Ros in de praktijk

Alten ontwikkelt software voor bladsnijrobot Priva

Priva bouwt aan een robot die automatisch tomatenbladeren kan wegknippen. Alten schreef in opdracht van VDL ETG Projects de besturingssoftware waarbij het zich baseerde op het opensource ontwikkelraamwerk Robot Operating System (Ros).

Berend Kupers

Het wordt steeds moeilijker om geschikt personeel te vinden voor werk in kassen. Daarom is Priva de uitdaging aangegaan om een deel daarvan te automatiseren. In samenwerking met een aantal telers heeft Priva de Kompano-bladsnijrobot ontwikkeld om het verwijderen van tomatenbladeren te automatiseren. Met het prototype wist Priva de Greentech Innovation Award 2016 in de wacht te slepen. Priva heeft de hulp van VDL ETG Projects ingeroepen om de robot door te ontwikkelen, te bouwen en productierijp te maken. Alten werd ingeschakeld om de besturingssoftware te schrijven.

De Kompano-robot rijdt op bestaande verwarmingsbuizen tussen de rijen tomatenplanten door. Hij gebruikt twee stereocamera's om beelden te maken van het gewas. Op basis van deze beelden bepaalt een door Priva ontwikkeld visionalgoritme de positie en oriëntatie van de bladstelen. Met een uitschuifbare arm worden de bladstelen vervolgens vastgepakt en afgeknipt. Om ziekteverspreiding te voorkomen, wordt het mes regelmatig gedesinfecteerd.

Dit complexe gedrag en de integratie van alle verschillende onderdelen zijn uitdagingen voor de softwareontwikkeling. Daarnaast zijn de uitbreidbaarheid en modulariteit van de software van belang, want Priva heeft aangekondigd dat de Kompano-robot slechts de eerste is in een lijn van agrorobots.

Met de stap van prototype naar autonoom dagelijks gebruik worden er hoge eisen gesteld aan de software. De robot moet intelligent genoeg zijn om te kunnen reageren op uitzonderingssituaties, hij moet bestand zijn tegen het vuil en de onvoorspelbaarheid van de omgeving en hij moet natuurlijk veilig zijn in gebruik.

Voor de ontwikkeling van de software is de keuze gevallen op het opensource framework Ros. Dit framework biedt de gevraagde modulariteit en een goede ontwikkelmethodiek maakt de software betrouwbaar genoeg voor toepassing in de veeleisende omgeving van een kas.

Tools

Ros is een open, platformonafhankelijk raamwerk voor de ontwikkeling van robotsoftware. Sinds de start in 2007 heeft het sterk aan populariteit gewonnen en inmiddels wordt het wereldwijd op duizenden robots gebruikt. Het idee achter Ros is simpel: bied software-engineers een framework voor de ontwikkeling van robotapplicaties zodat ze kunnen voortbouwen op het werk van hun collega's en ze ideeën en algoritmes kunnen uitwisselen.

Het framework is breed opgezet en omvat veel aspecten die met de ontwikkeling van robotsoftware te maken hebben. Ceo Brian Gerkey van de Open Source Robotics Foundation, die Ros onderhoudt, legt Ros uit als de combinatie van infrastructuur, functionaliteiten, tools en ecosysteem. De infrastructuur van Ros bestaat uit verschillende modules of nodes, die elk een specifieke functionaliteit implementeren, en gestandaardiseerde communicatie tussen deze nodes, die loopt via messages of services. De standaard functionaliteiten worden meegeleverd met Ros via softwarebibliotheken en pakketten, Ros-packages geheten. Voorbeelden van deze standaard functionaliteiten zijn: navigatie, kinematica, vision en *collision checking*. Daarnaast zijn er veel Ros-packages beschikbaar gesteld vanuit fabrikanten en de opensourcegemeenschap. Denk hierbij aan drivers voor robotcontrollers en sensoren of een state-of-the-art algoritme voor objectherkenning.

