

CMS(Content Management System)의 Access Structure에 관한 설계 및 구현

문남미, 유소연
이화여자대학교 정보과학 대학원

Design and Implementation of Access Structure-based Contents Management System

NamMi Moon, SoYeon Yoo
Dept. of InfoScience Ewha womans Univ.
E-mail : mnm@ewha.ac.kr, syyoo@twonine.co.kr

요 약

대량의 콘텐츠에 대한 정보 관리 및 잦은 사이트 변화에 따른 퍼블리케이션의 용이성으로 CMS(Content Management System)의 도입이 많이 이루어 지고 있다. 본 논문에서는 CMS의 백엔드와 프론트 엔드를 연결하는 Access Structure 에 대해 고찰해보고, 이를 통해 시스템의 콘텐츠 검색을 효율적으로 지원할 수 있는 방안과 카테고리 수의 무분별한 증가를 막고자 하며, 퍼블리케이션에서의 확장된 아키텍처를 지원하기 위한 모델을 제안하고자 한다.

1. 서론

인터넷이 대중화 되기 시작한지 얼마 안되어, 이제 웹은 또 다른 변혁기를 맞고 있다. 놀라운 속도로 인터넷은 기업의 운영방식을 재정립하고 있으며 웹사이트는 기존의 단순 홍보 개념에서 기업의 e-biz목표를 달성하는데 하나의 핵심요소로 자리잡고 있다. 이러한 변화의 움직임 속에 e-biz를 위한 웹사이트는 고객에게 빠르게 갱신된 정보를 제공하여야 하며 다양한 콘텐츠 및 대량의 콘텐츠를 제공할 수 있어야 하고, 신속하고 정확한 처리를 요구하고 있다.

하지만 이제까지 대부분의 웹사이트는 체계 없이 관

리되어 왔다. 웹 관리자들은 하루에도 수 십번 콘텐츠를 만들어 내고 복사하며 메인 페이지부터 하부 페이지 곳곳에 작성된 콘텐츠를 배치한다. 이러한 처리는 콘텐츠 양이 상당히 많아질수록 사이트의 일관성이 결여되기 쉽고, 사이트가 적정시기에 갱신되는데 많은 시간과 노력을 필요하게 한다[6]. 이렇게 비체계적으로 작성된 웹 페이지는 재사용의 어려움도 안고 있다. 또한 사이트 수가 많거나 규모가 커질수록 콘텐츠 제작에는 많은 부사와 인력들이 유관되어짐에 따라 어떻게 효율적으로 콘텐츠를 관리하고 운영체제를 구축할 것인가가 기업의 관심사가 되었다.

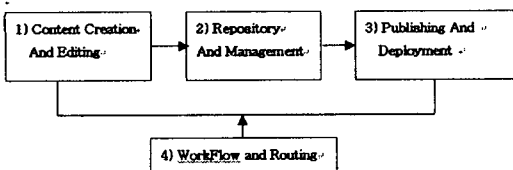
이에 CMS(Content Management System)라는 개념이 도입되었으며 이는 콘텐츠의 생산, 관리, 배포 일련의 과정을 자동화해 효율적이며 체계적인 관리를 도모하자는 것이다 [1][6]. CMS에 대한 많은 연구가 이루어지고 있지만, 본 논문에서는 백엔드와 프론트엔드를 연결시키는 CMS의 Access Structure에 대해 논하고자 한다. 이러한 Access Structure는 CMS를 이루는 근간이 되며 작업을 최소화 시킬 수 있으므로 매우 중요하다. CMS Repository에 있어서, Access Structure는 어떻게 콘텐츠를 분류하고 접근할 수 있는가에 대한 방법을 제시해야 하고, 퍼블리케이션에 있어서는 사용자들에게 페이지나 섹션을 이동할 수 있는 네비게이션 역할을 하여야 한다.

본 논문에서는 이러한 Access Structure를 AS_Category(Category for Access Structure)라 정의하고, CMS의 백엔드와 프론트엔드를 좀 더 효율적으로 관리 할 수 있는 Access Structure에 대한 방법론을 제시해 보며 효율적인 아키텍처를 구현하고자 한다.

논문의 구성은 다음과 같다. 2장에서는 관련연구로서, CMS의 개념 및 CMS의 Access Structure를 살펴 볼 것이다. 3장에서는 CMS에 있어서 Access Structure에 대한 모델을 제시하고자 한다. 이어서 4장에서는 AS_Category에 대한 설계 및 구현을 다룰 것이다. 마지막으로 5장에서는 본 논문의 결과를 정리하고 앞으로의 연구방향을 제시한다.

