

# The Web Information system for the Italian National Institute for Astrophysics: the experience by using the Zope/Plone framework

Serena Pastore, Caterina Boccato  
INAF-Astronomical Observatory of Padova  
National Institute of Astrophysics  
Vicolo Osservatorio 5 – 35122 - PADOVA  
ITALY

*Abstract:* - The complexity of modern web information systems is steadily increasing; the provision of platform where the flow of information may be easily managed and controlled by actors (creators, editors and reviewers of content) and where additional techniques may improve collaboration and online information sharing is a key requirement for modern web system. The paper describes the design and implementation of a web information system for the INAF research institution that, in the same multi-tiered framework that keeps content, presentation and logic separate, joins both the requirement of a flexible and multiuser publication tool for web actors and users demand on additional services. The software solution adopted is mainly based on Plone open source software, an application of the Zope web application server that, together with others products, allows to create an integrated environment where information process is highly supported. We describe specifically the membership and workflow systems of the framework that help to design the system information flow and control the quality of information given to site visitors. Furthermore we analyze the techniques used to provide additional services (i.e. news and searching systems, web feeds) that, added to satisfy user needs, may enhance every portal.

*Key-Words:* - Content Management System, Three-tier architecture, web application server, membership and workflow system, content lifecycle, web feeds

## 1 Introduction

The complexity of web information systems [1] used in information-intensive domains is steadily increasing. Initially such information join together a web page that was very concretely mapped to specific file, located at specific place in the file system: this approach do not scale managing a lot of documents of different types and the need to provide in the same system a vast range of techniques and services (i.e. collaborative web software, news and searching systems, web feeds) that is nowadays required by a website to be effectively useful and visited [2]. The implementation should focus on two standpoints: on one hand there are people acting on information that need to be published, on the other hand there are final users that access that system whose needs should be evaluate. A web publishing system should provide an easy method, for people acting on it, to facilitate content creation, editing and management; nevertheless a membership system should guarantee that actors play a distinct role and have not the same rights in manipulating content. This last requirement gives a control quality on information published. Web Content Management Systems (Web CMS) [3] have been developed to address these requirements:

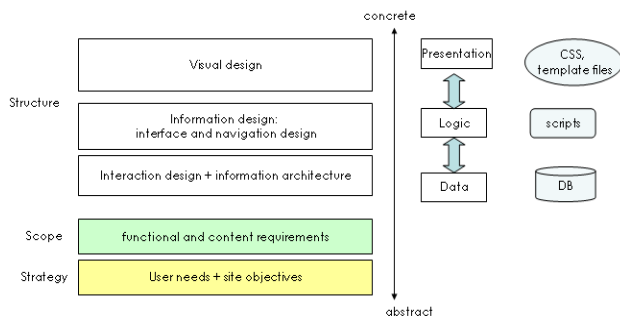
with CMSs, the content is not typed into a file with HTML/XML tags, but is inserted in a database. The CMS produces a web page on the fly that looks like ones handmade, while is a web page generator that takes an index into the database, retrieves the content for that index and uses it to populate the web page it's going to create. CMS is a solution for web sites that have lots of frequently changing items. For users instead, it is necessary that the information system provide a series of services that allow both a easy retrieving of information (i.e. a searching system, a news service, web feeds) and an active interaction (i.e. forum, weblogs [4]). In the recent definition this loose collection of concepts and technologies including weblogs, wikis, podcasts [5], web feeds and other forms of collaborative publishing is defined as the Web 2.0 world [6]. Having all these software available on the same framework could greatly enhance every portal. In this scenario, the paper describes the design and implementation of the web information system for the National Institute of Astrophysics (INAF) users needs and site objectives. The requirements for a system that distributes carefully controlled information and provides additional services to final users both for an optimal site use and for

collaborative goal, demand for a multi-tiered framework that host different application. The adopted solution make use of the Plone software (<http://plone.org>) that is developed as a CMS application within the Zope (<http://www.zope.org>) application web server, while web hosting is served by the Apache HTTP server (<http://httpd.apache.org/>) to provide more flexibility to the entire information system. In the context the CMS application is not only used as a flexible and multiuser publication tool for web portal management; but as a tool that allows to control the quality of the information. For this aim, a membership and workflow system has been configured for managing each type of content. Furthermore we analyze services, added to this system as Zope/Plone products, to satisfy some users needs.

## 2 Collaboration and final users demands

The design of the web information system requires several steps that starting from its strategy and scope define its structure (figure 1).

