

Tarkvaraarhitektuuri dokumenteerimise nõuded

- [1. Sissejuhatus](#)
 - [1.1. Terminoloogia](#)
- [2. Äriarhitektuur](#)
 - [2.1. Andmearhitektuur](#)
 - [2.1.1. Äriandmearhitektuuri näide](#)
- [3. Süsteemi arhitektuur](#)
 - [3.1. Süsteemi loogiline andmearhitektuur](#)
 - [3.2. Loogilise andmearhitektuuri näide](#)
 - [3.3. Süsteemi füüsiline andmearhitektuur](#)
 - [3.5. Rakendusliidesed](#)
 - [3.6. Pakutavad ja vajatavad liidesed/teenused](#)
 - [3.7. Näide 1](#)
 - [3.8. Näide 2](#)
 - [3.9. Rakenduse isendite kirjeldused \(a.k.a füüsiline arhitektuur\)](#)
 - [3.9.1. Näide](#)

1. Sissejuhatus

1.1. Terminoloogia

Eestikeelne termin	Eestikeelne seletus	Ingliskeelne termin	Ingliskeelne seletus	Viide standardile
rakendus		application		
süsteem		system		
liides		interface		
protokoll		protocol		
komponent / osis		component		
viiteaeg		latency		
edastusmaht		bandwidth		
isend		instance		

2. Äriarhitektuur

2.1. Andmearhitektuur

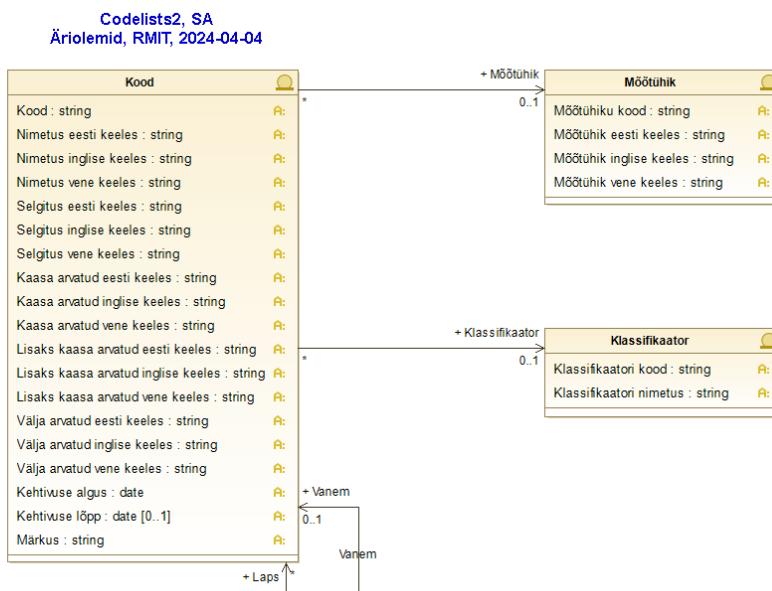
Äriline andmearhitektuur annab ärilise info selle kohta, millised andmed on äriprosessis kasutusel. Äriline andmearhitektuur peaks vastama küsimustele:

1. Mida üks või teine sõna tähendab (st mõistete kirjeldused)?
2. Millised andmed on tööprosesside sammudes vajalikud?
3. Millised andmed tekivad tööprosesside sammudes?
4. Kuidas on seotud dokumentides või andmestikes olevad andmekirjad?
5. Mis on erinevate äriandmete olemiste struktuur/sisu?

Äriline andmearhitektuur kirjeldatakse UML klassimodelina. See tähendab, et:

1. Äriolemid modelleeritakse **UML klassidena**, millel on Togaf stereotüüp **<>business entity**.
2. Äriolemite vahelised seosed modelleeritakse **UML sidemetena**. Seose paljusus kirjeldatakse **UML võimsustikuna** ning **UML roll** määrab, kuidas viidatakse olemis viidatud olemile.
3. Äriolemite sisu modelleeritakse **UML atribuutidena**.

2.1.1. Äriandmearhitektuuri näide



Ärilise andmearhitektuuri andmemudeli skeemi näidis.

3. Süsteemi arhitektuur

Süsteemiarhitektuuri kirjeldamist alustame rakenduse enda kirjeldamisega. Rakenduse mudeli eelistatud autor on rakenduse autor.

