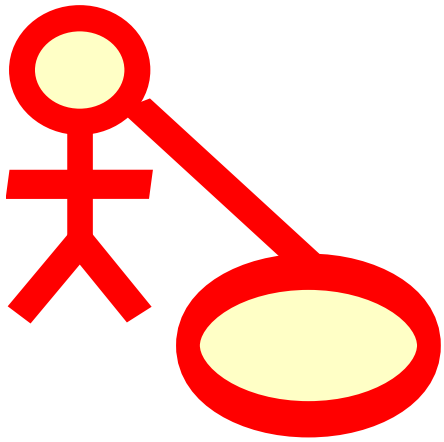
Het handboek van Umbrello UML Modeller



Het handboek van Umbrello UML Modeller

Inhoudsopgave

Samenvatting

Umbrello UML Modeller helpt bij het softwareontwikkelingsproces door gebruikmaking van de industrie standaard Unified Modelling Language (UML). opdat u diagrammen kunt maken om uw systeem te ontwerpen en te documenteren.

Hoofdstuk 1

Introductie

Umbrello UML Modeller is een UML diagramgereedschap dat u bij het proces van softwareontwikkeling kan ondersteunen. In het bijzonder tijdens de analyse en ontwerpfase van dit proces, kan Umbrello UML Modeller u helpen een kwalitatief hoogwaardig product te verkrijgen. UML kan ook benut worden om uw software ontwerpen te documenteren ten dienste van u en collega ontwikkelaars.

Over een goed model van uw software beschikken is de beste manier om te communiceren met andere ontwikkelaars die aan het project werken, en met uw klanten. Een goed model is buitengewoon belangrijk voor middelgrote en zeer grote projecten, maar is ook heel nuttig voor de kleinere. Zelfs als u werkt aan een klein eenmansproject, zult u welvaren bij een goed model omdat het u het overzicht verschaft dat u helpt om de zaken de eerste keer meteen goed te coderen.

UML is de diagramtaal die gebruikt wordt om zulke modellen te beschrijven. U kunt uw ideeën in UML representeren met verschillende soorten diagrammen. Umbrello UML Modeller 2.11 ondersteunt de volgende soorten:

- Klasse-diagram
- Volgorde-diagram
- Collaboratie-diagram
- Use Case-diagram
- Toestandsdiagram
- Activiteitsdiagram
- Component-diagram
- Uitzettingsdiagram
- Entiteitrelatiediagram

Meer informatie over UML kunt u vinden op de website van OMG, http://www.omg.org die de UML standaard gecreëerd hebben.

Wij hopen dat u veel plezier zult hebben van Umbrello UML Modeller en dat het u helpt bij het maken van kwalitatief hoogwaardige software. Umbrello UML Modeller is Vrije Software en gratis verkrijgbaar, het enige dat wij van u vragen is om bugs, problemen of suggesties te melden aan de ontwikkelaars van Umbrello UML Modeller op umbrello-devel@kde.org of https://bugs.kde.org.

Hoofdstuk 2

Grondbeginselen van UML

2.1 Over UML ...

Dit hoofdstuk geeft u snel een overzicht van de grondbeginselen van UML. Houdt u voor ogen dat dit geen allesomvattende studie is over UML maar veeleer een korte inleiding in UML die men kan lezen als een basiscursus UML. Wilt u meer te weten komen over de Unified Modelling Language, of in bredere zin over software-analyse en -ontwerp, raadpleeg dan een van de vele boeken die over dit onderwerp voorhanden zijn. Er zijn ook een groot aantal cursussen op het Internet voorhanden, waarmee u een goede start kunt maken.

De Unified Modelling Language (UML) is een taal om diagrammen te maken of een notatiewijze om modellen van objectgeoriënteerde softwaresystemem te specificeren, te visualiseren en te documenteren. UML is geen ontwikkelmethode, d.w.z. het vertelt u niet wat u eerst moet doen en wat daarna, of hoe u uw systeem moet ontwerpen, maar het helpt u om uw systeem te visualiseren en te communiceren met anderen. UML staat onder toezicht van de Object Management Group (OMG) en is de industriestandaard voor het grafisch weergeven van software.

UML is gemaakt voor het ontwerpen van objectgeoriënteerde software en heeft beperkt nut voor andere programmeerparadigma's.

UML is opgebouwd uit vele modelelementen die de verschillende delen van een softwaresysteem vertegenwoordigen. De UML-elementen worden gebruikt om diagrammen te maken, die een bepaald deel of een gezichtspunt van een systeem voorstellen. De volgende soorten diagrammen worden ondersteund door Umbrello UML Modeller:

- *use case diagrammen* tonen actoren (mensen of andere gebruikers van het systeem), use cases (de scenario's wanneer zij het systeem gebruiken), en hun relaties
- *Klassediagrammen* tonen klassen en hun onderlinge relaties
- *Volgordediagrammen* tonen objecten en een volgorde van methode-aanroepen die zij doen naar andere objecten.
- *Collaboratiediagrammen* tonen objecten en hun relaties, met nadruk op de objecten die deelnemen aan de berichtenuitwisseling
- *Toestandsdiagrammen* tonen toestanden, toestandsveranderingen en gebeurtenissen van een object of een deel van het systeem
- *Activiteitsdiagrammen* tonen activiteiten en de overgang van de ene activiteit naar de andere samen met de gebeurtenissen die in een bepaald deel van het systeem optreden
- *Componentdiagrammen* tonen programmacomponenten op het hoogste niveau (zoals bijv. KParts of Java Beans).

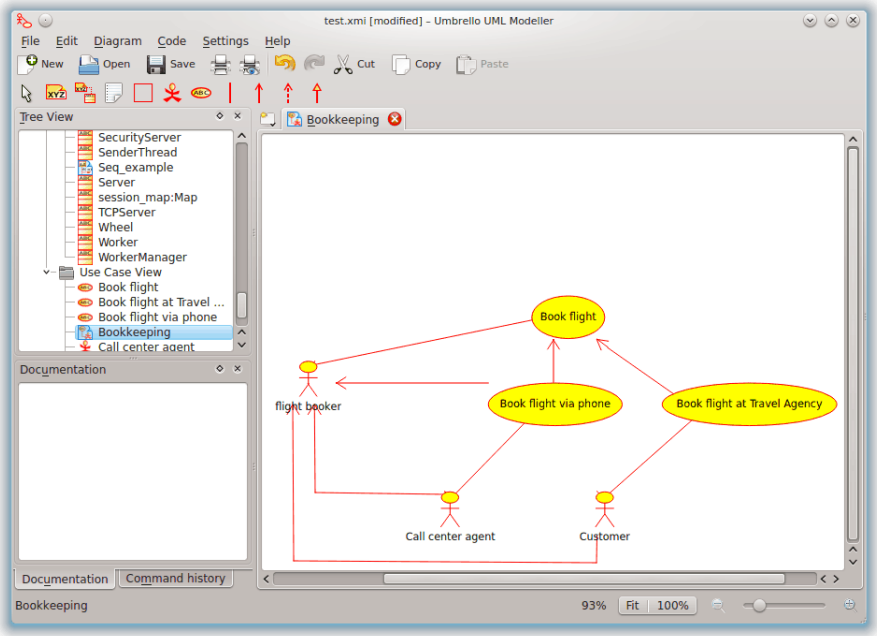
- *Deploymentdiagrammen* tonen de instanties van de componenten en hun relaties.
- *Entiteitsrelatie-diagrammen* tonen gegevens en de relaties en beperkingen tussen de gegevens.

2.2 UML-elementen

2.2.1 Use case diagram

Use case-diagrammen beschrijven de relaties en afhankelijkheden tussen een groep van *use cases* en de actoren die deelnemen aan het proces.

Belangrijk om op te merken is dat use case-diagrammen niet geschikt zijn om het ontwerp te representeren, en niet het inwendige van een systeem kunnen beschrijven. Use case-diagrammen zijn bedoeld om de communicatie met de toekomstige gebruikers van een systeem, en met de klant, te vergemakkelijken, en zijn in het bijzonder behulpzaam bij het vaststellen van welke benodigde kenmerken een systeem moet hebben. Use case diagrammen vertellen *wat* het systeem moet doen maar specificeren niet — en kunnen dat ook niet — *hoe* dit gerealiseerd moet worden.



Umbrello UML Modeller toont een use case-diagram

2.2.1.1 Use case

Een *use case* beschrijft — vanuit het standpunt van de actoren — a groep activiteiten in een systeem die een concreet, tastbaar resultaat oplevert.

Use cases zijn beschrijvingen van kenmerkende interacties tussen de gebruikers van een systeem en het systeem zelf. Zij representeren de externe interface van het systeem en specificeren een soort pakket van eisen diehet systeem moet uitvoeren (onthoud: alleen wat, niet hoe).

Bij het werken met use cases is het belangrijk om enkele eenvoudige regels in acht te nemen:

- Iedere use case is gerelateerd aan tenminste één actor

- Iedere use case heeft een initiator (bijv. een actor)
- Iedere use case leidt tot een relevant resultaat (een resultaat met 'waarde')

Use cases kunnen ook relaties met andere use cases hebben. De drie meest karakteristieke relaties tussen use cases zijn:

- *«include»* geeft aan dat een use case zich *binnen* een andere use case afspeelt
- *«extends»* geeft aan dat in bepaalde situaties, of op een zeker moment (ook wel het uitbreidingspunt genoemd) een use case uitgebreid zal worden met een andere.
- *Generalisatie* geeft aan dat een use case de karakteristieken erft van de 'super'-use case, en sommige ervan kan herdefiniëren of nieuwe kan toevoegen, op eenzelfde wijze als bij de overerving bij klassen het geval is.

2.2.1.2 Acteur

Een actor is een externe entiteit (buiten het systeem) die samenwerkt met het systeem door deelname aan (en veelal door initiëren van) een use case. In de dagelijkse werkelijkheid kunnen actoren mensen zijn (bijvoorbeeld gebruikers van een systeem), andere computersystemen of externe gebeurtenissen.

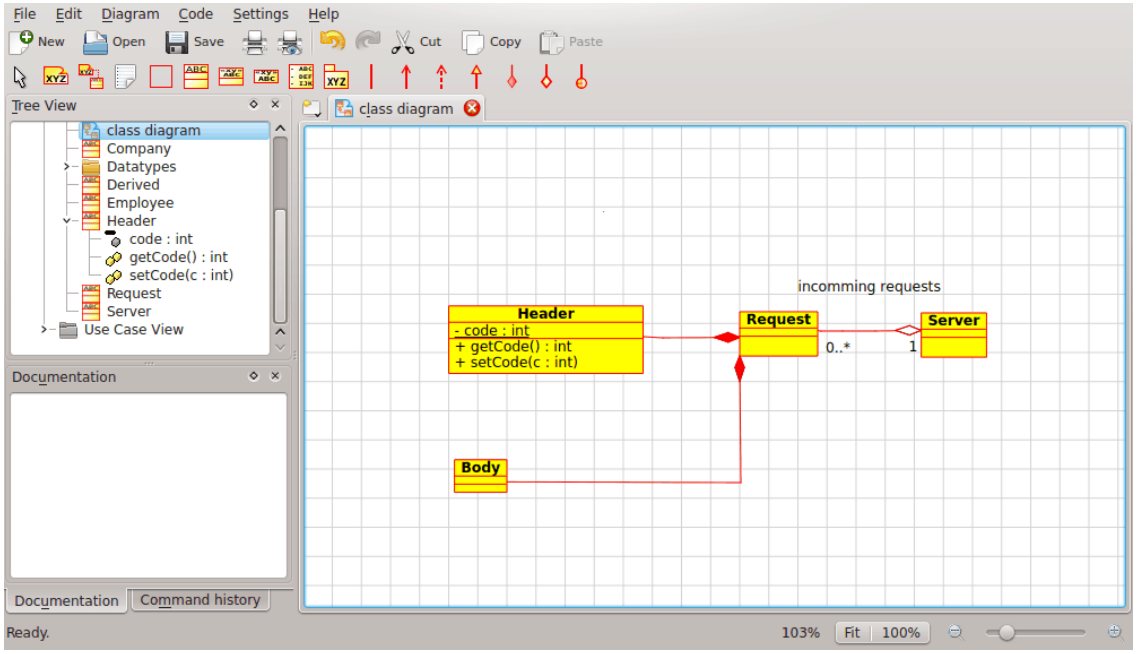
Actoren representeren niet de *fysieke* mensen of systemen, maar hun *rol*. Dit impliceert dat als een persoon met het systeem samenwerkt op verschillende manieren (hij meet zich verschillende rollen aan) hij door meerdere actoren voorgesteld zal worden. Bijvoorbeeld een persoon die telefonische klantenondersteuning geeft en orders invoert van de klant in het systeem, voorgesteld wordt door een actor 'afdeling ondersteuning' en een actor 'afdeling verkoop'

2.2.1.3 Use case-omschrijving

Use case-omschrijvingen zijn textuele verhandelingen van de use case. Zij nemen gewoonlijk de de vorm van een aantekening of een document aan dat op een bepaalde manier gekoppeld is aan de use case, en de processen of activiteiten die in de use case plaatsvinden, nader toelicht.

2.2.2 Klassediagram

Klassediagrammen tonen de verschillende klassen waaruit het systeem is gemaakt, en hoe zij aan elkaar gerelateerd zijn. Van klassediagrammen zegt men dat zij 'statische' diagrammen zijn omdat zij weliswaar de klassen weergeven, samen met hun methoden en attributen, alsmede de statische relaties tussen hen (i.e. welke klassen 'hebben weet' van welke klassen of welke klassen 'maken deel uit' van een andere klasse), maar niet de methode-aanroepen tussen hen onderlingweergeven.

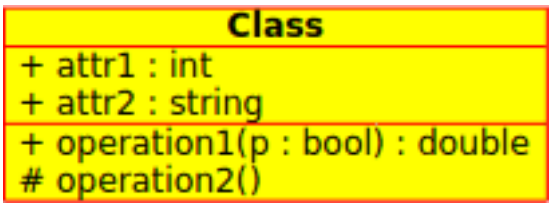


Umbrello UML Modeller toont een klassediagram

2.2.2.1 Klasse

Een klasse definieert de attributes en de methoden van een set objecten. Alle objecten die van deze klasse afgeleid zijn (instanties van deze klasse) hebben eenzelfde gedrag, en hebben overeenkomstige verzamelingen met attributen (ieder object heeft zijn eigen set). I.p.v. klasse wordt ook wet de term 'type' gebezigd, maar het moet nadrukkelijk gesteld worden dat deze twee niet identiek zijn, type is een algemenere term.

In UML, worden klassen door rechthoeken gerepresenteerd, met de naam van de klasse; hierin kunnen ook de attributen en operaties van de klasse weergegeven worden in twee andere 'vakken' binnen de rechthoek.



