

Utveckling av tema för Drupal

Case: Bolag X

Marcus Aschan

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Medieteknik
Identifikationsnummer:	3330
Författare:	Marcus Aschan
Arbetets namn:	Utveckling av tema för Drupal
Handledare (Arcada):	Johnny Biström
Uppdragsgivare:	Jack Russell & Terrier Co. Oy
<p>Sammandrag:</p> <p>Detta examensarbete handlar om utvecklingen av ett tema för innehållshanteringssystemet Drupal. Syftet med detta arbete är att få fördjupade kunskaper i hur man bygger upp teman för Drupal i samband med att bygga upp en webbplats för företaget Bolag X.</p> <p>Examensarbetet består av flera delar och kommer först att behandla vad innehållshanteringssystem är och berätta allmänt om Drupal, hur systemet är uppbyggt och om de olika funktionerna som finns i Drupal, och vad som menas med ett tema. Det här är en mera teoretisk del med avsikten att ge läsaren en bas för vad som kommer att behandlas i de senare kapitlen.</p> <p>I den andra delen behandlas mera på djupet hur utvecklingen av ett tema för Drupal går till och vilka metoder man kan utnyttja. Denna del är mera praktisk och ger läsaren en insikt i vad som krävs av ett Drupal tema och vilka metoder man kan använda för att få det utseendet man önskar. Här behandlas vilka filer som behövs då man skapar ett tema och vad det finns för filer man kan utnyttja för att få större kontroll på temat.</p> <p>Till sist ingår det en diskussionsdel var den färdiga produkten diskuteras och hur den lämpar sig för sitt ändamål.</p>	
Nyckelord:	Drupal, teman, design, underlag
Sidantal:	45
Språk:	Svenska
Datum för godkännande:	19.12.2011

DEGREE THESIS	
Arcada	
Degree Programme:	Media Technology
Identification number:	3330
Author:	Marcus Aschan
Title:	Development of a theme for Drupal
Supervisor (Arcada):	Johnny Biström
Commissioned by:	Jack Russell & Terrier Co. Oy
Abstract:	
<p>This thesis deals with the development of a theme for the content management system Drupal. The purpose of this work is to obtain a thorough knowledge of how to build themes for Drupal in connection with building a website for the company Bolag X.</p> <p>The thesis consists of several parts, and will first examine what CMS is and tell us generally about Drupal, how the system is structured and about the various features available in Drupal, and what is meant by a theme. This is a more theoretical part with the intention to give the reader a basis for what will be discussed in later chapters.</p> <p>The second part deals more in depth how the development of a theme for Drupal goes to and what methods you can use. This part is more practical and gives the reader an insight into what is required of a Drupal theme and what methods you can use to get the look you desire. Here are treated the files needed when creating a theme and what are the files you can use to get more control on the theme.</p> <p>Finally, it includes a discussion part were the final product is discussed and how it is fit for purpose.</p>	
Keywords:	Drupal, themes, design, templates
Number of pages:	45
Language:	Swedish
Date of acceptance:	19.12.2011

INNEHÅLL / CONTENTS

1	Inledning	9
1.1	Utgångspunkt	9
1.2	Målsättning	10
1.3	Metod	11
1.4	Begränsningar	12
2	Vad är ett CMS	12
3	Kort om Drupal	12
3.1	Vad är ett tema	13
3.2	Installera Drupal	13
3.2.1	<i>Systemkrav</i>	14
3.2.2	<i>Installation</i>	14
3.3	Drupals struktur	15
3.3.1	<i>Noder</i>	15
3.3.2	<i>Fält</i>	15
3.3.3	<i>Block</i>	16
3.3.4	<i>Moduler</i>	16
3.4	Installera ett tema	17
4	Hur ett Drupal tema är uppbyggt	17
4.1	Temats struktur	18
4.2	Phptemplate	19
4.3	Hjälpmedel	19
4.3.1	<i>Firebug</i>	19
4.3.2	<i>Devel</i>	21
4.4	Planering	22
4.5	Skapa de nödvändiga filerna	23
4.5.1	<i>Info-filen</i>	24
4.5.2	<i>Regioner</i>	26
4.5.3	<i>Funktioner</i>	28
4.6	Template filer	29
4.6.1	<i>page.tpl.php</i>	30
4.6.2	<i>html.tpl.php</i>	33
4.6.3	<i>node.tpl.php</i>	34
4.6.4	<i>Övriga template filer</i>	36
4.6.5	<i>template.php</i>	37
4.7	CSS	38
4.8	Javascript	39
5	Diskussion	42
5.1	Drupal	43

6	Källor / References	44
----------	----------------------------------	-----------

Figurer

Figur 1. En photoshop fil som beskriver målsättningen.....	11
Figur 2. Användargränssnittet för placering och skapande av block.....	16
Figur 3. Strukturen på ett Drupal tema.....	18
Figur 4. Firebug aktiverat på Bolag X sida.....	20
Figur 5. Devel kan användas för att hitta ett visst värde i en array.....	21
Tabell 1. Metod för att skriva ut värdet i ett fält.....	21
Figur 6. Placering av egna temakataloger och filer.....	24
Tabell 2. Grundläggande nycklar i .info filen.....	25
Tabell 3. Ytterligare nycklar i .info filen.....	25
Figur 7. Administrationssidan för temats inställningar i Drupal.....	26
Tabell 4. Metod för att exkludera en region.....	27
Tabell 5. Metod för att lägga till en region.....	27
Tabell 6. Utseendet på Bolag X .info fil.....	28
Figur 8. Inställningar för olika funktioner på administrations sidan.....	29
Tabell 7. Aktiverandet av funktioner i .info filen.....	29
Figur 9. Var man hittar page.tpl.php i Drupal core.....	31
Figur 10. Standard utseendet på en page.tpl.php fil.....	32
Figur 11. Standard html.tpl.php.....	34
Figur 12. Strukturen på node.tpl.php.....	35
Figur 13. Region.tpl.php.....	36
Figur 14. Funktion för att ändra på sökfältet.....	37
Tabell 8. Exempel på olika CSS filer för olika medietyper.....	38
Tabell 9. Funktion för att lägga till en extern CSS fil.....	39
Tabell 10. Metod för att lägga till skript.....	39
Tabell 11. Funktion för att lägga till externa skript.....	40
Tabell 12. Exempel på hur Javascript ska skrivas på sidan.....	40
Tabell 13. Exempel på hur man ska använda JQuery.....	41
Figur 14. Framsidan för webbplatsen.....	42

Ordförklaringar

Drupal core = En basinstallation av Drupal utan några extra moduler eller teman installerade

Node = Innehåll i Drupal. Alla sidor i Drupal består av noder.

Modul = Ett tillägg till Drupal som utökar systemets funktionalitet.

Region = Ett område på webbsidan så som ett sidhuvud eller sidfot.

Block = Block är mindre "rutor" med innehåll som kan flyttas till olika regioner.

HTML = Hypertext Markup Language

CSS = Cascading Style Sheets

PHP = PHP: Hypertext Preprocessor

CMS = Content Management System

Array = Används i programmering för att spara flera värden i en variabel. En samling med data.

1. INLEDNING

Ett innehållshanteringssystem kan vara till stor hjälp då man bygger upp en webbplats. I större innehållshanteringssystem, eller CMS:ar, finns ofta också inbyggt mycket som man kanske inte normalt tänker på eller kanske ens känner till då man bygger upp en webbplats, som t.ex. sökmotor optimering och säkerhet. Det finns också en massa funktioner som kan vara svåra och tidskrävande att utveckla på egen hand.

Det brukar gå relativt lätt att få de funktioner man behöver och forma webbplatsen enligt eget behov, däremot kan det vara en utmaning att få webbplatsen att se ut precis som man vill.

Mitt arbete kommer att bli i form av en manual som berättar om de olika stegen för att bygga ett eget tema i CMS:en Drupal. Utmaningen kommer att vara att förstå uppbyggnaden och logiken bakom Drupal, och det kommer att finnas en hel del begrepp och funktioner att lära sig.

1.1. Utgångspunkt

Arbetet är gjort som ett projekt som jag har fått via min nuvarande arbetsgivare Jack Russell & Terrier Co. Projektet utförs åt Bolag X som har beställt en ny webbplats. Bolag X ville ha en webbplats som är enkel att uppdatera och med förnyat utseende. För att hålla uppbyggnaden kostnadseffektiv blev valet att bygga upp webbplatsen med innehållshanteringssystemet Drupal (version 7).

Med innehållshanteringssystem kan man lägga mindre tid på själva funktionerna och mera tid på utseendet. Med Drupal går det också mycket enkelt att kontrollera användarrättigheter, och dela upp användare i olika grupper för att förhindra att någon gör ändringar som skulle ha stora konsekvenser för webbplatsens utseende.

1.2. Målsättning

Målsättningen för examensarbetet är att få utveckla ett tema för Bolag X som går att använda med en Drupal installation. Det är också meningen att få fördjupade kunskaper i hur man bygger upp ett tema och hur man kan ändra på det enligt eget behov.

Då projektet började fick jag en bild i photoshop format där alla lager ännu var användbara. Jag använde den bilden som utgångspunkt då jag började bygga på temat, och fick genom den information om färger, fonter och bilder. För strukturen hade jag en såkallad sitemap, eller sidkarta som beskrev hur de olika sidorna på webbplatsen ska vara sammanbundna.



