

XML oplossingen bouwen. Het is zo langzamerhand een doel op zich geworden voor elke ontwikkelaar. Voor de Java ontwikkelaar is het dankzij de JaxPack alvast een fluitje van een cent, als we Sun mogen geloven. In dit derde artikel over de Jaxpack besluit Marc Portier met een overzicht van de nieuwste toevoegingen aan de bundel Java API's voor het behandelen van XML.



JaxR: business repository's en interface metadata

Java API's voor de verwerking van XML (deel 3)

“Dat hangt er vanaf.” Zo ongeveer klinkt de oervorm waar alle ingenieurs-antwoorden in volle eer en geweten van zijn afgeleid. Het soort intellectuele eerlijkheid waar we al te vaak onze besluiteloosheid achter wegstoppen. In het slechtste geval is het allicht een houding die serieuze vooruitgang in de weg gaat zitten. Maar ook de omgekeerde houding, het zonder consideratie ‘blind’ erin vliegen, is nauwelijks een beter alternatief te noemen. Het beheersen van de ‘afhankelijkheden’ blijft een nobel doel dat onze aandacht verdient. Dat is voor onze computersystemen niet anders. In de huidige dagen van de webservices-hoogconjunctuur worden in dit verband al snel twee stevige technieken tot ont koppeling voorgedragen. In het vorige artikel van deze reeks werden ze aangehaald als ‘ontkoppeling in beheer’ en ‘ontkoppeling van technisch platform’, en werd aangereikt hoe XML messaging in het algemeen en JAXM in het bijzonder de ont koppeling najaagt. Al met al blijft het hier bij een redelijk recht-toe recht-aan gebeuren dat eenduidige boodschappen stuurt tussen duidelijk gekende partijen: we weten misschien niet wanneer de dienst actief is, en op welke technologie ze draait, maar we zijn duidelijk nog wel afhankelijk van wie de dienst waar uitbaat, en welke versie er actief is (zie ook kader ‘Ontkoppeling van versies met XML’).

VOLLEDIG AUTONOM

Onafhankelijkheid van de locatie van de dienst (ook wel ‘location transparency’) wordt klassiek gerealiseerd door het gebruik van een directory service. Binnen de webservices context heet het alvast iets prozaïscher ‘business registry’. Het is alvast, in de beste dotcom tra-

ditie, een term die er goed in slaagt hoge verwachtingen te wekken en ze vervolgens niet waar te maken. Of de gelanceerde verwachtingen ook echt in de corporate noden zijn, is nog maar zeer de vraag. Vermoedelijk kunnen we het antwoord nu al uit de ervaringen van de verschillende “virtual marketplace” initiatieven leren. Het tot stand laten komen van actieve klant-leverancier-relaties blijft een typisch menselijk proces van eerste contact tot eerste contract. Een protocol voor vertrouwen blijkt er nog niet te zijn. Het is vanuit het andere oogpunt ook één van de eerste lessen die de uitbaters van de publieke UDDI servers hebben gemerkt: heel vaak worden de nodige beschrijvende velden van de bedrijven en hun diensten daar gebruikt om eerder te adverteren welke diensten je zou willen aanbieden dan wel welke je al actief in het portfolio hebt. In het publieke forum is er nu eenmaal heel duidelijk een incentive verbonden aan het ‘rekken’ van de waarheid. Een volledig autonoom systeem bouwen dat in een dergelijk milieu toch het menselijk aanvoelen van betrouwbare kwaliteit kan gebruiken in zijn besluitvorming veronderstelt dan het soort artificiële intelligentie waarvan men de voorbije vijftig jaar heeft beloofd dat ze in de komende tien jaar effectief zal zijn.¹

NOVELL Het doet natuurlijk allemaal niets af aan de zinnigheid van een zekere mate van nuchtere “location transparency” in onze systemen. Je dient niet te weten hoe je naar elke straat in het land reist als je maar weet

¹ Zie ook <http://andstuff.org/TurtlesAllTheWayDown> voor het klassieke verhaal omtrent de zoektocht naar betekenis.

