組織的 Web サイト用 CMS の開発

04H104 水野貴弘

目次

1 はじめに 1

2 卒業研究 2
2.1 目的 . . . 2
2.2 開発環境 . . . 2

3 CMS について 3
3.1 コンテンツを統合的に管理するということ . . . 3
3.1.1 CMS の具体的な動作について . . . 3
3.2 動的生成型 . . . 3
3.3 静的生成型 . . . 4
3.4 現在最適と考えられる生成方法は . . . 4

4 現在ある CMS の分類と開発する CMS の理想像 5
4.1 現在ある CMS を分類する . . . 5
4.2 現在ある CMS の問題点 . . . 5
4.3 開発する CMS の理想像 . . . 5

5 開発したシステム 7
5.1 CMS include Pure Plugin System - CiPS とは . . . 7
5.1.1 CiPS のファイル・ディレクトリ構造 . . . 7
5.2 SysPoS とは . . . 8
5.2.1 SysPoS の独自機能 . . . 8
5.2.2 Site P-C Model とは . . . 8
5.2.3 TopDown/ButtomUp Entry とは . . . 8
5.2.4 TopDown Entry の説明 . . . 8
5.2.5 BottomUp Entry の説明 . . . 9
5.2.6 TopDown/ButtomUp Entry と TrackBack との違い . . . 9
5.2.7 SySPoS を構成するプラグイン . . . 10

6 SysPoS の具体的な動作 11

7 考察 12
7.1 利点 . . . 12
7.2 問題点 . . . 12
7.3 発展 . . . 12

8 まとめ 13

付録 A 付録 14
A.1 要件定義書 . . . 14
A.2 コーディング規約 . . . 16
A.3 SQL データ構造 . . . 16

A.3.1 database syspos 16
A.3.2 table ENTRY 16
A.3.3 table SITE 17
A.3.4 table USER 17
A.4 ソースコード 18
A.4.1 index.php 18
A.4.2 cips/base_core.php 19
A.4.3 cips/cips_core.php 22
A.4.4 cips/plugins/entry/index.php 26
A.4.5 cips/plugins/usermanager/index.php 29
A.4.6 cips/plugins/sitemanager/index.php 31

1　はじめに

卒業研究として CMS の開発を行う。CMS とは Contents Management System(コンテンツマネージメントシステム) の略称であり、テキストや画像などのデジタルコンテンツを統合的に管理し、配信など必要な処理を行うシステムの総称である。コンテンツ管理システムとも呼ばれている [1]。

CMS は Web サイト開発においてテキストエディタで HTML を記述する、もしくはソフトウェアで全ての Web ページを作成する方法とは異なり、CMS をサーバ内に構築した後はコンテンツ (この場合 Web ページでは無く、画像、本文などである) のみ作成することで Web サイトを作成・更新することができる。また、サイト管理者がサーバ内に CMS を構築すれば、複数のコンテンツ管理者が Web ブラウザからサイトを更新することができる。このような利点から今後サイト構築において CMS は広く一般化していくと考えたからである。

2　卒業研究

2.1　目的

卒業研究を行うことで、CMS についての知識を深め、自分が望む、世に出ている数多の CMS とは異なる CMS を開発するのが目的である。また、いかにして目的のソフトウェアを開発するかという開発手法、開発に関する経験を養うことも卒業研究の目的に含んでいる。

2.2　開発環境

開発環境を以下に示す。

- ハードウェア... ... ds3.fken(研究室内ネットワークに存在するマシン)
- OS... ... Vine Linux 4.0
- 開発言語... ... PHP5 [2, 3]
- データベース... ... MySQL 5 [4, 5]

3　CMS について

3.1　コンテンツを統合的に管理するということ

CMS で言う、コンテンツを統合的に管理する、ということについて考える。これはコンテンツ自身と他のコンテンツとの関係性を含めて管理する (保存、削除する) ということだと考えることができる。具体的に言うと、一つのエントリ[*1]とそのエントリが参照している画像、映像を並行に管理することであり、それはつまり単純に言えば「必要なものは全部纏めておく」ということである。

3.1.1　CMS の具体的な動作について

CMS はどういったソフトウェアなのか。具体的な動作を考えてみる。

まず最初に、CMS は単なるソフトウェアでしかなく、それ自体は Web サイトではない。Web ブラウザによって解釈、表示できるのは HTML のみである。よって CMS は何らかのタイミングでコンテンツを HTML として生成しなければならない。生成された HTML が Web ブラウザによって Web サイトとして表示されるのである。

その生成の動作として CMS を実行した時に生成する場合と、既に HTML を生成してあり、CMS を実行せずに Web サイトを表示する場合に分けられる。前者を動的生成型、後者を静的生成型と定義する。

3.2　動的生成型

動的生成型というのは、CMS を実行た時 CMS 内部で HTML を生成し、それを Web サイトとして表示するものである。例として Nucleus CMS[*2]がある。これはユーザーがアクセスするのは CMS 内のファイル (index.php など) だけで、引数などによって表示する Web ページをその時生成し、表示する。

