Projectbeheer en geavanceerde processen in softwareontwikkeling.

Ontwikkeling volgens MDA

Professoren: Jan Wirix

Kristof Coninx

16 mei 2012

Inhoudsopgave

Lijst van figuren

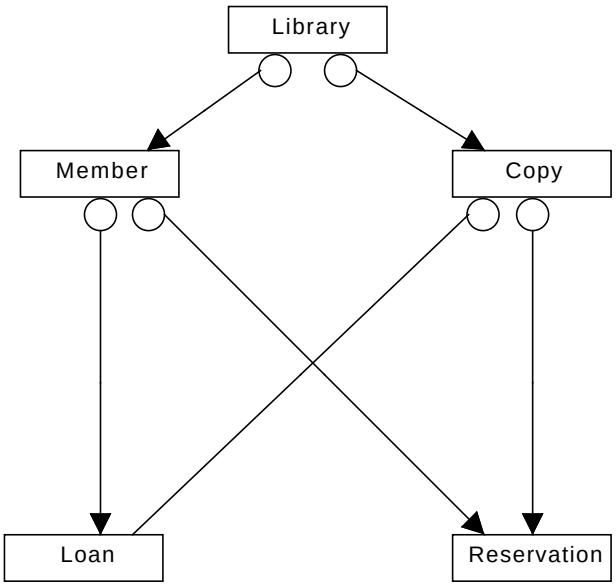
1 Merode Deliverables

In deze sectie worden de merode deliverables besproken waarvan gewerkt wordt om tot een oplossing te komen binnen het versata framework. Deze deliverables zijn voornamelijk geïnspireerd door het voorbeeld in hoofdstuk 7 van de merode documentatie. Hier en daar zijn er echter wel enkele aanpassingen en/of uitbreidingen doorgevoerd.

Zo is er ook een library object toegevoegd om de member en copy klassen in een context te plaatsen. Verder werd ook de object event tabel uitgebreid met events voor het aanmaken en vernietigen van library objecten. Tenslotte werd ook de life cycle van het copy object wat uitgebreid om een moeilijker scenario te illustreren bij de omzetting van deze modellen binnen het versata framework.

1.1 Existence Dependency Graph

Dit EDG-diagram is hetzelfde als wat er in de opgave getoond werd, met als toevoeging de library-component. Deze component zal verder ook invloed hebben op de OET zoals besproken in de 1.2.



Figuur 1: EDG for the library

1.2 Object Event Table

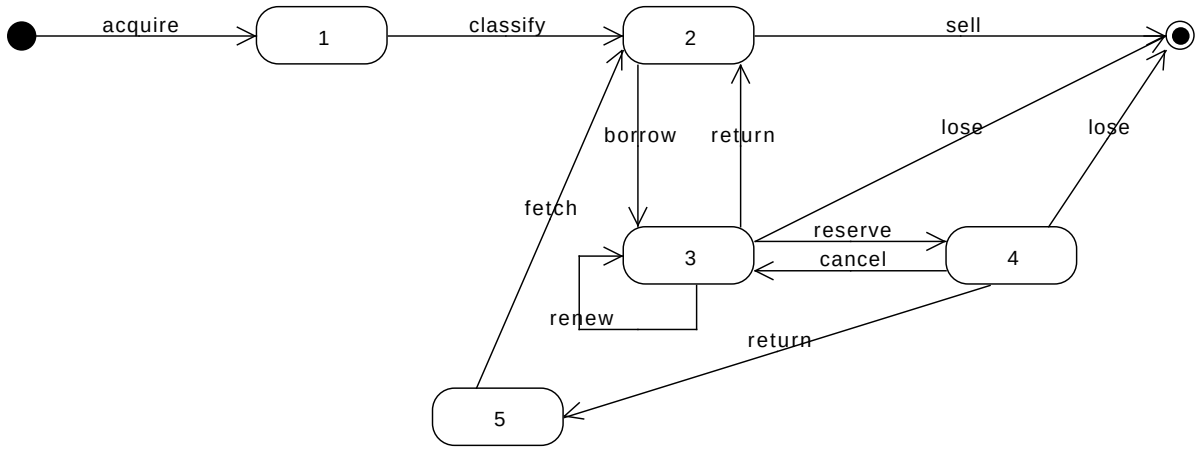
In deze sectie wordt de Object event table besproken. Per event wordt er in deze tabel weergegeven welke types van acties er worden uitgevoerd op verschillende objecten. (Deze acties zijn met de letters C, M en E letters afgekort weergegeven voor Create, modify en End.)

Tabel 1

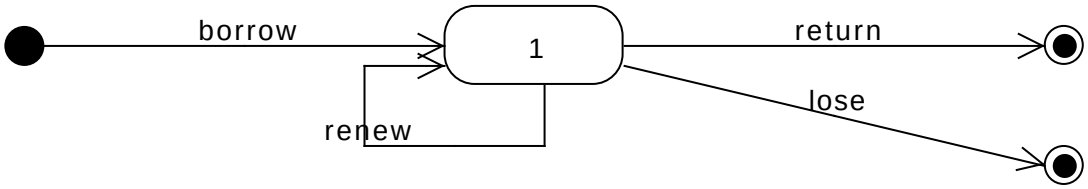
	Library	Member	Copy	Loan	Reservation
create	C				
enter	M	C			
leave	M	E			
acquire	M		C		
classify			M		
borrow		M	M	C	
renew		M	M	M	
return		M	M	E	
sell	M		E		
reserve		M	M		C
cancel		M	M		E
fetch		M	M	C	E
lose		M	E	E	
destroy	E				

1.3 Life cycles

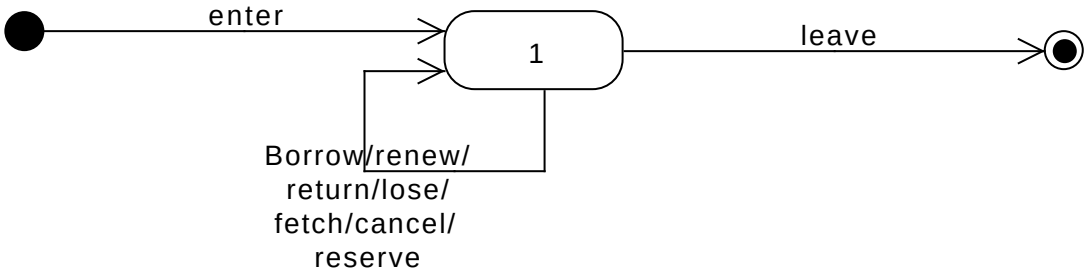
In deze sectie worden de lifecycles besproken voor de objecten in de voorgaande Merode modellen. Deze life cycles worden voorgesteld in toestandsdiagramma en bepalen de toestanden waarin een bepaald object kan geraken door middel van het afvuren van bepaalde events op deze objecten.



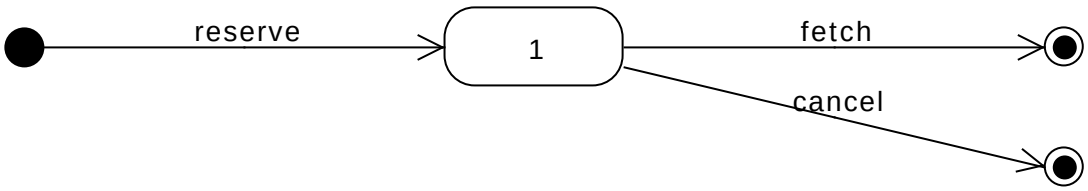
Figuur 2: Lifecycle for Copy



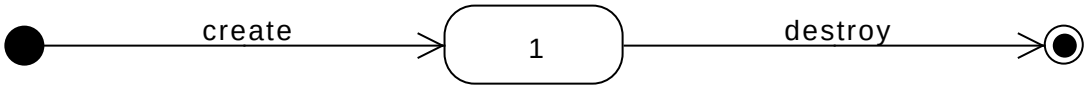
Figuur 3: Lifecycle for Loan



Figuur 4: Lifecycle for Member



Figuur 5: Lifecycle for Reservation



Figuur 6: Lifecycle for Reservation

1.4 Precondities

In deze sectie worden enkele precondities besproken waaraan bepaalde events moeten voldoen alvorens te vuren.

Allereerst worden er voor alle events constraints opgesteld gebaseerd op de lifecycles waarin hun doelobjecten voorkomen. Voor elke event wordt er een constraint bepaald voor het nagaan of het doelobject zich in de juiste toestand bevindt.

Een volgende conditie die uitgewerkt wordt als preconditie, is het feit dat een lid maar maximum 5 boeken mag ontlenen. Alsook de conditie dat een boek enkel kan uitgeleend worden wanneer het nog niet is ontleend wordt, wordt uitgedrukt als een preconditie.

In deze context wordt ook de regel toegevoegd dat een boek niet uitgeleend mag worden wanneer er nog een openstaande reservatie voor bestaat (die nog niet gefetched is).

Betreffende de bibliotheekobjecten zelf werd er bepaald dat alle boeken verkocht moeten worden, en alle leden het pand verlaten moeten hebben alvorens een bibliotheek te vernietigen.

Een boek kan ook niet verkocht worden wanneer er nog openstaande reservaties zijn voor dit boek, of wanneer dit boek nog uitgeleend is aan een lid.

Tenslotte kan een lid ook enkel de bibliotheek betreden wanneer deze bibliotheek effectief boeken heeft verworven.

Hoe deze precondities uiteindelijk worden afgedwongen wordt beschreven in sectie 2.4

1.5 Acties

In deze sectie worden bepaalde acties besproken die gepaard gaan met het afvuren van bepaalde events.

Allereerst worden hier alle acties opgesteld gebaseerd op de lifecycles waarin hun doelobjecten voorkomen. Voor elke event die een object in een andere toestand brengt, moet er als actie de regel toegevoegd worden die deze toestandsverandering uitvoert.

Verder moet er ook nog afgedwongen worden dat bij de events die een creatie-eigenschap hebben, de juiste business objecten moeten aangemaakt worden. Dit wordt verder besproken in sectie 2.5

2 Transformatie van MERODE naar VERSATA

In deze sectie wordt systematisch de mechanische omzetting van de Merode modellen uit sectie 1 in een oplossing binnen het Versata framework besproken.

2.1 EDG

De omzetting van het EDG diagram naar een oplossing binnen versata wordt besproken in sectie 3

2.2 OET

Bij het omzetten van de merode modellen naar een oplossing binnen het versata framework, vertrekken we van de object-event-table. Deze tabel zal gebruikt worden om de versata objecten te construeren. Voor elke kolom van de tabel wordt er een versata business object aangemaakt. Deze objecten zullen als het ware de datacontainers zijn voor de domein elementen die ze voorstellen. Deze objecten worden ook voorzien van attributen die relevant kunnen zijn voor deze context. Deze attributen worden niet expliciet in de merode documentatie vermeld, maar kunnen zelf nog toegevoegd worden waar nodig.

Verder worden de objecten die aangemaakt worden in versata studio ook nog voorzien van:

- Primary key: Unieke primaire sleutel voor elk object.
- Foreign key: Verwijzing naar de primaire sleutels van elk object waar dit object een relatie mee heeft.
- Afleidende velden: Attributen die het aantal relaties tellen of een som maken. Deze objecten dienen te worden aangemaakt wanneer nodig in een voorheen besproken constraint.
- ObjectState: Een teller die bijhoudt in welke toestand een object zich bevindt.

Voor elke rij in de OET (events) worden event object aangemaakt in versata studio. Deze objecten verschillen volgens versata niet per definitie van business objecten, maar worden in onze context wel op een andere manier gebruikt. Voor elke mogelijke event worden er dus event objecten aangemaakt. Bij deze objecteventen kunnen immers constraints en acties worden gespecifieerd die de beperkingen en handelingen die gepaard gaan bij een event, voorstellen.

Een bepaald subtype van deze event objecten, namelijk de create events zorgen echter ook voor de aanmaak van hun parent-object (wanneer natuurlijk aan alle precondities is voldaan). Verder hebben deze event objecten de volgende eigenschappen(/velden).

- Primary key: Unieke primaire sleutel voor elk event object.
- Foreign key: Voor elk object waarop deze event een handeling uitvoert.
- Attributes: Voor een create event object, een attribuut veld voor elk te isntantiëren veld in het aan te maken parent object.

Het is mogelijk om nog meer informatie bij te houden in deze event objecten zoals tijdstip van uitvoeren of de initiator van het event. Dit is echter niet strikt noodzakelijk en kan mogelijk gezien worden als administratieve keuze die gemaakt moet worden.

2.3 Life cycles

2.4 Precondities

2.5 Acties

3 Transformatie van EDG naar een UI

3.1

3.2

4 Toepasbaarheid van MDA

4.1 Lessen

4.2 Overige noodzakelijkheden

4.3 Conclusies