

学内向け施設予約システムの更新と開発

田村 亮太 中島 潤
北海道情報大学

The Update and Redevelopment of the Facility Reservation System
for HIU

Ryota TAMURA and Jun NAKAJIMA
Hokkaido Information University

2020年12月

北海道情報大学紀要 第32巻 第1号別刷

<報告>

学内向け施設予約システムの更新と開発

田村 亮太* 中島潤†

The Update and Redevelopment of the Facility Reservation System for HIU

Ryota TAMURA* Jun NAKAJIMA†

要旨

本学では2005年から教室や会議室等の施設予約管理を、OSSをベースとして独自開発し長年運用してきた。施設予約システムは、教職員が各種学内施設の予約管理を行う目的の他に、学生ポータルサイトを通じて実習室等の利用予定等を学生に提示するなど、学内で幅広く活用され、大学運営のための重要な情報システムの一つとして活用されてきた。しかし、運用開始から15年近く経過し、その間にOS等の稼働条件となる各種ソフトウェアが陳腐化してきたこと、周辺の学内情報システムの更新やセキュリティ対策等、早期に大規模なシステム更新が必要とされていた。本稿ではこの度、従来の施設予約システムを新システムへ更新・移行したのでその概要について報告する。

Abstract

Since 2005, Hokkaido Information University (HIU) has independently developed and operated a facility reservation management system for classrooms and conference rooms based on open-source software (OSS). The facility reservation system is widely used at HIU, for example, to provide students with information on the scheduled use of laboratories and other facilities through the student portal site, as well as for faculty and staff to manage reservations for various on-campus facilities, and as an essential important information system for university management. However, 15 years have passed since the system was introduced. The software that forms the operating conditions for the system has become obsolete, and it has become necessary to update the university's internal information systems in order to improve security measures. In this report, we outline the renewal and migration of the reservation system to its newer from.

キーワード

施設予約管理 (facilities management) OSS (open-source software)

* 北海道情報大学情報センター事務室, Information Center Office

† 北海道情報大学経営情報学部システム情報学科教授, Professor, Department of System and Informatics, HIU

1. はじめに

1-1 背景と開発概要

本学には、教室や会議室などの共用施設が約 70 存在し、またそれらの施設の利用者も多数存在する。そのため各施設の予約情報を共有・管理するシステムが必要不可欠であり、本学では 2005 年に中島らのボランティアが OSS をベースに独自開発した施設予約管理システム（以下、「旧システム」と称す。）を、XOOPS Cube で構築した教職員ポータルサイト上で稼働し、サービスを行ってきた（図 1）。

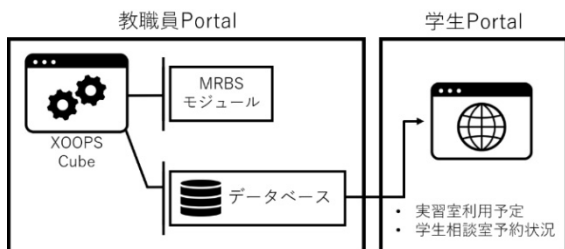


図 1 旧システムの構成

旧システムで管理される施設予約情報は、教職員による予約管理の他に、Web ポータル（HIU Portal）での「実習室の利用状況」や「学生相談室の予約状況」を提示するなど、幅広く活用されている（図 2）。

実習室 1	実習室 2	実習室 3	画像メディア実習室	LL実習室	105教室
実習室 1 の利用予定					
月曜日 (01/20)					
火曜日 (01/21)			キャリアデザイン III (夜間)		ソフトウェア工学(明)
水曜日 (01/22)	プログラミング基礎 (谷川)		情報科教育法 II (山北)		基礎英語 B (共通)
木曜日 (01/23)			情報システム学概論 II (南神)		
金曜日 (01/24)	ビジネスセミナー II (竹内)				プログラミング演習 I (徳井)
土曜日 (01/25)					
日曜日 (01/26)					

図 2 HIU-Portal に掲示される実習室利用状況

しかし、旧システムのベースとした MRBS は、当時の段階で開発やサポートが終了し、後の OS や PHP（稼働環境）のアップデート・パッチの適用が行えない状態であった。そのため、重大な脆弱性を内包していることが確認できていても対応が行えず、セキュリティ上の問題を抱えている状態がしばらく続いていた。また、予約情報の入力の方法によって画面デザインが崩れてしまう、といった不具合や、ユーザが予約可能な施設をハードコーディングで設定している、といった保守性の悪さも持っていた。

そこで施設予約システムを新たに構築した。本稿ではその過程で起きたデータ移行や講義情報の反映に関する課題解決について報告する。

1-2 MRBS (Meeting Room Booking System)

