Systematiek I (versie 1.2)

24 oktober 2008

Contents

Element typen

AppendixType

Attributen: id
Elementen: description, startDate, endDate, state, dateLamu, userLamu, language, category, helpInfo, code
Referenties: complexElements

Een AppendixType bevat de definitie van een bijlage. Welke gegevens bijgehouden worden bij een bijlage is te definiëren in het XML veld. Simpel voorbeeld:

```
<AppendixType id="StandardAppendixType">
  <description>Standaard appendix type</description>
  <startDate>2007-01-23T00:00:00Z</startDate>
  <endDate>2007-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLamu>2007-01-23T00:00:00Z</dateLamu>
  <userLamu>bapa</userLamu>
</AppendixType>
```

ComplexElementType

Attributen: id
Elementen: description, startDate, endDate, state, dateLamu, userLamu, language, category, helpInfo
Referenties: elements

Een ComplexElementType is een verzameling van SimpleElementTypes, elk genoemd SimpleElementType komt precies het aantal keer voor dat hij genoemd wordt. Simpel voorbeeld:

```
<ComplexElementType id="MenukaartItem">
  <description>Item op menukaart</description>
  <startDate>2007-01-23T00:00:00Z</startDate>
  <endDate>2007-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLamu>2007-01-23T00:00:00Z</dateLamu>
  <userLamu>bapa</userLamu>
  <elements>
    <SimpleElementTypeRef idref="Naam"/>
    <SimpleElementTypeRef idref="Prijs"/>
    <SimpleElementTypeRef idref="Omschrijving"/>
    <SimpleElementTypeRef idref="Kalorieen"/>
  </elements>
</ComplexElementType>
```

ElementCondition

Attributen: id
Elementen: description, requiredNotify, minValue, maxValue, format, helpInfo
Referenties: message, element

De conditie op een SimpleElementType gebruikt binnen een specifiek MessageType. Simpel voorbeeld:

```
<ElementCondition id="Prijsrestrictie">
  <description>Minimale prijs van een menukaart item</description>
```

```
  <requiredNotify>0</requiredNotify>
  <minValue>5.00</minValue>
  <element>
    <SimpleElementType>Prijs</SimpleElementType>
  </element>
  <message>
    <MessageTypeRef idref="VerstrekkenVanMenukaartBericht"/>
  </message>
</ElementCondition>
```

GroupType

Attributen: id
Elementen: description, startDate, endDate, state, dateLamu, userLamu, language, category, helpInfo

De definitie van de groep voor het opslaan van bijlagen verzonden met een bericht binnen een transactie. Op het moment wordt in de praktijk geen functionaliteit door leveranciers toegekend aan dit element. Een GroupType maakt wel onderdeel uit van de structuur van een raamwerk.Simpel voorbeeld:

```
<GroupType id="StandardGroupType">
  <description>Standaard groep</description>
  <startDate>2007-12-20T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLamu>2007-12-20T00:00:00Z</dateLamu>
  <userLamu>bapa</userLamu>
</GroupType>
```

MessageInTransactionType

Attributen: id
Elementen: requiredNotify, dateLamu, userLamu, received, send, state, initiatorToExecutor, openSecondaryTransactionsAllowed
Referenties: message, previous, group, transaction, transactionPhase

De instantiatie van een MessageType binnen een TransactionType behorende bij een bepaald groep type (GroupType). Simpel voorbeeld:

```
<MessageInTransactionType id="MiTT_002">
  <requiredNotify>0</requiredNotify>
  <dateLamu>2008-01-23T00:00:00Z</dateLamu>
  <userLamu>bapa</userLamu>
  <received>true</received>
  <send>true</send>
  <state>active</state>
  <initiatorToExecutor>false</initiatorToExecutor>
  <openSecondaryTransactionsAllowed>true</openSecondaryTransactionsAllowed>
  <message>
    <MessageTypeRef idref="VerstrekkenVanMenukaartBericht"/>
  </message>
  <previous>
    <MessageInTransactionTypeRef idref="MiTT_001"/>
  </previous>
  <transaction>
    <TransactionTypeRef idref="MenukaartVerkrijgenTransactie"/>
  </transaction>
  <transactionPhase>
    <TransactionPhaseTypeRef idref="MenukaartGegeven">
```

```
  </transactionPhase>
  <group>
    <GroupTypeRef idref="StandardGroupType"/>
  </group>
</MessageInTransactionType>
```