Figur1. En photoshop fil som beskriver målsättningen

1.3. Metod

Forskningsmetoden för detta arbete kommer dels att bestå av litterär forskning i form av böcker och olika webbplatser samt även diskussionsforum. Jag kommer att utveckla temat medan jag skriver detta arbete och kommer därmed också att inkludera en praktisk del.

1.4. Begränsningar

I detta arbete kommer jag att utgå från att läsaren har en baskunskap i html och css och kommer inte att beskriva hur html och css filerna ser ut. Jag kommer att koncentrera mig på att berätta vad vilka filer som behövs för att skapa ett fungerande tema för Drupal och hur man kan utnyttja olika funktioner och vilka sidbotten som finns till förfogande. Arbetet kommer också endast att omfatta uppbyggnaden av ett tema med Drupals inbyggda themeengine, eller tema motor, phptemplate.

2. VAD ÄR ETT CMS

Ett CMS är ett innehållshanteringssystem som används för att publicera material på internet. Innehållshanteringssystem används för att underlätta uppbyggnaden av webbplatser och hjälpa till att göra webbplatserna mer dynamiska. Då man tidigare byggde webbplatser med skilda html-sidor för varje sida, kan man med en CMS enkelt fylla på innehåll eller lägga till nya sidor utan att behöva göra en ny html-sida, med all kod det skulle innebära för att inte nämna de ändringar som måste göras på tidigare sidor med nya menyer osv. Populära innehållshanteringssystem är Wordpress, Joomla och Drupal, som baserar sig på PHP och MySQL, eller Liferay som baserar sig på Java och PostgreSQL. (Wikipedia 2011a)

3. KORT OM DRUPAL

Drupal är ett CMS, eller innehållshanteringssystem med öppen källkod. Drupal är programmerat i PHP och använder sig av en databas för att spara data. Med öppen källkod menas att det är fritt för vem som helst att använda sig av systemet och göra ändringar enligt egen önskan. Drupal distribueras under GNU-licensen vilket innebär att Drupal är fritt att använda för valbart syfte, man kan göra ändringar och förbättringar och man har rättighet att vidare distribuera kopior.

Drupal är ett mycket modulärt och flexibelt innehållshanteringssystem som lätt går att utvidga med kontribuerade moduler som man hittar på drupal.org. Redan med Drupals standard installation kan man bl.a. skapa en eller flera bloggar, diskussionsforum, olika användarprofiler och bygga upp websidor. Drupal stöder också flera webbplatser på samma installation, som kan dela databas, ha skilda databaser eller delvis dela på databaser.

Drupal används av 1,6% av alla webbplatser internationellt, och av de webbplatser som använder sig av innehållshanteringssystem står Drupal för 6,3% (w3techs 2011). Kända webbplatser som använder sig av Drupal är bl.a. whitehouse.org, hbl.fi och data.gov.uk. Drupals community är också mycket stor vilket resulterar att det finns mycket olika synvinklar på hur systemet ska utvecklas och många som testar att allt fungerar som det ska och att det inte finns säkerhetsluckor. Dessutom går det enkelt att få stöd ifall man stöter på problem, och det finns oändlig mängd dokumentation att läsa.

(Wikipedia 2011b, Drupal.org 2011a)

3.1. Vad är ett tema

Då man talar om ett tema i samband med Drupal handlar det om en samling relaterade filer som ansvarar för utseende och den grafiska upplevelsen av Drupal. Temat inkluderar bland annat html, css och javascript filer. Kort sagt bestämmer ett tema hur webbplatsen visas för användaren. Ett tema kan beskrivas som en visuell behållare som används för att formatera och visa data på skärmen. Jag kommer att referera till tema för att beskriva de filer styr den visuella sidan av Drupal.

(Ric Shreves 2011)

3.2. Installera Drupal

För att kunna använda Drupal måste det installeras på en server. Installations processen är relativt enkel och användaren blir instruerad nästan hela vägen igenom. För en ovan användare kan det dock vara en utmaning.

3.2.1. Systemkrav

För att kunna installera Drupal behöves en webbserver, PHP-server och databas-server. Som webbserver rekommenderas Apache version 1.3 eller 2.x, även om det finns andra servrar som Drupals basinstallation ska fungera på som t.ex Microsoft IIS. Rekommenderade databas-servrar är MySQL 5.0.15 eller högre, eller MariaDB. PHP version 5.3 rekommenderas, och under 5.2.5 godkänns inte av Drupal. Drupal är inte beroende av någon viss platform så det går att installera Drupal på en PC såväl som på en MAC eller Linux baserad server. (Drupal.org 2011c)

3.2.2. Installation

Det första steget är att ladda ner en kopia av Drupal filerna från drupal.org. Filerna packas upp i den katalog man önskar ha sidan på, web root katalogen i de flesta fallen. Nästa steg är att skapa en databas och en användare för databasen som man ger fulla rättigheter åt. Drupal kommer med ett installations skript som gör installationen mycket enkel. Då man packat upp filerna och skapat en databas är det bara att navigera skriva adressen på katalogen var Drupal filerna ligger och installationen påbörjas automatiskt.

Då installationen är avklarad rekommenderas det att man navigerar till *Status report* sidan inom Drupal och granskar möjliga problem. De problem som brukar uppstå handlar för det mesta om olika inställningar på servern.

(Drupal.org 2011d)

3.3. Drupals struktur

Drupals system bygger på olika moduler som baserar sig på så kallade *hooks* eller krokar som är funktioner i Drupal. I en funktion med namnet `foo_bar()` står `foo` för modulens namn (vars filnamn skulle vara `foo.module`) och `bar` är namnet på kroken.

3.3.1. Noder

Allt innehåll på Drupal webbplatser behandlas som noder. Det finns olika innehållstyper, så som *Basic page* eller *Article*, i en basinstallation av Drupal och alla dessa är noder. Genom att behandla allt innehåll som noder finns möjligheten till större flexibilitet. T.ex. kan användaren skapa egna innehållstyper för att ha olika sidbotten för olika sorters sidor.

För att skapa en ny innehållstyp navigerar man till *Administration > Structure > Content types > Add content type*. På det här viset kan man ha ett skilt sidbotten för en sida som skiljer sig mycket till utseendet från andra sidor. T.ex. kan man skapa en innehållstyp som heter *Gallery* som man använder sig av för att göra sidor med bildgallerier. (Drupal.org 2011k, Todd Tomlinson 2010)

3.3.2. Fält

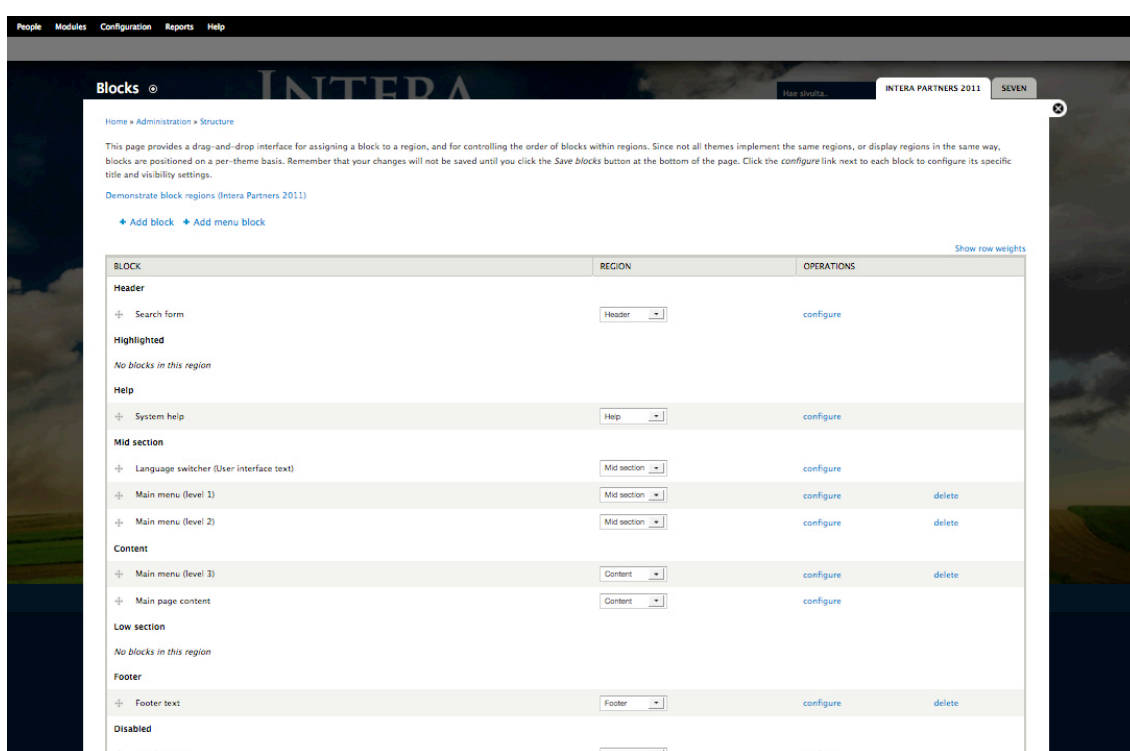
Innehåll i Drupal byggs upp av olika fält. En nods rubrik är ett fält, likaså brödtexten. Administratören har möjlighet att lägga till nya fält till olika innehållstyper så som *Basic page* och *Article*. Dessa fält kan t.ex. vara en bild eller en underrubrik. Det går också att ändra på ordningen på fälten.

