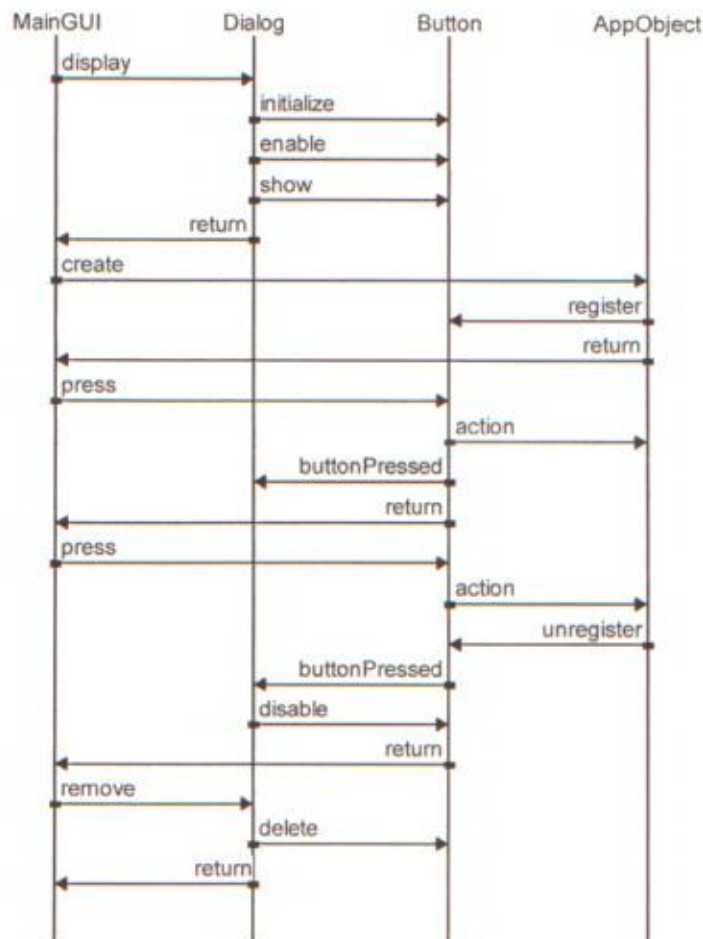


Ohjelmistoarkkitehtuurit

Loppukoe 20.11.2007

Kaikista tehtävistä voi saada max 6 pistettä. Tehtävä 6 on vierailuluentoja koskeva jokeritehtävä: myös siitä voi saada 6 pistettä, mutta kokeen kokonaispistemäärä on kuitenkin max 30 pistettä. Jokeritehtävällä voi näin korvata muissa tehtävissä mahdollisesti tapahtunutta pistehävikkiä.

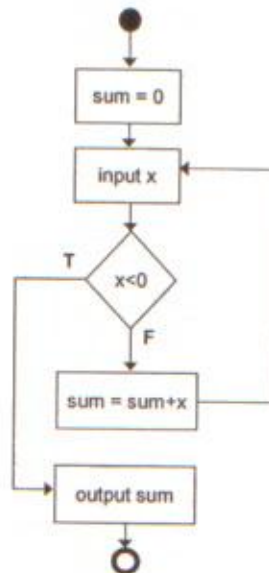
1. Alla oleva sekvenssikaavio kuvaa erään tyyppillisen vuorovaikutuksen, johon komponentti Button osallistuu (kaikkia paluunuolia ei ole merkitty näkyviin). Päättele tämän perusteella sopivat roolirajapinnat komponentille Button. Anna UML:n komponenttikaavio, jossa on komponentti Button, sen tarjoamat ja vaativat rajapinnat, sekä näitä käyttävät muut komponentit.



2. Oletetaan, että olet suunnittelemassa visuaalista komponenttityökalua, jonka avulla olemassa olevista komponenteista voidaan muodostaa uusia sovelluksia liittämällä komponenttien kuvakkeita yhteen graafisesti. Yhteen liittäminen tarkoittaa, että yhden komponentin aiheuttamaan tapahtumaan reagoidaan kutsumalla toisen komponentin jotakin operaatiota. Mitä suunnittelumalleja käytät tällaisen komponenttien yhteenliittämisen toteuttamiseen? Kuvaa lyhyesti nämä suunnittelumallit (kuva +selitys).

KÄÄNNÄ

3. Yksinkertaisia kontrollivuokaavioita voidaan piirtää graafisesti käyttämällä toimintaelementtejä (suorakaide, jonka sisällä on tekstuaalinen toimintakuvaus), haarautumaelementtejä (vinoneliö, jonka sisällä on ehtolauseke), ja niiden välisiä kontrollisiirtymiä (nuoli). Jokaisesta toimintaelementistä lähtee täsmälleen yksi kontrollisiirtymä, mutta sellaiseen voi tulla useita (vähintään yksi) siirtymiä. Haarautumaelementistä lähtee täsmälleen kaksi kontrollisiirtymää, jotka on varustettu symboleilla T(true) ja F (false). Haarautumaelementtiin voi tulla useita (vähintään yksi) siirtymiä. Lisäksi notaatiossa on täsmälleen yksi alkusymboli (musta pallo), josta lähtee täsmälleen yksi siirtymä eikä tule yhtään, ja nolla tai useampi loppusymboli (pieni ympyrä), johon voi tulla useita (vähintään yksi) siirtymä mutta ei lähde yhtään. Anna tälle kontrollin kuvauskielille metamalli. Kielen tekstuaalisia osia (toiminta- ja ehtotekstit) ei tarvitse tarkentaa metamallissa, vaan ne voidaan esittää String-tyyppisinä attribuutteina. Alla on esimerkki kuvauskielen käytöstä.



