

# A Survey on Workflow Aspects in Content Management Systems

Pedro Pico, Alberto Rodrigues da Silva  
pedro.coelho.pico@gmail.com, alberto.silva@acm.org

Instituto Superior Técnico / Universidade Técnica de Lisboa

**Abstract.** Content Management Systems (CMS) are software platforms that strongly contribute to make organizations more agile, flexible and dynamic concerning the management of their contents: business-oriented structured and no-structured information. A CMS's extra feature is Workflow support since it can allow task automation, ultimately increasing organizations productivity. While there are different kinds of Workflow platforms, this paper concentrates mostly in Content Management Workflow, analyzing key features like: Workflow definition, representation, instance management, content mapping and third-party application communication. Finally, it is also analyzed and discussed the workflow support in existing CMS such as Alfresco, Typo3, OpenADMS and Vignette.

**Keywords:** Content Management System, CMS, Workflow, Workflow Management System, Business Process.

## 1. Introduction

Workflow is an important concept and technology that is relevant within Software Engineering as well as Organizational Engineering. Nowadays, there are a relevant number of organizations that are increasingly embracing it. Due to the richness and abstraction of the concept, Workflow will only be mentioned in the Software Industry endeavor, meaning that it will only be applied to Software Application issues. Bearing that in mind, the Workflow Management Coalition, WFMC <http://www.wfmc.org> is a reference organization responsible for the definition of standard specifications regarding Workflow. The WFMC defines Workflow as “the computerized facilitation or automation of a business process, in whole or part” [1]. On the other hand, Marshak defines Workflow as “The automation of the processes we use every day to make our business through. A Workflow Application does automatically the sequence of actions, activities and tasks to run a process, including all the routing within the stages of each instance of a process, as well as the tools to manage the process itself” [2].

Automation and business processes are concepts that are mentioned on both definitions, leading to a first definition of *Workflow* as a “business process automation”.

WFMC still introduces the concept of a *Workflow Management System*, as being a “system that completely defines, manages and executes the workflow through the execution of software whose order of execution is driven by a computer representation of the workflow logic” [1].

These definitions indicate the Workflow logic has to be represented in a formal language, so that running software may be able to build, manage and execute that workflow logic.

Workflow technologies may be applied with several purposes and application contexts, such as (1) system’s integration support, (2) user interface and (3) content workflow in CMS (Content Management Systems).

**Workflow support in systems integration.** Information system’s high-level abstractions and business service’s interactions may be seen as a Workflow system [13]. Currently, Service Oriented Architecture (SOA) [14] is an example of this approach, in which technology is only a tool for orchestration of business processes and services. According to the SOA approach there should be a Service repository which agglomerates services – which according to the OASIS [32] organisation is defined as “a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description.”. These services, in their turn, communicate among them and with front and back-end applications, through a communication channel. To accomplish this, each service application must expose a programming or computing interface. Workflow plays the role of orchestrating the interactions amongst all these services. Microsoft Biztalk [17] and BEA WebLogic [31] are the leading application servers that support SOA in the industry.

**Workflow support in User Interface Software Applications.** The Workflow concept can also be applied to the end-user interface definitions for software applications. The links between interaction spaces (e.g. web page, window screen) are events that may be triggered by human or third-party applications. For example, the submission of a web form is a human interaction, while the presentation of a RSS (Really Simple Syndication) feed is a third-party application interaction.

To make the most out of this vision of software design and implementation, some companies have been publishing Interface Workflow Managers. This functionality is usually associated with the software developing process since the tools that support these features are usually within the Integrated Developing Environments, IDE, of the existing platforms. With Microsoft Visual Studio[18] the users have the ability to visually edit and see their application’s interfaces as a tree of nodes, in which each node is an interaction space and the connection among nodes define the hierarchy of the nodes in the tree. On the other hand, Sun released in 2006 JAVA Web Studio Creator[19], supporting the same functionality for J2EE applications.

