

Suunnittelumallit turva-automaatiossa

Timo Vepsäläinen, Jari Rauhamäki, Seppo Kuikka

Tampereen teknillinen yliopisto, Systeemiteknikan laitos, PL 692, 33101 TAMPERE

Puh. +358408490062, {timo.vepsalainen, jari.rauhamaki, seppo.kuikka}@tut.fi

AVAINSANAT suunnittelumalli, mallipohjainen kehitys, turvallisuus

TIIVISTELMÄ

Turva-automaatiojärjestelmät toteuttavat automatisoitavien prosessien turvallisuuden vaatimat toiminnot. Perinteisesti turva-automaatiossa on suosittu mm. mekaanisia ja sähköisiä ratkaisuja, jotka ovat ohjelmistoja helpompia kehittää. Ohjelmistoihin perustuvat ratkaisut niin turva- kuin käyttöautomaatiossa kuitenkin yleistyvät mm. teollisen internetin kehityksen myötä. Niiden kehityksessä korostuvat käytettävät kehitysprosessit, -tekniikat ja -ratkaisut sekä työn dokumentointi. Tämä artikkeli keskittyy ohjelmistotekniikasta lähtöisin olevien suunnittelumallien, joita voidaan käyttää hyviksi havaittujen ja koeteltujen ratkaisujen kuvaamiseen, hyödyntämiseen turva-automaatiossa. Suunnittelumalleille kehitetty työkalutuki sekä avustaa suunnittelumallien käyttöä että automatisoi tarkastuksia ja koostaa dokumentaatiota suunnittelumallien käytöstä.

1 JOHDANTO

Turva-automaatiojärjestelmät ovat järjestelmiä, jotka toteuttavat ohjattavien, automatisoitavien prosessien turvallisuuden edellyttämät turvatoiminnot. Perinteisesti turva-automaatiossa on suosittu mm. mekaanisia ja sähköisiä, ei-ohjelmoitavia ratkaisuja, jotka ovat ohjelmistoja helpompia kehittää standardien mukaisesti. Teolliseen internetiin perustuva älykkäiden koneiden reaaliaikainen tiedonvaihto sekä mittaus- ja ohjausinformaation välitys mahdollistavat puhtaasti ohjelmistoihin perustuvat automaation hajautusratkaisut sekä prosessi- että liikkuvien koneiden automaatiossa. Teollisen internetin sovellusten ja niihin liittyvien prosessien turvallisuuden suunnittelu on kuitenkin aiempaa haastavampaa ja lisää paitsi mallinnuksen myös uudelleen käytön tarvetta kehityksessä. Ohjelmistojen käyttö turva-automaatiossa tuo mukanaan myös ohjelmistojen kehitykseen ja dokumentointiin liittyviä erikoispiirteitä.

Esimerkiksi sähköisistä järjestelmistä poiketen ohjelmistot eivät vikaannu satunnaisesti vaan niiden viat ovat systemaattisia ja voivat ilmetä, kun viallista ohjelmakoodia suoritetaan. Tilasto- ja todennäköisyyslaskennan sijaan ohjelmallisten turvatoimintojen kehityksessä korostuvatkin ohjelmistojen kehitysprosessit, kehityksessä käytettävät määrittely-, suunnittelu-, toteutus- ja testausmenetelmät sekä työn huolellinen dokumentointi. Standardien edellyttämiä tekniikoita käyttämällä pyritään välttämään systemaattisilta virheiltä ja takaamaan ohjelmistojen laatu sekä haluttujen turvallisuuden tasojen toteutuminen. Esimerkiksi keskeinen toiminnallisen turvallisuuden standardi, IEC 61508 /1/, esittää ohjelmallisesti toteutettaville osille ns. vesiputousmallia muistuttavan kehitysprosessin ja vaatimuksia prosessin eri vaiheissa tehtävälle työlle, tekniikoille ja tuloksille.