動的生成型の利点

- Web サイトに変更を加えた際に自動的に Web サイトが更新される。(Web サイト管理者が更新を意識しない)
- Web サイトのファイルが少なくて済む (メンテナンス性の向上)

動的生成型の欠点

- 表示する際に CMS が実行されることから、サーバーの負荷が静的生成型に比べて多い。
- アクセス先は常に特定のファイルであるので、SEO[*3]が難しくなる。
- CMS が実行されるサーバーの環境が変わった場合、対策を講じないと CMS が動かず、サイトを表示できない可能性がある。

[*1] サイト内でコンテンツとして表示される文字列。Entry、書き込みと同意である

[*2] ベルギーの Wouter Demuynck などによって作られた、MySQL と PHP によって動作するサーバ設置型のオープンソースな CMS。WeBlog 向け。ライセンスは GPL となっていて、ソースコードは無償で配布されている。

[*3] サーチエンジンオプティマイゼーション、(SEO)。ある特定の検索エンジンを対象として検索結果でより上位に現れるようにウェブページを書き換えること。また、その技術。サーチエンジン最適化とも言われる。

3.3 静的生成型

静的生成型というのは、CMS が Web サイトとなる HTML ファイル郡を管理者の操作によって一括で生成することでサイトを構築するものである。例として MovableType[*4]がある。これはユーザーが Web サイトとしてアクセスするのは生成後の HTML ファイル郡 (index.html など) である。

静的生成型の利点

- ユーザがアクセスするのは HTML ファイルであるので、サーバの負荷が動的生成型に比べて少ない。
- HTML ファイルはそれぞれが内容を特定できるファイル名を持っているので SEO として効率的である。

静的生成型の欠点

- サイトに変更を加えた際、必ず HTML ファイルを生成 (再構築) しなければいけない。
- HTML ファイルが増え、サーバの保存領域の圧迫やメンテナンス性の低下を招く。

3.4 現在最適と考えられる生成方法は

では動的生成型・静的生成型それぞれの利点・欠点を踏まえてどちらのほうが現在最適と考えられるのか。私は動的生成型ではないかと考えている。理由は、現在の高性能なサーバー環境、ネットワーク環境において負荷はそれほど問題ではなく (勿論極力少なくしなければいけないが)、それよりも管理者の作業に関するレスポンス (静的生成型の再構築待ち時間など) が速いほうが良いと考えるからである。また、動的生成型の欠点として挙げた SEO に関しても例えば Nucleus CMS の FancyURL[*5]のようにアドレスを偽装することで Web サイトが HTML ファイルで存在するように装うなどの手段を取ることで SEO を行うことが可能になっているからである。

[*4] シックス・アパート社が開発・提供する WeBlog ソフトウェア。プラグインによってあらゆる機能を拡張することが可能であり、WeBlog ソフトウェアとしては定番的存在である。

[*5] バージョン 2.0 からグローバル設定において設けられたオプション。URL を http://example.org/index.php?itemid=1234 から http://example.org/item/1234 に変えることができる。

4　現在ある CMS の分類と開発する CMS の理想像

4.1　現在ある CMS を分類する

CMS と言っても様々な種類がある。それらを以下に列挙し、その違いを考える。

- 汎用型... ... プラグインを導入することで、どんな形の CMS としても動作する [6]。 (XOOPS[*6]、DotNetNuke[*7]など)
- 機能特化型... ... 日記 (WeBlog) など、特定のサイト形式を作成するために特化した CMS [7]。(MovableType、Nucleus CMS など)
- 一部管理型... ... 既存のサイトのコンテンツ部分のみを管理する CMS [8]。(CMS Designer[*8]など)

4.2　現在ある CMS の問題点

こうして分類したとき、現在の CMS には大きな問題点があると考えた。それは全ての CMS が中途半端であるということである。

一つの Web サイトを構築するとき、大まかなイメージを組み立てる。そしてそれを CMS を用いて構築しようとしたとき、それぞれの CMS の型によって壁にぶつかってしまうのである。

汎用型はプラグインを入れなければただのプラグイン管理システムに過ぎず、構築したい Web サイトに合わせてプラグインを調べ、選別して導入しなければいけない。これは容易ではない。同様の機能を持ったプラグインが複数ある場合、それらがどう異なるのか調べなければいけない。

機能特化型は特定の Web サイトのスタイルにとっては容易に Web サイトを構築することができるが、少しでもそのスタイルから離れた場合は適応することができない。例えば WeBlog として作った Web サイトに記事としてでは無く、プロフィールページなどを作りたい場合、(全く作れないわけではないが) 非常に困難である。

一部特化型は従来の Web サイトが既にあり、その一部分だけ管理する場合は有効だが、Web サイト全体を管理したいと考えたとき選択肢にさえならない。

4.3　開発する CMS の理想像

ここから、卒業研究として開発する CMS を考えることにした。まず私が CMS に求めるのは三点である。

- CMS は Web サイト全体を管理すべきである。
- CMS は自由に Web サイトを構築できるべきである。
- CMS は煩わしさを管理者に与えてはならない。

