

# ProRail onderzoekt mogelijkheden GML

ProRail, de beheerder van de Nederlandse Railinfrastructuur, voerde onlangs een proef uit met gegevensuitwisseling via de open standaard GML. ProRail wil GML en andere open standaarden in de toekomst gebruiken voor gegevensuitwisseling met leveranciers – ter vervanging van de ‘gesloten’ CAD-standaarden als Autocad en Microstation. De proef moest zo breed mogelijk de haalbaarheid van GML testen. Daarom zijn ook partners van ProRail uitgenodigd aan de proef deel te nemen. Zo kon de volledige informatieketen worden getest.

ProRail is verantwoordelijk voor de kwaliteit van het Nederlandse spoor in termen van onder meer betrouwbaarheid, veiligheid en capaciteit. ProRail geeft invulling aan die verantwoordelijkheid, door als een regisseur ontwikkelopdrachten aan ingenieursbureau's, en bouw- en onderhoudsopdrachten aan aannemers te verstrekken. De laatste jaren worden ook geïntegreerde opdrachten verstrekt aan consortia in zogeheten ‘procescontracten’ en ‘design & construct’ opdrachten.

ProRail moet uiteraard voortdurend inzicht hebben in haar infrastructuur, en dat inzicht ook kunnen geven aan partijen die opdrachten voor ProRail uitvoeren. Daarom zijn alle leveranciers verplicht om, aan het einde van hun opdracht, gegevens aan te leveren over de gebouwde of aangepaste infrastructuur: welke objecten staan waar en welk type object is het? Zulke gegevens worden nu doorgaans geleverd als digitale tekeningen in CAD-formaten zoals Microstation en Autocad. ProRail ontvangt op deze wijze vele duizenden tekeningen per jaar en verstrekt er ook weer vele duizenden.

In een mooie toekomst is alle informatie over de volledige ProRail-infrastructuur vastgelegd in één grote objectgerichte database. Met een druk op de knop kunnen selecties worden gemaakt van gegevens waardoor ProRail-medewerkers in één oogopslag kunnen zien hoe de infrastructuur er bij ligt en waar beheersmatige probleempunten liggen. Bovendien kan iedereen beschikken over alle informatie die zich in de organisatie bevindt.

Helaas is de huidige situatie minder rooskleurig. De gegevensinfrastructuur van ProRail laat zich het beste samenvatten als ‘divers’. Gegevens zijn doorgaans juist niet onderling gekoppeld en daardoor lastig terug te vinden. We spreken van ‘collecties’ van vaak losse data zonder duidelijk verband. Om te beginnen heeft ProRail honderdduizenden ‘platte’ tekeningen: analoge tekeningen en rasterformaten.



*H. Dekker,  
ProRail;  
co-auteurs  
R. Kamp, ProRail  
en W. Quak,  
TU Delft*

Daarnaast zijn er vele CAD-bestanden in hoofdzakelijk Microstation en AutoCad. Ook deze bestanden zijn van sterk variërend allooi. Soms zijn ze net zo plat als de canvas kaarten waarvan ze zijn overgetekend; in sommige gevallen zijn er projecten geweest waardoor ze meer informatie bevatten over de getekende objecten en relaties tussen die objecten. Naast de kaarten en CAD-bestanden zijn er binnen ProRail ook nog enkele databases die soms onderling aan elkaar of aan CAD-bestanden gerelateerd zijn. Soms ook niet.

## Uitdagingen

De uiteenlopende kwaliteit en samenhang van de gegevenscollecties weerspiegelt de huidige en vroegere organisatiestructuren. De infrastructuur is opgebouwd met behulp van vele verschillende technieken en disciplines: energievoorziening, baan- en bovenbouw, telecom, civiele techniek, beveiliging, enz. Elke discipline heeft in de loop der jaren een eigen werkwijze opgebouwd van omgaan met gegevens. De enorme hoeveelheid gegevens bij ProRail reflecteert dat: de collecties met tekeningen en databases zijn voor een groot deel nog niet goed samenhangend en daardoor kan ook de kwaliteit sterk variëren.

ProRail staat dus voor twee uitdagingen. Ten eerste: alle gegevens uit het verleden omzetten naar een samenhangende digitale omgeving. De tweede uitdaging is te zorgen dat gegevens die in de toekomst worden geleverd, direct op de juiste manier op de juiste plaats terecht komen.

