Hissi - Teoria

1 Oppimistavoitteet

- Projektinhallinta, syventyen koneautomaatio-alan käytäntöihin
- V-malli ja kyky liittää suunnitteludokumentaatio V-mallin vaiheisiin; testauksen vaiheistus
- Koneautomaation suunnitteludokumentaatio: käyttötapauskuvaus, riskianalyysi, järjestelmätoimintojen määrittely
    - Kyky lukea ja kirjoittaa koneautomaation suunnitteludokumentaatiota osana V-mallin mukaisesta projektia
- PLC-kehitys koneautomaation suunnitteludokumentaatiota vastaan ja V-mallin vaiheiden mukaisest

2 Teoria

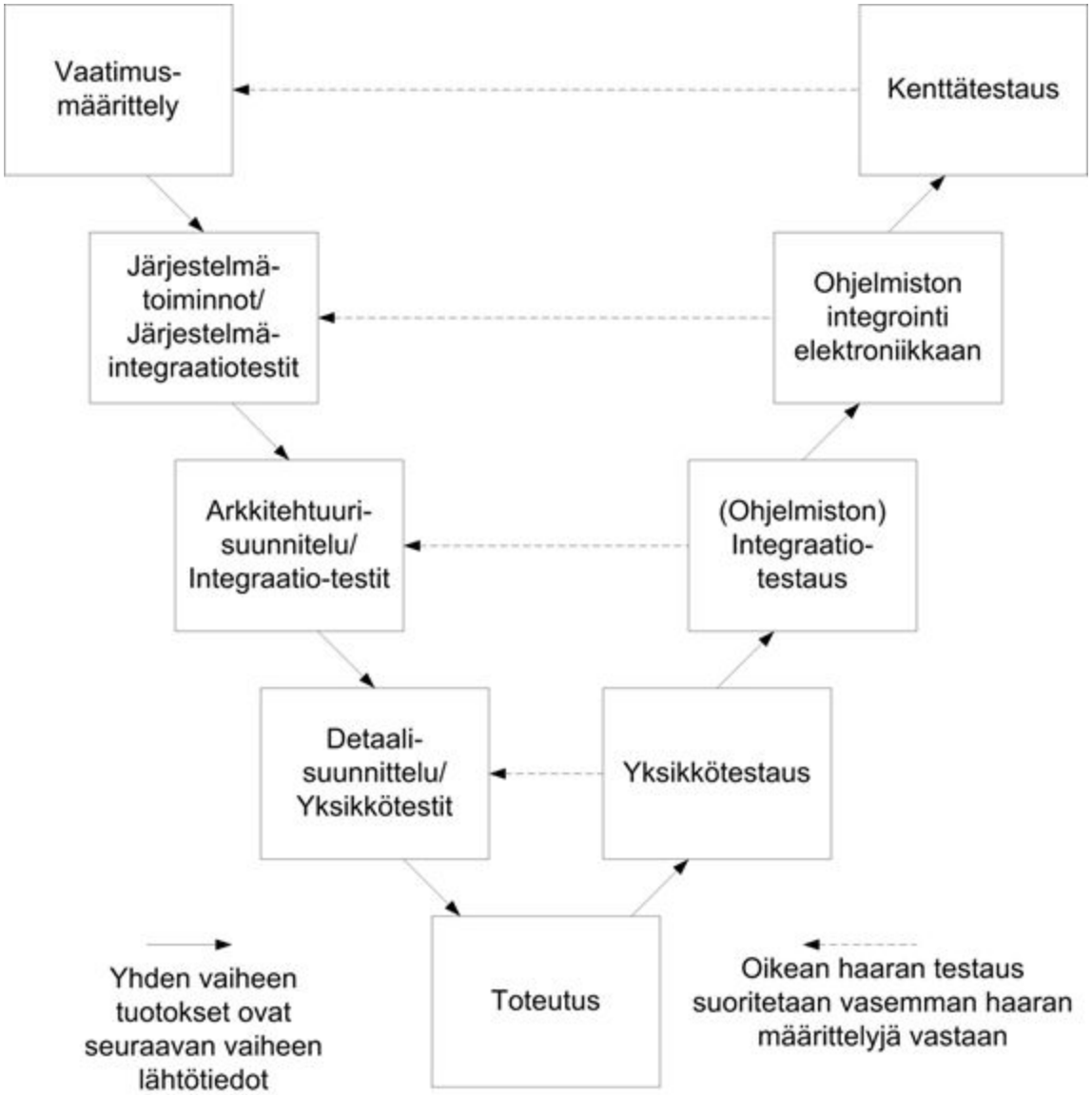
Projekti voidaan määritellä kertaluontoisena tehtävänä, jolla on tavoite, aikataulu, resurssit ja suunnitelma. Yksittäisellä teollisuussektorilla muodostuu kokemuksen kautta hyviä käytäntöjä projektin suunnitteluun, minkä ansiosta suunniteltu aikataulu toteutuu myös käytännössä. Suomalaisten teollisuusautomaatio-alan vientiyritysten kannalta haasteena ei ole että saadaanko projekti ennemmin tai myöhemmin toimimaan – tämä ei riitä nykyisessä kilpailutilanteessa. Haasteena on että saadaanko projekti valmiiksi suunnitellussa aikataulussa. Mitä kireämmäksi aikataulut voidaan suunnitella, sitä kannattavampaa liiketoimintaa voidaan harjoittaa.

