

Drupal 8 Modules: Translation Management Tool and Paragraphs

Saša Nikolić

Faculty of Mathematics, Science and Information Technologies, University of Primorska
Glagoljaška 8, SI-6000 Koper, Slovenia
E-mail: nikolic.sasa09@gmail.com

Jurij Šilc

Computer Systems Department, Jožef Stefan Institute
Jamova cesta 39, SI-1000 Ljubljana, Slovenia
E-mail: jurij.silc@ijs.si

Technical paper

Keywords: content management system, Internet, website, open source, module, Translation Management Tool, Paragraphs

Received: September 11, 2015

As the Web has grown in the last few decades, we now have nearly one billion websites online and most of them offer rich information, that is usually difficult to manage by normal users. In order to simplify the process of creating and managing website content with relative ease and with an user friendly experience, lots of content management systems were developed. These are software applications that provide capabilities for multiple users with different permission levels to seamlessly create, edit, review and publish website content. They offer a web-based graphical user interface, enabling publishers to access the content management systems online using only a web browser. Because of good security, quality, customisability and great support by the community of developers, open source content management systems are becoming extremely popular and many of them available on the market. The most noticeable are Drupal, Joomla, Magento and Wordpress. A lot of different surveys have been done to determine which one of them is the best, but this question still remained unanswered. Because of our personal involvement into developing Drupal and helping the community, this article describes a totally new version of Drupal, named Drupal 8. Firstly, a short overview about Drupal and all its key parts is presented, followed by a chapter describing all the new features and the current status. Two subtopics of this article will include modules that we lately contributed to – Translation Management Tool and Paragraphs. Some of our main issues are described at the end of each module.

Povzetek: V zadnjem desetletju smo priča bliskovitemu napredku spletnih tehnologij in posledično spletnih strani ter aplikacij. Te so danes polne bogatih informacij, a jih je navadno težje upravljati. Za poenostavitev kreiranja in upravljanja s podatki na spletnih straneh so bili razviti različni sistemi za upravljanje vsebin. Danes jih je na tržišču veliko, med njimi so najbolj opazni Drupal, Joomla, Magento in Wordpress. V sestavku je govor o novi različici odprtokodnega sistema za upravljanje vsebin Drupal 8, pri razvoju katerega smo sodelovali. Najprej je na splošno predstavljen Drupal z glavnimi lastnostmi in funkcijami, nato sledi poglavje Drupal 8, kjer so opisane novosti in posodobitve. Po kratkem poglavju o trenutnem stanju sistema ter prihodnjih korakih sledi poglavje o dveh modulih, pri razvoju katerih smo sodelovali: Translation Management Tool, za lažje prevajanje vsebin, in Paragraphs, za boljše strukturiranje vsebin. Nekaj naših rešitev je opisanih na koncu vsakega od obeh modulov.

1 Introduction

The Internet is probably one of the most profound achievements in human history. We became so addicted to it, that we hardly even notice it, unless it happens to be unavailable. With the help of great innovative technologies, the internet has simply dropped the barriers of time and geographical distance to turn the entire world into a local community centre. In the last decade millions of people tend

to share their lives and experiences with others through their personal blog sites. Others use the web to show off their work, art or music. Still others found an opportunity to promote their companies and be more noticeable to a wider variety of people. Whatever the need is, there exists a great solution that for its installation and administration does not require any programming skills. Drupal is one of the most common solutions for building anything from simple user blogs to fully-customizable, interactive

and mobile-responsive websites in several languages. Assuming that you are interested in the open source community and in learning more about Drupal, this article briefly describes what Drupal is and mainly, introduces you to the new version, Drupal 8. Last, but not least, it describes two new modules, Translation Management System and Paragraphs, their structure and my personal contribution to in their development. We have been an active Drupal member for more than 7 months now and contributed to more than 80 various issues, from core to contributed modules.

2 Drupal

2.1 What is Drupal?

Drupal was developed and released in 2001 under the open GNU General Public Licence (GPL), which means anyone is free to download it and share it with others. It is a PHP and MySQL-based system for managing websites and is used by hundreds of thousands of web developers around the world [3]. It serves as a back-end framework for more than 2% of all websites worldwide – from personal blogs to corporate, political and government sites, including the official website of the White House and various UK Government projects [2].

