Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op www.nioc2025.nl voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

NIOC 2002 "ICT onderwijs van e-motion naar e-mission"
6e Nationaal Informatica Onderwijs Congres

SAXION Hogeschool Enschede

NIOC 2002-029 workshop

Realtime Embedded Systems

J.B.van de Vrie

1 Inloggen op Sun systemen

Op de Sun systemen in lokaal Wb59 zijn useraccountsbeschikbaar in het kader van het NIOC2002.
Elke useraccount bestaat uit nioc met een volgnummer, dus:
nioc1, nioc2,…..,nioc32
Het bijbehorende password is nioc2002
Gebruik svp de niocaccount met **een nummer dat overeenkomt met het Sun-blade-nr** in het welkom/loginscherm. Zo houdt ieder zijn eigen homedir.

Vanuit de taakbalk is nu Netscape direct op te starten of ook een terminalvenster indien gewenst.

VRIENDELIJK VERZOEK
Sluit een sessie af met alleen uit te loggen en schakel de systemen nooit uit.
Dit voorkomt lange opstartprocedures voor volgende gebruikers

2 Opdrachten

2.1 Kennismaking UML-RT

2.1.1 Extra elementen: capsules, ports, protocols

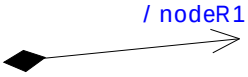
Start vanuit de taakbalk Netscape op en open de pagina /hio/appl/RoseRT/Tutorials/unix/index.htm .Open van hieruit met de verwijzingen de verschillende animaties over capsules, ports , protocollen etc.

2.1.2 Extra diagrammen: hierarchische structure diagrams

Start vanuit een terminalvenster Rational ROSE RT met het commando: rosert
Accepteer de defaultwaarden van opstartkeuzes, zoals Framework RTC++.
Open in ROSE RT (vanuit het filemenu: Open) een simpel demomodel:
TokenDemo2002.rtmdl vanuit de dir /hio/praktikum/i4/IG0_RTOGO
Voor het aanbrengen van wijzigingen: eerst saven naar homedir: /home/nioc#

Dit betreft een network bestaande uit twee nodes: een masternode en een (gewone) node, die als een signal een TOKEN uitwisselen en dit op basis van start en afloop van een timer deze enige tijd vast houden en weer terugsturen naar de eerdere zender. De masternode heeft als extra taak bij opstart het token in het systeem te brengen.

Open vanuit de browser in ROSE RT voor dit model in de map Logical View het main class diagram. In dit classdiagram zijn capsules, ports en protocols te vinden. Na een rechtermuisklik op de capsule TokenNetwork is een



structurediagram te selecteren. In de structuur van deze capsule zijn de compositionrelaties zoals hierboven weergegevenmet een symbool, vertaald in een (hierarchische) strucuuur.

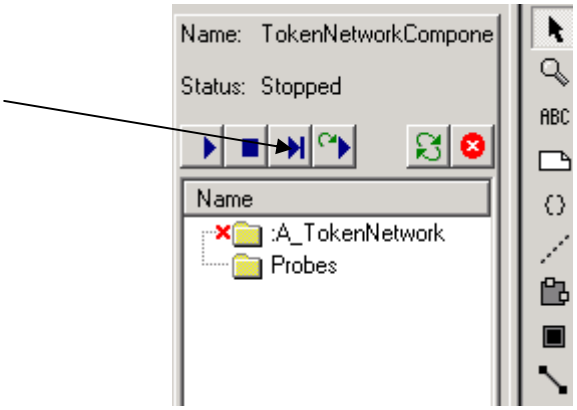
VRAGEN
Uit welke capsules is het TokenNetwork opgebouwd?
Hoe zijn capsules en protocols te onderscheiden?
Waar zijn de ports terug te vinden?

Bekijk ook eens het statediagram van de masternode via de role R1 ervan.

VRAAG:
Welk verschil is er nu eigenlijk tussen een masternode en een gewone node van het tokennetwork?