Prosessiautomaatiossa ja koneautomaatiossa on joitain eroja projektin vaiheistamisen käytännöissä. Tällä kurssilla syvennytään koneautomaation käytäntöihin hissi harjoitustyön kautta. Noudatetaan V-mallia, joka on koneautomaation lisäksi erittäin laajassa käytössä esim. ydinvoima-automaatiossa, autoteollisuudessa, raideliikennesektorilla, puolustusteollisuudessa, ilmailuteollisuudessa, sulautetuissa järjestelmissä sekä yleisesti IT alalla. Tämän lisäksi V-malliin pohjautuu automaatio-alan kannalta tärkeimmät turvallisuusstandardit. Näin ollen kun näet kuvan V-mallista, sinun tulee olla tietoinen siitä että mallista on useita variaatioita. Sinun tulee ensiksi selvittää että onko kysymyksessä pelkästään ohjelmistonkehitystä tukeva malli vain onko kohteena ohjelmistoa sekä fyysistä laitteistoa sisältävä järjestelmä. Tämän jälkeen kannattaa miettiä että onko malli suunnattu jollekin tietylle teollisuussektorille.

Kuva 1 esittää koneautomaatio-alan V-mallin. Vasen laskeva haara sisältää työvaiheet, joissa selvitetään mitä asiakas haluaa, miten nämä tarpeet voidaan kuvata teknisesti yksikäsitteisessä muodossa, ja mitä korkean ja matalan tason suunnitteluratkaisuja aiotaan noudattaa. Vasta tämän jälkeen toteutetaan järjestelmä. Vmallin oikea, nouseva haara puolestaan sisältää laadunvarmistukseen liittyviä vaiheita. Yleisin laadunvarmistustekniikka on testaus. Jokaisen

laadunvarmistusvaiheen suunnittelu edellyttää lähtötietoja. V-malli on järjestetty niin, että jokaisen oikean haaran vaiheen lähtötiedot tuotetaan samalla tasolla olevassa vasemman haaran vaiheessa. Esimerkiksi yksikkötestaussuunnitelmat voidaan laatia heti kun detaalisuunnittelu on tuottanut moduulien tarkat rajapintakuvaukset. Testisuunnitelmat voidaan siis laatia ennen ohjelmointityön aloittamista.

