Betrouwbaar betalen in Perspectives

Joop Ringelberg 18-07-18 Versie: 1

Introductie

Hoe modelleren in Perspectives we een financiële transactie waarbij A een bedrag aan B overmaakt? Geld kunnen we eenvoudig modelleren als een rekening waarop bijgeschreven en waarvan afgeschreven wordt. De administratie is simpel genoeg, maar hoe voorkom je fraude? Een belangrijk probleem voor virtuele munten is ‘double spending’. Een virtuele munt wordt gerepresenteerd met informatie en die kan eenvoudig gekopieerd worden.

Eerste modellering

1. Een gebruiker heeft geen rekening, alleen een serie opeenvolgende transacties.
2. Een transactie is een Context waar A en B in zijn betrokken.
3. Een transactie heeft als 'resultaat' een cumulatieve rekeningstand voor A en B, d.w.z. de standen ná de transactie.
4. Als A geld overmaakt aan B dan betreft de transactie een credit-actie voor de cumulatieve rekeningstand van B en een debet actie voor die van A. A vult de rol `debitor` en B de rol `creditor`.
5. Elke transactie bouwt voort op twee voorafgaande transacties. De eindstanden van de voorafgaande transacties zijn de beginstanden van de transactie. De debet- en credit acties worden op deze beginstanden toegepast om de eindstanden te verkrijgen. Een transactie heeft daarom een (functionele) rol `creditorPredecessorTransaction`, die gevuld wordt met de vorige Transactie van de `creditor`, en zo ook een rol `debitorPredecessorTransaction`.

Double spending in Perspectives

Het 'double spending' probleem vertaalt zich in deze modellering naar het tweemaal gebruiken van dezelfde transactie als basis voor een volgende transactie. Oftewel, een transactie mag maar één keer voorkomen als vuller van een de rol `creditorPredecessorTransaction`. Maar om dat vast te stellen, zou een partij toegang moeten hebben tot **alle** Transacties, wereldwijd. Dat is niet realistisch en daarom breiden we de modellering van de Transactie uit:

6. Een Transactie heeft een rol `creditorSuccessorTransaction` en een rol `debitorSuccessorTransaction`. De eerste is het vervolg voor de `creditor`, de tweede het vervolg voor de `debitor`. Merk op dat transacties dus 'dubbel gelinkt' zijn: `creditorSuccessorTransaction` is de omgekeerde van één van de `PredecessorTransaction` rollen van de volgende transactie van de `creditor`

(welke precies hangt af van de rol die de creditor in die volgende Transactie speelt!).

Hoewel we nu kunnen representeren dat een Transactie een vervolg heeft, hebben we nog niet uitgesloten dat A zo'n vervolg tegenover B zou kunnen verzwijgen (door een `successorTransaction` te wissen of niet door te geven aan B).

Double spending in Perspectives voorkomen

Om te waarborgen dat B betrouwbaar geïnformeerd wordt over de werkelijke toestand van de voorafgaande transactie van A, roepen we de hulp van getuigen in. Als A transactie Ta als vorige transactie inbrengt bij zijn transactie T met B, moeten de getuigen in effect kunnen nagaan dat Ta nog niet door A op die manier is gebruikt. Dit betekent dat de getuigen geïnformeerd moeten worden over de transacties.

Zowel A als B brengen een getuige in. Zo kan elk zich ervan vergewissen dat er minstens één betrouwbare getuige bij een Transactie meekijkt. We breiden daarom de modellering van een Transactie als volgt uit:

7. Een Transactie heeft een rol `debitorWitness` en `creditorWitness`.

Dit is nog niet voldoende. Om double spending te voorkomen, moet een getuige kunnen vaststellen dat de *vorige Transacties* niet al een keer gebruikt zijn, dus dat de juiste `SuccessorTransaction` leeg is.

Maar in Perspectives is een partij slechts geïnformeerd over een Context als hij er een rol bij speelt en dan slechts voor zover het bij die rol horende Perspectief en de Views op de rollen hem toestaan. We moeten de getuigen van een Transactie dus óók betrekken bij de vorige Transacties.