The default release of Drupal, known as Drupal core, contains only basic features, like user account registration and maintenance, menu management, taxonomy, page layout customization and system administration. This is enough for a simple website, a user blog or an Internet forum.

Currently, there are more than 30,000 free community-contributed add-ons, also known as contributed modules [4]. By adding different modules and features like advanced search, content translation, external text editors (WYSIWIG), many different jQuery libraries, etc. users can experience the web in totally new and different ways.

2.2 Why choose Drupal?

Here are some of the reasons why people may opt to use Drupal: Drupal is a CMS which allows users to update their websites without technical knowledge and ensures that it fits any organization's workflow. It is modular, extensible and scalable, which means it can grow over time as user's needs expands. Also, the website can be customized depending on the content or company features. Drupal can be installed in multiple languages, allowing users and administrators to view and administer the site in their own language. From the hardware point of view, it runs on any computing platform that supports a web server capable of running PHP (e.g., with Apache, Nginx, LiteSpeed) and a database to store content and configurations.

2.3 Core features

Administer: Drupal comes with various options for user accounts and permissions. The administrator can set up one or more roles to users, specifying with different permissions, what each user can and can not do.

Build: easily build websites without any programming knowledge. It also comes with pre-defined configuration, so that website building is much easier than before.

Collaborate: social publishing and interaction with the content on your site can be easily controlled by the administrator.

Connect: using aggregation, feeds, search engine connection capabilities and social media integration is widely supported to help users connect with wider audience.

Creative Content: Drupal's flexibility supports many content types including video, text, blog, podcasts and polls with an user-friendly web interface.

Design & Display: there are lots of themes created by professionals and free to use, since Drupals presentation layer allows designers to create their own interactive experiences that engage users and increase traffic.

Extend with more than 16,000 available modules developers can create and adapt the site to any requirements. Everyone is encouraged to contribute modules for others to use.

Organize & Find: many tools are available for organizing, structuring, finding and re-using website's content. Friendly path urls, custom lists, categorization with taxonomy and linking content with other content on the site are just some of the options.

2.4 Community

The main reason, why Drupal is more popular and secure than other open source CMS is a huge developer community. It counts more than 1 million members and provides support, constant development (and bug fixing), testing and documentation. Drupal community members make Drupal better and better every day. More than 950 people contributed code and ideas to the Drupal 7 release and even more are responsible for developing and maintaining the so-called "contrib" modules. The main website that provides a place for groups to organize, meet and work on various projects is <https://groups.drupal.org>. It is mainly based on geographic location and interest. This is a great way to get fast support, learn more by local people and to easily get involved. Drupal events and meet-ups are also very frequent, which makes it easier to exchange

knowledge face to face, get ideas for new projects and making friends along the way. IRC is another fast and effective way of communication and interaction with other developers, mainly in use for support. Forums, mailing lists and social media are also available for sharing information about Drupal.

3 Drupal 8

Since its creation in 2001, Drupal has grown and developed year by year to meet new changing demands and needs of all its global users and to achieve that, new big, forward-looking changes needed to be made. The result is that Drupal has stayed relevant to new technologies, unlike nearly every other Open Source CMS over the years. The downside is that with every major release, Drupal developers have gone through a lot of pain adjusting to these changes. In 2006, the founder of Drupal – Dries Buytaert wrote: “*So let’s capture that thought for future reference. Sweeping changes are required to make major advances in technology, and often times there is a lot of pain before the payoff.*” [1]

Drupal 7 is a very popular CMS amongst users, but there are quite some big limitations, including incomplete Entity API, no separation between logic and presentation in the theme layer, and so on. Contributed modules tried to solve many of these problems, but they were mostly incomplete. With Drupal 8, these problems were solved with a head-on approach – through the Configuration Management Initiative, Twig templating layer and a new, complete Entity API.