Kuva 1:en mallissa kaksi ylintä kerrosta koskevat koko järjestelmää. Tämän jälkeen prosessi haarautuu erillisiin ohjelmisto-, elektroniikka- ja mekaniikka suunnittelu prosesseiksi. Näiden prosessien tuotoksen voidaan myöhemmin integroida toisiinsa, jos kaikki ovat noudattaneet samoja järjestelmätoimintojen määrittelyjä. Näistä määrittelyistä esim. PLC ohjelmoija näkee että mihin I/O moduulin paikkaan tietty anturi tullaan kytkemään, eikä hänen tarvitse tietää elektroniikkasuunnittelun tarkempia yksityiskohtia. Elektroniikka- ja mekaniikkasuunnittelu tapahtuvat niitä varten erityisesti suunnitelluilla CAD ohjelmistoilla, jotka ovat tämän kurssin ulkopuolella ja joita ei näytetä Kuva 1:ssä. Kuva 1:en kolme alinta kerrosta kuvaavat siis PLC-kehitysprosessia.



Kuva 1: Koneautomaatio-alan V-malli

Jokainen V-mallin vaihe tuottaa artefakteja (artifact). Artefakti on arkeologien ja kulttuuritutkijoiden keksimä termi, joka kuvaa mitä tahansa ihmisen tekemää esinettä. Sittemmin termi on otettu laajaan käyttöön IT-alalla, jossa sillä kuvataan ohjelmistonkehitysprosessissa syntyviä konkreettisia tuotoksia, kuten kaavioita, tekstidokumentteja ja ohjelmakoodia. Esimerkkejä Kuva 1:en V-mallia noudattavan projektin tuottamista artefakteista ovat järjestelmätoimintojen määrittely, moduulitestisuunnitelmat, testiraportit sekä PLC koodit. Suurin osa artefakteista on dokumentaatiota, joita niitä tarvitaan projektin systemaattiseen läpivientiin sekä siihen, että kehittäjät, asiakkaat ja turvallisuudesta vastaavat viranomaiset voivat vakuuttua siitä, että valmiin tuotteen laatu ja turvallisuus täyttävät sille asetetut odotukset.

V-malli ei ratkaise sellaisia ongelmia, että kuinka muutoksia tai korjauksia voidaan tehdä mahdollisimman nopeasti. Teollisuuden projekteissa muutostilanteissa ei yleensä iteroida

kaikkia V-mallin vaiheita perusteellisesti uudestaan. Mutkia oikaistaan joko kokemukseen ja hiljaiseen tietoon perustuen tai systemaattisemmin ketteriä (agile) kehitysmenetelmiä käyttäen. Nämä asiat ovat tämän kurssin aihepiirin ulkopuolella ja niihin sisältyy vielä runsaasti tutkimuksellisia haasteita, etenkin kun kysymyksessä on turvallisuuskriittiset tuotteet, joiden kehitysvaiheet tulee dokumentoida perusteellisesti. Mutkia ei kuitenkaan kannata alkaa oikomaan ennen kuin osaa noudattaa V-mallia oikeaoppisesti ja on hankkinut jonkun verran kokemusta teollisuuden projekteista.

2.1 Vaatimusmäärittely

2.1.1 Käyttötapaus

Käyttötapaus (use case) on IT-alalla yleisesti käytetty tapa kuvata järjestelmän ja sen käyttäjän välistä interaktiota. Näin ollen käyttötapausten laatiminen on hyvä tapa varmistua siitä, että tuote tulee täyttämään loppukäyttäjien ja asiakkaan tarpeet. Käyttötapaus ei pyri toimimaan insinööreille suunnattuna spesifikaationa, joten käyttötapauksista pitää johtaa järjestelmätoimintojen määrittelyt seuraavassa vaiheessa. Koneautomaatiossa käyttötapaus kuvaa koneen ja sen käyttäjän välistä interaktiota, joten se toimii myös hyvänä lähtötietona, kun koneen turvallisuuteen liittyvässä riskinarvioinnissa kartoitetaan tästä interaktiosta syntyviä riskejä. Riskinarviointi edellyttää laajempia lähtötietoja, kuin mitä IT-alan käyttötapauspohja määrittelee. Tämän takia suomalaisessa työkoneteollisuudessa on laajassa käytössä VTT:n KOTOTU (Koneiden Ohjausjärjestelmien TOiminnallinen TUrvallisuus) hankkeessa luoma käyttötapauskuvaus.