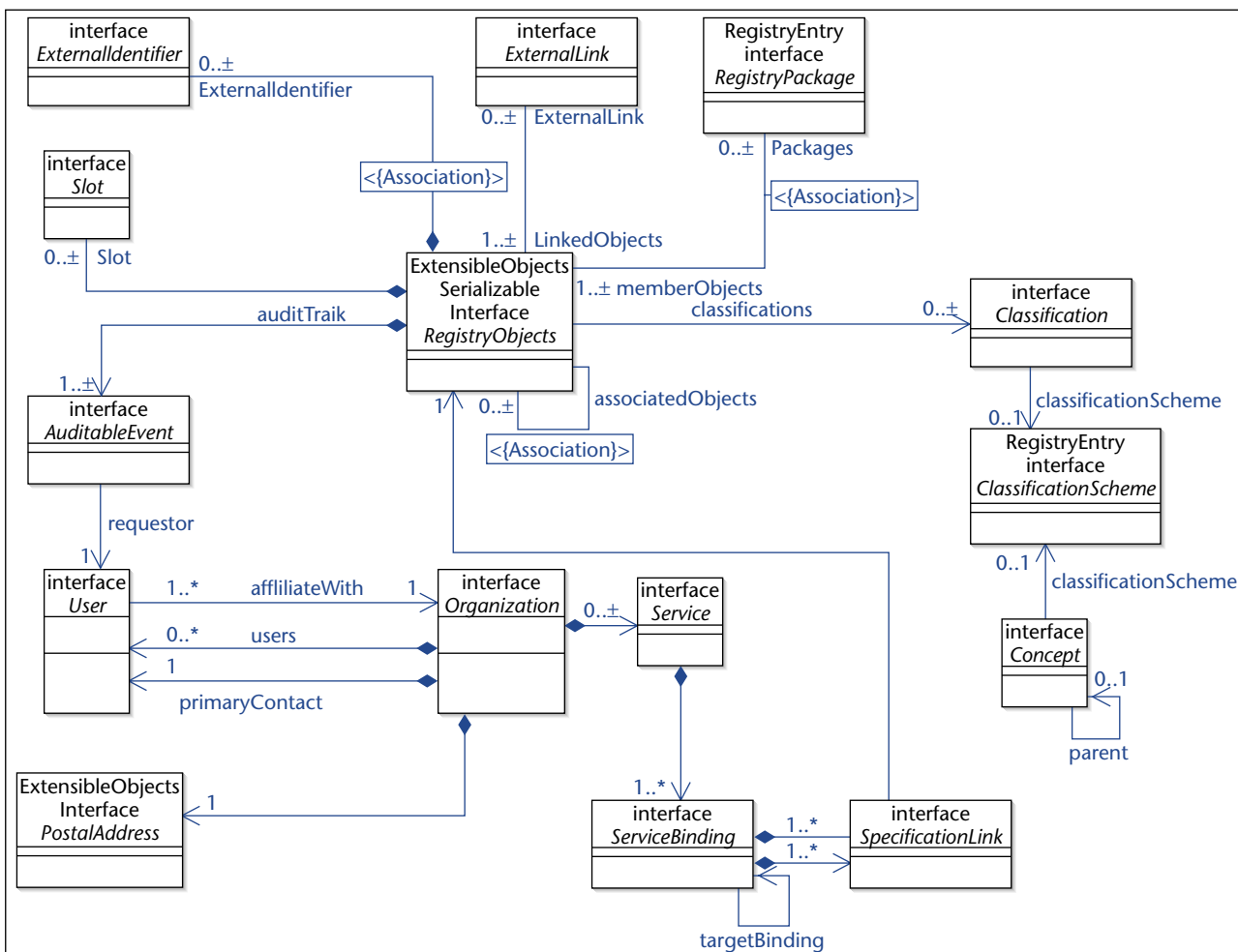
Adv.
Sitraka

waar je kaarten kunt kopen (en die kunt lezen). Volgens directory service specialist Novell moeten we niet allemaal terug naar school om dat te realiseren. Zij menen hun corporate ervaring met klassieke directory services als LDAP best toe te kunnen passen op de moderne webservice noden. Die stelling maken ze ook waar getuige een aantal recente acties. Novell heeft een voorstel ingediend bij het IETF dat een LDAP schema voor het opslaan van UDDI informatie aandragt, en staat mede aan de basis van de adoptie van DSML (directory services markup language). DSML maakt een bestaande directory service ondervraagbaar over SOAP, en wordt zo snel een werkbaar webservice enabled alternatief voor het gebruik van UDDI. Het gaat dan wel weer om directory services (en dus locatie onafhankelijkheid) dan om 'business registry's'.