Ohjelmistojen laatua ja kehitysprosessia voidaan pyrkiä parantamaan myös järjestelmien mallinnukseen panostamalla. Mallipohjainen kehitys on viime vuosina laajalti tutkittu ohjelmistokehitysparadigma, joka käyttää malleja ohjelmistokehitysprosessin pääasiallisina suunnittelukohteina. Mallipohjaisen kehityksen graafiset mallit ovat tekstispesifikaatioita intuitiivisempia ja mielekkäämpiä kehittää. Standardoitujen kielten, kuten UML, mukaisia malleja voidaan käsitellä automaattisesti tuoden niihin tietoa muista malleista tai tietoformaateista tai generoiden esimerkiksi sovelluskoodia. Mallien voidaan nähdä toimivan myös kehitettävien järjestelmien dokumentaationa ja niistä voidaan koostaa tietoa erilaisiin tarpeisiin. Ohjelmallisten turva-automaation osien mallipohjainen kehitys edellyttää kuitenkin turvallisuusstandardien erikoispiirteiden huomioimista. Kehitystyössä tulisi olla mahdollista hyödyntää turva-automaatiokehitykseen soveltuvia ratkaisuja ja tekniikoita sekä tuottaa järjestelmiltä vaadittava turvallisuusdokumentaatio. Eräs keino näiden haasteiden saavuttamiseksi ovat suunnittelumallit, joilla ohjelmistotekniikassa kuvataan uudelleen käytettäviä ratkaisuja. Suunnittelun lisäksi – ja soveltuvaa työkalutukea hyödyntäen – ne voivat kuitenkin helpottaa myös dokumentaation tuottamista.

Tämä artikkeli käsittelee turva-automaation suunnittelumalleja sekä työkalutukea suunnittelumallien käytölle, jonka avulla voidaan helpottaa paitsi suunnittelumallien käyttöä myös niistä hyötymistä. Artikkelin perustuu Sulava-, ReUse- ja Ohjelmaturva-projekteissa tehtyyn työhön.

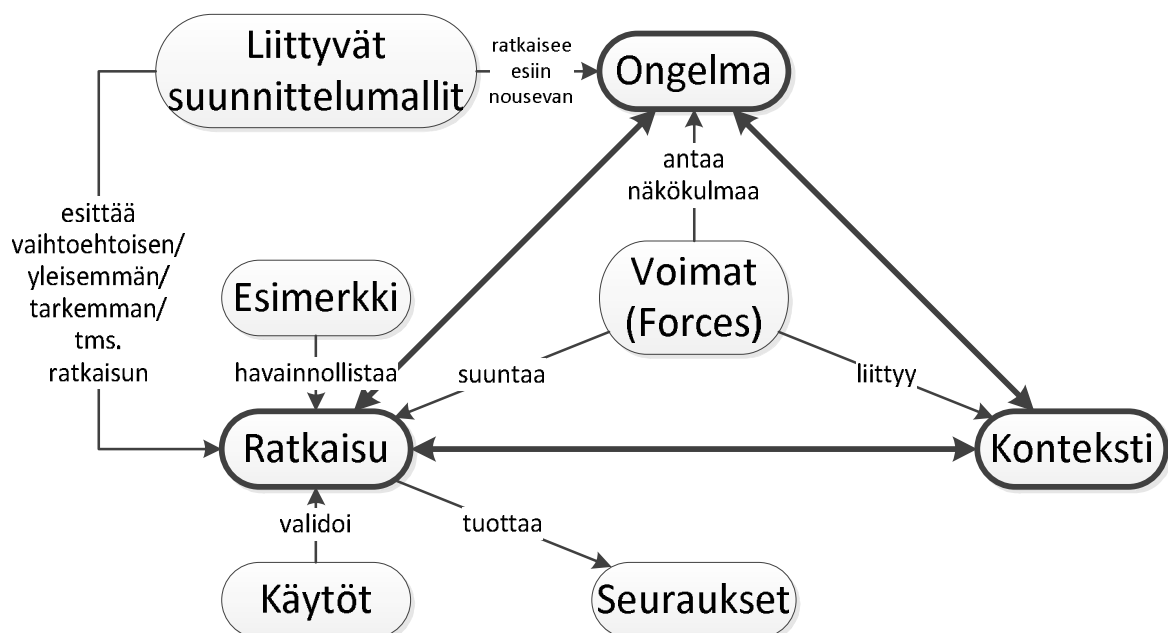
2 SUUNNITTELUMALLEISTA JA NIIDEN KEHITYKSESTÄ

Suunnittelumallit ovat dokumentoituja, määritellyissä konteksteissa käyttökelpoisia ratkaisuja suunnittelutyössä toistuviin haasteisiin. Suunnittelumalleihin liittyvä tutkimus sai alkunsa Christopher Alexanderin työstä /2, 3/, joka keskittyi rakennusarkkitehtuuriin ja jossa ajatuksena oli hyödyntää systemaattisesti hyviksi havaittuja ratkaisuja. Ohjelmistotekniikassa suunnittelumallit alkoivat herättää kiinnostusta ns. Gang of Four -kirjan /4/ julkaisun jälkeen, missä sovellusalueena oli ohjelmistotekniikka ja erityisesti olio-ohjelmointi. Myöhemmin suunnittelumalleja on kuitenkin julkaistu myös erikoissovellusalueille, kuten turva-automaatioon. Tukea suunnittelumallien käyttöön on kehitetty myös ohjelmistotekniikassa yleisesti käytettyyn UML-mallinnuskieleen.

