

# 熊本大学ポータル

中野 裕司, 喜多 敏博, 杉谷 賢一  
熊本大学総合情報基盤センター

2007年8月

## 概要

2006年度から、統合認証との連携で熊本大学ポータルのサービスを開始した。CASやuPortalをベースにカスタマイズしたシステムで、熊本大学の全学生・教職員が、一度の認証でそのユーザにあった全てのWebサービスを利用可能になる。2006年度中に10以上のWebサービスが熊本大学ポータルに対応した。

**keyword:** ポータル, CAS, uPortal, シングルサインオン

## 1 はじめに

本学は、学務情報システム SOSEKI、1300 台以上の教育用 PC システム、全学無線 LAN 等の IT 基盤整備、学習管理システム (LMS: Learning Management System) である WebCT の全学導入、情報リテラシー教育の充実、CALL (Computer Assisted Language Learning) 教材と LMS の連携、インストラクショナル・デザインを中核とした e ラーニング専門家養成インターネット大学院である教授システム学専攻の設立等、e ラーニング等、教育への IT 活用に積極的に取り組んできた [1]。

この取組の過程で、複数の e ラーニングシステムを活用する必要が生じ、2004 年度からは、SOSEKI と WebCT 間の履修データ連携 [2]、及び、SOSEKI、WebCT、教育用 PC、全学無線 LAN 等のユーザ ID とパスワードの共通化を NIS や LDAP の活用で実現してきた。

ユーザ ID とパスワードの共通化は、一組のユーザ ID とパスワードで多くのシステムへログインでき、パスワードを変更した場合も全てのシステムのパスワードが変更され、以前より利便性が向上した。しかしながら、多くのシステムを利用する場合、その都度ユーザ ID とパスワードをキーボードから入力しなければならない。

そこで、一回の認証だけで多くのシステムが利用可能になる「シングルサインオン (Single Sign-On, SSO)」を、2006 年度から、CAS (Central Authentication Service) [3] をベースに追加・変更を加え、全学的に導入、運用を開始した。

また、シングルサインオンに対応したシステムをシームレスに使えるように、同時に、uPortal [4] をベース

に追加・変更を加えた大学ポータルの全学運用を開始した。

本稿では、2006 年度運用を開始した熊本大学ポータルについて、その準備段階からの、各種アプリケーションの開発や対応等も含めて解説する。

## 2 利用方法

図 1 に、熊本大学ポータルの構造の概略を示す。ユーザは、熊本大学ポータルの Web アドレス

`http://uportal.kumamoto-u.ac.jp/`

に Web ブラウザでアクセスすると、Web ブラウザを起動してから初めてのアクセスであれば、図 1 中上の統合認証の画面が表示される。画面表示に従って、全学生・教職員が各々保有しているユーザ ID とパスワードを入力すれば、図 1 右上のようなポータル画面になる。

ポータル画面は、ログインした人物の所属や職等のプロパティ、及び学内からのアクセスかどうか等によって、変化する。表示されるタブや領域ごとに、そのユーザのためのお知らせや、シングルサインオンに対応した利用権限のあるシステムを再ログインすることなくシームレスに利用可能なリンクが置かれている。

図 1 に示すように、既存の各種システムや新規システムのシステムのシングルサインオン (CAS) への対応が順調に進み、2006 年度終了時点までに 12 システム以上が対応している。