2. 관련연구

2.1 CMS의 모델 정의



[그림 1] CMS를 이루는 4개의 Subsystem

1) Content Creation And Editing

콘텐츠 생성의 단계는 다양한 경로를 통해 이루어

지는데, 자체 생성을 통해 이루어 지거나 외부 CP(Content Provider)등에 의해 도입되는 단계이다. 생성의 단계에 CMS는 콘텐츠의 내용과 표현양식을 분리해 다룬다. 이러한 접근 방법은 콘텐츠의 재사용성을 높이게 된다 [2][6].

2) Repository And Management

생성과 유입의 단계를 마친 콘텐츠는 적절한 분류체계를 통해 콘텐츠 Repository로 전달된다, 저장소에 담아야 하는 데이터는 실제 콘텐츠와 그에 대한 정보를 담고있는 메타데이터의 데이터 저장소, 이들을 접근, 관리하는 프로세스로 구성되어 있다.

콘텐츠 관리는 다양한 기능을 포함한다. 각 콘텐츠와 페이지 사이의 링크관리, 버전관리, 역할기반의 보안관리, 개인화 등 실로 다양하다[2][6].

3) Publishing And Deployment

Publication System이란 콘텐츠가 생성 및 승인 과정이 끝난 후, 저장소(Repository)에서 콘텐츠 내용 요소들을 추출하여 그에 따른 Access Structure를 추출, 적절한 출판매체(예를 들면 웹 브라우저 등 다른 채널)에 표현(design) 요소와 결합하여 단일화된 출판물을 만드는 것을 말한다.

이러한 Publication System은 다음과 같은 상세구조를 가지고 있어야 한다.

- publication의 platform :시스템은 원하는 출판물을 생성할 수 있어야 함을 의미한다. (web publication, print publication, e-mail, Syndication, other publications)
- Templates: Template은 콘텐츠가 보여지는 외형을 말하지만 이것은 publication작업을 자동적으로 생성하는 프로그램이다.
- Files and Directories:출판물이 요구하는 file과 directory 구조를 만들어야 한다.[3]

4) WorkFlow and Routing

위의 모든 단계의 프로세스는 권한과 역할기반에 의한 워크플로우로 통제하고 있다[2][6].

2.2 CMS의 Access Structure

CMS의 Access Structure는 다음과 같은 매우 중

요한 의미를 갖는다

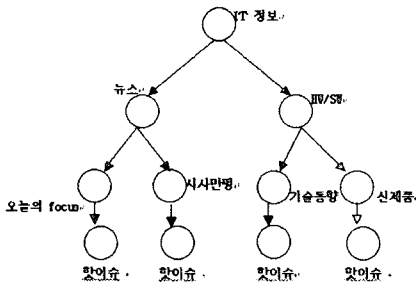
- CMS Repository에 있어 어떻게 컴포넌트들을 체계화하고 조직화하는가에 대한 역할
- 퍼블리케이션이 된 후 사용자들에게 페이지나 색션을 통해 움직이도록 제공하는 네비게이션에 대한 역할

3. CMS의 Access Structure에 대한 모델

CMS의 Access Structure는 백엔드에서 대분류와 소분류 등으로 구조화 된 전형적인 계층적 분류체계를 의미하며 퍼블리케이션에 있어서 서비스 하는 시스템에서의 디렉토리 구조를 의미한다. 따라서 본 논문에서 제안하고자 하는 AS_Category 모델은 분류체계와 검색, 퍼블리케이션에 있어 네비게이션 모두를 지원할 수 있는 확장된 아키텍처를 보여주고자 한다.

3.1 AS_Category 모델

카테고리는 공통된 주제를 기준으로 콘텐츠를 나누어 놓은 계층 구조이다. 대부분의 사이트는 트리 형태의 단선 구조를 이루고 있다. 이와 같은 분류체계는 콘텐츠들은 하나의 카테고리에 속할 수 밖에 없으며, 여러 카테고리에 속하려면 해당 카테고리에 콘텐츠를 중복시킬 수 밖에 없다. 이와 마찬가지로 하위 카테고리가 상위 카테고리를 여러 개 가지려면 트리 구조의 형태상 상위 카테고리마다 같은 이름의 하위 카테고리를 만드는 수 밖에 없다.