**Workflow support in CMSs.** Content management environments are also a relevant application for Workflows. These systems evolved from meta-applications and frameworks which were used to produce other applications. The urge for Documental Management Applications left only one step further what would later be known as Content Management Applications. The difference between these two is the object that is managed, while the former manage documents, the latter manage contents, which is an abstraction in which documents can be included.

In these environments a Workflow can be seen as the set of stages that content may assume since its creation until it is made available. Consider the following example: (1) a document is created and submitted to an application; (2) then it is approved in chain by a set of users; (3) until it becomes visible for all convenient users.

In this example the stage transitions are mainly triggered by human interaction. Users trigger stage transitions so that contents may evolve through the hierarchy structure of the organization, being progressively approved by users with more and more responsibilities. This type of Workflow systems has to make sure that if content is not approved by a user, it must return to its previous stage. Stage transitions also have to be able to trigger automatic actions. For instance, sending e-mails so users may be notified of pending decisions they have to make. Stage transitions may not only be triggered by human interaction, but also by third-party applications.

This paper analyzes and discusses the Workflow support that is required in enterprise applications, designed on the top of CMS platforms. In particular it identifies (1) Workflow concepts as well as their common use cases; (2) the way Workflows can be implemented; and (3) functionalities available for their end-users.

After analyzing and comparing relevant CMS, we propose a generic reference model, based on which we discuss them in what concerns their Workflow's features.

## **2. Content Workflow in CMS**

This section describes the Workflow mechanisms supported by Content Management Systems: its background, its elements and its functionalities.

### **2.1. Technological Support Aspects**

Content Management Systems (CMS)[20] promote the separation between contents and services. The latter are responsible for content's presentation, manipulation and access, while the former are the artefacts that are passed throughout services. Pictures, texts, links, news, videos, and documents are all examples of contents.

The main goal of Workflows in CMS is to provide a path for contents since their creation, until they are made available for other users to see – often defined as publication. The path can be described by a set of consecutive evaluations, usually referred as stages, in which users, defined by a specific business role may, or may not, approve contents. For example, in a newspaper, each article, after written by a journalist, must be reviewed by the journalist's supervisor. If the article is approved, it will go on to the next evaluation, and an upper supervisor will have to evaluate the article; otherwise it will return to the previous stage, and the journalist will have to rewrite the article. The evaluations will go on in chain until eventually some supervisor will approve the article's publication. A Workflow in a CMS is exactly the chain of evaluations that content undergoes since its creation until its publication. It is now assumed a Workflow has several stages. While each stage is associated with one role – set of users – who are responsible for evaluating the content – either approving it, or declining it. When a supervisor declines content, the content will return to the

previous stage so it can be reviewed or rewritten again. If, on the other hand, the content is approved it will move on to the next stage. If it achieves the last stage, the content will be published. While in the physical world articles are printed in paper, in the digital world contents may either be created or replace existing versions of the content. As an example it can be considered any page available in the *wikipedia* web site, <http://wikipedia.org>. In this web site each page may be replaced by a new version, as well as new pages may be created. Pages are the contents in this example.

The multiplicity of instances of the same content that have to co-exist lead to the need of content versioning. This happens since at least two versions of the same content will be needed. One that is published, and thereby the one all users can see, and another one which is evaluated by supervisors, and thereby available for supervisors to see. The former will be referred as the published version, while the latter will be referred as the draft version. This can lead us to the conclusion that content is characterized by its *version*.

Finally, there is also another aspect to be added, the possible existence of predefined actions that may occur every time a stage is achieved or departed from. A simple notification to the author every time the content he submitted is approved by a supervisor is an example of what a predefined action can be.

From the above observations it is concluded that in the CMS domain a Workflow has several stages. While each stage has one responsible *role* and a set of predefined actions that may be triggered either when the content arrives or leaves the stage. Finally the content, which is the artefact that goes through the stages of the Workflow, has to be identified by a version, since in the most simplistic scenario at least two instances of the same content will have to co-exist.

