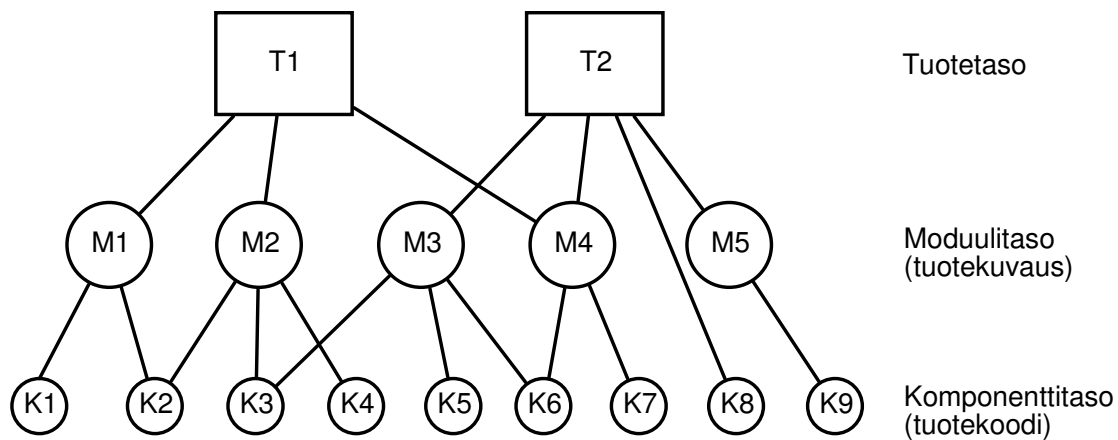
$$\text{Varianttien määrä} = n_1 \cdot n_2 \cdot n_3 \cdot \dots \cdot n_{k-1} \cdot n_k \quad (2)$$

Jos viiden valinnan sijaan konfigurointimallissa olisikin kuusi eri valintaa, kasvaisi varianttien määrä jo 65536 kappaleeseen. Tästä voidaan päätellä, että monimutkaisten konfigurointimallien kuvaaminen kiinteillä tuotekodeilla ei ole järkevää, sillä tuotekoodien hallinta olisi lähes mahdotonta. Tuotekoodin määrittelystä on kerrottu lisää luvussa viisi.

## 2.2. Modulaarinen tuotekoodi

Modulaarisen tuotekoodin yhteydessä on tietomallissa listaus moduuleista ja komponenteista, mutta ei tuotekodeista. Komponentti on alin taso, mihin tuotekoodi hajoaa ja tyypillisesti yksi merkki tuotekoodissa vastaa yhtä komponenttia. Moduulilla tarkoitetaan isompaa komponenttipakettia, ja tätä tasoa käytetään usein apuna tuotanto- ja hinnoittelutasolla. Tuotteen kuvaus ei ole tuotetasolla vaan moduulitasolla. Jokaisella moduulilla on oma kuvauksensa, joten tuotteen kuvaukseksi muodostuu näiden moduulien kuvausten yhdistelmä (Jørgensen 2003: 7).

Tuotekoodi muodostuu komponenttien osoittamista merkeistä. Tyypillisesti kutakin komponenttia varten on varattu yksi merkki. Esimerkiksi valitussa tuotteessa ensimmäinen komponentti voi saada arvokseen merkin *H*, toinen komponentti arvon *J* ja niin edelleen. Kun nämä valittuja komponentteja vastaavat merkit yhdistetään, muodostuu tuotekoodi. Komponentin ei välttämättä tarvitse sisältyä moduuliin, jos komponentin ilmoittama tieto ei ole hinnoittelun tai tuotannon kannalta oleellinen. Modulaarisen tuotekoodin malli on esitetty kuvassa 2.



**Kuva 2.** Modulaarisen tuotekoodin malli

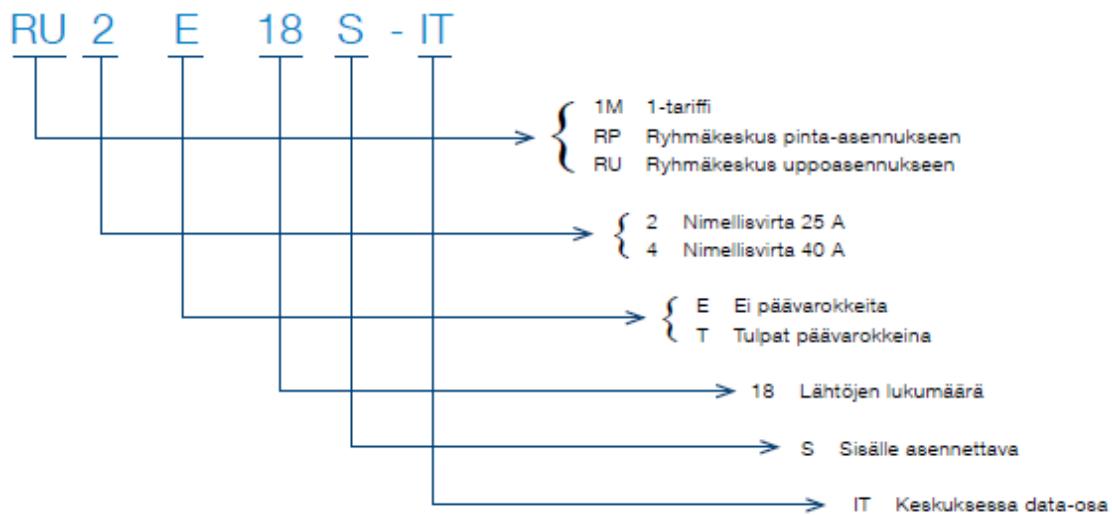
Modulaariset tuotekoodit ovat välttämättömiä, mikäli tuotteessa olevien parametrien ja näiden arvojen määrä kasvaa suureksi. Tuotekoodissa on tunnistava osa, jonka avulla tiedetään mihin tuotteeseen tuotekoodi kuuluu. Voidaan esimerkiksi määrittellä, että tuotekoodin kaksi ensimmäistä merkkiä on varattu tuotteen tunnistusta varten (Mukherjee, Ryan, Wason 1994: 3).

Tuotteista voi olla eri versioita. Tällainen tarve syntyy, kun esimerkiksi asiakaspalautteen vuoksi huomataan, että tuotteeseen tarvitaan yksi ylimääräinen komponenttivalikoima. Tällöin tulisi versioida tuotteen konfigurointimalli. Tämä saavutetaan siten, että konfigurointimallia ei käsitellä tuotetasolla vaan versiotasolla. Tällöin tuotekoodiin on lisättävä tunnistavan osan lisäksi myös versioiva osa. Versioiva osa voi olla esimerkiksi tuotekoodin viimeinen merkki.



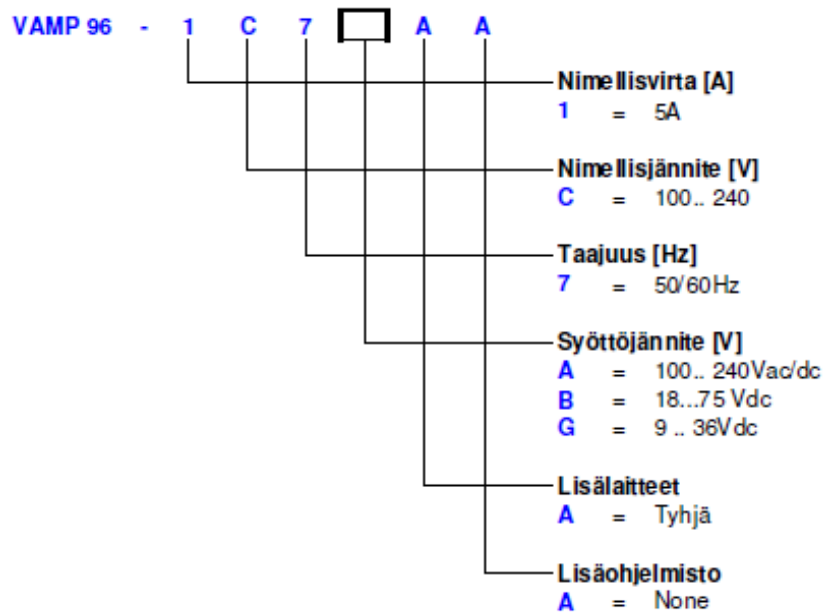
### 2.3. Lajimerkkiavain

Lajimerkillä tarkoitetaan sellaista kaaviokuvaa tai taulukkoa, josta näkee mitä tuotekoodin merkit tarkoittavat. Lajimerkkiavaimet, kuten myös tuotekoodit, ovat usein huonosti standardisoituja. Merkintätavat vaihtelevat suuresti valmistajien ja jopa saman valmistajan tuoteperheiden kesken. Kuvassa 3 on esitetty ABB Oy:n pienjännitekaapin lajimerkkiavain (ABB Oy 2011). Komponenttivalintoja on vähän, jonka vuoksi tuotetiedot voitaisiin kuvata tietomallissa kiinteinä tuotekoodeina. Jokainen merkki vastaa komponenttivalintaa, eikä tuotekoodissa ole tunnistavaa tai versioivaa osaa.



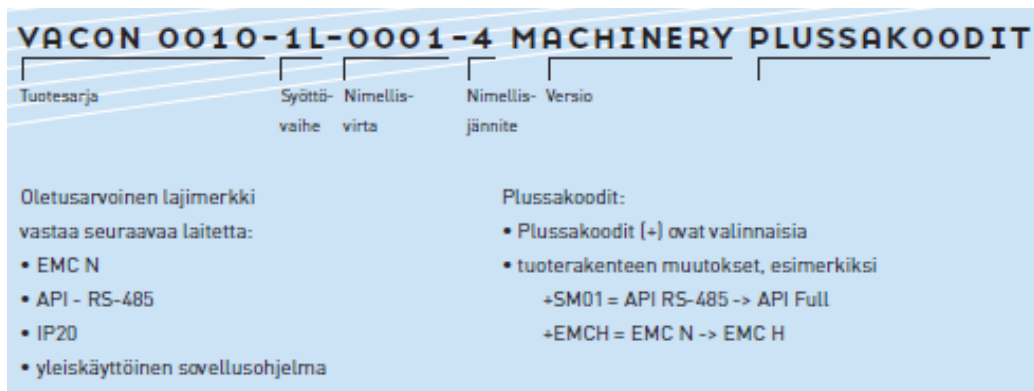
**Kuva 3.** ABB Oy:n pienjännitekaapin lajimerkkiavain

Kuvassa 4 on esitetty Vamp Oy:n lajimerkkiavain suojareleelle (Vamp Oy 2011). Edelliseen esimerkkiin verrattuna siinä on lisäksi tuotekoodin alussa osa *VAMP96*-. Tällä määritellään tuoteperhe, ja se on samalla myös tunnistava osa.



**Kuva 4.** VAMP Oy:n suoja releen lajimerkkiavain

Vacon Oy:n taajuusmuuttajan lajimerkkiavain on esitetty kuvassa 5 (Vacon Oy 2008). Siinä on tunnistava osa ja versioiva osa. Lisäksi tuotekoodin lopussa on ”plussakoodit”. Näillä koodeilla ilmaistaan, miten tuotteen konfigurointi eroaa oletusarvoisesta tuotekoodista.



**Kuva 5.** Vacon Oy:n taajuusmuuttajan lajimerkkiavain

## 2.4. Validoinnin periaate

Validointi on prosessi, jossa tarkistetaan että tuote, palvelu tai järjestelmä vastaa sille asetettuja määrittelyitä ja täyttää tarkoituksensa (Boehm 1984). Jos konfigurointimalliin liittyy parametrien välisiä sääntöjä, eli validointisääntöjä, vähenee samalla varianttien määrä. Taulukon 1 esimerkissä voisi olla sellainen validointisääntö, että jos ensimmäisen parametrin arvoksi valitaan *kyllä*, täytyy myös toisen parametrin arvo olla *kyllä*.

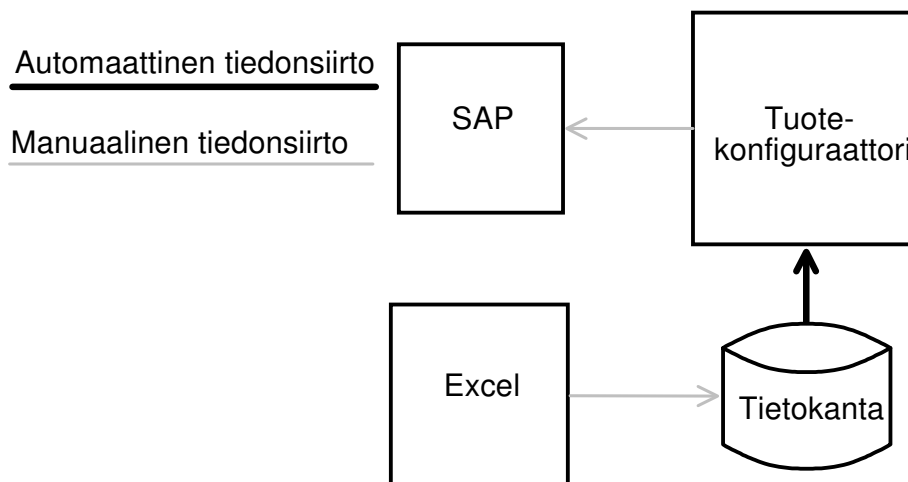
Kiinteillä tuotekoodilla ei tarvita validointisääntöjä. Silloin yksinkertaisesti riittää, että määritellään tietueeseen vain ne tuotekoodit, jotka ovat valideja. Modulaaristen tuotekoodien yhteydessä tämä ei ole järkevää varianttien suuren määrän vuoksi. Tällöin tietueeseen ei lisätä valideja tuotekoodia, vaan validointisäännöt. Tuotekoodin validointia on käsitelty tarkemmin luvuissa 5.4. ja 6.3.

## 2.5. Konfigurointimallit

Tuotekonfiguraattori on työkalu, jolla käyttäjä konfiguroi tuotteen joka vastaa käyttäjän vaatimuksia. Tuotteen konfigurointi tarkoittaa sitä, että käyttäjä vastaa käyttöliittymässä esitettyihin kysymyksiin, jotka voivat olla esimerkiksi suoria kysymyksiä siitä, mitä komponentteja tuotteessa tulee olla, tai vaatimuskohtaisia kysymyksiä, esimerkiksi millä käyttöjännitteellä tuotteen tulee toimia. Tuotekoodin validointi liittyy olennaisesti konfigurointiin. Tuotekonfiguraattoreilla voi olla useita eri tapoja käsitellä tuotteen konfigurointiin liittyviä sääntöjä. Sääntöpohjainen tapa tarkoittaa sitä, että käyttäjän tekemät valinnat laukaisevat sääntöjä, jotka pakottavat muiden parametrien arvoja säännön mukaiseksi (Bakker, Dikker, Tempelman, Wogmim 1993). Constraint Satisfaction Problem (CSP) –tyyppisessä konfigurointimallissa säännöt koostuvat rajoitteista. Ongelmana on löytää parametreille sellainen sijoitus, että kaikki rajoitesäännöt tyydyttyvät (Bensana, Mulyanto, Verfaillie, 2000). Oliopohjainen konfigurointimalli perustuu siihen, että konfiguroinnin tiedot jäsennetään olioiden, attribuuttien, luokkien, viittausten ja periytymisen avulla (Bensana ym. 2000).

### 3. AIEMMAN JÄRJESTELMÄN KUVAUS

Järjestelmässä on useampi työvaihe perustunut manuaaliseen tiedonsiirtoon. Tämä tarkoittaa sitä, että käyttäjä on kopioinut tietoa käsin järjestelmien välillä tai pahimmassa tapauksessa kirjoittanut tietoa ulkomuistiin luottaen. Tästä aiheutuvat suurimmat haitat ovat virhealttius ja hitaus. Tuotteen määrittely on tehty Excel-  
taulukkolaskentaohjelmassa. Nämä tiedot on jälkiprosessoinnin jälkeen siirretty tietokantaan. Jälkiprosessointi on työvaiheena ollut erityisen virhealtis. Kun tiedot on viety tietokantaan, on tuotekonfiguraattoriin toteutettu näiden tietojen käsittely, jolloin tuote ilmestyy tuotekonfiguraattoriin. Tämän jälkeen käyttäjä on voinut tuotekonfiguraattorilla määrittää haluamansa tuotteen, jonka jälkeen tuotekoodi on siirretty käsin toiminnanohjausjärjestelmään. Samalla on pitänyt kopioida tuotantoa avustava osaluettelo, jota käytetään myös hinnoitteluun. Toiselta nimeltä tätä osaluetteloä sanotaan hinnoittelukoodiksi. Kuvassa 6 on tätä prosessia esittävä kommunikaatiokaavio.



**Kuva 6.** Aiemman järjestelmän kommunikaatiokaavio

### 3.1. Tuotteen määrittely

Määrittely tehdään Excel-taulukkolaskentaohjelmassa. Excel-tiedostot on jaettu Lotus Notes -järjestelmän avulla, jolloin kaikki yrityksen työntekijät, joilla on oikeus nähdä tiedostot, pääsevät niihin käsiksi. Tiedostot on jaettu välilehtiin. Esimerkiksi ensimmäisellä välilehdellä on tiedoston muutoshistoria ja toisella välilehdellä on määritelty mitkä eri arvot ovat mahdollisia tuotekoodin merkkien eri indekseille. Tuotepäällikkö kertoo millainen tuotteen tulee olla, mutta itse määrittelytiedoston tekee tuotteistuskoordinaattori.

### 3.2. Tuotekonfiguraattori

Tuotteen konfigurointi perustuu Selectica-tuotekonfiguraattoriin, joka on kolmannen osapuolen tuote. Se koostuu kolmesta eri pääkomponentista: sääntömoottorista, säännöstön hallinnan käyttöliittymästä ja käyttöliittymästä. Selectica sallii monenlaisia eri sääntöjä, kuten yksinkertaisia ehtosääntöjä tai massakonfigurointiin liittyviä taulutyyppejä sääntöjä. Säännöstön hallinnan käyttöliittymä on Java-ohjelmointikielellä toteutettu ohjelma. Itse konfiguroinnin käyttöliittymä on web-sovellus, jolle on Selectica-tuotekonfiguraattorissa oma syntaksikielensä. Tällä syntaksikielellä pystyy toteuttamaan yleisimmät tuotekonfiguraattorin asettamat vaatimukset. Sen lisäksi käyttöliittymässä voidaan käyttää Java-ohjelmointikielellä toteutettuja ohjelmistokomponentteja, joiden avulla voidaan käyttöliittymässä toteuttaa sellaisia ominaisuuksia, mihin Selectica-tuotekonfiguraattorin oma syntaksikieli ei pysty (Selectica, Inc. 2005: 210–222, 314).

Jotta tuote tulee näkyville tuotekonfiguraattoriin, on tuotetiedot ensin pitänyt lisätä tietokantaan. Tuotekonfiguraattori oli alun perin suunniteltu tukemaan vain kiinteitä tuotekoodeja. Modulaaristen tuotekoodien tullessa kuvioihin, oli kiire saada tuote konfiguroitavaksi eikä tuotekonfiguraattorin arkkitehtuuriin ehditty panostamaan. Tästä syystä modulaaristen tuotekoodien tuotteet lisättiin tuotekonfiguraattoriin tavalla, joka ei tukenut myöhempiä vaatimuksia.

### 3.3. Toiminnanohjausjärjestelmä

Käytössä oleva Enterprise Resource Planning (ERP) -järjestelmä, eli toiminnanohjausjärjestelmä, on Systeme, Anwendungen und Produkte (SAP) -ohjelmistotalon valmistama sovellus. Yleisesti järjestelmää kutsutaan SAP-järjestelmäksi. Toiminnanohjausjärjestelmä on käytössä koko tilaus-toimitusketjussa kaikissa vaiheissa aina tilauksesta tuotteen ja laskun lähettämiseen. Myöhemmin esiin tuleva materiaalikoodi on toiminnanohjausjärjestelmän vaatima koodi, mutta se toimii samalla myös tuotteiden eri versioiden hallinnointiin sopivana koodina (SAP AG 2001: 7).

## 4. UUDEN JÄRJESTELMÄN VAATIMUKSET

Uudessa järjestelmässä tiedon tulee välittyä järjestelmien välillä niin, että käsin tehtävän työn osuus on mahdollisimman pieni. Lisäksi tuotteiden hallinnan tulee olla standardoituja ja skaalautuvaa. Tämä tarkoittaa sitä, että uutta tuotetta lisättäessä ei sitä varten tarvitse luoda uutta tietuetta, vaan riittää että tiedot lisätään jo olemassa olevaan skaalautuvaan tietueeseen.