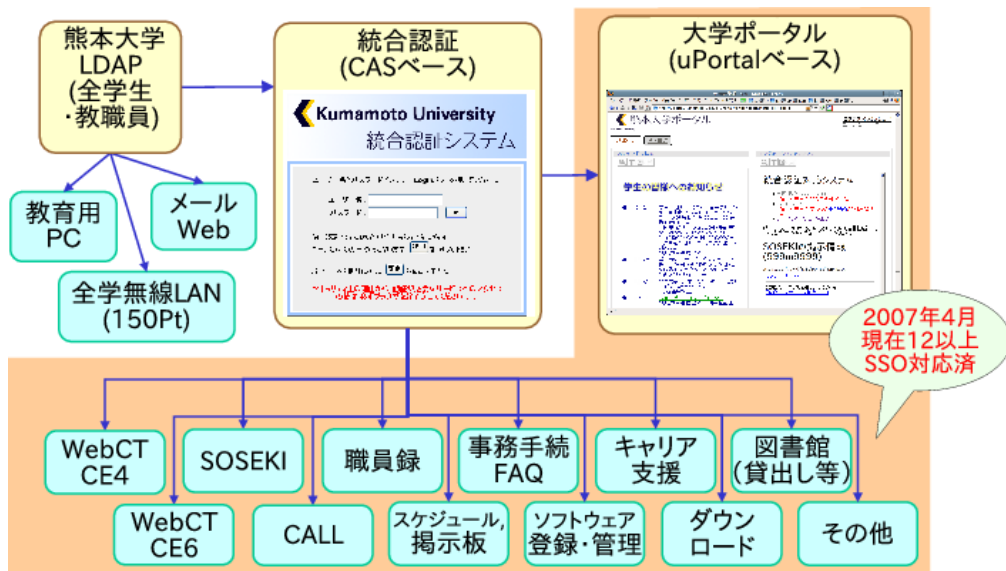


図 1: 熊本学ポータル概要

### 3 技術的背景

熊本大学ポータルは、統合認証として CAS[3] をベースに、大学ポータルとして uPortal[4] をベースにしている。これらのシステム (Web アプリケーション) について簡単に紹介する。

#### 3.1 CAS

CAS[3] は、Yale 大学で開発されたユーザ認証システムで、現在は JA-SIG プロジェクト [5] に開発が引き継がれている。

CAS のユーザ認証を簡単に以下に示す。

1. 図 2(1) に示すように、CAS に対応した Web アプリケーションにユーザがアクセスすると、認証がまだの場合は、(2)CAS サーバへユーザをリダイレクトし、そこでユーザ ID とパスワードによる認証を行う。熊本大学の場合は認証に必要なデータは、全学 LDAP サーバから取得している。
2. 認証が通った場合、図 2(3) に示すように、CAS サーバから Web アプリケーションに、再リダイレクトされる。今度は、CAS サーバで取得した Ticket を持って、(4)Web アプリケーションにアクセスする。有効期限内にユーザ認証が既に成立している場合も、Ticket を持ってアクセスすることになる。
3. Ticket を持って Web アプリケーションにアクセスした場合、図 2(5) に示すように、Web アプリ

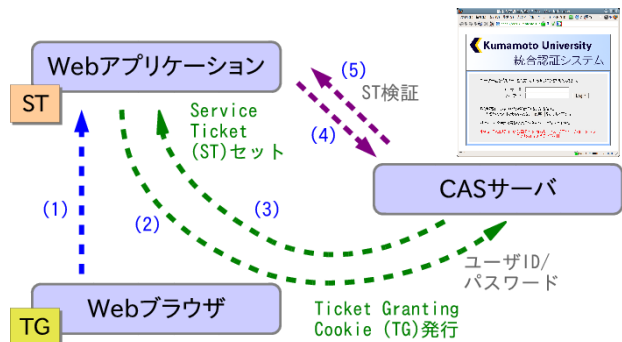


図 2: CAS によるシングルサインオンの概略

ケーションは CAS サーバにアクセスし、その正統性を検証する。

4. 検証ができた場合、CAS サーバは Web アプリケーションにユーザ ID を返すことで、Web アプリケーションはユーザ認証を終了し、同時にユーザ ID を取得する。

このように、比較的単純なシステムであるにもかかわらず、Web アプリケーション自体はユーザのパスワードに関する情報をなにも持つ必要がなく、高いセキュリティを持つ。Java、.Net、PHP、Perl、Apache、uPortal 等多くの言語やシステムに対応しており、本研究で学習者ポータル構築のベースとして用いた uPortal をはじめ、ポータル環境を支える SOSEKI 等、種々の Web アプリケーションも対応している。一例として、よく Web アプリケーション構築に利用される Apache Tomcat[6] においては、casclient クラスライブラリ [7] を利用することで、web.xml に filter を追加するだけで、ほとん