## 2.2. Content Types

In general terms, content management workflows may manipulate two types of contents: unitary and aggregators. The former are the basic cells manipulated by the CMS, the ones which are processed as a single unit and thereby elementary operations are made upon them. An image, a defined piece of html code, or a file, are examples of unitary contents. On the other hand, aggregator contents are sets of other contents,. The aggregators provide the “glue”, which connect its sub-contents. Examples of this type of contents are a list of links, a list of documents, or a custom content which aggregates one image, one link and one text.

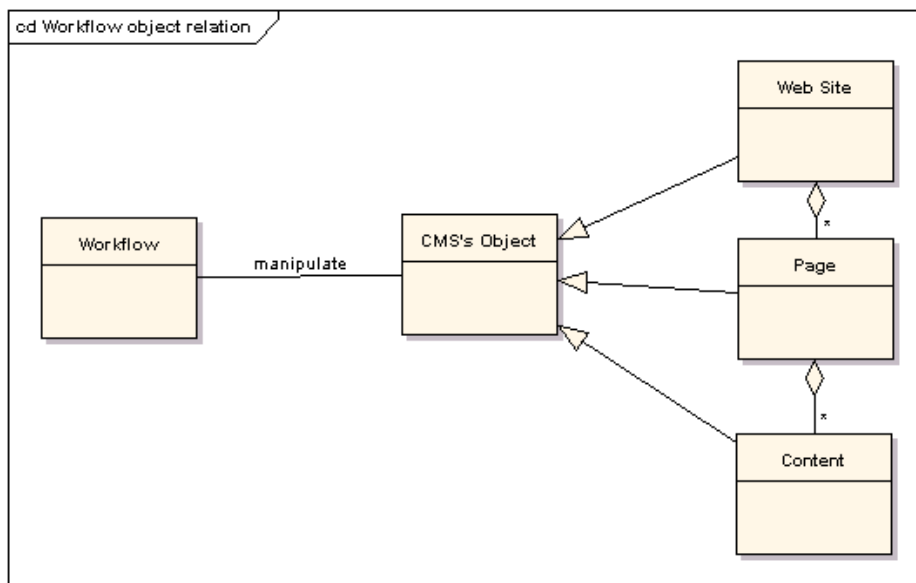
The existence of aggregators brings up another issue: the hierarchy which is formed from multi-level aggregator contents. This issue will have an important impact in the Workflow implementation within the CMS, because, like it is previously stated, contents are the artefacts which go through stages of a Workflow. So far content was always assumed to be unitary, but if it is a set of contents, in a several level hierarchy, a much more careful approach has to be done.

In order to escape from the abstraction of this issue, a scenario will be drawn to materialize it, making it easier to understand.

In a web CMS context it may be considered the following hierarchy of contents: the web site, the page, and the unitary content, as illustrated in Figure 1. Each of these types of Contents has their own characteristics as described below.

The **web site** is the highest granularity content and it aggregates page contents. This type of content must be used in a limited way since a Web site should be stable and any content under it should be updated without having to update the whole Web site. Nevertheless the Website may be submitted to a Workflow when it is created, so that the process may be monitorized by the organization's website administrators.

The **web page**, or page for short, aggregate unitary contents. In spite of not being the most used content type within the Workflow mechanism, it is fairly more used than the portal, since creation, removal, edition and configuration of its attributes are all operations that are made in the portal life cycle. It is also important to make a distinction from Static to Dynamic pages because of the way their content is processed. The former are built in compile time and do not require any level of interaction, while the latter have contents that are created in *runtime* and that may require some level of interaction. Therefore static pages may be seen a unitary content which allows to edited a huge portion of html code; while dynamic pages can be seen as an aggregator content, since they provide several outputs to the user according to what was given as an input.



**Figure 1: Workflow content relation**

At last, the **unitary content** is the lowest granularity content in this type of Workflow. This content have its own attributes, some of them are even common to all contents – such as the name or page in which it is included. The attributes which vary from content to content are the ones that identify the content itself (e.g. an image has a filename, width and height, or a byte array; while an html element does only have the html source code; in its turn a link is formed by the text and the url). The unitary

content is the most often processed content type by the Workflow Manager because of its independence from content to content and from the pages in which they are included.