With more than 200 new features and improvements, will definitely be the most significant update in Drupal history. Easier customizations of data structures, listings and pages will be the first thing that the users will notice at the beginning. Countless new capabilities for displaying data on mobile devices, building APIs and adapting the website to multilingual needs are also some of the other things that will make a huge impact on the usability and diversity. With a much more efficient core, easier migration from earlier versions and inline content editing tools it will become a cutting-edge platform that will set new standards for other CMS. To not forget various new modules and themes, made available by a modern Object Oriented Programming (OOP) approach on the back end side. All those new features can be summed up in different categories, but the categorisation based on what affects different types of users seems to be the most important one [7].

4 Drupal 8 development

One of the biggest challenges with Drupal, is that it is hard for organizations of all sizes to find Drupal talent (developers, themers, site builders, etc). Drupal 7 didn’t address this problem (e.g., using procedural programming instead of object-oriented programming), and in fact made it a bit worse with the introduction of even more Drupal-specific

development (e.g., excessive use of structured arrays). For most people new to Drupal, Drupal 7 could be really complex. The most effective way to address the Drupal talent issue, as well as the complexity issue, is to update Drupal with modern frameworks and platforms, so there is less Drupal-specific knowledge to learn in order to become proficient. Modern PHP concepts and standards, object-oriented programming, and the Symfony framework were adopted for that matter. While a lot of the Drupal concepts (Fields, Views, Entities, Nodes) continue to exist in Drupal 8, they are now implemented using object-oriented programming design patterns.

The advantages and disadvantages of object-oriented programming are well-known. The disadvantages are verbosity, size, slower performance and the amount of work it takes to write (including the design planning that goes into it). For people that are new to object-oriented programming the learning curve could be steep; some of the key programming techniques, such as inheritance and polymorphism, can be challenging initially. The advantages are encapsulation (both to avoid tampering with internal values and to hide implementation details), faster development thanks to re-use, extensibility, and better maintainability. Compared to procedural programs, object-oriented programs are easier to maintain, extend and refactor. So although a lot of work is spent to write the code, less work is needed to maintain it over time. For Drupal 8 this means that the code will be more abstract, more verbose, and slower, but also more maintainable, more modular, and more accessible to non-Drupal developers. The end goal is that Drupal 8 should help attract new people to Drupal in a way Drupal 7 did not. As an example, exactly the same happened with other projects like Symphony. Symphony 2 was a complete refactor and re-architecture from the previous version. People had different opinions about that. A lot of people were alienated, yet at the same time Symphony 2 was a big success.

The same thing has happened with the major releases of Drupal as well, despite how much change each one brings. Many of Drupal 8 development changes are described below based on the users.

4.1 End users and clients

As mentioned, Drupal 8 is a powerful platform that requires very little to no technical knowledge. It’s purpose is to let content administrators use the website as they want; “getting the right content to the right people in the right language at the right time”.

Comparing Drupal to its competitors, the leading Drupal-based company has set some priorities to fill in the most important gaps. The biggest shortcoming in Drupal was set to be the authoring experience. Drupal 8 expands previous functionality by allowing users to do the following:

- Easily create lists and image galleries using views.

- Using WYSIWIG editor in core to create well-structured pages and still having the option to see the rich text format.
- Inline editing is possible by default. It works with image fields, taxonomy, files, regular text, formatted text, etc.
- Use two column layout, which makes it easier to separate the essentials on the left and meta/admin data on the right.
- Preview the newly created content on the front end.

Since the mobile technology evolved so much lately, responsive designs are a must. Drupal 8 supports adding and editing content from any mobile device and comes with all built-in themes that are fully responsive. Since mobile was also a big priority, everything from the installer to the modules page has been re-designed with mobile in mind. There were made lots of big improvements:

- Responsive core themes.
- It comes with picture module that uses HTML5 picture element for responsive images.
- It is based on configurable breakpoints - courtesy of the new Breakpoint module.
- The new toolbar is mobile friendly with large tap areas, and a nice vertical sidebar or app icons at the top.
- Responsive tables make sure that the most important columns are also displayed on smaller screens.

4.2 Site builders

All new out-of-the-box features like views, configuration management and an improved user interface (UI) make the job of site builders a lot easier. Drupal 8 allows site builders to:

- Attach new field types (like Entity Reference) to different new types of content.
- Customize the look and feel of data entry forms on Form displays.
- Use views and customize the default front page, admin listings, sidebar content, image galleries, slide shows, etc. with 0 lines of code.
- Have a totally new administration experience, since the interface got lots of attention. The elements are redesigned for responsiveness, consistency and better accessibility.