2.1.2 Turvallisuus ja riskinarviointi

Prosessiautomaatio ja koneautomaatio eroavat merkittävästi toisistaan siinä, miten turvallisuusasiat on ratkaistu. Yksi lähestymistapa on jakaa automaatio käyttöautomaatioon ja siitä täysin erilliseen turvallisuuteen liittyvään järjestelmään (TLJ). TLJ monitoroi prosessia ja puuttuu peliin jos se havaitsee riskitilanteen. Tällöin käyttöautomaatio voidaan toteuttaa ilman turvallisuusvaatimuksia ja sertifiointia. Toinen lähestymistapa on toteuttaa käyttöautomaatio siten, että voidaan osoittaa sen täyttävän sovellusalueelle relevanttien turvallisuusstandardien mukaiset määräykset. Prosessiautomaatiossa yleensä on erillinen TLJ kun taas suomalaisissa työkonealan yrityksissä on miltei aina päädytty siihen ratkaisuun, että on kustannustehokkaampaa olla käyttämättä erillistä TLJ:tä. Ydinvoima-alalla puolestaan on käytössä useita eri automaatiojärjestelmiä, joilla on eritasoisia turvaluokituksia. Tällä kurssilla keskitytään koneautomaation käytäntöihin, koska nämä perusperiaatteet tulevat vastaan muillakin sovellusalueilla jos joku vaikka diplomityövaiheessa paneutuu turvallisuusaiheeseen. (Teollisuuden teettämät turvallisuusaiheiset diplomityöt ovat tässä koulutusohjelmassa melko yleisiä.) V-mallin mukainen kehitysprosessi on keskeisten turvallisuusstandardien