MRBS は OSS（オープンソースソフトウェア）として公開されていた Web アプリケーションで、施設や備品などの共有資源を予約管理することが可能である。MRBS は元々 Daniel Gardner によって書かれたものであるが、近年では John Beranek の支援を受け Campbell Morrison が、稼働環境・条件を変更しながら多くの機能を開発している。MRBS の持ついくつかの機能を以下に記す。

- Web ブラウザを通して様々な端末から予約可能
- 無制限のエリア・施設登録
- 様々な状況に対応した繰り返し予約設定
- 日・週・月ごとの表示切り替え
- 時間や期間での予約
- 予約の種別設定
- 複数の認証権限（読み取りのみ、一般ユーザ、管理ユーザ）
- 予約情報の一覧出力

- ・ 管理者への予約情報メール通知
- ・ 複数言語に対応

OSS として、当時これらの機能を持つ施設予約管理機能 MRBS のみであったことから、本学では MRBS をベースとして、本学独自に必要なとされる機能を実装し、利用することとした。

1-3 XOOPS と XOOPS Cube

XOOPS (eXtensible Object Oriented Portal System) は、Web サーバー上で PHP と MySQL を用いて動作する CMS (コンテンツ管理システム) で、本学では教職員ポータルシステムや (旧) 学生ポータルシステムの基盤として採用してきた。特徴として、各種機能モジュールを組み合わせることで、Web サイトに様々な機能を持たせることができ、非常に高い拡張性を持つ。XOOPS Cube は XOOPS から派生した、日本語等マルチバイト環境に対応した CMS で、日本では XOOPS Cube の方が広く普及している。多くの開発者によってモジュールが開発されており、XOOPS 日本公式サイトにはおよそ 400 件のモジュールが公開されている。本学でも過去に中島らが XOOPS 用モジュールを多数開発し、履修登録システムやシラバス登録・公開システム、文書管理システムを開発し運用してきた。

2. 施設予約システム更新における課題と解決

2-1 更新方法の検討

旧約システムの更新にあたり以下の 4 つの方法を検討した。

- ① XOOPS 用の古い MRBS モジュールを、不具合が生じないように最新の稼働環境に合わせて独自に書き直す

- ② 現在も開発が続いている MRBS を単体のシステム (サイト) として再構築する

- ③ 商用のパッケージ製品を含め、全く別の施設予約システムを探し導入する

- ④ 一から本学の要求に見合う施設予約システムを開発する

①のモジュールを修正する方法では書き直すコード量が膨大であるため、開発コストが高いと判断し見送ることにした。②の MRBS を単体で再構築する方法はシステム UI が大幅に変わらないためシステム更新後もユーザが使いやすいという利点がある。また、データの移行に関してもテーブル構造が似ているため他のシステムに比べて行いやすいという点があり有力な選択肢となった。③で挙げた別の施設予約システムを用いる方法では、導入コストなども考慮し、多くの他大学でも導入されている NetCommons3 という OSS ベースの CMS を用いての導入を検討した。そこで本学で必要となりそうな機能について、NetCommons3 と MRBS で比較を行った (表 1)。

表 1 MRBS と NetCommons3 の機能比較

項目	MRBS	NetCommons3
施設登録数	制限なし	制限なし
予約データ管理	DB	DB
施設をまとめるエリア機能	有り	有り
繰り返し予約	有り	有り
期間 (講時) での予約	有り	無し
月/週/日の表示切り替え	全て可能	週・日のみ可
最終更新 (20/01/27 現在)	19 年 12 月	19 年 10 月

機能面からみると両者に大きな差はなく、どちらも使用するのには支障がないように考えた。そのため最終判断として、データ移行の難易度とカスタマイズ性を検討した。

データ移行について考えると、NetCommons3 と旧システムはテーブル構造が大きく異なるため、②で述べた通り MRBS の方が適していると考えられた。またカスタマイズ性から考えても MRBS のほうがシンプルなコードで構築されており、機能追加も比較的自由にできると感じた。④の一から予約システムを構築する方法も検討したが、①同様開発コストが高く、開発期間を考えると難しいと判断し、選択肢から外すこととした。

従って本開発では②の MRBS を単体で再構築する方法を採用することにした。

2-2 新システムの構成案

システム更新にあたり課題となるのは、旧システムからのデータ移行や講義に使用する施設の予約反映方法だった。この課題を解決するために、次のようなシステム構成とした(図3)。