Dit artikel, en de business case voor GML, richt zich met name op het laatste punt: leveringen van gegevens over de infrastructuur op de juiste plaats in de organisatie krijgen. Om te beginnen, is daarvoor een duidelijke afspraak tussen ProRail en haar leveranciers nodig over de wijze waarop gegevens aangeleverd dienen te worden aan ProRail. Aan een CAD-bestand of een excelsheet hebben we niets. De gegevens moeten gestructureerd en volgens een afgesproken structuur, syntax en semantiek worden geleverd. Duidelijk iets waar GML zich goed voor leent.

## Waarom GML?

ProRail heeft de beleidskeuze gemaakt om toe te werken naar uitwisseling van selecties van objectgerichte gegevens en tevens te werken naar het gebruik van open standaarden waar dat mogelijk is. Met andere woorden: ProRail wil de komende jaren voor de uitwisseling van gegevens het gebruik van leveranciersspecifieke, gesloten formaten zoveel mogelijk uitfaseren. Op dit vlak ontstaat overigens regelmatig spraakverwarring. Voor alle duidelijkheid vermelden we hier dat ProRail nadrukkelijk niet wil voorschrijven waarin leveranciers dienen te ontwerpen. Iedereen kan en mag ontwerpen en ontwikkelen in het softwarepakket van zijn keuze. Echter, de levering van gegevens aan ProRail dient plaats te vinden volgens de specificaties van ProRail.

ProRail ziet GML als het vervolg op tekeningen en digitale CAD-bestanden. Het is een vorm van gegevensuitwisseling die past in een wereld van objectgerichte databases. Wanneer GML-bestanden als uitwisselingsformaat worden gehanteerd tussen ProRail en de ingenieursbureaus en aannemers, kunnen de geleverde gegevens makkelijker worden gecontroleerd en in databases worden opgenomen. Ten opzichte van de traditionele CAD-formaten brengt GML belangrijke voordelen met zich mee:

- *efficiëntere verwerking*: de software voor verwerken van GML-bestanden is veel eenvoudiger te maken dan voor CAD-bestanden van meerdere verschillende software-platforms. CAD-bestandsformaten zijn bovendien vaak gesloten, dus moeilijk in te zien zonder gebruik van leveranciersspecifieke API's (Application Programming Interface) of andere speciale software;
- *betere informatie*: GML-bestanden zijn beter en makkelijker te specificeren en te verwerken waardoor ProRail meer informatie krijgt over haar infrastructuur en beter kan achterhalen wat de eventuele effecten zijn van beheersmaatregelen;
- *minder projectuitloop*: een standaard CAD-bestand bevat gegevens over soms honderden meters infrastructuur tegelijk. Infrastructurele projecten kunnen soms niet vooruit omdat een bepaalde tekening is uitgeleend aan een ander project.

Met GML kunnen véél specifiekere selecties (doorsnedes) worden gemaakt waardoor projecten minder vaak vertraging oplopen;

- *beschikbaarheid*: GML-bestanden zijn tot het einde der tijden te openen en te lezen. Hetzelfde kan niet worden gezegd voor leveranciersafhankelijke formaten. Dit kan van belang zijn vanwege de archiefwet. Door de open specificatie van het bestandsformaat kan in het uiterste geval zelfs een 'viewer' vanaf de grond worden opgebouwd. Voor gesloten bestandsformaten is dit niet mogelijk.



GML is bovendien een marktbrede standaard. ProRail profiteert als grote opdrachtgever in de Nederlandse markt van vergaande standaardisatie binnen een goed functionerende marktplaats waarin informatie over infrastructuur en infra-projecten met vele partijen kan worden uitgewisseld zonder telkens grote en kleine investeringen in specifieke bestandsformaten. Met behulp van GML kan voor ProRail een vaste gegevensdefinitie (schema) worden vastgelegd. De voordelen daarvan kunnen eenvoudig worden berekend wanneer men kijkt naar de tijd die wordt besteed om telkens opnieuw tot overeenstemming te komen met leveranciers, grote projecten en derden - met betrekking tot de te leveren informatie en de wijze waarop.

## Aanpak

Om de haalbaarheid en wenselijkheid van GML-gegevensuitwisseling in de praktijk te testen, besluit ProRail in het voorjaar van 2006 een proef te doen met deze standaard. Binnen deze proef moet worden bepaald of GML een toepasbare en praktische structuur biedt voor de uitwisseling van gegevens met leveranciers.