Figure 1: Structural model of a web information system and a three-layer architecture



We refer to a three-layer architecture model that keeps content, logic and presentation separate: this helps to implements the two main goal of our system that are a web publishing system for different actors and a set of additional tools for final users needs.

### 2.1 Web CMS: a collaborative approach

Content management, as a series of technologies and processes that supports the lifecycle of digital content created by more people, has many facets. A web CMS, besides providing document and content management through the web by essentially using a browser, may assist in automating various aspects of web publishing and managing. Addressing the main steps of managing content (creation, update, publish, storage and use), the collaborative process may be based on the following roles and privileges:

author (which is responsible of content creation), editor (which controls the formal aspect of content, style and representation), reviewer which controls the quality of information, publisher which publishes information and administrator that manage the entire system. The main functions that need to be provided for our information management process are:

- the creation of new or editing of existing information as part of a controlled generation and publishing process: this implies user and roles identification in content production and delivery and the definition of the workflow for the different types of content;
- the secure separation of access to public and non-public information;
- the automatic conversion for various display formats with specific visualization for internet presentation (browser, HTML, XML, etc.) by using a templating method. The aim is to provide an environment where web actors actively collaborate to create, edit, manage and upgrade content, but where each one follows its specific role and permission to avoid inconsistency, wrong data and general problems that may arise when this security features is not implemented.

### 2.2 Additional services

Final users needs to have information available in different way and format: besides a specific navigation tool and visual design that may easily direct them to the information, other tools should be available to facilitate their retrieval.

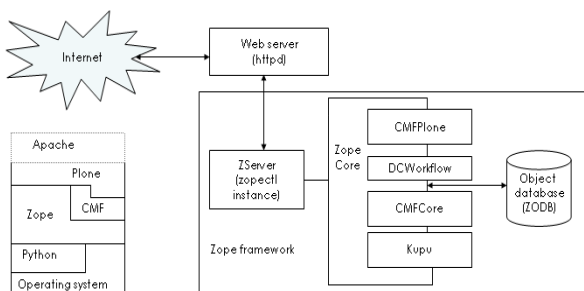
#### 2.2.1 Searching system, news services and web feeds

A main tool of an information system is a search engine software utility that can be used to index information, making it easier for visitors to find what they are looking. Moreover since providing a lot of new information is a primary requirement, managing news and similar techniques are a way to keep updated users. A news systems is an easy way to put small information directly available. A web feed is a document (XML-based) based on a simple data format (i.e. RSS – really simple syndication) [5] used to inform visitors or other sites about headlines or content. Syndication is used to describe the action of making available a feed for an information source. Users may use feed readers to be constantly in contact with a specific site. It may be used to be informed about upgrades of site sections or for site news to reduce the time and effort to regularly check it.

### 3 Software Solution: the Plone/Zope framework

The adopted solution that aims to give an integrated environment able to provide all these features, is based on Plone software [], a CMS application written in Python language (<http://www.python.org>) hosted in the Zope framework. The Zope application server infrastructure [] is an object-oriented platform that offers to its applications basic and advanced functionalities (i.e. user services, security): all data (content, HTML templates, scripts, the search engine and code) are store as objects in a transactional object database (called Zope Object database or ZODB) that can use the file system as its backing store. Numerous products (plug-in Zope components) are available to extend the basic set of tools through the “Products” tool which consists of packages of Python files put into a specific location. Plone itself comes as a set of products: it is built on a Zope application called CMF (content management framework) that provides many of the architectural elements that it uses: skins, tools, content definition and a web interface.

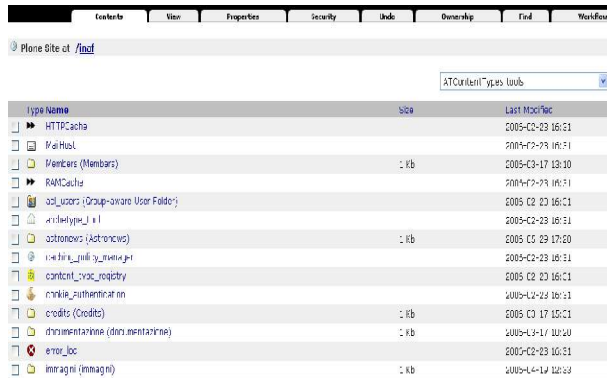
Figure 2: web information system hierarchy and architecture