TTY:n systeemitekniikan laitoksella tehty tutkimus pyrkii hyödyntämään suunnittelumalleja turva-automaation mallipohjaisessa kehityksessä eri kehitysvaiheissa ja eri näkökulmista. Sulava-, ReUse-R- ja Ohjelmaturva-projekteissa tehtyä työtä jatketaan yhteistyössä FIMA:n (Forum for Intelligent Machines) kanssa alkamassa olevassa DPSafe-hankkeessa (Suunnittelumallit ohjelmallisille turvatoiminnoille). Jo suoritettujen tutkimusten tuloksiin sisältyy paitsi turvallisuuteen liittyviä suunnittelumalleja myös tukea suunnittelumallien työkaluavusteiseen käyttöön ja suunnittelumallien hyödyntämiseen dokumentaation tuottamisessa. Turvallisuuskäsitteestä ajatellen työ pyrkii siis edistämään paitsi hyviksi havaittujen ratkaisujen käyttöä myös dokumentaation tuottamista vastaamaan kysymyksiin, missä ja millä tavoin kehitettävät järjestelmät hyödyntävät - standardien mukaisia ja muita soveltuvia - suunnittelumalleina kuvattuja ratkaisuja.

Systeemitekniikan laitoksella on viime vuosien aikana dokumentoitu joukko suunnittelumalleja turvallisuuden liittyvien järjestelmien kehitykseen /5, 6, 7, 8/, keskittyen turvallisuuteen liittyvien järjestelmien järjestelmä- ja ohjelmistotekniikkaan sekä toiminta- ja suunnitteluperiaatteisiin. Suunnittelumalleja on tähän asti etsitty nojautuen kirjallisuuslähteisiin kuten standardeihin, keskusteluihin yritysten edustajien kanssa ja havaintoihin olemassa olevista laitteista ja automaatio-ohjelmistoista. Ideaalisessa tilanteessa suunnittelumallit kannattaisi identifioida sertifioiduista, turvallisuuden liittyvistä järjestelmistä, mikä antaisi suunnittelumalleille vahvistusta.

Suunnittelumallien dokumentoinnissa käyttämämme rakenneosat on esitetty kuvassa 1, joka mukailee kanonista suunnittelumallimuotoa /9/. Suunnittelumallin ytimen muodostavat yleensä aina ongelma, konteksti ja ratkaisu. Lisäksi suunnittelumalliin usein dokumentoidaan voimat (forces), jotka esittävät ongelman ratkaisussa huomioon otettavia, mahdollisesti ristiriitaisia seikkoja ja ohjaavat valinnan juuri suunnittelumallissa esitettyyn ratkaisuun. Jokaisella ratkaisulla on seuraukset. Hyvin usein ratkaisulla on sekä positiivisia, että negatiivisia seurauksia. Suunnittelijan tehtävä on ottaa seuraukset huomioon suunnittelupäätösten tekemisessä. Ratkaisua voidaan havainnollistaa esimerkiksi, jonka ei tarvitse välttämättä olla tunnetusta käyttökohteesta. Sen sijaan (tunnetut) käytöt -kohta dokumentoi mielellään vähintään kolme tunnettua käyttökohdetta. Tunnetut käytöt vahvistavat



Kuva 1: Suunnittelumallin tyypilliset rakenneosat ja niiden väliset suhteet.

suunnittelumallissa esitetyn ratkaisun. Näiden lisäksi mainitaan usein vielä liittyvät suunnittelumallit, jotka kuvaavat esimerkiksi vaihtoehtoisia ratkaisuja tai suunnittelumalleja, jotka ratkaisevat esimerkiksi jonkin ratkaisun mukanaan tuoman uuden ongelman. Tarvittaessa suunnittelumallia voitaisiin laajentaa esimerkiksi tiedolla siitä, mille SIL- tai PL-tasolle IEC 61508 tai EN ISO 13849-1 puitteissa kyseinen ratkaisu soveltuu.