これらは、CMS とは「簡単で便利、面倒無く好きな Web サイトを構築できるシステム」であると考えているからである。

その結果、私は前述した三つの分類とは異なる、四つ目の分類へとたどり着いた。それは「パッケージ型」という発想である。

[*6] XOOPS（eXtensible Object Oriented Portal System、ズープス）は、GPL に基づいて開発された CMS。PHP-Nuke を元に日本人によって開発された。

[*7] 北米で開発された CMS。ASP.NET で記述されている。

[*8] コンテンツ部分のみを管理する、部品型 CMS。

「パッケージ型」とは、基本的には汎用型である。データベースにアクセスするなどの基幹システムにプラグインを導入することによって CMS を構築する。汎用型と異なるのは、プラグインを既に実装していることである。

例えと言うには少し異なるが、Linux のディストリビューションを考えてほしい。Linux のカーネルは資源の管理やハードウェアへのアクセスインタフェースなどしか提供せず、実際にユーザが操作するのはカーネル上で動作する各種ソフトウェアである。この Linux カーネルと各種ソフトウェア、ライブラリなどをニーズに合わせて纏めたものがディストリビューションである。[*9]

今回の「パッケージ型」というのはこれに近い発想である。作りたい Web サイトにあわせて必要最低限な機能を提供するプラグインと、その管理を行うシステムを纏めて一つのパッケージとして提供する。ユーザがさらに必要な機能があれば追加でプラグインを導入することで対応することができる。このようにすることで汎用型の煩わしさと機能特化型の制限された自由から解放されると考えた。次の章では設計した基幹システム「CiPS」とパッケージ「SysPoS」について詳しく説明する。

[*9] 勿論これだけの説明ではディストリビューションを説明しきれていないが、今回の例えとしてはこのようにとらえてほしい

5　開発したシステム

5.1　CMS include Pure Plugin System - CiPS とは

図 1　CiPS ロゴ

CiPS とは、「CMS include Pure Plugin System」の略で、純粋なプラグイン型の CMS であることを示している。(図 1:CiPS のロゴ)

これはつまり、基幹システム自身はデータベースアクセスやプラグイン管理などの管理機能しか持たず、その上に実装されたプラグインが具体的な CMS の機能を提供することを意味している。

この利点として以下のことが挙げられる。

- プラグインを入れ換えることで必要な CMS を構築することができる。
- プラグインにバグがあった場合、CMS 全体を変更することなく、対象のプラグインを差し替えるだけで対策することができる。
- CiPS を拡張・改良するだけで CMS 全体の性能を向上させることができる。

5.1.1　CiPS のファイル・ディレクトリ構造

CiPS は現在自身の存在するディレクトリ下に二つのディレクトリを持つ。

- layout …… Web ページレイアウトディレクトリ
- plugins …… プラグインディレクトリ

5.2 SysPoS とは

図 2 SysPos ロゴ

SysPoS とは、「Systematic Portal site management System」の略で、組織的な Web サイト群[*10]を管理する為のシステムの意味である。(図 2:SysPoS のロゴ) これは次に示すような独自機能によって、組織的な Web サイト郡の構築に特化した CMS である。

5.2.1 SysPoS の独自機能

SysPoS の独自機能として、以下の機能を実装している。

- 作成した複数サイト間での親子関係 (Site P-C Model) の保持
- 記事の共有 (TopDown/ButtomUp Entry)

5.2.2 Site P-C Model とは

CMS で複数のサイトを管理する場合、それぞれのサイトは独立しており、お互いに関係をもっていない。これは組織的な Web サイト郡を構築する際にはそれぞれの関係性を保持することができない。そこで SysPoS ではサイト同士の関係として親子関係を持つ。これを Site P-C Model(Site Parent-Children Model) と定義する。

図 3 を例にあげて説明する。サイト A はサイト B、サイト C の親 Web サイトであり、同時にサイト B、サイト C はサイト A の子 Web サイトである。また、サイト C はサイト D の親 Web サイトであり、同時にサイト D はサイト C の子 Web サイトである。この関係性が Site P-C Model である。

5.2.3 TopDown/ButtomUp Entry とは

Site P-C Model において親 Web サイトの記事を子 Web サイト全てに表示できればユーザは子 Web サイトのみ閲覧するだけで親 Web サイトの記事を知ることができる。また逆に、子 Web サイトの記事を親サイトで表示することができれば親 Web サイトを閲覧するだけで子 Web サイトの記事を知ることができる。それが TopDown/ButtomUp Entry である。TopDown/ButtomUp Entry それぞれの例を次に示す。

5.2.4 TopDown Entry の説明

TopDown Entry とは親 Web サイトの記事を自動的に子 Web サイトで表示する機能である。図 4 にあるように上にある親 Web サイトの記事を子 Web サイトで表示する。

[*10] 例えば、企業において全社向けサイトと、各部署ごとの複数のサイトが一つのサーバ内に存在しており、それがツリー型のように関係を持っているような状態のサイト郡

