

WebアプリケーションのCAS対応

熊本大学総合情報基盤センター
計算機援用教育研究部門
兼
eラーニング推進機構アプリケーション開発室
准教授 永井 孝幸

内容

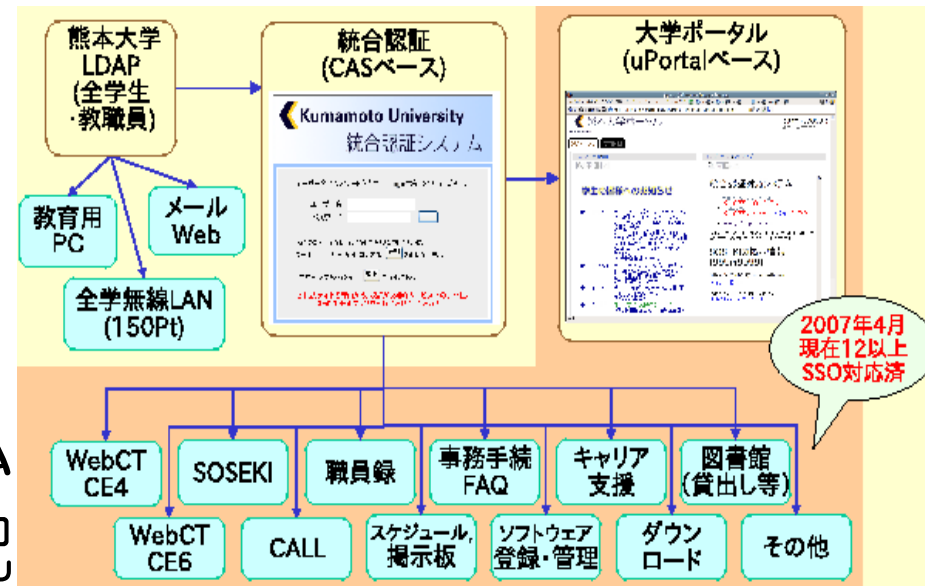
-
- CAS対応アプリケーションの紹介
 - 熊本大学の事例
 - CAS対応サイト・アプリの構築方法
 - <http://www.jasig.org/cas/client-integration/cas-ifying-apps>

熊本大学でのCAS利用

- 大学ポータルを中心とした情報サービス

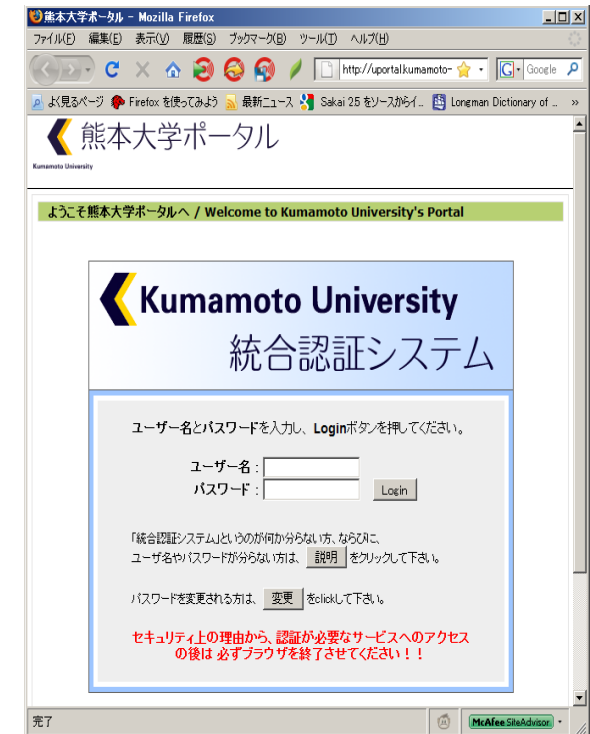
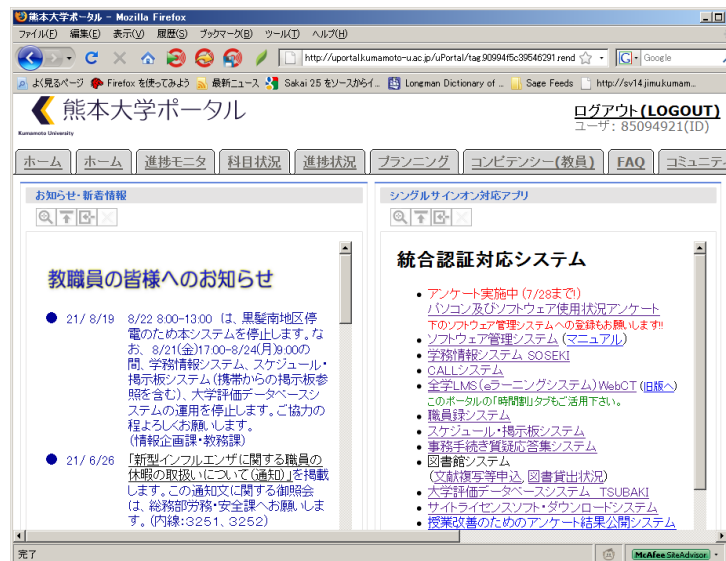
- uPortal
- 全学LMS
- 学務情報システム
- ソフトウェア管理システム
- サイトライセンスソフト配布システム
- 等々