### 4.1. Tuotteen määrittelyn hallinta

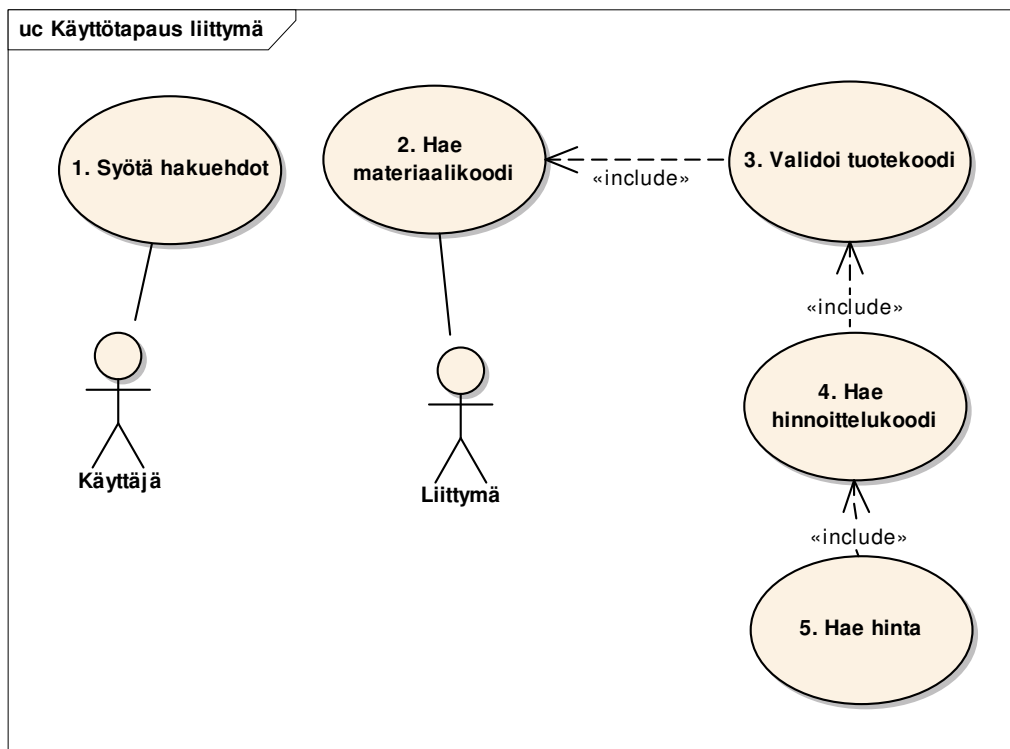
Tuotteen määrittely halutaan pitää Excel-taulukkolaskentaohjelmassa samaan tapaan kuin aiemmassa järjestelmässä. Tätä puoltavat helppo käytettävyys, muokattavuus ja käyttäjien tottumus ohjelmaan. Määrittely on kuitenkin tehtävä standardoidulla tavalla. Tämä tarkoittaa käytännössä sitä, että kaikkien tuotteiden määrittelyssä käytetään samaa taulukkopohjaa.

### 4.2. Rajapinta muihin järjestelmiin

Rajapinta toiminnanohjausjärjestelmään voidaan toteuttaa normaalilla Web Service -tekniikalla. Se on koneelta koneelle yhteensopivaa vuorovaikutusta tietoverkon yli (Haas, Brown 2004). Samaa rajapintasovellusta voidaan tarvittaessa käyttää tulevaisuudessa myös muiden järjestelmien välillä.

Selectica-tuotekonfiguraattoriin tuotteet tulee näkyä samalla tavalla kuin muihinkin järjestelmiin. Koska Selecticaan tarvitaan tuotteista tarkemmat tiedot kuin toiminnanohjausjärjestelmään, on järkevää että Selectica tuotekonfiguraattorilla on suora yhteys tietokantaan. Tämä lisää nopeutta ja Selectica ei varsinaisesti tue rajapintatekniikoita kuten Web Service –tekniikkaa.

Kuvassa 7 on esitetty rajapintaan liittyvät käyttötapaukset. Itse käyttäjän vuorovaikutukseen liittyy vain yksi käyttötapaus ja muut käyttötapaukset ovat liittymään kuuluvia. Käyttötapausten tarkemmat kuvaukset on esitetty taulukoissa käyttötapauskaavion jälkeen. Tämän tutkielman käyttötapauskaavioissa on virhetilannekenttään listattu virhetilanteet, jotka ovat voineet johtua asioista, joiden virheenkäsittelyyn voi vaikuttaa sovelluskoodissa. Virhetilanteisiin ei siis ole lisätty harvinaisempia tilanteita, kuten tietokantapalvelimen kaatuminen, sen korruptoituminen tai sähkökatkos. Käyttötapaukset 2–5 liittyvät toisiinsa määreellä *include*, eli sisältyminen. Se tarkoittaa sitä, että jos jokin toinen käyttötapaus sisältää tämän käyttötapauksen, tulee myös tämä sisällytettävä käyttötapaus suorittaa (Haikala, Märijärvi 2004: 157–159).



**Kuva 7.** Rajapinnan käyttötapauskaavio



## Käyttötapaus 1 – Syötä hakuehdot

**Taulukko 2.** Syötä tuotekoodi -käyttötapaus

|                |   |
|----------------|---|
| Alkuehdot      | Käyttäjä tietää, minkä tuotekoodin, hintalistan nimen ja valuutan syöttää.  |
| Syötteet       | Tuotekoodi, hintalistan nimi ja valuutta.   |
| Kuvaus         | Käyttäjä syöttää toiminnanohjausjärjestelmässä omiin tekstikenttiinsä tuotekoodin, hintalistan nimen ja valuutan. Käyttäjä painaa lopuksi tuotekoodi-kenttä aktiivisena enter-painiketta, jolloin tietojen haku käynnistyy. |
| Virhetilanteet | -   |
| Tulokset       | Materiaalikoodi, hinnoittelukoodi ja hinta, tai virheilmoitus.  |

## Käyttötapaus 2 – Hae materiaalikoodi

**Taulukko 3.** Hae materiaalikoodi -käyttötapaus

|                |  |
|----------------|--|
| Alkuehdot      | Tuotekoodi on syötetty.                              |
| Syötteet       | Tuotekoodi.  |
| Kuvaus         | Materiaalikoodi selvitetään tuotekoodin perusteella. |
| Virhetilanteet | Materiaalikoodia ei saatu selvitettyä.               |
| Tulokset       | Materiaalikoodi tai virheilmoitus.                   |

### Käyttötapaus 3 – Validoi tuotekoodi

**Taulukko 4.** Validoi tuotekoodi -käyttötapaus

|                |  |
|----------------|--|
| Alkuehdot      | Materiaalikoodi on selvitetty. Toiminnanohjausjärjestelmä palauttaa maakoodin.   |
| Syötteet       | Materiaalikoodi, tuotekoodi ja maakoodi. Maakoodi tulee kiinteänä syötteenä toiminnanohjausjärjestelmästä eikä käyttäjä pysty vaikuttamaan siihen.       |
| Kuvaus         | Tuotekoodi validoidaan ja tarkistetaan että kyseisellä maakoodilla on oikeus tilata sitä.  |
| Virhetilanteet | Tuotekoodi ei ole validi tai valitulla maakoodilla ei ole oikeutta tilata materiaalikoodia. Materiaalikoodin mukaiset tiedot on poistunut tietokannasta. |
| Tulokset       | Numero yksi jos tuotekoodi on validi. Muussa tapauksessa virheilmoitus.  |

### Käyttötapaus 4 – Hae hinnoittelukoodi

**Taulukko 5.** Hae hinnoittelukoodi -käyttötapaus

|                |   |
|----------------|---|
| Alkuehdot      | Tuotekoodi on validi.   |
| Syötteet       | Materiaalikoodi ja tilauskoodi.                                 |
| Kuvaus         | Haetaan tuotekoodia vastaava hinnoittelukoodi.                  |
| Virhetilanteet | Materiaalikoodin mukaiset tiedot ovat poistuneet tietokannasta. |
| Tulokset       | Hinnoittelukoodi tai virheilmoitus.                             |

## Käyttötapaukset 5 – Hae hinta

**Taulukko 6.** Hae hinta -käyttötapaus

|                |  |
|----------------|--|
| Alkuehdot      | Hinnoittelukoodi on selvitetty. Hintalistan nimi ja valuutta on syötetty toiminnanohjausjärjestelmään. Toiminnanohjausjärjestelmä palauttaa maakoodin.   |
| Syötteet       | Hinnoittelukoodi, hintalistan nimi, valuutta ja maakoodi.  |
| Kuvaus         | Haetaan hinnoittelukoodia ja muita syöttöparametreja vastaava hinta. Lisäksi tarkistetaan, onko valitulla maakoodilla oikeus valittuun hintalistan nimeen. Jos jokin hinnoittelukoodin osa on tietokannassa arvoltaan <i>NULL</i> , ilmoitetaan tästä käyttäjälle virheilmoituksen muodossa eikä hintaa palauteta. |
| Virhetilanteet | Hintaa ei saatu selvitettyä.   |
| Tulokset       | Hinnoittelukoodia vastaava hinta tai virheilmoitus.  |

### 4.3. Tuotekonfiguraattorin luomat rajoitukset

Toiminnanohjausjärjestelmään riittää tieto, onko tuotekoodi validi, eli oikeellinen, vai ei. Tällöin tuotetta ei konfiguroida rajapinnan kautta. Tuotekonfiguraattorissa sen sijaan muodostetaan tuotekoodi konfiguroimalla, joka tarkoittaa että tuotteeseen valitaan eri komponentteja ja vastataan kysymyksiin jotka liittyvät käyttäjän vaatimuksiin. Valinnat ovat vuorovaikutuksessa keskenään, eli aikaisempi valinta rajaa pois muiden valintojen arvoja. Näiden valintojen perusteella tuotekoodi muodostuu pala palalta. Toimintaperiaatteen erilaisuudesta huolimatta tuotekoodin validointisääntöjen tulee olla vain yhdessä paikassa, josta ne näkyvät samalla tavalla kaikkiin järjestelmiin.

Suurin rajoite tuotekonfiguraattorin osalta on, ettei se tarjoa mitään valmista rajapintaa tuotekoodien validointiin. Ideaalisin tilanne olisi sellainen, että viestillä voisi lähettää tiedot tuotekonfiguraattorille, ja paluuviestinä tulisi tieto siitä, onko tuotekoodi validi vai ei. Tällainen rajapinta voisi olla mahdollista toteuttaa, jos toteuttaisi Java-ohjelmointikielellä liitännäisen rajapintasovellusta varten. Tämä kuitenkin olisi hankalaa ja työlästä. Lisäksi se vaatisi, että tuotekonfiguraattori olisi aina päällä. Tuotekonfiguraattorin vikaantumisriski on tavallista suurempi sen monimutkaisuudesta johtuen.

Tuotekoodien validointisäännöt tuodaan tuotekonfiguraattoriin yleensä tietokannasta. Säännöt voidaan myös kirjoittaa kiinteästi, mutta silloin dynaamisuus kärsii ja muutosten tekeminen on hidasta. Tuotekonfiguraattorin sääntömallin muokkaustyökalu vaatii, että tietokannasta tietoa haettaessa kyselylause alkaa komennolla *SELECT* (Selectica, Inc. 2005: 356). Tämä rajaa *Stored Procedure*, eli tallennettu proseduurin -ohjelmien, käytön pois, sillä tallennettu proseduurin -ohjelma ajetaan komennolla *EXEC*. Tallennettu proseduurin on säilytyspaikka Transact-SQL -kieliselle ohjelmalle. Sen sijaan tietokantafunktiota pystyy käyttämään, sillä se ajetaan *SELECT*-komennolla. Tietokantafunktio on tallennetun proseduurin kaltainen, mutta on ominaisuuksiltaan rajoittuneempi (Vieira 2003: 359).

#### 4.4. Tuotteiden näkyvyys

Tuotteiden näkyvyydelle on kaksi päävaatimusta. Ensinnäkin tulee pystyä määrittelemään, mille järjestelmälle tuote näkyy. Tämä tarkoittaa sitä, että esimerkiksi toiminnanohjausjärjestelmään näkyvä tuote ei näy tuotekonfiguraattorille. Tästä on hyötyä silloin, kun halutaan tilata tuotetta, mutta sitä ei ole ehditty vielä toteuttamaan tuotekonfiguraattoriin. Toiseksi tuotteen näkyvyys tulee pystyä rajaamaan eri maille. Esimerkiksi jos toiminnanohjausjärjestelmää käytetään Kiinasta, tulee kiinalaisten pystyä tilaamaan ainoastaan Kiinan markkinoille tarkoitettua tuotteita.

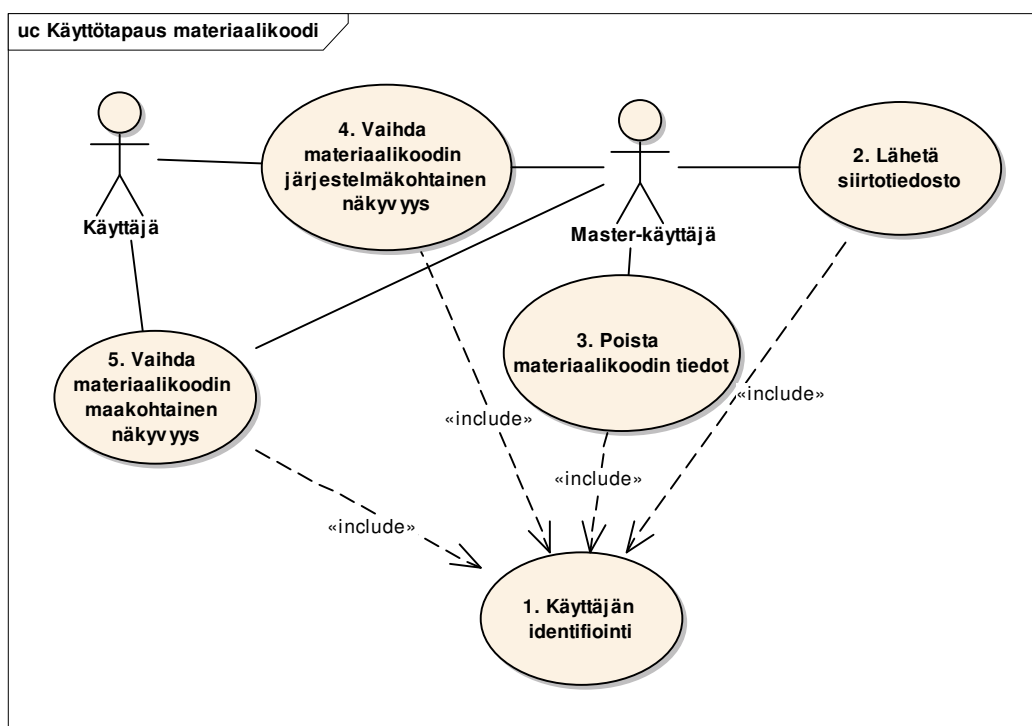
#### 4.5. Ylläpitoliittymä

Ylläpitoliittymä on tarkoitettu tuotetietojen hallintaan. Tärkein toiminto ylläpitoliittymässä on tuotetietojen syöttäminen järjestelmään. Tuotetiedot sisältävät lähes kaiken tuotteeseen liittyvän tiedon, eli validointisäännöt, sallitut merkit tuotekoodissa ja niiden kuvaukset, hinnoittelukoodit ja otsikkotason tiedot. Yllä mainitut tiedot syötetään kertaluonteisesti eikä niitä tarvitse pystyä muuttamaan ajonaikaisesti. Tietojen muuttaminen hoidetaan Excel-taulukkolaskentaohjelmalla, jonka jälkeen tiedot siirretään uudestaan ylläpitoliittymään.

Tämän lisäksi ylläpitoliittymässä tulee pystyä muuttamaan ajonaikaisesti tuotteiden näkyyystietoja. Näin tuotetiedot voidaan ladata järjestelmään etukäteen ja asettaa tuote näkyväksi vasta tuotteen julkaisupäivänä. Tuotteen validointia ja hinnanhakua tulee pystyä testaamaan suoraan ylläpitoliittymässä. Tämä luo nopean väylän tuotetietojen hakemiseen ilman, että käyttäjän tarvitsee hakea tiedot raskaamman järjestelmän kautta. Ylläpitoliittymä on oleellinen tuotetietojen tehokkaan testaamisen kannalta.

#### 4.6. Ylläpitoliittymän käyttötapaukset

Käyttötapaukset on jaettu käyttöliittymänäkymien mukaan. Ensimmäisessä käyttötapauskaaviossa on siirtotiedoston lähettämiseen ja materiaalikoodien hallintaan liittyvät näkymät. Tässä kaaviossa on lisäksi käyttäjän identifiointi –käyttötapaus. Sama käyttötapaus on myös muissa näkymissä, mutta se käsitellään toiston välttämiseksi vain ensimmäisessä kaaviossa. Käyttäjä on pystyttävä identifioimaan, eli käyttäjän on täytynyt kirjautua järjestelmään sisään, jotta tämä voi suorittaa muita käyttötapauksia. Lisäksi käyttötapauksissa on huomioitava, että ”master-käyttäjällä” on oikeus lähettää siirtotiedosto, mutta tavallisella käyttäjällä ei. Käyttötapaukset on esitetty kuvassa 8.



**Kuva 8.** Tuotetietojen hallinnan käyttötapauskaavio

### Käyttötapaus 1 – Käyttäjän identifiointi

**Taulukko 7.** Käyttäjän identifiointi -käyttötapaus

|                |  |
|----------------|--|
| Alkuehdot      | Käyttäjä on kirjautunut verkossa erikseen määriteltyyn Windows Server Domain –toimialueeseen, ja käyttäjän käyttäjätunnukselle on annettu ylläpitoliittymässä tarvittavat käyttöoikeudet.  |
| Syötteet       | Windows Server Domain –toimialueen nimi ja käyttäjätunnus muodossa <i>DOMAINKÄYTTÄJÄTUNNUS</i> .   |
| Kuvaus         | Käyttäjän identifiointi toteutetaan Windows Authentication –menetelmällä. Tällöin riittää, että käyttäjä on kirjautunut sisään Windows-käyttöjärjestelmään, eikä ylläpitoliittymässä tarvita erillistä sisäänkirjautumisnäkyvää. |
| Virhetilanteet | Käyttäjä ei ole kirjautunut Windows Authentication –menetelmällä, käyttäjä on väärässä Windows Server Domain –alueessa tai käyttäjälle ei ole määritelty oikeuksia ylläpitoliittymään.   |
| Tulokset       | Käyttäjä kirjautuu sisään ylläpitoliittymään ja saa käyttäjärooliansa vastaavat oikeudet käyttäen sitä. Sisäänkirjautuneen käyttäjän käyttäjätunnus ja käyttäjärooli tulostetaan käyttöliittymän yläosaan.                       |

## Käyttötapaus 2 – Lähetä siirtotiedosto

**Taulukko 8.** Lähetä siirtotiedosto -käyttötapaus

|                |  |
|----------------|--|
| Alkuehdot      | Käyttäjä kuuluu käyttäjärooliin, jolla on tarvittavat oikeudet lähettää siirtotiedosto.  |
| Syötteet       | XML-muotoinen siirtotiedosto joka on erikseen määritellyn XML-skeeman mukainen.  |
| Kuvaus         | Käyttäjä lähettää siirtotiedoston, jonka mukaiset materiaalikoodin tiedot lisätään tietokantaan.   |
| Virhetilanteet | XML-muotoinen siirtotiedosto ei ole erikseen määritellyn XML-skeeman mukainen.   |
| Tulokset       | Materiaalikoodin mukaiset tiedot lisätään tietokantaan. Vakiona materiaalikoodi on siirron jälkeen ”Ei julkaistu” –julkaisutilassa. Käyttäjä voi myös valita olla lisäämättä tietoja tietokantaan, jolloin käyttäjälle tulostetaan tiedot, millaisena ne olisi menneet tietokantaan, jos käyttäjä olisi näin valinnut. |

## Käyttötapaus 3 – Poista materiaalikoodin tiedot

**Taulukko 9.** Poista materiaalikoodin tiedot -käyttötapaus

|                |  |
|----------------|--|
| Alkuehdot      | Käyttäjä on ”Materiaalikoodit”-näkyvässä. Käyttäjä on määritelty sen mukaiseen käyttäjärooliin, jolla on oikeudet kyseiseen toimenpiteeseen.   |
| Syötteet       | Käyttäjä painaa ”Poista materiaalikoodi” –painiketta, joka välittää materiaalikoodin nimen.  |
| Kuvaus         | Käyttäjä poistaa valitsemansa materiaalikoodin tiedot tietokannasta.   |
| Virhetilanteet | Toinen käyttäjä oli jo poistanut materiaalikoodin tiedot tietokannasta. Käyttäjän käyttörooli vaihdettiin kesken prosessin sellaiseksi, ettei tällä ole oikeuksia kyseiseen toimenpiteeseen. |
| Tulokset       | Materiaalikoodin mukaiset tiedot poistuu tietokannasta ja tästä ilmoitetaan käyttäjälle.   |