ユーザは Shibboleth 認証を通して本システム

にアクセスし、認証情報から与えられた権限に基づいて予約の作成・参照・変更・削除処理を行うことができるシステムとする。

また、旧システムからのデータ移行用プログラムと、講義に使用する施設の情報を e アシーナ(出欠管理システム)から取得し予約として反映するプログラム、をそれぞれ新たに開発することにした。なお、本開発は次の環境で行った(表2)。

表2 新システムの開発環境

OS	CentOS 7.6
Web サーバ	Apache 2.4.6
DB	mysql 14.14
開発言語	PHP 7.3.9
使用ソフト	git 1.8.3.1, Vagrant 2.2.6

2-3 Shibboleth 認証から取得する属性の利用

新システムを開発するにあたり、次の3点の課題を解決する必要があった。

- ・ 学生の本システムの利用を禁止する
- ・ ロール(部署や役職)ごとに予約可能なエリアを制限する
- ・ 予約にグループの概念を持たせる

これらの機能を実装するために Shibboleth

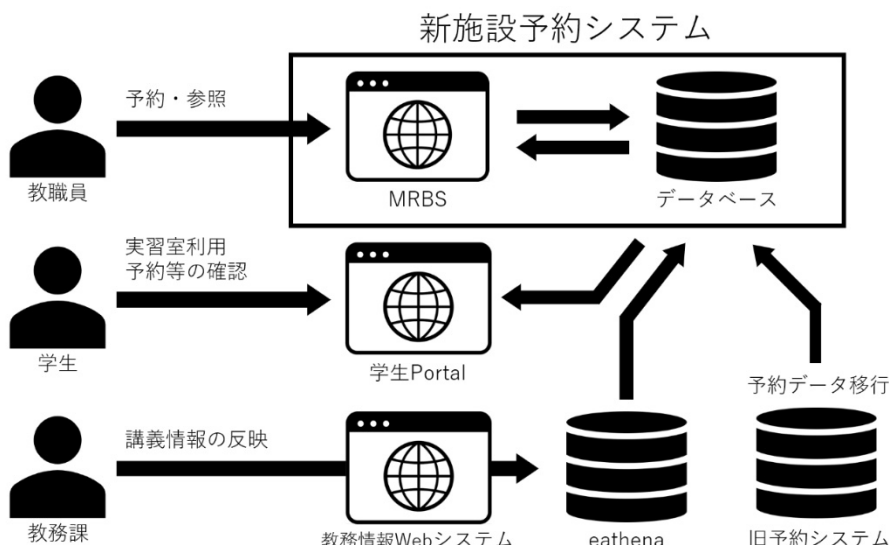


図3 新システムの構成

認証を利用する。本学の Shibboleth の運用では、認証後に次のような値が返却される（表 3）。

属性名	属性値	属性値の例
unscoped-affiliation	所属	Student
uid	ユーザ ID	s1612999
displayName	氏名	情報 太郎

表 3 Shibboleth から得られる属性(例)

Shibboleth により取得できる値は属性名をキーとしてサーバー変数として設定される。しかし、Shibboleth からロールや予約可能エリアの情報を取得することはできないため、サーバー変数に設定されたユーザ ID を用いることにより、既存の教務情報 Web システムで利用されているロール情報を取得し、そのロール情報をキーとして、新システムの予約エリア管理テーブル(図 4)から予約可能エリアを取得する構造とした(図 5)。

mrbs_area_auth	
role_name:	char(32)
area_id:	int(11)

role_name	area_id
admin	1
admin	2
admin	4
admin	6

図 4 予約可能エリア管理テーブル

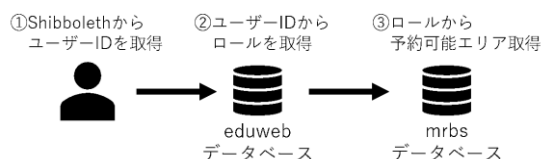


図 5 予約可能エリア取得フロー

ロールを持たないユーザや予約エリア管理テーブルにロールを登録していない場合は、デフォルトの予約可能エリアの値が設定され、この値を参照することで予約や予約変更の際に適切な権限を付与する構造とした。取得した値をシステム内で利用しやすくするために新たに Shibboleth ユーザクラス (図 6) を生成し、セッションに保持する。システムはセッションから Shibboleth ユーザクラスに設定された値を参照することで認証に利用した。

Shibboleth ユーザー
<ul style="list-style-type: none"> • ユーザーID:string • 氏名:string • 所属:string • ロール:array • 予約可能エリア:array
<ul style="list-style-type: none"> • ユーザーIDを返却する • 氏名を返却する • 所属を返却する • ロールを返却する • 予約可能エリアを返却する