2.1.3 executeerbare modellen

Met hetzelfde model nog geopend in Rose RT kunnen we ook uitproberen welke mogelijkheden het runnen van executeerbare modellen biedt.
In de browser is eerst een component gedefinieerd op basis van de eerder onderzochte capsules. In de specificatie van TokenNetworkComponent is de top level capsule geselecteerd (rechter muisklik in de browser op deze capsule: Open Specification) via het tabblad C++Executable. Willen we het hele model gaan executeren dan moet hier de selectie van de omvattende capsule Token Network aangehouden worden. Het is middels deze configuratie van componenten ook mogelijk om slechts een deel van een model te executeren door een andere capsule als top level capsule te selecteren. Op het tabblad C++ compilation kan het target geconfigureerd worden waar deze component voor zal worden gegenereerd. Het Sun systeem waar we op werken is in deze situatie voor ons nu zowel host als target. Vandaar de selectie voor Sun5T.sparc-gnu-2.95.1
Deze component kan nu op een target worden gerund door in de ROSERT browser een Target aan te maken waarop een instantie van de component kan runnen. In het geopende model is dat reeds uitgevoerd.
Voer nu de modelexecutie uit voor deze componentinstance op het aangegeven SunTarget. Door in de RunTime View in de browser het proces een stap uit te



laten voeren met de stepknop , worden in de browser de capsules zichtbaar van TokenNetwork. Met de rechteruisknop zijn hier na selectie ook de statediagrams te volgen tijdens verdere (stapsgewijze) executie.

Voor wie nu zelf een model wil bouwen/aanpassen een paar kleine opdrachten :

OPDRACHT
Breid het gegeven Tokensystem uit tot een systeem bestaande uit drie nodes, één masternode en twee gewone nodes. Laat het token rondgaan tussen de nodes, met een tijdsduur voor het vasthouden van de token, als in de originele opgave.

VRAAG:
Is een tokensystem ook op te bouwen uit nodes met slechts één poort waar zowel een token verstuurd als ontvangen kan worden?
Is dan nog inheritance toe te passen?

2.2 Eenvoudige demo koffiemachine:

2.2.1 Modellen en productieproces:

Open vanuit /hio/appl/RoseRT/Examples/Models/C++/CoffeeMachine het model Coffeemachine.rtmdl

Traceer in de browser vanuit de verschillende Views

1. requirements:	use cases,
2. analysis :	uc realization, sequence diagrams, protocollen, ports
3. design:	class diagram, state
4. implementatie:	packages, triggers & actions, components
5. deployment:	executable models, probes & traces, debugging
6. testen:	testcases, states

VRAAG:
Op welke manier hangen de testcases samen met de usecases?
Let hierbij ook eens op het traceability diagram te vinden onder Logical View – TestHarnesses-MarkI Tests- Scenarios_MarkI

2.2.2 Architecture centric

In de zelfde CoffeeMachinedemo zijn een paar architecture patterns toegepast:

Hardware abstractie is te vinden door in de structuur van CoffeMachine Mark1 te duiken:
Browser- Logical View- Brewers- Main class diagram : selecteer met rechter muisklik het structuurdiagram van CoffeMachine_Mark I.
Binnen de container capsule structuur is zowel Hardware abstractie als een Mediator pattern te vinden.

VRAGEN:
Welke capsule is in feite de "mediator" en waarom?
Hoe is hardware abstractie te herkennen?

Een vorm van Layering is op het eerste niveau binnen de top level capsule te vinden door deaangebrachte scheiding tussen hardwarelaag en softwarelaag.

Hiervan wordt dankbaar gebruik gemaakt in de testcapsules.
Per TestCase is er een stub gemaakt voor de hardware en een simulatie van de bijbehorende hardwaresignals.

2.3 Voor "gevorderden"

2.3.1 Ontwerpen van state diagrams

Er is voor wie daar aan toekomt nog een extra opdracht Verwarmingssysteem.

2.3.2 Toepassing van patterns

Voor geinteresseerden is er ook nog een extra opdracht Safe Dynamic Structures, waarin verschillende patterns kunnen worden toegepast.