Specifically the framework includes a Zope core which provide the basic platform and several possible instances: an instance contains data and software that make a running Zope server unique. Each web request is treated as a separate transaction by the object database. It includes its own HTTP serving capabilities (called ZServer), even if the integration into Apache web server is to prefer. A Plone site stands as a hierarchy of objects inside the Zope structure: the Zope administration interface (called Zope Management Interface or ZMI) provides a way to view and manage all the objects that make up a site (figure 3). Besides the specific folder which constitute the site section with the specific content inserted by web actors, there are the main portal\_\* objects, called tools, which provide different features and behavior of the site (figure 3). The Plone web interface has a default layout, but everything may be customizable. Content (inside

portal\_types tool) that may be of different types, is separate from logic which is provided by python scripts and in turn by the content presentation that is provided by templating languages (inside portal\_skins). Layout and appearance is almost decided by Cascading Style Sheet (CSS).

Figure 3: the ZMI interface for managing and viewing the site



The main features is that every customization in terms of the three layers is done by adding specific object in the custom sub-object of the portal\_skins tool without modifying source code. The resulting layout is shown in figure 4.

Figure 4: A screenshot for resulting site layout



Administrators are the only authorized to make such modifications, together with the management of site properties, users and products management . In fact web actors may act only on content and not on the site behavior, layout or properties. The end result is a site on which content can be maintained entirely over the web; web pages are generated by unique combination of plaintext authoring, clever plug-ins and site-wide style.

### 4 Membership management and workflow system

The key aspect of such information system is the setting of the content publishing process: this requires the establishment of specific actors to

distinguish who can create content and who can approve whoever content actually gets published and the implementation of a workflow on content. Zope implements the delegation of editing or authoring responsibilities of a section of publication to others: its security architecture, built around the concept of “safe delegation of control”, allows to turn control over parts of a site to others. Objects provides a much richer set of possible permissions than a conventional file-based system: permissions vary by object type based on capabilities of that objects. We can set restriction so that a user can only create certain kinds of objects.

Plone provides a membership system so that users can have a specific identity on the site: such identity may be used to make sure that only authorized people can act on certain content. Moreover user are provided a place to work keeping track of changes without touching the site. Respect the basic user management that allows for users to join in the site, in this system users are added only by site administrators that also set their properties in term of rights and group membership. Figure 5 shows the interface used to configure users.

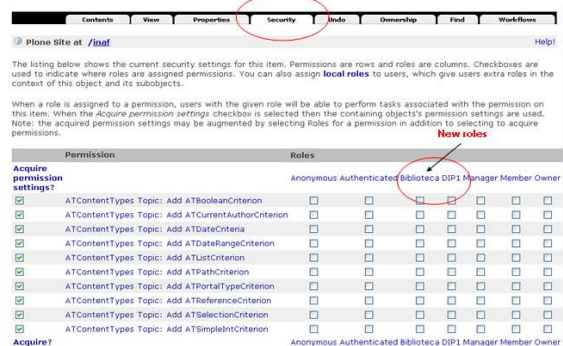
Figure5. The Plone interface to manage users



With this configuration each users has a security policy that is set by his/her role in terms of a set of permissions that are assigned either for the site in general or for specific content. Basic security mechanism lies on user credentials (a login and password to access the system), even if the system recognize a special user called anonymous and special roles (anonymous and authenticated) that help to discriminate about certain actions done in the system. Every action is guarded by a permissions: they are named after the action they control and are assigned to roles. Permissions can be viewed and set in a special tab (called security) of any ZMI object (figure 6).

Advantaging of this feature, we have defined a set of new roles to have different permissions on different section of the system besides the ones of Plone that are manager, owner, member and reviewer.

Figure 6. ZMI interface to manage global role and permissions.



Roles can be assigned across the entire site (global role) or on a per-object basis (local role) to set a much finer control over access to content or features. Global roles are assigned as seen through the ZMI interface checking the action in specific section of the site, while local roles are set inside the Plone interface by using a specific tab called sharing that allows another person to act with a specific role in that section. Finally users can be collected in groups: a group can have roles, can own content and can otherwise act like a member. A group is created to collect the same permissions. A user interacting with the system has his/her own space and interface (figure 7): after the login, he/she works with content by using a set of actions and tools included in toolbars or buttons.