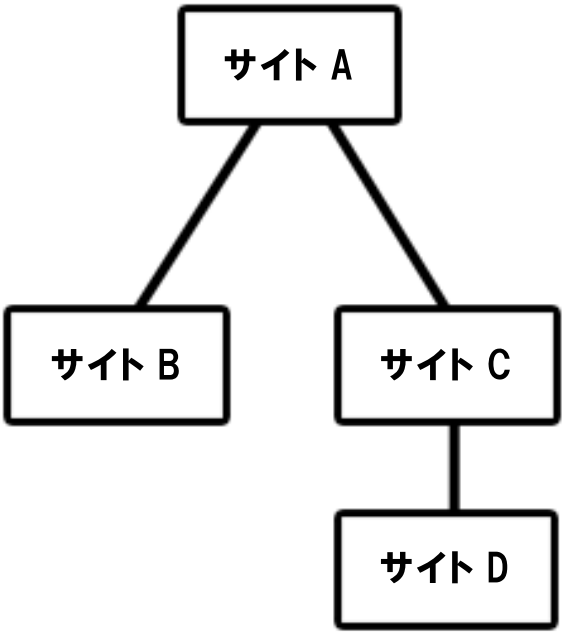

図 3　Site P-C Model

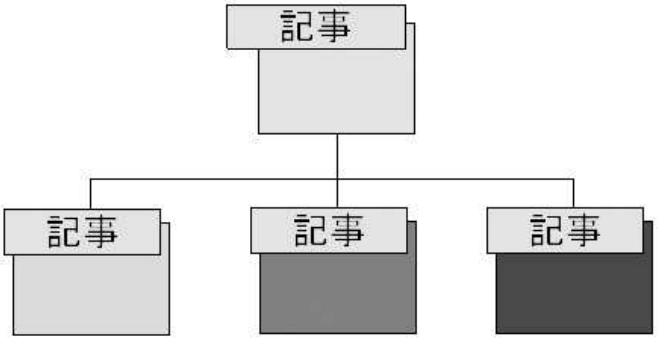

図 4　TopDown Entry

5.2.5　BottomUp Entry の説明

BottomUp Entry とは TopDown Entry の逆で、子 Web サイトの記事を自動的に親 Web サイトで表示する機能である。図 5 にあるように上にある子 Web サイトそれぞれの記事を親 Web サイトで表示する。

5.2.6　TopDown/ButtomUp Entry と TrackBack との違い

TrackBack という機能がある。これはある他者の Web サイトの記事に対して、その記事に関連する内容の記事を自分の Web サイトに作成したとき、他者の Web サイトの記事に自分の Web サイトの記事へのリンクを表示する機能である。これによって他者の Web サイトの記事を読んだとき、関連する記事としてユーザは自分の Web サイトの記事を知ることができる。

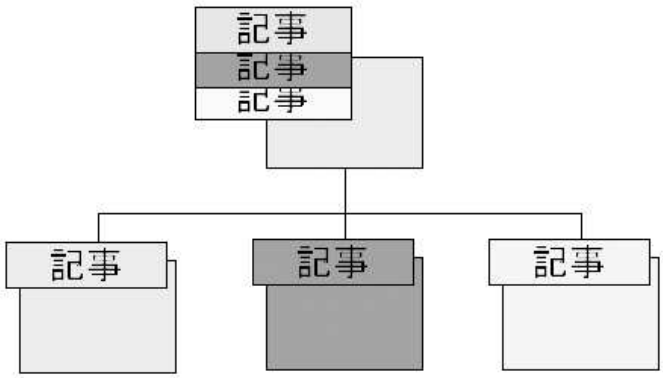

図 5　BottomUp Entry

これは他者の Web サイトに自分の Web サイトの記事を表示するという点では TopDown/ButtomUp Entry とよく似ているように感じられる。だが、TrackBack は他者の Web サイトに自分の Web サイトの記事を表示するのが本来の目的ではなく、他者の Web サイトの記事に対する自分のコメントを自分の Web サイトに残し、その関係をリンクとして示す機能である。

TopDown/ButtomUp Entry は TrackBack とは異なり、単純に親子関係を持っている他の Web サイトに自分の Web サイトの記事を表示する機能である。リンクを行う訳ではなく、自分の Web サイトの記事を他の Web サイト内部に表示し、それを含めて他の Web サイトを表現する機能である。

5.2.7　SySPoS を構成するプラグイン

- userManagerPlugin... ... ユーザー管理プラグイン
- siteManagerPlugin... ... 複数サイト管理プラグイン
- entryPlugin... ... エントリ管理プラグイン

6　SysPoS の具体的な動作

この章では SysPoS の具体的な動作について説明する。だが現在は SysPoS はデータベースからデータを読み出し、表示する部分しか実装されていない。よってこの章では Web サイトの表示を行った際の動作を説明する。

1. まずはじめに、index.php 上で各種設定が読み込まれ、プラグインの連結が行われる。
2. 次に、表示する Web ページのレイアウトの元となるレイアウトファイル (*.slayout) が読み込まれる。
3. 各プラグインがデータベースにアクセスし、取得したデータとレイアウトファイルを元に HTML の生成が行われる。
4. 生成が終わると Web ブラウザに Web サイトが表示される。