Naast deze functionaliteiten biedt Ros tools om de ontwikkeling en het gebruik van software te vereenvoudigen. Voorbeelden hiervan zijn: simulatie, visualisatie, geautomatiseerde tests en loggingtools. Het ecosysteem is een van de belangrijkste redenen voor de populariteit. Er bestaat een grote online community van ontwikkelaars die nieuwe functionaliteiten beschikbaar stellen, die de software op verschillende platforms testen en verbeteringen aandragen.

Gazebo

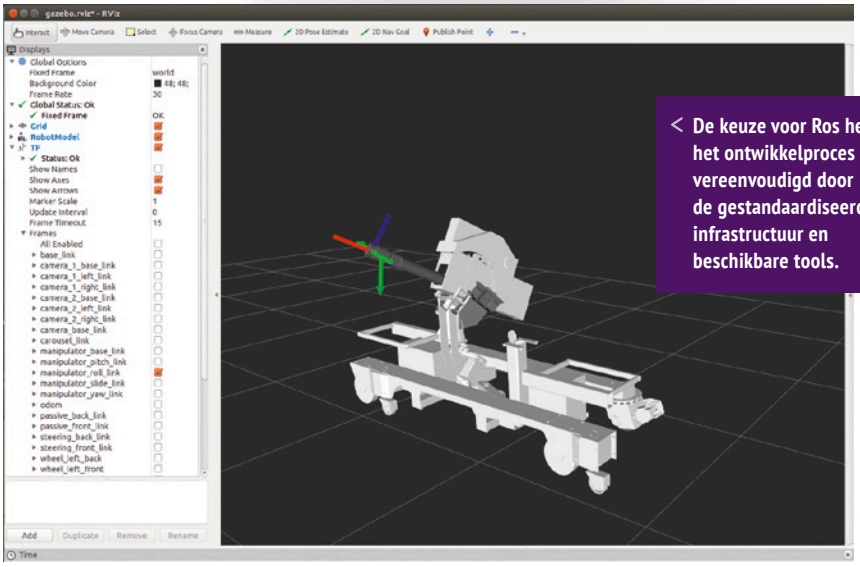
Waarom hebben we in dit geval voor Ros gekozen? De modulaire opbouw van software via Ros-nodes sluit goed aan bij de wens van Priva om de software uitbreidbaar en multi-inzetbaar te maken. Als een volgend model in de serie een andere snijkop krijgt om andere gewassen te kunnen snoeien, kunnen het model en de aansturing van de grijper worden uitgewisseld en kunnen de overige nodes ongemoeid blijven. Als er in een later stadium wordt besloten om over te stappen op een andere motorleverancier, hoeft enkel de driver te worden vervangen.

De gestandaardiseerde communicatie tussen nodes zorgt er bovendien voor dat alle delen van dit complexe systeem eenvoudig kunnen worden geïntegreerd. De programmeur kan nodes onafhankelijk van elkaar ontwikkelen en testen. Door van tevoren de communicatie tussen de vision en de rest van het systeem af te

stemmen, konden Priva en Alten deze delen onafhankelijk van elkaar ontwikkelen.

De ontwikkeling van de software wordt verder vereenvoudigd door de tools binnen Ros. De hardware kan in detail worden gesimuleerd met de Gazebo-simulator voordat de echte hardware nodig is. De visualisator kan de locatie van gedetecteerde bladstelen en het berekende pad van de robotarm tonen, zodat de programmeur kan zien wat er gebeurt in het 'brein' van de robot.

> Met het prototype van zijn bladsnijrobot won Priva de Greentech Innovation Award 2016.



< De keuze voor Ros heeft het ontwikkelproces vereenvoudigd door de gestandaardiseerde infrastructuur en beschikbare tools.

