

Takties Agenda Component

Dagmar Hofman

October 5, 2018

Contents

1	inleiding	3
2	over het component	5
2.1	Het Model-View-Controller design pattern	6
2.2	De logische opbouw van het component	7
3	gegevensanalyse en ontwerp	8
4	functioneel ontwerp	9
4.1	Klassenhierarchie	10
4.2	Klassendiagram	12
4.3	Dialogstructuur	12
5	technisch ontwerp	13
6	implementatie en testen	14
7	status van het project	14

1 inleiding

Web bureau Takties heeft mij gevraagd het takties agenda component te maken voor Joomla.

Het component werkte al in de site voor Gezondheidscentrum Kloosterveen. Echter, bevatte het nog het een en ander aan bugs.

Het eerste wat ik voor Takties heb gedaan is de ontwikkelomgeving ingericht. Dit is een Debian Linux distributie.

Het hele project heb ik met behulp van het **git** versiebeheersysteem ingericht. Ik heb overlegd of het niet beter is alles op een server te hosten die van Takties zelf is. Vooralsnog is de huidige oplossing voldoende.

Omdat ik nieuw ben op het gebied van het schrijven van Joomla componenten heb ik eerst wat onderzoek gedaan. Dit was naar het schrijven van Joomla modules en (nog) niet van Joomla componenten.

Vervolgens hebben we besloten het hele ding te reconstrueren. Dit gebeurde omdat het component misschien voor andere sites beschikbaar moet worden gemaakt. En daarnaast om zo volledig inzicht te krijgen hoe een Joomla component werkt en hoe er een te bouwen.

Takties heeft inmiddels een eigen git server op Bitbucket. Er is voor Bitbucket gekozen omdat de projecten die er mee bijgehouden worden alleen zichtbaar zijn voor een beperkt publiek. In dit geval de medewerkers van Takties.

Het product staat onder de GNU licentie versie 2. Dit wil zeggen dat als het product verkocht wordt aan een klant, dat ook de broncode er bij geleverd moet gaan worden. Dit behoeft verder geen aparte taak, omdat de taal die gebruikt wordt PHP is. Het lijkt me gepast om een eventuele klant de zipfile aan te bieden.

De zipfile is inmiddels gereed. De bugs zijn uit het programma, en de eerste release is beschikbaar. Er zijn nog wel wat op- of aanmerkingen over het produkt, het updaten van afspraken in de agenda gaat nog niet zo heel goed en misschien laat de user interface wat te wensen over. De update routine, daar wordt aan gewerkt. Wat de user interface betreft wil deze aan Takties voorleggen, om te kijken of het ook wat mooier kan.

Dit document neemt verder het volgende in beschouwing:

- Een korte beschrijving van het component.
- Gegevensanalyse en ontwerp
 - Het entiteiten relatie diagram.
- Functioneel ontwerp
 - Het klassendiagram, die schetst uit welke objecten het programma bestaan.
 - De dialoogstructuur.
- Technisch ontwerp
- Implementatie en testen
 - De testcases

Wat het laatste bevat (de testcases) zou ik ze in ieder geval nog eens grondig overwegen.

Merk op dat ik bij het schrijven van dit document mijn eigen techniek gebruik, en dat het misschien nuttig is bij verdere projecten te overleggen hoe een project gedocumenteerd moet gaan worden.

2 over het component

Het component zelf heet `com_tks_agenda`. De prefix **com_** is daar om te duiden dat het om een component gaat en de prefix **tks_** is gekozen voor alle eventuele componenten die door Takties geschreven gaan worden. Verder verlangt de goede werking van het component de modules:

- `mod_tks_agenda`
- `mod_tks_downloads`
- `mod_tks_news`

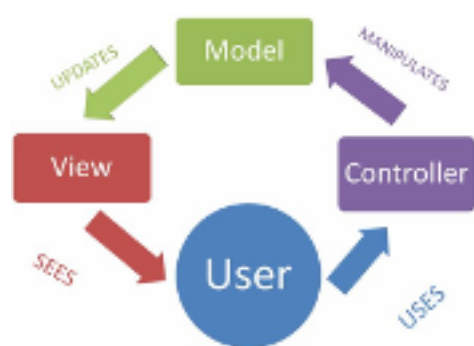
Ook is er een component nodig van een andere auteur. Het **profile5** component. Deze heb ik in de huishouding meegenomen en opgeslagen, maar er verder niets aan hoeven doen.