Rakenduse enda kirjeldamise järel tuleb ära kirjeldada rakenduse isendid ehk rakenduse erinevad kasutusjuhud. Selle käigus kirjeldatakse ära rakenduste serverite nõuded ja loogilised keskkonnad.

Piirangud arhitektuuri kirjeldamisel

Rakenduse arhitektuur viitab mitmes kohas äriarhitektuurile ja tehnilisele arhitektuurile. Kui äriarhitektuuris või tehnilises arhitektuuris on puudusi, siis tuleb need vastava arhitektuuri taseme eest vastuvatele isikutele välja tuua.

Kuni puuduste kõrvaldamiseni võib puuduvaaid elemente kirjeldada ka süsteemiarhitektuuri tasemel või jäätta arhitektuuri juurde märkus, et arhitektuuri kirjelduses ei ole kõike kirjeldatud ning mudel vajab täiendamist esimesel võimalusel.

3.1. Süsteemi loogiline andmearhitektuur

Süsteemi loogiline andmearhitektuur kirjeldab andmebaaside struktuure, andmeliidestes liikuvate andmestike struktuure, nende seoseid äriandmetega ning andmete seoseid ärialiste nõuetega (nt viited seadustele, mis reguleerivad andmete säilitusaega jms). Süsteemi loogilise arhitektuuri juures saab soovi korral kirjeldada ka vastavate andmete andmekaitsenõudeid. Loogiline mudel kirjeldab ära ka andmekirjetele rakendatavad piirangud (nt pikkus, kuju jms). NB! Need piirangud ei sisalda füüsilise andmemudeli spetsifikast lisanduvaid andmeid (nt lõpu nullbait sõnedel multibait andmekujust tingitud mahukasv jms).

Süsteemi loogilise andmearhitektuuri andmeolemeid modelleeritakse **UML klassidena**. Andmeliidese puhul lisatakse klassile vastavalt liigile Togaf stereotüüp:

- **<<Service Data>>** - teenuse andmeolem (nt sõnumi juurelement või sõnumi lihttüüp/
UML standardandmetüüp).
- **<<Message fragment>>** - teenuse andmeolemi alamolem.

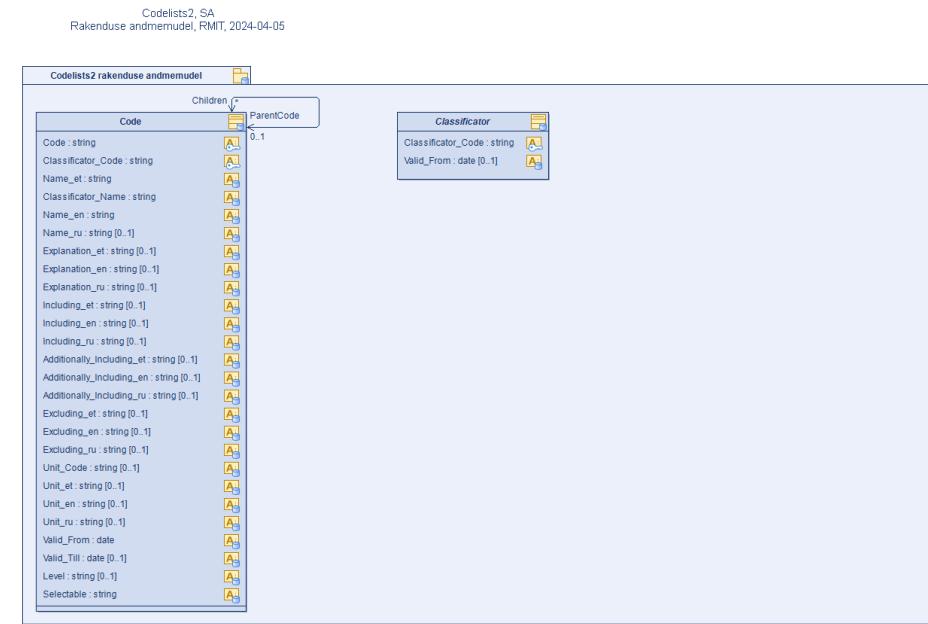
Teenuste andmeliidestest andmeolemid seostatakse modelleerimisel **UML kompositsooniga**. Andmebaasi andmeolemid seostatakse **UML sidemetena**. Seose paljusus kirjeldatakse **UML võimsustikuna** ning **UML roll** määrab, kuidas viidatakse olemis viidatud olemile. **Seose nimi** peaks vastama realtsionilise andmebaasimudeli puhul välisvõtme nimele.
Andmeolemites sisu kirjeldatakse **UML artibuutidena**.
Andmebaasiskeemid modelleeritakse **UML pakettidega**.