Architectuur

Omdat Ros een opensource platform is met roots in de academische wereld wordt vaak gedacht dat het niet betrouwbaar genoeg is voor industriële toepassingen. Door gebruik te maken van de ingebouwde tools en best practices vanuit de softwarewereld heeft het team ervoor gezorgd dat de Kompano-robot voldoet aan alle eisen.

De eerste stap was niet anders dan voor elk ander softwareproject: een systematische ontwikkelmethode volgen. Priva, VDL ETG Projects en Alten hebben in een vroeg stadium de functies en requirements afgestemd. Daarna is er aan de hand van deze requirements een overzichtelijke architectuur opgezet.

Door de modulaire opzet van Ros lieten de functies zich eenvoudig vertalen naar nodes. Er zijn nodes gemaakt voor bijvoorbeeld de navigatie, de motionplanning van de arm, motordrivers en de vision. Door ze te groeperen in Ros-packages is de software overzichtelijk geworden en daarmee beter onderhoudbaar. Doxygen heeft de documentatie van de software automatisch gegenereerd.

Als de requirements in de toekomst veranderen of er moet onverhoopt een functie worden toegevoegd, dan hoeft dit slechts op nodeniveau te worden aangepast. De wijziging heeft daarmee weinig effect op de rest van het systeem.

Betrouwbaarheid

Alten heeft de benodigde ontwikkeltijd kunnen inkorten door gebruik te maken van opensource packages. Om de betrouwbaarheid van het systeem te garanderen, zijn we hier selectief mee omgegaan. De standaard Ros-packages worden door de Open Source Robotics Foundation getest met behulp van een buildfarm gebaseerd op het *continuous integration framework* Jenkins. Deze packages worden pas vrijgegeven wanneer alle tests slagen. Wij konden deze zonder problemen gebruiken. Doordat de basis van Ros, zowel als de communicatie, op een reeks verschillende platforms over de hele wereld wordt gebruikt, komen eventuele problemen snel aan het licht en is het inmiddels een zeer betrouwbaar platform.

Ontwikkelaars van nieuwe Ros-packages kunnen dezelfde buildfarm gebruiken om hun packages te testen voordat ze deze beschikbaar stellen. De site van Ros geeft per package aan of er unittests zijn geïmplementeerd en of de code is gedocumenteerd. Tijdens de ontwikkeling heeft Alten deze indicatoren gebruikt bij de beslissing om een opensource package te gebruiken of om de code zelf te schrijven.

Alleen wanneer er geen geschikte package beschikbaar was of wanneer de code moest worden geoptimaliseerd voor een specifieke toepassing, heeft de programmeur zijn eigen nodes geschreven. Om de betrouwbaarheid van deze nodes vast te kunnen stellen, zijn unittests en integratietests gebruikt. Ros biedt frameworks voor unittests in C++ en Python en heeft een eigen framework voor integratietests.

Hoewel via deze methodes de software betrouwbaar genoeg is gemaakt voor industriële toepassingen, voldoet hij niet aan de strengere eisen voor een veiligheidssysteem. Net zomin als dat de normale industriële pc waar Ros op draait, voldoet aan Europese normen voor veiligheid. Een goed veiligheidssysteem wordt ontworpen door de richtlijnen te volgen en is in dit specifieke geval geïmplementeerd via gecertificeerde veiligheidscomponenten.

In simulatie

Door de bovengenoemde methodes te volgen hebben we Ros gebruikt om een systeem te ontwikkelen dat even betrouwbaar is als elk ander systeem. Daarnaast heeft de keuze voor Ros het ontwikkelproces vereenvoudigd door de gestandaardiseerde infrastructuur en beschikbare tools. Door selectief gebruik te maken van opensource packages is de ontwikkeltijd bovendien flink verkort.

De software voor de Priva-robot is inmiddels uitvoerig getest in simulatie, maar de echte test komt als de eerste Kompano-bladsnijrobots klaar zijn om getest te worden in de kassen.

Berend Kupers is consultant bij Alten Technology.

Redactie Alexander Pil