Om de haalbaarheid op een zo breed mogelijk vlak te beproeven, worden aan de pilot de volgende eisen gesteld:

1. ten eerste moeten er gegevens worden uitgewisseld van een scala aan technische disciplines: civiele techniek, energievoorziening, baan- en bovenbouw, enz. Wie immers een stuk spoor aanlegt, gebruikt daarbij al deze technieken - dus wie gegevens uitwisselt, zal dat ook moeten. Met andere woorden: de pilot mag de huidige praktijk, met al haar versnippering van gegevens, niet als randvoorwaarde accepteren;
2. de tweede belangrijke eis is dat het gehele proces van gegevensuitwisseling geraakt wordt: gegevens exporteren bij ProRail, gegevens versturen naar de leverancier, aldaar gegevens wijzigen en vervolgens weer terugleveren en importeren in de ProRail-database.

Voor de uitvoering van de pilot wordt een team ingericht van vier personen die, ieder vanuit hun eigen discipline, een bijdrage kunnen leveren: techniek, informatie-analyse, geografie en projectmanagement. De planning is om in een periode van een half jaar de volledige pilot af te ronden en de resultaten te presenteren aan het management.

Een mogelijk knelpunt in het project is dat er op het raakvlak van XML/GML en objectmodellering weinig kennis beschikbaar is – net zo min bij ProRail als in de markt. Deze kennis moet dus tijdens het project worden ontwikkeld. Hier moet een belangrijk risico worden onderkend: wat als de gevraagde kennis te diepgaand of specifiek is om zelfstandig op te bouwen? Om dit risico te verminderen, wordt de mogelijkheid open gehouden om externe kennis van buiten ProRail in te zetten.

De uitvoering van de pilot wordt vervolgens ingericht aan de hand van de volgende stappen:

1. ontwerp van het gegevensmodel;
2. inrichting van een database;
3. opstellen van een GML-schema;
4. ontwikkelen van een export-script;
5. exporteren van gegevens naar GML;
6. opstellen en versturen van de wijzigingsopdracht naar leveranciers;
7. importeren van gewijzigde GML-gegevens.

### Stap 1: Ontwerp van het gegevensmodel

Alvorens er geëxporteerd kan worden, moet er een database zijn met gegevens over de railinfrastructuur. Zoals eerder geconstateerd, is er binnen ProRail niet één database met gestructureerde gegevens die daarvoor als uitgangspunt kan dienen: er zijn juist vele databases die vaak eerder CAD- en rasterbestanden bevatten, dan objectgerichte gegevens voorzien van syntax en semantiek. Er moet dus een database worden ontworpen die als bron kan dienen voor de gegevensexport.

De volgende belangrijke constatering is dat het best gemakkelijk zou zijn wanneer het gegevensmodel van de database en het gegevensmodel van het GML-bestand overeen zouden komen. Binnen de pilot wordt daarom al snel besloten om eerst een gegevensmodel te ontwerpen dat als basis zal dienen voor zowel de database als het exportbestand.

Gelukkig is het project voor dat gegevensontwerp niet volledig op zichzelf aangewezen. Er is namelijk binnen ProRail een zogenaamde 'Objectenstructuur' beschikbaar. Deze objectenstructuur zou de basis moeten zijn voor de grootscheepse objectisering van de informatiesystemen. Theoretisch dus een mooie basis voor het gegevensmodel voor de pilot.

Na enig onderzoek wordt vervolgens besloten om het gegevensmodel te definiëren in de UML-schematiek (Unified Modelling Language). Deze beslissing is genomen op basis van de volgende argumenten:

- vastleggen in een strak omliggende standaard als UML voorkomt onduidelijkheden en vaagheden. Dubieuze of multi-interpretabele constructies komen in UML vrij snel bovendrijven - dit zou wellicht anders pas bij het technisch ontwerp van de database of het GML-schema aan het licht komen;
- Het gegevensmodel zou omwille van uitwisselbaarheid van gegevens, met andere grote civiele opdrachtgevers, moeten aansluiten op de NEN3610-standaard. Deze standaard is op het moment dat de pilot start net definitief geworden - en vastgelegd in UML. Dit werk is grotendeels uitgevoerd op de TU Delft, waar ProRail ook contacten heeft en daarmee de aanwezige kennis en hulp kan gebruiken;

- via de TU Delft wordt ook ontdekt dat er een open source Java-programma beschikbaar is dat UML-bestanden direct converteert naar een GML-schema.

