

## Inleiding

Op 30 augustus 2013 is er een uitgebreid overleg geweest bij de Waarderingskamer over het omgaan met historie in de LV WOZ en de problemen die daarmee zijn in de StUF-standaard. Omdat er op de korte termijn, die nodig is voor het aansluiten op de LV WOZ, geen patch op de StUF-standaard kan worden uitgebracht is ervoor gekozen voorlopig alleen het sectormodel woz0312 te patchen. Er is in de StUF0301 namespace een extra schema gemaakt met daarin het attribute sleutelSynchronisatie. Dit attribute is vervolgens opgenomen in de laatste patch van woz0312. Dit document beschrijft het werken met het attribute StUF:sleutelSynchronisatie. Daarnaast geeft dit document vooruitlopend op de definitieve patch c.q. een nieuwe versie StUF0302 van de onderlaag op een aantal punten een aanscherping van de specificaties in StUF0301.

Dit document is leidend ten opzichte van de op 30 augustus gegeven presentatie. Dit document is onder nogal wat tijdsdruk tot stand gekomen en daarmee nog niet ideaal. De plaatjes uit de presentatie zijn er bijvoorbeeld nog niet in verwerkt. Het is hopelijk wel een voldoende startpunt voor de betrokken partijen om daadwerkelijk verder te gaan met de lastige implementatie van historie en met het genereren van synchronisatieHistorie berichten uit de database. Eventuele opmerkingen kunnen het beste via het forum gedeeld worden en er zal vervolgens zo spoedig mogelijk op gereageerd worden in de vorm van aanpassingen/bijstellingen van dit document.

## Inhoudsopgave

Inleiding.....	1
Het werken met het attribute StUF:sleutelSynchronisatie.....	1
Aanscherpingen in het werken met wijzigingen en correcties.....	3
Notatiewijze.....	3
Eisen aan een los Lk01-bericht met mutatiesoort W.....	3
Eisen aan een los Lk01-bericht met mutatiesoort F.....	4
Het corrigeren dat een relatie nooit bestaan heeft.....	4
Lk01-berichten binnen Sh01/02-bericht.....	5
Het corrigeren van de oudste beginGeldigheid.....	6
Synchronisatiebericht voor relaties.....	6
Corrigeren tijdvakRelatie.....	6
Het afleiden van een synchronisatiebericht uit de “foto's”.....	7

## Het werken met het attribute StUF:sleutelSynchronisatie

In een synchronisatieActueel en een synchronisatieHistorisch bericht is het verplicht om in de topfundamenteel, in alle relaties van de topfundamenteel en in alle relaties van relaties in de topfundamenteel het attribute StUF:sleutelSynchronisatie op te nemen met als waarde een sleutel die de topfundamenteel of de relatie in combinatie met zijn entiteittype uniek identificeert binnen het hele synchronisatiebericht. Het is toegestaan dat bijvoorbeeld de topfundamenteel en een relatie of twee relaties met verschillende entiteitstypen worden geïdentificeerd met dezelfde sleutelSynchronisatie. Twee verschillende relaties van hetzelfde entiteitstypen mogen nooit dezelfde sleutelSynchronisatie hebben.

Het is niet toegestaan om het attribute StUF:sleutelSynchronisatie op te nemen binnen gerelateerden of binnen relaties van gerelateerden. Als een relatie uniek geïdentificeerd is, dan is ook bekend wat de gerelateerde is. Het in het bericht identificeren van de gerelateerde is daarom niet nodig. Het is

bovendien niet altijd mogelijk om een gerelateerde te identificeren met StUF:sleutelSynchronisatie, omdat de gerelateerde afkomstig kan zijn uit een ander sectormodel dat dit attribute niet kent.