詳しくはこちらのページ

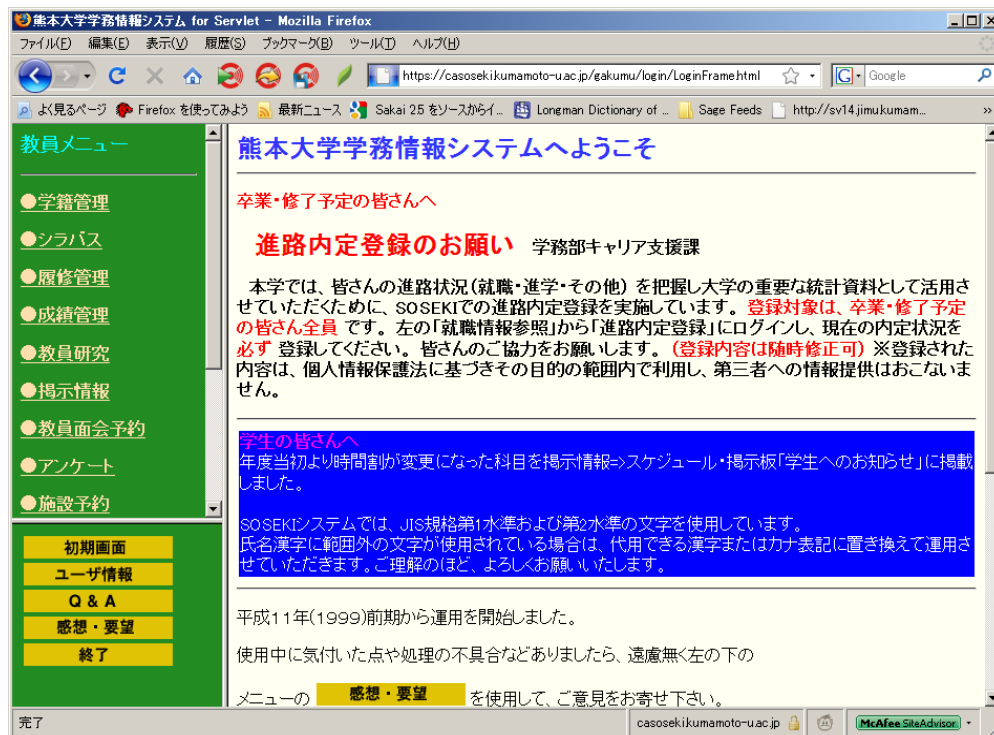


大学ポータル

- <http://uportal.kumamoto-u.ac.jp>
- uPortalを利用
 - 利用者属性に応じた情報の提示
 - 一般学生、教職員、特定グループ
 - アクセス元IPに応じた情報の提示
 - 学内、学外