MessageType

Attributen: id
Elementen: description, startDate, endDate, state, dateLamu, userLamu, language, category, helpInfo, code
Referenties: complexElements

De definitie van een type bericht (MessageType), hierin is ook gedefineerd hoe dit bericht gestructureerd is en welke verzameling van SimpleElementType's (via ComplexElementType's) hierbij horen. Simpel voorbeeld:

```
<MessageType id="VerstrekkenVanMenukaartBericht">
  <description>Bericht welke de menukaart bevat.</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLamu>2008-01-23T00:00:00Z</dateLamu>
  <userLamu>bapa</userLamu>
  <complexElements>
    <ComplexElementTypeRef idref="Menukaart"/>
  </complexElements>
</MessageType>
```

OrganisationType

Attributen: id
Elementen: description, startDate, endDate, state, dateLamu, userLamu, language, category, helpInfo, code
Referenties: complexElements

Definitie van een bepaalde groep organisaties, in het algemeen één instance aanwezig in een raamwerk met als reden het definiëren van de structuur van elementen die voor elke instance van dit tot object gepromote OrganisationType ingevuld moet worden. Simpel voorbeeld:

```
<OrganisationType id="StandardOrganisationType">
  <description>Standaard organisation type</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLamu>2008-01-23T00:00:00Z</dateLamu>
  <userLamu>bapa</userLamu>
</OrganisationType>
```

PersonType

Attributen: id
Elementen: description, startDate, endDate, state, dateLamu, userLamu, language, category, helpInfo, code
Referenties: complexElements

Definitie van een bepaalde groep personen, in het algemeen één instance aanwezig in een raamwerk

met als reden het definiëren van de structuur van elementen die voor elke instance van dit tot object gepromote PersonType ingevuld moet worden. Simpel voorbeeld:

```
<PersonType id="StandardPersonType">
  <description>Standaard persoons type</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLamu>2008-01-23T00:00:00Z</dateLamu>
  <userLamu>bapa</userLamu>
</PersonType>
```

ProjectType

Attributen: id
Elementen: description, startDate, endDate, state, dateLamu, userLamu, language, category, helpInfo, code
Referenties: complexElements

Definitie van een bepaalde groep projecten, in het algemeen één instance aanwezig in een raamwerk met als reden het definiëren van de structuur van elementen die voor elke instance van dit tot object gepromote ProjectType ingevuld moet worden. Simpel voorbeeld:

```
<ProjectType id="StandardProjectType">
  <description>Standaard project type</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLamu>2008-01-23T00:00:00Z</dateLamu>
  <userLamu>bapa</userLamu>
</ProjectType>
```

RoleType

Attributen: id
Elementen: description, startDate, endDate, state, dateLamu, userLamu, language, category, helpInfo, code, responsibilityFeedback, responsibilityScope, responsibilitySupportTask, responsibilityTask

De definitie van een bepaald roltype, belangrijk voor TransactionType. Simpel voorbeeld:

```
<RoleType id="ober">
 <description>Verantwoordelijk voor het opnemen en uitzetten van
bestellingen</description>
 <startDate>2008-05-04T00:00:00:00Z</startDate>
 <endDate>2008-05-04T00:00:00.00Z</endDate>
 <state>active</state>
 <dateLamu>2008-05-04T00:00:00.00Z</dateLamu>
 <userLamu>MMA</userLamu>
 <language/>
 <category/>
 <helpInfo/>
 <code/>
 <responsibilityScope/>
 <responsibilityTask/>
 <responsibilitySupportTask/>
 <responsibilityFeedback/>
</RoleType>
```

SimpleElementType

Attributen: id
Elementen: description, interfaceType, state, dateLamu, userLamu, language, category, helpInfo, valueList
Referenties: composedOf, userDefinedType