Een voor de hand liggende waarde voor StUF:sleutelSynchronisatie is de sleutelVerzendend van het object. Indien er geen sleutelVerzendend beschikbaar is, dan zal bij het maken van het synchronisatiebericht een sleutel voor het object gegenereerd moeten worden die in combinatie met het entiteitstype uniek is binnen het synchronisatiebericht. Elk voorkomen van de topfundamenteel of een relatie binnen de topfundamenteel wordt geïdentificeerd met dezelfde sleutelSynchronisatie. Vanuit het synchronisatieActueel bericht kan het object op basis van zijn actuele gegevens in de database gevonden worden. Bij het opbouwen van de historie is altijd bekend om welke relatie het gaat door zijn sleutelSynchronisatie.

Wanneer een topfundamenteel of relatie voor het eerst voorkomt binnen oudste of wijziging, dan dienen alle elementen met de initieel geldige waarde te worden opgenomen. Gerelateerden worden altijd opgenomen met de actuele gegevens. In volgende berichten waarin de topfundamenteel of een relatie gewijzigd of gecorrigeerd wordt, worden uitsluitend de elementen relevant voor de wijziging opgenomen. Kerngegevens die voor de wijziging of correctie niet relevant zijn mogen niet worden opgenomen.

De sleutelSynchronisatie is nodig in het synchronisatieActueel bericht binnen een synchronisatieHistorisch bericht. Om de specificatie uniform te houden is sleutelSynchronisatie ook verplicht in een los synchronisatieActueel bericht, ook al is sleutelSynchronisatie daar functioneel niet nodig. Het is verplicht om in een synchronisatieActueel bericht binnen een synchronisatieHistorisch bericht ook alle relaties op te nemen die beëindigd zijn (eindRelatie gevuld), omdat anders deze relaties niet geïdentificeerd kunnen worden. In een beëindigde relatie is eindRelatie gevuld. Wanneer een relatie per ongeluk is opgevoerd, maar in de werkelijkheid nooit heeft bestaan, dan wordt deze in het synchronisatieActueel bericht opgenomen met eindRelatie-gelijk aan beginRelatie de gegevens zoals deze golden op het moment dat werd geconstateerd dat de relatie nooit heeft bestaan. In dat geval dienen Het nooit bestaan hebben van de relatie wordt aangegeven door de elementen beginGeldigheid en eindGeldigheid gevuld te worden te vullen met geenWaarde, omdat tijdvakGeldigheid geen betekenis heeft voor een relatie die nooit bestaan heeft. De StUF-standaard specificeert op dit moment niets over beëindigde relaties in een synchronisatieActueel bericht.

Binnen het element wijziging in een synchronisatieHistorisch bericht worden alleen elementen en relaties opgenomen die nodig zijn om de wijziging of correctie door te geven. Kerngegevens mogen alleen worden opgenomen, als ze voor de wijziging of correctie relevant zijn en anders niet.

Er is voor gekozen om StUF:sleutelSynchronisatie verplicht te maken. Dit impliceert dat in 'oud' in relaties met `xsi:nil="true"` en `StUF:noValue="geenWaarde"` `StUF:verwerkingssoort="T"` c.q. in nieuw in relaties met `xsi:nil="true"` en `StUF:noValue="geenWaarde"` `StUF:verwerkingssoort="V"` of "E" het attribute StUF:sleutelSynchronisatie moet worden opgenomen. De waarde heeft geen betekenis, maar er wordt voorgeschreven om deze te vullen met dezelfde waarde als in 'nieuw' in geval van verwerkingssoort "T" c.q. als in 'oud' in geval van verwerkingssoort "E" of "V".

## Aanscherpingen in het werken met wijzigingen en correcties

### Notatiewijze

Een wijzig- of correctiekennisgeving kan genoteerd worden als

$$A(t_{bo}, t_{eo}) \rightarrow Z((t_{bn}, t_{en}); t_r)$$

Hierin staat:

A voor de waarden in oud;

Z voor de waarden in nieuw

$t_{bo}$  en  $t_{eo}$  voor beginGeldigheid en eindGeldigheid in oud

$t_{bn}$  en  $t_{en}$  voor beginGeldigheid en eindGeldigheid in nieuw

$t_r$  voor het tijdstipRegistratie in nieuw.