JAVA OPLOSSINGEN Het laatste woord in deze zaak lijkt nog niet gezegd en de voorgelegde dualiteit zien we dan ook terugkeren in de bestaande Java API's. Enerzijds is er het sinds jaar en dag (ook heel sterk binnen EJBs bijvoorbeeld) gebruikte JNDI (Java Naming & Directory Interface) dat toelaat om het even welk object uit een directory op te halen op basis van een hiërarchisch

opgebouwde naam. Naast wat configuratie aspecten zit de essentie ervan in twee methodes: bind() en lookup() die werken op de zeer basale objecttypes String en Object als argument. Anderzijds is er de JAXR (Java API for XML Registry's) die via z'n API een heus infomodel aanreikt om te praten over die zaken die business registry's plegen te bevatten (zie figuur 1).

VOEDSTERVADER De immer politiek correcte benadering van Java ontkoppelt dit algemene zicht op de business registry dan weer volledig van de effectieve implementatie. Tussen de API en de effectieve registry die men wil consulteren nestelt zich een zogenaamde JAXR provider die de aanvragen op de API omzet naar effectieve bevragingen van het type business registry waarop men connecteert. In de officiële JAXR distributie worden er meteen twee providers meegeleverd. Eén provider is voor de ebXML Registry en Repository, omdat voedstervader Sun de echte wereldstandaard nog steeds een zeer warm hart toedraagt. De andere is UDDI provider omdat de rest van de wereld het blijft hebben over dit (toch slechts) voorstel van een industrieel consortium (waar Sun niet in mocht meespelen). Wereldschokkend wordt de JAXR volgens mij niet.



FIGUUR 1. JAXR fungeert ook als 'infomodel' over zaken die business registry's plegen te bevatten.

Ontkoppeling van 'versies' met XML

De grap in softwarekringen gaat dat God er slechts in geslaagd is de wereld in zes dagen te scheppen, omdat hij zich van geen enkele 'legacy' diende aan te trekken. Onafhankelijkheid van de eigen legacy wordt op het einde van de rit meestal enkel gerealiseerd door 'echte mannen' die het aandurven een volledig bestaand systeem te gaan vervangen door een nieuw. Exit legacy. En dapper leren leven met een overgangperiode waarin het nieuwe systeem stevast nieuwe onhebbelijkheden binnenbrengt. Niet in het minst omdat het vroeg of laat ook achteloos het label 'legacy' krijgt opgeplakt. Een ieder die zich nog eens opmaakt voor een volgende dergelijke veldslag, krijgt in de XML school alvast een nieuw strategisch inzicht aangeleerd. Een relatief jonge stroming daar bekwaamt zich namelijk in het schrijven van langlevende en toekomstgerichte schema's. In wezen gaat het om een tweede jeugd voor waarschijnlijk het designprincipe dat aan de grondslag van de eXtensible Markup Language lag: uitbreidbaarheid. In de hoogdagen van de *web browser wars* (u herinnert zich vast nog de jongensachtige flair waarmee Marc Andreessen het jasje van de Microsoft-bedreiging droeg) werd het serieus en de wetenschappelijke aanpak van het W3C ernstig op de proef gesteld. Het jonge standaardisatieorgaan werd met elke versie van die elektronische zeilplanken geconfronteerd met de relativiteit van hun eigen in steen gebakken standaarden. De verschillende HTML's volgden elkaar op in een moordend tempo en slaagden er nog niet in de expansiedrang van de browser leveranciers enigszins bij te benen. Voornamelijk onder die druk (W3C medewerkers wilden ook wel weer eens zorgeloos op verlof) werd gezocht naar een techniek om de ongebreidelde uitbreidingsdrang minstens binnen de grenzen van de standaard te laten verlopen. Ondertussen is niet alleen de eensgezindheid omtrent het gebruik van XML schrikbarend snel gegroeid, maar is helaas ook duidelijk geworden dat een correct uitbuiten van de X (extensibility) in het ontwerp van schema's niet van dezelfde vanzelfsprekendheid is. Technisch komt het erop neer dat men gepast gebruik maakt van enerzijds optionele elementen, en anderzijds het ANY keyword bij het ontwerp van een schema, en zijn toekomstige versies. Als het goed gebeurt is zullen de verschillende XML-instanties allen kruisgewijs ten opzichte van de verschillende versies van de DTD kunnen worden gevalideerd. Of het goed gebeurt, hangt dan niet zozeer van de kwaliteit van je glazen bol af, als wel van je visie en domeinkennis die je technisch in het schema uitdrukt in een vrijblijvende waas van 'dit doe ik nu nog niet'. In operationele XML messaging creëert deze 'flou artistique' alvast de mogelijkheid om versie-1 berichten door te sturen naar een versie-2 verwerkend systeem. Zo laat dit soort architecturen toe af te stappen van de draconische transitie-projecten waar alle applicatie systemen binnen je organisatie in één klap moeten vervangen worden. Het blijft tenslotte bang aanschrijven aan het bankloket, die maandag na het 'weekend van algemene systeem transitie'.