Visuele representatie van een klasse in UML

2.2.2.1.1 Attributen

In UML, worden attributen minsten met hun naam weergegeven, maar ook kunnen hun type, beginwaarde en andere eigenschappen weergegeven worden. Attributen kunnen ook worden getoond met hun zichbaarheid:

- + staat voor *public* attributen
- # staat voor *protected* attributen
- – staat voor *private* attributen

2.2.2.1.2 Operaties

Operaties (methoden) worden eveneens met minstens hun naam getoond; Ook zij kunnen hun parameters en return typen tonen. Operaties kunnen, net als attributen, hun zichtbaarheid weergeven:

- + staat voor *public* operaties
- # staat voor *protected* operaties
- - staat voor *private* operaties

2.2.2.1.3 Sjablonen

Klassen kunnen sjablonen hebben, een waarde die gebruikt wordt voor een niet gespecificeerde klasse of type. Het sjabloontype wordt gespecificeerd wanneer de klasse geïnstantieerd wordt (bijv. een object wordt aangemaakt). Sjablonen komen voor in modern C++ en zullen in Java 1.5 geïntroduceerd worden onder de naam generics.

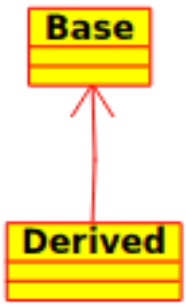
2.2.2.2 Klassenassociaties

Klassen kunnen op verschillende wijze aan elkaar gerelateerd zijn (met elkaar geassocieerd zijn):

2.2.2.2.1 Generalisatie

Overerving is een van de fundamentele concepten van objectgeoriënteerdprogrammeren, waarbij een klasse 'toegang krijgt tot' (bijna) alle attributen en operaties van de klasse waar het van erft en sommige ervan opnieuw kan implementeren (overriding), alsook meer attributen en operaties van zichzelf kan toevoegen.

In UML, plaatst een *generalisatie*-associatie tussen twee klassen hen in een hiërarchie die het concept van overerving van een afgeleide klasse van een basisklasse representeert. In UML, worden generalizaties weergegeven door een lijn, die de twee klassen met elkaar verbindt, met een pijlpunt aan de kant van de basisklasse.



Visuele representatie van een generalisatie in UML

2.2.2.2.2 Associaties

Een associatie vertegenwoordigt een relatie tussen klassen, en voorziet in de gemeenschappelijke semantiek en structuur voor vele soorten'connecties' tussen objecten.

Associaties zijn het mechanisme dat objecten in staat stelt met elkaar te communiceren. Het beschrijft de connectie tussen verschillende klassen (de connectie tussen de eigenlijke objecten noemt men een objectconnectie, of *link*).

Associaties kunnen een rol hebben die het doel van de associatie specificeert en zowel uni- als bi-directioneel zijn (dit geeft aan of de twee objecten die een relatie hebben, berichten naar elkaar kunnen sturen, of dat slechts één van hen weet heeft van de ander). Ieder uiteinde van de associatie heeft ook een multipliciteit, die oplegt hoeveel objecten aan deze kant van de associatie gerelateerd kunen zijn aan telkens een object aan de andere kant.

In UML, worden associaties weergegeven als lijnen, die de klassen die aan de relatie deelnemen, met elkaar verbinden. Hierbij kan ook de rol en de multipliciteit van elk van de deelnemers worden getoond. De multipliciteit wordt afgebeeld als een bereik [min..max] van niet-negative waarden. Een asterisk (*) aan de maximumkant geeft oneindig weer.

Visuele representatie van een associatie in UML

2.2.2.2.3 Aggregatie

Aggregaties zijn een speciaal type associaties waarin de twee deelnemende klassen geen gelijkwaardige status hebben, maar een 'geheel-deel' relatie vormen. Een aggregatie beschrijft hoe de klasse die de rol van geheel heeft, samengesteld wordt uit de andere klassen, die de rol van deel hebben. Voor aggregaties heeft de klasse die optreedt als geheel, altijd de multipliciteit één.

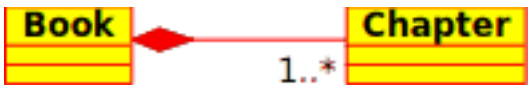
In UML, worden aggregaties gerepresenteerd door een associatie die een ruit heeft aan de kant van het geheel.

Visuele representatie van een aggregatie-relatie in UML

2.2.2.2.4 Compositie

Composities zijn associaties die *zeer sterke* aggregaties vertegenwoordigen. Dit betekent dat composities evengoed geheel-deel relaties vormen, maar de relatie is zo sterk dat de delen niet op zichzelf kunnen bestaan. Zij bestaan slechts binnen het geheel, en als het geheel vernietigd wordt, gaan de delen er ook aan.

In UML, worden composities gerepresenteerd door een gekleurde ruit aan de kant van het geheel.



2.2.2.3 Andere klassediagram-onderdelen

Klassediagrammen kunnen verscheidene andere onderdelen bevatten naast klassen.

2.2.2.3.1 Interfaces

Interfaces zijn abstracte klassen, wat wil zeggen dat instanties niet direct uit hen gemaakt kunnen worden. Zij kunnen operaties bevatten maar geen attributen. Klassen kunnen overerven van interfaces (via een realisatie-associatie) en pas dan kunnen klassen van deze diagrammen gemaakt worden.

2.2.2.3.2 Gegevenstypen

Gegevenstypen zijn primitieven die normaal gesproken ingebouwd zijn in een programmeertaal. Bekende voorbeelden zijn integers en booleans. Zij kunnen geen relatie met klassen hebben, maar klassen kunnen wel een relatie met hen hebben.

2.2.2.3.3 Enumeraties

Enumeratie is een eenvoudige lijst met waarden. Een typisch voorbeeld is een enumeratie voor de dagen van de week. De keuzemogelijkheden binnen een enumeratie noemt men enumeratieconstanten. Evenals gegevenstypen kunnen zij geen relatie met klassen hebben, maar klassen kunnen wel een relatie met hen hebben.

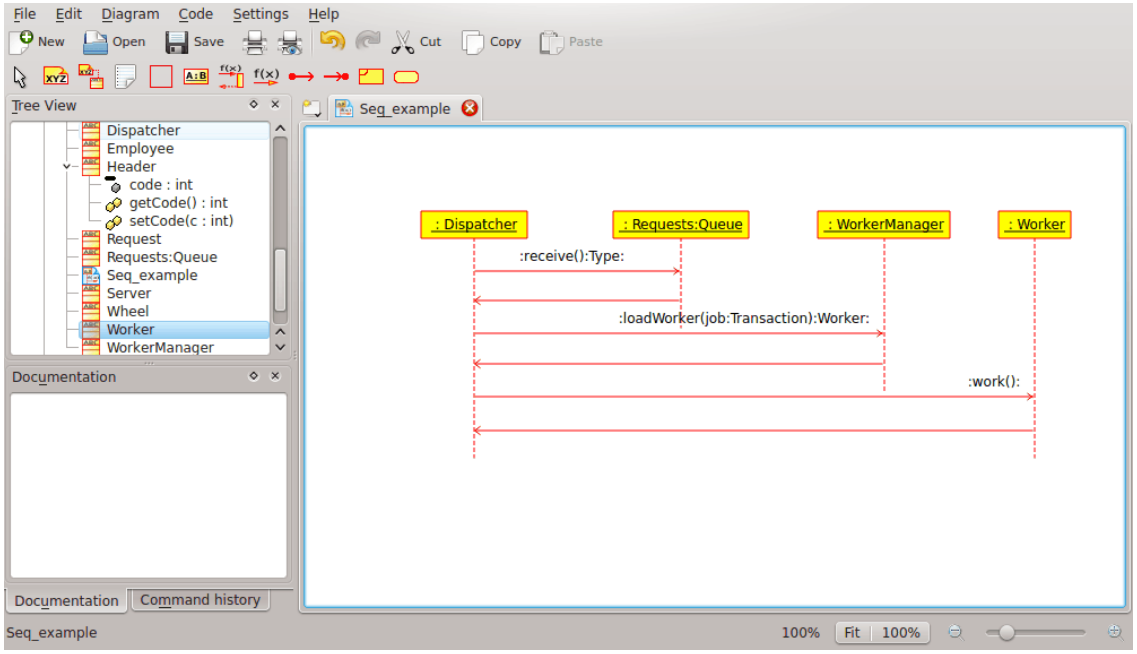
2.2.2.3.4 Pakketten

Pakketten vertegenwoordigen een naamruimte in een programmeertaal. In een diagram worden zij gebruikt om delen van een systeem te representeren, die meer dan één klasse, wellicht zelfs honderden klassen, kunnen bevatten.

2.2.3 Volgordediagrammen

Volgordediagrammen geven de berichtenuitwisseling weer (bijv. methode-aanroep) tussen verscheidene objecten in een specifieke tijd-begrensde situatie. Objecten zijn instanties van klasses. Volgordediagrammen leggen speciale nadruk op de volgorde waarin en de tijdstippen waarop de berichten naar de objecten verstuurd worden.

In volgordediagrammen worden objecten gerepresenteerd door vertikale onderbroken lijnen, met de naam van het object bovenaan. De tijd-as loopt ook verticaal, en neemt toe naar beneden, zodat berichten verstuurd worden van het ene object naar de nadere in de vorm van pijlen met de namen van de operatie en de parameters erbij.



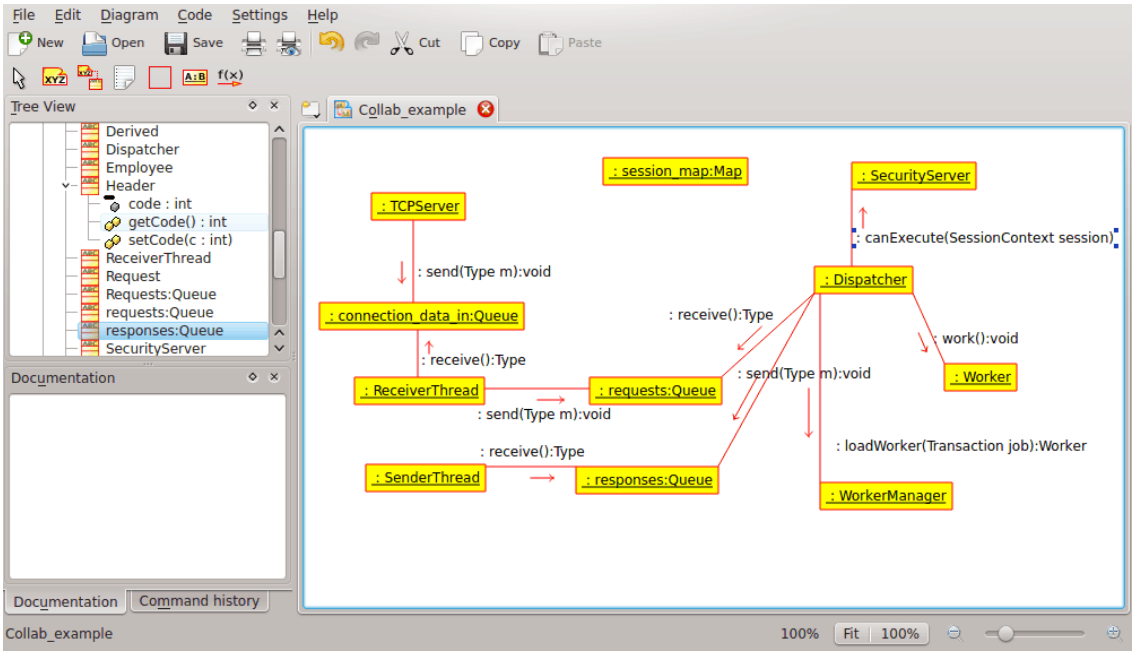
Umbrello UML Modeller toont een volgordediagram

Berichten kunnen ofwel synchroon zijn, het normale type bericht aanroep waarbij de besturing wordt doorgegeven aan het aangeroepen object totdat de betreffende methode voleindigd is, of asynchroon, waarbij de besturing direct weer teruggeven wordt aan het aanroepende object. Synchrone berichten hebben een verticaal kader aan de kant van het aangeroepen object om het verloop van de programmabesturing te laten zien.

2.2.4 Collaboratiediagrammen

Collaboratiediagrammen laten de interacties zien die plaatsvinden tussen objects die participeren in een specifieke situatie. Dit is min of meer dezelfde informatie als weergegeven bij volgorde-diagrammen maar daar valt de nadruk op hoe de interacties plaats vinden in de tijd, terwijl bij Collaboratiediagrammen de relaties tussen de objecten en hun topologie op de voorgrond treden.

In collaboratiediagrammen worden berichten die van het ene object naar het andere worden gestuurd, voorgesteld door pijlen, waarbij de naam van het bericht, de parameters, en de bericht-volgorde, weergegeven wordt. Collaboratiediagrammen zijn bij uitstek geschikt om een specifiek programma-verloop of situatie weer te geven, en zijn een van de beste diagramtypen om snel een process in de programmalogica te demonstreren of toe te lichten.



Umbrello UML Modeller toont een collaboratiediagram

2.2.5 Toestandsdiagram

Toestandsdiagrammen geven de verschillende toestanden van een object weer gedurende zijn bestaan, en de stimuli, die er voor zorgen dat het object zijn toestand wijzigt.