(Todd Tomlinson 2010)

3.3.3. Block

Block är olika komponenter på en sida. Huvudmenyn är ett exempel på ett block som kan placeras i olika regioner, som *header* eller *sidebar* genom Drupals grafiska användargränssnitt. Andra exempel på block är inloggnings- och sökfält. Många block skapas av moduler men användaren har också möjlighet att skapa egna block och placera dem enligt eget tycke.

(Todd Tomlinson 2010)



Figur 2. Användargränssnittet för placering och skapande av block.

3.3.4. Moduler

Moduler står för Drupals funktionalitet, och kan aktiveras eller avaktiveras enligt användarens eget behov. Vissa moduler finns med i basinstallationen och krävs för att systemet ska fungera, medan andra går att avinstallera om man inte har behov för dem. Ett exempel på en modul som följer med Drupals basinstallation är *Block*, som ger användaren möjligheten att skapa block. Ett annat exempel är *Field* som gör att användaren kan

skapa nya fält för innehållstyper. Med hjälp av moduler kan webbplatser hållas smidiga genom att bara använda de moduler som behövs, medan systemet är flexibelt och enkelt går att utvidga. (Todd Tomlinson 2011)

3.4. Installera ett tema

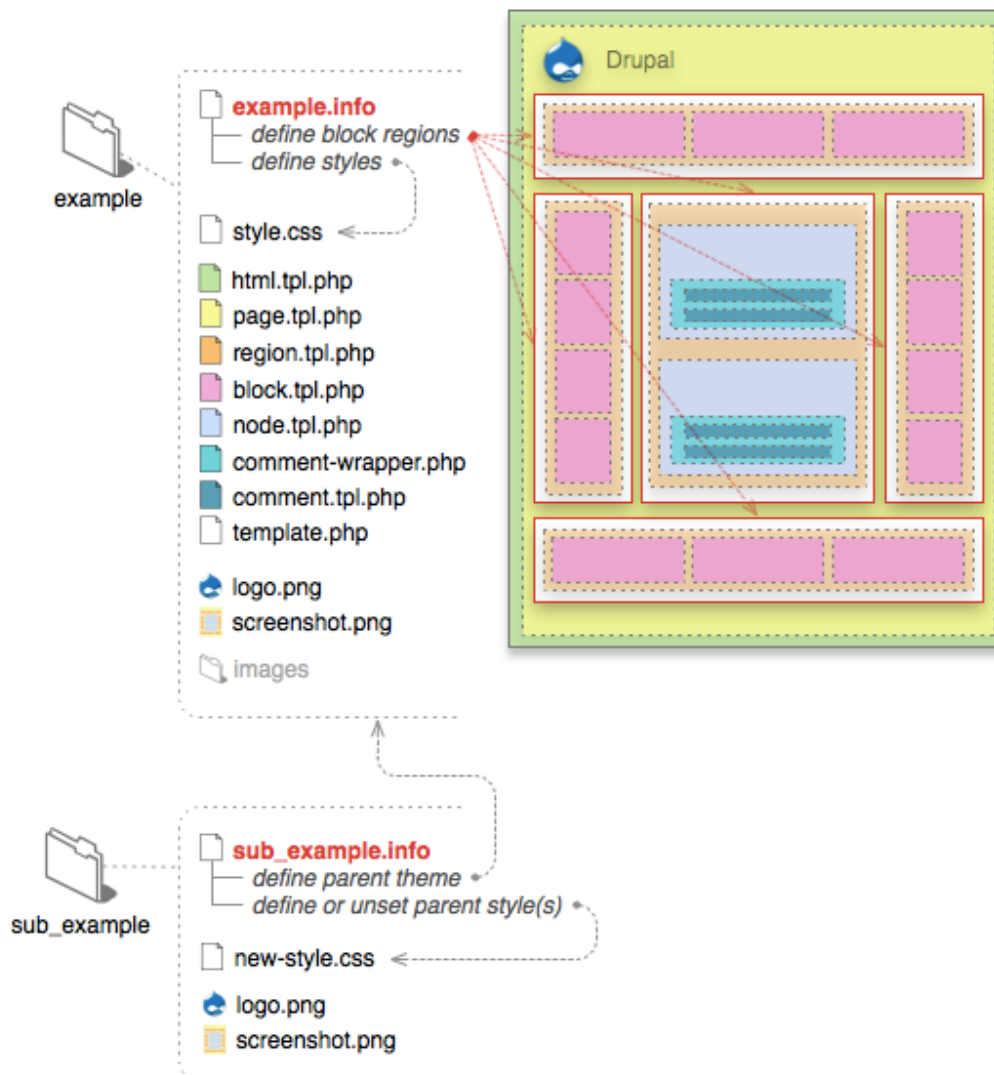
Det finns ett stort utbud av teman för Drupals webbplats som är fria att använda. För att installera ett nytt tema kan man gå till <http://drupal.org/project/themes> och välja ett tema. Temat laddas ner i packat format och så packar man upp temat och placerar det i `sites/all/themes`, varefter man navigerar till `/admin/appearance/` och väljer sitt nya tema. Det går att använda flera teman på samma gång men man måste ha ett standard tema.

(Ric Shreves 2011)

4. HUR ETT DRUPAL TEMA ÄR UPPBYGGT

I följande kapitel kommer jag att beskriva hur Drupal teman normalt är uppbyggda och vilka som är de vanligaste byggklossarna.

4.1. Temats struktur



Figur 3. Strukturen på ett Drupal tema. (Drupal.org 2011g)

Inuti temats katalog bör finnas en fil med samma namn som katalogen och ändelsen `.info`, som är konfigurationsfilen för temat. Där definieras css och javascript och ett antal andra funktionaliteter.

Drupal söker också efter såkallade template filer med ändelsen `.tpl.php`. Dessa krävs inte eftersom de finns från tidigare i Drupal core, men är nyttiga om man vill göra större ändringar på temat. De viktigaste är `page.tpl.php` som står för sidans ram och de element som inte växlar mellan varje sida, så som menyer, och `node.tpl.php` som ansvarar för det

innehåll som varierar mellan sidorna, t.ex text och bilder. Drupal använder sig av många olika .tpl.php filer och en stor del innehåller vanlig html med php variabler för att skriva ut dynamiskt innehåll.

För att sköta om förhandsprocessering finns det en fil som heter template.php. Denna fil behövs inte men är nyttig ifall man vill generera variabler för att använda i template filerna eller kringgå Drupals egna funktioner. (Drupal.org 2011g)

4.2. Phptemplate

Theme engine, eller temamotor är en samling script och filer som samverkar med Drupals kärna och tolkar dess programmeringsspråk. Drupal distribueras med phptemplate som dess tema motor. Själva programmeringsspråket PHP är egentligen en template engine. (Wikipedia 2011c, Ric Shreves 2011)

4.3. Hjälpmedel

Det finns många hjälpmedel man kan använda sig av då man utvecklar teman för Drupal. En del är sådana som kan vara bra även om det handlar om annan webbutveckling och andra är specifika för Drupal.

4.3.1. Firebug

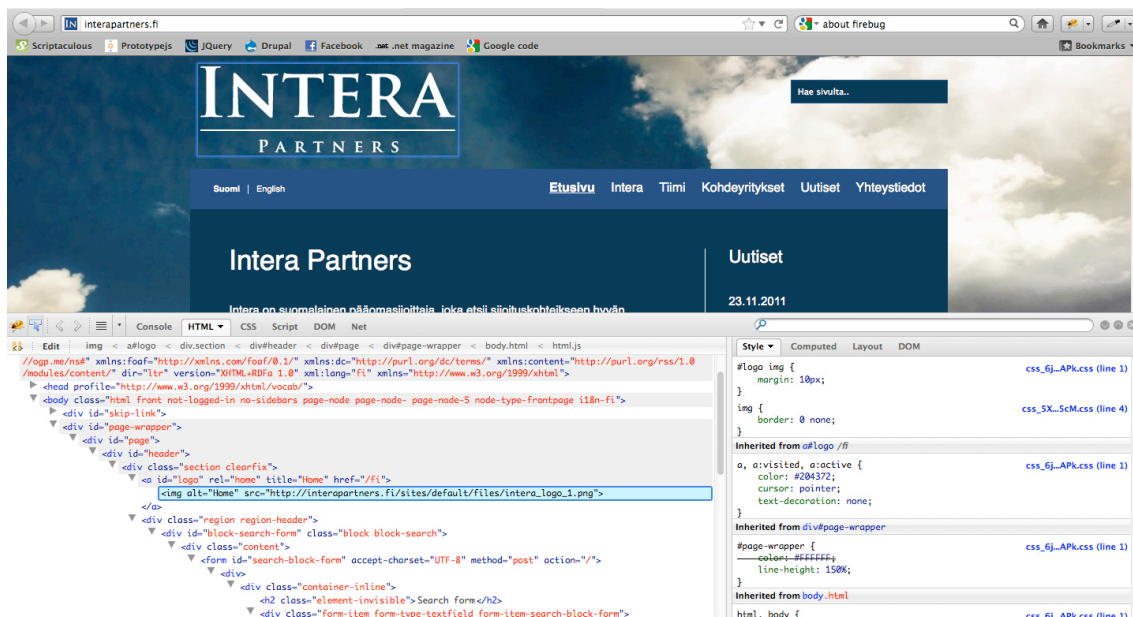
Firebug är ett tillägg för webbläsaren Firefox, som är till som hjälpmedel för webbutvecklare. Firebug ger utvecklaren möjlighet att inspektera HTML element som är djupt inne på sidan och välja ett element för att se information om elementet, så som CSS. Med Firebug går det att göra ändringar som syns direkt i webbläsaren, och underlättar på det viset så att utvecklaren inte behöver ladda upp sina ändringar och uppdatera sidan för att se dem.