Considering the above scenario it is clear that content has to be able to follow a Workflow despite its type. It is also clear that content representation is definitely an important issue that a concrete implementation of Workflow in a CMS has to address since contents may vary on their building blocks, assuming complex hierarchies. Finally it can be stated that despite content representation is a custom problem each CMS has to address (and that is not the subject of this paper), all types of contents which desirably will be able to follow Workflows in that CMS have to be understood by its Workflow engine as contents, meaning they have to share the concept of content which is accepted by the Workflows of that CMS.

### 2.3. Workflow Elements

This section lists and summarizes the concepts explored so far, as well as defines other relevant elements. Figure 2 relates these concepts, forming a reference model, with concrete CMS Workflow elements.

**Workflow.** A Workflow has several stages. The concrete number of stages should be set when a Workflow definition is created. A CMS should allow managers to create Workflow definitions, giving them the opportunity to choose then the number of stages that that particular Workflow definition should have. In other words the number of stages of a Workflow should be dynamic since it allows different number of stages for different Workflow definitions. One tricky way of achieving the dynamic number of stages is to not determine it when the Workflow definition is created. Instead, each stage is responsible to determine the next stage, also determining when the Workflow should end. This type of workflow definition will be referred as not having a defined number of stages as in opposite of static and dynamic number of stages. The drawback of such solution is that the path between stages is not memorized and if the same stages are always used, they have to be defined every time a Workflow instance executes. Finally a Workflow definition with a static number of stages has the drawback of not allowing two different Workflow instances to run with different number of stages.

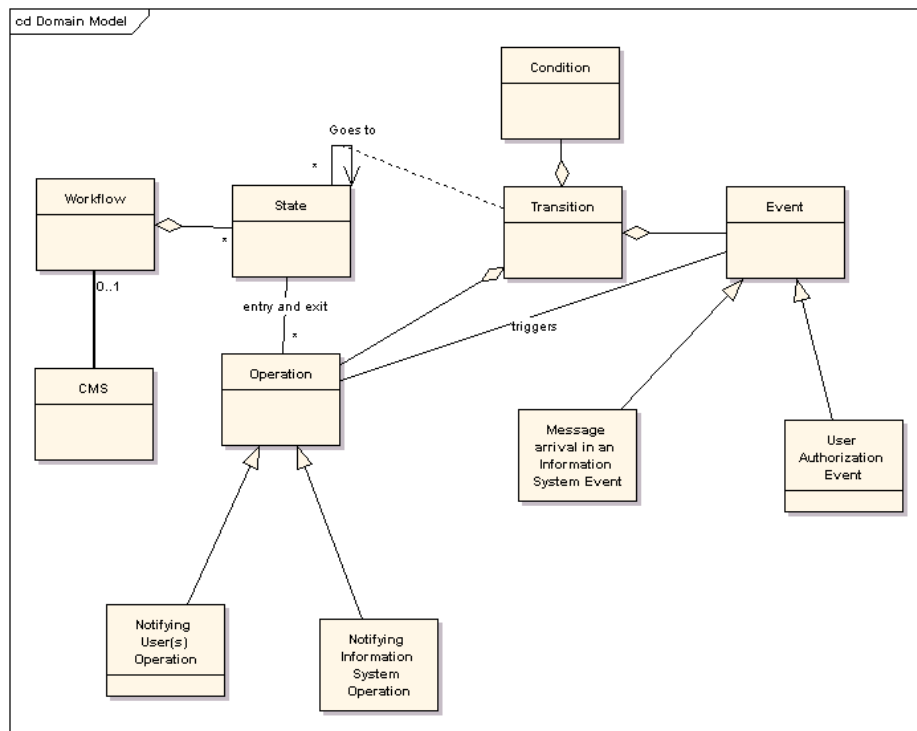
**Stage.** Every stage, as previously stated, has a supervisor role, who can determine if the content is accepted or declined. If the former is picked the next stage is reached, while the previous stage is reached if the latter is chosen. There are two special stages: the initial and the final. The initial stage is the one that starts the workflow, and the final is the one that defines when the workflow execution comes to an end. Every stage may have a set of associated operations that can fall into two categories: entry and exit operations. The former are executed when the content gets to a stage, while the latter are executed when the content leaves the stage.