Een specificatie van een simpel elementtype (SimpleElementType). Dit elementtype beschrijft een eigenschap die binnen verschillende objectstructuren zoals bijv. in MessageType kan voorkomen (zie ook AppendixType, ProjectType, PersonType en OrganisationType), de relatie is dan altijd via ComplexElementType. Simpel voorbeeld:

```
<SimpleElementType id="Naam">
  <description>Naam van het menuitem</description>
  <interfaceType/>
  <state>active</state>
  <dateLamu>2008-01-23T00:00:00Z</dateLamu>
  <userLamu>bapa</userLamu>
  <userDefinedType>
    <UserDefinedTypeRef idref="String"/>
  </userDefinedType>
</SimpleElementType>
```

TransactionPhaseType

Attributen: id
Elementen: description, startDate, endDate, state, dateLamu, userLamu, language, category, helpInfo, code

Het definiëren van de transactiefasetypes waarin een transactie zich kan bevinden. Over het algemeen worden in VISI raamwerken de volgende transactiefasetypes gebruikt: "Start", "Verzocht", "Beloofd/Executie", "Wijziging/Hold", "Melding Gereed" en "Einde". Leveranciers van VISI-compatible software hebben vooralsnog weinig tot geen functionaliteit aan de transactiefasetypes verbonden. TransactionPhase is optional.

Simpel voorbeeld:

```
<TransactionPhaseType id="WachtenOpMenukaart">
  <description>Menukaart gevraagd maar nog niet gegeven</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLamu>2008-01-23T00:00:00Z</dateLamu>
  <userLamu>bapa</userLamu>
</TransactionPhaseType>
```

TransactionType

Attributen: id
Elementen: description, startDate, endDate, state, dateLamu, userLamu, language, category, helpInfo, code, result, basePoint
Referenties: initiator, executor, subTransactions

De definitie van een transactietype (TransactionType). Een transactietype kan tevens zelf weer naar transactietypes verwijzen. Een transactie wordt geïniteerd door een persoon behorend bij een organisatie in een bepaalde rol. Op dit niveau geven we aan welk roltype deze initiator moet bezitten (het gepromote schema zal dit vervolgens afdwingen) idem voor executor. Simpel

voorbeeld:

```
<TransactionType id="MenukaartVerkrijgenTransactie">
  <description>De transactie om te komen tot het verkrijgen van de juiste
menukaart</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLamu>2008-01-23T00:00:00Z</dateLamu>
  <userLamu>bapa</userLamu>
  <initiator>
    <RoleTypeRef idref="Consument"/>
  </initiator>
  <executor>
    <RoleTypeRef idref="Werknemer"/>
  </executor>
</TransactionType>
```

UserDefinedType

Attributen: #id
Elementen: description, state, dateLamu, userLamu, baseType, xsdRestriction, language, helpInfo

Een specificatie van een datatype (v.e. SimpleElementType). Dit DataType geeft de vorm aan invulvelden in het uiteindelijk bericht, zoals bijvoorbeeld een postcode begint altijd met 4 cijfers en dan volgen verplicht 2 letters. Simpel voorbeeld:

```
<UserDefinedType id="String">
  <description>Standaard string</description>
  <state>active</state>
  <dateLamu>2007-12-20T00:00:00Z</dateLamu>
  <userLamu>bapa</userLamu>
  <baseType>STRING</baseType>
  <xsdRestriction/>
</UserDefinedType>
```

Attributen

id

Unieke 'korte' naam van de instantiatie, deze naam zal na de promotorstap een objectnaam zijn, voorbeeld:

Raamwerkinstance

```
  <description>De attributen van een organisatie</description>
  <startDate>2007-12-12T00:00:00Z</startDate>
  <endDate>9999-12-31T00:00:00Z</endDate>
  <state>actief</state>
  <dateLamu>2007-12-12T00:00:00Z</dateLamu>
  <userLamu>visitestbeheer</userLamu>
  <language>NL</language>
  <category>S</category>
  <helpInfo>http://x/</helpInfo>
  <code>DEFAULT</code>
</OrganisationType>
```

Berichtinstance

```
  <name>TNO Building & Construction</name>
  <state>active</state>
  <dateLamu>2007-12-02T00:00:00Z</dateLamu>
  <userLamu>Peter Bonsma</userLamu>
</Organisatie>
```