7　考察

7.1　利点

TopDown/BottomUp Entry を実装したことで、親子関係にある Web サイト同士の記事の共有を行うことに成功し、子 Web サイトの (親 Web サイトに関連のある) 記事を親 Web サイトで確認することができるのが利点である。

これにより子 Web サイトを見るだけで親 Web サイトの子 Web サイトユーザに知らせたい記事を見せることができる。また、親 Web サイトの記事も子 Web サイトのデザインで表示されるので子 Web サイトのデザインを崩すことなく表現することができる。

また、TopDown/ButtomUp Entry で表示される記事はあくまで記事を共有しているのみなので、元の記事を変更した際はそのままその記事を表示している Web サイトに反映される。

7.2　問題点

オブジェクト指向プログラミングをイメージしながら開発したことで、コードが冗長になってしまっている。また、時間の都合でオブジェクト指向プログラミングになりきっていない部分が多数存在する。もっと正しくオブジェクト指向プログラミングを行うことで、可読性に優れたコードになるだろう。

7.3　発展

今後 CiPS に対応したプラグインを開発することで、新たな機能を実装することが出来る。たとえば、お絵かき BBS の機能を実装したプラグインを開発し、draweBlog[*11] としてのパッケージにすることも可能である。また、他の CMS(例えば XOOPS) のプラグイン形式に対応できるように CiPS を改良することで大量にある既存の資産を活用し、機能拡張することが可能である。

他にも、現在の CiPS の構想では一つのサーバ内で動作するが、複数のサーバでそれぞれ動作する CiPS が相互連結することでサーバの垣根をこえたサイト構築も可能である。

[*11] draw WeBlog。絵を描くことに特化した WeBlog。造語。

8 まとめ

今回の卒業研究では CMS というもの、その役割、それによって管理者にもたらされるメリットなどを考え、自分自身が作りたい CMS のイメージを形付けるところから始まった。その時点から現在の CMS に対する不満があり、それを解決するためにはどのようなものにするべきなのか検討した。結果たどり着いたのがパッケージ型 CMS であり、CiPS と SysPoS であった。その後予想外の忙しさにかまけて進行が遅れ、満足にスケジュールをこなせずデスマーチに陥ったことや理想のクオリティに到達できなかったことなど非常に残念である。ただし最低限目的としていた機能を表現することができたのは嬉しい限りである。

CiPS、SysPoS は成長していくシステムである、と私は考える。その時々の技術や発想を吸収し、CiPS の改良や新しいプラグインの開発などを行うことでその時代に即したシステムへと変化を遂げることができる。それは私が行うこともできるが、私の手を離れて誰かが行うこともできる。ここに今日のオープンソースなどの開かれた開発における最大のメリットがあると考える。最初の開発者以外でも自由に改良・開発することができ、ときに最初の開発者では考えつかなかった機能や、開発できなかった機能を追加することが可能になる。そういった意味でひとり立ちできるシステムとして土台作りにはなったのではないか。

この論文・研究内容を知り、私以外の研究生がさらなる改良や機能を加えて SysPos、CiPS が他の CMS と肩を並べるソフトウェアとしてインターネット上で公開されるようになることを期待する。

謝辞

本論文執筆及び研究作業等、研究室での活動の際以外でもご指導及びご協力頂きました大垣 斉講師、藤井 信夫教授、情報安全工学実験室のメンバーの方々、その他見守ってくださった方々に深く感謝いたします。

参考文献

[1] ウィキペディア, http://ja.wikipedia.org/
[2] 日本 PHP ユーザ会, http://www.php.gr.jp/
[3] [改訂版] PHP ポケットリファレンス (Pocket reference) (単行本), 技術評論社 (ISBN:978-4774125022)
[4] MySQL AB, http://www.mysql.com/
[5] MySQL 入門以前, 毎日コミュニケーションズ (ISBN:978-4839916572)
[6] DotNetNuke, http://dotnetnuke.jp/
[7] ウィキペディア, http://ja.wikipedia.org/
[8] CMS Designer, http://cms.al-design.jp/

付録 A　付録

A.1　要件定義書

SysPos 考案時に作成した要件定義書である。

0. 用語定義

CMS
　コンテンツマネジメントシステムの略です。

システム
　本文中では開発する CMS を意味します。

管理
　作成・修正・削除を意味します

デザインテンプレート
　サイトのデザインに関するもの（スタイルシート、レイアウトなど）をサイト
　ごとに纏めたものです。

1. 開発するソフトウェアについて

開発するソフトウェアは CMS(コンテンツマネジメントシステム) です。サーバー内で動作し、
インターネットを用いてサイトを管理することができます。

2. 対象

インターネットへの接続が可能なコンピュータを持ったユーザーで、管理者登録された人が対象です。

作成したサイトに関しては、インターネットへの接続が可能なコンピュータを持った
ユーザーであれば誰でも閲覧することができます。（ただし、作成したサイトが
関係者以外の閲覧を禁じている場合は見ることができません）