Då man jobbar med Drupal som är ett system som skriver ut största delen av HTML taggarna genom PHP skulle det vara ytterst svårt att veta vilka klasser och ID:n som de olika taggarna har utan ett hjälpmedel som Firebug.

Firebug kan också användas för att analysera nätverksaktivitet och få information om hur länge olika filer och bilder tar att laddas. På det viset kan man enkelt hitta flaskhalsar i sin kod som gör att sidan blir långsam.

De flesta webbläsare har sina egna motsvarigheter till Firebug men få är lika genomtänkta och lätta att använda som Firebug. Till Internet Explorer finns det t.ex. ett verktyg som heter Developer Toolbar. (Firebug 2011)

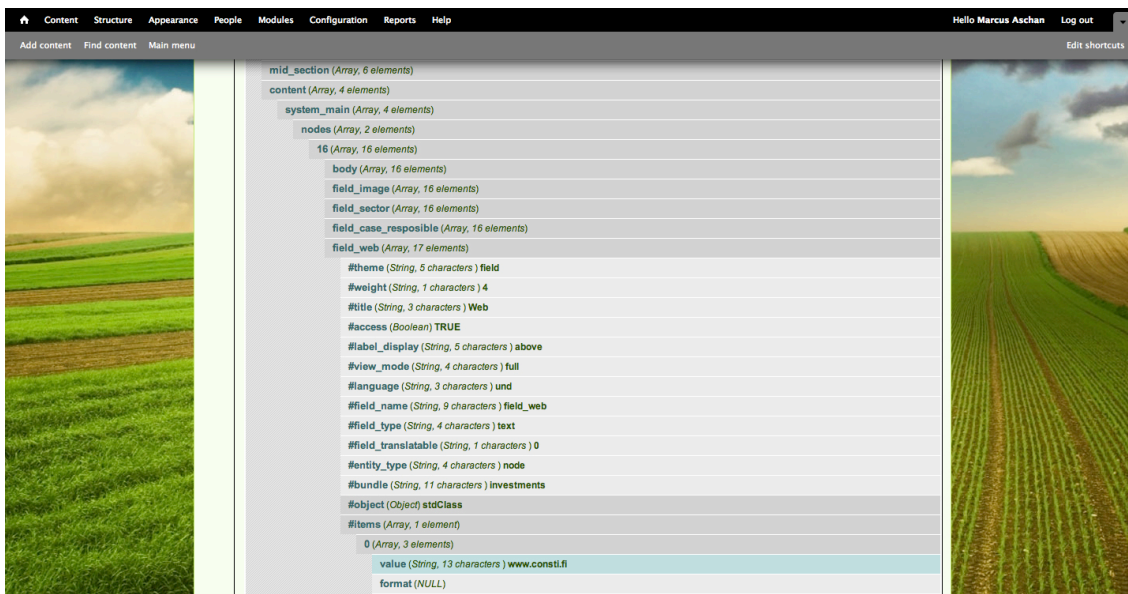
Firebug finns att hämta från <http://getfirebug.com>.



Figur 4. Firebug aktiverat på Bolag X sida.

4.3.2. Devel

Devel är en tilläggsmodul för Drupal som kan hämtas från drupal.org/project/devel. Devel är en modul som ger utvecklaren möjlighet att se vilka variabler och samlingar (arrays) som en viss sida använder sig av för att skriva ut data. Det går också att granska databasfrågor med hjälp av *Devel*.



Figur 5. Devel kan användas för att hitta ett visst värde i en array.

I bilden ovanför användes *Devel* för att hitta den array som innehåller värdet för ett fält som skapats tidigare i Drupal's grafiska användargränssnitt. Värdet kunde sedan skrivas ut med PHP på följande sätt.

Tabell 1. Metod för att skriva ut värdet i ett fält.

```
foreach ($field_web as $value) {  
    print $value['value'];  
}
```

Field web är ett fält som skapades genom Drupal's grafiska användargränssnitt och används för att fylla i en eller flera webbadresser. Variabeln *\$field_web* har fått sitt namn från fältets maskinläsbara namn. Det skulle ha varit utmanande att reda ut på vilket djup

i array:en som värdet ligger utan *Devel*. Orsaken till behovet att få ut värdet på detta vis kan variera, men var i detta fall att få möjligheten att lägga till extra HTML.

4.4. Planering

Det första man bör göra innan man börjar bygga sitt tema är att planera hur det ska se ut. Det kan spara mycket tid och besvär om man går igenom i vilka delar sidan ska byggas upp. Det hjälper att bara skissa upp på papper vad för regioner som ska ingå. Ska det finnas ett sidhuvud och var ska menyn vara placerad? Kommer innehållet att delas upp i flera områden och vill man ha sidobalkar?

Standardregioner i Drupal är följande:

- Header
- Highlighted
- Help
- Content
- Left sidebar
- Right sidebar
- Footer

Man är dock inte bunden till dessa regioner och kan ta bort eller lägga till regioner enligt eget tycke, vilket jag kommer att behandla senare i arbetet.

För det egna projektet ville jag ha fyra regioner.

- header
- menu bar
- content
- footer

Header regionen ska innehålla logon och ett sökfält, menu-bar ska innehålla huvudmenyn, content innehåller allt innehåll som växlar mellan sidorna och footer innehåller en nedre meny och kontaktuppgifter.

4.5. Skapa de nödvändiga filerna

Då planeringen är avklarad kan man börja bygga ett eget tema för Drupal genom att skapa en katalog som innehåller alla filer som temat använder sig av och placera den i `sites/all/themes`. Drupal är uppbyggt så att systemet automatiskt söker genom `sites/all/themes` efter `.info` filer som definierar ett tema. Då man bygger ett tema ska man endast skapa och ändra på filer i denna katalog, vilket underlättar att skilja på egna filer och systemets filer. Det är viktigt att veta skillnad på de egna filerna och Drupals filer då man t.ex. gör uppdateringar.

Nästa steg är att skapa `.info` filen som krävs för att skapa ett tema som är användbart av Drupal. Filen ska ha samma namn som katalogen som innehåller temat. Om man t.ex. skapar följande folder: `sites/all/themes/intera`, så letar drupal efter en fil med följande namn: `intera.info`. Namnet på både katalogen och `.info` filen måste vara läsbart av datorn och måste därmed börja med en bokstav och får endast innehålla bokstäver, siffror och understreck men inte punkter eller mellanstreck.

För att kunna göra större ändringar på utseendet behövs också en CSS fil. I Drupal 6 räckte det med att skapa `style.css` och placera den i samma katalog som `.info` filen, men i Drupal 7 måste CSS filen definieras. Det är nästa steg då man börjar fylla på information i `.info` filen. (Ric Shreves 2011)

Name	Date Modified	Size	Kind
▼ drupal-7.8	Today 7:31 PM	--	Folder
CHANGELOG.txt	Aug 31, 2011 9:55 PM	61 KB	Plain Text
COPYRIGHT.txt	Aug 31, 2011 9:55 PM	996 bytes	Plain Text
INSTALL.mysql.txt	Aug 31, 2011 9:55 PM	1 KB	Plain Text
INSTALL.pgsql.txt	Aug 31, 2011 9:55 PM	2 KB	Plain Text
INSTALL.sqlite.txt	Aug 31, 2011 9:55 PM	1 KB	Plain Text
INSTALL.txt	Aug 31, 2011 9:55 PM	18 KB	Plain Text
LICENSE.txt	Feb 24, 2011 2:47 AM	15 KB	Plain Text
MAINTAINERS.txt	Aug 31, 2011 9:55 PM	8 KB	Plain Text
README.txt	Aug 31, 2011 9:55 PM	3 KB	Plain Text
robots.txt	Aug 31, 2011 9:55 PM	2 KB	Plain Text
UPGRADE.txt	Aug 31, 2011 9:55 PM	9 KB	Plain Text
authorize.php	Aug 31, 2011 9:55 PM	7 KB	PHP:...cument
cron.php	Aug 31, 2011 9:55 PM	720 bytes	PHP:...cument
index.php	Aug 31, 2011 9:55 PM	529 bytes	PHP:...cument
install.php	Aug 31, 2011 9:55 PM	688 bytes	PHP:...cument
update.php	Aug 31, 2011 9:55 PM	18 KB	PHP:...cument
xmlrpc.php	Aug 31, 2011 9:55 PM	417 bytes	PHP:...cument
▶ includes	Aug 31, 2011 9:55 PM	--	Folder
▶ misc	Aug 31, 2011 9:55 PM	--	Folder
▶ modules	Aug 31, 2011 9:55 PM	--	Folder
▶ profiles	Aug 31, 2011 9:55 PM	--	Folder
▶ scripts	Aug 31, 2011 9:55 PM	--	Folder
▼ sites	Today 7:31 PM	--	Folder
example.sites.php	Aug 31, 2011 9:55 PM	2 KB	PHP:...cument
▼ all	Today 7:31 PM	--	Folder
README.txt	Aug 31, 2011 9:55 PM	347 bytes	Plain Text
▶ modules	Aug 31, 2011 9:55 PM	--	Folder
▼ themes	Today 7:32 PM	--	Folder
README.txt	Aug 31, 2011 9:55 PM	161 bytes	Plain Text
▶ intera	Today 7:32 PM	--	Folder
▶ default	Aug 31, 2011 9:55 PM	--	Folder
▶ themes	Aug 31, 2011 9:55 PM	--	Folder
web.config	Aug 31, 2011 9:55 PM	2 KB	Document

Figur 6. Placering av egna temakataloger och filer.