TTY:n systeemitekniikan laitoksen suunnittelumallikokoelma keskittyy (ohjelmallisten) turvatoimintojen suunnitteluun. Keskeisiä näkökulmia ovat muun muassa turvatoimintojen kehitysprosessi, riskinpienennysstrategiat, kehitysperiaatteet, järjestelmä- ja ohjelmistoarkkitehtuuri sekä turva- ja käyttöautomaation yhteistoiminta ja eriyttäminen. Yksittäiset suunnittelumallit kuuluvat pääsääntöisesti johonkin mainituista näkökulmista ja ne yhdessä muodostavat kokoelman.

Kokoelmasta on tietoisesti rajattu pois mm. vikasietoisuuteen liittyvät seikat, joiden suunnittelumalleja on tutkittu, dokumentoitu ja julkaistu varsin runsaasti. Tämä ei tietenkään tarkoita sitä, että vikasietoisuus ei kuuluisi osaksi turvatoimintoja toteuttavaa järjestelmää, päinvastoin. Vikasietoisuuden suunnittelumallit voidaan sitoa osaksi turvatoimintojen suunnittelumalleja määrittelemällä yhteyksiä kokoelmien ja niissä kuvattujen suunnittelumallien välille. Toinen luonnollinen rajapinta erityisesti ohjelmallisten turvatoimintojen suunnittelumalleille on hajautettujen ohjausjärjestelmien suunnittelumallit.

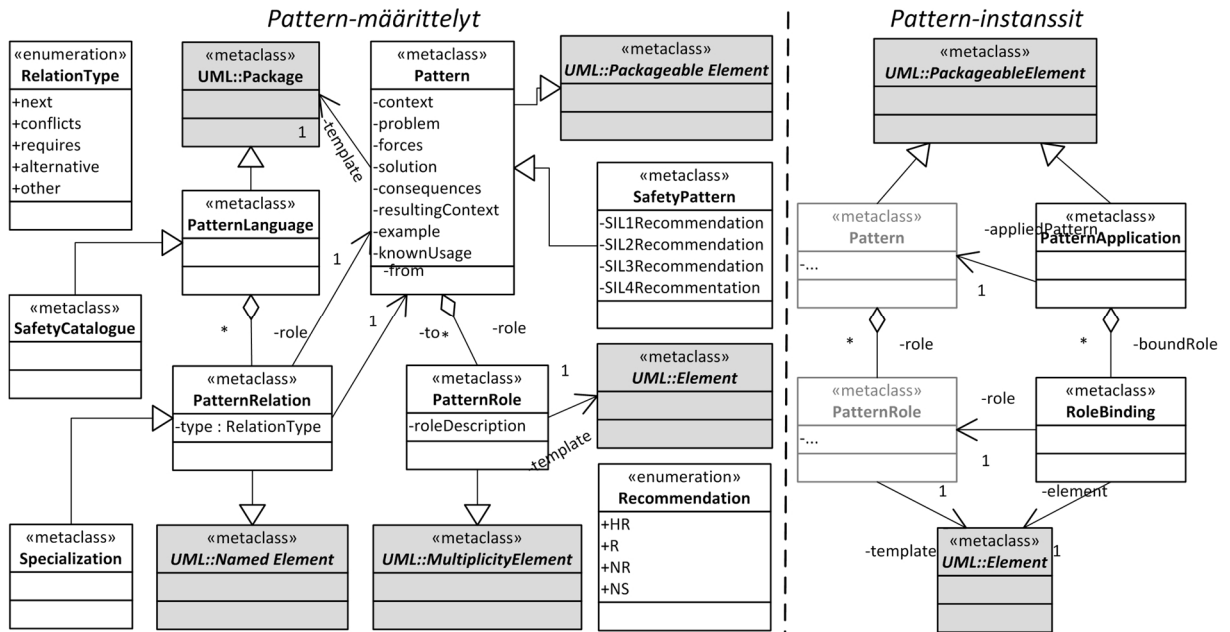
Suunnittelumallikokoelman yksittäisten mallien välille on määritelty yhteyksiä. Nykyisellään kokoelmassamme on noin 40 suunnittelumallia tai suunnittelumalliaihiota, joista osa on julkaistu. Yhteydet kuvaavat ensisijaisesti suunnittelumallien potentiaalista käyttöjärjestystä, eli ”tämän suunnittelumallin jälkeen kannattaa harkita seuraavia malleja”. Yhteyksiä voidaan myös erikoistaa määrittämällä esim. yhteys ’ratkaisee ongelman’, joka kuvaa, että suunnittelumalli B ratkaisee suunnittelumallin A esille tuoman ongelman tai B on erityistapaus A:n ratkaisusta johonkin tiettyyn ympäristöön sovellettaessa. Suunnittelumallien väliset yhteydet tukevat suunnittelijaa tarjoten mahdollisia etenemispolkukuja suunnittelumallikokoelman sisällä ja suunnittelumallikokoelmien välillä.