3. 開発環境

Vine Linux、PHP,mySQL を用いて開発します。

4. 機能内容

・複数のサイトを管理することができます。
・複数の管理者でサイトを管理することができます。
・管理者を登録・権限設定・削除することができます。
・記事を管理することができます。
・画像をアップロードすることができます。
・記事の共有化をすることができます（5. 特有機能参照）。
・デザインテンプレートで簡単にデザインを管理することができます（5. 特有機能参照）。
・作成したサイトに閲覧制限をかけることができます。

5. 特有機能

記事の共有化として登録した他サイトへ記事へのリンク、概要を表示することができます。
これはサーバー内外を問いません。ただし、同じシステムを使用して管理しているサイトに限ります。
技術的には Trackback Ping と同じ方法を利用するか、もしくは独自に規格を制定します。

複数のサイトを管理しても、混乱しないようにデザインテンプレートをわかりやすく構成します。

A.2　コーディング規約

コーディング規約
last update:2007/05/30

アンダースコア（_）は原則使用しない。ただし、メンバ、定数には使用する。

関数名

関数名は先頭小文字で単語ごとに大文字で表記する。
例）function showPage();

変数名

変数名は先頭小文字で単語ごとに大文字で表記する。
メンバ変数には先頭にm_をつけ、次の文字を大文字にして表記する。
例）\$userData

\clearpage

A.3　SQL データ構造

A.3.1　database syspos

```
+------------------+
| Tables_in_syspos |
+------------------+
| ENTRY            |
| SITE             |
| USER             |
+------------------+
```

A.3.2　table ENTRY

```
+------+----------+--------------+--------+--------+------+
| id   | title    | body         | author | siteid | link |
+------+----------+--------------+--------+--------+------+
|    0 | あああ   | いいいい     |      0 |      0 |    0 |
|    1 | test     | test-test    |      1 |      0 |    0 |
```

```
|    2 | link     | tertest-test |      0 |      1 |    1 |
|    3 | nonlink  | tertest-test |      0 |      1 |    1 |
|    4 | nonlink  | tertest-test |      0 |      1 |    0 |
|    5 | nonlink  | tertest-test |      0 |      0 |    1 |
|    6 | bottomup | tertest-test |      0 |      2 |    2 |
+------+----------+--------------+--------+--------+------+
```

A.3.3 table SITE

```
+------+---------------+--------+--------+-------+
| id   | sitename      | enable | parent | child |
+------+---------------+--------+--------+-------+
|    0 | syspos        |      1 | 1      | 2     |
|    1 | syspos_parent |      1 | -1     | 0     |
|    2 | syspos_child  |      1 | 0      | -1    |
+------+---------------+--------+--------+-------+
```

A.3.4 table USER

```
+------+-------+
| id   | name  |
+------+-------+
|    0 | hogeA |
|    1 | hogeB |
+------+-------+
```

A.4 ソースコード

A.4.1 index.php

SysPoS 操作用プログラム

```
<?php
header('Content-Type: text/html; charset=EUC-JP');
header('Content-Language: ja');

require_once("cips/base_core.php");
require_once("cips/cips_core.php");
require_once("cips/plugins/entry/index.php");
require_once("cips/plugins/sitemanager/index.php");
require_once("cips/plugins/usermanager/index.php");

$sqlset = new CIPS_cMySqlProperties()$sqlset->m_dbName = "syspos";
$sqlset->m_serverAddr = "localhost";
$syspos = new CIPS_cSystem($sql,"SYSPOS");



$syspos->createDB();
$plgin = new cEntryPlugin($sql);
$syspos->addPlugin($plgin);
$plginSM = new cSiteManagerPlugin($sql);
$syspos->addPlugin($plginSM);

$syspos->showPages($_GET);

?>
```

A.4.2 cips/base_core.php

CiPS 基幹コアプログラム

```
<?php
// (Static) Class

class CIPS_cStatic{
function output($str){
if (is_object($str)){
var_dump($str);
}else{
print($str . "<br />");
}
}
function getDirList($str){
$dirlist = array();

if ($dir = opendir($str)){
while(($file = readdir($dir)) !== false){
if ($file != "." && $file != ".." ){
if (is_dir("$str$file/")){
$dirlist[] =  $file;
}
}
}
closedir($dir);
}
return $dirlist;
}
}

class CIPS_cMySqlProperties{
var $m_dbName;
var $m_serverAddr;
var $m_serverPort;
var $m_userName;
var $m_password;
}

class CIPS_cMySql{
var $m_id;
```

```
var $m_setting;

function CIPS_cMySql($set){
$this->m_setting = $set;
$this->connect();
}

function connect(){
$id = mysql_pconnect($this->m_setting->m_serverAddr . ":" . $this->m_setting->m_serverPort,$this->m_set
if (!$id ) die("not connection!");
$this->m_id = $id;
}

function isConnect($addmsg = ""){
if (!$this->m_id){
CIPS_cStatic::output("not connected DB error in " . $addmsg);
return false;
}
return true;
}

function query($query){
if (!$this->isConnect()) return false;
$queryid = mysql_query($query,$this->m_id);

if (mysql_num_rows($queryid) == 0){
   return false;
}

$objs = array();

while($obj = mysql_fetch_object($queryid)){
   $objs[] = $obj;
}

mysql_free_result($queryid);
return $objs;
}

function createDB(){
if (!$this->isConnect()) return false;

return $this->query("create database". $this->m_setting->m_dbName . ";");
```

```
}

function selectDB(){
if (!$this->isConnect()) return false;

$ret = mysql_select_db($this->m_setting->m_dbName);

return $ret;
}

function searchTable($tablename){

if (!$this->isConnect()) return false;

$m_dbName = (string)$this->m_setting->m_dbName;

$sql = "show tables like '" . $tablename . "';";

$ret = $this->query($sql);

return ($ret != false);

}

function createTable($tablename,$tableColumn){
if (!$this->isConnect()) return false;


$columnstr = "";

$columncount = count($tableColumn);

$columnstr = $columnstr . $tableColumn[0][0] ." ". $tableColumn[0][1];

for($i=1;$i<$columncount;$i++){
$columnstr = $columnstr . "," . $tableColumn[$i][0] ." ". $tableColumn[$i][1];
}

$this->query("create table $tablename ($columnstr);");
}
}
?>
```

