

De ANW-artikeleditor: software als strategie

Jan Niestadt (Instituut voor Nederlandse Lexicologie)

Inleiding

Lexicologen hebben software als gereedschap de laatste jaren echt ontdekt. Ieder jaar komen er innovatieve applicaties bij die hun werk op de een of andere manier vergemakkelijken.

De efficiëntie waarmee gewerkt kan worden, neemt hierdoor snel toe. Dit stelt de initiatiefnemers van nieuwe projecten in staat steeds ambitieuzer doelen te stellen, wat vervolgens weer zwaardere eisen aan de software stelt.

Het is een proces met positieve feedback, net als een microfoon die gaat piepen wanneer hij te dicht bij de luidspreker staat. Deze piep van de lexicologie wordt steeds oorverdovender en voorlopig lijkt nog niemand geneigd de plug uit de microfoon te trekken. Niet alleen is het 'definitieve' dictionary writing system nog niet gebouwd, het zal er waarschijnlijk nooit komen. Er blijft altijd wat nieuws te wensen.

Automatiseerders kunnen op twee manieren aan de vragen van lexicologen voldoen: door software aan te kopen of door zelf software te ontwikkelen. In dit artikel presenteer ik een 'casestudy' van de tweede optie: de ontwikkeling binnen het INL van de artikeleditor voor het *Algemeen Nederlands Woordenboek* (ANW). Dit is een applicatie waarin medewerkers van het ANW de woordenboekartikelen bewerken. De artikeleditor vormt op zichzelf nog geen compleet dictionary writing system, maar vervult wel een aantal belangrijke functies die je in zo'n systeem verwacht.

Ik zal de artikeleditor beschrijven, vergelijkingen maken met andere beschikbare software en de vraag proberen te beantwoorden of het zinvol is voor een wetenschappelijk instituut om zelf software te ontwikkelen.

Het oude systeem

De ANW-artikeleditor is ontwikkeld ter vervanging van ons vorige systeem, dat gebruik maakte van de XML-editor Altova XmlSpy met een binnen het INL ontwikkelde formulierweergave. Peter van der Kamp beschreef in 2007 in zijn artikel 'De ontwikkeling van het ANW-artikelformulier: grenzen aan de groei?' reeds enkele beperkingen van deze aanpak.

Een van de problemen waar we tegenaan liepen was dat niet alle gewenste functionaliteit binnen het Altova-systeem eenvoudig bewerkstelligd kon worden. Met name communicatie met externe databases en andere applicaties vormde een probleem. Verder wilde men in één oogopslag de artikelstructuur kunnen zien, sneller door het artikel kunnen navigeren, en bijvoorbeeld automatisch een lijst met suggesties krijgen als men enkele letters in een veld tikte.

Er kwamen regelmatig nieuwe versies uit van Altova XmlSpy, waarvoor opnieuw moest worden betaald. Meer dan eens bleken de nieuwe versies trager te draaien en incompatibel te zijn met bepaalde aspecten van het bewerkingsformulier. Achterblijven in versie was geen realistische optie, omdat we vroeger of later toch een keer mee zouden moeten. Zo zaten we opgesloten in de 'vernieuwingstredmolen' van Altova.

Over deze wensen en problemen is meerdere malen contact gezocht met de Altova-ontwikkelaars, maar die konden uiteindelijk niet ingaan op onze specifieke verzoeken, omdat wij slechts een van hun vele klanten waren.

Kortom, deze aanpak had stevige nadelen en die vertraagden de bewerking van de artikelen en zorgden voor invoerfouten. Er was dus behoefte aan een gebruiksvriendelijker interface die beter aansloot bij de werkzaamheden van het ANW.

Deze bezwaren en beperkingen hebben ons doen concluderen dat het aanpassen van een ‘off-the-shelf’-gereedschap als Altova XmlSpy interessant kan zijn om een prototype te bouwen, maar voor de ontwikkeling van een complex gereedschap op lange termijn vaak tekort schiet.