4.5.1. Info-filen

Info-filen är temats konfigurations fil, och det finns ett antal så kallade nycklar som måste definieras i info filen för att temat skall kunna användas av Drupal.

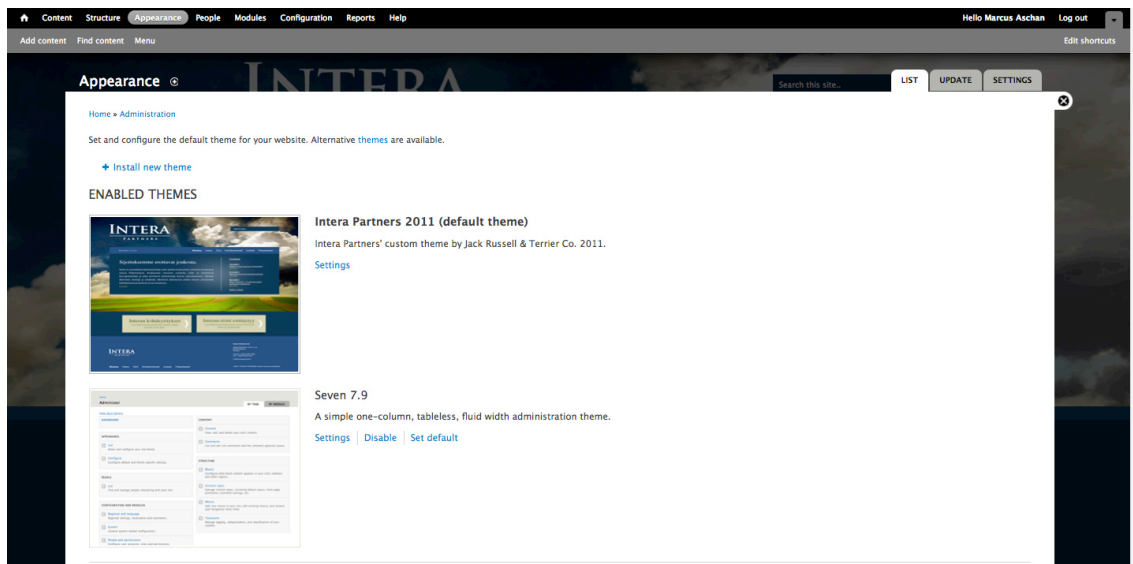
Tabell 2. Grundläggande nycklar i .info filen.

Nyckel	Beskrivning
name = Intera (krävs)	Det här fältet krävs. Här kommer namnet på temat som det syns i Drupals grafiska användargränssnitt.
description = Intera's fixed width theme (rekommenderas)	Description är ett rekommenderat fält. Här ska man beskriva temat. Beskrivningen syns i Drupals grafiska användargränssnitt då man väljer tema.
core = 7.x (krävs)	Fältet krävs. Det berättar vilken version av Drupal temat är ämnat för.
engine = phptemplate (rekommenderas)	Fältet definierar phptemplate som tema motor. Detta ska vara phptemplate.
stylesheets[all][] = css/style.css	Här definierar man sökvägen till CSS filen som skapades tidigare. I detta fall finns filen i katalogen css inuti tema katalogen.

Det här skulle räcka för att skapa ett eget tema. Man skulle kunna navigera till admin/appearance/list i sin Drupal installation och aktivera sitt nya tema. Sedan skulle det bara vara att fylla på med CSS. Det finns dock mycket mer man kan göra med sitt tema.

Tabell 3. Ytterligare nycklar i .info filen.

Nyckel	Beskrivning
screenshot = screenshot.png	Screenshot används om man vill ha en bild vid beskrivningen av temat. Detta fält krävs ej. Här ges sökvägen till bilden som i detta exempel befinner sig i temats katalog /sites/all/themes/intera.
scripts[] = js/script.js	Sökvägen till script på samma sätt som med CSS. Det går att definiera fler script genom att fylla på flera rader.
php = 5.3	Detta fält behövs inte ifall man inte har ett tema som kräver nyare version av PHP än Drupals standard installation kräver, vilket är 5.3.



Figur 7. Administrationsidan för temats inställningar i Drupal.

Figur 7 visar hur det ser ut då man har aktiverat temat och använder Drupals eget tema Seven som administrations tema. (Drupal.org 2011e)

4.5.2. Regioner

Man kan enkelt dela upp sin sida i olika regioner genom att definiera dem i temats .info fil. En region kan t.ex. vara ett sidhuvud, ett sidofält eller en sidfot, osv. Då man har definierat en region kan man placera olika block som t.ex. en meny eller sökfält i den regionen. Som tidigare nämnt finns det ett antal standard regioner.

- Header
- Highlighted
- Help
- Content
- Left sidebar
- Right sidebar
- Footer

Dessa inkluderas av Drupal automatisk om man inte definierar något annat. Om man inte vill ha med alla regioner som Drupal inkluderar måste man definiera de som man vill ha kvar. Om man t.ex. önskar lämna bort Right sidebar skulle det se ut enligt följande i .info filen.

Tabell 4. Metod för att exkludera en region.

```
regions[header] = Header
regions[highlighted] = Highlighted
regions[help] = Help
regions[content] = Content
regions[sidebar_first] = Left sidebar
regions[footer] = Footer
```

Det finns dock en region som inte får lämnas bort och det är content. På samma sätt som man tar bort regioner kan man också lägga till regioner. Låt oss säga att man vill ha en region för menyn under sidhuvudet (Header). Då skulle det se ut på följande sätt i .info filen.

Tabell 5. Metod för att lägga till en region.

```
regions[header] = Header
regions[menu] = Menu
regions[highlighted] = Highlighted
regions[help] = Help
regions[content] = Content
regions[sidebar_first] = Left sidebar
regions[footer] = Footer
```

Min .info fil skulle då se ut enligt tabell 6.

Tabell 6. Utseendet på Bolag X .info fil.

```
name = Intera
description = Intera's fixed width theme
core = 7.x
engine = phptemplate
screenshot = screenshot.jpg

regions[header] = Header
regions[highlighted] = Highlighted
regions[help] = Help
regions[menu_bar] = Menu bar
regions[content] = Content
regions[footer] = Footer

stylesheets[all][] = css/style.css

scripts[] = js/placeholder.js
```

För att kunna dra nytta av de regionerna måste de skrivas ut med PHP. Det behandlas i nästa kapitel. (Drupal.org 2011e, Ric Shreves 2011)

4.5.3. Funktioner

Det finns olika element på sidan som går att aktivera och stänga av genom Drupals grafiska användargränssnitt. Om man navigerar till *Administer >> Apperence >> Settings >> Tema namn* så har man möjlighet att välja dessa element.

TOGGLE DISPLAY

Enable or disable the display of certain page elements.