A.4.3 cips/cips_core.php

CiPS コアプログラム

```
<?php

class CIPS_cSystem{
var $sql;
var $prefix;
var $plugins = array();
var $layoutdir;

function CIPS_cSystem($sql,$prefix,$layoutdir=""){
$this->sql = $sql;
$this->prefix = $prefix;
if ($layoutdir == ""){
   $layoutdir="default";
}
$this->layoutdir = "./cips/layout/$layoutdir";
}

function loadPlugins(){
 $dirlist =
CIPS_cStatic::getDirList("./cips/plugins/");

 foreach ($dirlist as $pluginName){

 }

}


function addPlugin($plugin){
 $this->plugins[] = $plugin;
}

function replacePlugin($plugin){

}

function createDB(){

if (!$this->sql->isConnect("createDB")){
```

```
}

}


function showPages($args){
 if ($args['page'] == ""){
    $args = "index";
 }

$layoutfile = $this->layoutdir . "/" . $args['page']. ".slayout";

if ( file_exists($layoutfile) ) {
$layout = file_get_contents($layoutfile);
}else{
$layout = file_get_contents($this->layoutdir."/default.slayout");
}

 foreach($this->plugins as $value){
$layout = $value->replaceLayoutTag($args,$layout);
 }

 print($layout);
}
}


class CIPS_cPluginTableBase{
var $m_tableTitle;
var $m_tableColumn = array(array(),array());
var $m_parentPlugin;

function createTable(){

if (!$this->m_parentPlugin->sql->isConnect("plugin createTable")){
}

// table check and create

if (!$this->m_parentPlugin->sql->searchTable($this->m_pluginTableTitle)){
$this->m_parentPlugin->sql->createTable($this->m_pluginTableTitle,$this->m_pluginTableColumn);
}
```

```
}
}


class CIPS_cPluginBase{
var $sql;
var $m_pluginName;
var $m_pluginVersion;
var $tables = array();
var $chainPlugins = array();

function CIPS_cPluginBase($sql){
 $this->sql = $sql;

 return;
}

function getName(){ return $m_plugnName; }

function getData($index){
 return;
}

function chainSystem($system){
 $this->sql = $system->sql;
}

function addTable($table){

$table->m_parentPlugin = $this;
$this->tables[] = $table;


}

function showParentMessage(){

}

function replaceLayoutTag($args,$layout){
// make html code;
// return preg_replace(/TAGS/i,htmlcode,$layout);
```

```
}
function addChainPlugin($plugin){
 $chainPlugins[] = $plugin;
}
function releaseChainPlugin($plugin){
 if (in_array($plugin,$chainPlugins)){

 }
}

function configure(){
 return;
}
}
?>
```