Elementen

basePoint

Op het moment is aan dit element geen betekenis verbonden.

baseType

Geeft het basistype van een SimpleElementType aan. Voorbeeld:

```
<SimpleElementType id="Hoogte">
  ...
  <userDefinedType>
    <UserDefinedType id="...">
      ...
      <baseType>INTEGER</baseType>
      ...
    </UserDefinedType>
  </userDefinedType>
</SimpleElementType>
```

hierbij is dus het SimpleElementType *Hoogte* altijd een integer (eventueel met als restrictie xsdRestriction).

category

Categorie waar deze object instantiatie toe behoort, dit is een optionele waarde.

code

Binnen een raamwerk af te spreken code voor deze objectinstantiatie. Voorbeeld:

```
<... id="...">
  ...
  EAN 33156
  ...
</...>
```

dateLamu

Datum en tijd van laatste mutatie aan deze objectinstantiatie.

```
<... id="...">
  ...
  <dateLamu>2007-12-02T00:00:00Z</dateLamu>
  ...
</...>
```

description

Omschrijving van het geïnstantieerde object. Voorbeeld:

```
<... id="Deurblad">
 ...
  <description>Het blad van een vlakke deur.</description>
   ...
</...>
```

endDate

Eind datum en tijd van geldigheid van deze objectinstantiatie. In de praktijk wordt hier door levernaciers van VISI-compatible software nog geen functionaliteit aan toegekend. Voorbeeld:

```
<... id="...">
 ...
  <endDate>2007-02-03T00:00:00Z</endDate>
   ...
</...>
```

format

De opmaak van een element (optioneel).

helpInfo

Een URL/URI naar meer informatie over deze objectinstantiatie. Voorbeeld:

```
<... id="...">
 ...
  <helpInfo>http://www.visi.nl/helpInfo_object0001.html</helpInfo>
   ...
</...>
```

initiatorToExecutor

Een boolean waarde welke aangeeft in welke richting een bericht geacht wordt verstuurd te worden. Voorbeeld:

```
<MessageInTransactionType id="...">
  ...
  <initiatorToExecutor>false</initiatorToExecutor>
  ...
  <message>
    <MessageType id="OfferteAcceptatie">
      ...
    </MessageType>
  </message>
  <transaction>
    <TransactionType id="OfferteTraject">
    ...
      <initiator>
        <RoleType id="Uitvoerende">
          ...
        </RoleType>
      </initiator>
      <executor>
```

```
        <RoleType id="Opdrachtgever">
          ...
        </RoleType>
      </executor>
        ...
    </TransactionType>
  </transaction>
    ...
  <MessageInTransactionType id="...">
```

hier verwachten we dus dat bericht *OfferteAcceptatie* van Opdrachtgever (executor) naar Uitvoerende (initiator) wordt gestuurd.

interfaceType

Type interface c.q. view op dit SimpleElementType voor dit specifieke bericht. Bijvoorbeeld als het gegevenselement bedoeld is als invoer (inputText)of slechts een vaste inhoud bevat en niet aangepast mag worden (label). OP het moment wordt hier in de praktijk geen functionaliteit aan verbonden.

language

Taal die gebruikt wordt voor instantiatie van deze tot object te promoten instance, bijvoorbeeld:

```
<... id="...">
 ...
  <language>NL</language>
   ...
</...>
```

maxValue

Maximale waarde die een instance op berichtniveau uiteindelijk aan mag nemen, komt alleen voor in ElementCondition. Voorbeeld:

```
<ElementCondition id="...">
  ...
  <maxValue>999</maxValue>
  ...
</ElementCondition>
```

minValue

Minimale waarde die een instance op berichtniveau uiteindelijk aan mag nemen, komt alleen voor in ElementCondition. Voorbeeld:

```
<ElementCondition id="...">
  ...
  <minValue>0</minValue>
  ...
</ElementCondition>
```

openSecondaryTransactionsAllowed

Optional Boolean waarde die de mogelijkheid aangeeft of secundaire transacties nog niet afgerond hoeven te zijn voordat met de primaire transactie kan worden verder gegaan. De interpretatie voor "TRUE" is dat niet alle instanties van secundaire transacties hoeven te zijn afgerond voordat met de primaire transactie kan worden verder gegaan. Als de waarde "FALSE" is dienen alle instanties van secundaire transacties te worden afgerond voordat de primaire transactie hervat kan worden. Indien openSecondaryTransactionsAllowed niet is gedefinieerd wordt dit geïnterpreerd als "TRUE".