3 TYÖKALUTUKEA SUUNNITTELUMALLEILLE

Lähtökohta kehitetyissä työkaluavusteissa suunnittelumallien käyttöön on, että suunnittelumallin käyttö on aina suunnittelijan tietoinen ratkaisu. Jotta tietoa suunnittelumalleista voitaisiin myöhemmin hyödyntää, tulisi tieto niistä dokumentoida eksplisiittisesti kehitettäviin malleihin. Tämä ajatus vaikuttaisi olleen myös UML:n suunnittelumallikäsitteiden taustalla. UML:n tuki suunnittelumallien määrittelyyn (Collaboration) ja käyttöön (CollaborationUse) kuitenkin rajoittuu rakenteisiin luokkiin rajaten täten ulkopuolelle mm. UML:n pakkaukset ja komponentit, joita voitaisiin käyttää mm. arkkitehtuurisuunnittelumallien, kuten Layers /10/, kuvaamisessa. Myös monet muut suunnittelumallit, kuten Observer /4/, joudutaan esittämään kirjallisuudesta poikkeavassa muodossa /11/. UML:n suunnittelumallikäsitteistö soveltuu siis vain osalle tunnetuista suunnittelumalleista.

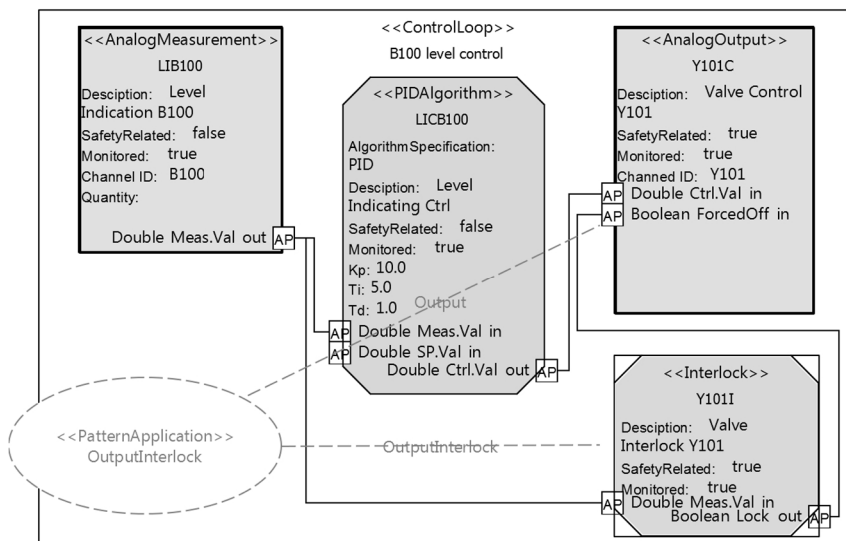
ReUse-hankkeessa kehitetty suunnittelumallien mallinnuskäsitteistö /11/, katso Kuva 1, on toteutettu kokeellisessa UML AP -työkalussa ja tarkoitettu mahdollistamaan suunnittelumallien joustava määrittely. Suunnittelumallit voivat koostua mistä tahansa rakenneosista ja ne voivat kuvata myös muiden kuin rakenteiden luokkien sisäisiä toteutuksia. Suunnittelumalleista määritellään niihin yleisesti tekstinä liittyvä tietosisältö (mm. konteksti ja ongelma), roolielementit lukumäärineen sekä roolikohtaiset mallielementit (template), joita voidaan käyttää automatisoitaessa ratkaisujen käyttöönottoa. Toisaalta käsitteet mahdollistavat suunnittelumalli- instanssien merkinnän niitä käytävissä malleissa ja niissä olevien mallinnuselementtien toteuttamien roolien määrittelyn. Käsitteistöön kuuluu myös turvallisuuteen liittyvä suunnittelumalli, SafetyPattern, joka sisältää samat perustiedot kuin tavallinen suunnittelumalli, mutta voi lisäksi sisältää tiedon ratkaisun suositeltavuudesta eri turvallisuuden eheystasoille (SIL). Luonnollisesti käsite eroaa tavallisista suunnittelumalleista (Pattern) myös siinä, että se on tarkoitettu turva-automaatioon soveltuviin suunnittelumallien kuvaamiseen.