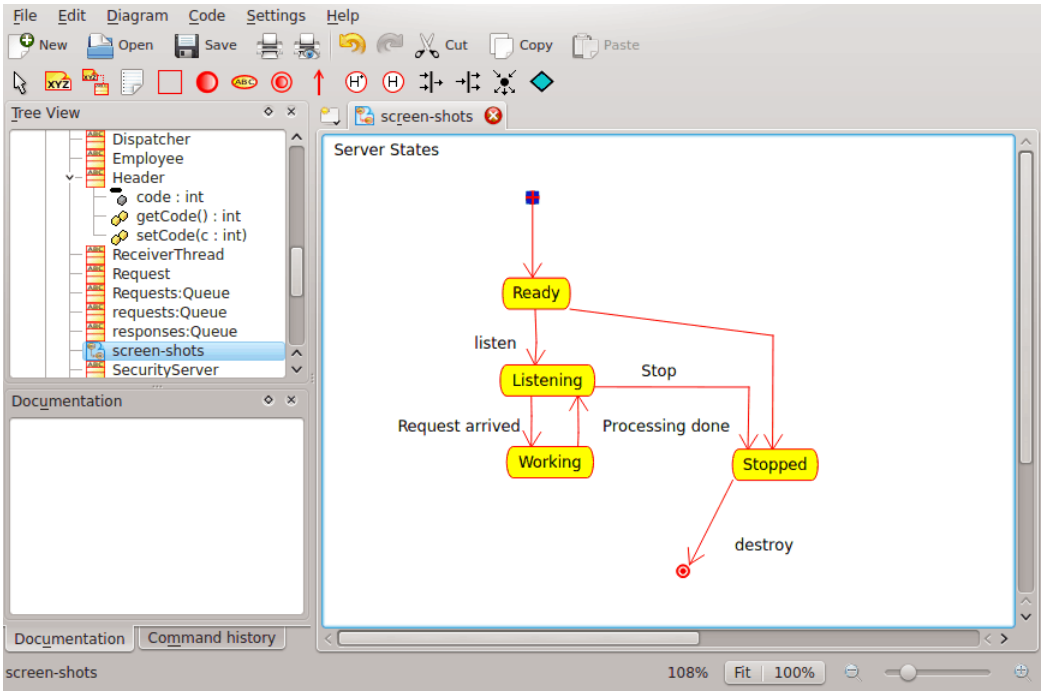
Toestandsdiagrammen kijken naar objecten als *toestandsmachines* of eindige automaten, die in een van een set eindige toestanden kunnen verkeren en hun toestand kunnen veranderen middels een van een eindige set stimuli. Bijvoorbeeld een object van het type *NetServer* kan in een van de volgende toestanden voorkomen tijdens zijn bestaan:

- Klaar
- Luisterend

- Werkend
- Gestopt

en de gebeurtenissen die ervoor kunnen zorgen dat het object van toestand verandert, zijn

- Object is aangemaakt
- Object ontvangt bericht "luisteren"
- Een cliënt vraagt een verbinding aan over het netwerk
- Een cliënt beëindigt een aanvraag
- De aanvraag is uitgevoerd en beëindigd
- Object ontvangt bericht "stoppen"
- etc



Umbrello UML Modeller toont een toestandsdiagram

2.2.5.1 Toestand

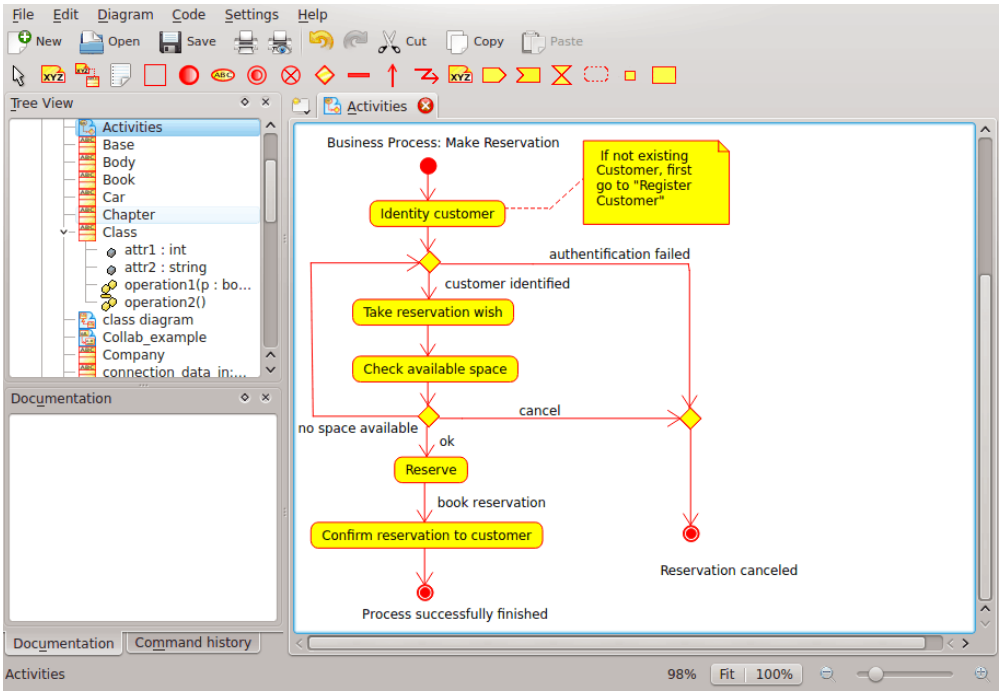
Toestanden zijn bouwstenen voor toestandsdiagrammen. Een toestand hoort bij exact één klasse en vertegenwoordigt een opsomming van waarden die de attributen van een klasse kunnen aannemen. Een UML-toestand beschrijft de interne toestand van een object van een bepaalde klasse

Merk op dat niet iedere verandering in een van de attributen van een object door een toestand gerepresenteerd dient te worden, maar slechts die veranderingen die significant de werking van het object kunnen beïnvloeden

Er zijn twee bijzondere toestandstypen: begin en eind. Zij zijn bijzonder omdat er geen gebeurtenis bestaat die ervoor kan zorgen dat een object terugkeert naar zijn begin-toestand, evenals er geen gebeurtenis bestaat die mogelijkerwijs een object uit zijn eind-toestand kan halen, wanneer het die eenmaal bereikt heeft.

2.2.6 Activiteitsdiagram

Activiteitsdiagrammen beschrijven de volgorde van aktiviteiten in een systeem net behulp van activiteiten. Aktiviteitsdiagrammen zijn een bijzondere vorm van toestandsdiagrammen, die alleen (of voornamelijk) aktiviteiten bevatten.



Umbrello UML Modeller toont een activiteitsdiagram

Activiteitsdiagrammen zijn vergelijkbaar met procedurele fluxdiagrammen, met dit verschil dat alle aktiviteiten duidelijk gekoppeld zijn aan objecten.

Activiteitsdiagrammen worden altijd geassocieerd met een *klasse*, een *operatie* of een *use case*.

Aktiviteitsdiagrammen ondersteunen zowel sequentiële als parallelle activiteiten. Parallelle uitvoering wordt gerepresenteerd m.b.v. pictogrammen voor afsplitsen/wachten, en voor de parallel verlopende activiteiten, is het niet van belang in welke volgorde zij worden uitgevoerd (zij kunnen tegelijkertijd uitgevoerd worden, of na elkaar)

2.2.6.1 Activiteit

Een activiteit is een enkele stap in een process. Een activiteit is een toestand in het systeem met interne activiteit en, ten minste, één uitgaande transitie. Activiteiten kunnen ook meer dan een uitgaande transitie bezitten als zij verschillende voorwaarden hebben.

Activiteiten kunnen hiërarchieën vormen, dit wil zeggen dat een activiteit samengesteld kan zijn uit meerdere 'detail'-activiteiten, in welk geval de in- en uitgaande transities zouden moeten overeenstemmen met de in- en uitgaande transities van het detaildiagram.

2.2.7 Hulpelementen

Er zijn een paar elementen in UML die geen reële semantische waarde voor het model, maar helpen bij het verhelderen van delen van het diagram. Deze elementen zijn

- Tekstregels

- Notities en ankers
- Vakken

Tekstregels zijn nuttig om korte tekstuele informatie aan een diagram toe te voegen. Het is vrijstaande tekst en heeft geen betekenis voor het model zelf.

Notities zijn nuttig om meer gedetailleerde informatie over een object of een specifieke situatie toe te voegen. Zij hebben het grote voordeel dat notities aan UML-elementen verankerd kunnen worden om te laten zien dat de notitie tot een specifiek object of situatie 'behoort'.

Vakken zijn vrijstaande rechthoeken die gebruikt kunnen worden om groepen onderdelen samen te voegen en zo de diagrammen leesbaarder te maken. Zij hebben geen logische betekenis voor het model.

2.2.8 Componentdiagrammen

Componentdiagrammen tonen de softwarecomponenten (ofwel component technologieën zoals KParts, CORBA-componenten of Java Beans ofwelsecties uit het systeem die duidelijk te onderscheiden zijn) en de artefacten waaruit zij gemaakt zijn zoals broncodebestanden, programmabibliotheken en relationele databasetabellen.

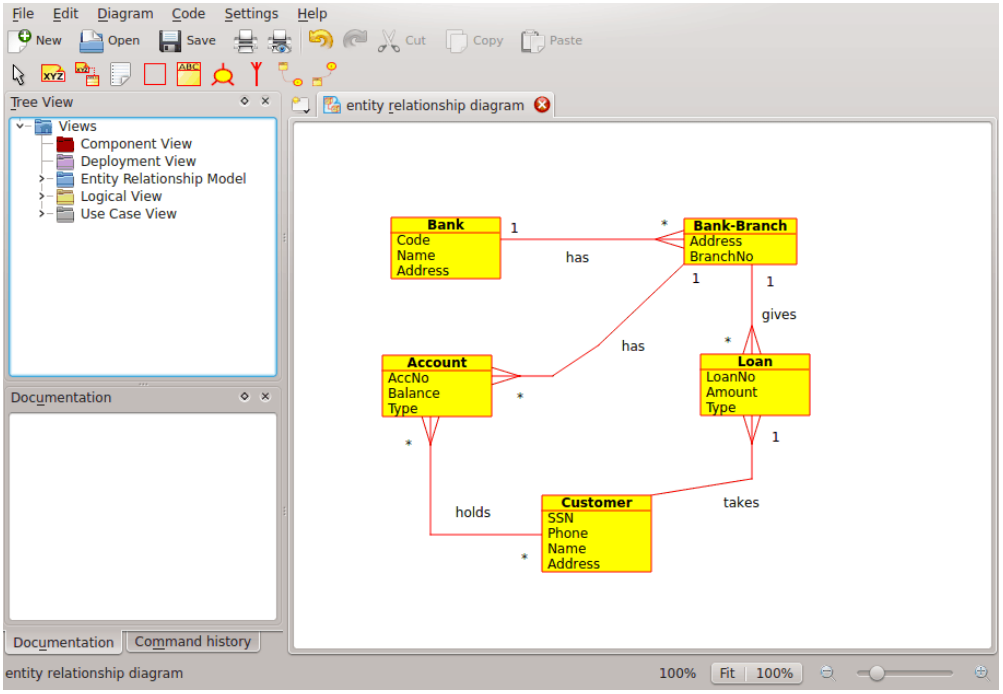
Componenten kunnen interfaces bezitten (bijv. abstracte klassen met operaties) die associaties tussen componenten toelaten.

2.2.9 Deploymentdiagrammen

Deploymentdiagrammen tonen de ″runtime″ componentinstanties en hun associaties. Zij omvatten nodes, die fysieke resources zijn, in het typische geval een enkele computer. Zij tonen ook interfaces en objecten (instanties van klassen).

2.2.10 Entiteitsrelatie-diagrammen

Entity Relationship Diagrams (ER Diagrams) tonen het conceptuele ontwerp van databasetoepassingen. Zij laten de verschillende entiteiten (concepten) in het informatiesysteem zien en de bestaande relaties en randvoorwaarden ertussen. Een extensie van Entity Relationship Diagrams genaamd 'Extended Entity Relationship Diagrams' of 'Enhanced Entity Relationship Diagrams' (EER), worden gebruikt om object georiënteerde ontwerptechnieken mee te nemen in ER Diagrams.



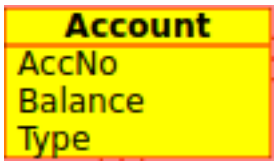
Umbrello toont een entiteitrelatiediagram

2.2.10.1 Entiteit

Een *Entiteit* is elk concept in de echte wereld met een onafhankelijk bestaan. Het kan een object zijn dat fysiek bestaat (voorbeeld, computer, robot) of het kan een object zijn met een conceptueel bestaan (bijv.: universitaire cursus). Elke entiteit heeft een set attributen die de eigenschappen van de entiteit beschrijven.

Opmerking: Er bestaat geen standaard notatie voor het weergeven van ER-diagrammen. Verschillende teksten over dit onderwerp gebruiken verschillende notaties. De concepten en notaties voor EER-diagrammen die in Umbrello worden gebruikt komen uit het volgende boek : *Elmasri R. and Navathe S. (2004). Fundamentals of Database Systems 4th edn. Addison Wesley*

In een ER-diagram worden entiteiten door rechthoeken gerepresenteerd, met de naam van de entiteit bovenaan. Hierin kunnen ook de attributen van de eintiteit in een andere 'vak' binnen de rechthoek.



Visuele representatie van een entiteit in een ER-diagram

2.2.10.1.1 Entiteitattributen

In ER-diagrammen, worden entiteitattributen getoond met hun naam in een apart vak van de entiteit waarbij hij behoort.

2.2.10.1.2 Voorwaarden

Voorwaarden in ER-diagramen specificeren de voorwaarden van gegevens in het informatieschema.

Er zijn vier typen ondersteunde voorwaarden in Umbrello :

- *Primaire sleutel:* De set attributen gedeclareerd als *primaire sleutel* zijn uniek voor de entiteit. Er kan maar één primaire sleutel in een entiteit zitten en geen van attributen die er onderdeel van zijn kan NULL zijn.
- *Unieke sleutel:* De set attributen gedeclareerd als *uniek* zijn uniek voor de entiteit. Er kunnen vele unieke voorwaarden aan een entiteit hangen. Zijn attributen die er onderdeel van zijn kunnen NULL zijn. Unieke sleutels en primaire sleutels identificeren een rij in een tabel (entiteit)
- *Vreemde sleutel:* Een vreemde sleutel is een referentie voorwaarde tussen twee tabellen. De vreemde sleutel identificeert een kolom of een set kolommen in één (verwijzende) tabel die verwijst naar een kolom of set kolommen in een andere (verwijzende) tabel. De kolommen in de verwijzende tabel moeten een primaire sleutel of unieke sleutel vormen.
- *Controle voorwaarde:* Een controle voorwaarde (ook bekend als tabel controle voorwaarde) is een voorwaarde die geldige gegevens definieert bij het toevoegen of bijwerken van een item in een tabel van een relationele database. Een controle voorwaarde wordt toegepast op elke rij in de tabel. De voorwaarde moet een voorspellend zijn. Het kan refereren naar een enkele of meerdere kolommen van de tabel.
  Voorbeeld: prijs >= 0

2.2.11 Concepten van Extended Entiteitrelatie-(EER)-diagram

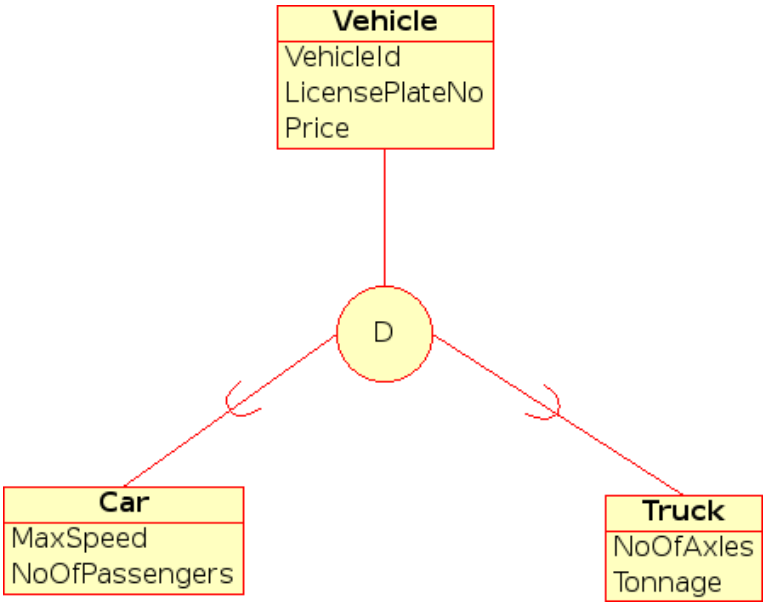
2.2.11.1 Specialisatie

Specialisatie is een manier om nieuwe entiteiten te vormen met gebruik van entiteiten die al gedefinieerd zijn. De nieuwe entiteiten, bekend als afgeleide entiteiten, nemen over (of erven) attributen van de vooraf-bestaande entiteiten, waarnaar gerefereerd wordt als basis entiteiten . Het is bedoeld om bestaande gegevens met weinig of geen modificatie.

In Umbrello, men kan losse en overlappende specialisatie

2.2.11.1.1 Losse specialisatie

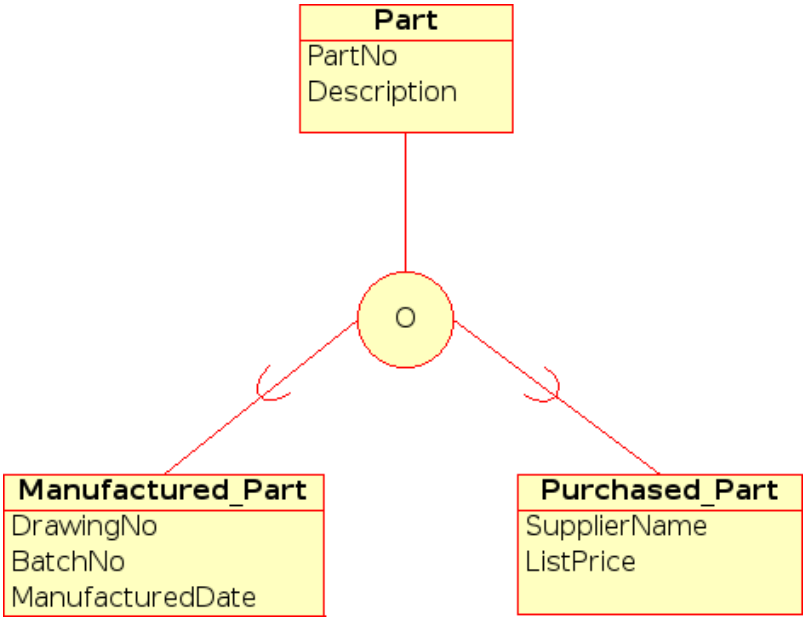
Disjoint Specialization (niet verbonden specialisatie) specificeert dat de subclasses van de specialisatie niet verbonden zijn. Dit betekent dat een entiteit een lid kan zijn van ten hoogste één van de afgeleide entiteiten van de specialisatie



Visuele representatie van losse specialisatie in EER-diagram

2.2.11.1.2 Overlappende specialisatie

Wanneer de afgeleide entiteiten niet voorwaarde bezitten om niet verbonden te zijn, dan betekent dat dat hun set entiteiten zich in overlappende specialisatie bevinden. Dit betekent dat dezelfde entiteit in de echte wereld lid kan zijn van meer dan één afgeleide entiteit van de specialisatie

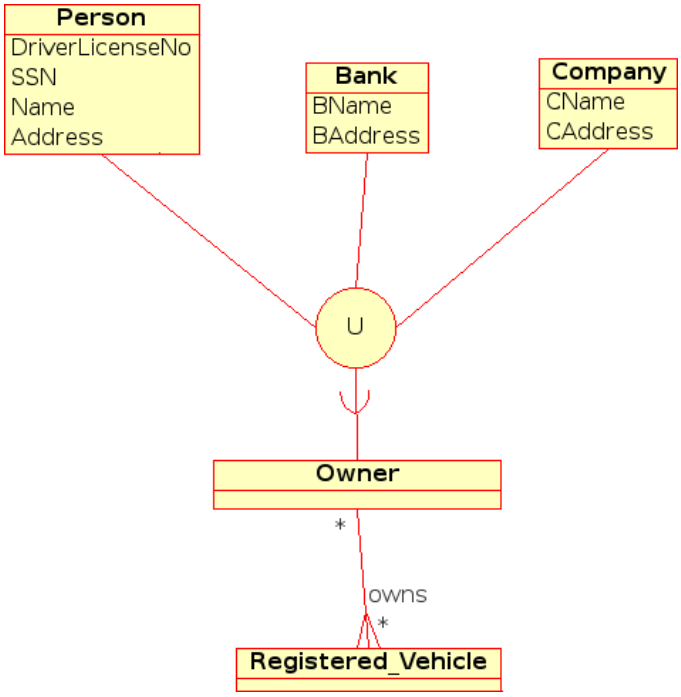


Visuele representatie van overlappende specialisatie in EER-diagram

2.2.11.1.3 Categorie

Een afgeleide entiteit behoort tot een *Categorie* wanneer het een verzameling objecten representeert die een subset zijn van vereniging van de onderscheidende entiteitstypen. Een categorie

wordt gemodelleerd wanneer het nodig is voor een enkele superclass/subclass relatie met meer dan één superclass, waar de superclasses verschillende entiteitstypen vertegenwoordigt. (Lijkt erg op overerving in object georiënteerd programmeren).



Visuele representatie van een categorie in EER-diagram

Hoofdstuk 3

Werken met Umbrello UML Modeller

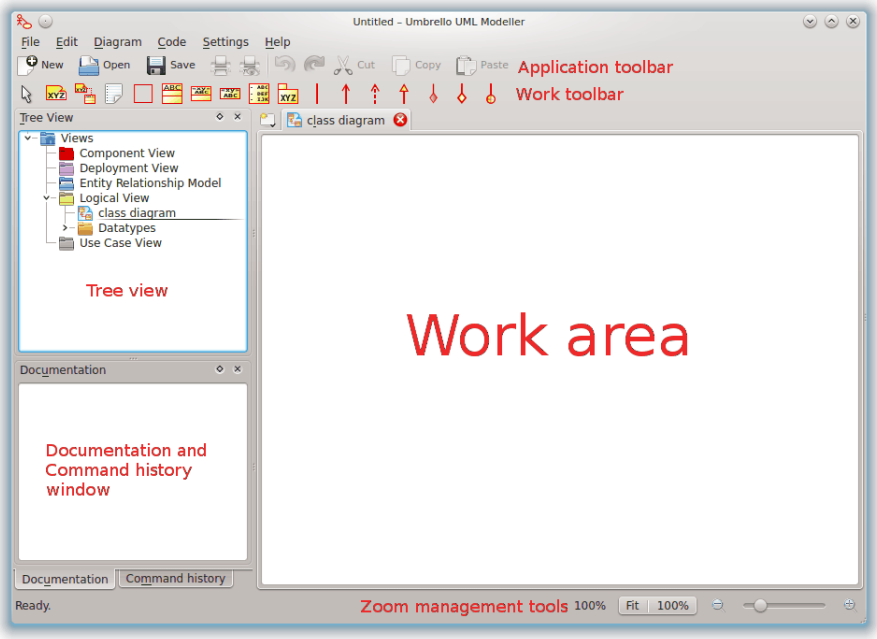
Dit hoofdstuk geeft u een inleiding in Umbrello UML Modeller's gebruikersinterface en vertelt u alles over wat u moet weten om een begin te maken met modelleren. Alle acties in Umbrello UML Modeller zijn toegankelijk via het menu en de werkbalken, hoewel Umbrello UML Modeller ook uitgebreid gebruik maakt van rechtermuisknop-contextmenus. U kunt met de rechtermuisknop klikken op vrijwel ieder element in Umbrello UML Modeller's werkblad of boomstructuur om een menu te krijgen met de handigste functies om toe te passen op het afzonderlijke element waar u op dat moment aan werkt. Sommige gebruikers vinden dit aanvankelijk enigszins verwarrend omdat zij er meer aan gewend zijn te werken met het menu of met werkbalken, maar als u er eenmaal aan gewend bent geraakt om rechts te klikken, dan zal uw werktempo aanzienlijk verhogen.

3.1 Gebruikersinterface

Umbrello UML Modeller's hoofdvenster is verdeeld in drie gebieden die u erbij helpen om zowel een overzicht te bewaren over uw gehele systeem als snel de verschillende diagrammen te benaderen terwijl u aan uw model werkt.

Deze gebieden heten:

- Boomstructuurweergave
- Werkblad
- Venster voor documentatie en geschiedenis van commando's



Umbrello UML Modeller's gebruikersinterface

3.1.1 Boomstructuurweergave

De boomstructuur bevindt zich gewoonlijk linksboven in het venster en toont alle diagrammen, klassen, actoren en use cases waaruit uw model is opgebouwd. De boomstructuur geeft u een snel overzicht van de samenstellende elementen van uw model. De boomstructuur maakt het u ook mogelijk om snel tussen de verschillende diagrammen in uw model te wisselen en elementen toe te voegen uit uw model in het actuele diagram.

Als u aan een model werkt met meer dan alleen maar een paar klassen en diagrammen, dan helpt de boomstructuur u om het overzicht te bewaren door de elementen van uw model in mappen te organiseren. U kunt mappen maken door de geeigende keuze uit het context menu te kiezen (klik met de rechtermuisknop op één van de mappen in de boomstructuur) en u kunt uw elementen organiseren door hen naar de van toepassing zijnde map te verplaatsen (slepen en neerzetten)

3.1.2 Venster voor documentatie en geschiedenis van commando's

Het venster voor documentatie en geschiedenis van commando's is het kleine venster linksonder in Umbrello UML Modeller, het geeft u een snel voorproefje van de documentatie voor het nu geselecteerde onderdeel en de geschiedenis van de commando's van uw werksessies. Het venster voor documentatie is tamelijk klein omdat het bedoeld is om u alleen maar een glimp van de documentatie van het element en een overzicht van de geschiedenis van de commando's te geven, waarbij het zo min mogelijk beeldschermruimte inneemt. Wilt u de documentatie in groter detail zien dan kunt u ten alle tijde de eigenschappen van het onderdeel openen.

3.1.3 Werkblad

Het Werkblad is het hoofdvenster in Umbrello UML Modeller waar de echte acties plaats vinden. U gebruikt het werkblad om te diagrammen in uw model te bewerken of te bekijken. Het werkblad toont het op dit moment actieve diagram. In de huidige versie kan op ieder moment slechts een diagram tegelijk getoond worden.

3.2 Modellen maken, laden en opslaan

Het eerste wat u moet doen om iets nuttigs met Umbrello UML Modeller te doen is het maken van een model waaraan u gaat werken. Als u Umbrello UML Modeller start, dan laadt het altijd het laatst gebruikte model of het maakt een nieuw, leeg model (dit hangt af van uw voorkeurinstellingen in het instellingenvenster). Dit stelt u in de gelegenheid direct aan de slag te gaan.

3.2.1 Nieuw model

Als u, op welk moment dan ook, een nieuw model moet maken, dan kunt u dit doen door de keuze **Nieuw** te selecteren uit het menu **Bestad** , of door te klikken op het pictogram **Nieuw** van de toepassingswerkbalk. Als u momenteel met een model bezig bent, dat is gewijzigd, dan zal Umbrello UML Modeller u vragen of het uw wijzigingen moet opslaan alvorens het nieuwe model te laden.

3.2.2 Model opslaan

U kunt op ieder moment uw model opslaan door de keuze **Opslaan** te selecteren uit het menu **Bestand** of door te klikken op de knop **Opslaan** uit de toepassingswerkbalk. Als u uw model onder een andere naam wilt opslaan, dan kunt u de keuze **Opslaan als** uit het menu **Bestand** gebruiken.

Voor uw gemak biedt Umbrello UML Modeller u ook de keuze om uw werk automatisch periodiek op te slaan. Als u er prijs op stelt dan kunt zowel deze keuze als het tijdsinterval instellen in **Instellingen** uit Umbrello UML Modeller

3.2.3 Model laden

Om een reeds bestaand model te laden, kunt u de keuze **Openen** selecteren uit het meun **Bestand** of het pictogram **Openen** aanklikken uit de toepassingswerkbalk. De meest recent gebruikte modellen zijn eveneens beschikbaar onder het submenu **Openen recent** van het menu **Bestand** om zo de toegangssnelheid tot uw meest gebruikte modellen te vergroten.

Umbrello UML Modeller kan slechts met één model tegelijk werken, dus als u het programma vraagt om een model te laden, en uw actuele model is gewijzigd nadat u het voor de laatste keer bewaarde, zal Umbrello UML Modeller uvragen of uw wijzigingen bewaard moeten worden om te voorkomen dat u werk kwijt raakt. U kunt op ieder moment twee of meer instanties van Umbrello UML Modeller starten, u kunt ook kopiëren en plakken tussen instanties.

3.3 Modellen bewerken

In Umbrello UML Modeller, zijn in principe twee manieren om de elementen in uw model te bewerken.

- Modelelementen direct bewerken via de boomstructuur
- Modelelementen bewerken via een diagram

Door gebruik te maken van de contextmenu's van de verschillende onderdelen in de boomstructuur bent u in staat om vrijwel alle elementen aan uw model toe te voegen, uit uw model te verwijderen en in uw model te wijzigen. Rechts klikken op de mappen in de boomstructuur geeft u de keuze om niet alleen de verschillende soorten diagrammen te maken maar ook, afhankelijk van of de gekozen map een *use case-weergave* of een *logische weergave* is, actoren, use cases, klassen enz..

Heeft u eenmaal elementen aan uw model toegevoegd, dan kunt u een element ook bewerken door middel van diens eigenschappenvenster. Dat kunt u vinden door de keuze *Eigenschappen* te selecteren uit het contextmenu dat verschijnt als u rechts klikt op de onderdelen in de boomstructuur.

U kunt uw model ook bewerken door elementen te maken of te wijzigen door middel van diagrammen. Meer details over hoe u dit moet doen worden in de volgende secties gegeven.

3.4 Diagrammen toevoegen en verwijderen

Uw UML-model bestaat uit een set UML elementen met hun onderlinge associaties. Alleen kunt u het model niet direct zien, u kijkt ernaar d.m.v. *diagrammen*.

3.4.1 Diagrammen maken

Om een nieuw diagram in uw model te maken selecteert u gewoon de diagramsoort die u nodig heeft uit het submenu **Nieuw** submenu van het menu **Diagram** en geeft u het een naam. Het diagram wordt gemaakt en wordt geactiveerd, en u krijgt het direct te zien in de boomstructuur.

Onthoud dat Umbrello UML Modeller uitgebreid gebruik maak van contextmenu's: u kunt evengoed rechtermuisknop klikken op een map in de boomstructuur en de juiste diagramsoort uit het submenu **Nieuw** van het contextmenu selecteren. Merk op dat u alleen use case diagrammen kunt maken in mappen van een use case-weergave, en de andere soorten diagrammen alleen in de mappen van de logische weergave.

3.4.2 Diagrammen verwijderen

Doet het zich voor dat u een diagram uit uw model moet verwijderen, dan doet u dit door het te activeren en **Verwijderen** te selecteren in het menu **Diagram**. U kunt dit ook bereiken door **Verwijderen** te selecteren in het diagrammen-contextmenu in de boomstructuur.

Omdat het verwijderen van een diagram een serieuze aangelegeheid is waarbij u per ongeluk werk kunt verliezen, vraagt Umbrello UML Modeller u om bevestiging van de verwijdereringsoperatie alvorens daadwerkelijk het diagram te verwijderen. Is het diagram eenmaal verwijderd dan is de file opgeslagen, dan kan deze actie op geen wijze ongedaan gemaakt worden.

3.4.3 Diagrammen hernoemen

Als u de naam van een bestaand diagram wilt wijzigen dan kunt u dit gemakkelijk doen door de optie ″Hernoemen″ uit diens rechtermuisknop menu te selecteren in de boomstructuur.

Een andere manier om een diagram te hernoemen is dit te doen via zijn eigenschappenvenster. U krijgt deze als u ″Eigenschappen″ uit zijn contextmenu selecteert of door erop dubbel te klikken in de boomstructuur.

3.5 Diagrammen bewerken

Bij het werken met een diagram probeert Umbrello UML Modeller u te assisteren door enkele eenvoudige regels toe te passen met betrekking tot toegestane elementen in de verschillende soorten diagrammen, en de relaties die tussen hun kunnen voorkomen. Als u een UML-expert bent dan zal u dit waarschijnlijk niet eens merken, maar dit helpt UML-nieuwelingen om aan de standaard voldoende diagrammen te maken.

Als u eenmaal uw diagrammen heeft gemaakt, dan wordt het tijd om ze te bewerken. Hier zou u het (voor beginners subtiele) verschil moeten zien tussen uw diagram bewerken, en het *model* bewerken. Zoals u reeds weet, zijn diagrammen *weergaven*van uw model. Bijvoorbeeld, als u een klasse maakt door een klassediagram te bewerken, dan bewerkt u in feite beide, uw diagram en uw model. Als u de kleur of andere weergavemogelijkheden van een klasse in uw klassediagram wijzigt, dan bewerkt u alleen uw diagram, en blijft uw model onveranderd.

3.5.1 Elementen toevoegen

Een van de eerste dingen die u zult doen wanneer u een nieuw diagram bewerkt bestaat uit het eraan toevoegen van elementen (klassen, actoren, use cases, etc.) Er zijn in principe twee manieren om dit te doen.

- Bestaande elementen naar uw model slepen uit de boomstructuur
- Nieuwe elementen in uw model maken en ze tegelijkertijd toevoegen aan uw diagram, door middel van de bewerkingsmiddelen in de gereedschappenbalk

Om reeds bestaande elementen toe te voegen aan uw model, hoeft u ze alleen maar uit de boomstructuur te slepen en ze te laten vallen op het diagram daar waar u ze wilt hebben. U kunt altijd elementen verplaatsen in uw diagram met het selectiehulpmiddel.

De tweede manier om elementen aan uw diagram toe te voegen bestaat uit het toepassen van de bewerkingsmiddelen in de gereedschappenbalk (merk op dat dit tegelijkertijd elementen aan uw model toevoegt).

De hulpmiddelenbalk bevond zich standaard aan de bovenkant van het venster. De in deze hulpmiddelenbalk beschikbare hulpmiddelen (de knoppen die u erop ziet) veranderen afhankelijk van het soort diagram waar u op het moment mee bezig bent. De knop voor het geselecteerde hulpmiddel is geactiveerd in de hulpmiddelenbalk. U kunt omschakelen naar het hulpmiddel voor selectie door op de toets **Esc** te drukken.

Wanneer u een bewerkingsgereedschap heeft geselecteerd uit de gereedschappenbalk (bijvoorbeeld, het gereedschap om klassen toe te voegen) dan verandert de muisaanwijzer in een kruis, en kunt u de elementen aan uw model toevoegen met een enkele muisklik in uw diagram. Merk op dat elementen in UML een *unieke naam* moeten hebben. Dus als u in een diagram een klasse heeft waarvan de naam 'KlasseA' is, en u gebruikt het klasse-toevoegen gereedschap om de klasse aan een ander diagram toe te voegen, kunt u niet deze nieuwe klasse eveneens 'KlasseA' noemen. Als verondersteld wordt dat deze beide twee verschillende elementen moeten zijn, dan dient u ze een unieke naam te geven. Als u wilt proberen om *hetzelfde* element aan uw diagram toe te voegen, dan is ″Klasse toevoegen″ niet het daartoe geschikte gereedschap. In plaats hiervan kunt u beter de klasse uit de boomstructuur verslepen.

3.5.2 Elementen verwijderen

U kunt ieder element verwijderen door de keuze **Verwijderen** te selecteren in het contextmenu.

Weer is er een *groot* verschil tussen een object uit een diagram verwijderen, en object uit uw model verwijderen: Als u een object uit uw diagram verwijdert, dan verwijdert u alleen dat object uit dat ene specifieke diagram: het element blijt nog steeds deel uitmaken van uw model model en als er andere diagrammen zijn die hetzelfde element gebruiken, dan zullen zij geen enkele wijziging krijgen. Als u daarentegen, het element verwijdert uit de boomstructuur, dat verwijdert u feitelijk het element uit uw *model*. Als derhalve het element niet langer in uw model voorkomt, dan wordt het automatisch verwijderd uit alle diagrammen waar het in voorkomt.

3.5.3 Elementen bewerken

De meeste van de UML-elementen in uw model en diagrammen kunt u bewerken door hun eigenschappenvenster te openen en de van toepassing zijnde mogelijkheden te selecteren. Om de eigenschappen van een object te bewerken, selecteert u **Eigenschappen** in zijn contextmenu (rechtermuisknop-klik). Ieder element heeft een dialoog die uit meerdere pagina's bestaat waarin u de instellingen van dat element kunt aanpassen. Voor sommige elementen, zoals actoren, kunt u slechts een paar instellingen, zoals de naam van het object en documentatie, een waarde geven, terwijl voor andere elementen, zoals klassen, u de attributen en operaties kunt bewerken, en selecteren wat er volgens u in het diagram getoond moet worden (complete operatiesignatuur of alleen namen van operaties, enz.), en zelfs de kleuren die u wilt gebruiken voor de lijnen en opvulling voor de klasserepresentatie in het diagram.

Voor UML-elementen kunt u het eigenschappenvenster ook openen door er dubbel op te klikken wanneer u het selectiegereedschap (pijl) gebruikt.

Merk op dat u de eigenschappen-menukeuze ook kunt selecteren in met contextmenu van de elementen in de boomstructuur. Hiermee kunt u ook de eigenschappen van de diagrammen bewerken, zoals de instelling of er al dan niet een rooster getoond moet worden.

3.5.4 Klassen bewerken

Alhoewel het bewerken van eigenschappen van alle objecten al besproken was in de vorige sectie, verdienen klassen een toegewijde sectie omdat zij wat gecompliceerder zijn en meer instelmogelijkheden hebben dan de meeste andere UML-elementen.

In het eigenschappenvenster van een klasse kunt van alles instellen, vanaf de kleur die het gebruikt tot en met zijn operaties en attributen.

3.5.4.1 Algemene klasse-instellingen

De algemene instelpagina van het eigenschappenvenster spreekt voor zich. Hier kunt u de naam van de klasse veranderen, de zichtbaarheid, documentatie, etc.. Deze pagina is altijd beschikbaar.

3.5.4.2 Klasse-attributen instellen

In de pagina voor attributeninstellingen kunt u klasse-attributen (variabelen) toevoegen, bewerken, of verwijderen. U kunt attributen omhoog en omlaag door de lijst verplaatsen met de pijlknoppen aan de zijkant. Deze pagina is altijd beschikbaar.

3.5.4.3 Klasse-operaties instellen

Net als in de pagina voor attributeninstellingen, kunt u in de pagina voor operatie-instellingen, klasse-operaties toevoegen, bewerken, of verwijderen. Bij het toevoegen en bewerken van een operatie, geeft u de basisgegevens op in het venster *Operatie-eigenschappen*. Als u parameters aan uw operatie wilt toevoegen, dan dient u op de knop **Nieuwe parameter** te klikken, waardoor het venster *Parameter-eigenschappen* getoond wordt. Deze pagina is altijd beschikbaar.

3.5.4.4 Klasse-sjabloon instellen

Met deze pagina kunt klasse-sjablonen toevoegen. Dit zijn niet gespecificeerde klassen of gegevenstypen. In Java 1.5 gaan deze generics heten.

3.5.4.5 De pagina "Klasse-associaties"

De pagina **Klasse-associaties** toont alle associaties van deze klasse in het huidige diagram. Dubbel klikken op een associatie laat zijn eigenschappen zien, en afhankelijk van de soort associatie kunt u hier enkele parameters wijzigen zoals instellen van menigvuldigheid en de naam van de rol. Als de associatie niet toelaat om dergelijke keuzemogelijkheden te wijzigen, dan geldt voor dit associatie-eigenschappenvenster alleen-lezen, en kunt u alleen de aan deze associatie gekoppelde documentatie wijzigen.

Deze pagina is alleen beschikbaar als u de klasse-eigenschappen vanuit een diagram opent. Als u de klasse-eigenschappen vanuit het contextmenu in de boomstructuur opent, dan is deze pagina niet beschikbaar.

3.5.4.6 De pagina "Klasse-weergave"

In de pagina **Weergave keuzemogelijkheden** kunt u instellen wat er in het diagram weergegeven moet worden. Een klasse kan weergegeven worden als alleen een rechthoek met de naam van de klasse erin (handig als u veel klassen in uw diagram heeft, of op het moment niet in de details van iedere klasse geïnteresseerd bent) of zo compleet als maar mogelijk met pakketen, stereotypen, attributen en operaties met volledige signatuur en zichtbaarheid

Afhankelijk van de hoeveelheid informatie die u wilt zien, kunt u in deze pagina de corresponderende keuzemogelijkheden selecteren. De wijzigingen die u hier maakt zijn slechts *weergave-keuzemogelijkheden* voor het diagram. Dit houdt in dat 'verbergen' van de klasse-operaties alleen tot gevolg heeft dat zij niet in het diagram getoond worden, en de operaties er nog steeds zijn als onderdeel van uw model. Deze keuzemogelijkheid is alleen beschikbaar als u de klasse-eigenschappen vanuit een diagram selecteert. Opent u echter klasse-eigenschappen vanuit de boomstructuur, dan ontbreekt deze pagina aangezien dergelijke keuzemogelijkheden in dat geval geen zin hebben.

3.5.4.7 Pagina "Klasse-stijl"

In de pagina **Widgetstijl** kunt u de kleuren instellen die u voor de lijnen en opvulling voor de widget wenst. Deze keuzemogelijkheid heeft overduidelijk alleen maar zin voor klassen in diagrammen, en ontbreekt wanneer u het klasse-eigenschappenvenster opent vanuit de boomstructuur.

3.5.5 Associaties

Associatie relateren twee UML objecten aan elkaar. Normaal worden associaties gedefinieerd tussen twee klassen, maar sommige soorten associaties kunnen ook voorkomen tussen use cases en actoren.

Om een associatie te maken selecteert u het geëigende gereedschap in degereedschappenbalk (generieke associatie, generalizatie, aggregatie, etc.). Vervolgens klikt u eenmaal op het eerste element dat deelneemt aan de associatie, en eenmaal op het tweede element dat eraan deelneemt. Merk op dat dit twee klikken zijn, eenmaal op elk van de objecten die aan de associatie deelnemen, het is *geen* slepen van het ene object naar het andere.

Als u probeert om een associatie aan te leggen tegen de regels van de UML-specificatie dan zal Umbrello UML Modeller weigeren de associatie te maken en krijgt u een foutmelding. Dit kan het geval zijn als, bijvoorbeeld, er een generalisatie bestaat van klasse A naar klasse B en u vervolgens probeert om een andere generalisatie van klasse B naar klasse A te maken.

Rechts klikken op een associatie brengt een contextmenu naar voren met de acties die u erop kunt toepassen. Als u een associatie wilt verwijderen, selecteer dan de optie **Verwijderen** uit dit contextmenu. U kunt ook de optie **Eigenschappen** selecteren, en afhankelijk van de soort associatie, de attributen, zoals rol en multipliciteit, bewerken.

3.5.5.1 Ankerpunten

Standaard worden associaties getekend als een rechte lijn die de twee objecten in het diagram met elkaar verbindt.

U kunt ankerpunten toevoegen om een associatie te buigen. Dubbelklik hiertoe ergens op de associatielijn. Dit voegt een ankerpunt toe (weergegeven als een blauwe punt wanneer de associatielijn geselecteerd wordt) welke u overal naartoe kunt verplaatsen om de associatie van vorm te veranderen.

Als u een ankerpunt wilt verwijderen, dubbelklik er dan weer om het te verwijderen

Merk op dat de enige manier om de eigenschappen van een associatie te bewerken, via het contextmenu loopt. Er op dubbelklikken, zoals bij andere UML-objecten, heeft alleen tot gevolg dat er een ankerpunt toegevoegd wordt.

3.5.6 Notities, tekst en vakken

Notities, tekstregels en vakken zijn elementen die in ieder soort diagram kunnen voorkomen en geen wezenlijke semantische waarde hebben. Maar, zij zijn zeer nuttig om extra commentaar of uitleg toe te voegen, waardoor uw diagram gemakkelijker te begrijpen wordt.

Om een notitie of een tekstregel toe te voegen, selecteert u het corresponderende gereedschap in de gereedschappenwerkbalk en klikt u eenmaal op het diagram waar u uw commentaar wilt hebben. U kunt de tekst bewerken door het element te openen via zijn contextmenu of in het geval van notities evengoed door er op te dubbelklikken.

3.5.6.1 Ankers

Ankers worden gebruikt om een tekstaantekening en een UML element aan elkaar te koppelen. Bijvoorbeeld, u gebruikt gewoonlijk een tekstaantekening om betreffende een klasse of een of andere associatie, iets te verklaren of er enig commentaar aan toe te voegen, in zo'n geval kunt u een anker gebruiken om aan te geven dat de notitie tot dat ene element 'behoort'.

Om een anker aan een notitie en een UML-element toe te voegen, gebruikt u het ankergereedschap uit de gereedschappenbalk. U dient eerst op de notitie te klikken en vervolgens op het UML-element waar u de notitie aan wilt verbinden.

Hoofdstuk 4

Code-import en code-generatie

Umbrello UML Modeller is een UML modelling gereedschap en als zodanig is zijn voornaamste doel u te helpen bij de *analyse en het ontwerp* van uw systemen. Echter, om de overgang van uw ontwerp naar uw *implementatie* te maken, kunt u met Umbrello UML Modeller broncode genereren in verschillende programmeertalen om u van start te laten gaan. Ook, als u UML wilt gaan toepassen in een reeds gestart C++ project, kan Umbrello UML Modeller u helpen bij het maken van een model van uw systeem op basis van uw broncode door deze te analyseren en de daarin gevonden klassen te importeren.

4.1 Code-generatie

Umbrello UML Modeller kan broncode genereren voor talrijke programmeertalen op basis van uw UML model om u te helpen van start te gaan met de implementatie van uw project. De gegenereerde code bestaat uit klasse declaraties, met hun methoden en attributen, dus u kunt de 'blanke ruimte invullen' door de functionaliteit van uw klassen-operaties te verschaffen.

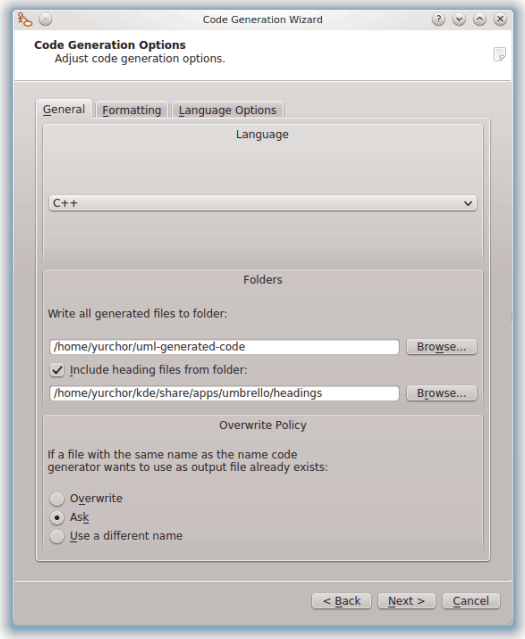
Umbrello UML Modeller 2 komt met code-generatie ondersteuning voor ActionScript, Ada, C++, C#, D, IDL, Java™, JavaScript, MySQL, Pascal, Perl, PHP, PHP5, PostgreSQL, Python, Ruby, Tcl, Vala en XMLSchema.

4.1.1 Code genereren

Om code te kunnen genereren met Umbrello UML Modeller, dient u eerst een model te maken, of er een te laden, met minimaal één klasse erin. Als u gereed bent om te beginnen met het schrijven van enigerlei code, dan selecteert u de **Code Generatie Wizard** menukeuze in het **Code** menu om een assistent te starten die u door het code-generatie proces zal loodsen.

De eerste stap is het selecteren van de klassen waarvoor u broncode wilt genereren. Standaard worden alle klassen van uw model geselecteerd, selected, en u kunt diegene, waarvoor u geen code wilt genereren, verwijderen, door ze naar lijst aan de linkerkant te verplaatsen.

In de volgende stap van de assistent kunt u de parameters wijzigen welke de codegenerator gebruikt bij het schrijven van uw code. De volgende keuzemogelijkheden zijn beschikbaar:



Keuzemogelijkheden voor de codegeneratie in Umbrello UML Modeller

4.1.1.1 Generatie keuzemogelijkheden

4.1.1.1.1 Uitvoerigheid commentaar

De keuzemogelijkheid **Schrijf documentatie commentaar zelfs indien leeg** instrueert de codegenerator om commentaar van de vorm /** blah */ te schrijven, zelfs als de commentaarblokken leeg zijn. Als u documentatie heeft toegevoegd aan uw klassen, methoden en attibuten in uw model, dan zal de codegenerator deze commentaren als Doxygen documentatie wegschrijven, ongeacht wat u daar formuleerde, maar als u deze keuzemogelijkheid selecteert dan zal Umbrello UML Modeller commentaarblokken schrijven voor alle klasses, methoden en attributen, zelfs wanneer er geen documentatie in het model is, in welk geval u uw klassen later dient te documenteren, rechtstreeks in de broncode.

Schrijf commentaar voor secties zelfs indien sectie leeg is zorgt ervoor dat Umbrello UML Modeller commentaar schrijf in de broncode om de verschillende secties van en klasse af te bakenen. Bijvoorbeeld 'public methoden' of 'Attributen' voor de corresponderende sections. Als u deze keuzemogelijkheid selecteert, dan zal Umbrello UML Modeller commentaar wegschrijven voor alle klasse-secties ook als de sectie leeg is. Bijvoorbeeld, het zou commentaar met de tekst 'protected methoden' wegschrijven ook als er geen protected methoden in uw klasse zijn.

4.1.1.1.2 Mappen

Schrijf alle gegenereerde files naar map. Hier dient u de map te selecteren waarin Umbrello UML Modeller de gegenereerde bronnen van u moet plaatsen.

De **Voeg heading-files toe uit map** keuzemogelijkheid geeft u de mogelijkheid om een heading aan het begin van iedere gegenereerde file toe te voegen. Headingfiles kunnen auteursrecht of licentie informatie bevatten maar ook variabelen die op het moment van genereren geëvalueerd worden. U kunt eens kijken naar het sjabloon headingfiles zoals die met Umbrello UML Modeller meegeleverd worden om te zien hoe dit soort variabelen gebruikt moeten worden, om uw naam en/of de huidige datum te vervangen ten tijde van het genereren.

4.1.1.1.3 Overschrijvingsprotocol

Deze keuzemogelijkheid vertelt tegen Umbrello UML Modeller wat het moet doen als het bestand dat het wil maken, reeds in de bestemmingsmap voorkomt. Umbrello UML Modeller *kan bestaande bestanden niet wijzigen*, derhalve moet u een keuze maken uit: of het bestaande bestand overschijven, of het genereren van dat ene aparte bestand overslaan, of Umbrello UML Modeller een andere bestandsnaam laten kiezen.Als u kiest voor de mogelijkheid van een andere bestandsnaam, dan zal Umbrello UML Modeller de bestandsnaam van een achtervoegsel voorzien.

4.1.1.1.4 Taal

Umbrello UML Modeller zal standaard code genereren in de taal die u als actieve taal heeft geselecteerd, echter met de codegenererings-assistent heeft u de keuzemogelijkheid om dit naar een andere taal te veranderen.

4.1.1.2 Genererings-assistent genereren

De derde en laatste stap van de assistent toont de status van het codegenererings proces. U hoeft alleen nog maar op de knop genereren te klikken om de klassen voor u te laten wegschrijven.

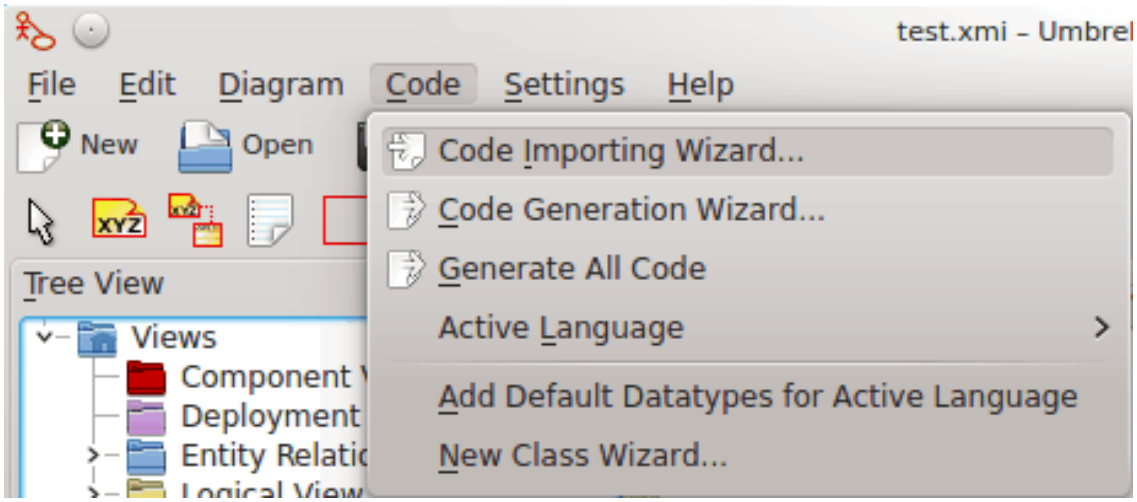
Merk op dat de keuzemogelijkheden die u selecteert in de codegenererings-assistent alleen gelden voor de huidige generering. Wanneer u de assistent een volgende keer gebruikt, dan moet u alle keuzemogelijkheden opnieuw selecteren (de map voor uw headers, overschrijvings protocol, enz.). U kunt de standaarden die Umbrello UML Modeller aamhoudt, instellen in de sectie **code genereren** van de Umbrello UML Modeller settings, beschikbaar in **Instellingen** → **Umbrello UML Modeller instellen...**

Als u uw codegenererings keuzemogelijkheden op de juiste waarden heeft ingesteld, en u zomaar enige code wilt genereren zonder door de assistent heen te moeten gaan, dan selecteert u het **Genereer alle code** in het **code**-menu. Dit zal code genereren voor alle klassen in uw model met gebruikmaking van de huidige instellingen (inclusief uitvoermap en overschrijvings-protocol, dus pas op met het gebruik ervan).

4.2 Code import

Umbrello UML Modeller kan broncode importeren vanuit uw bestaande projecten om u te helpen met het construeren van een model voor uw systemen. Umbrello UML Modeller 2 ondersteunt ActionScript, Ada, C++, C#, D, IDL, Java™, Javascript, MySQL en Pascal, PHP en Vala-broncode.

Om klasses in uw model te importeren, selecteert u de menukeuze **Assistent voor importeren van broncode...** uit het menu **Code** menu. Selecteer in de bestandsdialoog de bestanden die klasse-declaraties bevatten en druk op **Volgende >** daarna op **Importeren starten** en **Beëindigen**. De klasses worden nu geïmporteerd en u zult ze aantreffen als deel van uw model in de boomstructuur. Merk op dat Umbrello UML Modeller geen enkel soort diagram aanmaakt om uw klassen te tonen, ze worden alleen in uw model geïmporteerd opdat u ze naderhand in welk diagram dan ook, kunt gebruiken.



Menu om broncode te importeren in Umbrello UML Modeller

Hoofdstuk 5

Andere mogelijkheden

5.1 Andere Umbrello UML Modeller mogelijkheden

Dit hoofdstuk beschrijft kort enkele andere mogelijkheden van Umbrello UML Modeller.

5.1.1 Objecten als PNG-afbeeldingen kopieëren

Los van het gebruikelijke kopiëren, knippen en plakken om objecten te kopiëren tussen verschillende diagramme, kan Umbrello UML Modeller objecten ook kopiëren als PNG-afbeeldingen zodat u ze kunt invoegen in een ander type document. U hoeft niets byzonders te doen hiervoor. Selecteer simpelweg een object van een diagram (Class, Actor, etc.) en kopieer deze (**Ctrl-C**, of gebruik het menu), open vervolgens een Calligra Words-document (of een ander programma waarin je afbeeldingen kunt plakken) en selecteer **Plakken**. Op deze wijze kunt u sommige delen van uw diagram als simpele afbeeldingen hergebruiken.

5.1.2 Exporteren als een afbeelding

U kunt ook een compleet diagram exporteren als afbeelding. Het enige dat u hiervoor hoeft te doen is om het diagram te selecteren dat u wilt exporteren en vervolgens de menu-ingang **Als afbeelding exporteren...** te gebruiken uit het menu **Diagram**.

U kunt meerdere diagrammen tegelijk exporteren met de optie **Diagrammen als afbeeldingen exporteren...** uit het menu **bestand**. Hiermee kunt u ook de resolutie van de afbeelding instellen, zodat de afbeeldingen niet erg wazig zullen zijn.

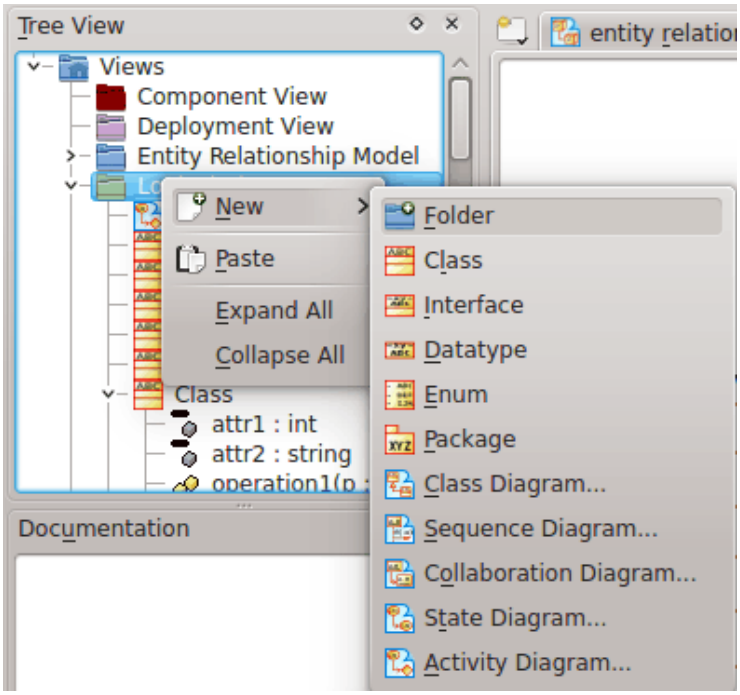
5.1.3 Afdrukken

Umbrello UML Modeller maakt het mogelijk om individuele diagrammen af te drukken. Druk op het icoon **Afdrukken** in de werkbalk of selecteer **Afdrukken** uit het menu **Bestand**. U krijgt dan het standaard KDE Afdrukken-venster, vanwaar u uw diagram kunt afdrukken.

5.1.4 Logische mappen

Om uw model in te delen, met name in grotere projecten, kunt u logische mappen aanmaken in de boomstructuur. Om ze te maken, selecteert u de keuzemogelijkheid **Nieuw** → **Map** in het contextmenu van de standaard mappen in de boomstructuur. Mappen kunnen genest zijn, en u

kunt objecten heen en weer verplaatsen door ze van de ene map weg te slepen en ze in de andere te laten vallen.

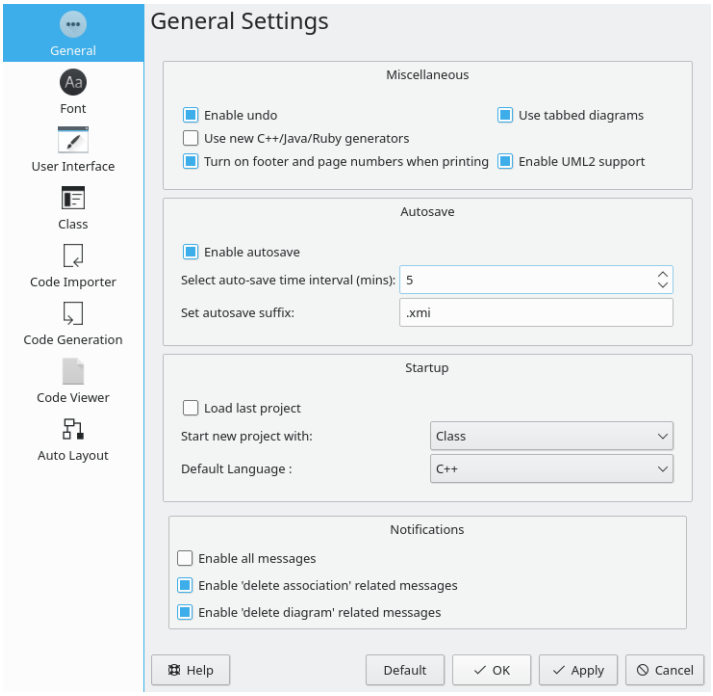


Uw model indelen met logische mappen in Umbrello UML Modeller

Hoofdstuk 6

Instellingen

6.1 Algemene instellingen



Opties voor Algemene instellingen in Umbrello UML Modeller

6.1.1 Diversen

- De optie **Voorgaande bewerking opnieuw doen inschakelen** biedt het ongedaan maken van een vorige actie.
- **Nieuwe C++/Java/Ruby-generatoren gebruiken** laat de gebruiker ofwel de oude of nieuwe codegeneratoren selecteren
- **Inschakelen van voettekst en paginanummer bij afdrukken** indien geselecteerd, drukt diagraminformatie af voor het diagram dat wordt afgedrukt en het paginanummer.

- **Getabte diagrammen gebruiken** geeft de optie om meerdere diagramvensters in tabbladen tegelijk open te hebben.

6.1.2 Automatisch opslaan

- **Automatisch opslaan activeren** biedt een keuze om het bestand automatisch op te slaan.
- **Selecteer interval voor automatisch opslaan (in minuten):** biedt het instellen van de tijd voordat het bestand automatisch wordt opgeslagen.
- **Achtervoegsel bij automatisch opslaan instellen:** is standaard .xmi maar biedt een andere instelling voor bestandsextensie.

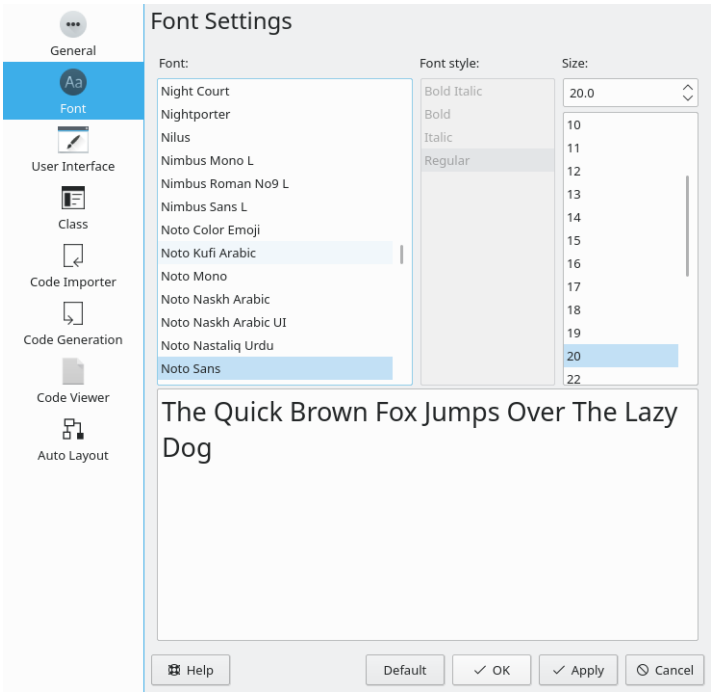
6.1.3 Opstarten

- **Laatste project laden** indien ingesteld, laadt altijd het laatste werkproject bij opstarten van project.
- **Nieuw project starten met:** geeft een keuze van met welk type UML-diagram te beginnen in een nieuw project.
- **Standaard taal:** is een instelling voor de standaard te gebruiken programmeertaal.

6.1.4 Meldingen

- **Alle meldingen inschakelen** is een optie om ofwel alle meldingen of een gereduceerde set meldingen te zien.
- **Meldingen gerelateerd aan 'associatie verwijderen' inschakelen** maakt zeker dat u alle meldingen van dit type zult ontvangen indien geactiveerd.
- **Aan 'diagram verwijderen' gerelateerde meldingen inschakelen** zal alle meldingen van dit type inschakelen indien geactiveerd.

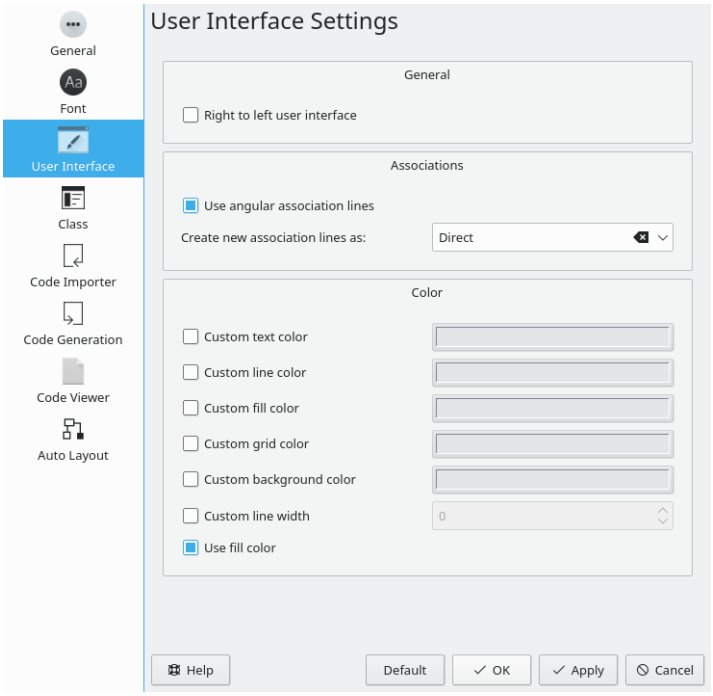
6.2 Lettertype-instellingen



Opties voor instellingen voor lettertypen in diagrammen in Umbrello UML Modeller

Deze instellingen voor lettertypen stelt de karakteristieken in van de tekst in de diagrammen. Stijl van het lettertype en grootte zijn de enige te selecteren opties.

6.3 Instellingen voor de gebruikersinterface



Instellingen voor de gebruikersinterface in Umbrello UML Modeller

6.3.1 Algemeen

Interface voor rechts naar links configureert het interface voor de rechts naar links talen.

6.3.2 Associaties

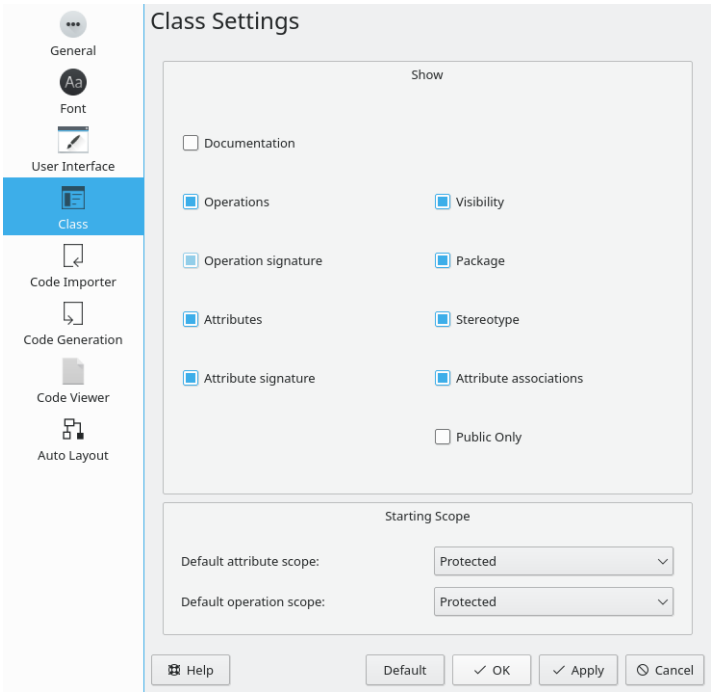
Hoekige associatielijnen gebruiken biedt associatielijnen met een willekeurig variabele hoek.

Nieuwe associatielijnen aanmaken zoals: biedt de mogelijkheid om de stijl van associatielijnen te wijzigen.

6.3.3 Kleur

De kleurensectie biedt verschillende opties om de tekst te wijzigen, lijn, vulling, raster en achtergrondkleuren evenals de lijnbreedte.

6.4 Klasse-instellingen



Opties voor instellingen voor klasse in Umbrello UML Modeller

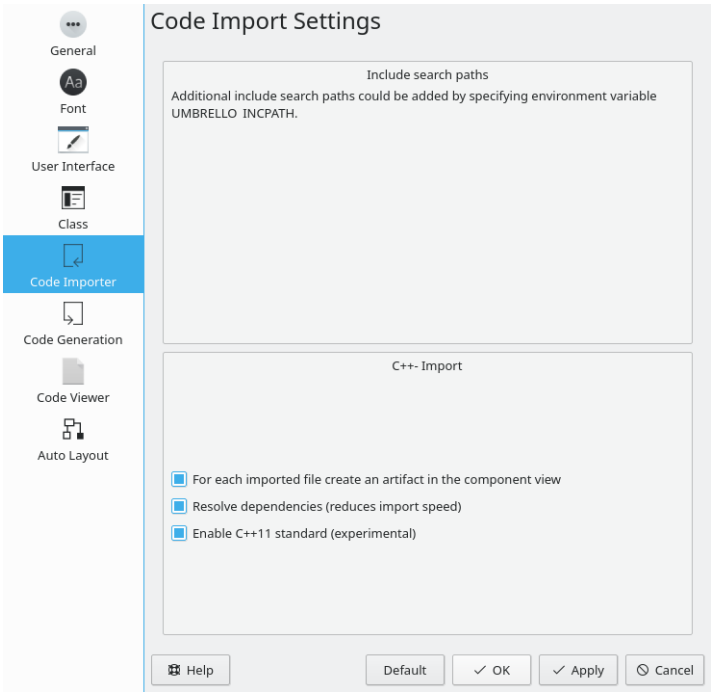
6.4.1 Tonen

De sectie Tonen heeft vele instellingen die bepalen welke klassekarakteristieken getoond worden in het klassediagram.

6.4.2 Startscope

Keuzes voor attribuut en standaardinstellingen voor werking, publiek, privé of beschermd.

6.5 Instellingen voor code importeren



Opties voor instellingen voor code importeren in Umbrello UML Modeller

6.5.1 Zoekpaden meenemen

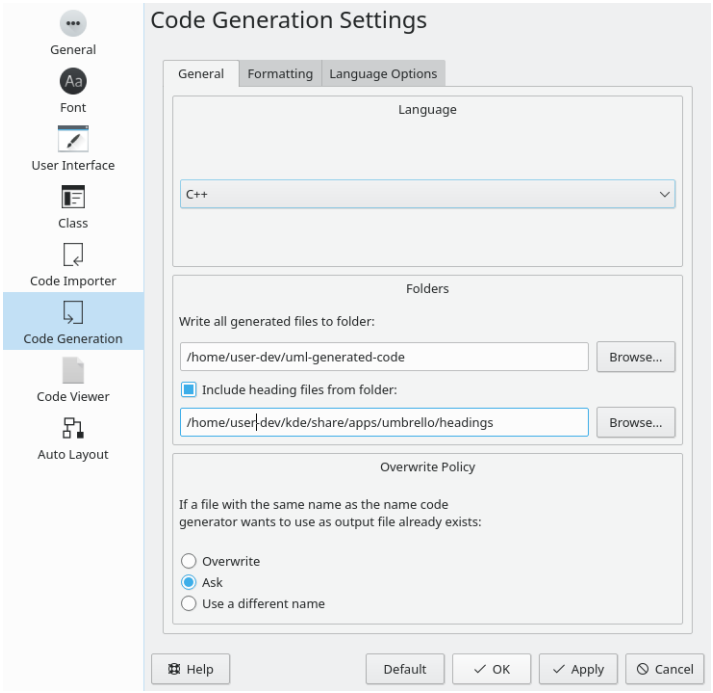
Er is een algemene aanbeveling gegeven om zoeken te verbeteren door UMBRELLO INCPATH als een omgevingsvariabele in te voegen.

6.5.2 Importeren van C++

- **Voor elk geïmporteerd bestand maak een artifact aan in de componentenweergave** Het aangemakkte artifact kan dan gesleept worden in de klassediagramweergave waar afhankelijkheden gemakkelijk gezien kunnen worden samen met de attributen en functies van elk bestand.
- **Afhankelijkheden oplossen (reduceert snelheid van importeren)** Verzekert dat alle afhankelijkheden van bestanden zijn opgelost die dan verschijnen in klasse-afhankelijkheden in het klassediagram.
- **C++11 standaard inschakelen (experimenteel)** Een experimentele functie om te voldoen aan C++11, schakel het uit indien niet nodig.

6.6 Instellingen voor de codegeneratie

6.6.1 Algemeen tabblad Instellingen voor de codegeneratie



Opties voor de algemene instellingen voor code generatie in Umbrello UML Modeller

Umbrello UML Modeller kan broncode genereren voor talrijke programmeertalen op basis van uw UML model om u te helpen van start te gaan met de implementatie van uw project. De gegenereerde code bestaat uit klasse declaraties, met hun methoden en attributen, dus u kunt de "blanke ruimte" invullen door de functionaliteit van uw klasse-operaties te verschaffen.

6.6.1.1 Taal

Kies de te gebruiken programmeertaal voor projecten. De geboden keuzes zijn ActionScript, Ada, C++, C#, D, IDL, Java, JavaScript, MYSQL, Pascal, Perl, PHP, PHP5, PastgreSQL, Python, Ruby, SQL, Tcl, Vala en XMLSchema

6.6.1.2 Mappen

Alle gegenereerde bestanden naar map schrijven: heeft een te bewerken veld voor het gewenste pad voor gegeneerde bestanden of optioneel een bladerknop om het pad te selecteren.

Header-bestanden uit deze map insluiten: indien geactiveerd, laat de gebruiker een pad specificeren in een bewerkbaar veld of kiest dit met een bladerknop.

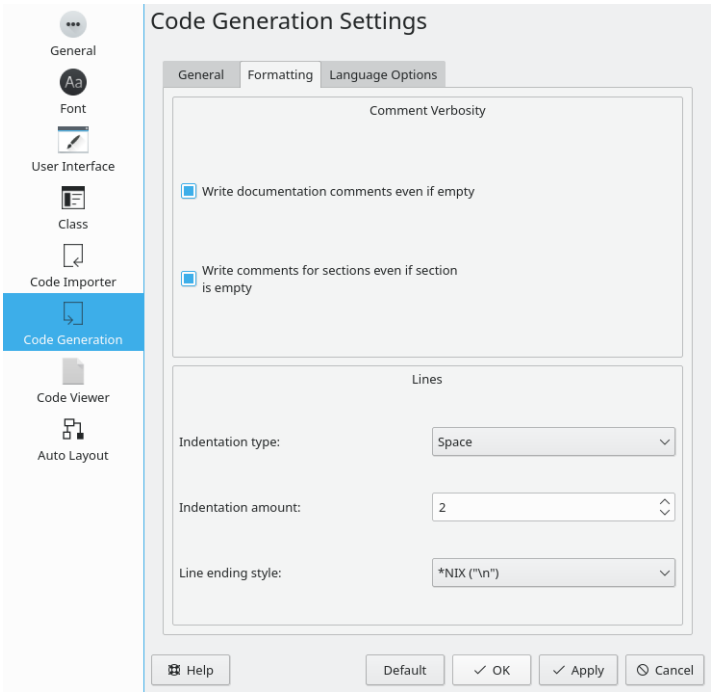
6.6.1.3 Overschrijvingsprotocol

Wanneer de code is gegenereerd in de gespecificeerde map, bepaalt deze instelling wat er gebeurt als een bestand met dezelfde naam wordt gevonden.

- **Overschrijven** van het bestand zonder een waarschuwing of optie.

- **Vragen** of het bestand wordt overschreven of het hernoemen.
- **Een andere naam gebruiken** wanneer een bestand al bestaat door het te herenoemen met een achtervoegsel.

6.6.2 Tabblad Instellingen voor het formaat van de codegeneratie



Opties voor de instellingen van het formaat voor code generatie in Umbrello UML Modeller

6.6.2.1 Uitvoerigheid commentaar

Documentatie-toelichting schrijven zelfs als deze leeg zijn Genereert toelichting voor klassen en functies zelfs als ze leeg zijn.

Toelichting schrijven voor secties zelfs als deze leeg zijn Schijft toelichtingen voor de privé, beschermde en publieke secties zelfs als ze leeg zijn.

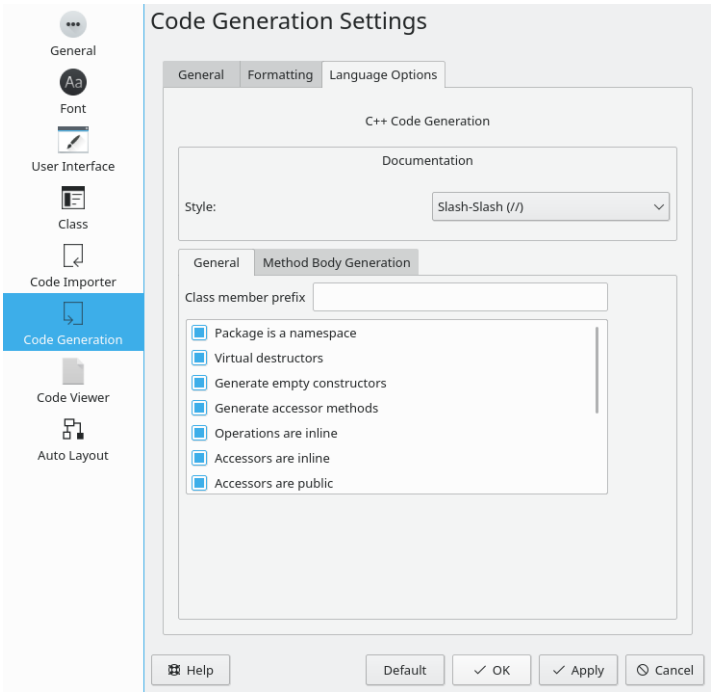
6.6.2.2 Lijnen

Inspringtype: biedt een keuze tussen niet inspringen, tab of spatie.

Inspringafstand: laat de gebruiker het aantal spaties specificeren voor de keuze inspringen met tab of spatie.

Type regeleinde: is een keuze tussen de stijl van regeleinde van *NIX, Windows of Mac.

6.6.3 Taalopties



Opties voor de algemene taalinstellingen voor code generatie in Umbrello UML Modeller

Deze pagina wijzigt voor elke programmeertaal geselecteerd onder het tabblad Algemeen opties. Op dit moment zijn alleen voor de C++ taal opties beschikbaar.

6.6.3.1 C++ codegeneratie

6.6.3.1.1 Documentatie

Stijl: geeft een keuze om ofwel "/** */" of "//" als de documentatiestijl te gebruiken

6.6.3.1.2 Algemeen

Onder het tabblad **Algemeen** van het tabblad **Taalopties**, staan verschillende opties voor code-generatie.

- **Voorvoegsel van klasselid**

  Een optie die een voorvoegsel biedt bepaald door de gebruiker, om toegevoegd te worden aan klasseleden wanneer code wordt gegenereerd.

- **Pakket als namespace**

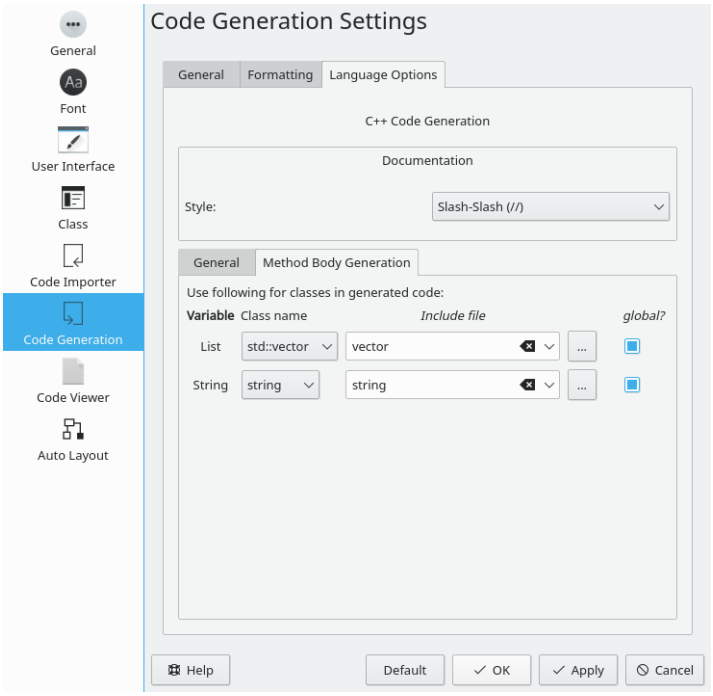
  Naamruimten bieden eem methode om naamconflicten in grote projecten te voorkomen. Symbolen gedeclareerd in een naamruimteblok worden placed in een genaamde scope geplaatst die voorkomt dat ze bij vergissing voor identiek genaamde symbolen in andere scopes worden gebruikt.

- **Virtuele destructors**

  Hoewel destructors niet geërfd worden, als een basis klasse zijn destructor virtueel declareert, zal de afgeleide destructor het altijd overschrijven. Dit maakt het mogelijk om dynamisch toegewezen objecten van veelvormig type via pointers pointers naar de basis te verwijderen.

- **Lege constructors aanmaken**
  Dit zal constructors genereren die lege accolades hebben.
- **Toegangsmethoden aanmaken**
  Zal methoden genereren om toegang te krijgen tot gegevenstypes.
- **Operaties zijn inline**
  Genereer de methoden als inline, maar compilers zijn vrij om niet de inline methode te kiezen.
- **Accessors zijn inline**
  Methoden die toegang bieden tot de gegevens van klasse zullen inline worden gegenereerd, maar compilers zijn vrij om niet de methode niet inline te gebruiken.
- **Accessors zijn openbaar**
  Methoden die gegenereerd zijn als publiek zullen beschikbaar zijn aan elk exemplaar van de klasse.
- **Ophalers aanmaken met voorvoegsel 'get'**
  Dit zal het voorvoegsel "get" op de methoden zetten die de klassegegevens ophalen/teruggeven.
- **Voorvoegsel '[a-zA-Z]_' verwijderen uit namen van toegangsmethode**
  Als een voorvoegsel was ingevoerd in **Voorvoegsel van klasselid**, zal dit verwijderd worden.
- **Toegangsmethoden beginnen met hoofdletters**
  Dit maakt de eerste letter van de methodenaam een hoofdletter.
- **'\' gebruiken als documentatie-tag in plaats van '@'**
  Een keuze van een tag om te gebruiken bij documenteren van parameters van een methode.

6.6.3.1.3 Genereren van de body van de Methode



Opties voor de algemene taalmethode-inhoud-instellingen voor code generatie in Umbrello UML Modeller

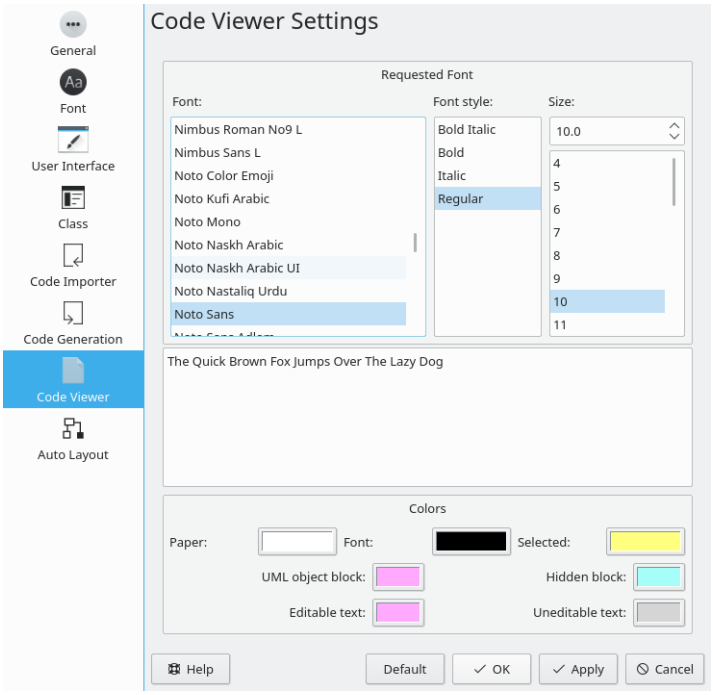
Lijst

Heeft opties van QPtrList, vector en std::vector voor de lijst met type. Een te bewerken of te selecteren veld volgt om het in te voegen bestand te specificeren samen met een bladerknop om het in te voegen bestand te zoeken en te selecteren. Er is ook een optie om de lijst globaal te maken.

Tekenreeks

Opties van tekenreeks of QString voor het type tekenreeks. Een te bewerken of te selecteren veld volgt om het in te voegen bestand te specificeren samen met een bladerknop om het in te voegen bestand te zoeken en te selecteren. Er is ook een optie om de tekenreeks globaal te maken.

6.7 Instellingen voor de codeviewer

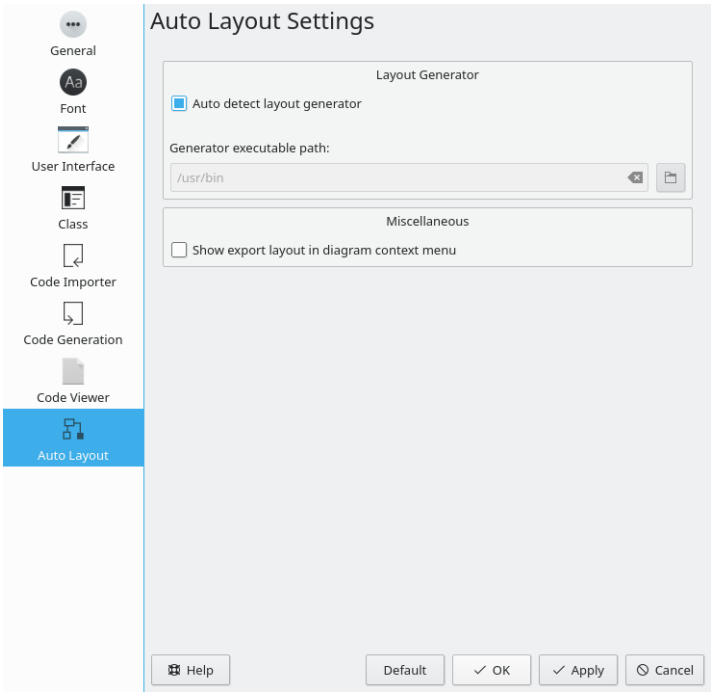


Opties voor instellingen voor code bekijken in Umbrello UML Modeller

Biedt aanpassen van de codeviewer. De sectie **Gevraagd lettertype** biedt de selectio van het lettertype, de stijl ervan en de grootte. Een representatie van uw keuzes wordt onder de keuzes getoond.

In de sectie **Kleuren**, kunnen wijzigingen gemaakt worden aan papier, lettertype, geselecteerd, UML-objectblok, verborgen blok, bewerkbare tekst en niet-bewerkbare tekst. Wijzigingen in de kleuren kunnen gemaakt worden door te klikken op het kleurvakje door het respectievelijke label.

6.8 Instellingen voor automatische opmaak



Opties voor instellingen voor automatische indeling in Umbrello UML Modeller

Generator voor opmaak automatisch detecteren

De functie automatische indeling hangt af van generators voor indeling geleverd door het pakket GraphViz, dat normaal naast Umbrello wordt geïnstalleerd door een pakketbeheerder. Umbrello is gebouwd met ondersteuning voor het detecteren van de locatie van de installatie van deze generators voor indeling. Voor gevallen waar deze afhankelijkheid niet beschikbaar is of niet past, kan een ander installatiepad worden geleverd.

Export-indeling in diagram-contextmenu

Een .dot-bestand exporteren wordt gedaan met de export-indeling. Met deze optie geactiveerd, wordt de export-indeling toegevoegd aan de beschikbare diagram-indelingen en schakelt een snel voorbeeld in van .dot exporteren.

Hoofdstuk 7

Auteurs en geschiedenis

Dit project was gestart door Paul Hensgen als een van zijn universitaire projecten. De oorspronkelijk naam van de toepassing was UML Modeller. Paul deed alle ontwikkeling zelf tot het eind van 2001 toen het programma versie 1.0 bereikte.

Versie 1.0 bood al een hoop functionaliteit, maar nadat het project was beoordeeld op Pauls Universiteit, konden andere ontwikkelaars zich erbij voegen en zij begonnen met het maken van waardevolle bijdragen aan de UML Modeller, zoals de overschakelen van een binair bestandsformaat naar een XML-bestand, ondersteuning van meer soorten UML-diagrammen, codegeneratie en code-import, om er een paar te noemen.

Paul moest zich terugtrekken uit het ontwikkelteam in de zomer van 2002 maar als Vrije en Open Source Software, bleef het programma zich verbeteren en ontwikkelen en wordt het onderhouden door een groep ontwikkelaars uit verschillende delen van de wereld. In September 2002 veranderde het project zijn naam van UML Modeller, in Umbrello UML Modeller. Er zijn meerdere beweegredenen voor deze naamsverandering, de belangrijkste is wel dat alleen 'uml' — waaronder het algemeen bekend was — een te algemene naam was en problemen met sommige distributies met zich meebracht. Een andere, niet minder belangrijke reden, is dat de ontwikkelaar van mening zijn dat Umbrello een veel coolere naam is.

De ontwikkeling van Umbrello UML Modeller evengoed als de discussies met betrekking tot waar het programma in voor moet gaan in toekomstige versies, is open en speelt zich af op het Internet. Als u een bijdrage wilt leveren aan het project, aarzel dan niet om contact op te nemen met de ontwikkelaars. Er zijn vele manieren waarop u kunt helpen bij Umbrello UML Modeller:

- Melden van bugs of suggesties tot verbeteringen
- Bugs oplossen of features toevoegen
- Goede documentatie schrijven of het vertalen in een andere talen
- En vanzelfsprekend ... programmeren met ons!

Zoals u ziet, zijn er veel manieren waarop u uw steentje kunt bijdragen. Zij zijn alle heel belangrijk en ieders deelname is welkom.

De ontwikkelaars van Umbrello UML Modeller zijn bereikbaar op umbrello-devel@kde.org.

Hoofdstuk 8

Copyright