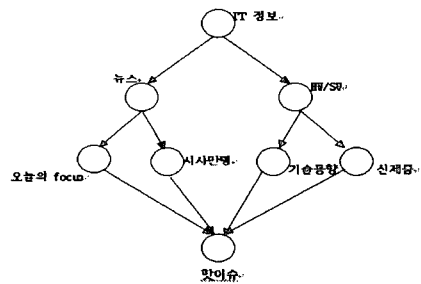


[그림 2] 단선구조의 예

[그림2]는 중복된 정보의 카테고리를 가지게 되어

사이트가 복잡해질수록 카테고리 또한 양적인 팽창이 일어나므로 WEB 환경에서 속도의 문제를 야기시키기도 한다.

따라서 제안될 모델은 [그림3]과 같은 네트워크 구조의 모델을 정의하고자 한다. [그림3]에서 핫이슈라는 카테고리가 상위의 카테고리를 중복해서 갖는다면 [그림2]에서 15개의 카테고리 구조가 8개의 카테고리로 줄일 수 있다. 이는 카테고리의 공유정책을 사용함으로써 해결을 하고자 한다.



[그림 3] 제안된 분류체계 모델

분류체계가 유연성과 확장성, 유동성을 가지기 위해서는 다양한 연산을 지원해야 한다. 카테고리는 시간이 지남에 따라 사이트의 성격상 변화를 요구하게 된다. 일부 카테고리는 없어질 수도 있고 다른 카테고리는 새로 생기기도 하며 어떤 카테고리는 병합을 요구하게 되기도 한다. 이에 AS_Category 모델은 카테고리의 수정, 이동, 복사, 삭제, 공유의 기능을 다양하게 지원하고자 한다[4].

3.2 검색범위의 구조를 줄임

카테고리는 콘텐츠를 분류별로 나누어 놓은 분류체계에 대한 검색 조건의 조합으로 나타내어 질 수 있다. 예를 들어, 그림 3의 오늘의 focus에 해당하는 콘텐츠 A가 있다 가정할 때, A라는 콘텐츠의 카테고리에 대한 검색조건은 (레벨1="IT 정보" AND 레벨2="뉴스" AND 레벨3="오늘의 focus") 와 같이 나타낼 수 있다. 카테고리의 내용이 검색조건의 형태로 표현되어 각 카테고리는 카테고리에 속한 콘텐츠에 대한 직접적인 연결 정보를 가질 필요가 없다. 이러한 검색방식은 전체의 Contents Repository를

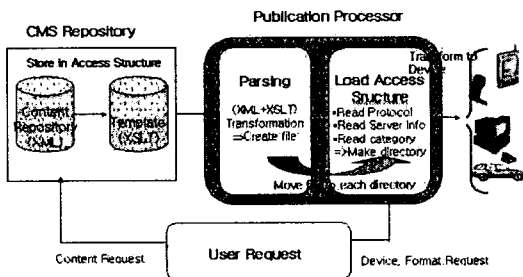
찾을 필요 없이 단계별 계층구조만 검색함으로 검색의 효율성을 높일 수 있다[5].

3.3 퍼블리싱을 위한 As_Category 모델

퍼블리싱은 백엔드의 정보를 보여주고자 하는 채널에 전달하는 작업을 포함한다. 각각의 CMS 솔루션마다 백엔드의 콘텐츠 저장 분류체계와 프론트에 게시될 네비게이션 디렉토리에 대한 나름대로의 정책을 가질 것이다. 하지만 서로 다른 분류체계 및 다른 디렉토리 구조를 가진다면 카테고리가 복잡해지고 관리해야 할 카테고리 수가 많아짐에 따라 프론트엔드와 백엔드를 관리하는 작업자는 많은 부담을 느낄 것이다. 이에 AS_Category의 모델은 백엔드와 프론트엔드의 분류체계를 일치시키고자 한다.

[그림 2]의 IT 정보>뉴스>오늘의 focus> 핫이슈의 백엔드 카테고리 정보는 프론트엔드에 똑 같은 구조의 디렉토리를 만들어야 한다는 것이다. 즉, CMS Repository에서 위의 카테고리 계층구조에 속한 콘텐츠들은 퍼블리싱 단계시 표현요소와 결합하여 서비스할 서버의 /IT/news/today_focus/hot_issue의 디렉토리 구조를 만들고 해당 디렉토리로 생성된 파일들이 옮겨져야 한다. 카테고리와 디렉토리는 CMS에 있어 무관하지 않으며 CMS에서 하나의 Access Structure인 것이다.

CMS의 백엔드 카테고리는 다양한 채널로도 서비스될 수 있도록 확장된 개념을 가져야 한다. 따라서, WAP이나 PDA 등 멀티 채널을 위한 프로토콜 및 해당 서비스할 서버의 정보에 대한 관리까지 AS_Category가 관리한다면, 멀티채널에 대한 Delivery에 대한 용이성을 제공할 수 있으리라 본다.