Alternatieve systemen

Uiteraard is ook overwogen om een van de bestaande dictionary writing systems (bijvoorbeeld TschwaneLex of iLex) in te zetten. Zoals gezegd zijn dit complete systemen met vaak een hoop nuttige extra functionaliteit die noch ons oude systeem, noch de nieuwe artikeleditor waarover dit artikel gaat, op dit moment kunnen bieden. Toch bleken deze systemen onvoldoende aan te sluiten bij de wensen van het ANW.

Het ANW wilde bijvoorbeeld wordeigenschappen uit de ‘kop’ van een artikel (d.w.z. het algemene gedeelte van het artikel, vóór de betekenissen) automatisch laten overerven naar de verschillende betekenissen. Dit scheelt veel invoerwerk, doordat de waardes van bepaalde velden bij meerdere betekenissen vaak hetzelfde zijn. De beschikbare systemen hebben hier (nog) geen ondersteuning voor.

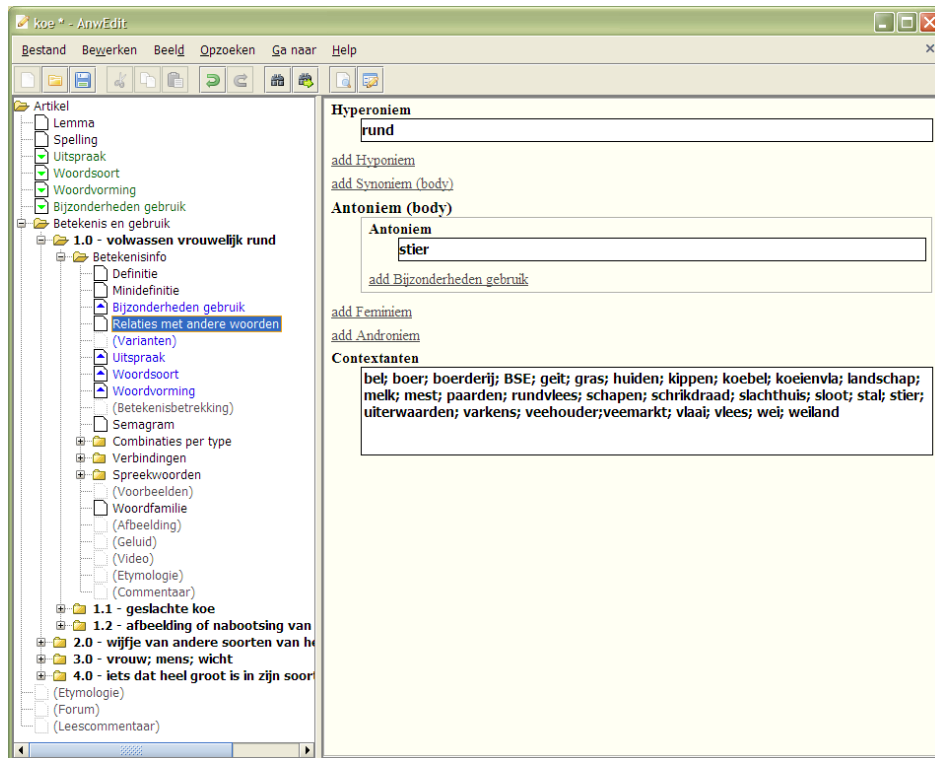
Ook gaven ze vaak een onvoldoende duidelijk overzicht van de (ingewikkelde, diep geneste) artikelstructuur. De meeste dictionary writing systems zijn ontworpen voor woordenboeken met een eenvoudiger artikelstructuur dan het ANW.