## Käyttötapaus 4 – Vaihda materiaalikoodin järjestelmäkohtainen näkyvyys

**Taulukko 10.** Vaihda järjestelmäkohtainen näkyvyys -käyttötapaus

|                |  |
|----------------|--|
| Alkuehdot      | Käyttäjä on ”Materiaalikoodit”-näkyvässä. Käyttäjä on määritelty sen mukaiseen käyttäjärooliin, jolla on oikeudet kyseiseen toimenpiteeseen.   |
| Syötteet       | Alasvetovalikosta valittava uuden järjestelmäkohtaisen näkyvyyden nimi. Tämän jälkeen käyttäjä painaa painiketta, jonka jälkeen tiedot syötetään tietokantaan.                               |
| Kuvaus         | Materiaalikoodin järjestelmäkohtainen näkyvyys vaihdetaan, jolloin materiaalikoodin tiedot näkyvät vain määriteltyihin järjestelmiin.  |
| Virhetilanteet | Toinen käyttäjä oli jo poistanut materiaalikoodin tiedot tietokannasta. Käyttäjän käyttörooli vaihdettiin kesken prosessin sellaiseksi, ettei tällä ole oikeuksia kyseiseen toimenpiteeseen. |
| Tulokset       | Käyttäjälle ilmoitetaan operaation onnistumisesta ja materiaalikoodilistauksessa näkyy uusi järjestelmäkohtainen näkyvyys –tieto.  |

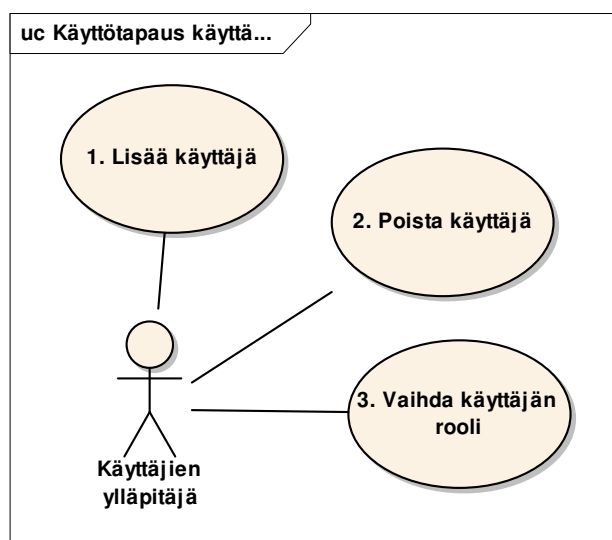
## Käyttötapaus 5 – Vaihda materiaalikoodin maakohtainen näkyvyys

**Taulukko 11.** Vaihda maakohtainen näkyvyys -käyttötapaus

|                |   |
|----------------|---|
| Alkuehdot      | Käyttäjä on ”Materiaalikoodit”-näkyvässä ja käyttäjä painaa painiketta, jolla päästään ”Maakohtaisten näkyvyyksien määrittely” –näkyvään. Käyttäjä on määritelty sen mukaiseen käyttäjärooliin, jolla on oikeudet kyseiseen toimenpiteeseen.              |
| Syötteet       | Käyttäjä valitsee alasvetovalikosta kaksimerkkisen maakoodin, jolle materiaalikoodi asetetaan näkyväksi. Maakoodi on ISO 3166-1 Alpha-2 code –standardin mukainen. Tämän jälkeen käyttäjä painaa painiketta, jonka jälkeen tiedot syötetään tietokantaan. |
| Kuvaus         | Materiaalikoodin maakohtainen näkyvyys vaihdetaan, jolloin materiaalikoodin tiedot näkyvät vain määriteltyihin maakoodien mukaisiin järjestelmiin.  |
| Virhetilanteet | Toinen käyttäjä oli jo poistanut materiaalikoodin tiedot tietokannasta. Käyttäjän käyttörooli vaihdettiin kesken prosessin sellaiseksi, ettei tällä ole oikeuksia kyseiseen toimenpiteeseen.  |
| Tulokset       | Käyttäjälle ilmoitetaan operaation onnistumisesta ja maakohtainen näkyvyys –näkyvässä näytetään tieto uudesta maakohtaisesta näkyvyydestä.  |



Käyttäjänhallintaan liittyvät käyttötapaukset on esitetty kuvassa 9. Käyttötapausten suorittamiseksi käyttäjän tulee olla määritelty ”käyttäjien ylläpitäjä” –käyttörooliin. Normaalilla käyttäjällä ei ole pääsyä ”Käyttäjät”-näkymään, mutta tämä näkee omat käyttäjätietonasa ylläpitoliittymän käyttöliittymän yläosassa.



**Kuva 9.** Käyttäjänhallinnan käyttötapauskaavio

### Käyttötapaus 1 – Lisää käyttäjä

**Taulukko 12.** Lisää käyttäjä -käyttötapaus

|                |  |
|----------------|--|
| Alkuehdot      | Käyttäjä on ”Käyttäjät”-näkyssä. Käyttäjä on määritelty sen mukaiseen käyttäjärooliin, jolla on oikeudet kyseiseen toimenpiteeseen.  |
| Syötteet       | Käyttäjän nimi, käyttäjätunnus ja käyttäjärooli.   |
| Kuvaus         | Käyttäjien ylläpitäjä lisää uuden käyttäjän järjestelmään.   |
| Virhetilanteet | Toinen käyttäjien ylläpitäjä oli jo lisännyt käyttäjän järjestelmään. Käyttäjän käyttörooli vaihdettiin kesken prosessin sellaiseksi, ettei tällä ole oikeuksia kyseiseen toimenpiteeseen. |
| Tulokset       | Käyttäjän tiedot lisätään tietokantaan ja ”Käyttäjät” –näkyman listaukseen ilmestyy uuden käyttäjän tiedot.  |

## Käyttötapaus 2 – Poista käyttäjä

**Taulukko 13.** Poista käyttäjä -käyttötapaus

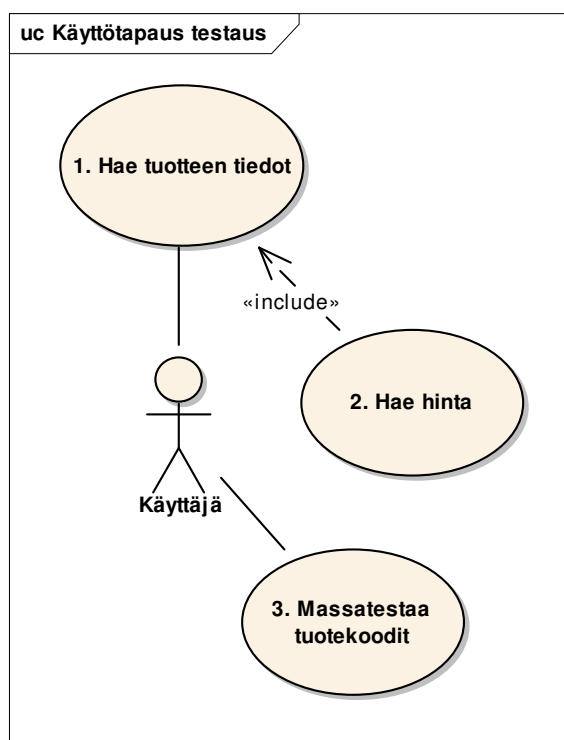
|                |  |
|----------------|--|
| Alkuehdot      | Käyttäjä on ”Käyttäjät”-näkyvässä. Käyttäjä on määritelty sen mukaiseen käyttäjärooliin, jolla on oikeudet kyseiseen toimenpiteeseen.  |
| Syötteet       | Käyttäjien ylläpitäjä painaa käyttäjälistauksessa valitun käyttäjän rivillä olevaa ”Poista käyttäjä” -linkkiä.   |
| Kuvaus         | Valitun käyttäjän tiedot poistetaan tietokannasta.   |
| Virhetilanteet | Toinen käyttäjien ylläpitäjä oli jo poistanut käyttäjän järjestelmästä. Käyttäjän käyttörooli vaihdettiin kesken prosessin sellaiseksi, ettei tällä ole oikeuksia kyseiseen toimenpiteeseen. |
| Tulokset       | Valittu käyttäjä poistuu tietokannasta ja käyttäjälle ilmoitetaan käyttäjän poiston onnistumisesta.  |

## Käyttötapaus 3 – Vaihda käyttäjän rooli

**Taulukko 14.** Vaihda käyttäjän rooli -käyttötapaus

|                |   |
|----------------|---|
| Alkuehdot      | Käyttäjä on ”Käyttäjät”-näkyvässä. Käyttäjä on määritelty sen mukaiseen käyttäjärooliin, jolla on oikeudet kyseiseen toimenpiteeseen.   |
| Syötteet       | ”Käyttäjän rooli” -alasvetovalikon mukainen käyttäjän roolin nimi.  |
| Kuvaus         | Vaihdetaan käyttäjän rooli, joka määrittelee ,mihin toimintoihin käyttäjällä on oikeus järjestelmässä.  |
| Virhetilanteet | Toinen käyttäjien ylläpitäjä oli poistanut käyttäjän järjestelmästä. Käyttäjän käyttörooli vaihdettiin kesken prosessin sellaiseksi, ettei tällä ole oikeuksia kyseiseen toimenpiteeseen. |
| Tulokset       | Käyttäjän rooli vaihtuu käyttäjälistaukseen. Onnistuneesta operaatiosta ilmoitetaan käyttäjälle.  |

Tuotetietojen testaamiseen liittyvät käyttötapaukset on esitetty kuvassa 10. ”Hae hinta” -käyttötapaus on riippuvainen ”Hae tuotteen tiedot” -käyttötapauksesta, koska tämä käyttötapaus palauttaa tarvittavia syötetietoja.



**Kuva 10.** Testaamisen käyttötapauskaavio

### Käyttötapaus 1 – Hae tuotteen tiedot

**Taulukko 15.** Hae tuotteen tiedot -käyttötapaus

|                |   |
|----------------|---|
| Alkuehdot      | Käyttäjä on ”Tuotekoodin testaaminen” -näkyvässä.   |
| Syötteet       | Tuotekoodi ja maakoodi.   |
| Kuvaus         | Päätellään materiaalikoodi tuotekoodin perusteella, validoidaan tuotekoodi ja haetaan hinnoittelukoodi. |
| Virhetilanteet | Materiaalikoodin mukaiset tuotetiedot olivat poistuneet tietokannasta.                                  |
| Tulokset       | Validointitieto ja hinnoittelukoodi.  |

## Käyttötapaus 2 – Hae hinta

**Taulukko 16.** Hae hinta -käyttötapaus

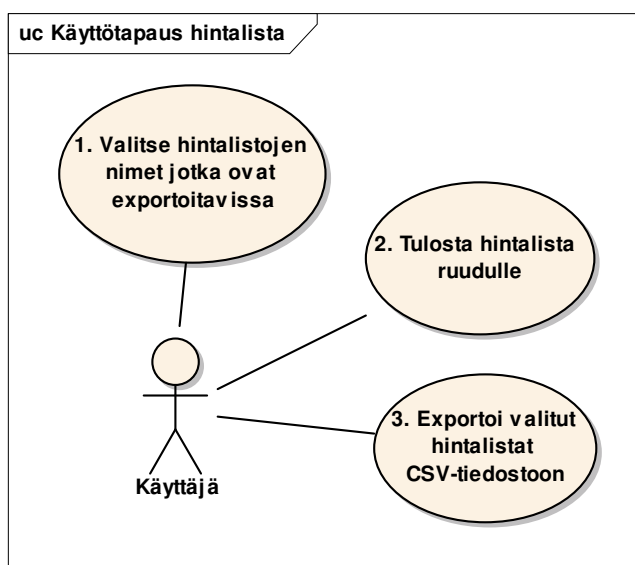
|                |  |
|----------------|--|
| Alkuehdot      | Käyttäjä on ”Tuotekoodin testaaminen”-näkyvässä.   |
| Syötteet       | Materiaalikoodi, hinnoittelukoodi, hintalistan nimi ja valuutta.   |
| Kuvaus         | Haetaan hinnoittelukoodia vastaava hinta.  |
| Virhetilanteet | Materiaalikoodin mukaiset tuotetiedot olivat poistuneet tietokannasta.   |
| Tulokset       | Hinnoittelukoodia vastaava hinta tai jos jonkin hintakoodin arvo on tietokannassa NULL, palautetaan tästä virheilmoitus. |

## Käyttötapaus 3 – Massatestaa tuotekoodit

**Taulukko 17.** Massatestaa tuotekoodit -käyttötapaus

|                |  |
|----------------|--|
| Alkuehdot      | Käyttäjä on ”Tuotekoodin testaaminen”-näkyvässä.                                 |
| Syötteet       | Lista tuotekoodeista rivinvaihtomerkillä erotettuna.                             |
| Kuvaus         | Käyttäjä syöttää usean tuotekoodin kerralla.                                     |
| Virhetilanteet | Materiaalikoodien mukaiset tuotetiedot olivat poistuneet tietokannasta.          |
| Tulokset       | Listaus syötetyistä tuotekoodeista, validointitiedoista ja hinnoittelukoodeista. |

Hintalistojen exportoiminen, eli ulosvieminen, ei ole keskeinen osa ylläpitoliittymää, mutta se on lisätty siihen siksi, että tiedot ovat helposti saatavilla ja ylläpitoliittymä tarjoaa tälle toiminnallisuudelle valmiin alustan. Käyttötapauskuvat on esitetty kuvassa 11.



**Kuva 11.** Hintalistan exportoinnin käyttötapauskaavio

### Käyttötapaus 1 – Valitse hintalistojen nimet jotka ovat exportoitavissa

**Taulukko 18.** Valitse hintalistojen nimet, jotka ovat exportoitavissa -käyttötapaus

|                |  |
|----------------|--|
| Alkuehdot      | Käyttäjä on näkyvässä, jossa valitaan hintalistojen nimet, jotka ovat exportoitavissa.   |
| Syötteet       | Hintalistojen nimet, joiden tulee olla exportoitavissa.  |
| Kuvaus         | Käyttäjä valitsee monivalintalistasta niiden hintalistojen nimet, jotka tulee olla exportoitavissa.  |
| Virhetilanteet | Valitun hintalistan nimi poistuu tietokannasta ennen kuin se syötetään tietokantaan.   |
| Tulokset       | Valitut hintalistan nimet syötetään tietokantaan. Syötön jälkeen valitut hintalistan nimet ilmestyvät valittujen hintalistanimien listaan. |

## Käyttötapaus 2 – Tulosta hintalista ruudulle

**Taulukko 19.** Tulosta hintalista ruudulle -käyttötapaus

|                |   |
|----------------|---|
| Alkuehdot      | Käyttäjä on ”Hintalistojen exportointi”-näkyvässä.  |
| Syötteen       | Hintalistan nimi.   |
| Kuvaus         | Käyttäjä valitsee hintalistan nimen, jonka mukainen hintalista tulostetaan ruudulle. Käyttäjä pystyy tulostamaan vain yhden hintalistan kerrallaan, johtuen hintalistojen massiivisesta koosta. Muutoin web-selainohjelman käyttö voi hidastua huomattavasti. |
| Virhetilanteet | Valitun hintalistan nimi poistuu tietokannasta hintalistan nimen valinnan ja tulostuspyynnön välissä.   |
| Tulokset       | Näytölle tulostetaan tuotteiden hintalista. Hintalistassa on tuotteen nimi ja hinta.  |

## Käyttötapaus 3 – Exportoi valitut hintalistat CSV-tiedostoon

**Taulukko 20.** Exportoi valitut hintalistat CSV-tiedostoon -käyttötapaus

|                |  |
|----------------|--|
| Alkuehdot      | Käyttäjä on ”Hintalistojen exportointi”-näkyvässä.   |
| Syötteen       | Hintalistojen nimet.   |
| Kuvaus         | Käyttäjä valitsee monivalintalistasta usean hintalistan nimen, joiden mukaiset hintalistat exportoidaan CSV-tiedostoon. Tätä tiedostoa käytetään erillisessä hintaohjelmassa.  |
| Virhetilanteet | Valittujen hintalistojen nimet poistuu tietokannasta hintalistan nimien valinnan ja export-pyynnön välissä.  |
| Tulokset       | Käyttäjä saa ladattavakseen CSV-tiedoston. Tiedostossa on hintalistojen sisältö CSV-muodossa, jossa on neljä kenttää; hintalistan nimi, valuutan nimi, tuotteen nimi ja hinta. |

## 5. TUOTETIETOJEN MÄÄRITTELY

Asiakas on päättänyt tuotekoodin määrittelyssä tiettyyn tapaan, jolla saadaan ilmaistua kaikki tarvittavat asiat tuotekoodiin liittyen. Määrittely on toteutettu Excel-taulukossa, johon on sisäänrakennettuna ohjelma, jolla tiedot pystytään siirtämään ylläpitoliittymän ymmärtämään muotoon. Tuotteen määrittely on aina materiaalikohtainen.

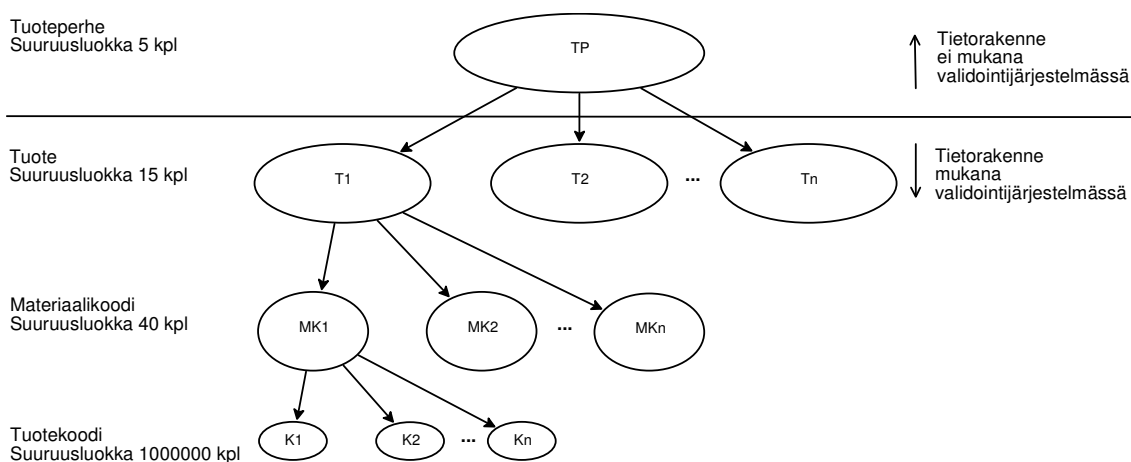
### 5.1. Tietomallin hierarkia

Ylin taso tietomallin hierarkiassa on tuoteperhe. Samaan tuoteperheeseen liittyy samankaltaisia tuotteita. Tuoteperheeksi voisi mieltää vaikkapa elektroniikkamessuja varten suunnitellun tietokoneen keskusyksikön malliston. Tuoteperhe tietorakenteena ei ole tarpeen validointijärjestelmässä, sillä käyttäjän hakiessa tietoja on tämä kiinnostunut tuotteen eikä tuoteperheen tiedoista. Tuoteperheitä ei ole kovin montaa vaan tällä hetkellä niitä on viisi kappaletta.

Tuote on validointijärjestelmästä löytyvistä tietorakenteista ylin taso. Tuotekohtaisia tietoja ovat esimerkiksi tuotekoodin pituus ja tuotokuva. Kaikki tuotteeseen liittyvät materiaalikoodit näkyvät tuotekonfiguraattorissa samassa näkymässä. Tuotteita on tällä hetkellä 15 kappaletta.

Materiaalikoodi on tuotetta yhtä alempi taso. Materiaalikoodi on tietorakenteista tärkein, sillä siihen on sidottu tuotekoodin sallitut merkit ja niiden kuvaukset, materiaalikoodin tunnistus tuotekoodista, validointisäännöt ja hinnoittelusäännöt. Materiaalikoodin voisi määrittellä tuotteen versioksi. Kun tuote kehittyy, niin materiaalikoodiin voidaan sijoittaa tuotteen versio ja esimerkiksi standardi, minkä maan markkinoille tuote on tarkoitettu. Materiaalikoodeja on tällä hetkellä noin 40 kappaletta.

Tuotekoodi on tietomallin alin taso. Tuotekoodi on modulaarinen, joka tarkoittaa, että ei ole olemassa tietuetta, johon on listattuna kaikki mahdolliset tuotekoodit. Tuotekoodin pituus on määritelty tuotteen tasolla, mutta eri tuotteilla voi olla eripituiset tuotekoodit. Tuotekoodin perusteella yritetään selvittää tuotteen hinta ja tuotantoa avustava osaluettelo. Tuotekoodeja voi olla siis dynaaminen määrä ja jos kaikki yhdistelmät käytäisiin läpi, olisi tuotekoodeja tuotekoodin pituudesta ja merkkien määrästä riippuen jopa miljoonia. Tietomallin hierarkia on esitetty kuvassa 12.



**Kuva 12.** Tietomallin hierarkia

## 5.2. Tuotekoodin merkkien sisältö

Tietyn tuotteen tuotekoodi on ennalta määrätyn pituinen. Materiaalikoodi on sidottu tuotteeseen ja tällöin tiedetään kuinka monta merkin indeksia tulee materiaalikoodille määritellä. Yhden merkin arvoksi ovat varattu kaikki numeeriset merkit väliltä 0–9 ja literaaliset merkit väliltä A–Z. Skandinaaviset merkit eivät ole sallittuja, sillä ne voivat aiheuttaa merkistöongelmia tietojärjestelmissä.