A.4.4 cips/plugins/entry/index.php

エントリプラグイン

```
<?php

class cEntryPlugin extends CIPS_cPluginBase{

      function cEntryPlugin($sql){

    $this->CIPS_cPluginBase($sql);

    $table = new CIPS_cPluginTableBase();

    $table->m_pluginTableTitle = "ENTRY";
    $table->m_pluginTableColumn[0][0] = "id";
    $table->m_pluginTableColumn[0][1] = "int";
    $table->m_pluginTableColumn[1][0] = "title";
    $table->m_pluginTableColumn[1][1] = "text";
    $table->m_pluginTableColumn[2][0] = "body";
    $table->m_pluginTableColumn[2][1] = "text";
    $table->m_pluginTableColumn[3][0] = "author";
    $table->m_pluginTableColumn[3][1] = "int";
    $table->m_pluginTableColumn[4][0] = "siteid";
    $table->m_pluginTableColumn[4][1] = "int";
    $table->m_pluginTableColumn[5][0] = "link";
    $table->m_pluginTableColumn[5][1] = "int";

    $this->addTable($table);
    $table->createTable();

      }

      function replaceLayoutTag($args,$layout){
    $temp = '';
    $sql = "select * from ENTRY where siteid='". $args['siteid'] ."';";
    $objs = $this->sql->query($sql);
    foreach( $objs as $obj ){
        $temp= $temp . "<h3>". $obj->title ."</h3>\n";
$temp = $temp ."<div>" . $obj->body . "</div>\n";

$usermanager = new cUserManagerPlugin($this->sql);
$obj = $usermanager->getData($obj->author);
```

```
$temp = $temp ."<div>" . $obj->name . "</div>\n";

    }
    $layout = preg_replace('/<%BLOCK_ENTRY%>/i',$temp,$layout);

    $temp = "";

    $objs = $this->sql->query("select * from ENTRY where link='1';");
    foreach( $objs as $obj ){

        $sitemanager = new cSiteManagerPlugin($this->sql);
        $siteinfo = $sitemanager->getData($args['siteid']);

        $parent = split("/",$siteinfo->parent);

if ($parent[0] == $obj->siteid){
    $temp = $temp . "<h3>". $obj->title ."</h3>\n";
    $temp = $temp ."<div>" . $obj->body . "</div>\n";
    $usermanager = new cUserManagerPlugin($this->sql);
    $obj = $usermanager->getData($obj->author);
    $temp = $temp ."<div>" . $obj->name . "</div>\n";
}
    }

    $layout = preg_replace('/<%BLOCK_TOPDOWNENTRY%>/i',$temp,$layout);

    $temp="";

    $objs = $this->sql->query("select * from ENTRY
    where link='2';");

    foreach( $objs as $obj ){
$sitemanager = new cSiteManagerPlugin($this->sql);
$siteinfo = $sitemanager->getData($args['siteid']);
$child = split("/",$siteinfo->child);

if ($child[0] == $obj->siteid){
    $temp= $temp . "<h3>". $obj->title . "</h3>\n";
    $temp = $temp ."<div>" . $obj->body . "</div>\n";

    $usermanager = new cUserManagerPlugin($this->sql);
    $obj = $usermanager->getData($obj->author);
    $temp = $temp ."<div>" . $obj->name . "</div>\n";
```

```
}
    }
    $layout = preg_replace('/<%BLOCK_BOTTOMUPENTRY%>/i',$temp,$layout);

    return $layout;
      }
}
?>
```

A.4.5 cips/plugins/usermanager/index.php

ユーザー管理プラグイン

```
<?php

class cUserManagerPlugin extends CIPS_cPluginBase{

      function cUserManagerPlugin($sql){

    $this->CIPS_cPluginBase($sql);

    $table = new CIPS_cPluginTableBase();

    $table->m_pluginTableTitle = "USER";
    $table->m_pluginTableColumn[0][0] = "id";
    $table->m_pluginTableColumn[0][1] = "int";
    $table->m_pluginTableColumn[1][0] = "name";
    $table->m_pluginTableColumn[1][1] = "text";
    $table->createTable();

      }

      function getData($userid){

    $sql = "select * from USER  where id='" . $userid ."';";

    $obj=$this->sql->query($sql);
    return $obj[0];
      }

      function replaceLayoutTag($args,$layout){
    $temp;
    $sql = "select * from SITE where id='" . $args['siteid'] . "';";
    $obj = $this->sql->query($sql);
    $temp= $obj[0]->sitename;
    $layout = preg_replace('/<%BLOG_TITLE%>/i',$temp,$layout);

    return $layout;
      }

}
```

```
?>
```

A.4.6 cips/plugins/sitemanager/index.php

サイト管理プラグイン

```
<?php

class cSiteManagerPlugin extends CIPS_cPluginBase{

      function cSiteManagerPlugin($sql){

    $this->CIPS_cPluginBase($sql);

    $table = new CIPS_cPluginTableBase();

    $table->m_pluginTableTitle = "SITE";
    $table->m_pluginTableColumn[0][0] = "id";
    $table->m_pluginTableColumn[0][1] = "int";
    $table->m_pluginTableColumn[1][0] = "sitename";
    $table->m_pluginTableColumn[1][1] = "text";
    $table->m_pluginTableColumn[2][0] = "enable";
    $table->m_pluginTableColumn[2][1] = "int";
    $table->m_pluginTableColumn[3][0] = "parent";
    $table->m_pluginTableColumn[3][1] = "text";
    $table->m_pluginTableColumn[4][0] = "child";
    $table->m_pluginTableColumn[4][1] = "text";

    $this->addTable($table);
    $table->createTable();

      }

      function getData($siteid){

    $sql = "select * from SITE where id='" . $siteid ."';";

    $obj=$this->sql->query($sql);
    return $obj[0];
      }

      function replaceLayoutTag($args,$layout){
    $temp;
    $sql = "select * from SITE where id='". $args['siteid'] . "';";
    $obj = $this->sql->query($sql);
```

```
    $temp= $obj[0]->sitename;
    $layout = preg_replace('/<%BLOG_TITLE%>/i',$temp,$layout);

    return $layout;
      }
}

?>
```