Een algemene indruk was dat de beschikbare producten niet altijd voldoende gebruiksvriendelijk zijn. De bediening wijkt vaak af van die van andere bekende applicaties, en er zijn relatief veel handelingen nodig om veelvoorkomende taken uit te voeren. Gebruiksvriendelijkheid was een zeer hoge prioriteit voor ons, omdat medewerkers sommige handelingen honderden keren per dag uitvoeren, en elke muisklik minder dus zorgt voor een prettiger gebruikservaring en hogere productiviteit.

Tot slot kunnen ANW-specifieke functies vaak niet eenvoudig in een bestaand product worden ingebouwd, en als het kan, is het duur en duurt het vaak lang voor de ontwikkelaar er tijd voor heeft. De communicatie met een externe ontwikkelaar die ons project niet kent, zagen wij ook als problematisch.

Onderscheidende functionaliteit

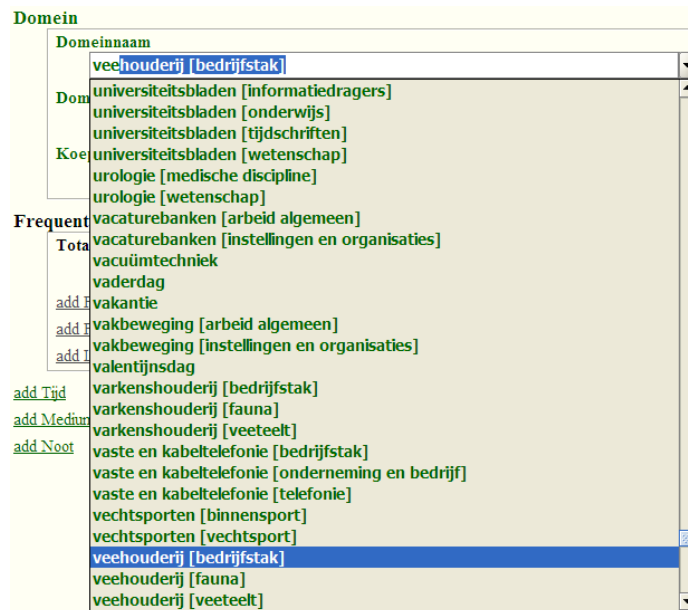
In het bestaande aanbod misten we dus het een en ander. Wat biedt de ANW-artikeleditor dan voor bijzondere functionaliteit, en op welke manier is de editor specifiek op het ANW toegespitst? Hieronder volgt een overzicht van de belangrijkste onderscheidende functies.



Figuur 1. De gebruikersinterface van de artikeleditor

Eén prettig aspect aan het ontwerp van de artikeleditor is dat de gebruikersinterface niet stap voor stap door een programmeur opgebouwd hoefde te worden; het wordt automatisch gegenereerd uit ons XML Schema (dat is een formele beschrijving van de precieze structuur van een ANW-artikel). Dit blijkt een zeer flexibele aanpak. Wijzigingen aan het XML Schema zorgen bijvoorbeeld zelden voor werk aan de editor; de wijziging is gewoon direct in de editor zichtbaar. Ook levert het een rustige, consequente interface op waar gebruikers snel aan wennen.

De automatisch gegenereerde interface is, op punten waar dat noodzakelijk is, aan te passen aan specifieke wensen. Zo is in plaats van een selectielijst voor het veld 'Domeinnaam' een tekstveld met automatische suggesties gebruikt. Verder worden bij de gekozen domeinnaam automatisch in een database de bijbehorende domeinklasse en het bijbehorende koepeldomein gezocht. Andere mogelijkheden voor afwijkende interface-elementen zouden kunnen zijn: een datumkiezer, een 'slider' voor getallen, keuzerondjes in plaats van een selectielijst als er maar een klein aantal mogelijke waarden zijn, etc.