4.3 Designers and themers

The most noticeable update regarding the UI is the usage of HTML5 markup for all themes and core components. The other most talked-about change is Twig, a templating engine that makes Drupal theming totally different in comparison with the older versions where template files were used for the HTML markup and PHP variables. Twig is a lot faster, more secure and makes the Drupal 8 markup a lot cleaner, while the admin UI provides a more consistent experience and is easier to use. It is also much easier to debug and view where information is coming from. It can be also seen as a development module for themers. Other than that, the front end libraries are updated. Drupal 8 offers the latest versions of jQuery, jQuery UI and Backbone, which is used as a front-end library and as in core for data modeling and state syncing.

4.4 Developers

The main goal was to make Drupal 8 developers do more, with fewer steps and with less knowledge about the framework itself. It was achieved with a totally new file-based configuration management system, which is a mix of some of the most modern and popular web technologies that we can use today; RESTful web services and the use of Symfony 2 framework. Developers can also benefit by the following updates:

- It uses the new Rest and serialization API, which means serialized data can be outputted as JSON and XML.
- New libraries: Assetic, Composer, PHPUnit, Guzzle, Zend Feed Component.
- Easily track changes in configurations with version control and update the production site cleanly.

5 Current Drupal 8 status and future releases

The Drupal community was working hard since March, 2011 on Drupal 8 and making progress constantly. The first ever stable version of Drupal 8 was released on 19th November 2015, followed by release parties all around the globe, organized by contributors themselves. This key milestone was achieved with the work of more than 2,300 people altogether. There have been more than 11,500 committed patches to 15 alpha releases. Semantic versioning will follow after Drupal 8.0.0 with a regular release schedule. Patch-level releases will follow monthly corresponding to Drupal 7's release windows. Minor releases are planned to follow every six months. The corresponding Drupal timeline is displayed in the Figure 1.

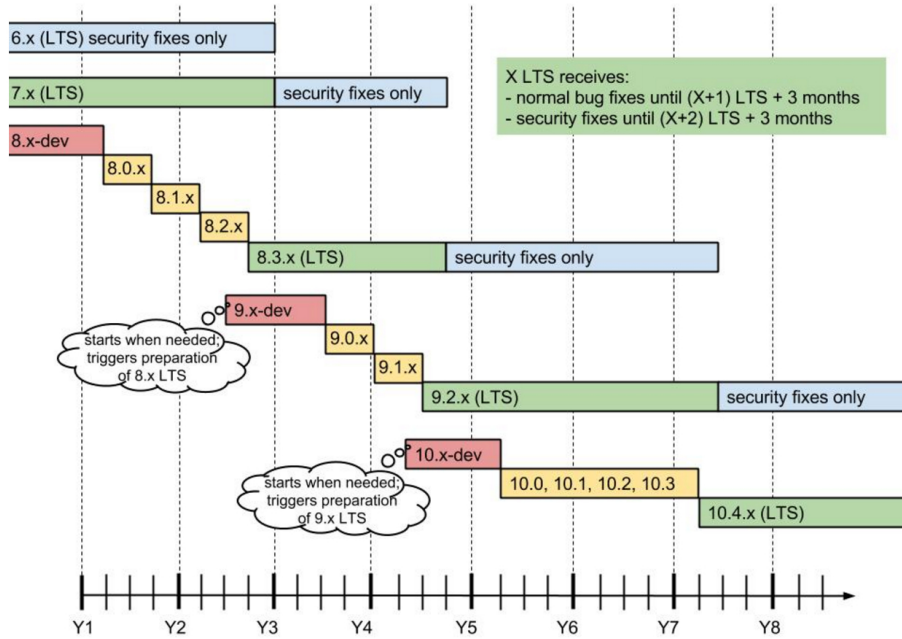


Figure 1: Drupal timeline plan for specific version releases. As seen, the plan is well defined, although there can be many variations.

6 Modules

Users can extend and customize Drupal functionality with contributed modules. If a module does not quite do what is supposed to do or if there are any bugs, everyone is invited to help the module maintainer and report and, if possible, also fix the opened issue. I am lucky to be part of the Drupal community and lately we were contributing to various modules, but our main focus is on Translation Management Tool and Paragraphs, which will be shortly described below.