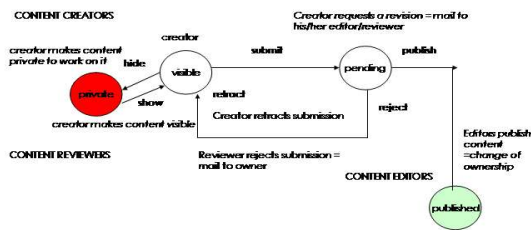
Figure 7: Actions and tabs available to authenticated users



But a CMS allows to manage the lifecycle of a content and a workflow system may be applied to automate the process that applying to content different access policies based upon certain criteria, makes content to follow a progression through various steps. Workflow are set by content type. Furthermore users have different rights on the content and their lifecycle: different workflows are available for folder and other contents. The central concept is the content objects have a state that determines how permissions are set on that content object and the states to which the content can move through the transitions. For example in our implementation (figure 8), that differs from the default one in the actors involved and in the script developed for specific tasks, we have some content creators works in their workspace adding information as different types of format (web pages, files, images, etc). Their number coincides with the

sections in which the site is structure and are related to the institution organization.

Figure 8: Lifecycle of a content: states and transitions



Furthermore each section has an editor which is responsible of position and layout of the content and a reviewer which is responsible of the content meaning. In this way a generic content follows a detail lifecycle that going from its creation to its publication involves all these actors. In the creation phase content is in the visible state: with the hide transition, it may be put in a “private” state that means that anyone can see it. This state permits creator to modify the content. When it is ready to be published, creator makes again the “show” transition and then chooses the “submit for review” action to send the content to be reviewed. This implies a changing to a pending state that is added with the action to an automatic mail to its reference editor. The editor in turn advertise the reviewer that may accept the content as it is or reject it to be modified motivating such action. At this point the editor has two action: in the first case publishes the content which passes to the published state by the “publish” transition, or rejects the submission with the reject state and the content returns to the visible state. As in the submit transition there is the “mail-to” action state, in the publish transition, we have associate a “change of owner” action in order that the published information may not be modified by creator.

### 3 Additional services

Among the default content, the news item is specifically target for such service. Plone provides elements in the site look and feel called portlets to provide site services: a specific one is dedicated to the news. Furthermore Zope has a built in search engine (ZCatalog) that allows to categorize and search objects. It provides a fast indexing and searching with drop-in ZCatalog objects: it allows indexing of any kind of object. The objects can be indexed by their content as well as properties. Moreover it supports a rich query interface to perform full text-searching and it keeps track of meta-data. Finally Plone can generate RSS data for

syndication of the content of the folders that uses RSS 1.0 and RDF-based format. Syndication may be applied to news, folders and topics, and properties managed are the time and frequency of update and the maximum number of items syndicated, : after its enablement, the object becomes an RSS channel. We have enabled for example channels for news, for new upgrade in order that RSS readers and aggregators let visitors to be informed about news and changes.

## 4 Conclusion

The analyzed Zope/Plone framework has been revealed a powerful and extensible web information system for the INAF requirements. The key aspect is the easy management by web users that hides the complexity of the entire system: the possibility that people with no particular technical knowledge can add information to the system has been very important to guarantee an often updated site that enhance its usefulness. From a technical point of view, the system, that up to now has a traffic of three thousand hits a day, was deployed in a machine with a 3 GHz processor and 2 GB of RAM: it is stable and robust and helps to keep separate web editors from administrator roles. Besides it has required a great effort in the first customization of layout, setting of templates and the workplace for users. Thanks to its extensibility, we are trying to integrate other products in the system such as a newsletter and a weblog.

### References:

- [1] Jesse James Garret, *The Elements of User Experience: User-Centered Design for the Web*, New Riders Press, 2002
- [2] Louis Rosenfeld, Peter Morville, *Information Architecture for the World Wide Web: Designing Large-Scale Web Sites*, O'Reilly, 2002
- [3] Russel Nakano, *Web Content Management: A Collaborative Approach*, Addison-Wesley Professional, 2001
- [4] Shelly Powers, et al. *Essential Blogging: Selecting and using Weblog Tools*, O'Reilly, 2002
- [5] Will Richardson, *Blogs, Wikis, Podcast and other powerful web tools for classrooms*, Corwin Press, 2006
- [6] Tim O'Reilly, *What is Web 2.0*, O'Reilly on-line, 2005
- [7] Ben Hammersley, *Content Syndication with RSS*, O'Reilly 2003
- [8] Andy McKey, *The definitive Guide to Plone*, Apress, 2004