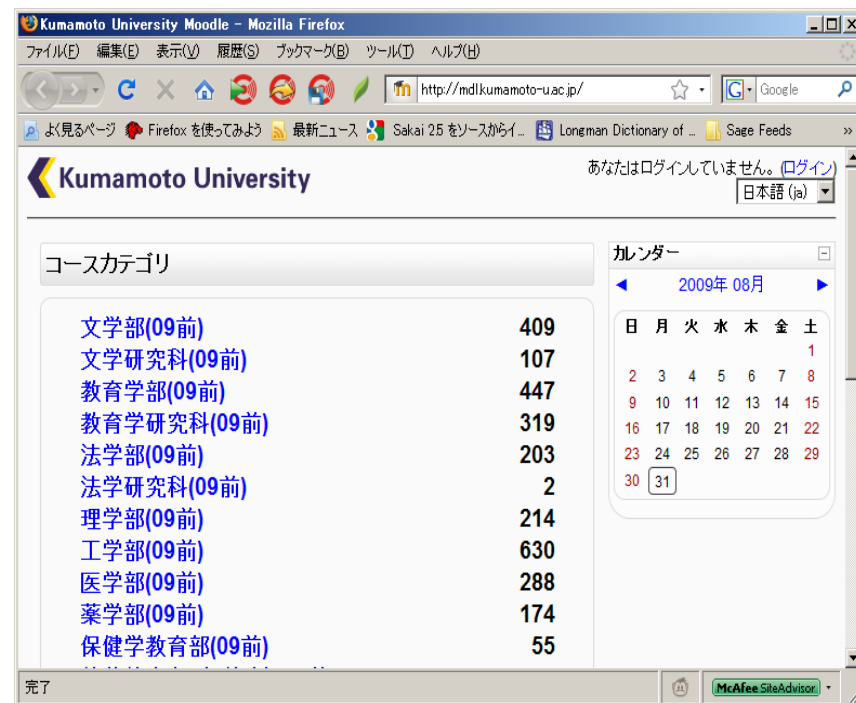
学務情報システム



- 履修登録
- シラバス
- 成績参照
- etc

全学LMS

- WebCT CE6
 - AutoSignonプロトコルを利用(詳細は中野)
- Moodle 1.9.5+



日常の利用手順

- 学生
 - 大学ポータル→時間割→全学LMS
- 教員
 - 事務連絡メール→全学掲示板
 - 課題提出メール→全学LMS

熊本大学ポータル
ログアウト (LOGOUT)
ユーザ: 85094921(ID)

ホーム ホーム 進捗モニタ 科目状況 進捗状況 プランニング コンピテンシー(教員) FAC

時間割

85094921 永井孝幸 (その他の施設)さんの時間割 | 2009年度前期

	月 (MON)	火 (TUE)	水 (WED)	木 (THU)	金 (FRI)	その他 (OTHERS)
1時限 08:40 - 10:10				情報基礎A 24 情報基礎A 24		
2時限 10:20 - 11:50			情報基礎A 18			
お昼休み (Lunch Break)						
3時限 12:50 - 14:20						

件名: gsbibp: Re: 検収完了しました
差出人: 宮川 崇仁 <noreply@mo.ielid.kumamoto-u.ac.jp>
返信先: このメールアドレス宛に返信しないでください <noreply@mo.ielid.kumamoto-u.ac.jp>
送信日時: 2009/08/25 11:13
宛先: 永井 孝幸