- Logo
- Site name
- Site slogan
- User pictures in posts
- User pictures in comments
- User verification status in comments
- Shortcut icon
- Main menu
- Secondary menu

Figur 8. Inställningar för olika funktioner på administrations sidan. (Drupal.org 2011f)

Dessa kryssrutor är synliga beroende på vad som är definierat i .info filen. Om inga funktioner är definierade så kommer alla att synas. Om man vill ta bort en funktion måste man på samma sätt som med regionerna definiera alla utom den man inte vill ha med. De tillgängliga funktionerna är följande. (Drupal.org 2011f)

Tabell 7. Aktiverandet av funktioner i .info filen.

```
features[] = logo
features[] = name
features[] = slogan
features[] = node_user_picture
features[] = comment_user_picture
features[] = comment_user_verification
features[] = favicon
features[] = main_menu
features[] = secondary_menu
```

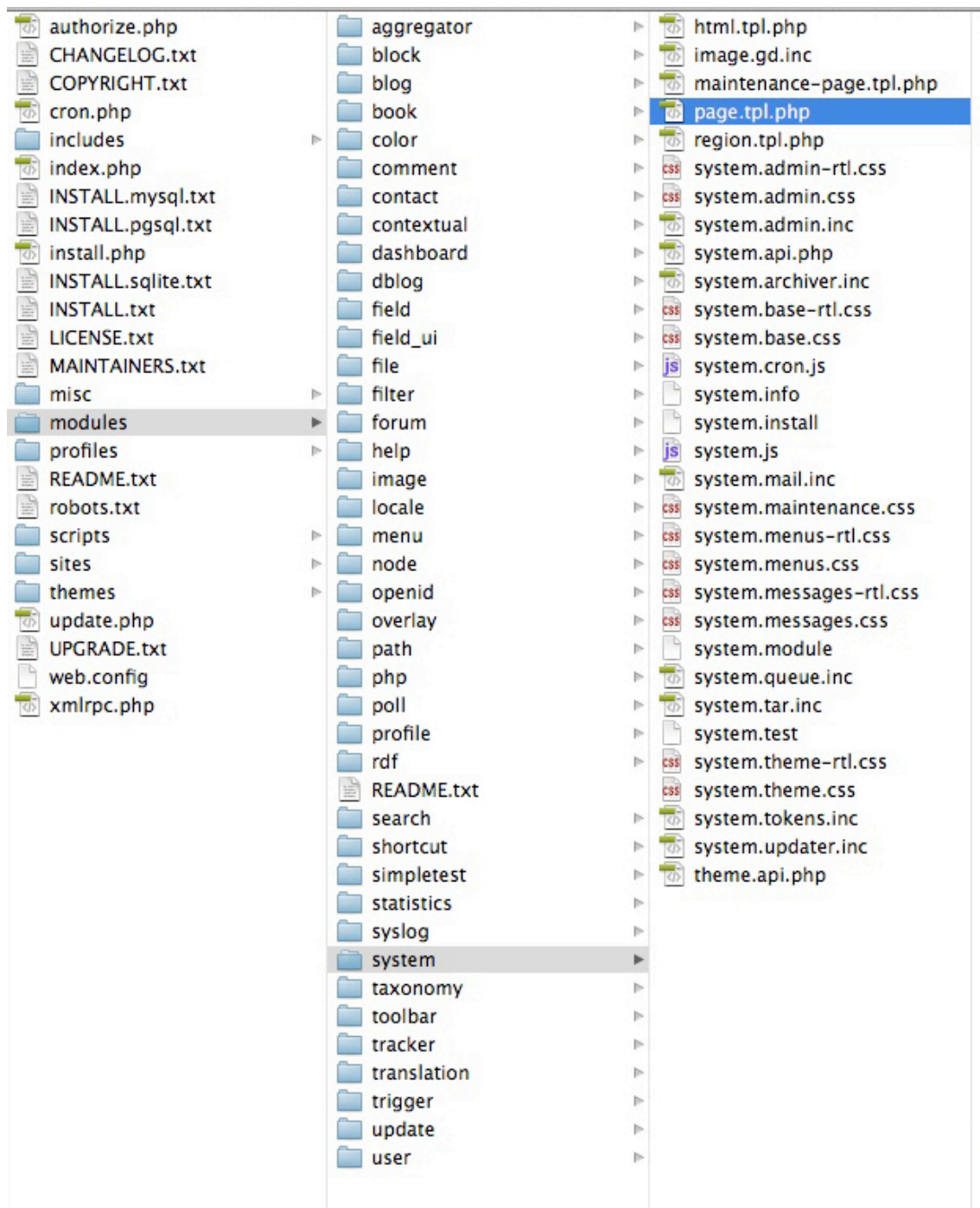
4.6. Template filer

Drupal använder sig av template filer eller mallar för att bygga upp sidan och de olika elementen som ska inkluderas. Drupal känner igen dessa filer på ändelsen .tpl.php. Filerna består av en blandning av HTML och PHP. Ingen av dessa filer behöver användas eftersom Drupal använder sig av dess egna filer ifall systemet inte hittar diverse mallar. Det ger dock en hel del flexibilitet att använda sig av template filerna.

4.6.1. page.tpl.php

Page.tpl.php är den viktigaste av alla template filer eftersom man med den kan ändra i stort sett på hela sidans utseende. Page.tpl.php finns med i Drupal core och den kommer att användas alltid då det inte finns en fil med samma namn i det aktiva temats katalog. Det är page.tpl.php som ska användas för att aktivera de regioner som deklarerades i kapitlet om regioner.

För enkelhetens skull kan man kopiera Drupals ursprungliga fil från *modules/system/* och placera kopian i det egna temats katalog. För att Drupal ska förstå att leta efter den nya filen bör man rensa cache minnet genom att navigera till *admin/config/development/performance* och välja *Clear all caches*. Normalt måste man alltid tömma cashe minnet då man har skapat en ny template fil, för att Drupal ska ta den i beaktande. (Ric Shreves 2011)



Figur 9. Var man hittar page.tpl.php i Drupal core.

```

<div id="page-wrapper"><div id="page">
  <div id="header"><div class="section clearfix">
    <?php if ($logo): ??
      <a href="#"><?php print $front_page; ??> title="<?php print t('Home'); ??>" rel="home" id="logo">
        <?php print $logo; ??> alt="<?php print t('Home'); ??>" />
      </a>
    <?php endif; ??

    <?php if ($site_name || $site_slogan): ??
      <div id="name-and-slogan">
        <?php if ($site_name): ??
          <?php if ($title): ??
            <div id="site-name"><strong>
              <a href="#"><?php print $front_page; ??> title="<?php print t('Home'); ??>" rel="home"><span><?php print $site_name; ??></span></div>
            </strong></div>
          <?php else: /* Use h1 when the content title is empty */ ??
            <h1 id="site-name">
              <a href="#"><?php print $front_page; ??> title="<?php print t('Home'); ??>" rel="home"><span><?php print $site_name; ??></span></div>
            </h1>
          <?php endif; ??
        <?php endif; ??

        <?php if ($site_slogan): ??
          <div id="site-slogan"><?php print $site_slogan; ??</div>
        <?php endif; ??
      </div> <!-- /#name-and-slogan -->
    <?php endif; ??

    <?php print render($page['header']); ??
  </div></div> <!-- /.section, /#header -->

  <?php if ($main_menu || $secondary_menu): ??
    <div id="navigation"><div class="section">
      <?php print theme('links__system_main_menu', array(
        'links' => $main_menu, 'attributes' => array(
          'id' => 'main-menu',
          'class' => array('links', 'inline', 'clearfix'),
          'heading' => t('Main menu')));
      ??
      <?php print theme('links__system_secondary_menu', array(
        'links' => $secondary_menu, 'attributes' => array(
          'id' => 'secondary-menu',
          'class' => array('links', 'inline', 'clearfix'),
          'heading' => t('Secondary menu')));
      ??
    </div></div> <!-- /.section, /#navigation -->
    <?php endif; ??

  <?php if ($breadcrumb): ??
    <div id="breadcrumb"><?php print $breadcrumb; ??</div>
  <?php endif; ??

  <?php print $messages; ??

  <div id="main-wrapper"><div id="main" class="clearfix">
    <div id="content" class="column"><div class="section">
      <?php if ($page['highlighted']): ??<div id="highlighted"><?php print render($page['highlighted']); ??</div><?php endif; ??
      <a id="main-content"></a>
      <?php print render($title_prefix); ??
      <?php if ($title): ??<h1 class="title" id="page-title"><?php print $title; ??</h1><?php endif; ??
      <?php print render($title_suffix); ??
      <?php if ($tabs): ??<div class="tabs"><?php print render($tabs); ??</div><?php endif; ??
      <?php print render($page['help']); ??
      <?php if ($action_links): ??<ul class="action-links"><?php print render($action_links); ??</ul><?php endif; ??
      <?php print render($page['content']); ??
      <?php print $feed_icons; ??
    </div></div> <!-- /.section, /#content -->

    <?php if ($page['sidebar_first']): ??
      <div id="sidebar-first" class="column sidebar"><div class="section">
        <?php print render($page['sidebar_first']); ??
      </div></div> <!-- /.section, /#sidebar-first -->
    <?php endif; ??

    <?php if ($page['sidebar_second']): ??
      <div id="sidebar-second" class="column sidebar"><div class="section">
        <?php print render($page['sidebar_second']); ??
      </div></div> <!-- /.section, /#sidebar-second -->
    <?php endif; ??

  </div></div> <!-- /#main, /#main-wrapper -->

  <div id="footer"><div class="section">
    <?php print render($page['footer']); ??
  </div></div> <!-- /.section, /#footer -->
</div></div> <!-- /#page, /#page-wrapper -->

```

Figur 10. Standard utseendet på en page.tpl.php fil.

Sidan packas in i två div-element och därefter kommer de olika regionerna och dess innehåll. Header regionen omges också av två div-element och sedan skrivs logon ut om en sådan finns. Allt det som man lägger till i header regionen genom det grafiska använ-

dargränssnittet skrivs ut genom funktionen *print render(\$page['header']);*. Med samma funktion kan man lägga till regioner som man har definierat i .info filen. T.ex menu bar skulle skrivas ut på följande sätt; *print render(\$page['menu-bar']);*. Om man använder sig av egna regioner är det bra att komma ihåg att ta bort de som inte finns definierade i .info filen från page.tpl.php.