6.1 Translation management tool

The idea behind Translation Management Tool (TMGMT) module grew-up in 2011. The plan was to build a “contrib” extension to support editors, publishers, translators and project managers during their process of content translation. It uses existing language tools and data structures. The purpose is to solve all the confusion and problems that were arising while doing translation in Drupal. Let’s say there were 100 “nodes” on the page and we need to translate it to 5 different languages To do this, we would end up with 500 nodes, which all contain the same content, but in different languages. To maintain all the nodes and translations was a real struggle and it was impossible to see and manage the status of the translations. Also, there was no workflow - external services were not supported and the translator had to log into the site configuration to do his job. All this ended up in a big mess. With Translation Management Tool most of these problems are solved and the

translation with Drupal is streamlined and user-friendly.

The architecture of the module is simple. It allows translation of any kind of text elements, from content, configuration and interface texts in just a few clicks. In TMGMT these sources are named source plugins and are added to the translation job, as seen in Figure 2. Each of the source plugins in a job is called job item. The translation of the job can be done by local or remote translators (also called Translation Plugins) of different kinds and the translation process can be totally automated. This means that external services can be used for creating a foreign language version of the source, but also the user himself can translate the text via the Local Translator and save it. Also automated translators are available. The module is based on a plugin architecture, that allows additional sources and translation services to be added by everyone [8].

The following services are part of the module:

- Local Translator gives the users the ability to translate the source on their own.
- File export and import via XLIFF and HTML.
- Gengo (human) as a remote translation service provider.
- Microsoft and Google Translate (machine) use their machine translation APIs.

With the installation of TMGMT the functionality of the page is extended. The user can then choose one or more languages to translate the node to and request a translation with the corresponding button. For each of the language chosen, a translation job is created. The user can

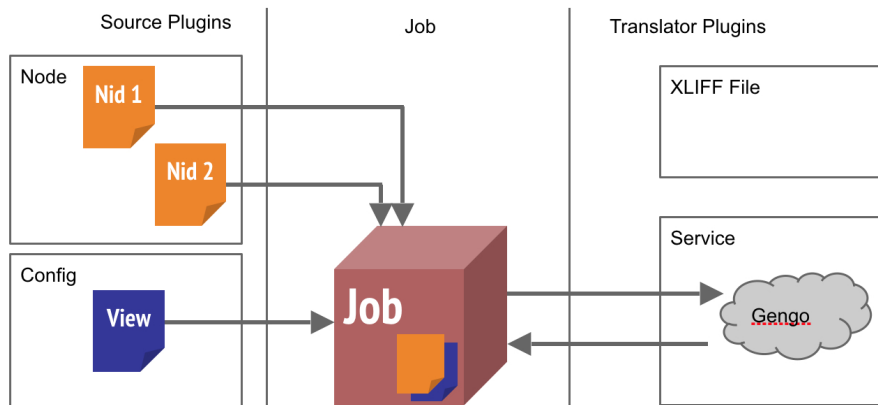


Figure 2: TMGMT architecture is made out of three basic parts; Source Plugins, Jobs (as TMGMT core functionality) and Translator Plugins.

then request a translation from the list of enabled translators. After getting the translation back, the job state can be processed, unprocessed, active, reviewed and finished. Translation overview offers a quick look at the list of all jobs with all the relevant information.

Our contribution for TMGMT is very vast. We got involved into this module at a very early phase, so our main points were discussing and implementing new functionalities, discovering and fixing bugs, extending web tests and reviewing other contributor's work – patches. We have also done a significant part of the improvements for the interface to reduce complexity and enhance the user experience.

One of our main topics was definitely the implementation of Gengo translator. It is a translator plugin for the TMGMT project, this means it extends its functionality by allowing the submission of translation jobs to Gengo, which is an external translation service provider [5]. Because this was a port from an older version, lots of code modifications needed to be made, such as syntax changes, different remote mappings, GET, POST and PUT requests, etc. Running the tests and checking the test errors helped me with the tasks mentioned above.