Taulukossa 21 on esitetty indeksikohtaisesti, mitä merkki tarkoittaa ja mitkä ovat sen mahdolliset arvot. Esimerkiksi merkin indeksissä kaksi on kerrottu tietokoneen keskusyksikön kotelon eri vaihtoehdot. Jos merkin indeksissä kaksi on merkki *B*, on kotelo tällöin *Full-tower ATX*.



Merkkien indeksien viisi ja kuusi kohdalla merkit menevät pareittain. Se tarkoittaa, että yhdelle ominaisuudelle, tässä tapauksessa prosessorijäähdyttimelle, on varattu kaksi perättäistä merkkiä. Tämä sallii sen, että yhdelle ominaisuudelle on varattu huomattavasti enemmän vaihtoehtoja kuin mitä olisi yhdellä merkillä. Kun yhdellä merkillä vaihtoehtoja on 67 kappaletta, on kahdella merkillä vaihtoehtoja 67 kertaa 67 eli 4489 kappaletta. Taulukon 21 esimerkissä eri vaihtoehtoja on ainoastaan yhdeksän kappaletta, jotta tuote pysyisi helpommin käsiteltävänä.

**Taulukko 21.** Tuotekoodin merkkien sisältö

| Otsikko                       | Merkki nro. | Kuvaus                 | Merkki |
|-------------------------------|-------------|------------------------|--------|
| Tuoteperheen identifioiva osa | 1           | Tuoteperhe Office      | O      |
| Kotelo                        | 2           | Miditorni ATX          | A      |
|                               | 2           | Full-tower ATX         | B      |
|                               | 2           | mATX desktop           | C      |
|                               | 2           | mATX torni             | D      |
| Virtalähde                    | 3           | 300 W                  | A      |
|                               | 3           | 400 W                  | B      |
|                               | 3           | 400 W Silent           | C      |
|                               | 3           | 550 W                  | D      |
| Proessori                     | 4           | AMD AM3 2000 MHz       | A      |
|                               | 4           | AMD AM3 2200 MHz       | B      |
|                               | 4           | AMD AM3 2500 MHz       | C      |
|                               | 4           | Intel LGA1155 2600 MHz | D      |
|                               | 4           | Intel LGA1155 2700 MHz | E      |
|                               | 4           | Intel LGA1155 3000 MHz | F      |
| Proessorijäähdytin            | 5,6         | Stock cooler           | NN     |
|                               | 5,6         | Alumiininen 32 dB      | AA     |
|                               | 5,6         | Alumiininen 28 dB      | AB     |
|                               | 5,6         | Alumiininen 25 dB      | AC     |
|                               | 5,6         | Alumiininen 22 dB      | AD     |
|                               | 5,6         | Kuparinen 32 dB        | BA     |
|                               | 5,6         | Kuparinen 28 dB        | BB     |
|                               | 5,6         | Kuparinen 25 dB        | BC     |
|                               | 5,6         | Kuparinen 22 dB        | BD     |
| Muisti                        | 7           | 1 GB DDR3              | A      |
|                               | 7           | 2 GB DDR3              | B      |
|                               | 7           | 4 GB DDR3              | C      |
| Emolevy                       | 8           | AM3 ATX                | A      |

|              |    |                     |   |
|--------------|----|---------------------|---|
|              | 8  | AM3 mATX            | B |
|              | 8  | LGA1155 ATX         | C |
|              | 8  | LGA1155 ATX + VIDEO | D |
| Näytönohjain | 9  | Ei näytönohjainta   | N |
|              | 9  | Radeon 512 MB DDR3  | A |
|              | 9  | GeForce 1 GB DDR4   | B |
| DVD-asema    | 10 | Ei DVD-asemaa       | N |
|              | 10 | DVD-ROM 48x         | A |
|              | 10 | DVD-RW 48x / 12x    | B |
| Versio       | 11 | Versio A            | A |

### 5.3. Materiaalikoodin tunnistaminen tuotekoodista

Kuten teoriaosuudessa kerrottiin, on tuotekoodissa tunnistava osa ja versio-osa. Tässä esimerkissä materiaalikoodi tunnistetaan näiden yhdistelmästä. Tuotekoodi on tuotteen perustuen dynaamisen mittainen, jolloin on mahdollista, että toisella tuotteella, jolla on pidempi tuotekoodi, tunnistetaan materiaalikoodi väärin. Tämä on kuitenkin varmistettu tuotteen määrittelyn tasolla, että tunnistusmerkit valitaan siten, että ristiriitaa ei pääse tapahtumaan. Taulukossa 22 on esitetty tuotteen tunnistus, jossa ensimmäinen merkki *O* kertoo tuoteperheen. Yhdestoista merkki *A* kertoo tuotteen version.

#### **Taulukko 22.** Materiaalikoodin tunnistaminen tuotekoodista

| <b>Merkki nro.</b> | <b>Kuvaus</b>     | <b>Merkki</b> |
|--------------------|-------------------|---------------|
| 1                  | Tuoteperhe Office | O             |
| 11                 | Versio A          | A             |

### 5.4. Validointisäännöt

Taulukossa 23 on esitetty tuotteen validointisäännöt. Siinä on yhteensä seitsemän eri sääntöä ja ne on eroteltu toisistaan sääntönumerosarakkeen arvon mukaan. Säännöt eroavat toisistaan riippuen siitä, mihin eri tuotekoodin merkkien indeksiin niissä otetaan kantaa. Esimerkiksi sääntö numero yksi ottaa kantaa indekseihin yksi, kaksi, kahdeksan ja yksitoista. Sallitut merkit on erotettu toisistaan pilkulla. Indeksit yksi ja yksitoista

kulkevat säännöissä mukana sen vuoksi, että koska ne ovat materiaalikoodin tunnistavat merkit, saadaan nämä säännöt kohdistettua vain halutulle materiaalikoodille. Seuraavissa tarkasteluissa ei toisteta näitä indeksejä, sillä ne ovat jokaiselle säännölle samat.

**Taulukko 23.** Validointisäännöt

| Sääntö nro. | M1 | M2      | M3      | M4    | M5  | M6    | M7    | M8      | M9  | M10   | M11 |
|-------------|----|---------|---------|-------|-----|-------|-------|---------|-----|-------|-----|
| 1           | O  | A,B     |         |       |     |       |       | A,C,D   |     |       | A   |
| 1           | O  | A,B,C,D |         |       |     |       |       | B       |     |       | A   |
| 2           | O  |         | A,B,C,D |       |     |       |       |         |     |       | A   |
| 3           | O  |         |         | A,B,C |     |       |       | A,B     |     |       | A   |
| 3           | O  |         |         | D,E,F |     |       |       | C,D     |     |       | A   |
| 4           | O  |         |         |       | N   | N     |       |         |     |       | A   |
| 4           | O  |         |         |       | A,B | A,B,C |       |         |     |       | A   |
| 5           | O  |         |         |       |     |       | A,B,C |         |     |       | A   |
| 6           | O  |         |         |       |     |       |       | D       | N   |       | A   |
| 6           | O  |         |         |       |     |       |       | A,B,C,D | A,B |       | A   |
| 7           | O  |         |         |       |     |       |       |         |     | N,A,B | A   |

Jos ensimmäisen säännön pilkut kertoo auki, saa säännön taulukon 24 esittämään muotoon. Ensimmäisessä säännössä siis otetaan kantaa merkkien yksi, kaksi, kahdeksan ja yksitoista mukaisten indeksien sisältöön. Indeksit kolme, neljä, viisi, kuusi, seitsemän, yhdeksän ja kymmenen eivät ole mukana säännössä.

Jos tarkastelemme keskusyksikön fyysisiä ominaisuuksia, niin kotelon standardi voi olla joko *ATX* tai *mATX*. Tällä tarkoitetaan mittoja, minkä kokoinen emolevy keskusyksikköön mahtuu sisälle. Emolevyjä on niin ikään joko *ATX* tai *mATX* standardin mukaisia. *ATX*-koteloon mahtuu sisälle joko *ATX*- tai *mATX*-standardin mukainen emolevy, mutta *mATX*-standardin mukaiseen koteloon ei mahdu sisälle *ATX*-standardin mukainen emolevy.

Taulukon 21 mukaisesti toisen indeksin mukaisen merkin arvot *A* ja *B* vastaavat *ATX*-standardin mukaista koteloa ja arvot *C* ja *D* *mATX*-standardin mukaista koteloa. Vastavasti kahdeksannen merkin arvot *A*, *C* ja *D* vastaavat *ATX*-standardin mukaista koteloa ja arvo *B* vastaa *mATX*-standardin mukaista emolevyä.

Taulukosta 24 näemmekin, että jos toinen merkki on arvoltaan *C* (*mATX desktop*), niin tällöin kahdeksannen merkin on oltava arvoltaan *B* (*AM3 mATX*). Asian voi myös ajatella toisinpäin, eli jos kahdeksannen merkin arvoksi valitaan *B* (*AM3 mATX*), niin silloin toinen merkki voi olla arvoltaan joko *A* (*Miditorni ATX*), *B* (*Full-tower ATX*), *C* (*mATX desktop*) tai *D* (*mATX torni*). Toisin sanoen, jos keskusyksikköön tulee emolevy *AM3 mATX*, niin kotelo voi olla mikä tahansa.

**Taulukko 24.** Ensimmäinen sääntö aukikerrottuna

| Sääntö nro. | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | M11 |
|-------------|----|----|----|----|----|----|----|----|----|-----|-----|
| 1           | O  | A  |    |    |    |    |    | A  |    |     | A   |
| 1           | O  | A  |    |    |    |    |    | C  |    |     | A   |
| 1           | O  | A  |    |    |    |    |    | D  |    |     | A   |
| 1           | O  | B  |    |    |    |    |    | A  |    |     | A   |
| 1           | O  | B  |    |    |    |    |    | C  |    |     | A   |
| 1           | O  | B  |    |    |    |    |    | D  |    |     | A   |
| 1           | O  | A  |    |    |    |    |    | B  |    |     | A   |
| 1           | O  | B  |    |    |    |    |    | B  |    |     | A   |
| 1           | O  | C  |    |    |    |    |    | B  |    |     | A   |
| 1           | O  | D  |    |    |    |    |    | B  |    |     | A   |

Säännöt ovat vuorovaikutuksessa keskenään. Tämä tarkoittaa sitä, että jos esimerkiksi ensimmäisen säännön nojalla toisen merkin arvo *C* oli rajannut merkin kahdeksan arvoksi *B*, koskee tämä rajaus automaattisesti myös niitä sääntöjä, jotka ottavat kahdeksanteen merkkiin kantaa. Tällaisia ovat säännöt numero kolme ja kuusi. Tällöin näiden sääntöjen nojalla neljännen merkin arvojen täytyy olla *A* (*AMD AM3 2000 MHz*), *B* (*AMD AM3 2200 MHz*) tai *C* (*AMD AM3 2500 MHz*). Samoin yhdeksännen merkin arvojen täytyy olla *A* (*Radeon 512 MB DDR3*) tai *B* (*GeForce 1 GB DDR4*). Toisin sanoen, jos emolevy on *AM3*-kantainen, käy emolevylle vain *AM3*-kantainen *AMD*-valmistajan prosessori. Samoin koneeseen täytyy valita erillinen näytönohjain, sillä emolevylle integroitu näytönohjain löytyy ainoastaan *LGA1155*-kantisesta emolevystä.

Säänöistä rajaavia ovat numero yksi, kolme ja kuusi. Tämä tarkoittaa sitä, että tietyn komponentin valinta rajaa jonkin toisen komponentin vaihtoehtoja. Säännöt numero kaksi, neljä ja seitsemän eivät ole rajaavia, sillä niissä vain luetellaan kaikki mahdolliset arvot, mitkä löytyvät taulukosta 21. Ei-rajaavat säännöt on esitetty taulukossa helpottamaan tuotekoodin muodostamisen hahmottamista. Jokaisen säännön täytyy toteutua ja jokaisen tuotekoodin merkin indeksin on saatava arvo, jotta tuotekoodi on validi. Esimerkiksi tuotekoodi *OAABNNABANA* on validi, sillä säännöt eivät ole ristiriidassa. Tuotekoodi *OAABNNABNNA* ei ole validi, sillä yhdeksännen merkin arvo on *N* (*Ei näytönohjainta*), mutta se ei ole mahdollista, sillä säännön kuusi nojalla *N*-merkki on mahdollinen vain, jos kahdeksännen merkin arvo on *D* (*LGAI155 ATX + VIDEO*). Tuotekoodi *OGABNNABANA* ei ole validi, sillä toinen merkki on arvoltaan *G* ja tätä merkkiä ei ole esitetty taulukossa 21 lainkaan.

### 5.5. Tuotekoodin varianttien määrän tarkastelua

Ellei tuotteella ole validointisääntöjä, lasketaan varianttien määrä suoraan tuloperiaatteen mukaisesti, kuten on esitetty kaavassa 2. Tällöin varianttien määrä on suurimmillaan. Varianttien määrä on laskettuna taulukossa 26, jossa niiden kokonaismääräksi saatiin 93312.

**Taulukko 25.** Varianttien määrä ilman validointisääntöjä

| M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | M11 | Varianttien määrä |
|----|----|----|----|----|----|----|----|----|-----|-----|-------------------|
| 1  | 4  | 4  | 6  | 9  | 3  | 4  | 1  | 3  | 3   | 1   | 93312             |

Jotta varianttien määrä voitaisiin laskea huomioon ottaen validointisäännöt, tulee ne yhdistää yhdeksi säännöksi, joka käsittelee kaikkia tuotekoodin merkkejä. Yhdistämisessä täytyy ottaa huomioon, että sääntörivit eivät saa olla ristiriidassa keskenään. Taulukossa 26 on yhdistetty taulukon 23 validointisäännöt yhdeksi säännöksi.

**Taulukko 26.** Validointisäännöt yhdistettynä yhdeksi säännöksi

| Sääntö nro. | M1 | M2      | M3      | M4    | M5  | M6    | M7    | M8 | M9    | M10   | M11 |
|-------------|----|---------|---------|-------|-----|-------|-------|----|-------|-------|-----|
| 1           | O  | A,B     | A,B,C,D | A,B,C | N   | N     | A,B,C | A  | A,B   | N,A,B | A   |
| 1           | O  | A,B,C,D | A,B,C,D | A,B,C | N   | N     | A,B,C | B  | A,B   | N,A,B | A   |
| 1           | O  | A,B     | A,B,C,D | D,E,F | N   | N     | A,B,C | C  | A,B   | N,A,B | A   |
| 1           | O  | A,B     | A,B,C,D | D,E,F | N   | N     | A,B,C | D  | N,A,B | N,A,B | A   |
| 1           | O  | A,B     | A,B,C,D | A,B,C | A,B | A,B,C | A,B,C | A  | A,B   | N,A,B | A   |
| 1           | O  | A,B,C,D | A,B,C,D | A,B,C | A,B | A,B,C | A,B,C | B  | A,B   | N,A,B | A   |
| 1           | O  | A,B     | A,B,C,D | D,E,F | A,B | A,B,C | A,B,C | C  | A,B   | N,A,B | A   |
| 1           | O  | A,B     | A,B,C,D | D,E,F | A,B | A,B,C | A,B,C | D  | N,A,B | N,A,B | A   |
| 1           | O  | A,B     | A,B,C,D | A,B,C | N   | N     | A,B,C | A  | A,B   | N,A,B | A   |
| 1           | O  | A,B,C,D | A,B,C,D | A,B,C | N   | N     | A,B,C | B  | A,B   | N,A,B | A   |
| 1           | O  | A,B     | A,B,C,D | D,E,F | N   | N     | A,B,C | C  | A,B   | N,A,B | A   |

Jotta varianttien määrä voitaisiin laskea näistä, tulee merkit korvata merkkien vaihtoehtojen määrällä ja tämän jälkeen laskea rivikohtaisesti varianttien lukumäärä tuloperiaatteen mukaisesti. Tämän jälkeen rivikohtaiset varianttien määrät lasketaan yhteen, jolloin saadaan tuotekoodin varianttien kokonaismäärä. Tämä on esitetty taulukossa 27. Validointisäännöt huomioon ottaen varianttien määräksi tulee 16632, joka on vain noin 18 % varianttien määrästä ilman validointisääntöjä.

**Taulukko 27.** Varianttien määrä huomioon ottaen validointisäännöt

| Sääntö nro.                      | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | M11 | Varianttien määrä |
|----------------------------------|----|----|----|----|----|----|----|----|----|-----|-----|-------------------|
| 1                                | 1  | 2  | 4  | 3  | 1  | 1  | 3  | 1  | 2  | 3   | 1   | 432               |
| 1                                | 1  | 4  | 4  | 3  | 1  | 1  | 3  | 1  | 2  | 3   | 1   | 864               |
| 1                                | 1  | 2  | 4  | 3  | 1  | 1  | 3  | 1  | 2  | 3   | 1   | 432               |
| 1                                | 1  | 2  | 4  | 3  | 1  | 1  | 3  | 1  | 3  | 3   | 1   | 648               |
| 1                                | 1  | 2  | 4  | 3  | 2  | 3  | 3  | 1  | 2  | 3   | 1   | 2592              |
| 1                                | 1  | 4  | 4  | 3  | 2  | 3  | 3  | 1  | 2  | 3   | 1   | 5184              |
| 1                                | 1  | 2  | 4  | 3  | 2  | 3  | 3  | 1  | 2  | 3   | 1   | 2592              |
| 1                                | 1  | 2  | 4  | 3  | 2  | 3  | 3  | 1  | 3  | 3   | 1   | 3888              |
| 1                                | 1  | 2  | 4  | 3  | 1  | 1  | 3  | 1  | 2  | 3   | 1   | 432               |
| 1                                | 1  | 4  | 4  | 3  | 1  | 1  | 3  | 1  | 2  | 3   | 1   | 864               |
| 1                                | 1  | 2  | 4  | 3  | 1  | 1  | 3  | 1  | 2  | 3   | 1   | 432               |
| <b>Varianttien kokonaismäärä</b> |    |    |    |    |    |    |    |    |    |     |     | 16632             |

## 5.6. Hinnoittelukoodin muodostaminen

Nimensä mukaisesti hinnoittelukoodin avulla muodostetaan tuotteen hinta, mutta se on myös lisätietoa tuotannolle, mitä eri komponentteja tuotteeseen tulee. Hinnoittelukoodi muodostuu koodien palasista, joista jokaisella on oma hinta. Kun näiden koodipalasten hinnat lasketaan yhteen, saadaan tuotteen kokonaishinta.

Hinnoittelukoodin muodostaminen edellyttää, että tuotekoodi on jo tiedossa. Taulukossa 28 on esitetty koodit, hinnat ja minkä tilauskoodien merkkien perusteella koodi lisätään hinnoittelukoodiin. Esimerkiksi, jos tuotekoodin toinen merkki on arvoltaan A, niin tällöin hinnoittelukoodiin lisätään koodi *CSE01A*.

Hinnoittelukoodi alkaa aina pääkoodilla. Pääkoodin sisällyttämiseen ovat samat kriteerit kuin materiaalikoodin tunnistamiseen, joten pääkoodi tulee aina hinnoittelukoodin alkuun. Pääkoodilla ei välttämättä ole edes hintaa, mutta pääkoodi lisätään selvennyksen vuoksi, jotta tuote on helpompi tunnistaa hinnoittelukoodin perusteella. Muut koodit muodostuvat hinnoittelukoodiin samassa järjestyksessä kuin ne ovat taulukossa 28.

Koodien välillä käytetään sovittua erotinmerkkiä, kuten +. Jos tuotekoodi on *OAABN-NABANA*, niin tällöin hinnoittelukoodiksi muodostuu *OFFICE01A+CSE01A+PWR01A+CPU02A+RAM01A+MBO02A+GPU01A+SFW01A+SFW02A+FANO1A*. Jos näiden palasten hinnat lasketaan yhteen, saadaan summaksi  $0+50+30+120+30+90+70+90+40+10$  eli 530. Hinnoittelukoodiin voidaan lisätä materiaalikoodista riippuen osia, jotka eivät muuten ilmene tuotekoodista. Taulukon 28 lopussa on kolme koodia, jotka tulevat hinnoittelukoodiin automaattisesti, kunhan materiaalikoodin tunnistamiseksi tarvittavat tiedot ilmenee tuotekoodista (tässä tapauksessa merkin indeksi yksi ja yksitoista). Esimerkiksi kaikissa *OFFICE01A* materiaalikoodin keskusyksiköissä toimitetaan aina mukana ohjelmistopaketti (*Microsoft Office 2010* ja *WinZip 15.0*) ja kotelon takatuuletin (*Takatuuletin 120mm 21 dB*). Näin tieto välittyy tuotannolle suoraan hinnoittelukoodin mukana eikä tuotannon tarvitse tehdä ylimääräisiä tarkistuksia jostain toisesta dokumentista.