**Stage transitions.** Empirically a transition is defined as a set of three elements: event, condition and operation. The event is external to the transition and when it happens it triggers the condition to be tested. The condition is the heart of the transition since it determines if the transition is executed. The operations are usually

performed if the condition is met. However events can also trigger Operations to be performed. Having this definition in mind, in CMS Workflows the events are content creation and edition, content approval (or disapproval), or messages from third-party applications. As to the conditions they consist of checking if the input given by the supervisor was an approving or disapproving instruction. When content is approved the condition is met and for example an operation of notifying the next supervisors may be executed.

**Users.** Users are responsible for approving or declining contents and so, firing events. Since content management systems deal with several users, they usually group users into roles, in which one role can have one or more users.

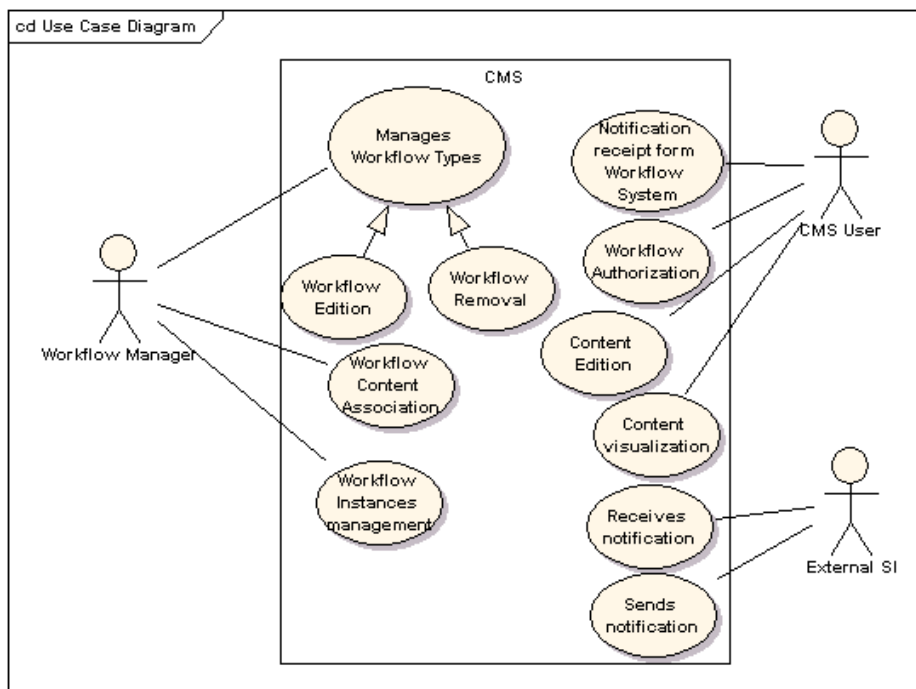
**Content.** The content, not regarding its type, is the object that runs through the Workflow. The content is needed to have a version so several instances of one content may co-exist, in order to supervisors and regular users may see different versions of the content.



**Figure 2: CMS's Workflow Support – Domain Model**

## 2.4. Functionalities

Figure 3 presents the use case model that reveals Workflow's main functionalities in a CMS. The actors present in the diagram are: (1) Registered User (URegistered), (2) Workflow Manager (UWorkflowManager) and (3) External Information Systems (IS-External). The UWorkflowManager can manipulate workflows, associate them to contents, as well as manage workflow instances. URegistered edit contents, receive notifications, authorize contents and monitor workflow instances. Finally, IS-External can notify and be notified by the Workflow Management System.