### Eisen aan een los Lk01-bericht met mutatiesoort W

Voor een losse wijzigkennisgeving (mutatiesoort W), waarmee historie wordt opgebouwd moet gelden:

$$\begin{aligned} t_{bo} &< t_{eo} \\ t_{eo} &= t_{bn} \\ t_{en} &= \infty \end{aligned}$$

Deze drie eisen kunnen gecontroleerd worden zonder de database te raadplegen.

Daarnaast moet gelden:

$t_{bo} \geq$  de grootste beginGeldigheid in de database van alle waarden met historie die ongelijk aan elkaar zijn in A en Z

$t_r >$  de grootste tijdstipRegistratie voor het object in de database

Als aan één van deze eisen niet wordt voldaan, dient een foutmelding (TODO: foutmelding definiëren) gegeven te worden en kan het bericht niet verwerkt worden.

Voor een wijzigkennisgeving (mutatiesoort W) geldt niet als eis dat  $t_{bo}$  groter moet zijn dan de grootste beginGeldigheid van alle waarden van het te wijzigen object. Het is toegestaan om een element te wijzigen met een  $t_{bn}$ , die kleiner is dan de beginGeldigheid van één of meer andere elementen. Het gaat erom dat een nu geldige waarde wordt gewijzigd en het is niet relevant of er andere waarden zijn met een grotere beginGeldigheid. Omdat dit voor sommige leveranciers lastig te realiseren is, wordt in de conformiteitstoets voor het aansluiten voorlopig toegestaan dat wijzigingen van een gegeven met een beginGeldigheid kleiner dan de beginGeldigheid van enig ander gegeven van dat object worden doorgegeven door middel van een synchronisatieHistorisch bericht in plaats van een wijzigkennisgeving met mutatiesoort W. Op termijn zullen ook de leveranciers die hiervoor het synchronisatieHistorisch bericht gebruiken overgaan naar het gebruik

van een wijzigkennisgeving met mutatiesoort W, omdat dit beter aansluit bij het NORA voorschrift dat waar mogelijk op basis van gebeurtenissen wordt gecommuniceerd.

### **Eisen aan een los Lk01-bericht met mutatiesoort F**

Voor een losse correctiekennisgeving (mutatiesoort F), waarmee historie wordt opgebouwd moet altijd gelden:

$$t_{eo} = t_{en} = \infty$$

$t_r$  > de grootste tijdstipRegistratie voor het object in de database

De correctie heeft uitsluitend betrekking op de elementen in A en Z, die als beginGeldigheid in de database  $t_{eo}$  hebben. Een correctiekennisgeving mag alleen verwerkt worden, als deze gegevens in de database als eindGeldigheid  $\infty$  (StUF:noValue="geenWaarde") hebben. Zo niet, dan dient een foutmelding gegeven te worden (TODO: nog definiëren).

Ter identificatie kan een bericht kerngegevens bevatten met een andere beginGeldigheid. Ook kan een bericht vanuit het schema verplicht gegevens bevatten met een andere beginGeldigheid. Deze gegevens met een andere beginGeldigheid worden niet geraakt door de correctie.

Een correctie kan alleen verwerkt worden als A en Z minstens één gegeven bevatten waarvoor geldt dat  $t_{bo}$  gelijk is aan de beginGeldigheid in de database.

Als  $t_{bo} = t_{bn}$ , dan gaat het om een correctie van één of meer waarden.