## 6. JÄRJESTELMÄN TOTEUTUS

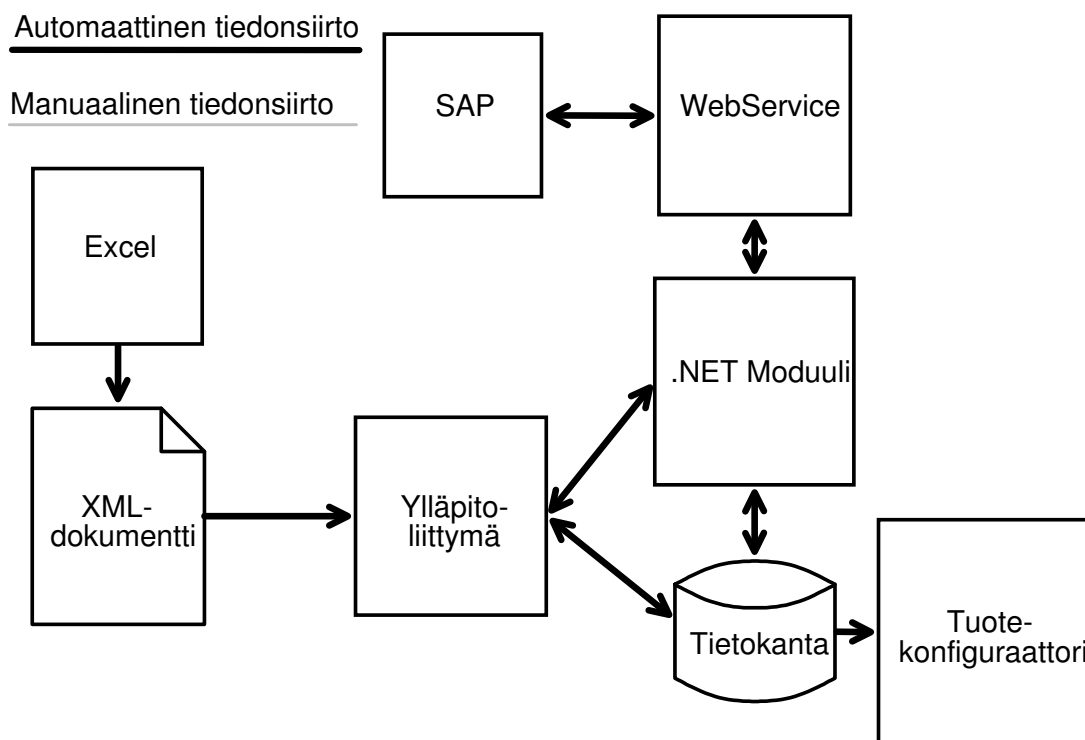
Toteutuksen kannalta olennaisinta on, miten tarjota validointitiedot samasta tietolähteestä useammalle eri järjestelmälle. Tässä jouduttiin ottamaan huomioon tuotekonfiguraattorin asettamat rajoitukset. Toteutuksessa on pyritty löytämään ne tavat ja tekniikat, jotka vastaavat juuri tämän järjestelmän vaatimuksiin.

### 6.1. Yleiskatsaus järjestelmään

Tuotteen määrittely tehdään Excel-taulukkolaskentaohjelmassa, josta erillisen ohjelman avulla tuotetaan XML-muotoinen (Extensible Markup Language) siirtotiedosto. XML on World Wide Web Consortiumin (W3C) laatima laitteisto- ja alustariippumaton tekstipohjainen kieli minkä tahansa tiedon kuvaamiseen (Quin 2011). Tämä siirtotiedosto ladataan ylläpitoliittymässä, joka lisää tuotteen tiedot tietokantaan. .NET moduuli on Microsoft .NET -ympäristössä tehty koodikirjasto, jossa on yleisimmin käytetyt metodit kuten tuotekoodin validointi ja hinnanhaku. Ylläpitoliittymässä voidaan käyttää tämän koodikirjaston metodeja, mutta ylläpitoliittymään voidaan myös hakea tietoa suoraan tietokannasta.

Tuotekonfiguraattori hakee tietoa suoraan tietokannasta, mutta ei tallenna tietokantaan mitään. Joissakin tapauksissa tuotekonfiguraattori käyttää samoja tietokantafunktioita kuin .NET moduuli. Tällä tavoin pyritään varmistamaan, että tiedot haetaan järjestelmiin samalla tavalla.

Rajapinta SAP-toiminnanohjausjärjestelmään toteutetaan Web Service -rajapintasovelluksella. Web Service -rajapintasovellus käyttää ainoastaan niitä metodeja, joita .NET moduuli tarjoaa. Web Service -rajapintasovellus luo rajapinnan, jonka avulla järjestelmä kommunikoi SAP-toiminnanohjausjärjestelmän kanssa (Haas, Brown 2004). Järjestelmän kommunikaatiokaavio on esitetty kuvassa 13.



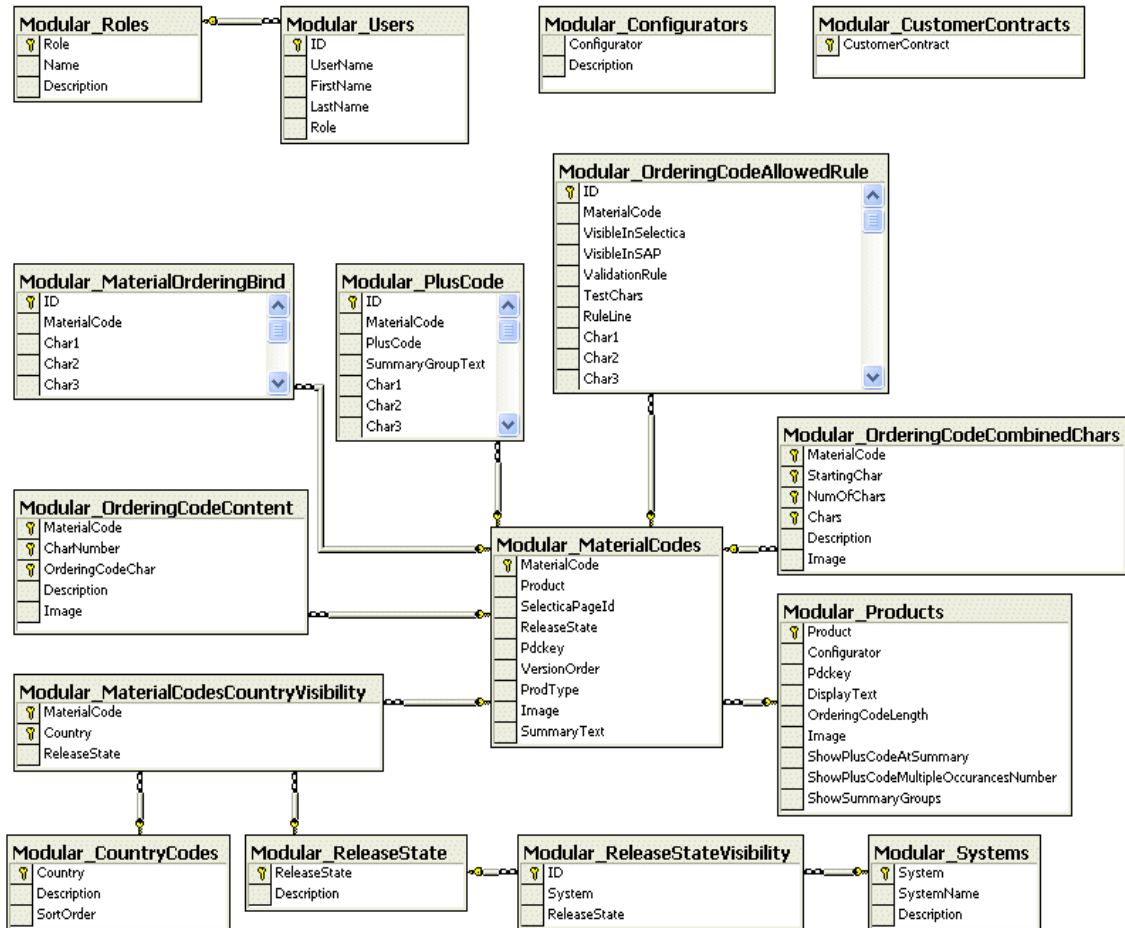
**Kuva 13.** Kommunikaatiokaavio

## 6.2. Tietokantamalli

Tietokantamallin taulut on nimetty etumääreellä *Modular\_*. Tällöin järjestelmään liittyvät taulut on helpompi erotella muista tauluista, sillä samassa tietokannassa on myös muiden järjestelmien tauluja. Tietokannassa keskiössä on taulu *Modular\_MaterialCodes*, koska useimmat tiedot ovat materiaalikoodikohtaisia. Taulu sisältääkin pääavaimen monelle muulle taululle. Korkeimmalla tietokantamallin hierarkiassa on kuitenkin taulu *Modular\_Products*, sillä yhdellä tuotteella voi olla useita materiaali-koodeja.

Kuvassa 14 on esitetty tietokantataulujen väliset suhteet (*FOREIGN* ja *PRIMARY KEY* -viittaukset). Lisäksi siitä selviää, mitä eri kenttiä tauluissa on. Taulujen *Modular\_MaterialOrderingBind*, *Modular\_PlusCode* ja *Modular\_OrderingCodeAllowedRule* kentistä ei näy viimeiset merkkikentät (*Char4–Char18*), jotta kuva ei veisi kohtuutto-

masti tilaa. Tietokantataulut siis rajaavat tuotekoodin pituuden maksimissaan 18 merkkiin, mutta tätä pidempiä tuotekoodeja ei ole toistaiseksi tarvittu. Jotta järjestelmässä voitaisiin käsitellä pidempiä tuotekoodeja, on merkkikenttien määrää lisättävä.



**Kuva 14.** Järjestelmän tietokantamalli

Kuvasta 14 selviää myös julkaisutilojen ja maakohtaisten näkyvyyksien riippuvuudet. Tuotteita ei ole sidottu käyttäjätunnuksiin millään tavalla, joten *Modular\_Roles* ja *Modular\_Users* -taulut ovat tuoteviittausten ulkopuolella. Näin myös taulujen *Modular\_CustomerContracts* ja *Modular\_Configurators* kohdalla, joista jälkimmäinen halutaan pitää erillään vielä sen vuoksi, että alkuperäisessä tuotekonfiguraattorissa ei ole vielä tukea tälle taululle. Oheisessa taulukossa on esitetty tarkemmat kuvaukset tietokantatauluista.

**Taulukko 29.** Tietokantataulut

| <b>Taulun nimi</b>                     | <b>Kuvaus</b>  |
|--|--|
| Modular_MaterialCodes                  | Materiaalikoodilistaus.  |
| Modular_OrderingCodeContent            | Materiaalikoodille määritellyt merkit tuotekoodissa.   |
| Modular_OrderingCodeCombinedChars      | Perättäisten merkkien yhdistäminen kun useampi perättäinen merkki tarkoittaa yhtä komponenttivalintaa. Näillä komponenttivalinnoilla on yhteinen kuvaus. Käytetty tuotekonfiguraattorin käyttöliittymässä. |
| Modular_PlusCode                       | Hinnoittelukoodin muodostamiseen käytetyt säännöt.   |
| Modular_MaterialOrderingBind           | Säännöt, joiden perusteella päätellään materiaalikoodi tuotekoodista.  |
| Modular_OrderingCodeAllowedRule        | Tuotekoodin validointisäännöt.   |
| Modular_Products                       | Tuotelistaus.  |
| Modular_ReleaseState                   | Julkaisutilalistaus.   |
| Modular_ReleaseStateVisibility         | Julkaisutilojen sitominen järjestelmänimiin.   |
| Modular_Systems                        | Järjestelmänimilistaus.  |
| Modular_MaterialCodesCountryVisibility | Materiaalikoodien sitominen maakodeihin. Tietyllä maakoodilla on mahdollisuus tilata vain tiettyjen materiaalikoodien mukaisia tuotteita.  |
| Modular_CountryCodes                   | Maakoodilistaus.   |
| Modular_Configurators                  | Tuotekonfiguraattorilistaus. Järjestelmässä voi olla useita eri tuotekonfiguraattoreita, jolloin on hyvä olla mahdollista määritellä, mitkä materiaalikoodit näkyvät millekin tuotekonfiguraattorille.     |
| Modular_CustomerContracts              | Hintalistojen nimi -listaus. Määrittelee mitkä hintalistojen nimet ovat nä-  |

|               |  |
|---------------|--|
|               | kyvissä ”Hintalistojen export” – näkymässä.  |
| Modular_Roles | Käyttäjien roolit. Esimerkiksi ylläpitäjä, käyttäjien ylläpitäjä ja normaali-käyttäjä. |
| Modular_Users | Järjestelmän käyttäjien listaus.   |

### 6.3. Tuotekoodin validointi

Validointi toteutetaan SQL-funktiolla, jolloin validointiin tehtävät muutokset on nopeampi tehdä ja järjestelmä saadaan pidettyä helpommin hallittavana. SQL-funktioon muutosten tekeminen on nopeaa, koska lähdekoodia ei tarvitse kääntää. Helpompi hallittavuus puolestaan saavutetaan sillä, että muutosten tekemiseen tarvitaan ainoastaan tietokannasta vastaava henkilö. Taulukon 23 esimerkissä on seitsemän eri validointisääntöä. Näistä jokainen sääntö täytyy toteutua, jotta tuotekoodi on validi. Itse validointifunktio suorittaa ennen validointitarkistusta myös muita ehtoja, joiden täytyy toteutua, jotta tuotetta voidaan tilata. Taulukossa 30 on esitetty validointifunktion parametrit.

**Taulukko 30.** Validointifunktion parametrit

| Syötteen nimi     | Tietotyyppi | Esimerkkiarvo |
|-------------------|-------------|---------------|
| Materiaalikoodi   | VARCHAR(50) | OFFICE01A     |
| Tuotekoodi        | VARCHAR(18) | OAABNNABANA   |
| Järjestelmän nimi | VARCHAR(50) | SAP           |
| Maakoodi          | VARCHAR(2)  | FI            |

Ensimmäinen tarkistus on, onko syötetietojen materiaalikoodia ja järjestelmän nimeä tietokannassa. Tämän jälkeen materiaalikoodin ja maakoodin perusteella tarkistetaan, onko materiaalikoodi asetettu nähtäväksi syötteen mukaiselle maakoodille. Seuraavaksi selvitetään tuotteen nimi materiaalikoodin perusteella. Tuote täytyy selvittää, jotta saadaan selville, minkä pituinen tuotekoodi kyseisellä tuotteella tulee olla. Pituuden tarkistuksen jälkeen selvitetään, onko tuote näkyvässä valitulle järjestelmälle.

Seuraavaksi validointifunktio etenee tuotekoodin validointiin. Ensimmäinen ja helpoin validoinnin vaihe on tarkastella, sisältääkö tuotekoodi ainoastaan merkkejä, jotka on määritelty taulukossa 21. Tämän jälkeen tarkistetaan, toteutuvatko validointisäännöt. Taulukossa 31 on esitetty kaikki validointisäännöt auki kerrottuna.

**Taulukko 31.** Validointisäännöt auki kerrottuna

| Sääntö nro. | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | M11 |
|-------------|----|----|----|----|----|----|----|----|----|-----|-----|
| 1           | O  | A  |    |    |    |    |    | A  |    |     | A   |
| 1           | O  | A  |    |    |    |    |    | C  |    |     | A   |
| 1           | O  | A  |    |    |    |    |    | D  |    |     | A   |
| 1           | O  | B  |    |    |    |    |    | A  |    |     | A   |
| 1           | O  | B  |    |    |    |    |    | C  |    |     | A   |
| 1           | O  | B  |    |    |    |    |    | D  |    |     | A   |
| 1           | O  | A  |    |    |    |    |    | B  |    |     | A   |
| 1           | O  | B  |    |    |    |    |    | B  |    |     | A   |
| 1           | O  | C  |    |    |    |    |    | B  |    |     | A   |
| 1           | O  | D  |    |    |    |    |    | B  |    |     | A   |
| 2           | O  |    | A  |    |    |    |    |    |    |     | A   |
| 2           | O  |    | B  |    |    |    |    |    |    |     | A   |
| 2           | O  |    | C  |    |    |    |    |    |    |     | A   |
| 2           | O  |    | D  |    |    |    |    |    |    |     | A   |
| 3           | O  |    |    | A  |    |    |    | A  |    |     | A   |
| 3           | O  |    |    | A  |    |    |    | B  |    |     | A   |
| 3           | O  |    |    | B  |    |    |    | A  |    |     | A   |
| 3           | O  |    |    | B  |    |    |    | B  |    |     | A   |
| 3           | O  |    |    | C  |    |    |    | A  |    |     | A   |
| 3           | O  |    |    | C  |    |    |    | B  |    |     | A   |
| 3           | O  |    |    | D  |    |    |    | C  |    |     | A   |
| 3           | O  |    |    | D  |    |    |    | D  |    |     | A   |
| 3           | O  |    |    | E  |    |    |    | C  |    |     | A   |
| 3           | O  |    |    | E  |    |    |    | D  |    |     | A   |
| 3           | O  |    |    | F  |    |    |    | C  |    |     | A   |
| 3           | O  |    |    | F  |    |    |    | D  |    |     | A   |
| 4           | O  |    |    |    | N  | N  |    |    |    |     | A   |
| 4           | O  |    |    |    | A  | A  |    |    |    |     | A   |
| 4           | O  |    |    |    | A  | B  |    |    |    |     | A   |
| 4           | O  |    |    |    | A  | C  |    |    |    |     | A   |
| 4           | O  |    |    |    | B  | A  |    |    |    |     | A   |
| 4           | O  |    |    |    | B  | B  |    |    |    |     | A   |

|   |   |  |  |  |   |   |   |   |   |   |   |
|---|---|--|--|--|---|---|---|---|---|---|---|
| 4 | O |  |  |  | B | C |   |   |   |   | A |
| 5 | O |  |  |  |   |   | A |   |   |   | A |
| 5 | O |  |  |  |   |   | B |   |   |   | A |
| 5 | O |  |  |  |   |   | C |   |   |   | A |
| 6 | O |  |  |  |   |   |   | D | N |   | A |
| 6 | O |  |  |  |   |   |   | A | A |   | A |
| 6 | O |  |  |  |   |   |   | A | B |   | A |
| 6 | O |  |  |  |   |   |   | B | A |   | A |
| 6 | O |  |  |  |   |   |   | B | B |   | A |
| 6 | O |  |  |  |   |   |   | C | A |   | A |
| 6 | O |  |  |  |   |   |   | C | B |   | A |
| 6 | O |  |  |  |   |   |   | D | A |   | A |
| 6 | O |  |  |  |   |   |   | D | B |   | A |
| 7 | O |  |  |  |   |   |   |   |   | N | A |
| 7 | O |  |  |  |   |   |   |   |   | A | A |
| 7 | O |  |  |  |   |   |   |   |   | B | A |

Taulukosta 31 ilmenee, että tietokantaan tulee lisätä 48 validointisääntöriiviä. Mikäli tuote olisi toteutettu kiinteillä tuotekoodilla, olisi tietokantaan pitänyt lisätä taulukon 27 mukaisesti 16632 validia tuotekoodia. Tässä tulee modulaarisen tuotekoodin etu hyvin esille. Validointi perustuu siihen, että validoitavaa tuotekoodia vasten täytyy taulukosta löytyä yhtä monta vastaavaa riviä kuin tuotteella on validointisääntöjä. Jos tämä ehto toteutuu, tuotekoodi on validi. Jos tarkastelemme esimerkiksi tuotekoodia *OAABN-NABANA*, niin taulukosta 32 saamme taulukon 8 mukaiset vastaavaisuudet.

**Taulukko 32.** Sääntöjen vastaavuudet suhteessa tuotekoodiin

| Sääntö nro.       | M1       | M2       | M3       | M4       | M5       | M6       | M7       | M8       | M9       | M10      | M11      |
|-------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 1                 | O        | A        |          |          |          |          |          | B        |          |          | A        |
| 2                 | O        |          | A        |          |          |          |          |          |          |          | A        |
| 3                 | O        |          |          | B        |          |          |          | B        |          |          | A        |
| 4                 | O        |          |          |          | N        | N        |          |          |          |          | A        |
| 5                 | O        |          |          |          |          |          | A        |          |          |          | A        |
| 6                 | O        |          |          |          |          |          |          | B        | A        |          | A        |
| 7                 | O        |          |          |          |          |          |          |          |          | N        | A        |
| <b>Tuotekoodi</b> | <b>O</b> | <b>A</b> | <b>A</b> | <b>B</b> | <b>N</b> | <b>N</b> | <b>A</b> | <b>B</b> | <b>A</b> | <b>N</b> | <b>A</b> |

Taulukosta 32 näemme, että tuotekoodi on validi, sillä vastaavaisuudet löytyivät seitsemästä eri säännöstä, joka on samalla myös materiaalikoodin sääntöjen kokonaismäärä. Tietokannassa ei välttämättä ole “säännön numero” -kentässä arvoa eikä siihen voi luottaa, sillä kentän arvo voi olla virheellinen jo määrittelyvaiheessa. Tämän vuoksi sääntöjen määrä on varmempaa laskea itse sääntötaulun rivien määrästä. Tämä tapahtuu siten, että haetaan kaikki materiaalikoodia vastaavat rivit. Vaihdetaan rivien merkkien arvot samaksi sovitukseksi merkiksi, esimerkiksi merkiksi X. Tyhjille merkille ei tehdä mitään vaan nämä pysyvät tyhjinä. Erilaisia rivejä haettaessa saamme materiaalikoodin sääntöjen määrän. SQL-kielessä tätä vastaa *SELECT DISTINCT* -määre. Tämä tilanne on esitettynä taulukossa 33.