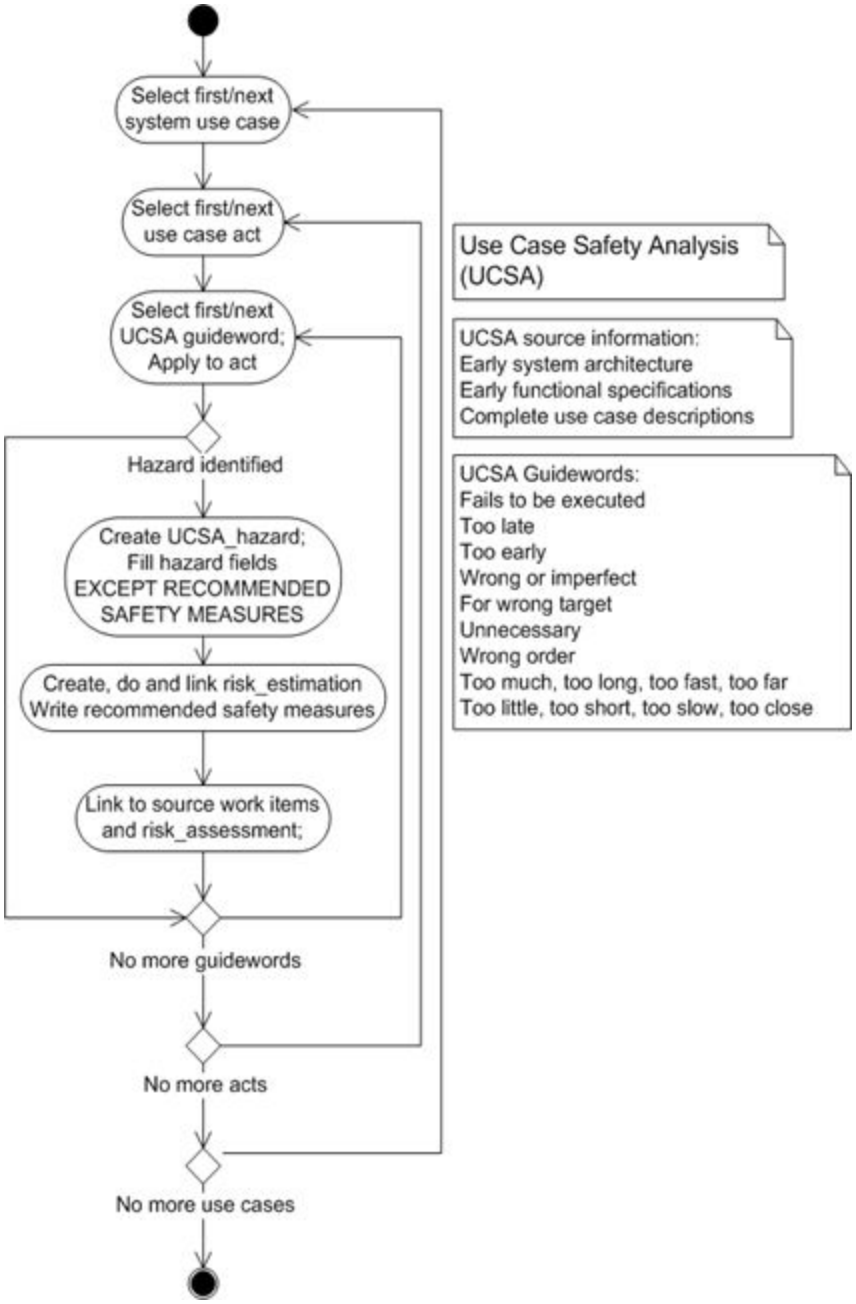
lähestymistapa turvallisuuteen. V-malli on laajasti käytössä myös ei-turvallisuuskriittisissä sovelluksissa, mutta turvallisuusaspektit laajentavat sitä esimerkiksi riskinarvioinnilla.

Turvallisuuskriittisten automaatiosovellusten osalta pitää voida osoittaa viranomaisille että riskit on vähennetty hyväksyttävällä tasolle. Toisin sanoen, on yleisesti ymmärretty, että kaupalliset järjestelmät ovat sen verran monimutkaisia että niistä ei voi saada täysin turvallisia. Näin ollen riskit pitää tunnistaa ja niiden suuruus arvioida, jotta voidaan tarvittaessa määritellä riskejä vähentävät turvatoiminnot ja sitten toteuttaa turvatoiminnot standardien edellyttämällä tavalla. Turvallisuuden yksityiskohdat eivät mahdu tämän kurssin aihepiiriin, joten turvallisuuteen annetaan johdanto siitä näkökulmasta, että miten se liittyy automaatiosovelluksen projektinhallintaan ja suunnitteluun, mikä on tämän kurssin ydinasiaa. Näin ollen keskitytään siihen miten suunnitteludokumentaatiosta tunnistetaan riskit ja arvioidaan niiden suuruus (riskinarviointi) ja määritellään tämän perusteella turvatoiminnot.

Riskinarvioinnin lähtötietona on suunnitteludokumentaatio. V-mallin vasemmassa haarassa tuotetaan erilaisia suunnitteludokumentteja, joten niiden systemaattiseen arviointiin on olemassa erilaisia riskinarviointimenetelmiä. OHA (Operational Hazards Analysis) on menetelmä, jonka avulla etsitään koneen ja sen käyttäjän interaktiosta syntyviä riskejä. OHAn heikkous on siinä että se ei edellytä lähtötietojen olevan missään tietyssä muodossa, joten OHA ei tarjoa systemaattista menetelmää lähtötietojen läpikäymiseen. Tämän takia VTT on kehittänyt OHAsta variaation ”käyttötapausanalyysi”, jonka lähtötietona ovat tämän dokumentin mukaiset käyttötapaukset. Käyttötapausanalyysia on sovellettu laajasti suomalaisessa työkoneteollisuudessa.