Det finns en stor mängd variabler till förfogande för att skriva ut diverse innehåll, och de finns alla listade i en kommentar i början av Drupals standard page.tpl.php. Ett exempel är *\$logo* som skriver ut sidans logo, eller *\$site_name* som skriver ut sidans namn. Utöver de som redan används av standard filen finns det en samling variabler listade som kan vara till hjälp för temats utvecklare. Variabeln *\$directory* ger sökvägen till det aktiverade temat. Resultatet skulle vara *sites/all/themes/intera*, och kan t.ex. användas för att hitta en bild i temats katalog. (Todd Tomlinson 2010)

4.6.2. html.tpl.php

Från och med Drupal 7 finns det en fil som heter html.tpl.php. Filen innehåller dokument typen, head taggen med allt dess innehåll och body taggen, och är med andra ord behållaren för allt innehåll som kommer att finnas på sidan. I html.tpl.php skrivs bl.a dokumentets rubrik, alla CSS filer och skript ut. Om man inte använder en html.tpl.php fil i sitt tema så kommer Drupal att använda sin egen. Filen kan dock vara användbar ifall man vill använda externa skript eller CSS filer, vilka inte går att definiera i .info filen. Drupals standard html.tpl.php hittas i katalogen */modules/system/*.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
"http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php print $language->language; ?>"
version="XHTML+RDFa 1.0" dir="<?php print $language->dir; ?>"<?php print $rdf_namespaces;
?>>

<head profile="<?php print $grddl_profile; ?>">
  <?php print $head; ?>
  <title><?php print $head_title; ?></title>
  <?php print $styles; ?>
  <?php print $scripts; ?>
</head>
<body class="<?php print $classes; ?>" <?php print $attributes;?>>
  <div id="skip-link">
    <a href="#main-content" class="element-invisible element-focusable"><?php print t('Skip
to main content'); ?></a>
  </div>
  <?php print $page_top; ?>
  <?php print $page; ?>
  <?php print $page_bottom; ?>
</body>
</html>

```

Figur 11. Standard *html.tpl.php*.

Filen innehåller en del variabler som största delen är ganska uppenbara som t.ex *\$head_title* som skriver ut sidans rubrik, *\$styles* som skriver ut CSS filer eller *\$scripts* som skriver ut javascript.

Innanför *body* taggen finns det tre variabler som ansvarar för sidans synliga innehåll och dessa är *\$page_top*, *\$page* och *\$page_bottom*. Den viktigaste är *\$page* som skriver ut allt innehåll som finns i *page.tpl.php* filen. De två andra är s.k. osynliga regioner vilket innebär att de inte syns på administrations sidan för block, däremot kan de användas av moduler för att skriva ut data. (Pingv.com 2011, Todd Tomlinson 2010)

4.6.3. *node.tpl.php*

Genom *node.tpl.php* kan man styra utseendet på en nod. Här kommer allt det innehåll som varierar beroende på vilken sida man befinner sig på, själva innehålls texten. Det här är en användbar fil då man skapar olika innehållstyper och vill presentera dessa på olika sätt.

I en basinstallation finns det två olika innehållstyper; *Article* och *Basic page*. Om man skulle önska att ha olika sidbotten för de olika innehållstyperna kan man skapa en fil som heter *node--article.tpl.php* och en som heter *node--page.tpl.php*. *Article* och *page* är respektive innehållstypers maskinläsbara (machine readable) namn. Det går också att ge en särskild nod ett eget sidbotten genom att använda dess id nummer. Om man t.ex. har skapat en nod med url:en *exempel.fi/node/2* så kan man skapa en fil med namnet *node--2.tpl.php*. Då man skapar nya sidbotten är det bra att komma ihåg att tömma cache minnet för att Drupal ska registrera den nya filen.

Node.tpl.php hittas i katalogen *modules/node/*. I den filen hittar man också diverse variabler som finns till förfogande då man vill skriva ut data. (Todd Tomlinson 2010)

```
<div id="node-<?php print $node->nid; ?>" class="<?php print $classes; ?> clearfix"<?php
print $attributes; ?>>

  <?php print $user_picture; ?>

  <?php print render($title_prefix); ?>
  <?php if (!$page): ?>
    <h2<?php print $title_attributes; ?><a href="<?php print $node_url; ?>"><?php print
$title; ?></a></h2>
  <?php endif; ?>
  <?php print render($title_suffix); ?>

  <?php if ($display_submitted): ?>
    <div class="submitted">
      <?php print $submitted; ?>
    </div>
  <?php endif; ?>

  <div class="content"<?php print $content_attributes; ?>>
    <?php
      // We hide the comments and links now so that we can render them later.
      hide($content['comments']);
      hide($content['links']);
      print render($content);
    ?>
  </div>

  <?php print render($content['links']); ?>

  <?php print render($content['comments']); ?>

</div>
```

Figur 12. Strukturen på *node.tpl.php*

För Bolag X tema skapades flera olika sidbotten för olika innehållstyper. De olika innehållstyperna är bl.a. *Investments*, *Team member* och *News*. Respektive sidbotten är *node--investments.tpl.php*, *node--teammember.tpl.php* och *node--news.tpl.php*. Något av dessa sidbotten används alltid då man skapar en ny sida, som är av någondera typen.

Till *node.tpl.php* finns också en hel del variabler till förfogande. Dessa är alla listade i standard filen som kommer med basinstallationen av Drupal. Exempel på dessa är *\$content* som skriver ut nodens innehåll, *\$title* som skriver ut nodens rubrik och *\$date* som skriver ut datumet som noden är skapad. (Todd Tomlinson 2010)

4.6.4. Övriga template filer

Region.tpl.php används när det finns innehåll i regionen, antingen via block systemet eller via en funktion. Standard filen hittas i */modules/system/* och är ganska enkel med två variabler.

```
<?php if ($content): ?>
  <div class="<?php print $classes; ?>">
    <?php print $content; ?>
  </div>
<?php endif; ?>
```

Figur 13. *Region.tpl.php*

Man kan skapa olika mallar för olika regioner genom att inkludera regionens namn i filnamnet. Om man har definierat en region i *.info* filen med t.ex. *regions[footer] = Footer*, så kan man skapa en fil med namnet *region--footer.tpl.php*, som styr utseendet för endast footer regionen. Det är namnet mellan hakparenteserna som man skall använda sig av i filens namn. *Region.tpl.php* är till skillnad från de tidigare mall filerna inte något Drupal använder sig av automatiskt.

Det finns många andra template filer som man kan använda sig av, som **block.tpl.php** vilken *block* modulen använder sig av och med vilken man kan ändra på utseendet på

olika block, eller **comment.tpl.php** som *comment* modulen använder sig av för att rendera kommentarer.

De flesta moduler använder sig de facto av olika template filer, men de är alla lätta att känna igen på sin ändelse, .tpl.php. (Todd Tomlinson 2010)

4.6.5. template.php

Template.php är en fil man kan använda för förhandsprocessering av temat. Genom denna fil kan man ändra på variablers data och vad olika funktioner gör. Relativt ofta vill man byta ut den html som någon modul skriver ut, eller lägga till nya klasser till html taggar för att göra det lättare att definiera en stil för något visst element, och det gör man genom template.php.

För Bolag X tema behövdes t.ex. en placeholder i sökfältet. Genom att använda en Drupal funktion som heter *hook_form_alter* kan man lägga till och ändra på värden som matas ut på sidan.

```
1  <?php
2
3  function intera_form_alter(&$$form, &$$form_state, $form_id) {
4      if ($form_id == 'search_block_form') {
5          $form['search_block_form']['#title_display'] = 'invisible';
6          $form['search_block_form']['#attributes']['placeholder'] = t('Search this site..');
7          $form['search_block_form']['#attributes']['value'] = t('Search this site..');
8          $form['search_block_form']['#attributes']['onfocus'] = 'clearText(this)';
9          $form['search_block_form']['#attributes']['onblur'] = 'clearText(this)';
10     }
11 }
```

Figur 14. Funktion för att ändra på sökfältet.

Man byter ut *hook* i funktionens namn till det egna temats namn och kontrollerar ifall det finns ett formulär som med id *search_block_form*. De nya värdena läggs in i en samling (array). (Drupal.org 2011, Todd Tomlinson 2010)

4.7. CSS

Cascading Style Sheets (CSS) är stilmallar som används som på de flesta webbplatser också av Drupal för att beskriva presentationsstilen. Det vanligaste sättet att lägga till en CSS fil till ett Drupal tema är att definiera det i .info filen, genom att skriva *stylesheets[all][] = css/style.css*.

På det här sättet kan man enkelt lägga till en css fil som inkluderas på alla sidor via html.tpl.php och skrivs ut med variabeln *\$scripts*. Nyckeln heter *stylesheets* och inom de första hakparenteserna kan man välja mediatyp. För skild CSS för skrivare (printer) skriver man på följande sätt *stylesheets[print][] = css/print.css*.

Tabell 8. Exempel på olika CSS filer för olika medietyper.