Äriolemid ja nende atribuudid seostatakse modelleerimisel loogilise andmemudeliga läbi **UML sõltuvuse**, millel on Togaf stereotüüp **<<migrates>>**. Sõltuvus on äriolemist loogilisse olemisse.

Eelnimetatud sõltuvus on vajalik terminoloogilise järgepidevuse saavutamiseks (st et ärilised mõisted läheksid kokku administraatorite ja arendajate nähtavate nimetustega).

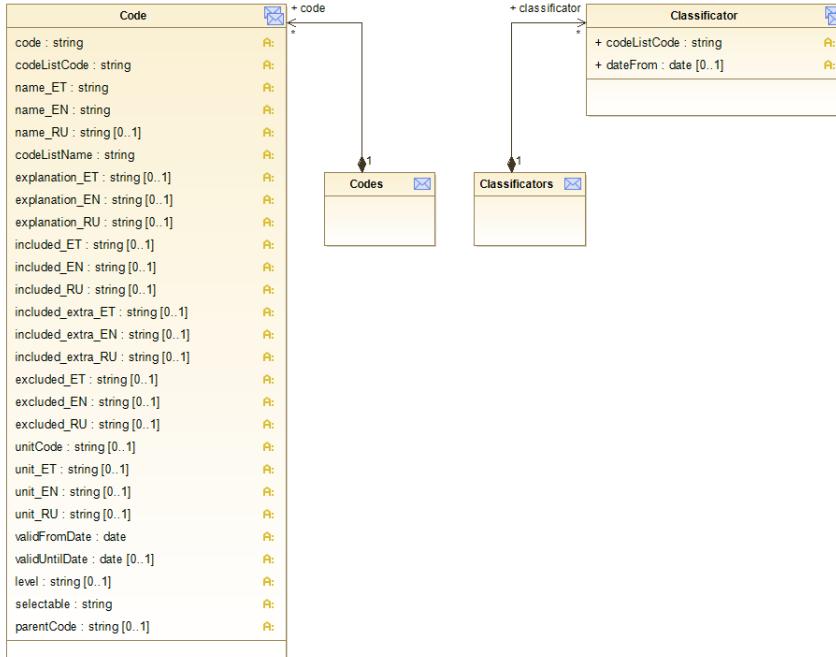
Loogiliste olemite vahelised andmete liikumised modelleeritakse läbi **UML sõltuvuse**, millel on Togaf stereotüüp <> .

3.2. Loogilise andmearhitektuuri näide

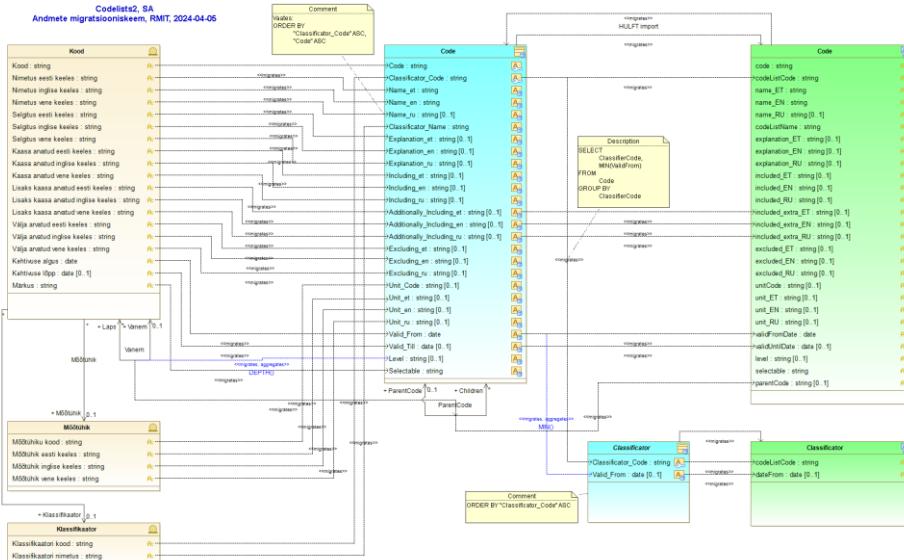


Loogilise andmebaasiskeemi näide.

