VISI
Richtlijn voor transactiekoppeling

Versie: **1.0**
Datum: **24 oktober 2008**
Status: **Definitief**

Inhoud

Inleiding 3
Deel 1: Communiceren met meerdere rollen over één onderwep 4
Deel 2: Meerdere instanties van een secundaire transactie 18

Inleiding

De VISI richtlijn voor transactiekoppeling bestaat uit twee delen. Het eerste deel behandeld de functionaliteit om met meer rollen over één onderwerp (transactie) te communiceren. Het tweede deel is gericht op de functionaliteit om een primaire transactie uit te zetten naar meerdere derde personen met verschillende instanties van een secundaire transactie.

De richtlijn is tot stand gekomen op basis van gevoerde discussies in het VISI Technisch Comité.

Deel 1: Communiceren met meerdere rollen over één onderwerp

VISI Transactions

Doel:
De huidige systematiek staat al toe transacties te definiëren welke afhankelijk zijn van andere transacties. Hiermee wordt een belangrijke vraag opgelost waarbij men met meerdere rollen (meer dan 2) over 1 onderwerp wil communiceren. Hoewel dit al mogelijk is tijdens het definiëren van een raamwerk, is de informatie die gegenereerd wordt tijdens het aanmaken van berichten onvoldoende om een 'gekoppelde transactie juist op te slaan en een juiste historie bij te houden. Dit wordt aangepast met een uitbreiding van Systematiek II (waarin wordt beschreven welke elementen een bericht bevat).

Aanpassingen Systematiek

Systematiek I behoeft geen aanpassingen

Systematiek II behoeft een aanpassing in MessageTemplate

Oorspronkelijke versie

```
ENTITY MessageTemplate;
 identification : STRING;
 dateSend : DATE;
 dateRead : DATE;
 state : STRING;
 dateLamu : DATE;
 userLamu : STRING;
 initiatorToExecutor : BOOLEAN;

 transaction : TransactionTemplate;
 template : ComplexElementTemplate;
END_ENTITY;
```

Voorgestelde aanpassing

```
ENTITY MessageTemplate;
 identification : STRING;
 dateSend : DATE;
 dateRead : DATE;
 state : STRING;
 dateLamu : DATE;
 userLamu : STRING;
 initiatingTransactionMessageID : OPTIONAL STRING;
 initiatorToExecutor : BOOLEAN;

 transaction : TransactionTemplate;
 template : ComplexElementTemplate;
END_ENTITY;
```

Note: initiatingTransactionMessageID is de ID van het voorgaande bericht
Het STRING veld bevat meer mogelijkheden dan de waarde van een ID, doch een constructie waar een exact ID ingevoerd kan worden is zo ingrijpend dat voor deze relatief simpele oplossing is gekozen.

Specifieke keuzes en achterliggende argumentaties

Voorgaande berichten worden niet als compleet bericht opgenomen:
Voordelen

- Een voorgaand bericht zou ook een attachment kunnen bevatten, als deze attachment groot is ontstaat er een hoop vervuiling
- Een voorgaand bericht zou ook weer een voorgaand bericht kunnen bevatten dit kan tot een opstapeling van voorgaande berichten leiden
- Huidige systemen zijn ingericht op 1 bericht per 12.xml file (m.u.v. project specifiek bericht, waar 0 berichten per 12.xml file aanwezig zijn). Deze keuze houdt dit valide
- Een systeem weet zelf zijn historie en heeft derhalve geen problemen met het terugvinden van de voorgaande berichten, de ID wordt gebruikt en deze is uniek .

Nadelen

- Een bericht bevat als losstaand bericht niet alle informatie zonder externe referenties zoals tot nu toe wel het geval was. Het is dus niet mogelijk om op basis van 1 bericht de gehele context van dit bericht weer te geven (verticaal),

Conclusie

- Het nadeel weegt niet op tegen de voordelen. Derhalve is gekozen om alleen referentie naar bovenliggend bericht aan te geven. Dit nadeel is (deels) goed te praten daar op basis van alleen 1 bericht al nooit gehele horizontaal context (zeg maar voorgaande en nakomende berichten) te achterhalen zijn, waardoor het ook plausibel is dat dit ook geldt voor verticale berichten (uit aanroepende transactie, dan wel berichten aangeroepen vanuit deze transactie)

Berichten die niet als direct voorgaand bericht een bericht uit een andere transactie hebben, hebben nog steeds de referentie naar het bericht uit deze transactie
Voordelen

- Conceptueel is het mooi bij overgang naar ander bericht zoveel mogelijk informatie gelijk te houden.
- Uit elk bericht is na te gaan of er een aanroepende transactie is waartoe als reactie op een bericht teruggestapt kan worden.