**Taulukko 33.** Sääntöjen määrän laskeminen

| Sääntö nro. | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | M11 |
|-------------|----|----|----|----|----|----|----|----|----|-----|-----|
| 1           | X  | X  |    |    |    |    |    | X  |    |     | X   |
| 2           | X  |    | X  |    |    |    |    |    |    |     | X   |
| 3           | X  |    |    | X  |    |    |    | X  |    |     | X   |
| 4           | X  |    |    |    | X  | X  |    |    |    |     | X   |
| 5           | X  |    |    |    |    |    | X  |    |    |     | X   |
| 6           | X  |    |    |    |    |    |    | X  | X  |     | X   |
| 7           | X  |    |    |    |    |    |    |    |    | X   | X   |

#### 6.4. .NET moduuli

.NET moduulia, toiselta nimeltä apukirjasto, käytetään rajapintasovelluksessa tietojen hakemiseen sekä ylläpitoliittymässä tuotetietojen testaaminen -näkyvässä. .NET moduuli on käytännössä DLL-tiedosto (Dynamic Link Library), johon viittaamalla saadaan sen tarjoamat metodit käyttöön muissa järjestelmissä (Groh 2002). Apukirjasto on toteutettu C#-ohjelmointikielellä ja sitä varten on tehty palvelinkohtainen asetustiedosto. Asetustiedosto on seuraavan esimerkin näköinen.



```

<configuration>
  <databaseConnectionString>Database=Selectica;User
ID=user_name;Pwd=password;Network Address=192.168.123.123
  </databaseConnectionString>

  <validateQuery>SELECT
dbo.validateOrderingCode (@MaterialCode,@OrderingCode,@SystemName,@
CountryCode)</validateQuery>

  <plusCodeQuery>SELECT
dbo.getPlusCode (@MaterialCode,@OrderingCode)
  </plusCodeQuery>

  <materialCodeQuery>SELECT
dbo.getMaterialCodeFromOrderingCode (@OrderingCode)
  </materialCodeQuery>

  <priceQuery>SELECT
dbo.getComponentTransferPriceMopo (@PriceList,@Items,@Currency,@Cou
ntryCode)
  </priceQuery>

  <systemName>sap</systemName>
</configuration>

```

*databaseConnectionString* on merkkijono, jossa on yhteystiedot tietokantaa varten. *validateQuery* on validoinnin SQL-funktion nimi. Tässä, kuten muissakin SQL-funktioissa on myös määritely funktion parametrit. Ne ovat samat, jotka on määritely käyttötapauksissa luvussa neljä. Toinen SQL-funktio kohdassa *plusCodeQuery* on tarkoitettu hinnoittelukoodin hakemiseen. Viimeinen SQL-funktio on määritely kohdassa *priceQuery*, jonka avulla haetaan tuotteen hinta. Viimeinen asetus *systemName* määrittelee järjestelmän nimen, jota varten apukirjasto on asennettu. Yllä olevassa esimerkissä järjestelmän nimeksi on määritely *sap*, jolla viitataan toiminnanohjausjärjestelmään. Apukirjastossa on asetustiedoston lukemiseen tarkoitettu metodi. Asetustiedoston sijainti palvelinkoneelle on ennalta määrätty. Tämä sen vuoksi, että apukirjasto tulee ensisijaisesti asentaa GAC (Global Assembly Cache) –välimuistiin. Tämän välimuistin tarkoitus on estää DLL-kirjastojen konfliktitilanteita. Välimuistiin ei kuitenkaan voi asentaa muita tiedostoja kuin DLL-kirjastoja, joten asetustiedoston täytyy olla jossain muussa hakemistossa (Microsoft Corporation, 2010).

Apukirjasto kirjoittaa kaikki virhetilanteet Windows Event Log -tietueeseen. Se on Windows-käyttäjärjestelmien sisäinen tapahtumatietue (Microsoft Corporation, 2010). Virhetilanteiden kirjoittaminen tapahtumatietueeseen on tarpeen etenkin silloin, kun apukirjastoa käytetään rajapintasovelluksessa. Tapahtumatietueeseen kirjataan kaikki parametrin arvot, joita käytettiin ajettavan metodin yhteydessä. Liitteessä 1 on esitetty materiaalikoodin hakemiseen liittyvä metodi virheenkäsittelyineen. Tämän lisäksi apukirjastossa on metodit tuotekoodin validointiin ja hinnanhakuun. Kaikki nämä metodit ovat toimintavaltaan samankaltaisia kuin liitteen 1 esimerkki.

## 6.5. Rajapintasovellus

Rajapintasovellus on toteutettu samalla ohjelmistokomponenttikirjastolla ja ohjelmointikielellä kuin .NET moduuli. Rajapintasovellus käyttää Web Service -tekniikkaa. Rajapintasovellukselle lähetettävä SOAP-viesti (Simple Object Access Protocol) on seuraavan XML-skeeman mukainen. SOAP on protokolla XML-pohjaisten viestien vaihtamiseen tietokoneverkossa.

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema xmlns="http://ModularProductInterface.ReceiveXML" targetNamespace="http://ModularProductInterface.ReceiveXML"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Root">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="MaterialCode" type="xs:string" />
        <xs:element name="OrderingCode" type="xs:string" />
        <xs:element name="PriceList" type="xs:string" />
        <xs:element name="Currency" type="xs:string" />
        <xs:element name="CountryCode" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Näiden elementtien sisältämät arvot saadaan toiminnanohjausjärjestelmästä. Materiaalikoodi ei ole pakollinen, sillä materiaalikoodi pystytään päättelemään tuotekoodin perusteella. Hintalistan nimeä ja valuuttaa tarvitaan hinnanlaskennassa. Maakoodin arvoon käyttäjä ei pysty itse vaikuttamaan ja sen avulla tarkistetaan, onko kyseisen maan toiminnanohjausjärjestelmällä oikeutta valittuun tuotteeseen tai hintalistaan. Rajapintasuorituksista saatava SOAP-viesti on seuraavan XML-skeeman mukainen.

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema xmlns="http://ModularProductInterface.ResponseXML" targetNamespace="http://ModularProductInterface.ResponseXML" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Root">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="MaterialCode" type="xs:string" />
        <xs:element minOccurs="0" name="OrderingCode" type="xs:string" />
        <xs:element minOccurs="0" name="PlusCode" type="xs:string" />
        <xs:element minOccurs="0" name="PriceList" type="xs:string" />
        <xs:element minOccurs="0" name="Currency" type="xs:string" />
        <xs:element name="CountryCode" type="xs:string" />
        <xs:element minOccurs="0" name="Price" type="xs:string" />
        <xs:element minOccurs="0" default=" " name="Error" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Tässä viestissä parametrit hintalistan nimi, valuutta ja maakoodi kopioidaan suoraan sisään menevästä viestistä. Ne ovat paluuviestissä lähinnä siksi, että ne voivat antaa lisäarvoa vikatilanteiden selvittelyyn. Verrattuna sisääntuloviestiin, paluuviestissä on lisäksi kolme lisäparametria. *PlusCode* on toinen nimi hinnoittelukoodille, *Price* on tuotteelle laskettu hinta ja *Error*-parametri sisältää mahdollisen virheilmoituksen.

Useimmiten Web Service –rajapintametodi palauttaa yhden parametrin kerrallaan. Tässä rajapintasovelluksessa halutaan kuitenkin saada kaikki tarvittavat tiedot yhdellä viestillä, jolloin toiminnanohjausjärjestelmän päähän tarvitsee toteuttaa vähemmän logiikkaa. Rajapintametodi voi palauttaa useamman parametrin kerralla määrittelemällä se palauttamaan *struct*-tietotyypin eli rakenteen, joka voi sisältää useita parametreja. Rakenne on kevyempi versio *class*-tietotyypistä, eli luokasta. Verrattuna luokkaan, rakenne ei tue käännönaikaista instanssienttien alustamista, rakenne ei voi periytyä muista luokista kuin *System.ValueType* ja muu luokka ei voi periytyä rakenteesta (C# Online.NET 2011). Seuraavassa esimerkissä on luotu rakenne nimeltä *Root*, joka sisältää samat parametrit kuin rajapinnan paluuviesti.

```
public struct Root
{
    public string MaterialCode;
    public string OrderingCode;
    public string PlusCode;
    public string PriceList;
    public string Currency;
    public string CountryCode;
    public string Price;
    public string Error;
}
```

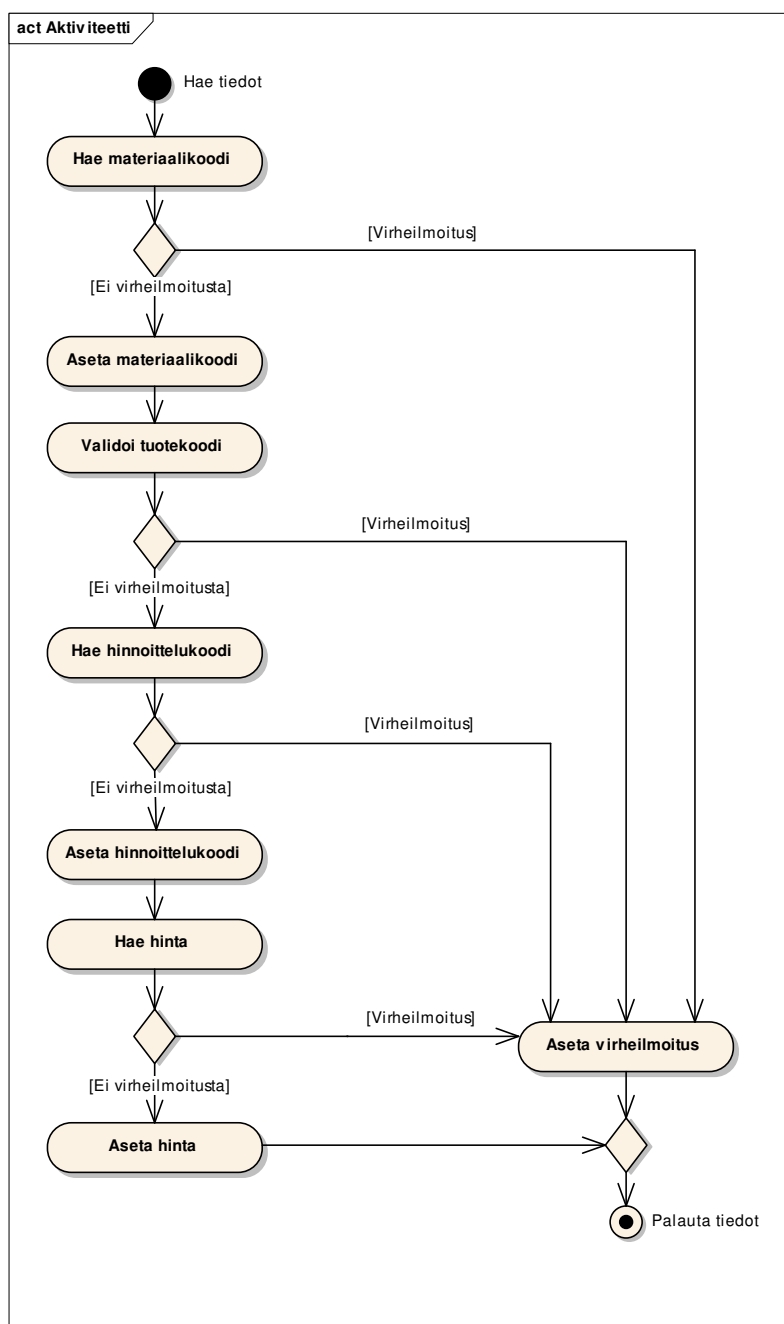
Web Service –metodi palauttaa rakenteen, kun sen alkuun lisätään määre [*SoapRpcMethod(ResponseElementName = "Root", Use = SoapBindingUse.Literal)*]. Tässä *ResponseElementName* määrittelee rakenteen nimen ja *Use = SoapBindingUse.Literal* määrittelee, että paluuviestin skeema noudattaa rakenteeltaan rakenteen *Root* muotoa kirjaimellisesti. Seuraavassa esimerkissä on käytetty *Root*-rakennetta paluuviestin skeemana (Stack Overflow 2010).

```
[SoapRpcMethod(ResponseElementName = "Root", Use = SoapBindingUse.Literal)]
[WebMethod]
public GetProductDataResponse GetProductData(string MaterialCode,
string OrderingCode, string PriceList, string Currency, string
CountryCode)
{
    GetProductDataResponse response = new GetProductDataResponse();

    // Tähän logiikka paluuviestin täyttämiseen
    //response = ...

    return response;
}
```

Rajapintasovellus ajaa .NET moduulin metodeja kuvan 15 aktiviteettikaavion mukaisesti. Toimenpiteitä ajetaan sarjamaisesti ja jokainen toimenpide on riippuvainen edellisen toimenpiteen syötteestä. Mikäli jokin toimenpide aiheuttaa virheilmoituksen, keskeytetään toimenpiteiden ajaminen, asetetaan virheilmoitus ja palautetaan paluuviesti. Toimenpiteet on kuvattu tarkemmin luvussa neljä.



**Kuva 15.** Tietojen haun aktiviteettikaavio

Aktiviteettikaavion virheilmoitus tulee SQL-funktioiden tasolta asti eikä virheilmoituksella tarkoiteta poikkeusta lähdekoodin suorittamisessa. Nämä virheilmoitukset ovat siis tarkoituksenmukaisia ja sellaisia, jotka antavat käyttäjälle tietoa syöteparametrien aiheuttamista ongelmista. Virheilmoitukset on kuvattu tarkemmin taulukossa 34.

**Taulukko 34.** Rajapintasovelluksen palauttamat virheilmoitukset

| <b>Virheilmoitus</b>   | <b>Kuvaus</b>   |
|--|---|
| Material code not found.   | Materiaalikoodia ei löytynyt.   |
| Cannot resolve product name from material code.                            | Materiaalikoodin perusteella ei löydetty tuotteen nimeä.  |
| Ordering code length check failure.  | Tuotekoodin pituus ei vastannut sitä pituutta, joka on määritelty tuotteelle.                       |
| System name not found (<system name>).                                     | Järjestelmän nimeä ei löytynyt.   |
| Material code not visible for the system (<system name>).                  | Materiaalikoodi ei ole määritelty näkyväksi valitulle järjestelmän nimelle.                         |
| Ordering code contains forbidden character(s).                             | Tilauskoodi sisältää merkkejä, joita ei ole määritelty kyseiselle materiaalikoodille.               |
| Invalid ordering code.   | Tuotekoodi ei ole validi, toisin sanoen jokaista validointisääntöä kohden ei löytynyt vastaavuutta. |
| Material code <material code> not visible for country code <country code>. | Materiaalikoodi ei ole asetettu näkyväksi valitulle maakoodille.                                    |
| No products found for ordering code length (<number>).                     | Tuotekoodin pituuttava vastaavia tuotteita ei löytynyt.   |
| Unable to resolve material code from ordering code.                        | Tuotekoodin perusteella ei pystytty selvittämään materiaalikoodia.                                  |
| Price list <price list> not visible for country code <country code>.       | Hintalista ei ole määritetty näkyväksi valitulle maakodille.  |

## 6.6. Ylläpitoliittymä

Ylläpitoliittymä ei aseta suuria vaatimuksia käyttöliittymäkomponenttien suhteen. Ylläpitoliittymän tulee olla helposti käytettävissä ja käyttöönottokynnys tulee olla minimaalinen. Nämä vaatimukset puoltavat web-käyttöliittymää, sillä tällöin käyttäjän koneelle ei tarvitse asentaa erillistä asiakasohjelmaa. Riittää, että käyttäjän tietokone on liitetty verkkoon ja koneelta löytyy web-selainohjelma.

### 6.6.1. Käytetyt tekniikat

Ylläpitoliittymän tulee olla web-sovellus. Tätä varten on useita eri alustoja, kuten Java EE, PHP, Ruby On Rails ja ASP.NET. Nämä ovat kaikki yleisesti käytössä olevia alustaratkaisuja, mutta koska ylläpitoliittymässä käytetään .NET moduulia, on ASP.NET luonteva ratkaisu, sillä myös se perustuu Microsoft .NET –arkkitehtuuriin. Lisäksi Microsoft .NET –arkkitehtuuriin on sisäänrakennettu natiivi tuki Microsoft SQL –tietokannoille ja järjestelmän toteuttamiseen osoitetuille ohjelmoijille oli yllä mainittu arkkitehtuuri tuttu jo entuudestaan. Lisäksi asiakkaalla on valmiina Microsoft Windows –pohjaisia palvelimia, joille ylläpitoliittymän voi asentaa. Ohjelmointikieleksi valittiin C#, sillä ohjelmoijilla oli siitä aiempaa kokemusta ja he kokivat sen helppotajuisimmaksi. Ylläpitoliittymä käyttää Microsoft IIS (Internet Information Services) –web-palvelinohjelmistoa.

Käyttäjän autentikointi toteutetaan Windows Authentication –menetelmällä. Tällöin riittää, että käyttäjä on kirjautunut sisään Windows-käyttöjärjestelmään eikä ylläpitoliittymässä tarvita erillistä sisäänkirjautumisnäkyä. Käyttäjän tulee olla kirjautunut asiakkaan Windows Server Domain –toimialueeseen. Jos käyttäjä kirjautuu ylläpitoliittymään toisesta kuin asiakkaan verkosta, näyttää www-palvelinkone sisäänkirjautumislomakkeen (Microsoft Corporation 2010). Tähän lomakkeeseen tulee käyttäjän kirjoittaa toimialueen nimi, käyttäjätunnus ja salasana. Näin ylläpitoliittymään voi kirjautua myös muista verkoista, joista on pääsy asiakkaan verkkoon. Toimialueen nimi ja käyttäjätunnus kirjoitetaan lomakkeeseen käyttäjän nimi –kenttään muodossa *DOMAIN\KÄYTTÄJÄTUNNUS*.



### 6.6.2. Tuotetietojen siirtotiedosto

Tuotetiedot viedään ylläpitoliittymän kautta tietokantaan XML-pohjaisen siirtotiedoston avulla. Siirtotiedostossa on kaikki tiedot, jotka on määritetty luvussa viisi. Siirtotiedoston XML-skeema on esitetty XSD-tiedostona (XML Schema Definition) liitteessä 2. XSD:n voi mieltää XML-skeeman sanastoksi (Salonen 2006). *ROOT*-elementti on nimensä mukaisesti juurielementti. Tämän alla on otsikkotason elementit, kuten *MaterialCode* ja *Product*, joka määrittelee mihin tuotteeseen materiaalikoodi kuuluu.