**Figure 3: CMS's Workflow Support – Use Case Model**

We set the focus to the main functionalities a Workflow Manager must have in order to support Content Workflow. The functionalities to be mention are only the ones that directly relate to the Workflow Engine.

**Workflow definition:** When performing it the number of stages should be set, as well as the responsible role for each stage.

**Association between Workflow and content:** With this functionality the user assigns a content to the Workflow.

**Workflow edition:** Existing Workflow definitions may be able to change.

However one should be very careful on how to handle existing executing instances of Workflows in order to avoid the loss of their contents.



**Workflow deletion:** it should be possible to delete a Workflow definition. The same situation happens as when Workflow definitions are edited, so the user must explicitly choose what to do to existing workflow instances.

**Workflow Instance Management:** Allowing administrators to stop workflow instances that didn't come to an end and that may be in a dead lock or starvation situation.

### 3. CMS with Content Workflow Support

Bearing in mind the concepts and functionalities discussed in section 2, four CMS were chosen to be analyzed and discussed and, consequently, to give us a better understanding of the problems in consideration. The criteria for choosing these CMS systems are the following: (1) provide Workflow functionalities; 2) distinct among them in which refers to being open-source or commercial; 3) distinct among them in which concerns their main purpose;

It should be stressed that it was surprisingly hard to find CMSs supporting Workflow technology.

#### 3.1. Alfresco

Alfresco [22] is an open-source CMS focused on documental management. Its highlight features are the version control, role support, content transformation, search engine and navigation either in the file system or via Web.

The contents which are handled by Workflows are exclusively unitary contents, meaning that it is not possible to apply workflows on aggregation contents. There may be three predefined stages: draft, review and published. The supported actions are authorization (approval or denial) by users and code actions (javascript). The latter are executed as soon as a stage is reached. There are also discussion forums, associated to each Workflow, so that the stakeholders may exchange opinions about the evolution of the given content. The notification system is based on the e-mail as well as through a module of pending actions that each user has access to, which shows the actions the user may execute. There is an administration console in which it is possible to see the stage, each Workflow is currently at. It is also possible to cancel the workflow in this administration console.

#### 3.2. Typo 3

Typo3 [23] is an open source PHP- based CMS. For Typo 3, a Workflow consists of a name, a user, and a deadline. There is the possibility of notifying users when the workflow starts. The only contents supported are also unitary contents. The user who is responsible for one stage determines the next transition – not defined number of stages - , whether accepting or declining the received content. The user may also schedule the deadline of the next stage. When the final stage is reached and the user

associated with that stage accepts the changes, the workflow ends, and the content is published.

Contents in intermediate stages are saved as drafts, while published contents are saved as final content, so there are always two versions of a determinate content: draft and final.

This Workflow model allows the existence of a variable number of stages. Despite the stages are built in execution time, while contents go through the Workflow. The disadvantage of this model is that equal Workflows have to be created every time there is a need of a new instance. In fact this means there is no workflow definition operation.

Bottom line is that this is a very simple and straightforward Workflow Managing System, which provides the content flexibility to the evolution within the Workflow. However it reveals a lack of automation, concerning stage definition, since stages are defined by its instances at runtime.

### **3.3. Altimate OpenEDMS**

Altimate OpenEDMS [24] is a CMS that allows Workflow definition and management.

The Workflow definition process is supported by an activity diagram visual editor, which enables the user to create stages and transitions. Each stage has a responsible role, automatic actions and destination stages. When the Workflow definition is completed there is a validation in which the system determines if the Workflow is, or is not, valid. Afterwards the Workflow is saved.

The notification system may be done in two ways, e-mail or private message (system internal messages), while it is possible any kind of combination of these types of notification.

Terminated and pending Workflows are possible to inspect, as well as to start a new Workflow. To perform this last one the user has to name the Workflow, insert a comment, describing it, and a starting date. When a workflow finishes it is possible to distribute its content to selected users.