Codelists2, SA
Teenuste andmemudel, RMIT, 2024-04-05



Loogilise andmeliidese



andmemudeli näidis.

Andemte migratsiooniskeemi näidis.

3.3. Süsteemi füüsiline andmearhitektuur

Süsteemi füüsiline andmearhitektuur kirjeldatakse kasutades selleks sobivat keelt. Levinud füüsilise andmearhitektuuri kirjeldamise keeled on näiteks: OpenAPI, WSDL, JSON Schema, XML Schema. Füüslist andmearhitektuuri ei modelleerita.

Füüsiline andmearhitektuuri seostamine loogilise andmearhitektuuriga käib läbi olemite ja atribuutide nimetuste.

3.5. Rakendusliidesed

Rakendusliidesed kirjeldavad, millised operatsioonid on teenuse poolt pakutud ja millised andmed (loogiline andmemudel) liiguvalt liidese vahendusel.

Rakenduse rakendusliidesed (*API*) modelleeritakse kui **UML liides**, millele on antud Togaf stereotüüp `<<IS service>>`.

Liidese pakutavad operatsioonid modelleeritakse kasutades **UML operatsiooni**, millele on lisatud Togaf stereotüüp `<<IS service operation>>`. Operatsiooni definitsioon peab viitama loogilises andmemudelis kirjeldatud andmemudelitele või standardandmetüüpidele.

Rakenduste kohustuslikud sõltuvused teistest rakendustest modelleeritakse **UML seosega**, millel on UML stereotüüp `<<use>>`.

3.6. Pakutavad ja vajatavad liidesed/teenused

Pakutavad ja vajatavad liidesed kirjeldavad, milliseid ühendusi saab rakenduse suunas teha ja milliseid ühendusi rakendus vajab?

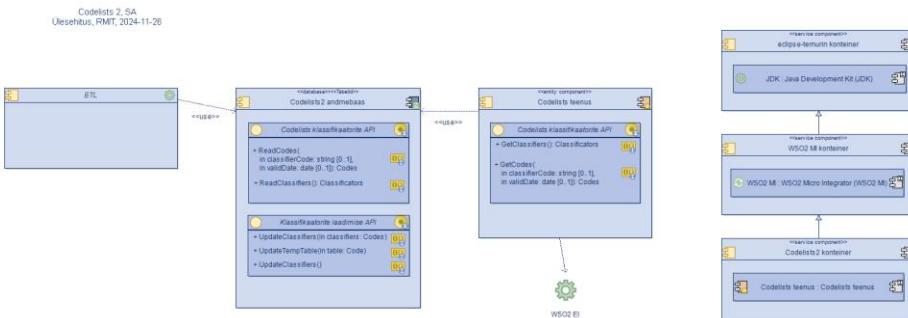
Pakutavad liidesed modelleeritakse kasutades **UML porti**, millele on antud Togaf stereotüüp `<<provided>>`. Pakutav liides/teenus seotakse rakendusliidesega **UML port** omaduse *Represented feature* kaudu.

Vajatavad liidesed modelleeritakse kasutades **UML porti**, millele on antud Togaf stereotüüp `<<required>>`. Vajatava liideses saab siduda rakendusliidesega läbi **UML port** omaduse *Represented feature*. Vajatav liides seotakse liidese pakkujaga **UML konnektoriga**.

Nii pakutavat kui vajatavat liidest saab tõpsustada stereotüüpidega:

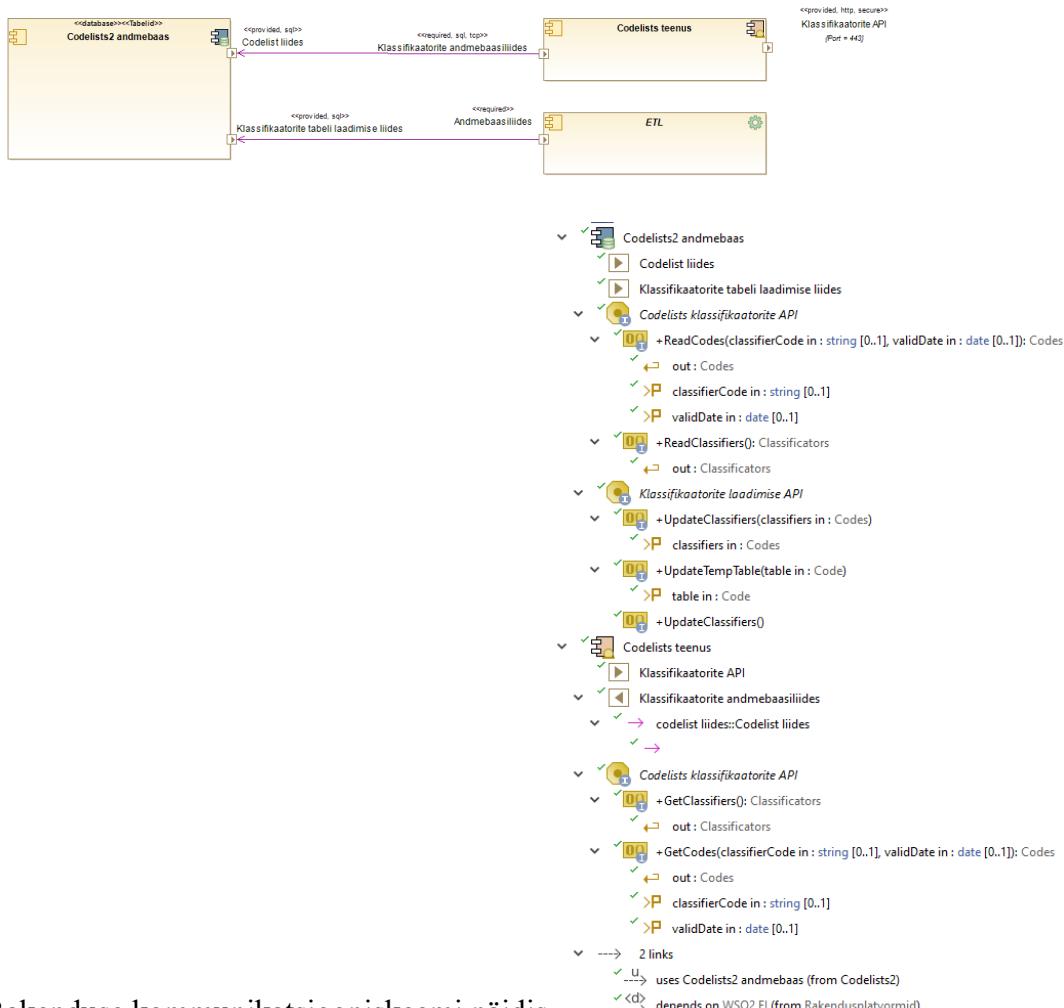
- `<<http>>` - tegemist on HTTP protokolli kasutava liidesega.
- `<<sql>>` - tegemist on SQL keelt kasutava liidesega.
- `<<tcp>>` - tegemist on TCP protokolli kasutava liidesega.
- `<<udp>>` - tegemist on UDP protokolli kasutava liidesega.
- `<<secure>>` - liides on turvatud (nt kasutusel on TLS või muu krüptograafiline lahendus).

3.7. Näide 1



Rakenduse ülesehituse skeemi näidis.

Codelists 2, SA
Kommunikatsiooniskeem, RMIT, 2024-02-01



Rakenduse kommunikatsiooniskeemi näidis

Rakenduse mudeli näidis.

Siiin näites on kirjeldatud põhjalikumalt kaks rakendust: Codelists2 andmebaas ja Codelists teenus.