received

Boolean waarde die aangeeft of het vorige bericht ontvangen zou moeten zijn. In de praktijk wordt dit element niet gebruikt.

requiredNotify

Op het moment wordt aan het element requiredNotify geen betekenis toegekend.

responsibilityFeedback

Terugkoppeling die vanuit de verantwoordelijkheid van de rol wordt verwacht richting andere rollen (OPTIONAL STRING).

responsibilityScope

Scope/kader waarbinnen de verantwoordelijkheden behorende bij de betreffende rol zijn gedefinieerd (OPTIONAL STRING).

responsibilitySupportTask

Taken die worden uitgevoerd om andere rollen te ondersteunen. Denk hierbij bijvoorbeeld aan gedelegeerde verantwoordelijkheden (OPTIONAL STRING).

responsibilityTask

Taken die voortkomen uit de verantwoordelijkheden van de betreffende rol (OPTIONAL STRING).

result

Resultaat.

send

Boolean waarde die aangeeft of het huigie bericht inmiddels verstuurd zou moeten zijn. In de praktijk wordt dit element niet gebruikt.

startDate

Startdatum en tijd van geldigheid van deze objectinstantiatie. In de praktijk wordt hier door leveranciers van VISI-compatible software nog geen functionaliteit aan toegekend. Voorbeeld:

```
<... id="...">
 ...
  <startDate>2007-02-03T00:00:00Z</startDate>
 ...
</...>
```

state

Status van deze objectinstantiatie, op dit moment mogelijke stadia:

```
<... id="...">
 ...
  <state>active</state>
   ...
</...>
```

en

```
<... id="...">
 ...
  <state>passive</state>
   ...
</...>
```

In de praktijk wordt door enkele leveranciers van VISI-compatible software de functionaliteit aan dit element verbonden om een element type zichtbaar of niet zichtbaar te maken in de software.

userLamu

Gebuiker die de laatste mutatie aan deze objectinstantiatie heeft uitgevoerd (gewoon een string met de naam).

```
<... id="...">
 ...
  <userLamu>Peter Bonsma</userLamu>
   ...
</...>
```

valueList

Puntkomma geseparate lijst van waarden die een instance op berichtniveau uiteindelijk aan mag nemen. Oorspronkelijk was dit element bedoeld als enumeration. In de huidige praktijk wordt dit opgelost met het element type UserDefinedType en het element xsdRestriction. In de xsdRestriction worden de enumeration values aangegeven. Aan het element valueList wordt in de huidige praktijk geen betekenis toegekend.

Voorbeeld:

```
<SimpleElementType id="...">
  ...
  <valueList>Groen;Rood;Oker Geel</valueList>
  ...
</SimpleElementType>
```

xsdRestriction

Dit is de restrictie die op berichtniveau door het berichtenschema zal worden uitgevoerd op de SimpleElementType's van dit UserDefinedType.

Referenties

complexElements

Een verwijzing naar een verzameling SimpleElementType's (verzameld in ComplexElementType).
Voorbeeld:

```
<... id="Abc">
 ...
  <complexElements>
    <ComplexElementType id="ElementenSet1">
     ...
      <elements>
        <SimpleElementType id="Element_A">
         ...
        </SimpleElementType>
        <SimpleElementType id="Element_B">
         ...
        </SimpleElementType>
      </elements>
    </ComplexElementType>
    <ComplexElementType id="ElementenSet2">
     ...
    </ComplexElementType>
    <ComplexElementType id="ElementenSet3">
     ...
    </ComplexElementType>
  </complexElements>
</...>
```