The most relevant feature of this System is the Workflow definition process, since it is completely visual and user friendly.

### **3.4. Vignette CMPortalSolution**

Vignette's [26] CMS is a commercial product and a reference among its peers in the industry.

Workflow is a central concept within this CMS, and it uses both unitary and aggregation contents. Workflow definition is achieved visually, via Microsoft Visio, and stage transitions may be triggered whether by user authorization or external IS actions. After edited, contents have to be approved, so they may be published. The instance management is done in a console that allows its edition and cancellation.

### 3.5. Comparative Analysis

The study of the referenced CMS leads to the conclusion that only one out of the four actually allows the use of aggregation contents on Workflow. Also, out of the four CMS, only Alfresco has a static number of stages. The Workflow definition process is different among the CMS. Typo3 is the only one that does not allow determining the number of stages at the Workflow definition, since the Workflow is defined in run time by the user that is responsible for the current stage. V7CMS is the only that allows external IS interaction, while all of them allow User interaction. Notifications and deadlines are important concepts, although not supported by all the CMS.

Concerning the supported functionalities, the most important are the Workflow definition, the content, Workflow association and Workflow deletion and management.

	Alfresco	Typo 3	Altimate OpenADMS	Vignette
<b>Concepts</b>				
Number of Stages	Static (3)	Not defined	Dynamic	Dynamic
Supported Contents	Unitary Contents	Unitary Contents	Unitary Contents	Unitary & Aggregation Contents
User interactions	Yes	Yes	Yes	Yes
IS interactions	No	No	No	Yes
User Notification Operations	Yes (E-mail)	Yes (E-mail)	Yes (E-mail, internal messaging system)	Yes (E-mail)
<b>WF Supported Operations</b>				
Definition	No	No	Yes (Visual Interface)	Yes (Microsoft Visio)
Association	Yes	Yes	Yes	Yes
Edition	No	No	Yes	Yes
Removal	Yes	Yes	Yes	Yes

## 4. Conclusions

The adoption of Workflow technology into contemporary CMSs would allow the automation of content's production as well as its better integration according the business interests. This will also lead to an optimization of the organisation's business processes.

Concerning CMS there are actually very few that support generically Workflow. Those which do, only have partial support with still limited and inflexible features.

The generic reference model introduced in section 2, allow us to analyze any of the given Workflow mechanisms, in section 3, defining additional concepts that may be useful for future releases.

Aggregation contents have little support for Workflows, since only one out of the four CMS supported it, which may lead to the conclusion that aggregation contents are not as relevant as contents from the Workflow perspective. Nevertheless, given the early stage that these classes of systems are, it may be plausible that such contents would be better supported in the future.

Despite the lack of Workflow support by CMS, there are many requests by the industry so that a standard solution of Workflow management for all content types may be supported by these classes of systems. This can be a reasonable indicator that this is an important topic for the software industry, which should emerge in the years to come.