Het component zelf is globaal onder te verdelen in drie functionaliteiten.

- De medewerker nieuws sectie.
Deze houdt in blog formaat bij welk nieuws er geadministreerd en getoond moet worden.
- De downloads sectie.
Deze houdt bestanden bij die nuttig zijn. (PDF of ODT formaat)
- De agenda sectie.
Dit is de agenda zelf. Het houdt bij welke vergaderruimtes er gereserveerd moeten worden. Ook toont het de agenda zelf.

2.1 Het Model-View-Controller design pattern

Joomla componenten maken gebruik van het MVC (Model-View-Controller) design pattern. Dit klinkt nogal complex maar is in feite erg eenvoudig.



De models, views en controllers in het component zijn de volgende:

Voor de **agenda** sectie:

- items
- itemform
- item

Voor de **download** sectie:

- newsitems
- newsitemform
- newsitem

Voor de **medewerker nieuws** sectie:

- downloads
- downloadform
- download

Daarnaast is er een **portal** view toegevoegd.

2.2 De logische opbouw van het component

Het Joomla component bevat een **site** kan en een **administratie** kant.

De opbouw van het component bevat een **xml** file die de metadata van het component bijhoud.

Dat wil zeggen:

- De gegevens voor de auteur, omschrijving, maakdatum en dergelijke.
- De gegevens voor de SQL interactie. (install en uninstall scripts). Deze gaat over de achterliggende gegevens. (zie ook hoofdstuk gegevensanalyse)
- De gegevens voor de **language files**. Deze leggen een vertaalslag voor verschillende labels in het component, zodat deze zowel in het engels, als in het nederlands beschikbaar wordt.
- De gegevens voor de MVC (Model-View-Controller) interface.
- De gegevens voor de **table classes** die de database interactie regelen.

Deze **xml** file regelt dus, op het moment dat de zipfile aan Joomla wordt aangeboden, hoe de installatie ervan zich dient te votrekken.

3 gegevensanalyse en ontwerp

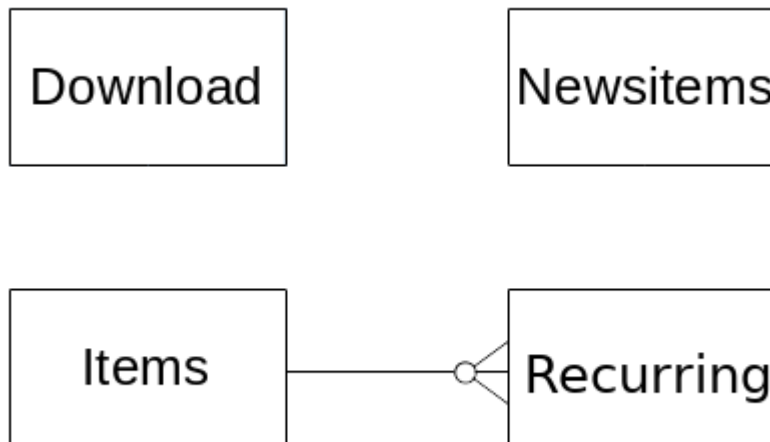
De achterliggende database bestaat uit vier tabellen.

- #__tks_agenda_download
- #__tks_agenda_newsitems
- #__tks_agenda_items
- #__tks_agenda_recurring

Waarvan de laatste twee een relatie hebben met elkaar.

Joomla vervangt de prefix #__ door de gegeven prefix van de namen van de Joomla tabellen, die bij installatie worden gegenereerd.

Het entiteiten relatie diagram is dan ook eenvoudig:



Waar de **items** tabel en de **recurring** tabel een relatie hebben met elkaar. Dit is alles wat er nodig is om de metadata van het component op te slaan.

4 functioneel ontwerp

Bij het schrijven van het functioneel ontwerp houd ik me niet aan enige standaard of (klassieke) ontwikkelingsmethodologieën. Dit niet alleen omdat het een reconstructie betreft van een bestaand produkt, waarbij dit ontwerp niet geleverd is, ook omdat het een eenvoudige opdracht betreft. Wel heb ik de overweging, dat mocht het geval zijn dat Takties mij vraagt componenten te maken, mij wel aan eenduidige richtlijnen te houden.