Kuva 2 esittää käyttötapausanalyysin työnkulun. Otetaan esimerkiksi käyttötapaus 1 ”Hissin kutsuminen kun käyttäjä seisoo hissin ulkopuolella”. Use case act viittaa ”tapauksen kulku” otsikon alta löytyviin numeroituihin vaiheisiin. Nämä vaiheet käydään yksitellen läpi ja jokaiseen sovelletaan vuorotellen käyttötapausanalyysin avainsanoja (Kuva 2: UCSA guidewords). Insinöörin tehtävä on päättää, että tulkitaanko vaiheen ja avainsanan kombinaatio vaaraksi (hazard). Jos vastaus on kyllä, siirrytään Kuva 2 ehdosta (salmiakkikuvio) ”hazard identified” haaraan ja muussa tapauksessa ohitetaan seuraavat vaiheet ja palataan nuolia pitkin takaisin. Näin ollen menetelmä edellyttää että jokaisen käyttötapauksen jokaiseen vaiheeseen yritetään soveltaa jokaista avainsanaa. Tässä esimerkissä huomataan seuraava vaara:

- Vaihe 3: Sitten kun tämä kutsu otetaan jonosta työn alle, ajetaan ylös/alas riippuen siitä että missä suunnassa kerros X on.
- Avainsana: Liian aikaisin (too early)
- Tulkinta: Hissi lähtee liikkeelle ennen kuin ovet ovat kiinni.
- HUOM: Käyttötapauskuvauksessa ei tässä tapauksessa ollut virhettä, koska ohjeistettiin sulkemaan ovet ennen liikkeelle lähtöä. Kuitenkin havaitaan, että mahdollinen tekninen virhe voi aiheuttaa vakavasti otettavan vaaratilanteen.



Kuva 2 Käyttötapausanalyysin työnkulku

Kuva 2 edellyttää että kun vaara on löytynyt, luodaan seuraavat artefaktit: vaarakuvaus ja riskin suuruuden arviointi (risk estimation).

2.1.3 Vaarakuvaus vaaralle ”Hissi lähtee liikkeelle ennen kuin ovet ovat kiinni”

Elinkaarivaiheet: Testaus, Käyttöönotto, Normaali operointi, Asetusten teko, Vianhaku, Puhdistus

Vaara-alue: Hissin oven ympäristö rappukäytävässä ja hissi

Vaarallinen tilanne: Hissi kutsutaan kun ovet eivät ole kiinni

Vaarallinen tapahtuma: Hissi lähtee liikkeelle kun ovet eivät ole kiinni

Vaara: Vaara-alueella oleva henkilö on hissin oviaukossa ja kaatuu tai jää puristuksiin.

Suositellut turvallisuus toimenpiteet [HUOM: TÄYTETÄÄN VASTA RISKIN SUURUUDEN ARVIOINNIN JÄLKEEN]: Tarvitaan elektronisella tasolla toteutettu. turvatoiminto, joka varmistaa että hissi ei lähde liikkeelle ellei ovet ole kiinni.

2.1.4 Riskin suuruuden arviointi vaaralle ”Hissi lähtee liikkeelle ennen kuin ovet ovat kiinni”

HUOM: Katso dokumentti ”Risk estimation guide IEC 62061”, joka kuvaa tämän koneturvallisuusstandardin mukaisen riskinarvioinnin. Huomaa että riskinarviointi tehdään pahimman realistisen skenaarion mukaan.