Nadelen

- Uit een bericht alleen is niet (altijd) af te lijden of het voorgaande bericht het bericht uit de andere transactie was.

Conclusie

- Het nadeel is alleen van toepassing als een gekoppelde transacties aangegeven moeten worden. In deze gevallen is de applicatie zich bewust van alle berichten en heeft het voldoende informatie uit alle berichten samen om de gekoppelde transactie aan te geven. De voordelen wegen dus op tegen het nadeel omdat het nadeel slechts slimmere applicaties eist en geen overall informatieverlies oplevert.

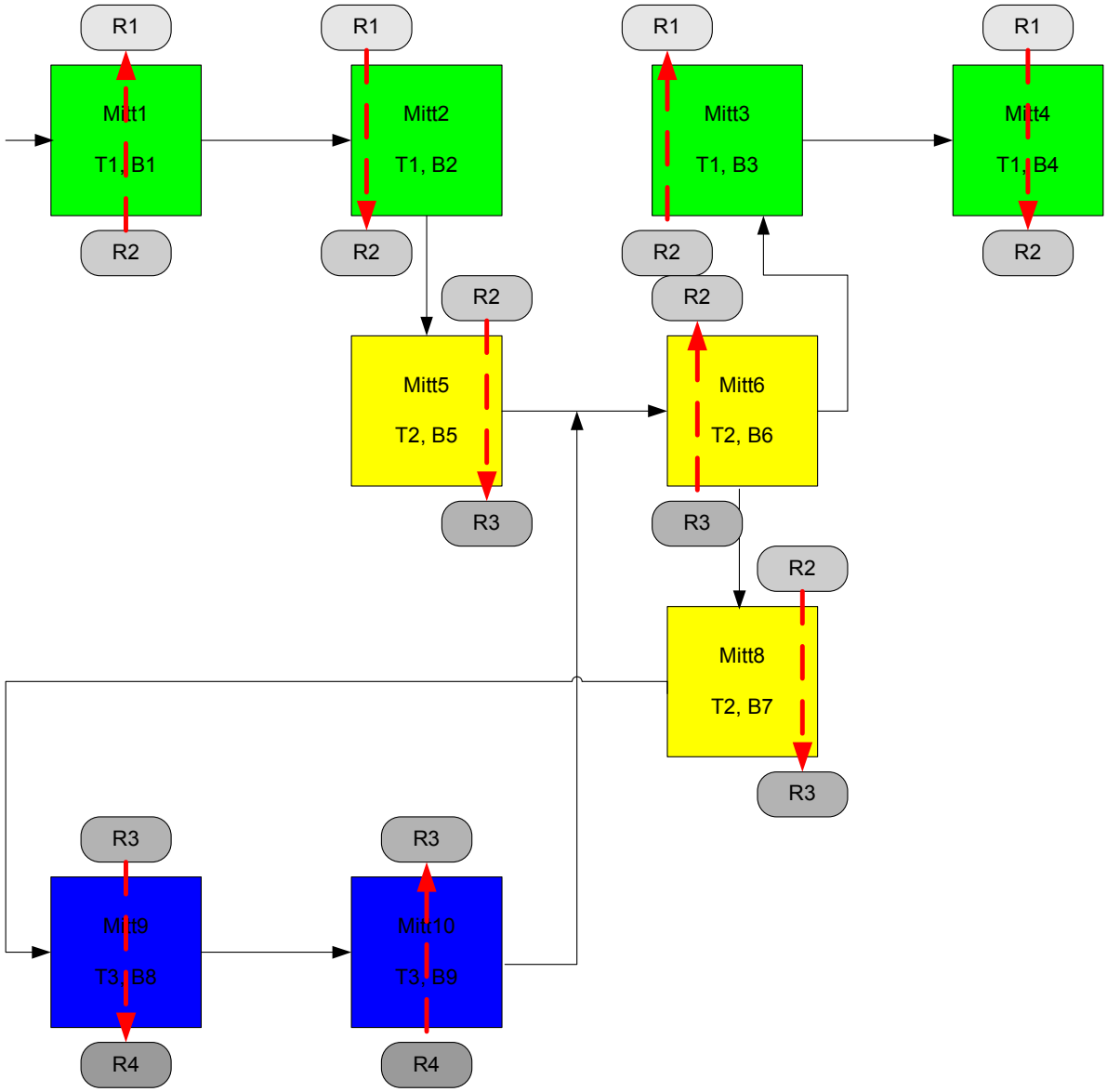
Voor bijlagen bij berichten is geen specifieke oplossing gezocht
Voordelen