Als  $t_{bo} \neq t_{bn}$ , dan gaat het om een correctie van beginGeldigheid, desgewenst gecombineerd met een correctie van één of meer waarden. Bij een correctie van beginGeldigheid mag  $t_{bn}$  elke waarde hebben behalve  $\infty$ . Bij een verschuiving naar de toekomst schuift de eindGeldigheid van het aansluitende blok in de materiële historie mee. Als bij een verschuiving naar het verleden de nieuwe beginGeldigheid groter is dan de beginGeldigheid van het aansluitende blok in de materiële historie, dan schuift de eindGeldigheid van het aansluitende blok in de materiële historie mee. Als de nieuwe beginGeldigheid ligt voor de beginGeldigheid van het eerste aansluitende blok of daaraan voorafgaande blokken, dan worden deze blokken verdrongen en schuift de eindGeldigheid van het eventueel daar nog voor liggende blok mee met de nieuwe beginGeldigheid.

Bij het corrigeren van een beginGeldigheid moet altijd gecheckt worden of er geen kerngegevens of andere verplichte gegevens zijn met beginGeldigheid gelijk aan  $t_{bo}$  waarvan het niet de bedoeling om de beginGeldigheid te verschuiven. Als dit het geval is, dan kan deze correctie van beginGeldigheid niet worden doorgegeven met een los Lk01-bericht met mutatiesoort F, maar zal deze onderdeel dienen te zijn van een synchronisatieHistorisch bericht.

### **Het corrigeren dat een relatie nooit bestaan heeft**

Soms wordt een relatie per ongeluk opgevoerd, maar heeft deze in de werkelijkheid nooit bestaan. Dit kan gecorrigeerd worden in een correctiekennisgeving met mutatiesoort F door de relatie te vervangen door de correcte relatie (verwerkingssoort "R") of door de relatie te verwijderen (verwerkingssoort "V"). In beide gevallen dient in 'oud' de relatie te worden opgenomen

- met het nu geldende tijdvakRelatie
- met de kerngegevens
- met ~~in tijdvakGeldigheid beginGeldigheid en eindGeldigheid beide als waarde “geenWaarde” (dit geeft aan dat de relatie nooit bestaan heeft) het nu geldende tijdvakGeldigheid (dwz met  $t_{eo} = \infty$ )~~
- met als tijdstipRegistratie het moment van het doorvoeren van de correctie in de registratie.

In geval van een vervanging bevat 'nieuw' de nieuwe relatie en in geval van een verwijdering bevat 'nieuw' een lege relatie.

### **Lk01-berichten binnen Sh01/02-bericht**

Bij de bespreking van het attribute StUF:sleutelSynchronisatie is al aangegeven dat kennisgevingen binnen het historie element in een synchronisatieHistorisch bericht alleen de gegevens mogen bevatten relevant voor het opbouwen van de historie. Kerngegevens waarmee niets gebeurt mogen niet worden opgenomen. In het eerste voorkomen van een topfundamenteel of relatie worden alle elementen opgenomen met de initieel geldige waarde. In de volgende voorkomens worden uitsluitend de gegevens relevant voor de wijziging opgenomen. Met uitzondering van het hier gestelde gelden voor het overige de hierboven gegeven regels voor wijzig- en correctiekennisgevingen binnen een synchronisatieHistorisch bericht.

Daarnaast mag binnen een kennisgeving slechts één waarde voor tijdstipRegistratie voorkomen. Het is niet toegestaan om binnen één kennisgeving wijzigingen voor een verschillend tijdstipRegistratie te combineren. Binnen het element oudste mogen dus geen relaties voorkomen die een ander tijdstipRegistratie hebben als het opvoeren van het object zelf.

Voor wat betreft de correctiekennisgeving zijn er nog wel een paar aanvullende regels. Binnen het Sh01/02-bericht mogen correctiekennisgevingen met mutatiesoort F voorkomen, waarbij  $t_{eo}$  een geldige waarde heeft in plaats van  $\infty$ .  $t_{en}$  kan zowel een geldige waarde hebben als  $\infty$  zijn. Het gaat hierbij om kennisgevingen die een correctie in de materiële historie doorvoeren. We zullen een correctiekennisgeving met voor  $t_{eo}$  een geldige waarde in het vervolg ook wel aanduiden als een historische correctiekennisgeving.