```
;Stilmall för alla typer av media
stylesheets[all][] = css/style.css

;Stilmall för print media
stylesheets[print][] = css/print.css

;Stilmall för skärmar med mindre bredd än 600 pixlar
stylesheets[screen and (maxwidth: 600px)][] = css/screen.css
```

Bra att komma ihåg:

- Det finns hakparenteser mellan *[medietyper]* och *= css/style.css*. Dessa är alltid tomma och betecknar att varje stilmall läggs till en samling (array).
- Ordningen som stilmallarna är i .info filen är den samma som de skrivs ut genom *\$styles* variabeln.
- Man kan välja att placera stilmallarna i temats egna katalog eller i underkataloger inom temats egen katalog. Det rekommenderas dock att man inte går djupare än en underkatalog eftersom det kan ställa till problem för vissa teman. (Drupal.org 2011)

Genom .info filen har man endast möjlighet att lägga till interna stilmallar. För externa stilmallar kan man använda en Drupal funktion som man lägger till i sin template.css fil.

Tabell 9. Funktion för att lägga till en extern CSS fil.

```
<?php
function intera_preprocess_html(&$variables) {
    drupal_add_css('http://fonts.googleapis.com/css?family=Syncopate',
array('type' => 'external', 'media' => 'all'));
}
?>
```

Funktionen är *template_preprocess_html* och för att den ska fungera i det egna temat måste man byta ut *template* till temats namn vilket i detta fall är *intera*.

Ett annat sätt att inkludera externa filer är att helt enkelt lägga till dem med html genom *html.tpl.php* filen. (Drupal.org 2011h)

4.8. Javascript

Det finns flera tekniker för att lägga till javascript i Drupal. Om man har ett internt skript är den enklaste metoden att göra det via .info filen med nyckeln *scripts[]*. Då man lägger till skript via .info filen kommer de att skrivas ut via *\$scripts* variabeln i *html.tpl.php*. Med den här metoden har Drupal också möjlighet att optimera skriptet, för att minska på tiden det tar att ladda en sida.

Tabell 10. Metod för att lägga till skript.

```
scripts[] = js/script.js
```

För att lägga till externa javascript finns det en funktion som heter `drupal_add_js`. Den ska man använda i sin `template.php` fil. På Bolag X sida kommer det att finnas en Google karta som behöver ett externt javascript för att fungera. Metoden för att inkludera detta skript är följande.

Tabell 11. Funktion för att lägga till externa skript.

```
<?php
function intera_preprocess_html(&$vars, $hook) {
  drupal_add_js('http://maps.googleapis.com/maps/api/js?sensor=true&libraries=places', 'external');
}
?>
```

På samma sätt som man lade till externa CSS filer så använder man sig här också av funktionen `template_preprocess_html` och byter ut `template` mot det egna temats namn.

Det går även att lägga till Javascript genom att inkludera den direkt i `html.tpl.php`, men detta är inte "the Drupal way" och hindrar Drupal från att optimera skriptet.

Då man vill använda Javascript inne på sidan utan att länka det från ett annat dokument, t.ex. i `page.tpl.php`, finns det Drupal specifika metoder. Dessa metoder är till för att hindra att de egna skripten är i konflikt med Drupals skript. I praktiken går det ut på att lägga till ett extra hölje (closure) i form av parenteser, som ser till att man inte skriver över variabler i Drupal.

Tabell 12. Exempel på hur Javascript ska skrivas på sidan. (Drupal.org 2011)

```
// Define a new function.
(function () {
  // Variables defined in here will not affect the global scope.
  var window = "Whoops, at least I only broke my code.";
  console.log(window);
  // The extra set of parenthesis here says run the function we just defined.
})();
// Our wacky code inside the closure doesn't affect everyone else.
console.log(window);
```


Det samma gäller också då man ska använda sig av JQuery, som är ett Javascript bibliotek, som används flitigt av Drupal. För att JQuery skript överhuvudtaget ska fungera i Drupal måste man packa in skriptet i extra parenteser och i slutet av funktionen skriva vilket Javascript bibliotek det handlar om. Detta är för att förhindra konflikter med andra Javascript bibliotek såsom Prototype eller MooTools. (Drupal.org 2011i, 2011j)

Tabell 13. Exempel på hur man ska använda JQuery.

```
(function ($) {  
  // jQuery kod här..  
})(jQuery);
```

5. DISKUSSION

Den slutliga produkten blev en webbplats som personalen på Bolag X lätt kan uppdatera. De kan lägga till nyheter, personal och enkelt ändra på texter på olika sidor utan kunskap i vare sig HTML eller CSS. Med hjälp av olika sidbotten ser de olika sidorna enhetliga ut. Webbplatsen är redan publicerad på adressen www.interapartners.fi och personalen på Bolag X har redan gjort sina första nyhetsuppdateringar.



Figur 14. Framsidan för webbplatsen.

Temat har inte krävt några avancerade omprogrammeringar, men utnyttjar ändå en stor del av de olika delområden som är bra att känna till då man bygger upp ett tema för Drupal, med allt från sidbotten, till utnyttjandet av variabler till små förhandsprocesser som ändrar på datan som matas ut.

Webbplatsen är för en vanlig besökare en statisk sida, men innehållshanteringssystemet utnyttjas för att göra sidan enkel att uppdatera. Att göra ett tema för en statisk sida är i vanliga fall enklare än för en interaktiv sida var besökare själva kan bidra med innehåll, eftersom man då inte behöver tänka på att besökaren inte kanske vet exakt vad den håller på med.

Bolag X tema går inte heller att jämföra med de teman som man hittar på Drupals egen webbplats. De teman som finns på Drupals webbplats är ämnade för att i stort sett vem som helst ska kunna använda dem, och de brukar inte använda sig av olika sidbottnar för olika innehållstyper. Användningen av olika sidbottnar för olika innehållstyper gör att man minskar på flexibiliteten, men å andra sidan är det mindre som webbplatsens upprätthållare kan stöka till utan att gå direkt in i temats filer och göra ändringar.

5.1. Drupal

Ett av målen med detta examensarbete var att få fördjupade kunskaper i hur teman fungerar i Drupal, och hur man bygger upp dem. Drupal är inte ett system som man behärskar på en dag utan det tar tid att förstå, dels hur administrations sidorna fungerar, och dels hur ett tema byggs upp. Flexibiliteten som är en av Drupals starkaste sidor är också en stor fallgrop för många, eftersom det medför att källkoden är väldigt komplex och inte helt enkel att förstå. Man kan dock klara sig ganska långt om man har förståelse för hur de olika template filerna fungerar och har baskunskap i HTML och CSS.

Då man börjar förstå sig på hur template filer fungerar kan man enkelt utnyttja dessa för att få det utseende man önskar. Då många av Drupals template filer dessutom innehåller kommentarer med en lista på olika variabler som finns till förfogande klarar man sig ganska långt utan att vara en expert på programmering eller ha någon djupare kunskap i Drupals funktioner.

KÄLLOR / REFERENCES

David Mercer. 2010. Drupal 7. Packt Publishing
ISBN 978-1-849512-86-2

Ric Shreves. 2011. Drupal 7 Themes. Packt Publishing
ISBN 978-1-849512-76-3

Todd Tomlinson. 2010. Pro Drupal 7 Development. Apress
ISBN-13 (electronic): 978-1-4302-2839-4

About Drupal | drupal.org 2011a, [www],
Hämtat 22.11.2011 <http://drupal.org/about>

History | drupal.org 2011b, [www],
Hämtat 22.11.2011 <http://drupal.org/about/history>

System Requirements | drupal.org 2011c, [www],
Hämtat 23.11.2011 <http://drupal.org/requirements>

Quick install for beginners | drupal.org 2011d, [www]
Hämtat 11.12.2011 <http://drupal.org/documentation/install/beginners>

Structure of the .info file | drupal.org 2011e, [www],
Hämtat 2.12.2011 <http://drupal.org/node/171205>

Theme settings | drupal.org 2011f, [www],
Hämtat 2.12.2011 <http://drupal.org/node/221905>

Overview of theme files | drupal.org 2011g, [www],
Hämtat 2.12.2011 <http://drupal.org/node/171194>

Adding style sheets | drupal.org 2011h, [www],
Hämtat 4.12.2011 <http://drupal.org/node/171209>

Managing JavaScript in Drupal 7 | drupal.org 2011i, [www],
Hämtat 4.12.2011 <http://drupal.org/node/756722>

JavaScript and jQuery | drupal.org 2011j, [www],
Hämtat 4.12.2011 <http://drupal.org/node/171213>

About nodes | drupal.org 2011k, [www],
Hämtat 20.12.2011 <http://drupal.org/documentation/modules/node>

Innehållshanteringssystem - Wikipedia 2011a, [www],
Hämtat 11.12.2011 <http://sv.wikipedia.org/wiki/Innehållshanteringssystem>

Drupal - Wikipedia 2011b, [www],
Hämtat 18.11.2011 <http://sv.wikipedia.org/wiki/Drupal>

Template engine (web - Wikipedia, the free encyclopedia 2011c, [www],
Hämtat 20.11.2011 [http://en.wikipedia.org/wiki/Template_engine_\(web](http://en.wikipedia.org/wiki/Template_engine_(web)

Usage Statistics and Market Share of Content Management Systems for Websites, December 2011, [www], Hämtat 18.11.2011
http://w3techs.com/technologies/overview/content_management/all

A peek at Drupal 7 theme system changes | PINGV Creative Blog 2011, [www],
Hämtat 2.12.2011 <http://pingv.com/blog/a-peek-at-drupal-7-theme-system-changes>

What is Firebug? : Firebug 2011, [www],
Hämtat 4.12.2011 <http://getfirebug.com/whatisfirebug>