Figuur 2. Automatische suggestie bij het veld Domeinnaam

Ook al hebben we goede ervaringen met het gebruik van het XML Schema voor validatie en voor het genereren van het gebruikersinterface, toch kan daarin niet alles worden beschreven wat je zou willen. Vaak wil je bijvoorbeeld aangeven dat een bepaald veld alleen mag worden toegevoegd als de geselecteerde woordsoort ‘substantief’ is; dit soort beperkingen zijn in het XML Schema niet uit te drukken. De editor maakt het mogelijk om deze additionele condities met behulp van Java-code in te bouwen. Dit zorgt voor nog meer eenvormigheid in de artikelen dan het XML Schema alleen kan garanderen, en dat is weer gunstig voor de automatische verwerking en doorzoekbaarheid van het woordenboek.



Figuur 3. De artikelstructuur in ‘boomvorm’

De ‘boomstructuur’ aan de linkerkant van het artikel maakt het mogelijk om in één oogopslag de structuur van het artikel te doorzien en er snel doorheen te navigeren. Bij betekenissen, spreekwoorden, e.d. worden ook het begin van de tekst getoond, zodat direct duidelijk is welk

element een label in de boom representeert. Door kleuring en gebruik van pictogrammen is verder snel te zien welke gedeeltes van het artikel zijn ingevuld en welke nog niet.

Een belangrijke aanleiding voor het ontwikkelen van de artikeleditor was zoals gezegd de wens om bepaalde gedeeltes van het artikel selectief te laten overerven van de ‘kop’ van het artikel naar de betekenissen. Tijdens de ontwikkeling is een hoop tijd gaan zitten in het goed en logisch laten werken van deze functionaliteit. Het doel was om niet alleen direct duidelijk te maken welke waarde op een betekenis van toepassing is, maar ook welke van de aanwezige waardes zijn overgeërfd en welke waardes juist afwijken van de over te erven waarde. Dit is gerealiseerd met doeltreffend kleurgebruik en verklarende pictogrammen.

Een groot voordeel van de volledige controle die we over onze artikeleditor hebben, is dat we eenvoudig mogelijkheden kunnen inbouwen voor informatieuitwisseling met andere applicaties. Het ANW-corpus wordt bijvoorbeeld geraadpleegd via de Sketch Engine van Lexical Computing Ltd. Een redacteur zoekt relevante citaten bij een woord en kopieert deze in de Sketch Engine naar het Klembord. Omdat de artikeleditor citaten uit de Sketch Engine herkent, kan het citaat met twee kliks worden ‘geplakt’ in het ANW-artikel. Uitgaande van de zogenaamde ‘checkbox lexicography’ in de Sketch Engine kan de lexicograaf zelfs met één handeling een aantal combinaties met voorbeelden in de editor plakken. In de toekomst kunnen nog talloze andere mogelijkheden tot informatieuitwisseling met andere applicaties ingebouwd worden. Het inbouwen van dit soort functies in de artikeleditor is relatief weinig werk, maar bespaart ANW-medewerkers een hoop tijd.

De artikeleditor maakt het ook mogelijk voor iedere medewerker om een artikel weer te geven op de manier die hij/zij het prettigst vindt. Artikelen kunnen worden bewerkt in een ‘hele artikel’-modus, waarbij een redacteur van begin tot eind door het artikel kan scrollen. Overigens kan nog steeds snel door dit lange artikel gesprongen worden door in de boomstructuur te klikken. De andere bewerkmodus toont het artikel steeds per onderdeel, bijvoorbeeld ‘Spelling’ of ‘Definitie van betekenis 1’; soms is dit overzichtelijker dan het hele artikel tegelijk te zien. Tot slot kan het artikel nog worden bekeken in de afdrukweergave of als XML-code.

Een artikeleditor moet uiteraard met een groep medewerkers tegelijk gebruikt kunnen worden. Hierbij moeten conflicten voortkomend uit het tegelijk bewerken van artikelen worden vermeden, moet de voortgang van artikelen in de gaten wordt gehouden, en moet worden geadmistreerd wie aan welke onderdelen van welke artikelen gewerkt hebben. Onze artikeleditor vermijdt bewerkingsconflicten door een artikel te vergrendelen zodra een medewerker het opent. Documentbeheer is beschikbaar in de vorm van metadata over auteurs en voortgang bij elk artikel.