- Conceptueel is het in lijn met andere VISI aanpak om de keuze van bijlagen zelf aan applicaties over te laten.

Nadelen

- Niet alle bijlagen staan automatisch op alle servers behorende bij rollen R1 t/m R4

Structuur van het raamwerk



De bovengenoemde zwarte pijlen geven de berichtenvolgorde in tijd weer. De rode pijlen geven weer welke rol het bericht verzendt en welke rol het bericht ontvangt.

De zwarte pijlen zijn niet aangegeven in de 12.xml files. Dit is informatie welke de applicatie moet onthouden indien vereist. Een uitzondering hierop vormen de pijlen waarbij een andere Transactie wordt geïnitieerd, dus van *B2 naar B5*, deze wordt opgeslagen in de instances van Mitt5, Mitt6 en Mitt8) en van *B7 naar B8*, deze wordt opgeslagen in de instances van Mitt9 en Mitt10).

De richting van de rode pijlen is aangegeven in elk 12.xml file. Elk bericht heeft altijd exact 1 rode pijl (en twee rollen).

Raamwerk in XML (_7.xml part)

Bericht B1 (12.xml part) T1

```
<?xml version="1.0" encoding="UTF-8"?>
<B1 id="">
        ...
        <initiatorToExecutor>false</initiatorToExecutor>
        <T1 id="">
                ...
                <initiator>
                        <R1 id="">
                                ...
                        </R1>
                </initiator>
                <executor>
                        <R2 id="">
                                ...
                        </R2>
                </executor>
                ...
        </T1>
        ...
</B1>
```

Bericht B2 (12.xml part) T1

```
<?xml version="1.0" encoding="UTF-8"?>
<B2 id=" messageB2">
        ...
        <initiatorToExecutor>true</initiatorToExecutor>
        <T1 id="">
                ...
                <initiator>
                        <R1 id="">
                                ...
                        </R1>
                </initiator>
                <executor>
                        <R2 id="">
                                ...
                        </R2>
                </executor>
                ...
        </T1>
        ...
</B2>
```

Bericht B5 (12.xml part) T2

```
<?xml version="1.0" encoding="UTF-8"?>
<B5 id="">
        ...
        <initiatingTransactionMessageID>messageB2</initiatingTransactionMessageID>
        ...
        <initiatorToExecutor>true</initiatorToExecutor>
        <T2 id="">
                ...
                <initiator>
                        <R2 id="">
                                ...
                        </R2>
                </initiator>
                <executor>
                        <R3 id="">
                                ...
                        </R3>
                </executor>
                ...
        </T2>
        ...
</B5>
```

Bericht B6 (12.xml part) T2

```
<?xml version="1.0" encoding="UTF-8"?>
<B6 id="">
        ...
        <initiatingTransactionMessageID>messageB2</initiatingTransactionMessageID>
        ...
        <initiatorToExecutor>false</initiatorToExecutor>
        <T2 id="">
                ...
                <initiator>
                        <R2 id="">
                                ...
                        </R2>
                </initiator>
                <executor>
                        <R3 id="">
                                ...
                        </R3>
                </executor>
                ...
        </T2>
        ...
</B6>
```

Bericht B7 (12.xml part) T2

```
<?xml version="1.0" encoding="UTF-8"?>
<B7 id="messageB7">
        ...
        <initiatingTransactionMessageID>messageB2</initiatingTransactionMessageID>
        ...
        <initiatorToExecutor>true</initiatorToExecutor>
        <T2 id="">
                ...
                <initiator>
                        <R2 id="">
                                ...
                        </R2>
                </initiator>
                <executor>
                        <R3 id="">
                                ...
                        </R3>
                </executor>
                ...
        </T2>
        ...
</B7>
```