NIOC 2002 *"ICT onderwijs van e-motion naar e-mission"*

6e Nationaal Informatica Onderwijs Congres

SAXION Hogeschool Enschede

NIOC 2002-029 workshop

Realtime Embedded Systems

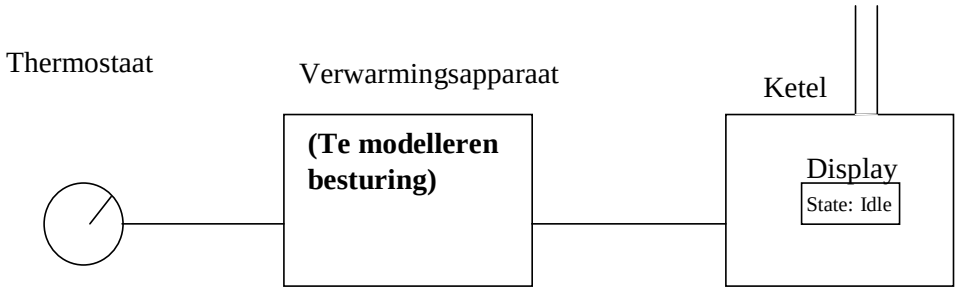
EXTRA OPDRACHTEN

J.B.van de Vrie

2.3 Voor "gevorderden"

2.3.1 Ontwerpen van state diagrams:Verwarmingssysteem

Het verwarmingsapparaat in deze opgave is een gasgestookte ketel die zorgt voor het verwarmen van een woning door het water in de radiatoren te verwarmen.
Het systeem wordt door de gebruiker bestuurd door een thermostaat.
Is de temperatuur in de kamer lager dan die van de thermostaat, dan moet het verwarmingsapparaat gaan stoken, anders niet.
Het systeem mag nooit stoken wanneer de temperatuur van het water boven de 90 graden Celsius komt of wanneer de waterdruk in de radiatoren te laag is (minder dan 0,5 atmosfeer).
Bovendien is er een anti-pendel mechanisme, wat betekent dat wanneer de ketel uitschakelt, deze tenminste 1 minuut uitgeschakeld moet blijven dit om te voorkomen dat de ketel steeds aan- en uitschakelt). Op het display van het verwarmingselement moet steeds af te lezen zijn in welke toestand deze zich



bevindt.

Bovenstaand verwarmingsapparaat gaat deel uitmaken van een computergestuurd systeem. Dit systeem wordt ontworpen op basis van object-georiënteerde technieken.
In het systeem is het verwarmingsapparaat afgebeeld als een capsule waarbinnen oa de waterdruk en watertemperatuur wordt gecontroleerd.
Bovendien kent het apparaat de methoden *schakel_in* en *schakel_uit*. Deze methoden worden aangeroepen door de thermostaat om de ketel te starten of te laten stoppen met stoken.

Opdracht : Ontwerp de besturing van de verwarmingsketel mbv Rational RoseRT

Er is al een basismodel beschikbaar verwarmingbasis.rtmdl in /hio/praktikum/i4/IG0_RTOGO , zodat nu de states, met transitions triggers en actions voldoende aandacht kunnen krijgen.

Een voorbeeld van mogelijke Use cases met scenario's:

Startsysteem :
- stap 1 initialiseer devices
- stap 2 maak devices operational
- stap 3 display : ready

VraagVerwarmen
- Stap 1 wacht op brander ready
- Stap 2 schakel brander in

Stopverwarmen
- Stap 1 schakel brander uit
- Stap2 start antipendel timer

SchakelSysteemUit
- Stap 1 schakelDevicesInactive
- Stap 2 start uitschakeltimer
- Stap3 wacht op timeout
- Stap4 switch off

SignaleerFout
- Stap1 schakel brander uit
- Stap2 schakel devices naar inAcitve
- Stap3 display error