Een historische correctiekennisgeving dient aan de volgende eisen te voldoen:

1.  $t_{bo} = t_{bn}$ , als  $t_r >$  de grootste tijdstipRegistratie voor het object in de database  
Op deze regel is één uitzondering: Als  $t_{bo}$  gelijk is aan de kleinste waarde van beginGeldigheid voor het object, dan mag  $t_{bn}$  een waarde ongelijk aan  $t_{bo}$  hebben. Zie voor meer details hieromtrent de paragraaf over het corrigeren van beginGeldigheid hieronder. Deze regel impliceert dat in een historische correctiekennisgeving alleen een eindGeldigheid of de eerste beginGeldigheid kan worden gecorrigeerd. Voor het verschuiven van eindGeldigheid gelden dezelfde regels als voor het verschuiven van beginGeldigheid in een los Lk01-bericht.  
Deze regel impliceert ook dat het in de StUF-standaard beschreven tussenvoegen van een waarde niet langer is toegestaan.
2. Voor alle elementen in de kennisgeving moet gelden dat het tijdvakGeldigheid van die waarden in de database gelijk is aan het tijdvakGeldigheid in oud (er is weer een uitzondering bij het corrigeren van de oudste beginGeldigheid). De verwerking van een

synchronisatieHistorisch bericht dient afgebroken te worden, als niet aan deze eis wordt voldaan. Er wordt dan een ongeldig synchronisatieHistorisch bericht geleverd.

Een historische kennisgeving hoeft niet aan de eis te voldaan te worden dat  $t_r$  groter is dan de grootste tijdstipRegistratie voor het object in de database, want ~~maar~~  $t_r$  mag ook gelijk zijn aan de grootste tijdstipRegistratie in de database. In dat geval worden gegevens domweg overschreven met nieuwe waarden. Er moet dan wel gelden dat hetzij  $t_{bo} \neq t_{bn}$ , hetzij  $t_{eo} \neq t_{en}$ , zodat slechts voor een deel van het oorspronkelijke tijdvakGeldigheid de oude waarden vervangen worden door de nieuwe waarden. Het tussenvoegen van een waarde zoals beschreven in de StUF-standaard wordt nu gerealiseerd met twee kennisgevingen. In een eerste kennisgeving worden eindGeldigheid en zonodig de waarde gecorrigeerd. In een tweede en zonodig volgende kennisgeving met dezelfde tijdstipRegistratie wordt vervolgens de al dan niet gecorrigeerde waarde deels overschreven met een nieuwe waarde, zie ook de voorbeelden in de presentatie.

In een historische correctiekennisgeving mag  $t_{en}$  de waarde  $\infty$  hebben. Dit maakt het mogelijk om een gegeven dat per ongeluk samen met een kerngegeven een andere waarde heeft gekregen te corrigeren naar de oorspronkelijke waarde.

Het voorbeeldbericht voorbeeldCorrectieHistorieRelatie.xml geeft in de vorm van een synchronisatieRelatie bericht een voorbeeld van het corrigeren van een historische oppervlakte in een relatie. Dit voorbeeld is eenvoudig onder te brengen in een synchronisatiebericht voor een WOZ-object.

### ***Het corrigeren van de oudste beginGeldigheid***

In geval van een correctie van de oudste beginGeldigheid dienen alle elementen van het object in het bericht te zitten met de waarde op beginGeldigheid. Het is toegestaan om tijdens het schuiven waarden nog te corrigeren. Omdat elementen met allerhande waarden voor eindGeldigheid in het bericht zitten, moeten  $t_{eo}$  en  $t_{en}$  als waarde hebben de kleinste eindGeldigheid voor de waarden geldig op de oorspronkelijke  $t_{bo}$ . Als beginGeldigheid kleiner wordt gemaakt dan mag deze elke waarde kleiner dan  $t_{bo}$  hebben. Als  $t_{bo}$  naar de toekomst wordt geschoven, dan moet  $t_{bn}$  kleiner of gelijk zijn aan  $t_{eo}$ . Zonodig dienen voorafgaand aan het schuiven met beginGeldigheid eerst andere correcties te worden doorgevoerd om te zorgen dat alle waarden een eindGeldigheid hebben die groter of gelijk is aan de nieuwe beginGeldigheid. De correctie van de oudste beginGeldigheid krijgt vervolgens dezelfde tijdstipRegistratie als de eerdere correcties.