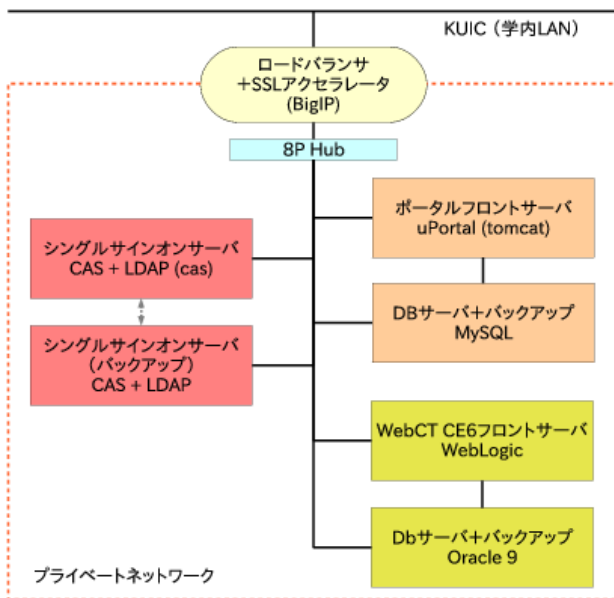


図 3: 熊本大学ポータル関係のネットワーク構成

どソースコードに変更を加えることなく CAS によるシングルサインオンを実現することができる。

世界的に多くの大学で、また国内では名古屋大学 [8] 等で、CAS および uPortal を利用した大学ポータルが活用されている。

### 3.2 uPortal

uPortal は、高等教育機関により開発されたオープンソース・フリーソフトウェアのポータルアプリケーションである。ユーザを複数のグループに所属させることができ、ユーザ又はグループ毎に、PUSH 又は PULL 型の情報提供 (fragment) を行ったり、情報のアクセス権限の設定が可能であり、ユーザ自身によるカスタマイズも可能となっている。情報提供方法もインラインフレーム、プロキシ、JSR168 規格 [9] に準拠した Portlet、Channel (uPortal 専用) 等が利用できる。

## 4 実装

以上のように、熊本大学ポータルは、CAS 及び uPortal をベースにしているが、どのように導入、修正、機能追加、開発等を行ったかを具体的に示す。

まず、図 3 にネットワーク構成の概略を示す。ロードバランサ兼 SSL アクセラレータ内のプライベートネットワークで構成され、セキュリティ的に強靱で、かつ、高速に動作できるようにしている。

## 4.1 シングルサインオン (CAS)

2007 年 8 月現在稼働中の CAS サーバのハードウェアは、1 台の Linux サーバ (2CPU, メモリ 2GB) 上に、CAS サーバと LDAP レプリカサーバを起動し、LDAP の認証データを用いて CAS が認証を行っているが、300 名以上の同時アクセスに対しても実用上の速度低下はなかった。また、全く同じハードウェア・ソフトウェアで構成されたもう 1 台のサーバを常にスタンバイ状態で待機、二重化し、トラブルに対応している。

以下、各々について示す。

### 4.1.1 CAS の導入と追加修正

CAS サーバは、1 台の Linux サーバ上に、LDAP、tomcat を起動し、CAS 自体は tomcat 上の J2EE アプリケーションとしてログインサービスを提供している。プライベートネットワーク外からのアクセスは SSL の必要があるが、SSL 変換や証明書の管理はロードバランサで行っている。

tomcat に関しては、ポート変更、アクセス URL の変更等、微小な変更で済んだ。一番大きな修正・変更は、ログイン画面のカスタマイズであり、熊本大学のブランド化に関係した修正が必要となった。

### 4.1.2 一般的クライアント対応

J2EE アプリケーションの CAS 対応に関して、例として、Servlet の Tomcat における設定ファイル web.xml の一部を図 4 に示す。ここに例では、casclient [7] の filter 機能を利用するだけで、CAS 認証を行うことが可能となり、CAS 認証に成功し filter を通過した場合は、Web アプリケーション内で、request.getRemoteUser() でユーザ ID を取得することができる。