Op berichtniveau ziet dat er dan als volgt uit:

```
<Abc id="...">
 ...
  <elementenSet1>
    <ElementenSet1>
      <Element_A>...</Element_A>
      <Element_B>...</Element_B>
    </ElementenSet1>
    ...
    <ElementenSet1>
      <Element_A>...</Element_A>
      <Element_B>...</Element_B>
    </ElementenSet1>
  </elementenSet1>
  <elementenSet2>
```

```
    <ElementenSet2>
     ...
    </ElementenSet2>
     ...
    <ElementenSet2>
     ...
    </ElementenSet2>
  </elementenSet2>
  <elementenSet3>
    <ElementenSet3>
     ...
    </ElementenSet3>
     ...
    <ElementenSet3>
     ...
   </ElementenSet3>
 </elementenSet3>
</...>
```

composedOf

Een SimpleElementType kan weer bestaan uit een verzameling SimpleElementType's (verzameld in ComplexElementType). Voorbeeld:

```
<SimpleElementType id="Deurblad">
  ...
  <composedOf>
    <ComplexElementType id="Maatgeving">
      ...
      <elements>
        <SimpleElementType id="Hoogte">
          ...
        </SimpleElementType>
        <SimpleElementType id="Breedte">
          ...
        </SimpleElementType>
        <SimpleElementType id="Dikte">
          ...
        </SimpleElementType>
      </elements>
    </ComplexElementType>
    <ComplexElementType id="Materiaalkeuze">
      ...
      <elements>
        <SimpleElementType id="Houtsoort">
          ...
        </SimpleElementType>
      </elements>
    </ComplexElementType>
  </composedOf>
</SimpleElementType>
```

Op berichtniveau ziet dat er dan als volgt uit:

```
<Deurblad>
  ...
  <Maatgeving id="...">
    <Hoogte>...</Hoogte>
    <Breedte>...</Breedte>
    <Dikte>...</Dikte>
  </Maatgeving>
  <Materiaalkeuze id="...">
```

```
    <Houtsoort>...</Houtsoort>
  </Materiaalkeuze>
</Deurblad>
```

element

De conditie op een SimpleElementType welke binnen een MessageType gebruikt wordt. Voorbeeld:

```
<ElementCondition id="...">
  ...
  <minValue>2000</minValue>
  ...
  <element>
    <SimpleElementTypeRef idref="Deurhoogte">
  </element>
  <message>
    <MessageType id="M">
      ...
      <complexElements>
        <ComplexElementType>
          ...
          <elements>
            <SimpleElementType id="Deurhoogte">
              ...
            </SimpleElementType>
          </elements>
        </ComplexElementType>
      </complexElements>
    </MessageType>
  </message>
</ElementCondition>
```

Hier hebben we gedefinieerd dat binnen MessageType *M* de *Deurhoogte* minimaal 2000 moet zijn, ondanks dat dit op SimpleElementType niveau niet afgedwongen wordt.

elements

Set van SimpleElementType's welke binnen een ComplexElementType aanwezig zijn. Voorbeeld:

```
<ComplexElementType id="Deur">
  ...
  <elements>
    <SimpleElementType id="Deurblad">
      ...
    </SimpleElementType>
    <SimpleElementType id="HangEnSluitwerk">
      ...
    </SimpleElementType>
    <SimpleElementType id="Bovenraam">
      ...
    </SimpleElementType>
  </elements>
</ComplexElementType>
```

Op berichtniveau ziet dat er dan als volgt uit:

```
<... id="...">
  ...
  <deur>
    <Deur>
      <Deurblad>...</Deurblad>
```

```
      <HangEnSluitwerk>...</HangEnSluitwerk>
      <Bovenraam>...</Bovenraam>
    </Deur>
    ...
    <Deur>
      <Deurblad>...</Deurblad>
      <HangEnSluitwerk>...</HangEnSluitwerk>
      <Bovenraam>...</Bovenraam>
    </Deur>
  </deur>
</...>
```

We zijn hierbij verplicht alle elementen precies één maal te noemen, deze deuren hebben dus altijd een bovenraam (VISI ondersteunt via composedOf relatie meer vrijheid). Het is zoals we zien wel mogelijk een onbeperkt aantal deuren aan te geven.