Tot slot bevat de artikeleditor nog een aantal specifieke ‘handigheidjes’ die het bewerken aangenaam en efficiënt maken, zoals het verwijderen van dubbele spaties uit citaten, het vanuit ieder onderdeel van een artikel snel de definitie kunnen opvragen (om bijvoorbeeld een gedeelte daarvan te kopiëren) en links naar diverse onlineapplicaties (het iWNT, het ANW-corpus, Google, Wikipedia, etc.) voor het opzoeken van informatie.

Het bovenstaande bestaat uit allemaal voorbeelden van hoe de artikeleditor ANW-medewerkers assisteert bij het uitvoeren van hun dagelijkse taken. Doordat we de artikeleditor zelf ontwikkelen, zijn we in staat in zeer kort tijdsbestek mogelijkheden in te bouwen die dan direct tijdsbesparingen opleveren bij bewerking. Hadden we gekozen voor een oplossing die niet volledig onder onze controle stond, dan hadden we veel van deze voordelen waarschijnlijk moeten missen.

Ontwikkeltraject

De ANW-artikeleditor is geschreven in de programmeertaal Java, met gebruikmaking van een aantal (open source) hulpcomponenten. De hele editor is door de auteur van dit artikel in ongeveer zeven maanden ontwikkeld.

Tijdens de gehele ontwikkeling was er zeer regelmatig overleg met de opdrachtgever (Fons Moerdijk, hoofdredacteur van het ANW). Tegen het einde van het traject is er enkele maanden getest door een klein groepje ANW-medewerkers. Deze twee vormen van feedback (nauwe samenwerking met de opdrachtgever en vrij lang, intensief testtraject) hebben de uiteindelijke gebruikersinterface sterk verbeterd.

De artikeleditor is door de medewerkers van het ANW goed ontvangen. Na een korte gewenningsperiode werkt men er nu zonder noemenswaardige problemen mee. Diverse medewerkers hebben aangegeven erg tevreden te zijn met de nieuwe artikeleditor.

Wensen

De artikeleditor is zo opgezet dat deze ook voor toekomstige projecten bruikbaar zal zijn. Er zal echter nog wel wat werk verzet moeten worden voor hij een breed inzetbaar product vormt.

Een belangrijke functie die nog moet worden ingebouwd is het bijhouden van een kruisverwijzing naar een betekenis binnen een ander artikel, die intact blijft als bijvoorbeeld de volgorde van betekenissen in het doelartikel gewijzigd wordt.

Ook bestaat er behoefte aan uitgebreide zoekmogelijkheden over het gehele woordenboek. Voorbeelden van vragen die snel te beantwoorden zouden moeten zijn: ‘Hoeveel spreekwoorden bevat het woordenboek?’, ‘Welke waardes zijn er allemaal ingevuld in het “Kleur”-semagramveld?’.

Soms is het noodzakelijk om in velden eenvoudige opmaak op te nemen, zoals vette en cursieve tekst. Ook aan subscript en superscript is behoefte, voor het correct weergeven van begrippen als CO₂ en x². Deze functionaliteit zal op termijn aan de artikeleditor worden toegevoegd.

Nog lang niet alle mogelijkheden van XML Schema worden ondersteund door de artikeleditor. We hebben ons tot nu toe beperkt tot die zaken die nodig waren voor het ANW. In de toekomst zal bredere ondersteuning hiervoor worden geïmplementeerd.

Tot slot is het belangrijk dat alle zaken die specifiek zijn voor het ANW worden geïsoleerd, zodat die eenvoudig te vervangen zijn door de specifieke vereisten van een ander project. Tot op zekere hoogte is dit al gebeurd tijdens de ontwikkeling.