[그림 4] 퍼블리싱을 위한 As_Category 모델

[그림 4]는 CMS Repository에서 내용요소와 표현요소를 가지고 와 프로세싱을 통해 파일로 만들고, 여기에 해당되는 Access Structure를 load하여 해당 device로 파일들을 퍼블리싱하는 AS_Category 모델이다.

3.4. AS_Category 설계

다음은 위에 제시한 모델을 바탕으로 설계한 AS_Category의 구조이다.

```
<folder id=1 name=" IT정보" dir_name=" IT"
Server_info=" WAP" >
<folder id=2 name=" 뉴스" dir_name=" news" >
<folder id=3 name=" 오늘의 focus"
dir_name=" today_focus" >
<item id=4 name=" 핫이슈"
dir_name=" hot_issue" />
</folder>
<folder id=5 name=" 시사만평"
common=" 4" dir_name=" current_topic" />
<folder id=6 name=" 기술동향"
common=" 4" dir_name=" tech_trend" />
<folder id=7 name=" 신제품"
common=" 4" dir_name=" new_product" />
</folder>
</folder>
```

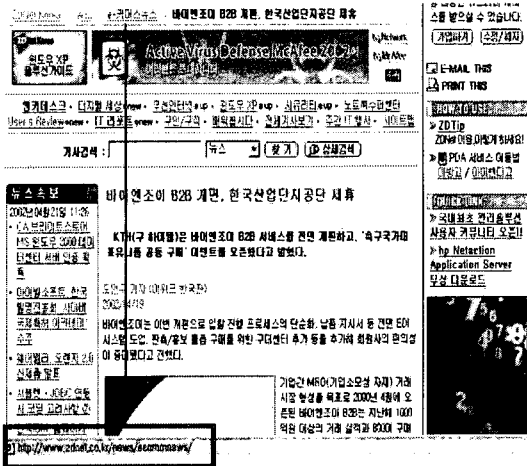
하위 카테고리를 가지는 경우는 folder로 나타내며 카테고리가 더 이상의 노드를 가지지 않을 경우 item으로 표현하고자 한다. common은 카테고리에 대한 네트워크 구조를 지원하기 위한 공유 정보이다.

id	카테고리 고유 식별자
name	카테고리 한글명, 검색 및 퍼블리싱 후의 네비게이션 이름
dir_name	해당 서버로 생성될 디렉토리명
server_info	퍼블리싱 할 서버 정보(프로토콜, 서버 IP..)

표 3-1 AS_Category의 속성 설명

4. AS_Category 구현

본 시스템은 Apache 1.3.2 환경에서 JDK1.3.1, 서블릿 엔진으로 Tomcat 3.2.2, 데이터베이스는 오라클 8i를 기준으로 구현하였다.



[그림 5] As_Category를 이용한 퍼블리싱 예

- [3] Bob Boiko, "Content Management Bible ", Hungry Minds Inc, 2001
- [4] 정지혜, " 전자 상거래를 위한 확장된 전자 카탈로그 질의 모델 제안", 석사학위논문, 서울대학교 전산학과, 2000
- [5] 홍영준, " 디지털 카탈로그 라이브러리 시스템을 위한 아키텍처", 석사학위논문, 서울대학교 전산학과, 2000
- [6] "CMS 솔루션 분석과 도입적략", 소프트뱅크 리서치, 2001
- [7] "WEB CONTENT MANAGEMENT", Ovum Ltd, 2000

5. 결론

본 논문에서는 CMS의 Access Structure에 대한 효율적인 방안을 제시해 보았다. 즉, 분류체계에 대한 새로운 모델을 제시 했으며, 검색범위의 구조를 줄인 계층적 검색, 확장된 퍼블리케이션의 방안을 다루었다.

AS_Category는 백엔드와 프론트엔드를 연결하는 근간이 되므로 이것의 확장 가능성은 무한하리라 본다. 컨텐츠의 생성은 현재 자체 생성과 CP의 전달로부터 이루어 지는데, 컨텐츠의 유입(Aggregation)이 활발해 진다면, 카테고리에 대한 외부 유입도 생각해 볼만하다. 이는 기존 체계와의 상호 연동성을 지원해야 하며, 운영상 표준화 문제도 함께 고려해야 할 것이다.

[참고문헌]

- [1] Gartner Group, "Web Content Management Overview"
- [2] Russel Nakano, "Web Content Management: A Collaborative Approach", Addison-Wesley Pub Co, 2001