(In zulks een geval zou ik een template schrijven voor zowel een component als de documentatie ervan, maar dit is vooralsnog niet aan de orde).

Ik zou bij functioneel ontwerp kunnen denken aan:

- UML, met uitgebreide use-cases en alle andere relevante schema's.
- System Development Methodology, oud maar bruikbaar. (daarnaast ben ik er goed in onderlegd).
- Scrum, Rapid Application Development en soortgelijke.

Een grote vraag die ik me daarbij kan stellen is "Welke techniek past het beste in de organisatie, en bij het product?"

In dit project ga ik eigenlijk de omgekeerde richting als gewend is. Ik maak aan de hand van het bestaande project een opsomming van de functionaliteiten, in plaats van omgekeerd.

Dit is nuttig voor analyse van het project, en verschaft inzicht over eventuele aanpassingen.

In deze versie van het document is de opsomming van functionaliteiten nog niet beschikbaar.

4.1 Klassenhierarchie

Het component gebruikt verschillende klassen uit de API, die Joomla definieert.

De onderstaande opsomming reflecteert de klassenhierarchie. De klassen die Joomla definieert beginnen met een hoofdletter J, de klassen die takties definieert beginnen met tks.

- JComponentRouterBase
 - tks_agendaRouter
- JControllerForm
 - tks_agendaControllerNewsitemForm
 - tks_agendaControllerDownloadForm
- JControllerLegacy
 - tks_agendaController
 - tks_agendaControllerNewsitem
 - tks_agendaControllerDownload
 - tks_agendaControllerDownload
- JModelForm
 - tks_agendaModelNewsitemForm
 - tks_agendaModelItemForm
 - tks_agendaModelDownloadForm
- JModelItem
 - tks_agendaModelItem
 - tks_agendaModelDownload
 - tks_agendaModelNewsitem
- JModelList
 - tks_agendaModelItems
 - tks_agendaModelDownloads
 - tks_agendaModelNewsitems

- JTable
 - tks_agendaTableitem
 - tks_agendaTablenewsitem
 - tks_agendaTabledownload
- JViewLegacy
 - tks_agendaViewDownloadform
 - tks_agendaViewitems
 - tks_agendaViewitems
 - tks_agendaViewNewsitems
 - tks_agendaViewNewsitem
 - tks_agendaViewItem
 - tks_agendaViewItem
 - tks_agendaViewDownload
 - tks_agendaViewPortal
 - tks_agendaViewItemform
 - tks_agendaViewDownloads
 - tks_agendaViewNewsitemform
 - tks_agendaViewtks_agenda
- tks_agendaController
 - tks_agendaControllerItem
 - tks_agendaControllerItems
 - tks_agendaControllerDownloads
 - tks_agendaControllerItemForm
 - tks_agendaControllerNewsitems

Merk op dat all klassen die hier opgesomd staan een extensie zijn van Joomla gedefinieerde klassen, op **tks_agendaController** na.
De klassen die geen extensie zijn van andere klassen de volgende:

- tks_agendaHelper
- tks_agendaSiteFrontendHelper

De hierarchie een laag hierboven teken ik niet, omdat dit Joomla API specifieke klassen zijn. Raadpleeg de Joomla API documentatie voor de details.

4.2 Klassendiagram

In deze versie van het document is het klassendiagram nog niet beschikbaar.

4.3 Dialoogstructuur

In deze versie van het document is de weergave van de dialoogstructuur nog niet beschikbaar.

5 technisch ontwerp

Het technisch ontwerp bestaat uit de **phpdoc** gegenereerde API reference. Deze is in site formaat en is gepubliceerd op de testomgeving.

6 implementatie en testen

Bij het implementeren en testen dien ik testcases op te tekenen die de functionaliteiten uit het functioneel ontwerp proberen om zo eventuele fouten aan het licht te brengen.

Op moment van schrijven zijn de testcases nog niet beschikbaar

7 status van het project

Momenteel is het project dus nog in volle gang. Essentieel is het goed onder de loep nemen van de update functie van agenda items.

Wat het documenteren en testen betreft, wil ik met Takties in overleg treden over wat een einddocument (zoals deze) allemaal moet reflecteren.

Voor toekomstige projecten die Joomla componenten zijn, is het misschien goed een apart document te maken met eenduidige richtlijnen en werkwijzen.