## References

1. David Hollingsworth, The Workflow Management Coalition Specification reference model, 1995.
2. Ronni T. Marshak, Workflow: applying automation to group processes, Groupware: technology and applications, Prentice Hall International (UK) Ltd., Hertfordshire, UK, 1995.
3. Tim Bray, Jean Paoli, Eve Maler, François Yergeau Extensible Markup Language (XML) 1.0, 4th Edition, 2006 .
4. Dr. Michael Kay, Building Workflow Applications with XML and XQuery available at <http://www.stylusstudio.com/xml/Workflow.html#> , last visited in 17-01-2007.
5. Robert Shapiro, Mike Marin, Roberta Norin, Workflow Management Coalition. XML Process Definition Language, 03-10-2005 available at [http://www.wfmc.org/standards/docs/TC-1025\\_xpdl\\_2\\_2005-10-03.pdf](http://www.wfmc.org/standards/docs/TC-1025_xpdl_2_2005-10-03.pdf), last visited in 21-07-2007.
6. Keith D. Swenson, Sameer Pradhan, Mike D. Gilger Wf-XML 2.0., Draft 08-10-2004, available at <http://www.wfmc.org/standards/docs/WfXML20-200410c.pdf>, last visited in 21-01-2007.
7. Keith D Swenson, ASAP/Wf-XML 2.0 Cookbook—Updated in Workflow Handbook, Layna Fischer, 2005.
8. Nilo Mitra, *SOAP Version 1.2 Part 0: Primer*. W3C Recommendation 24-06-2003. Available at: <http://www.w3.org/TR/2003/REC-soap12-part0-20030624/>, last visited in 21-01-2007.
9. Unified Modeling Language Resource Page 1997-2007, available at <http://www.uml.org/>, last visited in 11-12-2006.
10. Marlon Dumas, Arthur H. M. ter Hofstede, UML Activity Diagrams as a Workflow Specification Language, 2001.
11. Windows Workflow Foundation, 2006, available at <http://wf.netfx3.com/>, last visited in 04-12-2006.
12. Paul C. Zikopoulos , A Beginner's Recipe for a Service-Oriented Architecture, available at <http://www.db2mag.com/story/showArticle.jhtml?articleID=193200616>, last visited in 12-12-2006.
13. Michael Rowell, Understanding EAI: Enterprise Application Integration, Sams, Indianapolis, Indiana, 08-2001.
14. Dirk Krafzig, Karl Banke, Dirk Slama,. Enterprise SOA, Prentice Hall, 2005.
15. SAP Network: Standards and Enterprise SOA, available at <https://www.sdn.sap.com/irj/sdn/developerareas/esa/standards>, last visited in 12-12-2006.

16. Balwinder Sodhi, Implement a customizable ESB with JAVA, 08-08-2005, available at <http://www.javaworld.com/javaworld/jw-08-2005/jw-0808-esb.html>, last visited in 12-12-2006.
17. What is Biztalk Server 2006?, 07-11-2005, available at <http://www.microsoft.com/biztalk/evaluation/what-is-biztalk-server.msp>, last visited in 21-12-2006.
18. Dino Esposito, Building Web Solutions with ASP.NET and ADO.NET, Microsoft Press, 2002.
19. Gail Anderson, Paul Anderson, Java Studio Creator Field Guide, Sun Microsystems Press, 2006.
20. James Robertson, How to evaluate a content management system, KM Column 2002.
21. How to Choose a Content Management System, WebSideStory WHITE PAPER, 2005.
22. Alfresco CMS, available at <http://www.alfresco.com/>, last visited in 29-01-2007.
23. Typo3 CMS, available at <http://typo3.com/>, last visited in 29-01-2007.
24. Altimate OpenADMS CMS, documentação available at <http://www.altimate.ca/Workflowdemo.html>, last visited in 21-01-2007.
25. Adxstudio CMS, documentation available at <http://www.adxstudio.com/cms-product/features/Workflow>, last visited in 21-01-2007.
26. Vignette Solutions, available at <http://www.vignette.com/>, last visited in 29-01-2007.
27. Daniel Rubio, Biztalk Server: Microsoft's SOA building block, 24-01-2006, available at [http://searchwebservices.techtarget.com/tip/1,289483,sid26\\_gci1161311,00.html](http://searchwebservices.techtarget.com/tip/1,289483,sid26_gci1161311,00.html), last visited in 12-12-2006.
28. Wil van der Aalst e Kees van Hee , Workflow Management, MIT Press, 2002.
29. Edward A. Stohr, J. Leon Zhao Workflow Automation: Overview and Research Issues, Springer Netherlands, 2001.
30. Openflow: Open source Workflow management system, available at [http://www.openflow.it/EN/index\\_html](http://www.openflow.it/EN/index_html), last visited in 18-12-2006.
31. BEA WebLogic, available at <http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/weblogic/>, last visited at 12-09-2007.
32. Organization for the Advancement of Structured Information Standards, OASIS, available at [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=soa-rm](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm), last visited at 20-01-2008.