<filter></filter>の<init-param>指定は、CAS 認証を行うために必要な初期値で、casclient.jar [7] を利用可能な場所に配置するだけで、CAS 認証及びリダイレクト処理を行うことが可能になる。ここで、CAS サーバの URL は SSL であることが必要であるが、ローカルなネットワーク間で実現する場合等、SSL は不必要で時間のロスになるが、本環境では casclient.jar のソースコードを数ヶ所変更する以外には SSL を外す方法が見当たらなかった。

<filter-mapping></filter-mapping>は、どの Servlet にアクセスした時に CAS 認証を行うかを指定している。この場合は全ての Servlet としている。

```

...
</filter>
<filter-name>CAS Filter</filter-name>
<filter-class>
  edu.yale.its.tp.cas.client.filter.CASFilter
</filter-class>
<init-param>
  <param-name>
    edu.yale.its.tp.cas.client.filter.loginUrl
  </param-name>
  <param-value>
    CAS サーバ URL(SSL)/login
  </param-value>
</init-param>
<init-param>
  <param-name>
    edu.yale.its.tp.cas.client.filter.validateUrl
  </param-name>
  <param-value>
    CAS サーバ URL(SSL)/serviceValidate
  </param-value>
</init-param>
<init-param>
  <param-name>
    edu.yale.its.tp.cas.client.filter.serverName
  </param-name>
  <param-value>
    本 ervlet のサーバ URL(SSL)/認証させる Web アプリ
  </param-value>
</init-param>
<init-param>
  <param-name>
    edu.yale.its.tp.cas.client.filter.wrapRequest
  </param-name>
  <param-value>true</param-value>
</init-param>
</filter>

<filter-mapping>
  <filter-name>CAS Filter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
...

```

図 4: CAS 化 Servlet 設定 web.xml の一部

#### 4.1.3 WebCT CE4 への対応

2007 年度現在、全学 e ラーニングシステムとして利用している WebCT CE4 は、残念ながら CAS には対応していない。また、商用システムでありソースコードも公開されていないため、以下のようにアダプタの役割をするサーバを追加して対応した。

概念的には、名古屋大学が PHP で実装した方法とほぼ同様であるが、[10]、セキュリティ強度が高く大規模な運用が可能な servlet で実装するとともに、ポータル上でそのまま動作可能な Portlet[9] でも実装を行った [11]。以下に、その大まかな流れを示す。

1. ユーザが本 Servlet に認証が通っていない状態でアクセスすると、CAS サーバへユーザをリダイレクトし、そこでユーザ ID とパスワードによる認証を行う。なお、認証が通った状態でアクセスした場合は、この処理なしにユーザ ID が取得

きる。

2. 認証が通った場合、CAS サーバから本 Servlet 又は Portlet に、再リダイレクトされる。今度は、CAS サーバで取得したユーザ ID と、利用する WebCT CE4 サーバの固有に設定可能な秘密値 (Shared Secret Value) 等を元に、AutomaticSignon 機能 [12] を利用したユーザの myWebCT へ認証なしにアクセス可能な URI を合成する。
3. 本 Servlet 又は Portlet が上記で合成した URI へリダイレクトを行うことで、WebCT CE4 サーバの myWebCT ページへ移動する。

#### 4.1.4 WebCT CE6 への対応

2007 年度後期から全学公開を目指し、e ラーニングによる遠隔教育で e ラーニングの専門家を養成している熊本大学大学院社会文化科学研究科教授システム学専攻 (修士課程)[13] で先行利用を開始している、新しいバージョンの WebCT である、WebCT CE6 も、残念ながら CAS には対応していないため、WebCT CE4 と同様に、Servlet 及び Portlet で CAS に対応させた。

CE4 との違いは、AutomaticSignon 機能 [14] がセキュリティがより高い反面、さらに複雑で、WebCT CE6 サーバとのネゴシエーションが必要になる [11]。