Another important issue was regarding the stability of the module when deleting a translator with active translation jobs or translation items assigned. This issue involved extending the code by creating a new method called *hasTranslator()*. This simplifies the process of checking if a translator has a *target_id* and a plugin assigned. Also the translator class is simplified by adding a *hasPlugin()* method. This resulted in a lot of code refactoring through the whole module. *TranslatorTest* was extended to cover new possible cases for deleting a translator with jobs and job items, for example, for the jobs that are in the finished state - should be deleted, and active state – should not be deleted. As an addition, a success message was added after a successful removal of a translator.

6.2 Paragraphs

The Paragraphs module is a fairly new addition to the Drupal project. When it comes to content creation, it offers a rapid and straightforward path towards improving the quality of the website and user experience. It allows site builders to make things clearer for the end users and to give them more editing power while still have more control over their misbehavior. The old way was to put all the content in one WYSIWYG body field including images and videos. This approach had many issues:

- Inability to add rich content (galleries, accordions, parallax backgrounds etc.).
- Inadequate markup and/or undesirable inline styles.
- Users confusion.
- Bad-designed content (web pages).

With Paragraphs, the content is much more structured and easier to use. Users can choose from pre-defined paragraph types, which are independent from one another.

As a site builder, you can add an unlimited number of paragraph types to the site - each with its own fields and displays. In easy words, imagine Paragraph types as mini-content types which can be created on nodes. Since they are basic Drupal entities, it is easy to see what fields should they be composed of and how they should look like. So the biggest advantage of Paragraphs is the ability for a content editor to have total control over the flow of the content using drag-and-drop sorting.

For reference, part of my contribution to this module was providing a demo module as an example for new users which contains four different default paragraph types; a simple text field, a text with an image, an image gallery and a user, which outputs user information. These are all styled in the CSS files, so that the whole page with paragraphs is responsive on mobiles and tablets [6].

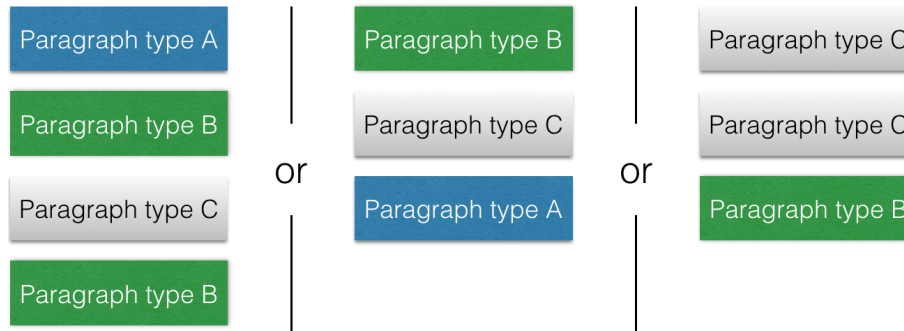


Figure 3: Options with combining paragraph types are endless. The order is easy to set with drag-and-drop functionality.

In many ways, Paragraphs module can be compared with Field Collection module, since it offers similar functionality. With Field Collection, a site builder creates a set of grouped fields and an editor can then add as many of those collections to a node, one after another. The problem here is that they all must be of the same type and there can not be any other type of content in between them. On the other side, with Paragraphs an editor can use more types together, in any order, thereby creating a flow of content without restriction. A simple example is presented in Figure 3.

Although not being “out in the wild” for a long time, this module became very popular among the users and is already being used on many web sites. One of them is also drupal.org (Easy Content Authoring page), which can be seen in Figure 4.

We mainly contributed to this module with implementing Paragraphs translation with TMGMT, so that users can easily translate structured data. This helps translators a lot, because they have the source in smaller bits, which are always easier to translate than bigger unstructured chunks of texts. For better history control, revisions are implemented so that users can easily spot the differences. For this we needed to extend TMGMT to depend on Paragraphs and add TMGMT specific configuration. Paragraphs needed to support translation on entity level. This was really challenging. In regular situations it is not the entity reference that should be set to translatable, because the wrapping field should maintain the same set of translated paragraphs. While translating some content, all paragraphs from the default language have to still be there and the user should have the ability to translate them into the target language. On save, the paragraph entities get updated with the proper language context to persist the translation.