executor

De 'uitvoerende' rol (RoleType) die behoort tot een bepaalde transactie. Voorbeeld:

```
<TransactionType id="OfferteTraject">
  ...
  <executor>
    <RoleType id="Opdrachtgever">
      ...
    </RoleType>
  </executor>
  ...
</TransactionType>
```

group

De GroupType waar een bericht binnen een specifieke transactie toe behoort. Voorbeeld:

```
<MessageInTransactionType id="...">
  ...
  <message>
    <MessageType id="M">
      ...
    </MessageType>
  </message>
  <group>
    <GroupType id="G">
      ...
    </GroupType>
  </group>
  <transaction>
    <TransactionType id="T">
      ...
    </TransactionType>
  </transaction>
  ...
</MessageInTransactionType>
```

we zien hier dat bericht *M* binnen transactie *T* behoort tot group *G* (er kunnen binnen deze transactie *T* meer berichten *M* zijn die tot dezelfde of een andere group behoren).

initiator

De 'initiërende' rol (RoleType) die behoort tot een bepaalde transactie. Voorbeeld:

```
<TransactionType id="OfferteTraject">
  ...
  <initiator>
    <RoleType id="Uitvoerende">
      ...
    </RoleType>
  </initiator>
  ...
</TransactionType>
```

message

Het bericht binnen een MessageInTransactionType instance. Voorbeeld:

```
<MessageInTransactionType id="...">
  ...
  <message>
    <MessageType id="...">
      ...
    </MessageType>
  </message>
  ...
</MessageInTransactionType>
```

previous

Een MessageInTransactionType kan afdwingen dat een eerder bericht binnen een specifieke transactie uitgevoerd moet zijn (dit voorgaande MessageInTransactionType hoeft niet per definitie tot hetzelfde TransactionType te behoren). Voorbeeld:

```
<MessageInTransactionType id="...">
  ...
  <previous>
    <MessageInTransactionType id="...">
      ...
      <message>
        <MessageType id="Offerte">
          ...
        </MessageType>
      </message>
      ...
    </MessageInTransactionType>
  </previous>
  ...
  <message>
    <MessageType id="OfferteAcceptatie">
      ...
    </MessageType>
  </message>
  ...
</MessageInTransactionType>
```

subTransactions

Transacties die binnen deze transactie vallen. Voorbeeld:

```
<TransactionType id="KomenTotWerk">
  ...
  <subTransactions>
    <TransactionType id="AcquisitieTraject">
      ...
    </TransactionType>
    <TransactionType id="OfferteTraject">
      ...
    </TransactionType>
  </subTransactions>
</TransactionType>
```

hier zien we dat TransactionType *KomenTotWerk* o.a. bestaat uit de TransactionType's *AcquisitieTraject* en *OfferteTraject*.

transaction

De transactie binnen een MessageInTransactionType instance. Voorbeeld:

```
<MessageInTransactionType id="...">
  ...
  <transaction>
    <TransactionType id="...">
      ...
    </TransactionType>
  </transaction>
  ...
</MessageInTransactionType>
```

transactionPhase

De TransactionPhase waarin een specifiek MessageType binnen een specifiek TransactionType. Voorbeeld:

```
<MessageInTransactionType id="...">
  ...
  <message>
    <MessageType id="M">
      ...
    </MessageType>
  </message>
  <transaction>
    <TransactionType id="T">
      ...
    </TransactionType>
  </transaction>
  ...
  <transactionPhase>
    <TransactionPhaseType id="TP">
      ...
    </TransactionPhaseType>
  </transactionPhase>
  ...
</MessageInTransactionType>
```

hier vinden we dus een MessageType *M* binnen een specifieke transactie *T* die TransactionPhaseType *TP* bepaalt.

userDefinedType

Referentie naar UserDefinedType, geeft de te gebruiken vorm van het SimpleElementType aan. Voorbeeld:

```
<SimpleElementType id="Hoogte">
  ...
  <userDefinedType>
    <UserDefinedType id="...">
      ...
      <baseType>INTEGER</baseType>
      ...
    </UserDefinedType>
  </userDefinedType>
</SimpleElementType>
```

hierbij geeft userDefinedType dus voor het SimpleElementType *Hoogte* aan dat dit altijd een integer moet zijn (eventueel met als restrictie xsdRestriction).