#### 4.1.5 その他のシステムへの対応

熊本大学学務情報システム SOSEKI に関しては、Servlet で動作しており、開発元の方で対応していただいた。また、CALL に関しては、大学教育機能開発総合研究センターの方で、.NET で対応いただいた。また、実習用や学外向けに利用している e ラーニングシステム Moodle [15] に関しては、最初から CAS に対応していた。

CAS への対応は、ユーザレベルで構築したシステムからも可能で、既に、Java[16] や、ActionScript[17] 等の演習教材に利用した例もある。また、今後利用をさらに進めるため、2007 年度設立された熊本大学 e ラーニング推進機構 [18] での全学的な利用方法等の情報共有や共同開発を進めたい。

## 4.2 ポータル (uPortal)

2007 年 8 月現在、uPortal サーバは、先の CAS サーバと同じプライベートネットワーク上に配置された、2 台の Linux サーバ (各 1CPU、メモリ 3G) で構成され、1 台は Tomcat 上に uPortal (バージョン 2.5.1) が起動し、

もう1台は uPortal 用データベース (MySQL) サーバおよびバックアップサーバとして利用している。プライベートネットワーク上にサーバ群を配置することで、CAS サーバへのアクセスを SSL をかけずに行えるため負荷の減少と速度向上を狙るとともに、データベースサーバをの隠蔽及び負荷分散が行える。ただし、CAS サーバを SSL を外してアクセスするために casclient クラスタライブラリ [7] の修正が若干必要であった。

uPortal は Java 仮想マシンのヒープサイズを 1GB 程度に拡大した Tomcat 上で運用することで、通常のアクセスであれば 300 名以上の同時アクセスに対して問題なく動作しているが、今回教授システム学用に関連した学習ポータル同時アクセスはまだ数名程度であり、その評価はできない。ただし、これは uPortal サーバに置く Portlet や Servlet の負荷と、データベースサーバの負荷に大きく依存すると考えられる。

なお、uPortal を運用する際には、大学の事情にあわせたカスタマイズが必要になる場合が多いと思われる。実際に熊本大学ポータルとしての運用開始にあたって、ソースコードを含むかなりのカスタマイズを行った。以下に各々概略を示す。

#### 4.2.1 uPortal の CAS 対応

ポータル (uPortal) から様々なシングルサインオン (CAS) に対応したシステムにアクセスすることから、uPortal と CAS の関係は特別に見えるが、実は uPortal も他のシステムと同様に CAS クライアントに過ぎない。uPortal を CAS に対応させるには、cas-client-java や cas-sec-prov の導入にはじまり、cas 関係 library の deploy、security.properties、filter 等の設定が必要となる。

さらに、Login / Logout の修正が必要になる。ログインページを CAS 化するために、CAS の Login ページのインラインフレーム表示、ログアウト時に SSO 終了となり、ログイン名の表示形式、一般ユーザの編集モード禁止等を含む uPortal のトップページを始め、かなりの変更・追加が必要であった。

#### 4.2.2 Tomcat の環境整備

デフォルトの Tomcat 設定では、OutOfMemory になることが多かったため、startup スクリプトに以下の環境変数設定を行ったところ、かなり動作が安定し、OutOfMemory も出なくなった。

```
export CATALIN_OPTS="-server -Xms1024m -Xmx2048m"
```



図 5: 一般的学生のポータルのトップページ例

#### 4.2.3 データベースサーバ

uPortal のバックエンドとしてのデータベースサーバとして、別のハードウェアを利用し、MySQL を起動している。セキュリティを考慮しプライベートネットワーク外からの直接アクセスが不可能な設定にしている。

#### 4.2.4 uPortal に関するその他修正等

運用にあたって、ユーザからの情報でデフォルトスキンのスクロールバーの色が薄いことが判明し、ソースツリーのレイアウト (AL\_TabColumn/integratedModes/等) の中の該当 CSS の修正で対応した。

Internet Explorer Ver.7 で動作しないという報告があり、ソースツリーのプロパティ(properties/browser.mappings) 及び source/org/jasig/portal/channels/CInlineFrame.java エントリを追加することで対応した。MySQL のバージョンは uPortal に依存する (properties/db/dbloader.xml) ので、運用する uPortal に適合したものである必要がある。