Bericht B8 (12.xml part) T3

```
<?xml version="1.0" encoding="UTF-8"?>
<B8 id="">
        ...
        <initiatingTransactionMessageID>messageB7</initiatingTransactionMessageID>
        ...
        <initiatorToExecutor>true</initiatorToExecutor>
        <T3 id="">
                ...
                <initiator>
                        <R3 id="">
                                ...
                        </R3>
                </initiator>
                <executor>
                        <R4 id="">
                                ...
                        </R4>
                </executor>
                ...
        </T3>
        ...
</B8>
```

Bericht B9 (12.xml part) T3

```
<?xml version="1.0" encoding="UTF-8"?>
<B9 id="">
        ...
        <initiatingTransactionMessageID>messageB7</initiatingTransactionMessageID>
        ...
        <initiatorToExecutor>false</initiatorToExecutor>
        <T3 id="">
                ...
                <initiator>
                        <R3 id="">
                                ...
                        </R3>
                </initiator>
                <executor>
                        <R4 id="">
                                ...
                        </R4>
                </executor>
                ...
        </T3>
        ...
</B9>
```

Bericht B6 (12.xml part)` T2

```
<?xml version="1.0" encoding="UTF-8"?>
<B6 id="">
        ...
        <initiatingTransactionMessageID>messageB2</initiatingTransactionMessageID>
        ...
        <initiatorToExecutor>false</initiatorToExecutor>
        <T2 id="">
                ...
                <initiator>
                        <R2 id="">
                                ...
                        </R2>
                </initiator>
                <executor>
                        <R3 id="">
                                ...
                        </R3>
                </executor>
                ...
        </T2>
        ...
</B6>
```

Bericht B3 (12.xml part) T1

```
<?xml version="1.0" encoding="UTF-8"?>
<B3 id="">
        ...
        <initiatorToExecutor>false</initiatorToExecutor>
        <T1 id="">
                ...
                <initiator>
                        <R1 id="">
                                ...
                        </R1>
                </initiator>
                <executor>
                        <R2 id="">
                                ...
                        </R2>
                </executor>
                ...
        </T1>
        ...
</B3>
```

Bericht B4 (12.xml part) T1

```
<?xml version="1.0" encoding="UTF-8"?>
<B4 id="">
        ...
        <initiatorToExecutor>true</initiatorToExecutor>
        <T1 id="">
                ...
                <initiator>
                        <R1 id="">
                                ...
                        </R1>
                </initiator>
                <executor>
                        <R2 id="">
                                ...
                        </R2>
                </executor>
                ...
        </T1>
        ...
</B4>
```

Deel 2: Meerdere instanties van een secundaire transactie

Problematiek

Huidige situatie:

- Alle transacties vinden plaats tussen twee partijen
- Het is mogelijk transacties te koppelen, hierbij gaat een transactie over naar 1 andere transactie en aan het einde van deze transactie terug naar de eerste transactie (dit kan genest)

Wens:

- Een transactie doorzetten naar meerdere derde partijen.

Voorbeeld

Stel we hebben de situatie volgens Figure 1.

Hierin vinden we 3 transacties. De transactieovergangen van T1 naar T2 en van T2 naar T3 is reeds beschreven in notitie ‘Richtlijn voor transactiekoppeling’.

Huidige situatie:
Op dit moment is iemand als projectleider (R2) vanuit B2 (T1) gemachtigd exact 1 adviseur te verzoeken een offerte op te stellen (B5). Op zijn beurt kan de persoon zijnde adviseur (R3) bij bericht B7 (T2) exact 1 specialist opdracht geven e.e.a. technisch uit te werken (B8).

Wens:
De wens is dat de projectleider (R2) vanuit B2 (T1) meerdere adviseurs kan raadplegen, daarnaast dat ook de adviseur (R3)vanuit B7 (T2) ook meerdere specialisten kan raadplegen.