*AllowedChars*-elementissä on määritelty tuotekoodien merkkien sisältö. *Item*-elementti vastaa yhtä riviä ja saman elementin *CharNo*-attribuutti kertoo merkin indeksin. *Desc*-attribuutti puolestaan kertoo merkkiä vastaavan kuvauksen. *Item*-elementin arvo kertoo sallitun merkin. Seuraavaksi on esitetty esimerkki *Item*-elementistä.

```
<Item CharNo="5,6" Desc="Alumiininen 28 dB">AB</Item>
```

*AllowedChars*-elementin jälkeen tulee *Rules*-elementti. Tämä koostuu kahdesta eri elementistä, jotka ovat *MaterialCodeRule* ja *ValidationRules*. *MaterialCodeRule*-elementissä on säännöt materiaalikoodin tunnistamiseksi tuotekoodista. Se sisältää *Item*-elementtejä samaan tapaan kuin *AllowedChars*-elementti. Seuraavaksi on esitetty esimerkki *MaterialCodeRule*-elementistä.

```
<MaterialCodeRule>
  <OrderingCodeChars>
    <Item CharNo="1">O</Item>
    <Item CharNo="11">A</Item>
  </OrderingCodeChars>
</MaterialCodeRule>
```

*ValidationRule*-elementti sisältää validointisäännöt. Se on jaettu useisiin *RuleLine*-elementteihin, jotka vastaavat sääntöriivejä. Sillä on kaksi attribuuttia *VisibleInSelectica* ja *VisibleInSAP*, jotka määrittelevät erikseen, mihin järjestelmiin sääntöriivi on näkyvisissä. *ValidationRule*-elementti sisältää niin ikään *Item*-elementtejä kuten aiemmissa esimerkeissä. Seuraavaksi on esitetty esimerkki *ValidationRule*-elementistä.

```

<ValidationRule>
  <RuleLine VisibleInSelectica="True" VisibleInSAP="True">
    <Item CharNo="1">O</Item>
    <Item CharNo="2">A, B</Item>
    <Item CharNo="8">A, C, D</Item>
    <Item CharNo="11">A</Item>
  </RuleLine>
  <RuleLine VisibleInSelectica="True" VisibleInSAP="True">
    <Item CharNo="1">O</Item>
    <Item CharNo="2">A, B, C, D</Item>
    <Item CharNo="8">B</Item>
    <Item CharNo="11">A</Item>
  </RuleLine>
</ValidationRule>

```

*Rules*-elementin jälkeen tulee siirtotiedoston viimeinen osuus *PricingCodes*. Se määrittelee säännöt hinnoittelukoodin muodostamiseen ja koostuu *PricingCode*-elementeistä. Tällä elementillä attribuutti *Value* kertoo hinnoittelukoodin osan arvon. *Desc*-attribuutti kertoo koodin kuvauksen ja attribuutit *SummaryGroupText*, *GroupOrderNumber* ja *GroupTextOrderNumber* ovat tuotekonfiguraattorispesifisiä. *PricingCode*-elementin sisällä on *Item*-elementtejä. Seuraavaksi on esitetty esimerkki *PricingCode*-elementistä.

```

<PricingCode Value="CSE01A" Desc="Miditorni ATX" Summary-
GroupText="" GroupOrderNumber="" GroupTextOrderNumber="">
  <Item CharNo="1">O</Item>
  <Item CharNo="2">A</Item>
  <Item CharNo="11">A</Item>
</PricingCode>

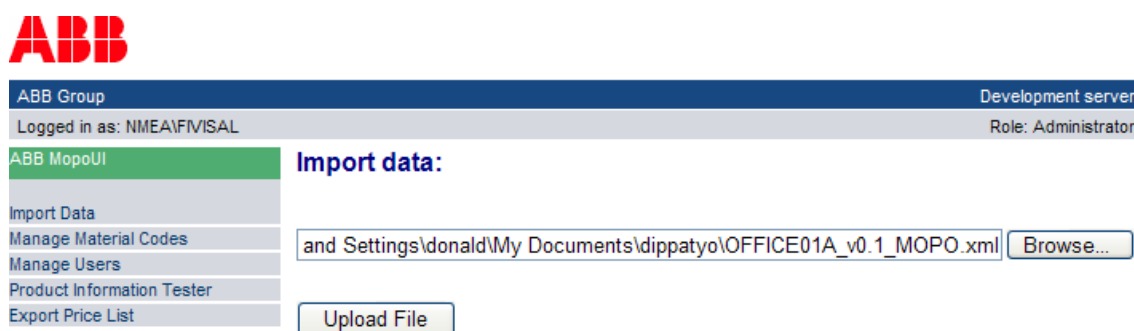
```

### 6.6.3. Käyttöliittymä

Käyttöliittymän sivupohjan sijoittelu ja ulkonäkö perustuvat asiakkaan valmiiseen sivupohjaan, sillä asiakkaan kaikkien sivustojen tyyli tulee olla yhtenevä. Sivupohjan yläosassa on asiakkaan logo, jonka alla ovat sivuston yläosan rajaavat siniset ja harmaat koko sivun levyiset solut. Näihin soluihin tulostetaan tiedot palvelimen nimestä, käyttäjätunnuksesta ja käyttäjäroolista. Käyttöliittymän vasemmassa laidassa on valikko, josta navigoidaan päänäkymien välillä. Tämän valikon oikealla puolella on sisältöalue. Ilmoi-

tukset operaatioiden onnistumisesta näytetään sisältöalueen vasemmassa ylälaudassa. Seuraavissa kuvissa on esitetty käyttöliittymän keskeisimmät näkymät.

Kuvassa 16 on esitetty siirtotiedoston lähettämiseen liittyvä näkymä. Siinä on tekstikenttä, johon kirjoitetaan siirtotiedoston sijainti käyttäjän koneella tai vaihtoehtoisesti käyttäjä voi painaa ”Browse...” –painiketta, jolloin tiedoston voi hakea käyttäjän tietokoneelta kansionäkymästä. Kun tiedosto on löydetty, käyttäjän painaessa ”Upload File” –painiketta lähetetään siirtotiedosto.



**Kuva 16.** Lähetä siirtotiedosto -näkö

Siirtotiedoston lähettämisen jälkeen näytetään käyttäjälle tieto siirtotiedoston lähettämisen onnistumisesta. Kuvassa 17 on tilanne, jossa siirtotiedoston sisältämän materiaalikoodin tiedot olivat jo tietokannassa. Lisäksi käyttäjä voi valita, mitkä tiedot siirtotiedostosta tuodaan tietokantaan. Tietoja ei ole pakko lisätä tietokantaan, vaan ne voidaan myös vain tulostaa ruudulle. Tällä tavoin siirtotiedostoa voi testata, ellei ole varma, haluaako tietoja tietokantaan asti. Painamalla painiketta ”Confirm import” suoritetaan valitut toimenpiteet tai vaihtoehtoisesti käyttäjä voi painaa painiketta ”Cancel”, joka peruuttaa aiotut toimenpiteet.



|                            |   |
|----------------------------|---|
| ABB Group                  | Development server  |
| Logged in as: NMEA\FMVISAL | Role: Administrator   |
| ABB MopoUI                 | Material Code OFFICE01A already exists.<br>Product OFFICE01 already exists. No changes will be made in table "Modular_Products".<br>All other tables will be updated (old values will be deleted and new values will be imported)<br>Please select what elements should be imported:  |
| Import Data                |   |
| Manage Material Codes      |   |
| Manage Users               |   |
| Product Information Tester |   |
| Export Price List          |   |
|                            | <input checked="" type="checkbox"/> Import Ordering Code Content and Combined Chars<br><input checked="" type="checkbox"/> Import Plus Code<br><input checked="" type="checkbox"/> Import Material Ordering Bind<br><input checked="" type="checkbox"/> Import Material Ordering Code Allowed Rules<br><input checked="" type="checkbox"/> Submit changes to database. Uncheck this box if you only want to test what would be written to database. |
|                            | <input type="button" value="Confirm Import"/> <input type="button" value="Cancel"/>   |

**Kuva 17.** Lähetä siirtotiedosto –näkyvä siirtotiedoston lähettämisen jälkeen

”Materiaalikoodien hallinta” –näkyvä on keskeisin näkyvä ylläpitoliittymässä. Sisäänkirjautumisen jälkeen se onkin vakionäkyvä, joka on esitetty kuvassa 18. Näkyvä koostuu taulukosta, jossa ensimmäisessä sarakkeessa on materiaalikoodin nimi. Sitä painamalla näkee samat materiaalikoodiin liittyvät tiedot kuin on mainittu luvussa viisi, eli tuotekoodin merkkien sisällön, säännöt materiaalikoodin tunnistamisesta tuotekoodista, validointisäännöt sekä hinnoittelukoodin muodostamiseen liittyvät säännöt. Toisessa sarakkeessa on kerrottu tuote, johon materiaalikoodi kuuluu. Kolmannessa sarakkeessa on alavetovalikko, josta voi valita materiaalikoodin järjestelmäkohtaisen julkaisutilan. Uusi julkaisutila syötetään tietokantaan ”Submit ReleaseState” –painikkeella. Neljännessä sarakkeessa on linkki, jota painamalla pääsee ”materiaalikoodin maakohdainen näkyvyys” –näkyvään. Viimeisessä sarakkeessa on ”Delete”-linkki, jota painamalla materiaalikoodiin liittyvät tiedot poistetaan tietokannasta.



ABB Group Development server  
 Logged in as: NMEA\FVISAL Role: Administrator

ABB MopoUI **Manage material codes:**

- Import Data
- Manage Material Codes
- Manage Users
- Product Information Tester
- Export Price List

| MaterialCode              | Product  | Release State                 | Country Visibility | Delete |
|---------------------------|----------|-------------------------------|--------------------|--------|
| <a href="#">OFFICE01A</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01B</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01C</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01D</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01E</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01F</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01G</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01H</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01I</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01J</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01K</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01L</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01M</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01N</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01O</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01P</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01Q</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01R</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01S</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01T</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01U</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01V</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01W</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01X</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01Y</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |
| <a href="#">OFFICE01Z</a> | OFFICE01 | Released in SAP and Selectica | ▼ Edit             | Delete |

**Kuva 18.** Materiaalikoodien hallinta -näkyvä

”Materiaalikoodin maakohtainen näkyvyys” –näkyvä on esitetty kuvassa 19. Sivun alaosassa on kohdassa ”Add country visibility” alavetovalikko, josta voi valita maakoodin, jolle materiaalikoodi asetetaan näkyväksi. ”Add”-painikkeella syötetään arvo tietokantaan. Näkymän yläosassa on lista maakoodeista, joille tuote on määritelty näkyväksi. Näille maakoodeille tulee vielä erikseen määritellä järjestelmäkohtainen julkaisutila, joka toimii samaan tapaan kuin ”materiaalikoodien hallinta” –näkyvässä. Jos siis haluaa, että tuote on julkaistu esimerkiksi Kiinassa, ei riitä että materiaalikoodille määrittelee maakoodin *CN*. Tämän lisäksi tulee materiaalikoodi määritellä maakoodilla *CN* järjestelmäkohtaiseen julkaisutilaan, jolla materiaalikoodi näkyy kyseisessä järjestelmässä. Maakoodimäärittelyn voi myös poistaa painamalla ”Delete”-linkkiä.



|  |  |  |                                    |
|--|--|--|------------------------------------|
| ABB Group  |  | Development server                       |                                    |
| Logged in as: NMEA\FMVISAL                                   |  | Role: Administrator                      |                                    |
| ABB MopoUI   |  | <b>Material Code Country Visibility:</b> |                                    |
| Import Data  | MaterialCode                                       | Country                                  | Release State                      |
| Manage Material Codes  | OFFICE01A  | CN                                       | Not released                       |
| Manage Users   |  |  | ▼ Delete                           |
| Product Information Tester                                   | <input type="button" value="Submit ReleaseState"/> |  |                                    |
| Export Price List  |  |  |                                    |
| <b>Add country visibility:</b>                               |  |  |                                    |
| MaterialCode   | Country  | Add CountryVisibility                    |                                    |
| OFFICE01A  | IN   | ▼  | <input type="button" value="Add"/> |
| <input type="button" value="Back to Manage Material Codes"/> |  |  |                                    |

### Kuva 19. Materiaalikoodin maakohtainen näkyvyys -näkyvä

”Käyttäjien hallinta” –näkyvässä pystyy lisäämään ja poistamaan käyttäjiä, sekä vaihtamaan näiden käyttäjäroolin. Näkyvän yläosa koostuu taulukosta, jossa ensimmäisessä sarakkeessa on verkon toimialueen ja käyttäjätunnuksen yhdistelmä. Toisessa sarakkeessa on käyttäjän etu- ja kolmannessa sarakkeessa sukunimi. Neljännessä sarakkeessa on alasvetovalikko, josta voi valita käyttäjäroolin. Se syötetään tietokantaan ”Submit Roles” –painikkeella. Viimeisessä sarakkeessa on linkki, jota painamalla käyttäjä poistetaan tietokannasta. Näkyvän alaosassa on kontrollit, joilla luodaan uusi käyttäjä. Näkyvä on esitetty kuvassa 20.



ABB Group Development server

Logged in as: NMEA\FIVISAL Role: Administrator

ABB MopoUI

**Manage users:**

| UserName      | FirstName | LastName | Role                  | Delete                 |
|---------------|-----------|----------|-----------------------|------------------------|
| NMEA\ORAVAIN  | Jari      | Oravain  | ReleaseStateWriteable | <a href="#">Delete</a> |
| NMEA\FIVISAL  | Juha P    | Kujala   | ReadOnly              | <a href="#">Delete</a> |
| NMEA\KORHONEN | Urho      | Korhonen | ReadOnly              | <a href="#">Delete</a> |
| NMEA\KARJUNEN | Pasi      | Karjunen | UserAdmin             | <a href="#">Delete</a> |
| NMEA\FIVISAL  | Ville     | Salomäki | Administrator         | <a href="#">Delete</a> |
| NMEA\KORHONEN | Juha      | Korhonen | Administrator         | <a href="#">Delete</a> |
| NMEA\KARJUNEN | Jari      | Karjunen | ReadOnly              | <a href="#">Delete</a> |
| NMEA\FIVISAL  | Antti     | Fivisal  | Administrator         | <a href="#">Delete</a> |

**Add user:**

| UserName             | FirstName            | LastName             | Role | Add User                           |
|----------------------|----------------------|----------------------|------|------------------------------------|
| <input type="text"/> | <input type="text"/> | <input type="text"/> | User | <input type="button" value="Add"/> |

**Kuva 20.** Käyttäjien hallinta -näkökulma

”Tuotetietojen testaaminen” –näkökulmassa käyttäjä pystyy testaamaan tietokantaan lisättyjä tuotetietoja. Se on esitetty kuvassa 21 ja näkökulmalla voi toisin sanoen simuloida, miten tiedot palautuisivat rajapinnan yli toiminnanohjausjärjestelmään. Painikkeella ”Fill sample data to inputs” asetetaan syötekenttiin testiparametrit. Ne on ennalta valittu sellaisiksi, joilla palautuu validin tuotekoodin testitulokset, hinnoittelukoodi ja hinta. Näin käyttäjä pystyy nopeasti testaamaan esimerkiksi sen, että validointifunktio on käytettävissä. Tekstikenttiin syötetään tuotekoodi, hintalistan nimi, valuutta ja maakoodi. Painamalla painiketta ”Validate” validoidaan tuotekoodi ja haetaan hinnoittelukoodi. Hinnan haku on oman painikkeen ”Get price” takana. Näin siksi, että hinnan haku on vaiheista raskain ja sen suorittamiseen vaaditaan eniten aikaa. Usein käyttäjä ei ole kiinnostunut hinnasta, vaan haluaa vain testata esimerkiksi validointisääntöjä.



| ABB Group  |  | Development server                 |   |
|--|--|------------------------------------|---|
| Logged in as: NMEA\FMVISAL   |  | Role: Administrator                |   |
| ABB MopoUI   |  | <b>Product Information tester:</b> |   |
| Import Data  |  | Ordering code                      | OAABNNABANA                                   |
| Manage Material Codes  |  | Pricelist                          | AU-AUD-1                                      |
| Manage Users   |  | Currency                           | AUD   |
| Product Information Tester   |  | Country code                       | FI  |
| Export Price List  |  | Material code                      | OFFICE01A                                     |
|  |  | Validation result                  | Valid <input type="button" value="Validate"/> |
|  |  | Plus code                          | OFFICE01A+CSE01A+PWR01A+CPU02A+RAM01A+MBO02A+ |
|  |  | Price                              | 530 <input type="button" value="Get price"/>  |
| <input type="button" value="Fill sample data to inputs"/>              |  |                                    |   |
| <div style="border: 1px solid gray; height: 80px; width: 100%;"></div> |  |                                    |   |
| <input type="button" value="Test elements"/>                           |  |                                    |   |

### Kuva 21. Tuotetietojen testaaminen -näkyvä

Saman näkymän alaosassa on tuotetietojen massatestaamiseen tarkoitettut kontrollit, jotka ovat paremmin esitetty kuvassa 22. Suureen tekstikenttään kirjoitetaan testattavat tuotekoodit rivinvaihtomerkillä erotettuna ja ne massatestataan ”Test elements” –painikkeella. Tämän jälkeen näkymän alaosaan ilmestyy taulukko, jossa on kerrottu testitulokset. Ensimmäisessä sarakkeessa on tuotekoodi, toisessa validointitieto ja kolmannessa hinnoittelukoodi. Kuvan esimerkissä kaksi ensimmäistä tuotekoodia ovat valideja. Ne eroavat vain siten, että ylempässä on toiseksi viimeinen merkki kirjain *N* ja alemmassa tuotekoodissa se on *A*. Merkki *N* vastaa tyhjää valintaa, mutta merkki *A* vastaa hinnoittelukoodia *DVD01A*. Tämä hinnoittelukoodin osa on lisätty alempaan hinnoittelukoodiin. Kolmannen rivin tuotekoodi ei ole validi, koska siinä toiseksi viimeinen merkki on kirjain *X*, jota ei ole määritelty kyseiselle tuotteelle.



|             |
|-------------|
| 0AABNNABANA |
| 0AABNNABAAA |
| 0AABNNABAXA |

Test elements

| OrderingCode | ValidationResult                                      | PlusCode  |
|--------------|---|---|
| 0AABNNABANA  | Valid   | OFFICE01A+CSE01A+PWR01A+CPU02A+RAM01A+MBO02A+GPU01A+SFW01A+SFW02A+FAN01A        |
| 0AABNNABAAA  | Valid   | OFFICE01A+CSE01A+PWR01A+CPU02A+RAM01A+MBO02A+GPU01A+DVD01A+SFW01A+SFW02A+FAN01A |
| 0AABNNABAXA  | ERROR: Ordering code contains forbidden character(s). | OFFICE01A+CSE01A+PWR01A+CPU02A+RAM01A+MBO02A+GPU01A+SFW01A+SFW02A+FAN01A        |

**Kuva 22.** Tuotetietojen massatestaamiseen liittyvät kontrollit

”Hintalistojen exportoiminen” –näkyessä käyttäjällä on kaksi vaihtoehtoa. Joko hän voi tulostaa yhden hintalistan sisällön kerrallaan suoraan ruudulle, tai sitten useamman hintalistan sisällön kerralla CSV-tiedostoon. Ensimmäinen vaihtoehto on esitetty kuvassa 23 ja siinä käyttäjä on valinnut valintalistasta hintalistan *AE-USD-1*, jonka jälkeen painikkeella ”Print to Screen” hintalistan sisältö on tulostunut ruudulle. Hintalista tulostuu taulukkomaisesti sivun alaosaan ja siinä on neljä saraketta; hintalistan nimi, tuotekoodi, tuotteen kuvaus ja hinta.



ABB Group Development server  
 Logged in as: NMEA\FMVISAL Role: Administrator

ABB MopoUI Command succeeded!

Import Data  
 Manage Material Codes  
 Manage Users  
 Product Information Tester  
 Export Price List

### Export price list:

AE-USD-1  
 AE-USD-16  
 AM-EUR-E01  
 AR-EUR-12  
 AR-USD-1  
 AT-EUR-1  
 AU-AUD-1  
 AU-AUD-17  
 AU-AUD-2  
 AU-EUR-16  
 AZ-EUR-1  
 BA-EUR-1

Instructions to export price list to CSV

Select contracts from the listbox. Click 'Export to CSV' button and after that click 'Download CSV File' button. You can download the last export at anytime as long as the browser is not closed.

[Print to Screen](#)

[Export to CSV](#)

[Download CSV file](#)

[Manage Customer Contracts](#)

| PriceListName | ItemName     | ItemDescription             | Price |
|---------------|--------------|-----------------------------|-------|
| AE-USD-1      | 100L2804H000 | 800 series accessories      | 5     |
| AE-USD-1      | 100L4002P000 | 80_100 Mounting Accessories | 40    |
| AE-USD-1      | 100L4002R000 | 80_100 Mounting Accessories | 11    |
| AE-USD-1      | 100L4004R000 | 80_100 Mounting Accessories | 20    |
| AE-USD-1      | 100L4004T000 | 80_100 Mounting Accessories | 21    |
| AE-USD-1      | 100L4006R000 | 80_100 Mounting Accessories | 20    |
| AE-USD-1      | 100L4007R000 | 80_100 Mounting Accessories | 11    |
| AE-USD-1      | 100L4007S000 | 80_100 Mounting Accessories | 40    |
| AE-USD-1      | 100L4007T000 | 80_100 Mounting Accessories | 21    |
| AE-USD-1      | 100L4007U000 | 80_100 Mounting Accessories | 21    |
| AE-USD-1      | 100L4007V000 | 80_100 Mounting Accessories | 21    |
| AE-USD-1      | 100L4007W000 | 80_100 Mounting Accessories | 21    |
| AE-USD-1      | 100L4007X000 | 80_100 Mounting Accessories | 21    |
| AE-USD-1      | 100L4007Y000 | 80_100 Mounting Accessories | 21    |
| AE-USD-1      | 100L4007Z000 | 80_100 Mounting Accessories | 21    |
| AE-USD-1      | 100L4008R000 | 80_100 Mounting Accessories | 72    |
| AE-USD-1      | 100L4008T000 | 80_100 Mounting Accessories | 120   |
| AE-USD-1      | 800C000-2    | 800 IPGates parts           | 21    |
| AE-USD-1      | 800C000      | 80_ mounting accessories    | 71    |
| AE-USD-1      | 800C000      | 80_ mounting accessories    | 20    |
| AE-USD-1      | 800C000      | 80_ mounting accessories    | 79    |
| AE-USD-1      | 800C000      | 80_ mounting accessories    | 71    |

### Kuva 23. Hintalistan tulostaminen ruudulle

CSV-tiedostoon vietäessä käyttäjä voi valita useamman hintalistan nimen kerralla valintalistasta. Tämä on esitetty kuvassa 24, jossa käyttäjä on valinnut vietäviksi hintalistojen nimet *AE-USD-1*, *AR-EUR-12*, *AU-AUD-1* ja *AU-AUD-17*. Hintalistojen nimien valinnan jälkeen käyttäjä painaa painiketta ”Export to CSV”. Tämän jälkeen hintalistojen sisältö haetaan tietokannasta ja kontrollit menevät *disabled*-tilaan, joka tarkoittaa että niitä ei voi käyttää. Kontrollien alapuolella myös kerrotaan käyttäjälle, mitä nyt on tapahtumassa.