Käytännössä suunnittelumallin määrittely sisältää Pattern-instanssin tekstitietoineen sekä rooli-elementteineen (PatternRole), jotka määritellään vastaamaan suunnittelumallikohtaisia rooleja. Suunnittelumallien määrittelyt sisältyvät pakkauksiin (Package) tai suunnittelumallikieliin (PatternLanguage). Suunnittelumalli-instansseja (PatternApplication) puolestaan käytetään merkitsemään käyttöön otettuja, yksittäisiä instansseja. Ne viittaavat suunnittelumallien, joiden instansseja ne ovat, määrittelyihin sekä määrittelyihin sisältyviin rooleihin. Malleissa säilyy täten tieto mm. siitä, mitä ja miten suunnittelumalleja on käytetty. Malleissa olevat merkinnät tehostavat dokumentointia indikoimalla standardiratkaisujen käyttöä, parantavat mallien ymmärrettävyyttä, tarkentavat elementtien roolia suunnittelussa ja kertovat myös mahdollisista suunnitteluhaasteista, joita suunnittelumallien käytöllä on pyritty ratkaisemaan. Osa tästä tiedosta on luonteeltaan sellaista, että sen säilyminen malleissa, jotka tyypillisesti kuvaavat ennemmin ratkaisuja kuin niiden taustoja, olisi muuten hankalaa.



Kuva 1 Suunnittelumallien ja suunnittelumalli-instanssien määrittelyssä käytettävät mallinnuskäsitteet.

Suunnittelumallien ja mallinnuskäsitteistön lisäksi TTY:llä on kehitetty työkaluja suunnittelumallien käyttöönottoon ja visualisointiin kaavioissa sekä malleista hyötymiseen. Koska suunnittelumallien käyttö tulisi merkitä malleissa, mikä on ylimääräinen tehtävä esimerkiksi rakenteisten mallien kopiointiin nähden (mikä sisältyy myös joihinkin kaupallisiin työkaluihin), on luonnollista, että tehtävä on työkaluavusteinen. Suunnittelumallien käyttöönotto tapahtuu työkaluympäristössä avustesovelluksen (wizard) avulla. Avuste mahdollistaa suunnittelumallien käyttöönoton käyden siihen liittyvät roolit läpi joko kopioiden mallielementtejä suunnittelumallin määrittelystä tai asettaen malleissa jo olevia elementtejä suunnittelumallikohtaisiin rooleihin. Suunnittelumalli-instanssien visualisoinnissa käytetään notaatiota, jossa suunnittelumalli-instanssit kuvataan ellipseinä, joihin eri rooleissa toimivat elementit kytketään relaatio-viivojen avulla. Visualisointia voidaan hyödyntää esimerkiksi dokumentoinnin tukena kopioitaessa kaavioita dokumentteihin ja suunnittelumallien opiskelussa visualisoimalla, miten muut suunnittelijat ovat käyttäneet suunnittelumalleja. Kuva 2 esittää esimerkin Output Interlock -suunnittelumalli-instanssin visualisoinnista. Vasemmassa alareunassa oleva ellipsi kuvaa suunnittelumalli-instanssia, johon liittyvissä rooleissa toimivat elementit ilmenevät ellipsistä lähtevistä katkoviivoista. Esimerkissä siis Y101 on lukittava ohjaus (Output) ja lukitus Y101I vastaava lukitus (OutputInterlock).



Kuva 2 Esimerkki Output Interlock -suunnittelumallin /11/ visualisoinnista.