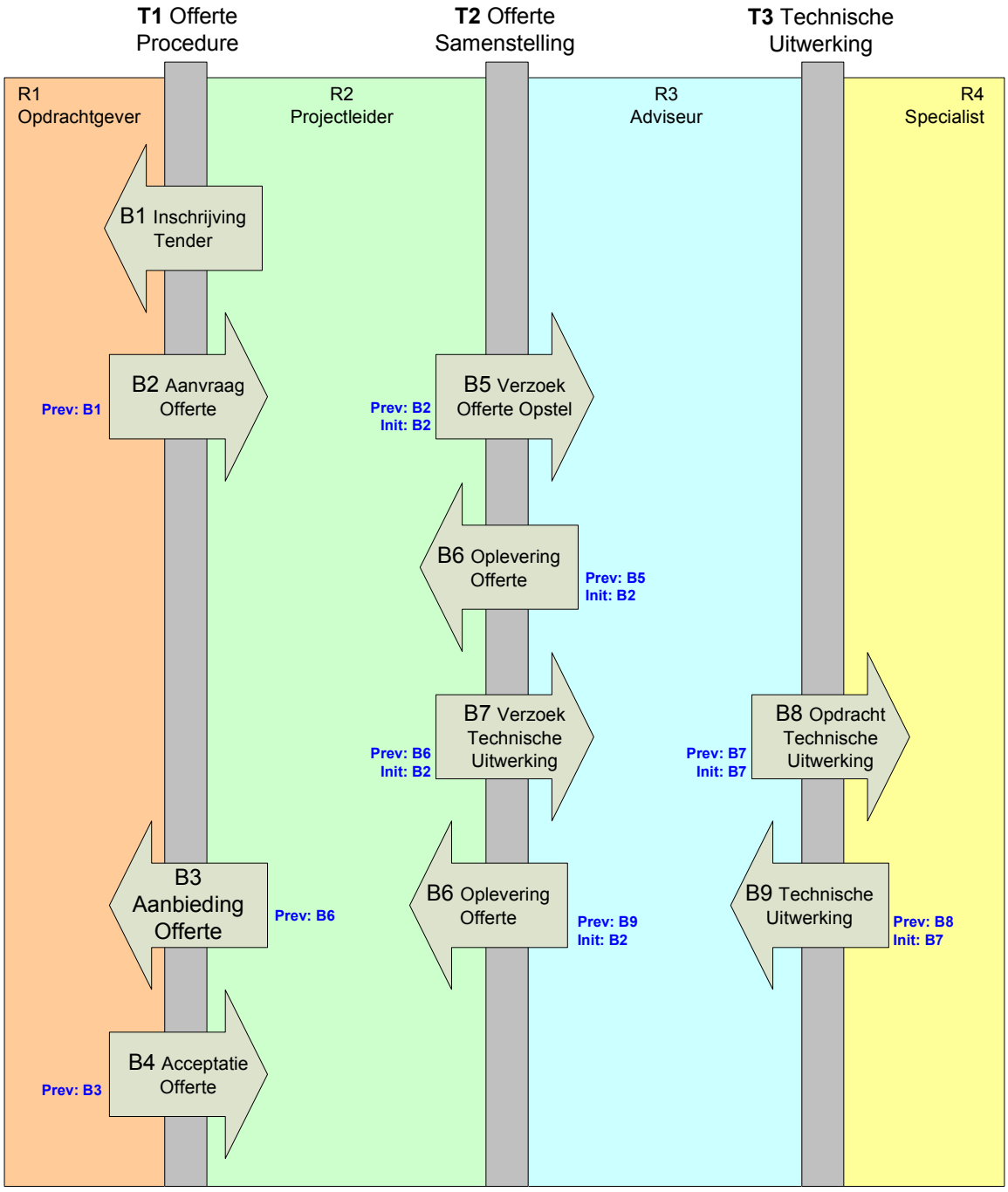


Figure 1, source Custom

Technische uitwerking

We onderscheiden 3 verschillende situaties die we willen ondersteunen:

1. vanuit een bericht in transactie X willen we meerdere transacties starten van type transactie Y waarbij we weer willen doorgaan met transactie X als ALLE transacties Y zijn afgerond
2. vanuit een bericht in transactie X willen we meerdere transacties starten van type transactie Y waarbij we weer willen doorgaan met transactie X als tenminste 1 van de type Y transacties is afgerond (1 OF MEER)
3. vanuit een bericht in transactie X willen we meerdere transacties starten van type transactie Y waarbij we weer willen doorgaan met transactie X ongeacht het aantal afgeronde transacties van het type Y zijn afgerond (0 OF MEER)

NOTE: Voor alle situaties gaan we ervanuit dat er een ongelimiteerd aantal secundary tranasacties (van het transactietype Y) mogen worden gestart.

