TO naamswijziging context

|  |  |  |
| --- | --- | --- |
| Joop Ringelberg | 19-12-17 | Versie: 1 |

# Introductie

Een modelleur die de editor gebruikt, geeft een context zijn naam door hem in te tikken. Dat moet een gekwalificeerde naam zijn, maar dankzij default namespacing kan hij aan die eis voldoen door de lokale naam te prefixen met : of #.

Maar de modelleur kan de naam van een context ook wijzigen. Wat we niet willen, is dat daardoor de onderliggende identifier verandert waarmee de representatie van de context in het systeem wordt bewaard en teruggevonden.

Om die reden hebben we een *displayName* property van de contextrepresentatie geïntroduceerd. Deze tekst geeft het ontwerp van de functie waarmee zo’n naamswijziging in de editor wordt uitgevoerd.

# Hoofdlijn

Conceptueel is de belangrijkste uitdaging dat we de tekstuele wijzigingen die de modelleur aanbrengt, in verband moeten brengen met de juiste interne context representatie. Dat doen we als volgt.

We laten het systeem letten op tekstwijzigingen die de contextnaam in een context declaratie raken. We bepalen dan de contextnaam van vóór de wijziging en leggen die vast. Vervolgens laten we het systeem letten op tekstwijzigingen waaruit we kunnen afleiden dat de modelleur klaar is met de naamswijziging (dat zal een wijziging zijn die eindigt op een andere regel). We lezen dan de nieuwe naam en leggen die vast als nieuwe displayNaam.

# Technische uitdagingen

De Ace editor signaleert voor onze doeleinden maar één soort gebeurtenis die we kunnen gebruiken, en dat is de functie onChange in de module Ace.Document[[1]](#footnote-1). Wat we goed hadden kunnen gebruiken is een event bij cursorverplaatsing, maar die is er niet. Zoals de naam al aangeeft, wordt Perspect pas ingeschakeld door Ace nadat een wijziging heeft plaatsgevonden. Om de oorspronkelijke naam te kunnen bepalen, zouden we de oorspronkelijke tekst volledig moeten bewaren – en dat bij elke toetsaanslag. Gelukkig levert de DocumentEvent data voldoende informatie om uit de tekst ná wijziging de tekst vóór wijziging te kunnen herstellen.

## Het DocumentEvent

De structuur van het DocumentEvent type is als volgt:

newtype DocumentEvent = DocumentEvent

{ action :: DocumentEventType

, start :: Position

, end :: Position

, lines :: Array String

}

data DocumentEventType = Insert | Remove

newtype Position = Position

{ row :: Int

, column :: Int

}

De gegevens worden als volgt geïnterpreteerd.

1. positie is zero-based, voor zowel regels als kolommen.
2. Tekst wordt rechts van de cursor ingevoegd. Na invoegen staat de cursor rechts van de ingevoegde tekst. Dus: als op (0,0) één karakter wordt ingevoegd, is de positie van de cursor daarna (0,1).
3. Bij een selectie geldt:
   1. de positie links van de eerste letter is de beginpositie;
   2. de positie rechts van de laatste letter is de eindpositie.
4. Bij remove staat de cursor
   1. oorspronkelijk op de eindpositie
   2. na bewerking op de beginpositie.  
      Alle karakters in de oorspronkelijke test tussen beide posities zijn dan verwijderd.
5. Bij insert staat de cursor
   1. oorspronkelijk op de beginpositie
   2. na bewerking op de eindpositie.  
      Alle karakters in het array lines staan tussen die twee posities in de nieuwe tekst.

# Algoritmes

1. Herstel de oorspronkelijke regels waar de mutatie in plaatsvond.
2. Bepaal of de eerste regel een contextdeclaratie is.

ZOJA:

* 1. bij een remove bevindt de cursor zich nu in de contextdeclaratie. De modelleur is dus nog niet klaar met de naamswijziging. Bepaal de oorspronkelijke naam en het regelnummer en sla op.
  2. bij een insert van één regel (begin- en eindpositie bevinden zich op dezelfde regel) is de gebruiker evenmin klaar. Afhandeling als bij a.
  3. bij een insert van meer dan één regel is de gebruiker klaar met de naamswijziging (als die al heeft plaatsgevonden). HERNOEM.

ZONEE:

* 1. als een naam en regelnummer bewaard zijn; als de positie van de cursor na de operatie niet gelijk is aan het bewaarde regelnummer, is de gebruiker nu klaar met de naamswijziging. HERNOEM.

HERNOEM

1. Haal de bewaarde, oorspronkelijke naam en zijn regelnummer op.
2. Bepaal de id van de context op basis van de oorspronkelijke naam. Dit vergt het parseren van omhullende contexten!
3. Bepaal de nieuwe naam (op de regel met het betreffende nummer).
4. Verander in de aldus geïdentificeerde context de displaynaam in de nieuwe naam.

HERSTEL TEKST NA REMOVE

1. Pas de functie getLine van Ace.Document toe op de row van de beginpositie. Noem het resultaat l.
2. Splits l op de column van de beginpositie in a en b.
3. Prepend lines[0] met a.
4. Postpend lines[n] met b.
5. Lines bevat nu de oorspronkelijke tekst.

HERSTEL TEKST NA INSERT

1. Pas de functie getLine van Ace.Document toe op de row van de beginpositie. Noem het resultaat lb.
2. Verwijder uit lb alle karakters na de column van de beginpositie. Noem het resultaat a.
3. Pas de functie getLine van Ace.Document toe op de row van de eindpositie. Noem het resultaat le.
4. Verwijder uit le alle karakters vóór de column van de eindpositie. Noem het resultaat b.
5. De oorspronkelijke regel is nu a + b.

# De gekwalificeerde naam bepalen

Als de modelleur default namespacing heeft gebruikt voor de contextnaam, moeten we die expanderen tot de gekwalificeerde naam. Dat doen we door op aan de contextdeclaratie voorafgaande regels te zoeken naar contextdeclaraties, totdat we er één vinden die volledig gekwalificeerd is.

Alle gedeclareerde namen die we van begin tot en met eind vinden, voegen we samen tot de gezochte, gekwalificeerde naam.

De gezochte gekwalificeerde naam construeren we door:

* te beginnen met de boven de betreffende contextdeclaratie gevonden gekwalificeerde naam (in geëxpandeerde vorm!).
* die te laten volgen door elke eronder gevonden lokale naam (waarbij we de segmenten telkens scheiden met #).

IS EEN NAAM GEKWALIFICEERD?

1. Als de naam begint met de default, lege prefix (d.w.z. de naam begint met :), dan is de naam gekwalificeerd. We kunnen hem expanderen door de domeinnaam van de default namespace (gedeclareerd in de Tekst) vóór de lokale naam te prependen.
2. Als de naam begint met een prefix (d.w.z. de naam begint met prefix:), dan is de naam gekwalificeerd. We kunnen hem expanderen door de domeinnaam horend bij de prefix (gedeclareerd in de Tekst) vóór de lokale naam te prependen.

In alle andere gevallen is de naam niet gekwalificeerd.

Merk op dat een naam hooguit uit drie segmenten kan bestaan.

1. Ik houd de naamgeving van het purescript-ace package aan. [↑](#footnote-ref-1)