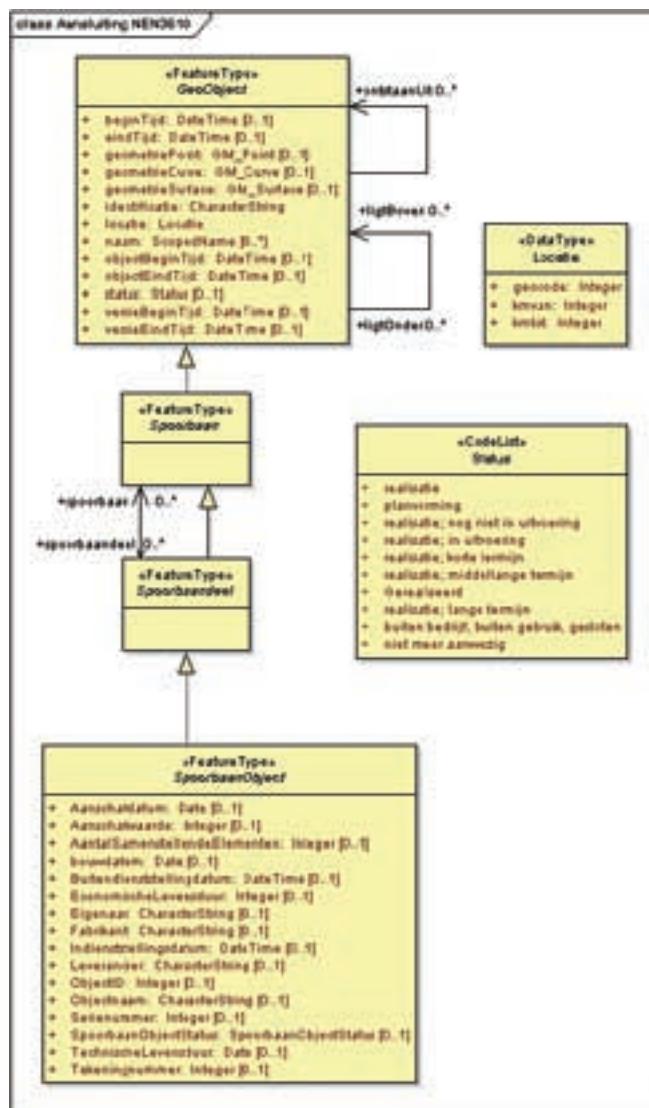


Fig. 1. Aansluiting RailGML en NEN3610.

Gegevensmodel: hoe de theorie de praktijk ontmoette

Het pilotproject gaat vervolgens hoopvol aan de slag om de ProRail-objectenstructuur vast te leggen in UML. Het uitgangspunt daarbij zijn de documenten in Microsoft Word en Excel die tot dat moment zijn gebruikt. Tijdens het proces van omzetting naar UML blijkt hoe strak de eisen zijn die UML stelt: verschillende onduidelijkheden en zaken die voor meerdere uitleg vatbaar zijn, blijken hun weg gevonden te hebben naar de objectenstructuur. Het zijn voor een informatie-analist herkenbare zaken:

- wat is in de oorspronkelijke structuur de relatie tussen elementen en direct ondergelegen elementen? Het blijkt de ene keer te gaan om de

relatie 'is onderdeel van' en in andere gevallen 'is een instantie van'. Helaas is dit nooit aangegeven en vaak lastig te achterhalen.

- van de (beschikbare) attributenlijsten is ook niet duidelijk op welk niveau deze moeten worden gekoppeld aan objecten. Ook deze onduidelijkheid kan uiteraard niet worden vertaald naar het UML-model.

Een ander knelpunt blijkt in deze fase juist minder ingewikkeld dan verwacht: de aansluiting van de ProRail-objectenstructuur op NEN3610. Het 'spoorse' gedeelte van NEN3610 is namelijk dermate algemeen van opzet dat zonder al te veel moeite een oplossing wordt gevonden (fig. 1).

Voor het vastleggen van de UML, wordt gebruik gemaakt van het softwarepakket Enterprise Architect. Dit pakket is voldoende gebruiksvriendelijk en beschikt bovendien over de mogelijkheid om de UML te exporteren naar een formaat dat kan worden omgezet naar een GML-schema.

### Stap 2: inrichten van de database

Nadat het gegevensmodel bekend is, wordt gestart met het inrichten van een Spatial database. Gelukkig staat de pilot er ook op dit vlak niet alleen voor. Op het moment dat de GML-pilot van start gaat, is er tevens een ander project waarop kan worden aangesloten: 'De Objectgerichte Basisbeheerkaart'. Dit project werkt aan een Spatial database met daarin een objectgerichte gegevensstructuur gevuld met basisobjecten van de spoorinfrastructuur. Deze gegevens worden door het project geconverteerd vanuit bestaande CAD-bestanden.