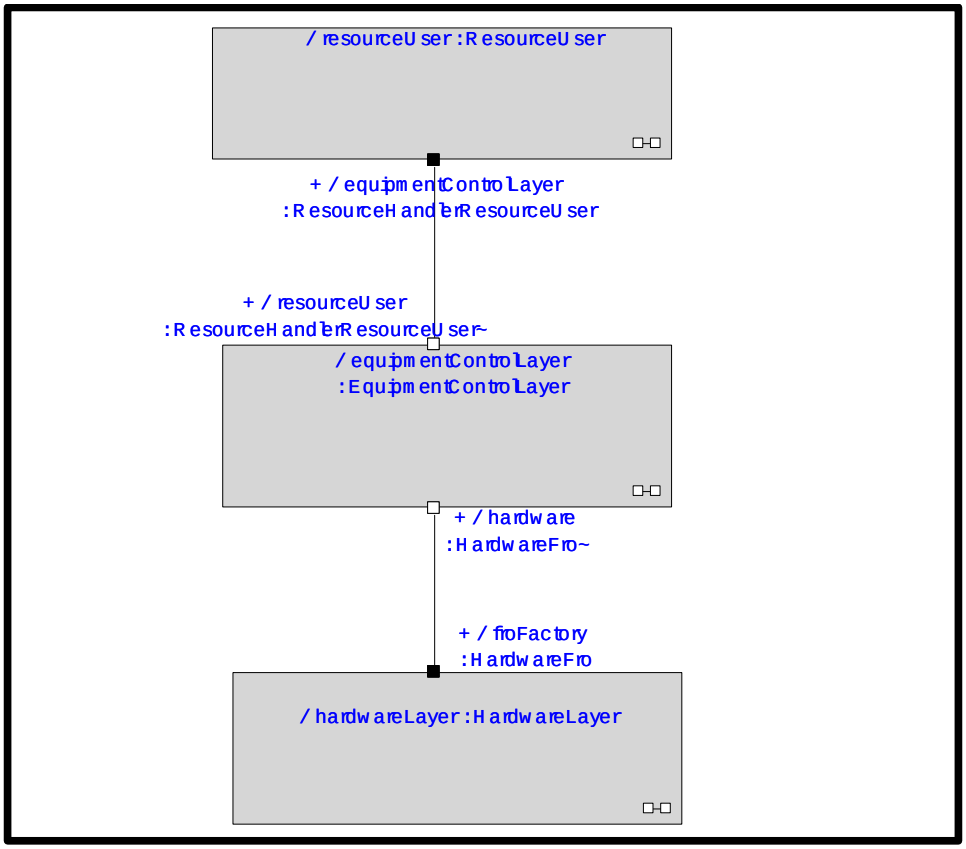
2.3.2 Toepassing van patterns:Safe Dynamic structure opdracht

Een algemeen probleem in veel systemen is het verkrijgen van toegang tot een schaarse resource. Het is dan noodzakelijk de toegang tot deze resource dynamisch te coordineren. Als de resource beschikbaar is dient er een dynamische verbinding te worden opgezet. Na gebruik dient de resource weer vrijgegeven te worden en moet de verbinding verbroken te worden.

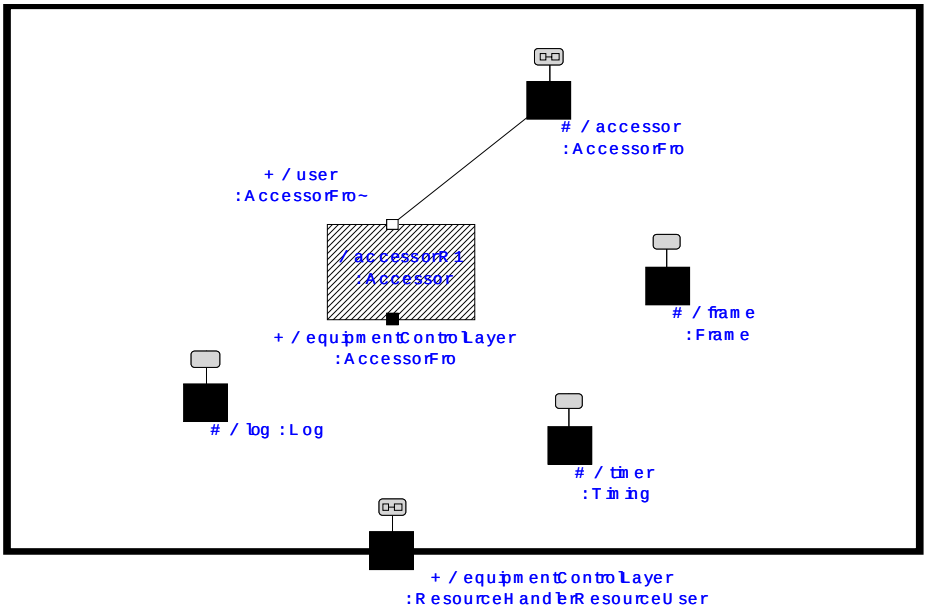
Dit probleem kent een algemene oplossing bekent onder de naam ***Safe dynamic structure pattern.***