Sisällyttämällä suunnittelumalleihin turvallisuuteen liittyvää tietoa mm. ratkaisujen standardien mukaisuudesta, voidaan suunnittelumalleja ja malli-instansseja hyödyntää myös turvallisuuskirjallisuuden kokoamisessa. Kehitetty suunnittelumallikäsitelistö (Kuva 1) mahdollistaa turvallisuuteen liittyvien suunnittelumallien erottamisen muista (oma mallinnuselementtityyppi) sekä mallien eri turvallisuuden eheystasolle (SIL-tasot) suositeltavuuden määrittelyn. Malleissa käytettäviä suunnittelumalli-instansseja ja niiden suosituksia SIL-tasolle voidaan verrata toisiinsa ja muuhun mallinnettuun tietoon ja näin automatisoida malleille suoritettavia oikeellisuustarkasteluja. Esimerkiksi arkkitehtuuriratkaisun, jota käytettävä standardi ei mainitse, hyödyntäminen turvallisuuteen liittyvässä järjestelmässä ei välttämättä ole suoranainen virhe, mutta järkevä varoituskohde, mikäli ratkaisulle olisi olemassa suositeltu, korvaava vaihtoehto.

Kuva 3 sisältää esimerkin mallinnuskäsitelstön mahdollistamasta tarkastelusta. Kuvan jäljitettävyyssmatriisi on koostettu automaattisesti esimerkimmallista lähtemällä liikkeelle mallinnetuista vaatimuksista (Requirement) ja niihin sisältyvistä turvallisuuden eheystason määrittämisestä tarkenteista. Vaatimukset on johdettu esimerkimmallin pakkauksiin (Package), joihin (tai joiden sisältämiin elementteihin) vaatimukset on mallissa yhdistetty jäljitysrelaatioiden avulla. Eheystasovaatimukset pakkauksille määräytyvät vastaavista vaatimusten eheystasoista siten, että taso vastaa korkeinta vaatimukseen liittyvää tasoa. Pakkaukset on puolestaan matriisissa johdettu suunnittelumalleihin, joita kyseisissä pakkauksissa käytetään. Matriisi esittää värikoodauksella, ovatko käytetyt suunnittelumallit suositeltuja (vihreä) tai ei-suositeltuja (punainen) niille eheystasolle, joita kyseisiltä pakkauksilta vaaditaan. Kuvan esimerkissä kaikki käytetyt suunnittelumallit ovat suositeltavia (R) vaaditulle eheystasolle SIL1.

REQUIREMENT:							
P100 Protection (SIL1)	↘						
P100IR (SIL1)		↘					
PACKAGE:	Software Safety Requirements (SIL1)	Control Structures (SIL1)	PATTERN:	SIL1	SIL2	SIL3	SIL4
				R	R	R	R
				R	R	HR	HR
				R	R	HR	HR
				R	R	HR	HR

Kuva 3 Esimerkki jäljitettävyyssmatriisista, joka on tuotettu mallinnuskäsitteitä hyödyntämällä.

Kuvan 3 matriisi ei kuitenkaan vielä vastaa kysymykseen, mitä suunnittelumalleja kussakin kehitysvaiheessa tulisi käyttää. Artikkelissa /11/ esitettyllä tavalla toisiinsa liittyviä suunnittelumalleja voidaan koota suunnittelumallikatalogeiksi, joihin sovelluksia voidaan verrata. Esittämällä esimerkiksi standardin tiettyyn suunnitteluvaiheeseen suosittelemat ratkaisut katalogiin kuuluvina suunnittelumalleina ja vertaamalla mallista löytyviä instansseja katalogiin, olisi mahdollista tuottaa suosituksia suunnittelumallien käyttöön. Toisaalta näin voitaisiin tuottaa kooste, mitä suositelluista ratkaisuksista ja tekniikoista on käytetty. Tällainen kooste voisi olla hyödyllinen esimerkiksi järjestelmän standardinmukaisuutta tarkistettaessa. Kuvassa 3 olevan matriisin tapaan koosteet voidaan tuottaa esimerkiksi erillisiin MS Excel-taulukoihin.

4 YHTEENVETO JA JOHTOPÄÄTÖKSET

Suunnittelumallit ovat tapa taltioida suunnittelua ja hyviksi havaittuja ratkaisuja uudelleenkäytettäväksi. Kukin suunnittelumalli dokumentoi tiettyyn kontekstiin soveltuvan ratkaisun suunnittelutyössä toistuvaan haasteeseen. Suunnittelumalleja voidaan myös kerätä kokoelmiin, joissa mallien välillä voi olla esimerkiksi potentiaaliseen käyttöjärjestykseen perustuvia yhteyksiä.