### **Synchronisatiebericht voor relaties**

Er is besloten om het voorgestelde synchronisatiebericht voor relaties nog niet te gebruiken bij de conformiteitstoets voor de LV WOZ. Een eventuele correctie voor een relatie wordt doorgegeven als onderdeel van het synchronisatiebericht voor de topfundamenteel waarbinnen de relatie voorkomt.

### **Corrigeren tijdvakRelatie**

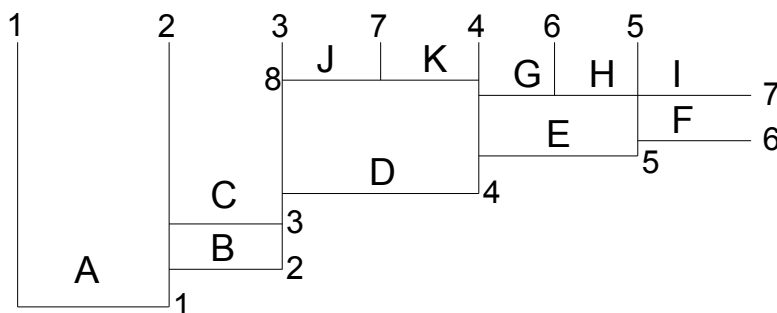
De tijd ontbrak om ook het corrigeren van een tijdvakRelatie nog in tekst te beschrijven. Hier zullen de voorbeelden uit de presentatie gevolgd moeten worden. Het voorbeeldbericht

voorbeeldCorrectieHistorieRelatie2.xml geeft een voorbeeld van het corrigeren van tijdvakRelatie in de vorm van een synchronisatieRelatie bericht. Dit is eenvoudig onder te brengen in een compleet synchronisatiebericht voor een WOZ-object.

## Het afleiden van een synchronisatiebericht uit de “foto's”

De op 30 augustus gegeven presentatie bevatte een aantal slides over het transformeren van een complexe linked list representatie met foto's naar een simpeler representatie. Op de terugweg in de trein heb ik me gerealiseerd dat ik daarmee een verkeerde insteek heb gekozen. Hieronder beschrijf ik mijn nieuwe inzichten voor de opbouw van een synchronisatieHistorisch bericht.

Onderstaande figuur geeft een voorbeeld linked list representatie die in het vervolg wordt gebruikt.



De letter in een blok staat voor de voor dat blok geldende set waarden. Per oplopend tijdstipRegistratie dienen de blokken geanalyseerd te worden.

Voor het eerste tijdstipRegistratie vinden we het losse blok A. Omdat er links geen blok meer is en rechts er een blok naast ligt met een grotere tijdstipRegistratie is dit blok oorspronkelijk opgevoerd in een toevoegkennisgeving:

$$(T^1) A((t_{b1}, \infty); t_{r1})$$

Voor het tweede tijdstipRegistratie vinden we het blok B met links ervan blok A en rechts ervan blok D met een grotere tijdstipRegistratie en eronder niets. Dit betreft een wijzigkennisgeving van A naar B:

$$(W) A(t_{b1}, t_{b2}) \rightarrow B((t_{b2}, \infty); t_{r2})$$

Voor het derde tijdstipRegistratie vinden we een blok C met daaronder blok B en links met een groter tijdstipRegistratie blok D. Dit betreft een correctiekennisgeving van B naar C:

$$(F) B(t_{b2}, \infty) \rightarrow C((t_{b2}, \infty); t_{r3})$$

Voor het vierde, vijfde en zesde tijdstipRegistratie vinden we weer een wijzigkennisgeving analoog aan de wijziging van A naar B.