Als deze aanpassingen worden doorgevoerd in de editor, is het resultaat een robuust, wendbaar platform voor toekomstige projecten.

Toekomst van de artikeleditor

Wat gaat er gebeuren met de artikeleditor, behalve gebruik door het ANW? Er zijn een aantal mogelijkheden. Natuurlijk zal de artikeleditor indien mogelijk in de toekomst ingezet worden voor nieuwe projecten van het INL. Het zou ook kunnen dat er interesse bestaat vanuit andere instituten voor het gebruik en de verdere ontwikkeling van deze applicatie.

Binnen het INL bestaat enthousiasme voor het idee om de artikeleditor samen met andere instituten verder te ontwikkelen, bijvoorbeeld door deze beschikbaar te stellen als open source. Het is interessant om te onderzoeken wat we kunnen bereiken als er andere partijen meehelpen aan de verdere ontwikkeling van de artikeleditor, en andere gereedschappen die er op aansluiten. Als een paar enthousiaste ontwikkelaars er samen de schouders onder kunnen zetten, lijkt een volwaardig en gebruiksvriendelijk open source dictionary writing system¹ een

¹ Er bestaat een open source dictionary writing system, Mātāpuna. Het is in Nieuw-Zeeland ontwikkeld voor een woordenboek van de taal van de Maori. Het is echter vrij bescheiden in opzet (ondersteunt geen XML Schema; geen diepe hiërarchische structuren mogelijk) en het project lijkt niet erg actief meer (de laatste wijziging was in

bereikbaar ideaal. Zo'n systeem zou een mooie basis vormen voor toekomstige lexicografische aspiraties.

Conclusies

Het INL heeft met het ontwikkelen van de ANW-artikeleditor tijd en geld geïnvesteerd. Dit is een goede investering gebleken: het intern ontwikkelen van de artikeleditor heeft ons voordelen opgeleverd die we anders nooit hadden gehad. Dankzij de artikeleditor kan het ANW nu efficiënter werken, verdere verbeteringen kunnen we snel zelf implementeren wanneer we dat willen, en we zitten niet langer vast aan de kostbare updates van een externe partij.

In het algemeen denk ik dat softwareontwikkeling noodzakelijkerwijs een van de taken van een taalkundig instituut is. Vernieuwend wetenschappelijk onderzoek steunt steeds vaker op nieuwe software met nieuwe mogelijkheden. Wanneer we ons beperken tot kant-en-klare software die we zelf niet kunnen wijzigen, laten we de ontwikkelaar van deze software onze technische mogelijkheden bepalen. Daarmee geven we het initiatief uit handen en kiezen we dus voor een volgende rol in plaats van een leidende. Als we onze eigen gereedschappen ontwikkelen, bepalen wij zélf hoe ver we onze grenzen kunnen verleggen. Het is dus een fundamenteel strategisch voordeel.

Hoe is dat te rijmen met de principes van open source? Een strategisch voordeel geef je toch niet zomaar weg?

Het antwoord is eenvoudig: natuurlijk doe je dat wel, als je verwacht dat je daardoor meer wint dan je verliest. Volgens mij hebben we niets te verliezen, en alles te winnen. Laten we samen een open platform creëren om daarvandaan onze visies voor vernieuwing te lanceren!

Bibliografie

Kamp, Peter van der (2007), 'De ontwikkeling van het ANW-artikelformulier: grenzen aan de groei?', in: Fons Moerdijk, Ariane van Santen en Rob Tempelaars (red.), *Leven met Woorden*. Leiden, Instituut voor Nederlandse Lexicologie / Koninklijke Brill Leiden., 151-160.

Altova XmlSpy, <www.altova.com/XMLSpy>

TshwaneLex, <tshwanedje.com/tshwanelex>

iLex, <www.emp.dk/ilexweb/>

Sketch Engine, <www.sketchengine.co.uk/>

Mātāpuna, <matapuna.thinktank.co.nz/matapuna/>