Na een kort onderzoek wordt besloten dat deze Spatial database over vol-



doende mogelijkheden beschikt om, naast de 'basisobjecten', ook ingericht te worden voor het meer omvangrijke gegevensmodel dat de pilot heeft ontwikkeld. Uiteraard wordt hiervoor een aparte omgeving binnen de database ingericht.

Vervolgens wordt de testdatabase gevuld met fictieve gegevens: een paar honderd meter spoor in een mooie boog.

### Stap 3: ontwikkelen van het GML-schema

Voor de ontwikkeling van het GML-schema wordt gebruik gemaakt van het open source Java-programma 'ShapeChange' (kader) dat tijdens de pilot beschikbaar is in versie 0.3. Vanuit de UML-ontwerpsoftware kan een bestand



worden geëxporteerd (voor de liefhebbers: een XML-bestand van het UML-ontwerp) dat ShapeChange 'begrijpt' en kan converteren naar een GML-schema.

Bij de ontwikkeling van het schema is een belangrijk uitgangspunt dat zoveel mogelijk van bestaande standaarden wordt uitgegaan. Zo kan dus al

Fig. 2. Samenhang tussen XML, GML en de Prorail-implementatie.

worden begonnen met het standaard GML-schema. Voor het inrichten van de infrastructuur kan ook gekeken worden naar NEN3610.

### Stap 4: ontwikkeling GML-exportscript

Na export van het UML-gegevensmodel naar GML kan worden begonnen met de ontwikkeling van het exportscript. Tijdens de ontwikkeling van dit script komen nog enkele knelpunten naar voren die hoofdzakelijk te maken hebben met de onbekendheid van de projectmedewerkers met XML-syntax.

Omdat GML tekst-gebaseerd is, zijn de opties voor de te kiezen scripttaal of exportsoftware feitelijk oneindig. Binnen de organisatie is echter het pakket FME beschikbaar dat specifiek voor dit doel is gemaakt. De keuze is daarom snel gemaakt om met deze software te gaan experimenteren. Met succes, zo blijkt al snel.

De FME-conversiesoftware beschikt over de mogelijkheid om databasetabellen te 'mappen' op velden van een zelf te bepalen GML-schema. Dit mappen gebeurt in een grafische omgeving waardoor het converteren zelf hoofdzakelijk een kwestie van 'point & click' is. Ook in deze fase van het pilotproject komen echter de nodige knelpunten boven. Ook nu weer moeten ze hoofdzakelijk worden gevat onder de noemer 'kennisontwikkeling'. En ook nu weer blijkt het doorzettingsvermogen van de pilotmedewerkers te zegevieren, ditmaal over gecompliceerde XML-syntax-kwesties.

Problemen met syntax komen gelukkig snel naar voren zodra XML gevalideerd wordt: met verschillende online- en offline-tools kan worden gecontroleerd of de XML-data en het XML-schema consistent zijn.

## Stap 5 & 6: exporteren gegevens en versturen opdracht aan leveranciers

Op het moment dat het gegevensmodel, de database en het exportscript afgerond zijn is natuurlijk de daadwerkelijke gegevensexport van het fictieve stuk infrastructuur (fig. 3) een koud kunstje. Het GML-schema en de GML-gegevens werden vervolgens verstuurd naar de vier deelnemende leveranciers. De bestanden werden bovendien vergezeld van een eenvoudige fictieve opdracht: het aanleggen van honderd meter rangeerterrein met baanlichaam, wissels, spoor, bovenleidingpalen, bovenleiding. Ook kregen alle deelnemers de beschikking over alle tot dan toe ontwikkelde kennis aangaande UML/XML/GML.

Al vroeg in de pilot moest het risico worden onderkend dat leveranciers, ondanks hun initiële enthousiasme, mogelijk niet konden voldoen aan de vraag om het GML-bestand te wijzigen. De materie is gecompliceerd, mogelijk te gecompliceerd om in enkele maanden bij te spijkeren zonder bezoldiging. Om deze reden wordt besloten om ook zelf de opdracht uit te voeren, zodat de volgende stap niet in gevaar komt.