4. Eräessä tulkkiaarkkitehtuuriin perustuvassa ESB-järjestelmässä annetaan tiettyyn XML-dokumenttityyppiin liittyvä liiketoimintaprosessin kuvaus seuraavalla syntaksilla:

```

Process ::= "PROCESS" Identifier ":" ArtifactSpec "DO" Activity { ";" Activity } "END"
ArtifactSpec ::= Identifier "FROM" Identifier | "ANY"
Activity ::= "ACTIVITY" Identifier "DO" { DocumentDecl ";" } Action { ";" Action } "END"
DocumentDecl ::= Identifier ":" Identifier
Action ::= "SUBMIT" Identifier | "LOG" Identifier | Identifier
           "IF" Cond "THEN" Action { ";" Action } "END | Identifier "=" Identifier "(" Identifier ")"
  
```

Tässä aaltosulut tarkoittavat toistoa n kertaa (n=0,1, ...) ja pystyviiva tarkoittaa vaihtoehtoista rakennetta. Rakennesymbolit Cond ja Expr tarkoittavat tässä tarkentamattomia lausekkeita. Esimerkkikuvaus voisi olla seuraavanlainen:

```

PROCESS Doc: Order FROM ClientOffice DO
  ACTIVITY CheckClient DO
    DOCUMENT CreditResult: CRSchema; StorageReport: SRSchema;
    IF Doc.Type /= "TRUSTED" THEN
      CreditResult = CreditChecker(Doc);
      SUBMIT CreditResult; LOG CreditResult
  
```

```

        END;
        SUBMIT CheckStorage(Doc);
        IF CreditResult.Status = "OK" THEN HandleOrder END
    END;
    ACTIVITY HandleOrder DO
        ...
    END
END

```

Kuvauksen tarkka semantiikka ei ole tässä olennaista. Anna prosessikuvauksen olioesityksen rakennemalli UML:n luokkakaaviona. Esiintyykö ratkaisussasi Tulkki-suunnittelumallin ilmentymä jossain muodossa? Merkitse mahdollinen ilmentymä luokkakaavioon.

5. Suunnittele ohjelmistoarkkitehtuuri auton kiinteän navigaatiojärjestelmän ohjausyksikölle. Järjestelmään kuuluu laitteina ohjausyksikön lisäksi GPS-vastaanotin, näyttö, muistikortin lukija, näppäimistö, kaiutin ja mikrofoni. Kaikki laitteet liitetään ohjausyksikköön. Järjestelmä käyttää muistikortilla olevaa kartastokantaa. Järjestelmällä on seuraavat keskeiset toiminnalliset vaatimukset:

- kohteiden määrittäminen
- kohdeluettelon ylläpito
- kohteeseen ohjaaminen (ääni+grafiikka)
- kohteen antaminen äänikomennolla
- auto ilmoittaa pyydettyä paikkansa huoltopisteelle (esim. varkaustapauksissa)

Järjestelmän tärkeimmät laatuvaatimukset ovat käytettävyyden lisäksi suorituskyky ja muunneltavuus (esim. laitteiden vaihtuminen). Ohjausjärjestelmän on pysyttävä periaatteessa toiminnassa, vaikka jokin oheislaitte (esim. mikrofoni) ja/tai sen ajuri poistetaan järjestelmästä. Tulevaisuudessa on suunniteltu laajennosta, jossa kohteen voisi lähettää järjestelmään ulkopuolelta (esim. kännykällä), ja jossa auto ilmoittaa pyydettyä paikkansa huoltopisteelle (esim. varkaustapauksissa). Myös muiden oheislaitteiden liittämistä järjestelmään tulevaisuudessa on ennakoitu. Kaikilla oheislaitteilla voidaan olettaa olevan oma laitekohtainen ajurinsa. Järjestelmän tulee huolehtia mm. puhesyntetisoinnista ja -tunnistuksesta.

Tehtävät:

- a) Mitä arkkitehtuurityyliä (tai -tyylejä) sovellat? Miksi?
 - b) Anna ohjausjärjestelmän ohjelmistoarkkitehtuurin kuvaus komponenttikaaviona.
 - c) Mitkä ratkaisut ohjelmistoarkkitehtuurissasi tukevat muunneltavuutta? Entä suorituskykyä?
 - d) Arvioi lyhyesti miten ratkaisusi tukee yllä kuvattuja mahdollisia tulevaisuuden skenaarioita.
6. Jokeritehtävä:
- a) Selosta lyhyesti käsitteet reaktiivinen agentti, tietoinen agentti ja hybridi agentti. Anna esimerkki kustakin. (3p)
 - b) Yrityksessä voi usein käydä niin, että järjestelmän arkkitehtuuri häviää: kukaan ei enää tiedä, mikä on järjestelmän arkkitehtuuri. Mitkä syyt voivat johtaa tähän, ja miten arkkitehtuuri voidaan rekonstruoida? (3p)