## 5 熊本大学ポータルのページ構築

以上、熊本大学ポータル自体の導入と種々のシステムに関するシングルサインオンへの対応に関して紹介したが、熊本大学ポータル上のページの構築についてはまだ紹介していない。

図 5 に、一般的学生のポータルのトップページを示すが、実際のポータル上のページは、Portlet、Servlet、他サーバ (インラインフレーム) 等で構築している。

uPortal 自体は、先に紹介したように、インラインフレーム、プロキシ、JSR168 規格 [9] に準拠した Portlet、Channel 等が、ユーザやグループ (例えば、学部・学科、学生・教職員等) 毎に設定できるばかりか、個人個人がカスタマイズすることも可能である。しかし、この個人によるカスタマイズ機能を提供すると、本来のペー

ジを消したり様々な弊害が予想され、現時点ではこの機能を提供してはいない。

uPortal の個人・グループ等による分岐機能を利用して、アクセスしたユーザが、学生なのか教職員なのか、どの学部等に所属なのか(現時点では、遠隔教育で受講の大学院社会文化科学研究科教授システム学のみ)に関して、グループ化し、対応したページを表示している。

ただし、uPortal の機能では、ユーザが学内からのアクセスか学外からなのかを判断することができない。また、グループ化がかえって複雑になる場合もある。それらの場合は、図 5 に示すように、ページを Servlet や Portlet で構築し、その中で条件分岐を行っている。

## 6 今後の取り組みと課題

最近、教育、学習、管理、学務等に関する様々なシステムが Web ベースで提供されており、その傾向は今後も拡大すると考えられる。これらを、便利に活用していくためには、認証が連携でき、ユーザの特定とその所属グループ等による分岐が可能な、シングルサインオンとポータルとの連携が最も効果的であると考えられる。また、この連携の上に乗れば、システムの開発も比較的容易になる。すなわち、新しいシステムを構築する際に、ユーザ認証やユーザの所属等による限定を行う必要がなくなる。

実際の開発は、図 3 のポータルサーバ等のあるプライベートネットワーク内のシステムである必要もなく、場合によっては学外のシステムでも対応可能である。開発は、Servlet や Portlet[9]、PHP 等様々な方法で行うことが可能である。

今後の方向性を考える先進事例として、熊本大学大学院社会文化科学研究科教授システム学専攻(修士課程)[13]で 2006 年度から構築・運用してきた「教授システム学専攻ポータル(以下、専攻ポータルと呼ぶ)」を紹介する。ここで得られた成果を、[18]等を通して、全学的に展開していきたい。

また、現状の熊本大学ポータルの課題と今後の計画に関しても紹介する。

### 6.1 教授システム学専攻ポータル

熊本大学大学院社会文化科学研究科教授システム学専攻(修士課程)は、2006 年 4 月、e ラーニングの専門家を e ラーニングで養成することを目的として開設された[13]。開設にあたり、フロリダ州立大学の先行

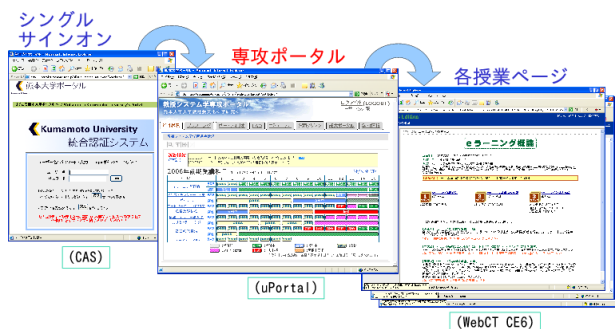


図 6: 専攻ポータルを利用した遠隔学習

事例調査[19]等も参考に、日本の e ラーニング専門家として修得すべき素養を分析・検討し、修了者に求めるコンピテンシーの設定と、それに基づいたカリキュラム及び授業設計がシステム的に行われた。このようにカリキュラムや授業内容がシステム的に設計されたため、専攻の授業科目体系が構造化され、前提科目やコンピテンシーと科目内容の関連が明確であることから、専攻の学習環境の入口である専攻ポータルも、これらに基づいて設計することが可能となった[20]。