Een nieuw element **openSecondaryTransactionsAllowed** in **MessageInTransactionType** geeft aan of bij het versturen van een bericht (uit transactie X) gewacht moet worden tot alle voorgaande berichten (indien komende uit een andere secondary transactie Y) moeten zijn voltooid. Default waarde is TRUE, dus indien TRUE of niet gedefinieerd is er slechts 1 voorgaand bericht nodig en mogen alle andere gestarte transacties van het type Y nog ergens bezig zijn. Indien FALSE dan zal gewacht moeten worden tot alle transacties van het type Y zover zijn dat ze aan dit bericht uit transactie X toe zijn.

Situatie 1

Als we ons voorbeeld van Figure 1 erbij pakken en in alle gevallen willen dat alle transacties T3 zijn afgerond voor B6 en alle transacties T2 zijn afgerond voor B3 dan zullen MessageInTransactionType voor B6 en voor B3 **openSecondaryTransactionsAllowed** op FALSE gezet moeten zijn**.**

Situatie 2

Als we ons voorbeeld van Figure 1 erbij pakken en in alle gevallen willen dat ten minste 1 transactie T3 is afgerond voor B6 en tenminste 1 transacties T2 is afgerond voor B3 dan zullen MessageInTransactionType voor B6 en voor B3 **openSecondaryTransactionsAllowed** op TRUE gezet moeten zijn. Daar TRUE de default value is, mag dit gewoon weggelaten worden.

Situatie 3

Als we ons voorbeeld van Figure 1 erbij pakken en in alle gevallen willen dat geen enkele T3 afgerond hoeft te zijn voor B6 en geen enkele transacties T2 afgerond hoeft te zijn voor B3 dan kunnen we eenvoudig B2 als voorgaande MessageInTransactionType van B3 definieren en B7 als voorgaand MessageInTransactionType van B6.