Probleembeschrijving:

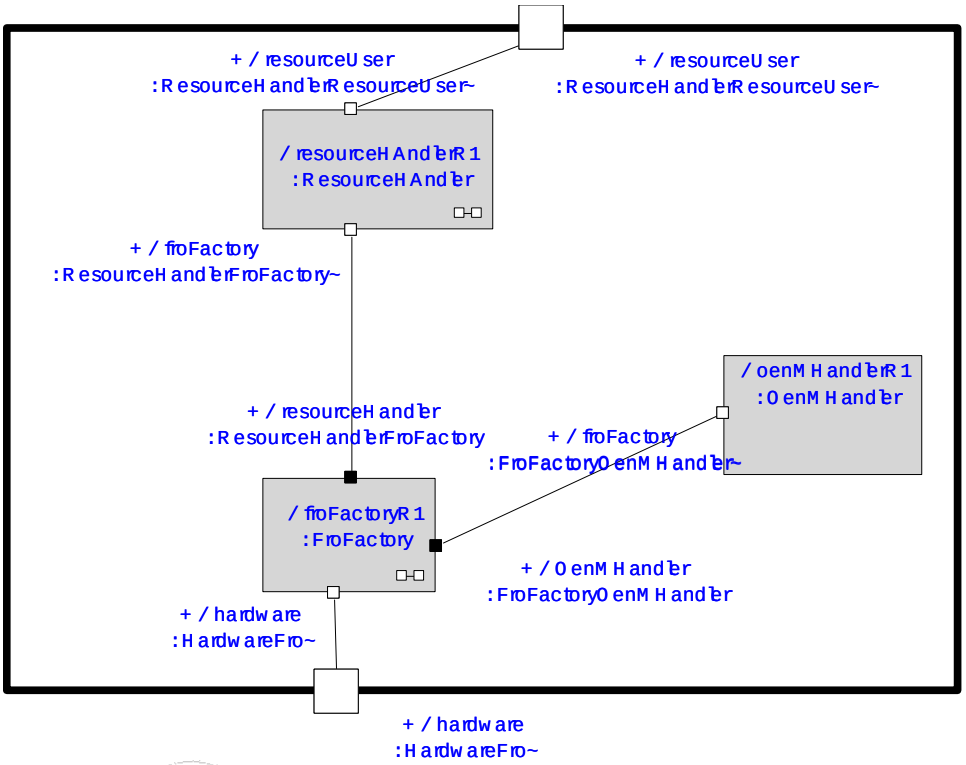
Gegeven een resource in de hardware waartoe een softwarecomponent toegang dient te hebben.
Om deze toegang mogelijk te maken is een gelaagde softwarearchitectuur opgezet bestaande uit de volgende lagen:



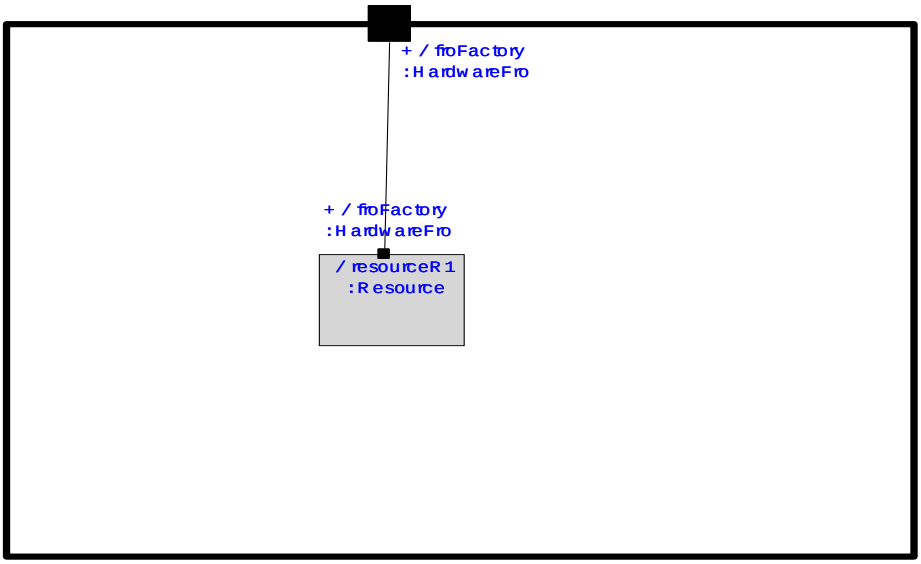
- **Logical Resource Layer**
  In deze laag bevindt zich de capsule ResourceUser die communicatie naar een resource aanvraagt of vrijgeeft. Om een dynamische verbinding te maken naar de gevraagde fro is binnen de ResourceUser de capsule Accessor aanwezig.



- **EquipmentControlLayer:**
  In deze laag boven de hardwarelaag bevindt zich per resource in de hardwarelaag een overeenkomstige fro (Facade Resource; de interface naar de resource). Wanneer er meerdere fro's zijn worden deze beheerd door de FroFactory. Verder is er een Resourcehandler waarmee de aanvrager communiceert. De resourcehandler handelt een aanvraag voor een resource af door de controleren of de gevraagde resource (eigenlijk de fro) beschikbaar is, en zo ja, een verbinding te leggen naar de fro. Ook is in deze layer de O&M-handler (Operation and Maintenance) aanwezig. Deze handler kan de FroFactory vragen bepaalde resources beschikbaar te stellen of juist niet.

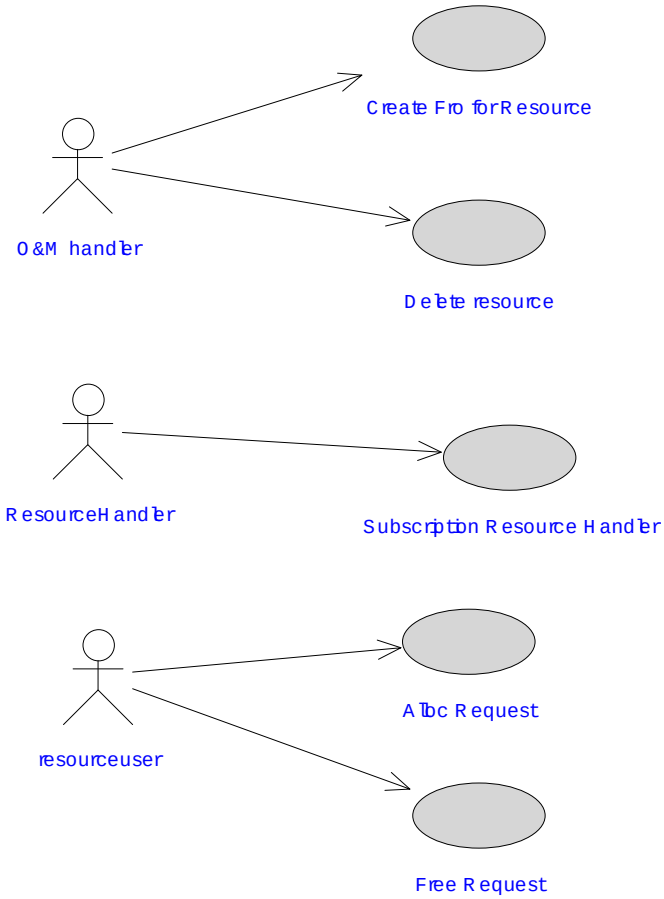


- **Hardware Layer:**



In deze laag bevindt zich de capsule RO (Resource) waarmee gecommuni-ceerd dient te worden.

Er zijn een vijftal Usecases te onderscheiden



a. Voor elk van deze usecases is een sequence diagram te maken. Voer dit uit

b. Voeg de states en transition triggers &actions toe en test het geheel.