ABB Group Development server

Logged in as: NMEA\FMVISAL Role: Administrator

ABB MopoUI Command succeeded!

Import Data

Manage Material Codes

Manage Users

Product Information Tester

Export Price List

### Export price list:

- AE-USD-1
- AE-USD-16
- AM-EUR-E01
- AR-EUR-12
- AR-USD-1
- AT-EUR-1
- AU-AUD-1
- AU-AUD-17
- AU-AUD-2
- AU-EUR-16
- AZ-EUR-1
- BA-EUR-1

Instructions to export price list to CSV

Select contracts from the listbox. Click 'Export to CSV' button and after that click 'Download CSV File' button. You can download the last export at anytime as long as the browser is not closed.

Print to Screen

**Export to CSV**

Download CSV file

**Processing ...**

**Getting information from database. This process may take up to ten minutes. Please be patient.**

#### **Kuva 24.** Hintalistojen exportoiminen CSV-tiedostoon, alkuvaihe

Kun hintalistojen sisältö on haettu tietokannasta, kontrollit menevät takaisin *enabled*-tilaan, joka tarkoittaa että ne ovat taas käytettävissä. Tämän jälkeen CSV-tiedosto on ladattavissa web-palvelimelta painamalla ”Download CSV file” –painiketta. Tiedoston voi ladata koska tahansa saman käyttäjäistunnon aikana. Jos käyttäjä painaa uudestaan saman istunnon aikana ”Export to CSV” –painiketta, korvautuu vanha tiedosto uudella. Tämä on esitetty kuvassa 25.



ABB Group Development server

Logged in as: NMEA\FVISAL Role: Administrator

ABB MopoUI Data created successfully! Click 'Download CSV File' to download the data.

Import Data

Manage Material Codes

Manage Users

Product Information Tester

Export Price List

### Export price list:

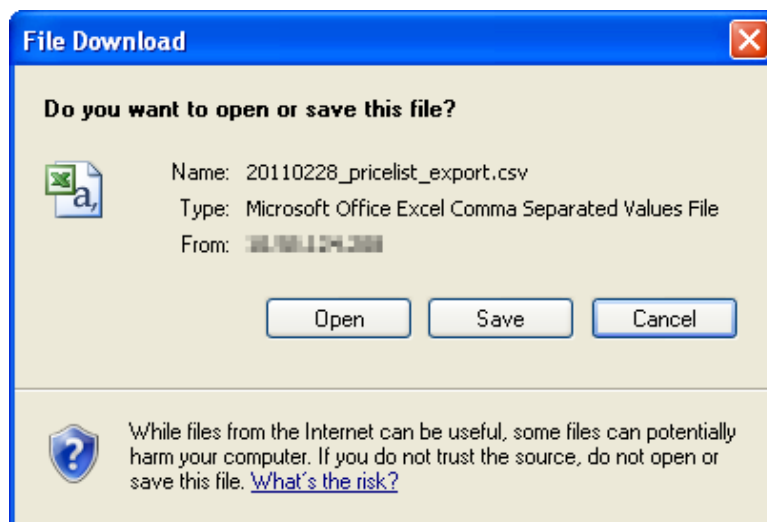
- AE-USD-1
- AE-USD-16
- AM-EUR-E01
- AR-EUR-12
- AR-USD-1
- AT-EUR-1
- AU-AUD-1
- AU-AUD-17
- AU-AUD-2
- AU-EUR-16
- AZ-EUR-1
- BA-EUR-1

Instructions to export price list to CSV

Select contracts from the listbox. Click 'Export to CSV' button and after that click 'Download CSV File' button. You can download the last export at anytime as long as the browser is not closed.

**Kuva 25.** Hintalistojen exportointi CSV-tiedostoon jälkivaihe

”Download CSV file” –painikkeen painamisen jälkeen selain näyttää tiedoston latausikkunan. Tiedoston nimi on muodossa *PÄIVÄMÄÄRÄ\_pricelist\_export.csv*. Kuvan 26 esimerkissä hintalistojen exportointi on suoritettu 28.2.2011.



**Kuva 26.** CSV-tiedoston lataaminen

## 7. JOHTOPÄÄTÖKSET

Tutkielmassa toteutettiin tuotekoodin validointijärjestelmä, joka tukee modulaarisia tuotekoodeja. Tutkielma on enemmän käytäntö- kuin teoriapainotteinen, johtuen siitä että aihetta tukevaa kirjallisuutta on julkaistu vähänlaisesti. Teoriaan liittyen tehtiin havainto, että tuotekoodin varianttien määrä pystytään laskemaan tuloperiaatteen avulla. Lisäksi käytiin läpi tuotekoodin määrittelyyn liittyvää kirjallisuutta.

Aiemmassa järjestelmässä yksi suurimmista ongelmista oli, että tietojen vieminen järjestelmien välillä oli pääosin manuaalista. Uudessa järjestelmässä tietojen tulikin liikkua järjestelmien välillä automaattisesti. Näin inhimillisten virheiden todennäköisyys tippui olennaisesti ja samalla prosessista tuli nopeampi. Tällöin kuitenkin prosessin vieminen alusta loppuun on riippuvainen siitä, että kaikki ketjuun liittyvät järjestelmät ovat tavoitettavissa. Toinen uuden järjestelmän vaatimus oli, että tuotetiedot näkyisivät kaikkiin järjestelmiin samalla tavalla ja tuotetietojen hallinta olisi vain yhdessä paikassa. Tämä ratkaistiin siten, että tiedot laitettiin tietokantaan niin, että tietomalli tukee uusia vaatimuksia ja suunniteltiin rajapinta, jonka yli tiedot viedään eri järjestelmiin.

Suurimmat ongelmat nykyisessä järjestelmässä liittyvät niihin prosessin osa-alueisiin, joissa käyttäjä tekee manuaalisesti työtä. Vaikka tieto siirtyykin järjestelmien välillä automaattisesti, tehdään tuotteen määrittely edelleen käsityönä käyttäen taulukkolaskentaohjelmaa. Tätä prosessia on kuitenkin mahdotonta automatisoida, sillä kone ei voi kertoa millainen tuotteen tulee olla. Tuotteen määrittelyssä eniten hämmennystä aiheuttaa validointisääntöjen määrittely. Niiden tulee olla määritelty juuri oikealla tavalla, koska muuten ne eivät toimi. Tätä voisi parantaa ainakin kolmella tavalla. Ensinnäkin, dokumentointia tulisi tarkentaa ja tuotteiden määrittelijöille tulisi järjestää aiheesta koulutus. Toiseksi, taulukkolaskentapohja tulee olla sellainen, että se ei salli määritellä validointisääntöjä kuin oikealla tavalla. Tässä voi kuitenkin tulla taulukkolaskentaohjelman omat tekniset rajoitteet vastaan. Kolmanneksi, ylläpitoliittymässä voisi olla parempi tarkistus validointisääntöjen oikeellisuudelle.

Järjestelmän käytön aikana törmättiin myös ongelmaan, että kun materiaalikoodin mukaiset tiedot päivitettiin järjestelmään vanhojen tietojen päälle ja uusissa tiedoissa olikin jotain vikaa, niin vanhojen tietojen palauttaminen onnistui vain palauttamalla varmuuskopio koko tietokannasta. Tätä voisi kehittää esimerkiksi siten, että siirtotiedostoa lähetettäessä tallennettaisiin tietokantaan itse siirtotiedosto sellaisenaan omaan tauluunsa. Samalla tallennettaisiin päivämäärä ja aika, kun siirtotiedosto lähetettiin. Siirtotiedoston versiohistoria näkyisi ylläpitoliittymässä omassa näkymässään, jonka kautta voitaisiin kantaan palauttaa halutun siirtotiedoston mukaiset tiedot.

Tutkielmassa käsitelty tuotetietojen esitystapa, tuotekoodin validointi, hinnanlaskenta modulaariselle tuotteelle ja rajapintasovellus ovat käytettävissä myös muissa tuotteissa. Tällöin vaatimuksena on, että tuotetiedot kuvataan määrittelyvaiheessa samoin kuin tässä tutkielmassa. Rajapintasovellus ei sinällään aseta rajoitteita toiminnanohjausjärjestelmälle, mutta materiaalikoodi on SAP-toiminnanohjausjärjestelmän nimitys tuotteiden erittelyyn. Tällä ei järjestelmän toiminnan kannalta ole merkitystä, vaan materiaalikoodia voi kutsua miksi haluaa. Selectica-tuotekonfiguraattorille tuotetietojen esitystapa on toimiva, muttei täydellinen. Ongelmaksi muodostuu, että tuotekonfiguraattoriin tulee lisätä käsin taulukkomuotoinen rajoitesääntö jokaista validointisääntöä kohden. Tämän vuoksi tuotekonfiguraattori on usein toiminnallisuuden osalta hieman jäljessä verrattuna siihen, miten tiedot näkyvät rajapinnan yli toiminnanohjausjärjestelmään. Rajoitesääntöjen lisäämisen osalta tuotekonfiguraattorissa voisikin olla enemmän automatisointia.

Järjestelmä on aktiivisessa käytössä ja sitä kehitetään edelleen. Se toimii vakaasti ja yleisimmät virhetilanteet liittyvät tyypillisesti palvelinten alasajoon. Rajapintasovellus ja etenkin ylläpitoliittymä tarjoavat valmiin alustan monille uusille toiminnoille. Tämän vuoksi ylläpitoliittymään suunnitellaan jatkuvasti uusia ominaisuuksia. Voisi sanoa, että järjestelmä ei ole valmis eikä se valmiiksi tulekaan. Tätä ei tule tulkita negatiivisesti, vaan pikemminkin kunnianosoitukseksi järjestelmän dynaamisuutta kohtaan.

## LÄHDELUETTELO

- ABB Oy (2011). *ABB Pienjännitetuotteet, Huoneistokeskukset* [online] [siteerattu 2.4.2011]. Saatavana Word Wide Webistä: <URL: [http://www05.abb.com/global/scot/scot209.nsf/veritydisplay/2266ec39b2925191c12577ee00430773/\\$file/58735\\_huoneistokeskukset\\_fi\\_lowres.pdf](http://www05.abb.com/global/scot/scot209.nsf/veritydisplay/2266ec39b2925191c12577ee00430773/$file/58735_huoneistokeskukset_fi_lowres.pdf)>.
- Bakker, R.R., Dikker, F., Tempelman, F., Wogmim, P.M. (1993). *Diagnosing and solving over-determined constraint satisfaction problems* [online] [siteerattu 3.4.2011]. Saatavana World Wide Webistä: <URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.82.835&rep=rep1&type=pdf&rct=j>>.
- Bensana, Eric, Mulyanto, Taufiq, Verfaillie, Gérard (2000). *Dealing With Uncertainty in Design and Configuration Problems* [online] [siteerattu 3.4.2011]. Saatavana World Wide Webistä: <URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.3278&rep=rep1&type=pdf&rct=j>>.
- Boehm, Barry W. (1984). *Software Engineering Economics*. Prentice Hall. 767 s. ISBN 978-0138221225.
- C# Online.NET (2011). *C# and Visual C# Development: C# Articles: struct* [online] [siteerattu 13.2.2011]. Saatavana World Wide Webistä: <URL: <http://en.csharp-online.net/struct>>.
- Groh, Michael (2002). *Creating Components in .NET* [online] [siteerattu 2.3.2011]. Informant Communications Group and Microsoft Corporation, PC Productivity Solutions. Saatavana World Wide Webistä: <URL: <http://msdn.microsoft.com/en-us/library/ms973807.aspx>>.
- Haas, Hugo & Brown Allen (2004). *Web Services Glossary* [online] [siteerattu 20.1.2011]. Saatavana World Wide Webistä: <URL: <http://www.w3.org/TR/ws-gloss/>>.

- Haikala, Ilkka & Märijärvi Jukka (2004). *Ohjelmistotuotanto*. 10. painos. Hämeenlinna: Talentum Media Oy ja tekijät. 440 s. ISBN 952-14-0850-2.
- Jørgensen, Kaj A (2003). *Modelling Configurable Products*. 16 s. Julkaisematon. Aalborg University, Department of Production.
- SAP AG (2001). *Material Master (LO-MD-MM)* [online] [siteerattu 11.2.2011]. 154 s. Saatavana World Wide Webistä: <URL: <http://help.sap.com/printdocu/core/print46c/en/data/pdf/LOMDMM/LOMDMM.pdf>>.
- Microsoft Corporation (2010). *MSDN Library: Global Assembly Cache* [online] [siteerattu 5.3.2011]. Saatavana World Wide Webistä: <URL: <http://msdn.microsoft.com/en-us/library/yf1d93sz%28v=vs.71%29.aspx>>.
- Microsoft Corporation (2010). *MSDN Library: Using Windows Event Log* [online] [siteerattu 28.2.2011]. Saatavana World Wide Webistä: <URL: <http://msdn.microsoft.com/en-us/library/aa385772%28v=VS.85%29.aspx>>.
- Microsoft Corporation (2010). *How To: Use Windows Authentication in ASP.NET 2.0* [online] [siteerattu 28.2.2011]. Saatavana World Wide Webistä: <URL: <http://msdn.microsoft.com/en-us/library/ff647405.aspx>>.
- Mukherjee, Sujan K., Ryan, James L., Wason, James R. (1994). *Method and system for product configuration definition and tracking*. U.S. patentti 5311424.
- Quin Liam (2011). *W3C: Extensible Markup Language (XML)* [online] [siteerattu 14.3.2011]. Saatavana World Wide Webistä: <URL: <http://www.w3.org/XML/>>.
- Salonen, Jaakko (2006). *Pikajohdanto XML-skeemoihin* [online] [siteerattu 3.3.2011]. Saatavana World Wide Webistä: <URL: <http://blite.iki.fi/artikkelit/pikajohdanto-xml-skeemoihin/>>.



Selectica, Inc. (2005). *Selectica Studio: Model Builder Guide (Release 8.0)*. 502 s. Teos on yhtiön tekemä manuaali eikä sitä ole julkisesti saatavilla.

Stack Overflow (2010). *Stack Exchange: Multiple output parameters in .NET web service (without complex type)* [online] [siteerattu 7.3.2011]. Saatavana World Wide Webistä: <URL: <http://stackoverflow.com/questions/3736270/multiple-output-parameters-in-net-web-service-without-complex-type>>.

Tiihonen Juha, Lehtonen Timo, Soininen Timo, Pulkkinen Antti, Sulonen Reijo & Rii-tahuhta Aso (1998). *Modelling Configurable Product Families*. 22 s. Julkaisema-ton. Delft University of Technology.

Vacon Oy (2008). *Vacon 10 Machinery* [online] [siteerattu 2.4.2011]. Saatavana World Wide Webistä: <URL: <http://www.vacon.fi/File.aspx?id=470362&ext=pdf&routing=396771&webid=396774&name=BC00286F>>.

Vamp Oy (2011). *Suojareleen lajimerkkiavain* [online] [siteerattu 2.4.2011]. Saatavana World Wide Webistä: <URL: <http://www.vamp.fi/Type%20designations/VTD96.FI002.pdf>>.

Vieira, Robert (2003). *Professional SQL Server 2000 Programming*. Indianapolis: Wi-ley Publishing, Inc. 1390s. ISBN 0-7645-4379-2.

Viertola, Raimo (2011). *Matematiikka, pitkä oppimäärä: MAA6 - Todennäköisyys ja tilastot* [online] [siteerattu 13.3.2011]. Saatavana World Wide Webistä: <URL: <http://opinnot.internetix.fi/fi/muikku2materiaalit/lukio/maa/maa06/index>>.

## LIITTEET

### Liite 1. Materiaalikoodin hakeminen C#-ohjelmointikielellä SQL-funktiota apuna käyttäen ja virheiden kirjoittaminen Windows event log:iin

```

public string GetMaterialCode(string orderingCode)
{
    // Lue konfigurointitiedostosta SQL-palvelimen käynnistyslause ja
    hinnoittelukoodin SQL-lause
    string connectionString = ReadConfig("databaseConnectionString");
    string plusCodeQuery = ReadConfig("materialCodeQuery");
    string connection = connectionString;

    // Luo tietokantayhteys
    SqlConnection myConnection = new SqlConnection(connection);
    SqlCommand myCommand = new SqlCommand(plusCodeQuery, myConnection);
    myCommand.CommandType = CommandType.Text;
    myCommand.CommandTimeout = connectionTimeout;

    // Aseta tietokantayhteyden parametrit
    SqlParameter sqlOrderingCode = new SqlParameter();
    sqlOrderingCode.ParameterName = "@OrderingCode";
    sqlOrderingCode.Value = orderingCode;
    myCommand.Parameters.Add(sqlOrderingCode);

    try
    {
        // Avaa tietokantayhteys
        myConnection.Open();
        // Aja SQL-lause ja palauta sen palauttama arvo
        return Convert.ToString(myCommand.ExecuteScalar());
    }
    catch (SqlException se)
    {
        // Kirjoita kaikki virheilmoitukset Windows event log:iin
        string appName = "SAPtoMopo ProductInformationQueryLibrary
GetMaterialCode";

        if (!EventLog.SourceExists(appName))
        {
            EventLog.CreateEventSource(appName, "Application");
        }

        string errorMessage =
CreateGetMaterialCodeExceptionMessage(se.Message, orderingCode,
connectionString, plusCodeQuery);
        EventLog.WriteEntry(appName, errorMessage, EventLogEntryType.Error);

        // Sql exception -tyyppinen virhe johtuu tyypillisesti
yhteysongelmasta tietokantaan
        return "ERROR:Connection failure between
ProductInformationQueryLibrary and database.";
    }
    catch (Exception e)
    {
        // Kirjoita kaikki virheilmoitukset Windows event log:iin
        string appName = "SAPtoMopo ProductInformationQueryLibrary
GetMaterialCode";

        if (!EventLog.SourceExists(appName))

```

```
{
    EventLog.CreateEventSource(appName, "Application");
}

string errorMessage = CreateGetMaterialCodeExceptionMessage(e.Message,
orderingCode, connectionString, plusCodeQuery);
EventLog.WriteEntry(appName, errorMessage, EventLogEntryType.Error);

// Kaikki muut virheet kuin Sql exception -tyyppiset virheet
return "ERROR:Unknown error. Please contact administrator.";
}
finally
{
    // Lopuksi sulje yhteys
    myConnection.Close();
}
}

private string CreateGetMaterialCodeExceptionMessage(string exceptionMessage,
string orderingCode, string connectionString, string plusCodeQuery)
{
    return "Error: " +
exceptionMessage + "\n\n" +
"Parameters: " + "\n\n" +
"OrderingCode: " + orderingCode + "\n" +
"ConnectionString: " + connectionString + "\n" +
"MaterialCodeQuery: " + plusCodeQuery;
}
```

## Liite 2. Siirtotiedoston XML-skeema

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDe-
fault="qualified">
  <xs:element name="ROOT">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Product"/>
        <xs:element ref="MaterialCode"/>
        <xs:element ref="AllowedChars"/>
        <xs:element ref="Rules"/>
        <xs:element ref="PricingCodes"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Product" type="xs:NCName"/>
  <xs:element name="MaterialCode" type="xs:NCName"/>
  <xs:element name="AllowedChars">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="Item"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Rules">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="MaterialCodeRule"/>
        <xs:element ref="ValidationRules"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MaterialCodeRule">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="OrderingCodeChars"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="OrderingCodeChars">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="Item"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ValidationRules">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="ValidationRule"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ValidationRule">
    <xs:complexType>
      <xs:sequence>

```

```

        <xs:element ref="TestChars"/>
        <xs:element maxOccurs="unbounded" ref="RuleLine"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="TestChars" type="xs:integer"/>
<xs:element name="RuleLine">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" ref="Item"/>
        </xs:sequence>
        <xs:attribute name="VisibleInSAP" use="required"
type="xs:NCName"/>
        <xs:attribute name="VisibleInSelectica" use="required"
type="xs:NCName"/>
    </xs:complexType>
</xs:element>
<xs:element name="PricingCodes">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" ref="PricingCode"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="PricingCode">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" ref="Item"/>
        </xs:sequence>
        <xs:attribute name="Desc" use="required"/>
        <xs:attribute name="GroupOrderNumber" use="required"/>
        <xs:attribute name="GroupTextOrderNumber" use="required"/>
        <xs:attribute name="SummaryGroupText" use="required"/>
        <xs:attribute name="Value" use="required" type="xs:NCName"/>
    </xs:complexType>
</xs:element>
<xs:element name="Item">
    <xs:complexType mixed="true">
        <xs:attribute name="CharNo" use="required"/>
        <xs:attribute name="Desc"/>
    </xs:complexType>
</xs:element>
</xs:schema>

```