

Er is één hele goede reden om, mocht u dat nog niet gedaan hebben, eens achter Visual Studio .Net te gaan zitten. Visual Studio is gewoon een groot, modern, geavanceerd ontwikkeltool zoals er wel meer grote, moderne, geavanceerde ontwikkeltools zijn. De belangrijkste reden om eens met Visual Studio te spelen is: webservices. Pas als je ziet hoe akelig eenvoudig het is om een webservice aan de praat te krijgen en te gebruiken, zie je in wat webservices kunnen gaan betekenen voor de volgende generatie systemen. Een paar klikken, wat code, druk op de knop en er is een webservice geboren. WSDL, SOAP, user interface (om te testen), alles wordt ongemerkt voor je gegenereerd.

Komt de geest uit de fles?

Wat zijn webservices? Een webservice is in feite een op XML gebaseerde website die niet gemaakt is om door mensen gebruikt te worden, maar door computers. Een interactieve website zonder enige opmaak, zonder plaatjes, zonder reclame. Pure functionaliteit. Het is aan het systeem dat gebruik maakt van de webservice om eventueel een user interface aan te bieden boven op deze webservice. Een organisatie kan diensten aanbieden in de vorm van webservices en andere organisaties kunnen gebruik maken van die diensten als onderdeel van hun eigen applicaties. En dat alles geheel volgens heldere en open standaards.

Terug naar Java land: Sun. Iedereen die inderdaad al eens met webservices gewerkt heeft voelt zich op het moment erg ongemakkelijk met de stilte bij Sun. We zijn al zo lang gewend dat Sun en Microsoft elkaar continu de loef af proberen te steken, dat het haast gaat lijken alsof Sun het nu opgeeft. IBM probeert uit plaatsvervangende schaamte nog het een en ander goed te maken, Sun murmelt wat over hoe je (in theorie) Java webservices zou kunnen maken en er

zijn een paar kleinere partijen die in het gat proberen te springen. Webservices gaan over standaards. Een Java implementatie van webservices wordt pas echt serieus wanneer Sun met een standaard komt, een referentie-implementatie, een set interfaces of een echte implementatie.

Maar, er is hoop! Er zijn geruchten! Er wordt gefluisterd... Er wordt gefluisterd dat Sun zit te broeden op iets nieuws, op iets nog mooiers, mooier dan webservices zoals .Net het voorstelt. Beter en mooier kan niet, zeggen sommigen. Verbeteren mag tenslotte niet ten koste gaan van de standaards waar webservices op gebaseerd zijn. Er gaan over een op Jini gebaseerde webservice architectuur. Jini, al een paar jaar oud maar nog nooit echt van de grond gekomen.

Wat kan Jini betekenen voor webservices en wat kunnen webservices betekenen voor Jini? Jini is een Java architectuur voor ad hoc netwerken van devices. Devices in heel brede zin; telefoons, printers, applicaties, software componenten, ijskasten... In de Jini architectuur weten deze devices elkaar te vinden, bieden ze elkaar diensten

aan en gaan onderling contracten aan over het gebruik van deze diensten. Wanneer deze diensten aangeboden worden in de vorm van webservices ontstaat er een hele nieuwe reeks aan mogelijkheden. Stelt u zich eens voor, grenzen vervagen, een wereld van devices wordt werkelijk mogelijk. Een wereld van kleine slimme apparaatjes die met elkaar kunnen communiceren, gebruik makend van elkaars diensten in de vorm van webservices.

Jini kwam niet echt van de grond omdat er niet werd nagedacht over de standaardisatie van de communicatie tussen Jini devices. Met webservices krijgt Sun die standaardisatie eigenlijk in de schoot geworpen. Er moet alleen nog nagedacht worden over een nette integratie van webservices in de Jini architectuur. Dat de heren en dames bij Sun daar wat lang over doen is ze wel gegund. Als ze straks maar met iets héél moois naar buiten komen...

J. Meermans

is Java-kenner en te bereiken via
meermans@cibit.nl