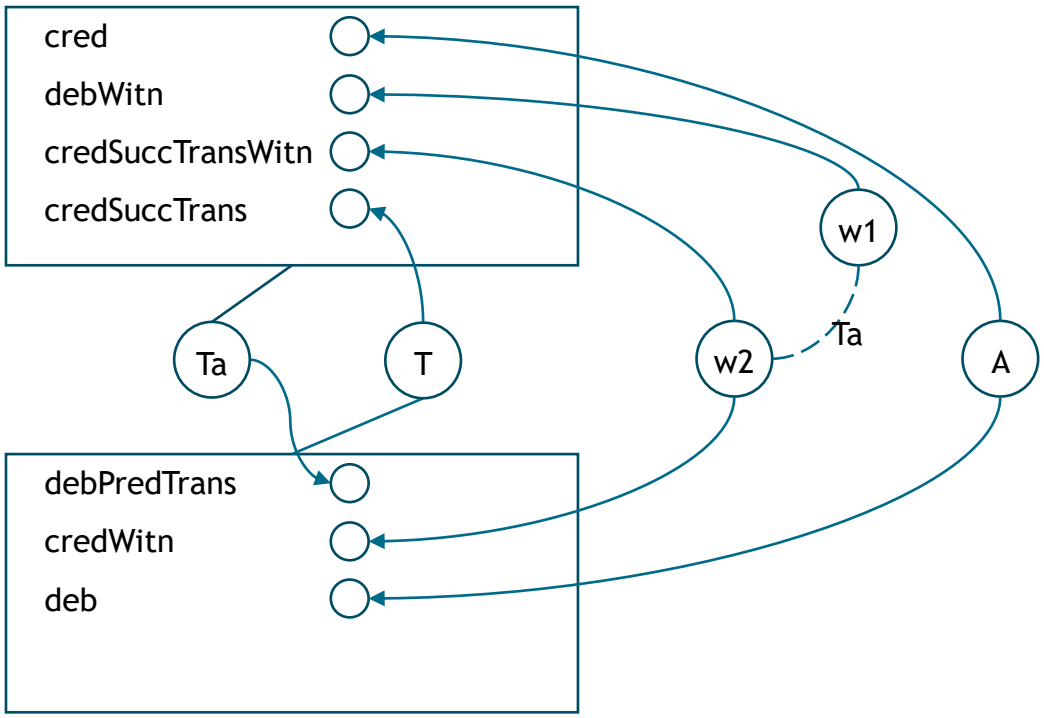
Daarom breiden we de Transactie uit met nog twee rollen:

8. Een Transactie heeft een rol `creditorSuccessorTransactionWitness` en een rol `debitorSuccessorTransactionWitness`.

Nu is bedrog door A uitgesloten. Immers:

- De getuigen van de voorgenomen Transactie T hebben óók een rol bij de vorige Transactie Ta.
- Deze getuigen worden over die vorige Transactie geïnformeerd door alle daarbij betrokken partijen, dus ook de getuigen van Ta.
- Daarom worden de getuigen van T onafhankelijk geïnformeerd over Ta.
- Als Ta al een opvolgende Transactie heeft, kunnen de getuigen dat dus onomstotelijk vaststellen.
- Zij gaan dan niet akkoord met de voorgestelde Transactie (wellicht vertrouwt B de getuige van A bij T niet, maar gelukkig heeft hij ook zijn eigen, betrouwbare getuige ingebracht!)

Met de rol `SuccessorTransactionWitness` rollen maken we als het ware een betrouwbaar kanaal waardoor de getuigen van opvolgende Transacties met elkaar communiceren. Zij kunnen elkaar volledig informeren en A is niet in staat één van die getuigen (of B) buiten te sluiten.