専攻ポータルには、シングルサインオン、メインページである新着情報と学習進捗状況、プランニング、ポートフォリオ、FAQ、コミュニティ、資料/リンクの機能(タブ)があり、ここではその中から、メインページである新着情報と学習進捗状況のみ紹介する。他は、[21]を参照されたい。

図 6 に専攻ポータルを利用した遠隔学習の流れを示す。専攻の学生がポータルにログインすると、学習の進捗状況を中心とした各種学習サポートページがあり、そこから e ラーニングシステムへ飛び学習を行うことになる。

専攻の学生が実装したポータルに入ると、最初に表示されるメイン画面として、図 7 のような、新着情報と学習進捗状況の表示となる。この画面のタブ以下は 1 つの Portlet で実装した。ただし、上部の新着情報は、Portlet 内で他の Web サーバ上のページをインラインフレームで表示することで実現しており、ポータル管理者が作業を行わなくとも新着情報の更新が行える。

新着情報の下は学習進捗状況の表示であり、現在履修中の授業科目全てに関し、そのタスク(15 回)と課題(3~5 個程度)の履修状況と、WebCT 上の授業科目やタスク、課題等へのシングルサインオンによるリンクを実現している。履修状況は、受付前、受付中、×切 1 週間前、×切超過、添削中、再提出、合格、および前提条件のクリア状況に関して、アイコンで表示している。さらに、マウスオーバーで詳細情報、例えば、×



- [8] 梶田・内藤・小尻・平野・間瀬：「CASによるセキュアな全学認証基盤による名古屋大学ポータル  
の運用」第3回 WebCT ユーザカンファレンス予  
稿集, pp.115-120, (2005).
- [9] JSR 168: Portlet Specification,  
<http://www.jcp.org/en/jsr/detail?id=168>
- [10] 小村・福山・梶田・山里：「名古屋大学における  
WebCTのCAS化による認証統合」, 第3回 WebCT  
研究会予稿集, pp.53-57, (2005).
- [11] 中野他, 「WebCT(4/6)-CAS-uPortal SSO 連携の  
Servlet/Portlet による実装」, 第4回 WebCT ユー  
ザカンファレンス予稿集, pp.1-6 (2006)
- [12] WebCT, Inc., WebCT Campus Edition 4.0 Technical  
Reference Guide, Chapter 2, 2003.
- [13] 大森他, 「インターネット時代の教育を切り拓く  
大学院を目指して -インストラクショナル・デザ  
インによるeラーニング専門家養成-」, 大学教育  
研究フォーラム発表論文集, pp.48-49 (2006).
- [14] WebCT, Inc., WebCT Automatic Signon Protocol,  
WebCT Vista 4.0 and WebCT Campus Edition 6.0,  
Document version 4.0.0.1, December 2, 2005.
- [15] Moodle home,  
<http://moodle.org/>
- [16] 白木, 菅尾, 中野, 喜多, 「CAS 統合認証下におけ  
る学習支援ツールの開発」第3回 CMS 研究会予  
稿集, pp.37-44 (2006).
- [17] 中野, 喜多, 杉谷, 「オンライン ActionScript 演習  
ツールの開発」第5回 CMS 研究会予稿集, pp.29-  
32 (2007).
- [18] 熊本大学 eラーニング推進機構,  
<http://www.ield.kumamoto-u.ac.jp/>
- [19] 鈴木, 「教授システム学専攻大学院先進事例の Web  
調査」教育システム情報学会第31回全国大会講  
演論文集, pp.201-202 (2006)
- [20] 根本, 北村, 鈴木, 「eラーニング専門家養成のた  
めのeラーニング環境の設計: 熊本大学大学院教  
授システム学専攻の導入教育事例」教育システム  
情報学会研究報告 21(1), pp.33-40 (2006)
- [21] 中野, 喜多, 杉谷, 根本, 北村, 鈴木, 「CMS を補完  
する学習ポータルの実装」, 第4回 CMS 研究会予  
稿集, pp.55-60 (2006).