

## 2章-0 CMSのアーキテクチャについて考える

Web Site Expert編集部

### ■ CMSのトレンド

1章の対談でもわかるように、ここ数年の動きからCMSのトレンドや方向性が見えてきました。とくに、CMSベンダ(プレーヤ)の顔ぶれが固定されつつあり、ニーズや用途によるCMSの選択ができる環境が整って来たと言えるでしょう。すなわちこれは、CMSの選択基準を明確にすることがCMS導入の成功につながる秘訣とも言えます。

本章では、CMS選択基準の1つとして、CMSのアーキテクチャ(設計思想)に注目し、中でもデータの構造化およびデータベース(DB)との連携についてフォーカスします。最初にまず、CMSのアーキテクチャについて、コンテンツリポジトリと汎用DB型/独自DB型の分類について解説します。

### ■ CMSのアーキテクチャを考える

CMSのアーキテクチャといっても非常に多岐にわたります。また、時代の移り変わりとともにその目的や用途が遷移してきました。

CMSが登場した当初は、ECM(Enterprise CMS)とWeb CMSという2つのカテゴリに分け、前者は企業・組織内での使用を強く意識

したもの、後者はWebサイトとして運用することを意識したもの、というおおまかな切り分けがされていました。

とくにWeb CMS(以降CMS)は、導入が始まった時期には、サイト管理者の作業軽減、コスト削減など、Webサイトの運用の効率化が重点的にフォーカスされていたことがありました。そのため、コンテンツとなるファイルの管理だったり、公開に関する権限、承認フローなど、さまざまな機能が実装されていました。その他、ページそのものを作成するエディタやビジュアル制作ツールなどが実装されていることもあります。

こうした中、最近とくに注目されているのが、情報設計を意識したデータ管理の仕組みです。データ量の肥大化に加えて、最近のWeb 2.0の考え方にもあるデータの流用、マッシュアップと言った概念を取り込むうえで、データの管理手法がとくに重要になっているのです。また、サイトリニューアルや改善といった実装面に関しても、データ管理をきちんと行えているか否かは大切になります。

そこで、コンテンツを保存するリポジトリを実装しているCMSへのニーズが高まっています。

### データの構造化 (コンテンツリポジトリの実装)

CMSで実装されているコンテンツリポジトリは、通常扱うデータを構造化して格納します。XMLのような汎用フォーマットや独自仕様など、形式はさまざまですが、データを構造化し階層的に管理することで、柔軟なデータ設計を実現できます。

コラムでも取り上げているJSR 283では、Javaによるコンテンツリポジトリの汎用フォーマットの策定を目指しており、今後の普及に注目が集まっています。

### データを構造化することの利点

データを構造化しリポジトリを実装する利点として、

- ・データ増に対する管理の効率化
- ・リニューアル時における作業軽減
- ・的確な情報設計による二次的効果(サイトブランドの統一、SEOなど)

など、非常にさまざまなメリットが挙げられます。これらはすべてWebサイトを活用していくうえでとても重要な項目です。

言い換えれば、リポジトリを実装したCMSに対して、きちんとしたデータ設計、情報設計、サイト設計を実現できるかどうか、というのがこれからのWebサイトに求められる課題でしょう。

### データの格納場所 (汎用DB型、独自DB型)

リポジトリの実装やフォーマットだけではなく、データの格納場所にも注意が必要です。今回の特集では、OracleやPostgreSQL、MySQLなど外部のDBを使うタイプを「汎用DB型CMS」、CMS独自で実装されているDBを利用するものを「独自DB型CMS」と分類して扱います。

これは、どちらが良いか悪いかということではなく、運用するWebサイトに合ったものがどちらなのか、を選ぶための選択基準の1つです。詳しくは、本章後半(2-1, 2-2)をご覧ください。

### ■ まとめ

以上、アーキテクチャという視点でCMSを捉えてみました。現在のWebサイトでは、CMSの導入は既定路線であり、今後は自分たちがどのようなWebサイトを作りたいのか、誰のためのWebサイトを作りたいのか、そういった視点でCMSの活用を考えていくフェーズに移ってきています。

そのときの判断のポイントの1つとして、リポジトリの概念やDB対応についても意識してみてください。WS

### コラム: JSR283の現在 JSR 283: Content Repository for Java Technology API Version 2.0

JSRとは、JCP(Java Community Process)によって提案されているJava標準のための仕様です。今回紹介するJSR283は「JSR 170: Content Repository for Java technology API」を拡張した仕様で、Javaで利用できるコンテンツリポジトリの標準実装となります。

今回の特集で紹介しているとおり、データの構造化やデータベースの連携は、今後のCMSにとり、とても重要なポイントになります。中でもJSR283は、エキスパートグループに、Apache Software FoundationやBEA Systems, IBM, Oracle, Sun Microsystemsなど、主要なサーバベンダおよびグループが参加していることが特徴で、JSR283に準拠することにより、Javaベースのデータベースやアプリケーションサーバと高いパフォーマンスでの連携が取れることが期待できます。今後のWebシステムを構築するうえで注目したい技術です。なお、2008年5月にFinal Releaseが予定されています。