7 Conclusions

The article describes general information about Drupal, and mainly talks about some of the most important new features in Drupal 8. In the end, two of the modules that we lately worked on are presented with a short description of our main contribution. With a combination of Drupal 8 core features, Paragraphs and Translation Manage-

ment Tool modules we can build a powerful multilingual website that is very easy to manage, since the content is structured with paragraphs and good looking, because of the flexibility of styling each item separately. And thanks to Translation Management Tool content translation is easier but at the same time more powerful and extendable than ever in just a few clicks. The Drupal 8 is following latest technology trends and with this new release it should attract even more site-builders and end-users. After all, the Drupal community has been working hard to achieve all defined goals for years.

Acknowledgment

I (S. N.) would like to thank my mentor, Assist. prof. dr. Jurij Šilc, for his support and MD Systems, as a leading contributor in open source, for initiating TMGMT, introducing and mentoring into Drupal, maintaining modules like Paragraphs and porting them to Drupal 8.

References

- [1] Buytaert, Dries (2006), The pain before he pay-off, personal blog. <http://buytaert.net/the-pain-before-the-payoff>, accessed January 14, 2016.
- [2] Buytaert, Dries (2010), The State of Drupal. DrupalCon, April 19–21, 2010, San Francisco, CA. <https://archive.org/details/Css3TheFutureIsNow>, accessed January 14, 2016.
- [3] Coombs, Karen (2009) Drupal Done Right, *Library Journal*, vol. 34, no. 19, pp. 30–32.
- [4] Drupal homepage. <https://www.drupal.org/>, accessed July 2 2015.
- [5] Professional Translation Services by Gengo. <http://gengo.com>, accessed January 14, 2016.

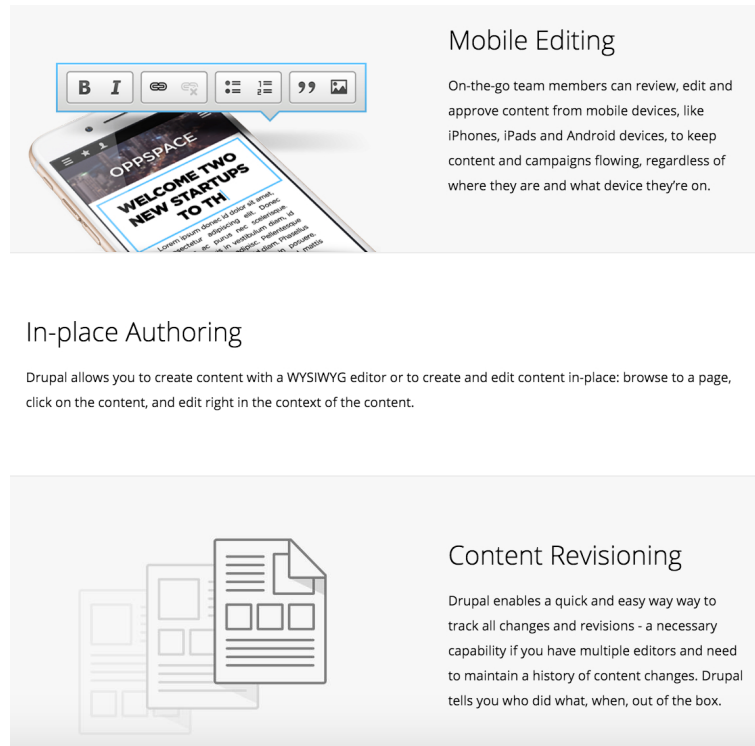


Figure 4: Paragraphs are already used on drupal.org. Images and text can be floated left or right, full width or styled as the user prefers.

- [6] Paragraphs module on Drupal.org. <https://www.drupal.org/project/paragraphs>, accessed January 14, 2016.
- [7] Patel, Savan K., Rathod, V. R., Prajapati, Jinga B. (2011) Performance Analysis of Content Management Systems - Joomla, Drupal and Wordpress, *International Journal of Computer Applications*, vol. 21, no. 4, pp. 39–43.
- [8] Translation Management Tool module on Drupal.org. <http://www.drupal.org/project/tmgmt>, accessed January 14, 2016.