Enkele maanden na de verstrekking van de opdracht blijkt de inschatting van dit risico terecht te zijn geweest. Van de vier deelnemende bureaus blijkt niet één in staat om aan de opdracht te voldoen. Dit is natuurlijk op zich een belangrijk signaal over de mate waarop leveranciers van ProRail klaar zijn voor GML – wellicht een van de belangrijkste resultaten van het pilotproject.

## Stap 7: importeren gewijzigde gegevens

Deze stap kan gelukkig wel worden uitgevoerd aangezien het project ook zelf een gewijzigd bestand heeft aangeemaakt. Voor de import van gegevens wordt ook gebruikt gemaakt van FME.

Hoewel het importeren van gegevens op zich een eenvoudige zaak is, komen op dit vlak nog enkele interessante kwesties aan het licht. Het terug geleverde bestand bevat, in het geval van de pilot, een groot aantal objecten: sommige zijn gewijzigd, sommige zijn verwijderd, sommige zijn nieuw. Vraag voor de organisatie is: op welke manier worden deze wijzigingen in de database opgenomen? Voor versiebeheer van objecten had de pilot nog geen voorzieningen getroffen. Verwacht wordt dat dit in een dienstdoende oplossing uiteraard wel het geval zal zijn. Andere vragen: wat te doen wanneer één object wordt gesplitst? Wordt het oorspronkelijke object bijvoorbeeld verwijderd en vervangen? En wat gebeurt er in zo'n geval in de database dan met onderhoudsgegevens? Enzovoorts.

## Conclusie

Het doel van het project was kennisontwikkeling over de mogelijkheden en wenselijkheid uitwisseling van spoor-specifieke gegevens (met name oplevering bij projecten) middels de open standaard GML. Dit doel is ruimschoots bereikt. Daarnaast zijn voor de organisatie enkele waardevolle feiten boven tafel gekomen.

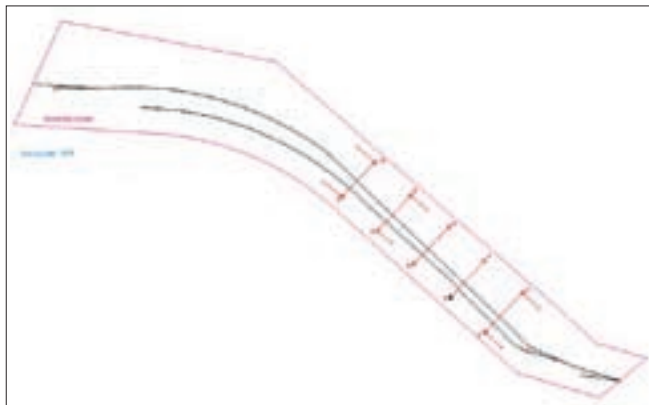


Fig. 3. Het fictieve stuk infrastructuur dat is gebruikt in de test.

Om te beginnen is er een goede businesscase voor de uiteindelijke overstap naar GML als uitwisselingsformaat. Zodra bovendien (een deel van) de infragegevens zijn opgenomen in een spatiaal database, is ProRail ook technisch in staat om die gegevens te exporteren naar dat formaat - en weer te importeren.

Informatiekundig gesproken moet er nog veel gebeuren. De objectenstructuur is in haar huidige vorm niet geschikt als basis voor grootschalig objectgerichte informatie-opslag c.q. informatie-uitwisseling. Er zijn teveel open einden en onduidelijkheden. Aanbevolen wordt de huidige methodiek op basis van Excel- en Worddocumenten te vervangen door een methodiek dat zich beter leent voor het beschrijven van gegevensstructuren, zoals UML. Daarnaast zijn de databases (denk onder meer aan tekeningcollecties) binnen ProRail doorgaans nog niet volledig ingericht volgens de eigen objectenstructuur, of gekoppeld daaraan.

Als laatste moet worden geconcludeerd dat ook de leveranciers van ProRail op dit moment nog niet in staat zijn om GML-bestanden te leveren. De benodigde expertise is schaars maar dit kan snel veranderen wanneer softwareleveranciers de ingeslagen weg blijven volgen. ■

ShapeChange is een Java-programma dat op dit moment helaas alleen via de command line kan worden aangestuurd. Dit betekent dat (in Windows) een DOS-box moet worden geopend om ShapeChange instructies uit te voeren. Uiteraard kan ook een batchbestand worden ingericht. De input van ShapeChange bestaat uit een XML-bestand waarin een UML-schema is vastgelegd. De output is eenvoudiger: een GML-schema.