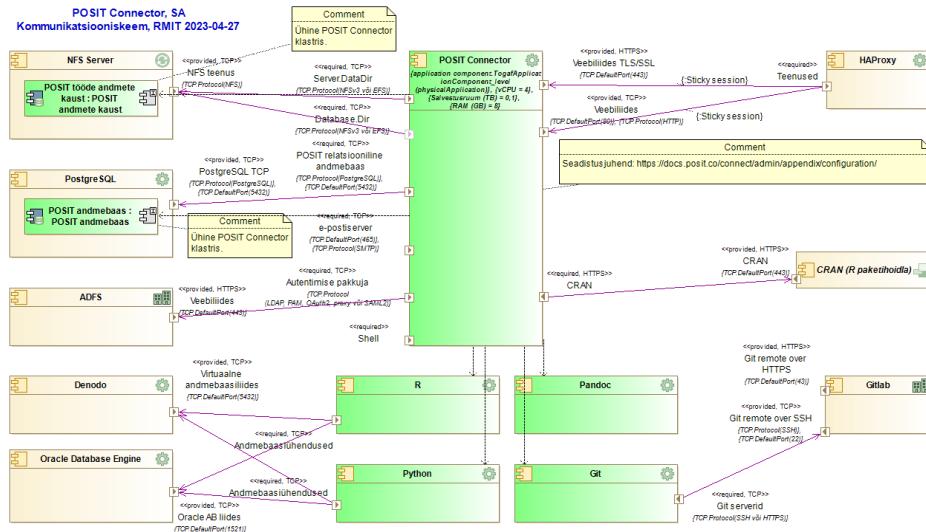
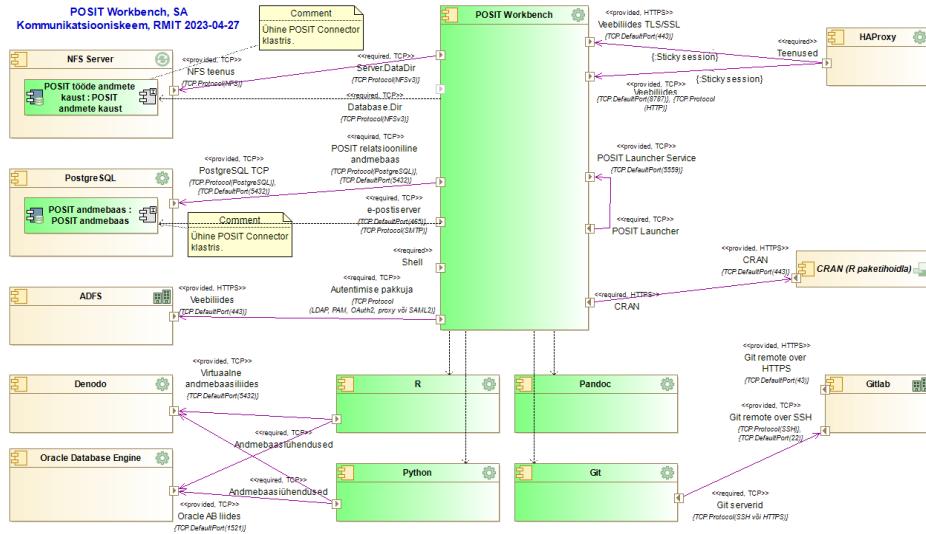
Codelists2 andmebaas on andmebasiteenus (Togaf stereotüüp <<database>>), mida on täpsustatud stereotüübiga <<tabel>> kuna see koosneb tabelitest. Antud juhul ei ole kirjeldatud alamstruktuuri (skeemid). Küll aga on kirjeldatud, et andmebaas pakub kahte liidest: Codelists klassifikaatorite API ja Klassifikaatorite laadimise API. Esimene neist pakub kahte operatsiooni, viimane kolme.

Codelists teenus on olemteenus (Togaf stereotüüp <<entity component>>), mis vahendab muuhulgas suhtlust Codelists2 andmebaas komponendiga (UML konnektor vastavasse liidesesse). See teenus pakub ühte rakendusliidest, ühte ühendust (Klassifikaatorite API) ja vajab ühte ühendust (Klassifikaatorite andmebaasiliiides). Rakenduse ülesehituse skeemi näidisel on näidatud, et Codelists teenus sõltub platvormist WSO2 EI.

Rakenduse ülesehituse skeemil on kujutatud ka rakenduse OCI konteinerite ülesehitust Codelists teenus komponendi konteineri osas. Seal on näidatud, et Codelists2 konteiner baseerub WSO2 MI konteineril (mis pakub WSO2 MI platvormi WSO2 Micro Integrator tarkvara põhjal), mis baseerub eclipse-temurin konteineril, mis pakub JDK platvormi.

3.8. Näide 2

Rakenduse kirjeldamiseks sobib hästi Togaf rakenduse kommunikatsioniskeem. Kommunikatsioniskeem toob välja komponentide vajavad liidesed, pakutavad liidesed ja sõltuvused teistest rakendustest. Keerulisema rakenduse puhul võib olla otstarbekas erinevate komponentide tarvis teha eraldi joonised. Alljärgnevas näites on kirjeldatava süsteemi komponendid toodud välja roheliselt:



Jooniste aluseks olev mudel on järgmine:

