

CQML-Schema - Entwurfsdokumentation

Ute Wappler

Dresden, den 9. April 2003

1 Einleitung

CQML ist für die Nutzung in der Analyse- und Entwurfsphase konzipiert. Für die Verarbeitung der QoS-Spezifikation durch einen Container ist es notwendig, diese in eine Laufzeitrepräsentation zu übersetzen. Dazu wurde die CQML-Grammatik auf ein XML-Schema abgebildet. Dabei wurden einige Vereinfachungen von Sprachkonstrukten vorgenommen, die im folgenden beschrieben werden.

Allgemein ist zu sagen, daß immer wenn von IDs gesprochen wird, diese im Schema als keys definiert sind, und immer wenn Verweise auf diese erwähnt sind, wurden dafür entsprechende keyrefs definiert. Diese keys und keyrefs sind im cqml-Element deklariert.

2 Qualitätscharakteristiken

Die Qualitätscharakteristiken wurden ohne große Veränderungen übernommen, wobei `< simpleQualityCharacteristic >`, welches innerhalb von `< qualityCharacteristic >` verwendet wird, der Regel entspricht:

```
< characteristic_declaration > ::= quality_characteristic < characteristic_name >
                                   [ ('< OCL :: formalParameterList >')]
                                   '{' [< individual_body >] '}'
```

`< specializedQualityCharacteristic >`, welches alternativ zu `< simpleQualityCharacteristic >` in `< qualityCharacteristic >` verwendet werden kann, entspricht folgender Regel:

```
< characteristic_declaration > ::= quality_characteristic < characteristic_name >
                                   < parent_characteristic >
                                   '{' [< specialization_body >] '}'
```

Alle weiteren verwendeten Tagnamen und Strukturen entsprechen denen der CQML-Grammatik. Jede Qualitätscharakteristik besitzt neben ihrem Namen noch eine ID, welche eine beliebige Zeichenkette sein kann, die innerhalb eines CQML-XML-Dokumentes für alle Qualitätscharakteristiken eindeutig sein muß, da sie als key definiert wurde. Diese ID wird zum Verweisen auf die jeweilige Charakteristik verwendet, z.B. bei Erweiterung, in Qualitätsstatements und in funktionalen Ableitungen, wobei die Qualitätscharakteristiken, auf die verwiesen wird, zwingend deklariert sein müssen.

3 Ressourcen

Ressourcen gibt es innerhalb der CQML-Grammatik [1] von Aagedal nicht. Innerhalb des Projekts COM-QUAD wurde diese Grammatik unter anderem um das Konstrukt der Ressourcen erweitert. Sie dienen dem Zusammenfassen bestimmter Qualitätscharakteristiken, die eine bestimmte Ressource charakterisieren. Das `< resource >`-Tag hat ein Attribut Name und eines ID, das zur eindeutigen Identifizierung einer Ressource in einem Dokument dient und für Verweise auf diese Ressource benutzt wird. Ansonsten enthält es eine Auflistung von Qualitätscharakteristiken `< qualityCharacteristic >`. Das `< resource >`-Tag wird innerhalb eines `< qualityCategory >`-Tags verwendet.

4 Qualitätsstatements

Auch die Qualitätsstatements besitzen genau wie die Charakteristiken und Ressourcen neben ihrem Namen noch eine ID, die der eindeutigen Identifizierung und zum Verweisen dient (z.B. in Profilen).

In der CQML-Grammatik findet sich folgende Regel: `< constraint > ::= < simple_constraint > | < qualified_constrained >`. Diese wurde ersetzt durch `< constraint > ::= < qualified_constrained >`, d.h. jeder Constraint, egal ob `< simpleConstraint >` oder `< qualifiedConstraint >`, besitzt ein Attribut `qualification`, welches die Werte `best-effort` oder `guaranteed` annehmen kann.

Außerdem wurde statt `< one_aspect >` die Bezeichnung `statisticalDerivation` gewählt, da dies aussagekräftiger ist. Die weiteren Tagnamen und Strukturen entsprechen der CQML-Grammatik.

Desweiteren wurde die `< value >`-Regel ergänzt:

```
< value >::=< element > | < range_value > | < characteristic_name >
['(' < OCL :: actualParameterList > ')' | '.' < one_aspect >']] < set_value >
```

`< set_value >` dient dem einschränken von Qualitätscharakteristiken, die eine Menge definieren und es gilt folgende Regel: `< set_value >::=< set_item > *`. Näheres zur XML-Syntax findet sich in der HTML-Dokumentation.

Es ist darauf zu achten, daß Qualitätcharakteristiken auf die verwiesen wird, auch im Dokument deklariert sind.

5 Profile

Im Gegensatz zur CQML-Grammatik gibt es im CQML-XML-Schema keine einzelnen Profile, sondern nur Profile, welche Profile, Transitionen und Präzedenzlisten enthalten. Sie entsprechen dem Tag `< compoundProfile >`. Außerdem hat auch jedes `compoundProfile` neben seinem Namen wieder eine eindeutige ID, welche zum Beispiel bei der Erweiterung von Profilen angegeben wird. Für jede Komponente darf es innerhalb einer Qualitätskategorie nur ein Profil geben. Eine solche Einschränkung ist in der Grammatik ebenfalls nicht vorgesehen.

Die Definition des Inhalts eines Profils wurde ebenfalls leicht geändert. Im Prinzip wurde aus der Regel: `< profile_body >::=< single_profile_body > | < one_profile > + < transition > + [< precedence >]` diese Regel: `< profile_body >::=< one_profile > + < transition > * < precedence >`, d.h. daß eine Präzedenzliste immer vorhanden sein muß. Sie muß auch die in [1](Seite 70) erwähnten implizit gegebenen Reihenfolgen enthalten. Transitionen hingegen, können auch fehlen und wie schon oben erwähnt gibt es keine einzelnen Profile außerhalb von zusammengesetzten Profilen (`compoundProfiles`). Desweiteren müssen die Namen eines Teilprofils innerhalb des zusammengesetzten Profils eindeutig sein und in den Transitionen und der Präzedenzliste können nur in diesem Profil deklarierte Teilprofile verwendet werden.

Die einzelnen Teilprofile eines zusammengesetzten Profils sind, bis auf die im folgenden beschriebenen Änderungen, entsprechend der CQML-Grammatik aufgebaut.

Die in der Grammatik mit `uses` bzw. `provides` gekennzeichneten Daten entsprechen im CQML-XML-Schema den Tags `< expectations >` bzw. `< offers >`. Desweiteren wurde ein Element `< resources >`, welches in der Grammatik nicht existiert, hinzugefügt. Es repräsentiert die benötigten Ressourcen, welche in disjunktiver Normalform angegeben werden. Die in `< expectations >`, `< offers >` und `< resources >` verwendeten Qualitätsstatements und Ressourcen müssen innerhalb des Dokuments deklariert sein.

Sowohl erwartete als auch angebotene QoS-Eigenschaften werden im XML in konjunktiver Normalform wiedergegeben. In der CQML-Grammatik sind beliebige logische Formeln gestattet. Außerdem wurde die in der `< expectation >`-Regel vorgesehene `worth`-Methode weggelassen, da ihre Benutzung nicht vorgesehen ist.

In den einzelnen Transitionen wird ein 'any' durch das Fehlen des entsprechenden Attributs ausgedrückt. Dies war durch die definierten keys und keyrefs nicht anders möglich.

6 Qualitätskategorien

Qualitätskategorien wurden entsprechend der Grammatik übernommen, wobei die Referenzen auf Qualitätscharakteristiken, Qualitätsstatements Profile und weitere Qualitätskategorien folgenden Tags entsprechen: *< qualityCharacteristicReference >*, *< qualityReference >*, *< compoundProfileReference >* und *< qualityCategoryReference >*. Zusätzlich wurden Ressourcen *< resource >* und Referenzen *< resourceReference >* auf diese hinzugefügt. Es wurde darauf verzichtet leere Qualitätskategorien nicht zu gestatten, da das Schema dadurch sehr unübersichtlich geworden wäre und diese beim Erzeugen des XML leicht zu vermeiden sind.

Die Namen der Qualitätskategorien müssen eindeutig sein.

Literatur

- [1] Jan Øyvind Aagedal. Quality of Service Support in Development of Distributed Systems. März 2001