Vakavuus/Severity: 4 (death, loss of vision or a hand)

Altistumistaajuus/Frequency of exposure: 5 (less than 1 hour)

Tapahtumistodennäköisyys/Occurrence probability: 4 (Probable) [Siksi, koska hissin käyttäjät voivat olla vanhuksia tai muita henkilöitä, joiden henkinen ja fyysinen suorituskyky on heikentynyt.]

Välttämisstodennäköisyys/Avoidance probability: 3 (Possible)

Risk index: CI=5+4+3=12 SIL=3

Standardin taulukon mukaan riskin suuruus edellyttää SIL (Safety Integrity Level 3) tason turvatoimintoa, mikä on korkein taso koneautomaatiossa. Näin ollen ei riitä että turvatoiminto toteutetaan ohjelmistolla. Tyypillinen ratkaisu olisi erillinen elektroninen (siis ei sisällä ohjelmistoa) turvapiiri, joka tunnistaa jokaisen oven osalta että onko se kiinni. Hissiä kutsuttaessa turvapiiri lukitsee ovet mikäli kaikki ovet ovat kiinni (saatat pystyä kuulumaan tähän liittyvän loksahduksen kun olet hississä). Turvapiiri on toteutettu elektronisesti niin että hissiä liikuttavat toimilaitteet eivät voi käynnistyä, ellei turvapiiri ole kiinni.

HUOM: turvapiirin suunnittelu ja toteuttaminen on tämän kurssin ulkopuolelle menevää asiaa, joten turvallisuusprosessia ei käydä läpi tämän pidemmälle. Tästä eteenpäin se on turvallisuusammattilaisten erikoisosaamista. Tähän asti käydyt vaiheet puolestaan ovat myös järjestelmäsuunnittelijoille kuuluvaa osaamista.

2.2 Järjestelmätoimintojen määrittely

Automaatiojärjestelmää ja sen ohjelmistoa ei ole yleensä mielekästä suunnitella siten, että jokaista käyttötapausta varten on sitä toteuttava komponentti tai moduuli, koska näin saattaa tulla paljon turhaa päällekäisyyttä. Käyttötapauskuvaukset kuvaavat koneen ja käyttäjän välisen interaktion, joten ne ovat hyödyllisiä kun selvitetään loppukäyttäjän tarpeita ja kun kartoitetaan tästä interaktiosta syntyviä riskejä. Järjestelmätoiminnot puolestaan kuvaavat miten sisääntulot prosessoidaan ja miten ulostulot muodostetaan, joten ne ovat askel kohti teknistä toteutusta. Järjestelmätoiminnot ovat yhteinen määritelmä ja sopimus ohjelmisto, elektroniikka ja mekaniikkainsinöörien välillä. Niiden määrittelyn jälkeen näiden aspektien kehittäminen jatkuu omassa haarassa.

2.3 Arkkitehtuurisuunnittelu

Arkkitehtuuri tässä tarkoittaa siis ohjelmistoarkkitehtuuria: tästä alaspäin V-malli koskee vain ohjelmistoa.

Tässä ratkaistaan arkkitehtuurin suunnitteluperiaatteet. Oman toimilohkotyypin määrittely on perusteltua jos samaa koodia instantioidaan useammassa kohdassa. Tässä tapauksessa tällaista tarvetta ei ole näköpiirissä. Hissin ohjauslogiikan voidaan jakaa järjestelmätoimintoon 1 ja 2, jotka voidaan sitten toteuttaa samaan ohjelmaan tai erillisiin ohjelmiin.

Järjestelmätoiminnot:

1. Ajetaan nykyisestä paikasta kerrokseen X ja tarvittaessa pysähdytään matkan varrella. Tämän toiminnon ei tarvitse tietää että onko komento kerrokseen X saatu rappukäytävän vai hissin sisäisillä kutsunapeilla.
2. Monitoroidaan hissin ja rappukäytävän kutsunappeja, tarvittaessa laitetaan kutsuja jonoon ja puretaan jonon vanhin kutsu kun toiminto 1 on tullut valmiiksi.