Legenda. Rolenames are built from:

- Cred (creditor)
- Deb (debitor)
- Witn (witness)
- Succ (succesor)
- Pred (predecessor)
- Trans (transaction)

e.g. credSuccTransWitn = witness of the next (successor) transaction of the creditor

Let op de volgende aspecten van de figuur:

- A is de crediteur (ontvangt geld) in Ta en de debiteur (betaalt geld) in T.
- B, de crediteur van Transactie T, is niet ingetekend.
- Getuige W2 is de getuige van de *crediteur* in Transactie T. A is de *debiteur* in deze Transactie. Met andere woorden: W2 is de 'tegengetuige' vanuit het standpunt van A.
- Het is juist deze tegengetuige die in de rol `credSuccTransWitn` van A's vorige Transactie wordt opgenomen.

- Ta heeft ook een tegengetuige voor A en dat is de `debWitn` (de getuige van de debiteur, terwijl A de crediteur is in Ta).
- Beide tegengetuigen van A (in respectievelijk Ta en T) communiceren onafhankelijk over Transactie Ta. A kan dus nooit liegen over Ta naar B, zonder gesnapt te worden.

Betrouwbaarheid

Bedrog is nog steeds mogelijk, als A samenspant met één der tegengetuigen. Dit is onwaarschijnlijk, omdat B tegengetuige W2 heeft gekozen en de (onbekende) transactiepartner van A in Ta W1. Belangrijk is dat elk van de partners de transactie zelf veiliger kan maken door goede, onafhankelijke getuigen te kiezen. De betrouwbaarheid van de getuigen is cruciaal.

Tenslotte is het mogelijk om meer dan twee getuigen te kiezen.

Uiteindelijke modellering

Hieronder de hoofdlijnen van de uiteindelijke modellering. De acties zijn schematisch, views ontbreken. Waarschijnlijk willen we een aantal business acties toevoegen die combinaties zijn van elementaire acties. Zo zal de Debitor bijvoorbeeld tegelijk de vorige transactie willen binden en daar direct de huidige tegengetuige aan willen toevoegen, alsmede van die vorige transactie de huidige als opvolger aanwijzen, enz.

```
Tekst "Betrouwbaar betalingsverkeer in Perspectives"

Domein model:Finance als fin: heeft

  zaken

    fin:Transactie heeft
      Properties
        intern
          $eindstandBetaler (Verplicht, Functioneel, Number)
          $eindstandOntvanger (Verplicht, Functioneel, Number)
          $bedrag  (Verplicht, Functioneel, Number)
          $transactieVoltooid (Verplicht, Functioneel, Boolean)
      Rollen
        $Debitor (Functioneel, Verplicht) gevuld door per:Persoon
        $Creditor (Functioneel, Verplicht) gevuld door per:Persoon
        $DebitorWitness (Functioneel, Verplicht) gevuld door psp:Partij
          Properties
            $akkoord (Verplicht, Functioneel, Boolean)
        $CreditorWitness (Functioneel, Verplicht) gevuld door psp:Partij
          Properties
            $akkoord (Verplicht, Functioneel, Boolean)
        $DebitorSuccessorTransaction (Functioneel, Verplicht) gevuld door fin:Transactie
        $CreditorSuccessorTransaction (Functioneel, Verplicht) gevuld door fin:Transactie
        $DebitorPredecessorTransaction (Functioneel, Verplicht) gevuld door fin:Transactie
        $CreditorPredecessorTransaction (Functioneel, Verplicht) gevuld door fin:Transactie
        $CreditorSuccessorTransactionWitness (Functioneel, Verplicht) gevuld door psp:Partij
        $DebitorSuccessorTransactionWitness (Functioneel, Verplicht) gevuld door psp:Partij
      Acties
        $Debitor bindt $DebitorWitness
        $Creditor bindt $CreditorWitness
        $Debitor bindt $DebitorPredecessorTransaction
        $Debitor bindt $DebitorSuccessorTransaction
        $Creditor bindt $CreditorPredecessorTransaction
        $Creditor bindt $CreditorSuccessorTransaction
        $creditor bindt $CreditorSuccessorTransactionWitness
        $debitor bindt $DebitorSuccessorTransactionWitness
        $Debitor betaalt
        $DebitorWitness acoordeert Transactie
        $CreditorWitness acoordeert Transactie
```