

OOP in PHP Drupal CMS

14/04/2008 20:26 by Sally Ahmed

Object Oriented Programming (OOP) in PHP

As most programming languages PHP supports Object Oriented Programming (OOP) through the use of classes which are the basis that forms objects that facilitate code reuse. The development of PHP was not planned with OOP, hence the fact that PHP was based on C which is a non-OOP language. Many PHP developers would continue for years without having to work with objects. Nevertheless, PHP is capable of providing modular code through objects and functions that allow large applications to run faster and benefit from implementations that support smooth upgrades.

The OOP characteristics of PHP are present in many of the Commercial off-the-shelf applications (COTS) that are made available in the market today; from Content Management Systems (CMS) and discussion forums, to Client Relationship Management (CRM) and Enterprise Resource Planning (ERP).

Drupal (www.drupal.org) is a content management system that was popularized with its wide range of modules and by the fact that it is distributed as an open source product under the General Public License (GPL). Drupal was based on PHP 4 which meant that many of the advanced features of OOP were not fully available for Drupal. New versions got improved with the enhancements that were deployed in PHP5. In spite of that, Drupal is one of few applications out there that might seem completely procedural, yet implements most if not all of the concepts of OOP.

Drupal makes use of components addressed as "Nodes"; that represent the various parts of the application such as pages, user objects, themes and plug-ins. Nodes use functions such as `node_invoke()` to call objects that provides operations on Nodes. The modular functions also determine what and who can perform actions on the node, which is predefined as the actions of viewing, deleting, updating or creation. A very intrinsic element in Drupal is it being built as a set of application programming interfaces APIs, which are defined in the form of modules. Modules in Drupal are components that can be thought of as plug-ins, and by themselves are object-like. Each module has a file associated with it, and incorporates its own set of variables and functions. Modules are also the main way to extend the use of Drupal and benefit from added functionality. Their interoperability is further extended by their compatibility with other modules and integration with different Drupal versions. Furthermore, modules can have additional behaviors comprised of other modules to be added extending the functionality of that particular module.

In terms of abstraction, Drupal uses a database abstraction layer, separating the integration with the database through a set of functions that allow developers to work with different databases without having to worry about SQL incompatibilities. These are taken care of by a set of functions such as `db_query`, `db_query_range`, `pager_query`, and `tablesort_sql`. Within the same concept, a Hook system is used within Drupal to build a framework that facilitates several operations for modules similar to the functionality provided to software applications by operating systems.

[-pagebreak-]

In terms of isolation, Drupal uses a naming convention to limit calling internal functions outside their object-like modules. This convention is maintained by the developers and thus is not as strict as the encapsulation provided by objects on their variables. Nodes in Drupal are polymorphic accepting different input and producing the desired output based on the internal construction of the item being passed on, for example, themes are displayed with a call to `node_view()` in spite of the fact that each theme might incorporate a different set of menus, site structure, and styling convention.

Furthermore, functions can redefine other functions within Drupal, a theme for example can benefit from the standard `theme.inc` file that has the main functions for Drupal interface so if the function is not overwritten the default values are rendered. Developers creating new modules or enhancing current ones need only identify the functions that are their and create their own versions using the same naming convention. In addition to all the above, Drupal provides well defined lines that separates structure and design. In programming, application framework structures are created to

handle responsibilities and assign tasks to the right entity. Issues arise when internal program work flow is not setup correctly to handle elements of the design regardless of that of the application structure. With Drupal, nodes provide the utmost abstraction and define proper set of responsibilities to avoid what others applications usually fall into.

My Conclusion OOP is a concept, in my choice of Drupal as an example of a framework that implements object oriented programming, I wanted to emphasize on the concept of OOP more than the implementation. From what has been discussed above, Drupal has more functions and procedure than it has objects and classes, however, all aspects of a well defined OOP application has been constructed using what was available to the developers at the time when the application was created. OOP is a way of thinking, even if the language is not equipped with a preset of objects and classes, developers can build modules and functions that allow interoperability, inheritance, and encapsulation. I could be wrong, but even a markup language like HTML allows for some modularity in its structure to leave room for reuse, inheritance, and interoperability. I am not saying HTML is an OOP language, I know many does not even consider it as a language, however, I think the characteristics of modular development and object orientation can be incorporated as part of the implementation. A very important element in Drupal is its expandability and its mind set anti-backward compatibility policy which generally affects many of the applications available in the market and results in many becoming too rigid or unexpectedly break and become fragile.

Drupal, as of this writing, has released version 7 which benefits from the advanced object oriented elements of PHP 5. We should expect a drastic change in the functional structure of Drupal by incorporating more classes and objects to simplify some of what has been master minded in earlier versions without actual OOP tools .

Drupal has a much cleaner core design and a very good API and theme engine. It impresses every developer with the capabilities it has in terms of organization and modularity. If there is one thing Drupal proves it would be that not every program using classes is an OO program, and not every procedural function means spaghetti code.