Tenzij je goudengids.nl heet en je jarenlang trouw opgebouwde "business registry" via een eigen JAXR Provider implementatie wilt gaan ontsluiten is er weinig kans dat de specifieke details van deze API je wereld danig gaan beïnvloeden. De meeste webservice georiënteerde Java IDE's zullen allicht een op JAXR gebaseerde repository browser meeleveren om vlot een bepaalde dienst te kunnen terugvinden. Dus voor de IDE bouwers wordt het ook een te lezen spec. Maar effectieve code die de

registry lookups ook dynamisch en at run time gaat doen, lijkt me de komende tijd echter meer uitzondering dan regel. Zelf geven de UDDI leveranciers aan dat zoiets binnen de bedrijfsgrenzen wel op een betrouwbare manier te realiseren is, alleen is die context natuurlijk weer deze waar de klassieke directory service oplossing ruimschoots voldoet, en vaak al aanwezig is. De tijden zijn er niet naar om nog wat extra licenties te gaan kopen.

JAXPACK Sinds het begin van deze artikelenreeks is het verre van stil geweest in XML land. Ook de Java community is aardig bezig geweest de belangrijkste bewegingen in nieuwe API's te vangen. Naast de JaxPack is er ondertussen een webservices pack dat naast de pure XML API's ook wat zinvolle implementaties en tutorials mee verpakt om meteen mee aan de slag te gaan. Maar ook voor de verdere toekomst zijn er nog heel wat werkgroepen onder de jcp vlag gestart aan specifieke XML gerelateerde API's. Een aantal daarvan gaan eerder richting het gebruik van XML in een specifieke functionele context zoals WEBDAV of WBEM. Voor de webservices wereld komt er ook een API om WSDL files te manipuleren, eveneens een voorbeeld van een API die enkel door de webservice IDE tool bouwers moet worden nagekeken. Tenslotte is ook nog fundamenteel parsing werk onderweg met de Pull Parsing API's. Dat zal naar verwachting onder de STaX (Streaming API for XML) vlag tegen het eind van het jaar worden voltooid. Er blijven dus voldoende alternatieven om tussen te kiezen. En omtrent de juiste keuze geldt er tot nader order slechts één wet:

"Would you tell me, please, which way I ought to go from here?"

"That depends a good deal on where you want to get to," said the Cat.

"I don't much care where —" said Alice.

"Then it doesn't much matter which way you go," said the Cat.

"— so long as I get somewhere," Alice added as an explanation.

"Oh, you're sure to do that," said the Cat, "if only you walk long enough."

(Lewis Carol, *Alice In Wonderland*)

Marc Portier (e-mail: mpo@outerthought.org)

Marc Portier is medeoprichter van Outerthought, een klein en spits-technologisch Java & XML competentiecentrum. Als Java-advocaat van het eerste uur volgt hij van nabij de nieuwe API's en richtingen (zoals J2EE) die de taal en het platform sinds de begindagen in 1996 heeft genomen. Daarbij gaat zijn interesse vooral uit naar de Internet en XML gerelateerde onderwerpen.