図 6 Shibboleth ユーザクラス

「学生のシステム利用を禁止する」という課題に対してはセッションに保持されているユーザの所属情報を利用することにし、本学の Shibboleth の運用では所属情報として教員、職員、学生のいずれかを返却するため、所属が学生ならば閲覧権限エラーページを表示するようにした。

「ロール(部署や役職)ごとに予約可能なエリアを制限する」という課題に対して、旧システムではハードコーディングでの対応を行っていたが、ハードコーディングでは保守性が悪く、また今後の運用の中で施設やロールの追加を行いたい際に対応が難しくなるため、新システムでは MRBS に備わっている標準の権限機能と前述した予約可能エリアを組み合わせることにより、ハードコーディングにならない形で解決することとした。MRBS には次の 3 つの権限がある (表 4)。

表4 MRBSの3つの権限

Level	ユーザ定義	権限詳細
0	未認証ユーザ	予約の参照のみ可能。 変更・消去などは不可
1	承認済ユーザ	予約可能。自身の予約のみ変更・消去可能
2	管理者ユーザ	すべての予約・変更・消去が可能。施設の追加が可能

新システムでは予約画面にアクセスすると、その URL から予約しようとしているエリアを取得し、セッションに保持されているユーザの予約可能エリアに該当エリアが含まれているか確認し、含まれていれば Level 1、含まれていなければ Level 0 を返すような認証機能を開発した。

「予約にグループの概念を持たせる」という課題も、セッションのロールを参照することで解決を試みた。MRBS では本来、管理者を除き予約の変更は自身が行った予約に対してのみ可能である。つまり、グループという概念は存在せず、予約は個々人のものであったが、これは急な予約変更や取り消しを行いたい場合に於いても、予約者が不在の場合は同部署内の職員であっても予約の操作をすることができない、という煩わしさが改善したい点の一つとして挙げられていた。新システムでは予約変更権限確認フローでもセッションに保持したロールを参照するように手を加え、次の場合で予約者以外でも予約の操作を行うことができるようにした。

- ・ admin ロールを持っている
- ・ 予約ユーザが自分と同じロールを持ち、自身の所属が職員
- ・ 予約ユーザが自分自身

この変更により例えば、ある部署の職員 A による施設予約を同じ部署に所属する職員 B

が変更するといったことが可能となった。

2-4 旧システムからのデータ移行

新システムへの移行にあたり、旧システムからのデータ移行が最重要課題だった。データ移行が難点となるのは、予約時間単位を変更したからであり、旧システムでは予約を時・分単位で管理していたが、新システムでは、講義に使用する施設を講義単位での管理とした（図 7）。これは旧システム運用時に講義に使用する施設では講義時間に合わせた予約が非常に多かったためである。

時間:	101(90)	102(90)	103(90)
09:00	●	●	生体計測装置学(清水)
09:15	●	●	
09:30	●	●	
09:45	●	●	
10:00	●	●	
10:15	●	●	
10:30	●	日本語	●
10:45	●	VI(田中)	●

期間	101 (90)	102 (90)	103 (90)
1講時(09:00-10:30)			生体計測装置学(清水)
2講時(10:40-12:10)		日本語 VI(田中)	
昼休み(12:10-12:55)			

図7 時・分単位での管理と講義単位での管理

図 7 で表した期間による管理は、システム内部では 12:00 を開始とした 1 分刻みで処理されている。わかりやすいように表示を旧システムと統一すると以下のようなになる（図 8）。

期間	101 (90)	102 (90)	103 (90)
12:00			生体計測装置学(清水)
12:01		日本語VI(田中)	
12:02			

図8 内部的に見た講時単位での管理

そのため、例えば予約開始時間が9:30であるとき、開始が1講時の予約として当てはめる、すなわち「9:30」という値を「12:00」(1講時)に変換する処理が必要となる。時・分で管理されていた情報を講時に変換したため、開発時は重複や想定と異なった講時への予約が多く行われてしまったが、データ移行は一度のみしか行わないため、こうした誤変換は多少であれば許容することができる考えた。しかし、余りにも誤変換が多かったため、開発環境にて閾値の調整を繰り返すことで、限りなく誤変換が起こらないようにし、最終的に表5に基づいて変換を行い、新システムへ既存の予約データを移行した。

表5 時・分単位から講時単位への変換テーブル

講時	予約開始時間	開始時間 変換後	予約終了時間	終了時間 変換後
1	~10:29	12:00	~10:30	12:01
2	10:30~12:14	12:01	10:31~12:15	12:02
昼休	12:15~12:44	12:02	12:16~12:45	12:03
3	12:45~14:29	12:03	12:46~14:30	12:04
4	14:30~16:14	12:04	14:31~16:15	12:05
5	16:15~17:44	12:05	16:16~17:45	12:06
6	17:45~	12:06	17:46~	12:07