gsbibp » フォーラム » 検収と検収に関する問合せ-6(発注責任者→開発スタッフ) » 検収完了しました

Re: 検収完了しました
2009年 08月 25日(火曜日) 10:42 - 宮川 崇仁 の投稿

教授システム学専攻 笠井様

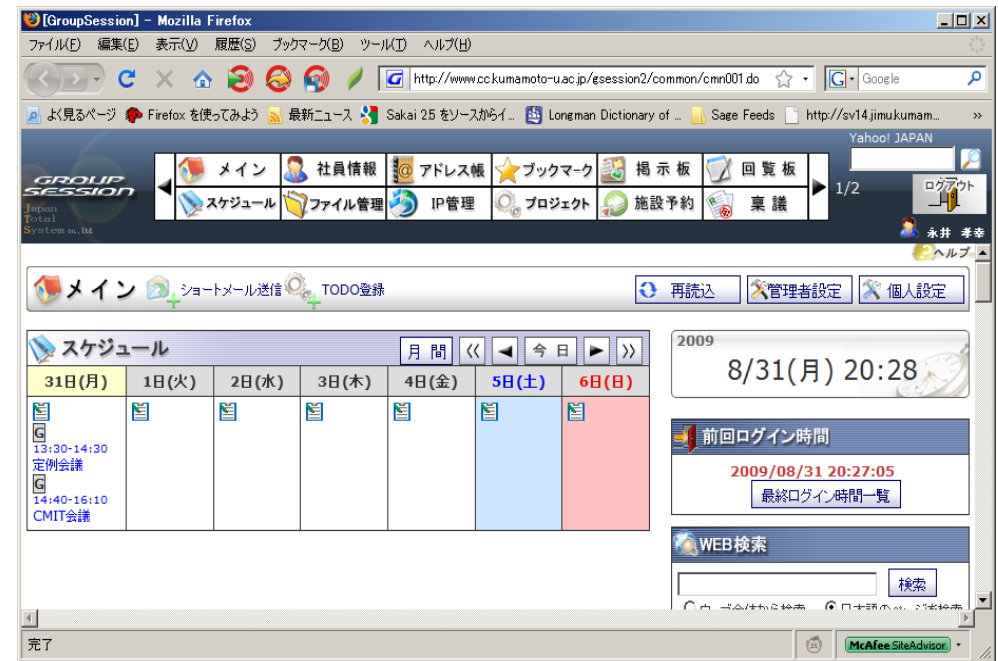
お世話になっております。eラーニング推進機構の宮川です。
ご確認いただき、ありがとうございました。

私自身至らない点などもあったかと思いますが、最後までありがとうございました。
宮下さま、朴さまにもどうかよろしくお伝えください。 [続きを読む](#) | [返信](#)

未読: 72 合計: 29589

総合情報基盤センター

- センターWeb (Drupal 6)
- グループウェア (GroupSession2)



CASサーバの構築

- CAS-3.3.3
 - tomcat上でサーブレットとして動作
 - ユーザ認証源:ActiveDirectory,JDBC,LDAP,etc
 - 使い方の詳細は[JA-SIGのページ](#)を参照
 - 各種設定 : web.xml,security,propertiesを編集
 - ログイン画面のカスタマイズ:JSPファイルを変更
- CAS²(CAS Square)
 - 名古屋大学にて開発
 - CASにauthorization機能を追加したもの

CAS対応アプリケーションの設定・構築

- CAS対応方法
 - PHP: CASライブラリを利用
 - J2EE: CASFilterを利用
- 元々CAS対応済みのもの
 - uPortal, Moodle, Sakai, Drupal, etc
- 自力で対応させる場合
 - Apache mod_auth_casの利用
 - アプリケーション個別対応(例: [MediaWiki](#))

PHP用CASクライアントライブラリ

- phpCAS
 - <http://www.ja-sig.org/wiki/display/CASC/phpCAS>
- サンプル
 - CASサーバをコンストラクタで指定
 - `phpCAS::forceAuthentication()` を実行
 - `phpCAS::getUser()` でユーザIDを取得
 - サンプルのソースコード