Speciale Situaties

In veel gevallen is een combinatie gewenst van beide, Zie Figure 2

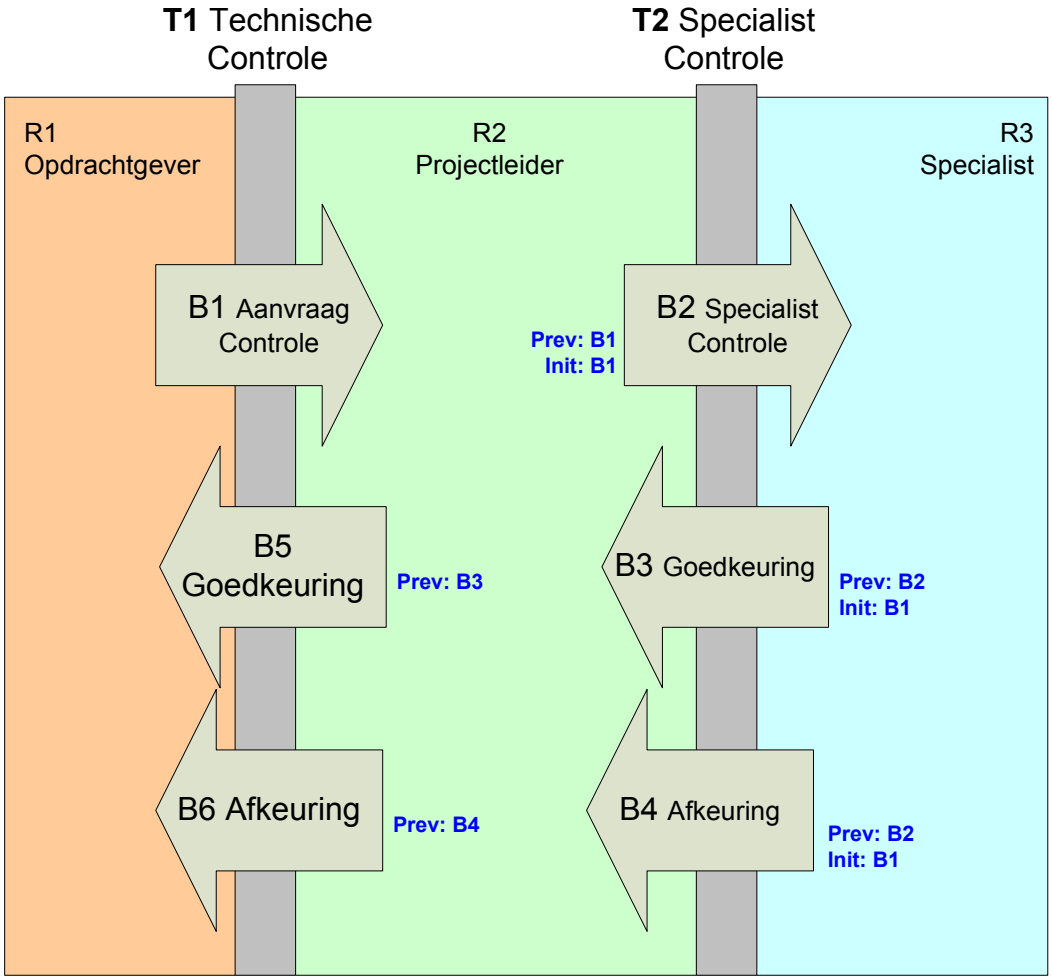


Figure 2

We zien hier een controle waarbij de opdrachtgever (R1) de projectleider (R2) vraagt een controle uit te voeren. De projectleider (R2) vraagt vervolgens verschillende specialisten (R3) een controle uit te voeren.

Wat we nu willen is de mogelijkheid iets af te keuren na 1 feedback van een specialist en pas goedkeuring te kunnen geven nadat alle specialisten goedkeuring hebben gegeven.

Dit betekend dat B5 zich in Situatie 1 bevindt en dat B6 zich in Situatie 2 bevindt. Technishc betekend dit dat MessageInTransactionType voor B5 **openSecondaryTransactionsAllowed** op FALSE heeft staan. (B6 is gewoon default en maakt net als alle andere MessageInTransactionTypes dus geen gebruik van dit element).

Aanpassingen Interpretatie

De volgende nieuwe aanname is gemaakt:

- bij een overgang naar een andere transactie (zie ook notitie 'Richtlijn voor transactiekoppeling') kan een ongelimiteerd aantal nieuwe instanties van de nieuwe Transactie worden aangemaakt
- er mogen geen andere transacties meer geïnitieerd worden op het moment dat de initierende transactie naar een volgend bericht is gegaan
- indien het nieuwe element **openSecondaryTransactionsAllowed** van **MessageInTransactionType** niet is gedefinieerd wordt hij geinterpreteerd als zijnde TRUE

Gevolgen:

- openstaande geinitieerde transacties kunnen dus blijven doorlopen ook al is de initerende transactie al verder of zelfs afgerond

Aanpassingen Systematiek

Systematiek I behoeft een aanpassing in MessageInTransactionType

Oorspronkelijke versie

```
ENTITY MessageInTransactionType;
 requiredNotify : INTEGER;
 dateLamu : DATETIME;
 userLamu : STRING;
 received : BOOLEAN;
 send : BOOLEAN;
 state : STRING;
 initiatorToExecutor : OPTIONAL BOOLEAN;

 message : MessageType;
 previous : OPTIONAL SET [0:?] OF MessageInTransactionType;
 transaction : TransactionType;
 transactionPhase : OPTIONAL TransactionPhaseType;
 previous : OPTIONAL SET [0:?] OF MessageInTransactionType;
END_ENTITY;
```

Voorgestelde aanpassing

```
ENTITY MessageInTransactionType;
 requiredNotify : INTEGER;
 dateLamu : DATETIME;
 userLamu : STRING;
 received : BOOLEAN;
 send : BOOLEAN;
 state : STRING;
 initiatorToExecutor : OPTIONAL BOOLEAN;
 openSecondaryTransactionsAllowed : OPTIONAL BOOLEAN;

 message : MessageType;
 previous : OPTIONAL SET [0:?] OF MessageInTransactionType;
 transaction : TransactionType;
 transactionPhase : OPTIONAL TransactionPhaseType;
 previous : OPTIONAL SET [0:?] OF MessageInTransactionType;
END_ENTITY;
```

Systematiek II behoeft geen aanpassingen