2-5 講義使用予定情報の反映

本学にはおよそ70の共有施設が存在するが、約半数の施設は講義に使用されている。そ

れらについても施設予約重複を起こさないために、また施設を最大限活用するためにシステムに予約情報として登録する必要がある。

本学では講義予定情報がeアシーナ(出欠管理システム)上のデータベース上に保存されており、このデータベースから必要な情報を取得し、新システムの予約情報として反映する必要がある。eアシーナのテーブルと新システムのテーブル構造は全く異なるため、それらの値を変換する処理が必要であるが、その変換処理は旧システムで利用されていたコードを再利用することにし、新システムでは反映プロセスの簡略化を目指した。

旧システムでは教務課がeアシーナから開講予定情報をCSVファイルとしてダウンロードし、教員Portalにアップロード、その後に予約情報反映プログラムを実行するという煩雑なプロセスを経る必要があった(図9)。本システムではこの「取得」、「変換」、「登録」の処理を一つの機能として纏め上げ実装した。ワンクリックのみでシステムはPDOを用いてeアシーナデータベースから開講予定情報を取得し、変換の処理を実行、そのまま施設予約として登録する。(図10)。反映時はeアシーナによる既存の予約情報を削除してから追加する処理を行うことで、予約の重複が起きないように工夫をした。また講義に使用する施設の予約情報は、随時更新が必要となるため、教務課職員が容易に実行できるよう、既存の教務情報Webシステムの管理メニューに配置した。

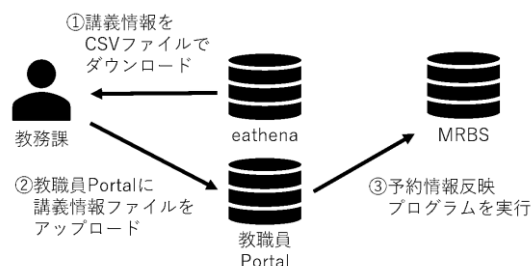


図9 旧システムにおける講義情報反映プロセス

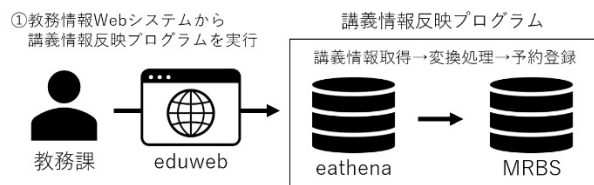


図 10 新システムでの講義情報反映プロセス

3. まとめ

セキュリティ問題の解決を第一とし、その他動作上の不具合を改善するため、施設予約システムの更新を行った。更新方法として複数の選択肢があったが、機能面、拡張性、更新にかかるコストから総合的に判断した結果、最新の MRBS を単体のサイトとして再構築する方法を選択した。

既存の他学内サイト同様に Shibboleth を用いたシングルサインオン認証を取り入れることにし、これにより、既存の教務情報 Web システムで使用していたロール(権限情報)を利用することが可能となり、ハードコーディングで行っていた予約権限周りの課題を改善することができた。また、ロールを参照することで本来 MRBS にはないグループの概念を持たせることができた。

多くのプロセスを経て行われていた教務課による講義に使用する施設予約情報の反映も簡単な操作で取得、変換、登録を行える機能を実装し、日常的な運用操作を簡素化することが出来た。

ユーザインタフェースの面から見ても、利便性や見やすさの向上を行うことができたと考えている。旧システムでは、表が枠からはみだしデザインが崩れていたり、空いている講時が見づらかったりなど利用者からも不満の声が上がっていたが、新システムでは講義に使用する施設を講時単位で予約できるように変更したため、空いているコマが見やすいことや、予約時に細かく時間を入力するなど

いった煩わしさを改善できたと考えている。また旧システムでは、予約時間の入力値条件によって表レイアウトが崩れて表示される不具合があったが、予約登録フォームを改善することで防止されるようにした。

参考文献

Geeknet MRBS <https://mrbs.sourceforge.io/>
(2019年12月19日アクセス)。

Shibboleth
<https://www.shibboleth.net/>
(2019年12月23日アクセス)。

sourceforge Meeting Room Booking System
<https://sourceforge.net/projects/mrbs/>
(2019年12月23日アクセス)。

XOOPS Cube 日本サイト
<https://xoopscube.jp/>
(2019年12月23日アクセス)。