Erityiseksi tavoitteeksi annetaan **testattavuus**, siten että kumpikin toiminto voidaan yksikkötestata. Tämä tarkoittaa sitä että kummallekin toiminnolle löytyy oma moduuli, joka toteuttaa toiminnon kokonaisuudessaan eikä sisällä muuta toiminnallisuutta ja että tällä moduulilla on hyvin määritelty rajapinta. Näin ollen kumpikin toiminto toteutetaan omassa ohjelmassaan. Kaikki nämä signaalit on määritelty globaaleiksi muuttujiksi, jotta ne näkyvät molemmissa ohjelmissa.

2.4 Detaalisuunnittelu

Tämä on selvästi haastavampi ohjelmointiharjoitus kuin mikään tähänastisista. Jotta ohjelmistonkehitys ei veisi liikaa aikaa, tässä annetaan vinkkejä miten ohjelmisto kannattaa suunnitella. Tässä luvussa mainittujen suunnitteluperiaatteiden noudattaminen on pakollista vain jos haluaa teknistä tukea.

Aluksi vaikuttaa siltä että SFC kieli olisi sopiva. Kuitenkin SFC soveltuu ainoastaan tapauksiin, joissa vaiheiden sekvenssi on ennaltamääritelty. Tässä tapauksessa ei voida tietää että missä järjestyksessä hissiä kutsutaan eri kerroksiin. FBD kieli ei puolestaan sovellu, jos on vähänkään enemmän ehtolauseita tai silmukoita. Näin ollen kannattaa käyttää ST:tä.

Taulukko 1: Järjestelmätoiminto 1:en toteuttavan ohjelman suunnittelu

<table>
<tr><th>Tilan nimi</th><th>Aktiivinen jos</th><th>Toiminta</th></tr>
<tr><td>Liikkeelle lähtö</td><td>Liikkeelle lähtö Ollaan kerroksessa Y, joka ei ole X ja Go on tosi ja kaikki ovet ovat kiinni Jos Y>X start motor down, muuten start motor up</td><td>Jos Y>X start motor down, muuten start motor up</td></tr>
<tr><td>Saavutaan matkan varrella kerrokseen 1 ja avataan ovi</td><td colspan="2" rowspan="4">???</td></tr>
<tr><td>Kerroksen 1 ovi kokonaan auki</td></tr>
<tr><td>Ollaan kerroksessa X tai saavutaan kerrokseen X ja avataan ovi</td></tr>
<tr><td>Kerroksen X ovi kokonaan auki</td></tr>
</table>

Huomataan että toiminto voidaan mallintaa tilakoneella. Taulukko 1 luettelee 5 tilaa, joissa ohjelma voi olla. IT alalla laajasti käytetyn tilakoneen (statechart) määritelmä [jos haluatte tietää formaalimman määritelmän tilakoneelle, suorittakaa tietojenkäsittelyteorian kurssi viereisessä talossa] on että järjestelmä on aina täsmälleen yhdessä tilassa. Tilakone myös määrittelee että millä logiikalla voidaan vaihtaa tilasta toiseen. Koska PLC kielet eivät tue tilakoneita, lähdetään siitä että mistä tahansa tilasta voidaan siirtyä mihin tahansa tilaan heti kun uuden tilan

”aktiivinen jos” ehto (kts Taulukko 1) toteutuu. Näin ollen ehdot pitää kirjoittaa siten että ne ovat toisensa pois sulkevia. Taulukon mukainen tilakone voidaan toteuttaa ST kielellä seuraavasti:

```
IF <tila1 ”aktiivinen jos” ehto tosi> THEN

  <tila1 toiminta>

ENDIF;

IF <tila2 ”aktiivinen jos” ehto tosi> THEN

  <tila2 toiminta>

ENDIF;
```

Jne.