Java用CASクライアントライブラリ

- Java CAS Client
 - <http://www.jasig.org/cas/client-integration/java-client>
- サブレットコンテナのフィルターに設定
 - `edu.yale.its.tp.cas.client.filter.wrapRequest=true`
 - `HttpRequest`の`remoteUser()`でユーザID取得
 - または自前でユーザIDを取得
 - `session.getAttribute(CASFilter.CAS_FILTER_USER);`
 - サンプル

uPortalの設定

- uPortal-3.1.1
 - <http://www.jasig.org/uportal>より
uPortal-3.1.1-quick-start.tar.gzを入手
- 設定対象ファイル
 - uPorta-3.1.1-quick-start/apache-tomcat-6-0.18/webapps/uPortal/WEB-INF/web.xml
 - security.properties

uPortalの設定項目

- web.xml
 - cas.proxyUrl
 - cas.client.filter.validateUrl
- security.properties
 - logoutRedirect.root
 - channels.cLogin.CasLoginUrl
- サンプルファイルは[こちら](#)から

Sakaiの設定

- 参考資料

- <http://confluence.sakaiproject.org/display/~st eve.swinsburg/CASifying+Sakai>

- 設定箇所

- sakai-login-tool
 - casclientライブラリの追加、web.xmlの設定
- sakai.properties
 - CAS認証だけを使うように設定
- 設定サンプル

Moodleの設定

- 「ユーザー→認証→CASサーバ(SSO)」で設定
- CASサーバ設定
 - 最初は「マルチ認証」をYesにするのがよい
(CASがうまく動作しないとログイン不能になる)
- LDAPサーバ設定も必要
 - ホストURI、バージョン、LDAPエンコーディング
- 既存ユーザーをCAS認証に移行するには？
 - データベースを直接更新する必要あり

Drupalの設定

- CASモジュール
 - <http://drupal.org/project/cas>
 - phpCASライブラリも別途必要
- 設定項目
 - CASサーバ、CASログイン用URI
 - 認証対象ページ

ApacheのCAS対応

- mod_auth_cas
 - .htaccessのユーザ認証にCASが使える
 - 静的HTMLのCAS認証に利用可能
- .htaccessでの使い方
 - AuthTypeにCASを指定
 - require user, require groupを指定
 - グループメンバの指定
 - AuthGroupFileで指定
 - LDAP(ActiveDirectory)連携もできる (らしい)

mod_auth_casの導入

- Apache用モジュールのインストール
 - http://www.ja-sig.org/wiki/display/CASC/mod_auth_cas
 - `svn co https://www.ja-sig.org/svn/cas-clients/mod_auth_cas/tags/mod_auth_cas-1.0.8`
- 設定
 - httpd.conf
 - .htaccess
 - 設定サンプル

httpd.confの設定

- モジュールの登録
 - LoadModule auth_cas_module
- CASパラメータの設定
 - CASLoginURL
 - CASValidateURL
 - CASValidateServer
 - CASCookiePath
- CASCookiePathに指定したディレクトリには
httpdのwrite権限を与えること

.htaccessの設定

- AuthTypeにCASを指定
AuthType CAS
require user casuser
- 該当URIにアクセスするとCAS認証が行われる
 - REMOTE_USER属性にも反映される
 - BASIC認証対応CGI等のCAS対応に使える

自前でCAS対応する場合

1. J2EE準拠の場合→CASFilterを利用

HttpRequestのremoteUser属性を見る

2. 独自環境の場合

自前でCASサーバに問い合わせ

1. tiketパラメータを取得

`http://myapp/?ticket=XXX`

2. CASサーバのvalidateを呼び出す(HTTP GET)

`https://casserver/validate?service=myapp&ticket=XXX`

3. CASサーバのレスポンスをチェック

「yes ユーザ名」が返ってくるか確認

まとめ

-
- 主要なWebアプリはCAS対応済み
 - 主な言語用のCASライブラリも整備済み
 - BASIC認証を使う古い物はApacheで対応可