

シラバスシステム引継文書 (2008-2009) (実装概要編)

シラバスシステム開発チーム 2008 M1 小宮山

最終更新日:2009 年 4 月 13 日

本文書の概要

著者の独断と偏見による、シラバスシステムにおける変更点と今後の展望を記述するチラシの裏。

1 2008 年度システムにおける変更の概要

1.1 実装

perl から php5 に実装が変更された。これにより、処理系における perl,bash,java,xalan,ruby の部分が php5 に統一された。しかし、perl システムをサブシステムとして並行運用する（互換性確保）ために、単体の設計では不要な仕様が存在している。これについては、文字の処理をはじめとしシステム全体の設計にまで波及している。

1.2 サーバ側の設定

php の設定についてデフォルトから変更を行っている。php.ini,.htaccess

1.3 tex のパッケージとフォント

UTF の文字を pdf(dvi も) に出力するために otf パッケージを用いている。

Vine の texmacro-otf ではうまくいかなかった（フォント）ため、<http://mytexpert.sourceforge.jp/>を参考にし、土村さん (<http://www.nn.ij4u.or.jp/~tutimura/tex/ptex.html>) の ptex3 を alpha にインストールしている。具体的には、tetex-texmf-3.0-1.noarch.rpm(tetex-texmf-3.0-1) と Vine4-ptetex3-20071003-1.i386.rpm(ptetex3-20071003-1) である。

1.4 XML エンコーディング

EUC-JP から UTF-8 に変更した。

2 問題点の概要と展望

ここでは 2008 年度のシステムにおいて発生または発覚した、あるいは改善されていない点について述べる。

3 文字データ

3.1 2008 年度システム内部の文字データ変遷

フォームから送信された文字列の文字コードは入力環境依存であるために、php で受取る段階で UTF-8 に変換している。これは php の mb_convert_encoding 関数を用いているが、期待の動作を得るために php の環境設定をデフォルトから変更している。また、後述の互換性問題により&、>、<、"、'、_を全角に置き換えている。

次にこれらの文字列を用いて XML を作成し保存する。XSLT をもちいてこの XML から tex ファイルを作

成するのだが、そのままでは対応する PDF、dvi 用のフォントがなく、期待する出力が得られない。そのため、otf パッケージの `\UTF{UTF-8 の文字コード}` という命令に変換する必要がある。ただし、出力時の文字の間隔や、tex の禁則処理のために、アルファベットと括弧や句読点などはそのままにしておく。また、この時点で tex における特殊文字の置換も行っている。これらの処理を行う関数 `str2otf` を実装している。

また XSLT を行うために、ここではじめて XML として扱われる。(半角空白の削除に注意)

XSLT を用いて作成された tex ファイルの文字コードは UTF-8 になっているが、現状では特に意味は無い。

その後 dvi、pdf と生成されていく。ここでフォントが埋め込まれるわけではないことに注意すべきである。

実際に該当するのはかなり特殊な文字だけであるが、Adobe-Japan のフォントが用いられる。この埋め込みはライセンスに抵触する恐れが高いため埋め込みは行っていない。これらのフォントである Adobe-Japan-1-5 程度は Adobe Reader 7.0 以上で閲覧、あるいは不足フォントとして自動ダウンロードが可能である。また Mac OSX の PDF ビューワ (何か忘れた) も対応していた。非対応のビューワでは当該文字が表示されない問題がある。

3.2 問題となる文字

文字 エスケープ等 現状

xml

<	<	<
>	>	>
&	&	&
"	"	"
'	'	'

tex

#	\#	\#
\$	\\$	\\$
%	\%	\%
&	\&	&
-	_	—
{	\{	\{
}	\}	\}
\	\$\backslash\$	\$\backslash\$
-	{-}	未対応
~	\~{}	\~{}
^	\^{}{}	\^{}{}
*	\$*\$	\$*\$
	\$	\$

括弧や句読点等 そのまま そのまま

php 文字列

```
'    \'
"    \"
\\   \\\n
t    \t
n    \n
$    \$
r    \r
```

3.3 文字における諸問題

3.3.1 入力

フォームから送信された文字列を php で受取ったときに UTF-8 に変換している。

フォームから送信されるデータは、クライアントの環境、主にブラウザに依存する。ここで文字コードの変換の精度や繰り返すデメリットを考えれば、システム内で文字コードについて一貫性を確保した方が良い。また、ユーザの意図した文字を扱えることを考慮すると、カバーする文字種から UTF-8 が適していると考えられる。

3.3.2 旧 perl システム互換

一部の半角文字を入力の際で全角に置き換えている問題である。置き換えられている文字は xml で文字実体参照で表現されるべき `&`, `<`, `>`, `"`, `'` と `_` である。互換性を考慮しなくてよい 2009 年度システム以降では改善されるべき点である。

3.3.3 XML 処理

前述の文字実体参照と空白の扱いである。これらは適切に扱われるようにすべきである。

3.3.4 tex 組版における制約

tex における特殊文字、禁則処理、otf パッケージの使用を考慮しなくてはならない問題である。現状では自作関数 `str2otf` で対処しているが、前述の文字実体参照を考慮すると実装を変更する必要がある。また、otf パッケージの命令に変換する部分では UpTeX, UplaTeX (現状では 版) 等を用いることでそのまま扱えるようにするのが望ましい。

4 来年度以降への対応策提案

4.1 Session 管理

現在のシステムでは旧来のシステムを元に行っているため、セッション管理が不十分である。

php で実装する利点を活かし、ログイン情報をブラウザを閉じるまで保持するようにセッションを用いることを推奨する。

4.2 コーディング

変数と出力部分の分離を行う等、読みやすいコーディングを行うべきであると考えられる。また、アーキテクチャを見直しにより変数の整理が可能だと考えられる。

4.3 セキュリティ

ファイル名をはじめ、ログイン時のエラーメッセージ等からユーザ名が存在するかどうか確認可能である。一般的には重大なセキュリティホールといっても過言ではないが、運用を考えると本件の重要度は低いと考えられる。

しかしシステム全体について、セキュリティの再検証を行う必要性は十分にあると考えられる。

4.4 php6.0

Unicode が扱えるようになる。関数が UTF-8 に対応するため、現状のような処理や設定などが不要になり、システムのメンテナンス性が向上できると考えられる。

4.5 UpTex/UplaTex

UTF で記述し、そのまま組版が可能。ただし現在 版。

4.6 PDFLib

php の有償ライブラリ。php から直接 PDF を作成できる。上位版ではテンプレートの PDF に文字列を流し込むことが可能である。