1 De letter tussen haakjes voor de formule staat voor een toevoegkennisgeving (T), een wijzigkennisgeving (W) of een correctiekennisgeving (F)

Voor het zevende tijdstipRegistratie treffen we een ingewikkelder situatie aan, met boven de blokken E, H en I en daaronder de blokken E en F. Deze situatie analyseren we door eerst voor de twee blokken met eindGeldigheid  $\infty$  het verschil in waarden te bepalen. Vervolgens bepalen we voor de waarden die verschillen de grootste beginGeldigheid (in het vervolg aangeduid met  $t_{bo}$ ). In dit voorbeeld vinden we dan een waarde kleiner dan de de beginGeldigheid van het blok E. Omdat op het te analyseren tijdstipRegistratie er boven het meest linkse blok zich een nieuw blok met de waarden set E bevindt is de beginGeldigheid van de mutatie de eindGeldigheid van het bovenste blok E (in het vervolg aangeduid met  $t_{bn}$ ). Als over de hele lengte de waarden in de blokken boven en onder tijdstipRegistratie verschillen, dan is de beginGeldigheid in de mutatie gelijk aan de beginGeldigheid van het meest linkse blok onder tijdstipRegistratie. Vervolgens checken we of de beginGeldigheid van de mutatie groter is dan de grootste beginGeldigheid voor de waarden die verschillen in de twee blokken met eindGeldigheid  $\infty$ . Zo ja, dan gaat het om een wijziging en zo nee, dan gaat het om een correctie waarbij ook beginGeldigheid is verschoven.

In geval van een wijziging wordt vervolgens nog gecheckt of voor alle blokken boven en onder geldt dat het verschil in overeenstemming is met het verschil op de twee blokken met eindGeldigheid  $\infty$ . Zo ja, dan is het een gewone kennisgeving

$$(W) F(t_{bo}, t_{bn}) \rightarrow I((t_{bn}, \infty); t_{r7})$$

Zo nee, dan gaat het om een correctie die we kunnen doorvoeren als:

$$(F) E(t_{b4}, t_{b5}) \rightarrow E((t_{b4}, \infty); t_{r7}) + (F) E(t_{b4}, \infty) \rightarrow H((t_{b6}, \infty); t_{r7}) + (F) H(t_{b6}, \infty) \rightarrow I((t_{b5}, \infty); t_{r7})$$

In geval van een correctie kan op soortgelijke wijze worden afgeleid of het gaat om een actuele correctie of een correctie die moet worden doorgevoerd als meer dan één correctiekennisgeving met hetzelfde tijdstipRegistratie. Op het moment van schrijven ontbreekt de tijd om dit in meer detail uit te werken.

Voor de laatste tijdstipRegistratie vinden we onder het blok D en daarboven de blokken J en K. Dit is een historische correctie, omdat zich rechts van D een blok bevindt met een tijdstipRegistratie kleiner dan de onderhanden tijdstipRegistratie. Het is geen eenvoudige correctie van een waarde, omdat zich boven twee blokken bevinden. Het is geen verschuiving van eindGeldigheid, omdat de waarden in het blok rechts van D niet voorkomen in een blok boven de onderhanden tijdstipRegistratie. De eenvoudigste manier om dit door te voeren is met twee kennisgevingen met hetzelfde tijdstipRegistratie:

$$(F) D(t_{b3}, t_{b4}) \rightarrow J((t_{b3}, t_{b4}); t_{r8}) + (F) J(t_{b3}, t_{b4}) \rightarrow K((t_{b7}, t_{b4}); t_{r8})$$

Ook hier is een uitgebreidere analyse nodig om te komen tot het complete algoritme.