Ohjelmallisesti toteutettavien turvatoimintojen kehityksessä korostuvat kehitysprosessit sekä kehityksessä käytettävät määrittely-, suunnittelu-, toteutus- ja testausmenetelmät sekä työn huolellinen dokumentointi. Esimerkiksi toiminnallisen turvallisuuden standardi IEC 61508 /1/ jakaa ohjelmistokehityksen vaiheisiin, joihin esitetään suosituksia käytettäville menetelmille ja ratkaisuille. Eräs tapa mallintaa standardien mukaisia ja muita

turva-automaatioissa toistuvia, hyviksi havaittuja ratkaisuja on kuvata ne suunnittelumalleina. Monet standardin esittämistä ratkaisuista – mm. arkkitehtuurisuunnitteluun liittyen – sisältyvätkin jo aihepiiriin kirjallisuuteen. Esimerkiksi standardin suosittamia redundanssi- ja (virheistä) toipumISRatkaisuja on kuvattu julkaisuissa /12/ ja /13/.

TTY:n systeemitekniikan laitoksella tehdyssä tutkimuksessa on paitsi koottu ja kehitetty turva-automaation suunnittelumalleja myös kehitetty työkalutukea niiden käytölle ja niistä hyötymiseen. Suunnittelumallit ovat keskittyneet turvallisuuteen liittyvien järjestelmien arkkitehtuuriin ja periaatteisiin. Samalla on tutkittu ja kehitetty työkalutukea suunnittelumallien hyödyntämiseen mallipohjaisessa kehitysprosessissa käyttäen lähtökohtana kokeellisesta UML AP -työkalua.

Työkalutuki suunnittelumalleille perustuu suunnittelumallien joustavan mallinnuksen mahdollistavaan käsitteistöön ja sisältää avustusta suunnittelumallien käyttöönottoon ja merkintöjen perusteella tehtäviin oikeellisuustarkastuksiin. Tarkastukset huomioivat mm. ratkaisujen suositeltavuuden eri turvallisuuden eheystasoille sekä sovelluksilta vaaditut tasot. Työkalut edistävät myös ratkaisujen jäljitettävyyttä ja mm. suunnittelun ymmärrettävyyttä mahdollistamalla standardiratkaisujen korostamisen. Merkintöjen perusteella voidaan tuottaa automaattisesti mm. jäljitettävyydsmatriiseja.

5 Kirjallisuusluettelo

1. International Electrotechnical Committee. IEC 61508: functional safety of electrical/electronic/programmable electronic safety-related systems. Parts 1-7. 2010.
2. Alexander, C., Ishikawa, S. and Silverstein, M. A Pattern Language: Towns, Buildings, Construction. Oxford University Press. 1977.
3. Alexander, C. The timeless way of building. Oxford University Press. 1979.
4. Gamma, E., Helm, R., Johnson, R. and Vlissides, J. Design patterns: elements of reusable object-oriented software. Addison-Wesley. 1994.
5. Rauhamäki, J., Vepsäläinen, T. and Kuikka, S. Patterns for safety and control system cooperation. VikingPLoP 2013 Conference. 2013.
6. Rauhamäki, J., Kuikka, S. Patterns for Hazard Mitigation Process. Presented in workshop of the EuroPLoP 2014 Conference, 9.-13.7.2014, Irsee, Germany. Proceedings version pending. 2014.
7. Rauhamäki, J., Kuikka, S. Patterns for Sharing Safety System Operation Responsibilities between Humans and Machines. VikingPLoP 2013 Conference, 10.-13.4.2014, Vihula, Estonia. ACM, New York, NY, USA, ISBN 978-1-4503-2660-5. pp. 68-74. 2014.
8. Rauhamäki, J., Kuikka, S. Patterns for control system safety. Presented in workshop of the EuroPLoP 2013 Conference, 10.-14.7.2013, Irsee, Germany. Proceedings pending. 2013.
9. Canonical Form. <http://c2.com/cgi/wiki?CanonicalForm>. Viitattu 12.2.2015.
10. Buschmann, F. Pattern Oriented Software Architecture: A System of Patters. Ashish Raut. 1999.
11. Vepsäläinen, T. and Kuikka, S. Safety Patterns in Model-Driven Development. 9th International Conference on Software Engineering Advances. October 12 - 16, 2014 - Nice, France.
12. Douglass, B.P. Real-Time UML: Developing Efficient Objects for Embedded Systems. Addison-Wesley. 1998.
13. Hanmer, R. Patterns for Fault Tolerant Software. John Wiley& Sons. 2013.