

MegaOakIBARS II における新旧ハイブリッドシステムの開発

真野 誠・菊地 浩人
柳原 圭佑・倉持 敦

要旨

「MegaOakIBARS II」は、大規模から中小規模の病院における医療事務パッケージソフトウェアであり、現在約800の医療機関で稼働している「MegaOakIBARS」の後継機として開発されました。旧製品から新製品への移行がスムーズに行えることが必須であり、これを解決する方法として採用した新旧システムのハイブリッド方式について紹介します。

キーワード

●医療事務 ●.NET ●.NET Framework

1. はじめに

近年、医療分野の情報化においては、患者受付、カルテ管理、保険点数の算出、医療保険の審査支払機関に提出するレセプト（診療報酬明細書）の作成などを行う医療事務のシステム化は、ほぼ100%に近い普及率になっています。

NECでは、現在までに、医療事務パッケージソフトウェア「MegaOakIBARS」を全国の約800の医療機関に導入し、診療報酬改定・医療制度改革へ確実に対応してきた実績があります。

しかし、「MegaOakIBARS」の開発から11年が経過し、診療報酬改定やユーザ要求実現のための数多くの大きな改造により、COBOL、Visual Basic[®]、C、ACCESSで作成されているモジュールは膨大なものとなり、メンテナンス工数や導入工数の増大を招いています。

そこで、上記の課題を解決し、ユーザから見た魅力的な機能の強化、電子カルテシステムMegaOakHRとの連携強化をめざし、.NET Framework上で動作する次期医療事務システム「MegaOakIBARS II」（メガオークアイバースセカンド）を製品化しました。

この新システムの開発に当たっては、前製品の必要な機能をすべて受け継ぐこと、旧システムを導入している約800ユーザのスムーズな新システムへの移行を必須条件として開発を行いました。最終的には旧システムの機能をすべて.NETへ移行する予定ですが、上記の必須条件を満たすため、ファーストバージョンでは、旧システムのマスタ、データベース

(DB)、モジュールなどを活用したハイブリッドシステムとして新システムを構築しました。

本稿では、新システムの特長の1つであるハイブリッド方式について紹介します。

2. マスタ・DBの新旧ハイブリッド方式

MegaOakIBARS II では、MegaOakIBARSで使用しているデータベースのテーブル構成をそのまま活用し、新システム

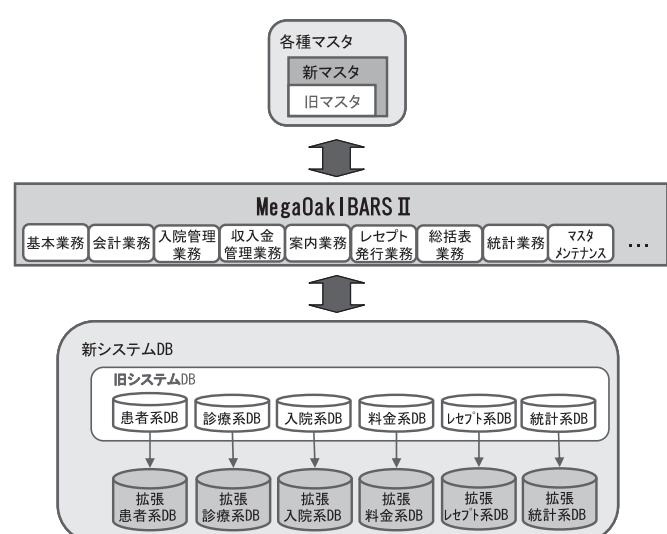


図1 MegaOakIBARS II のマスタ・DB構成イメージ

で新たに必要となる情報については、それぞれのテーブルにリンクする拡張テーブルを用意することで対応しました。また、マスタについては、旧システムのマスタを一部カラム拡張や追加を行うのみに留めています。

これにより、旧システムから新システムへの移行や旧システムのモジュールを活用することが容易になりました。

MegaOakIBARS IIのマスター・DB構成イメージを図1に示します。

3. 業務間連携の新旧ハイブリッド方式

3.1 MegaOakIBARS

(1) 業務管理と画面制御

MegaOakIBARSでは、各業務APがそれぞれ単体の画面を持ち、コントロールタスクと呼ばれるランチャーから起動する方式で作成されています。

業務管理モジュールは、各業務APからの起動・終了通知を管理し、各業務APの情報を共有メモリに保持します。

これらのアーキテクチャにより、コントロールタスク終了時に各業務APを終了する機能や、特定の業務を最前面に表示する機能を実現できます。

(2) 業務間通信

MegaOakIBARSでの業務間通信は、業務管理モジュールが中心となり、業務間の通信経路を確保しています。

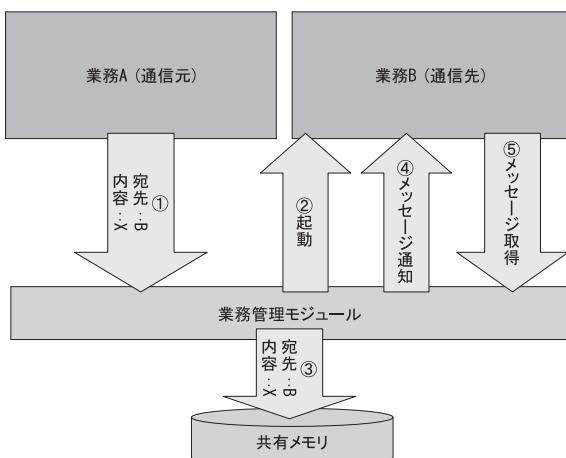


図2 MegaOakIBARSにおける業務間通信

また、通信先が存在しない場合は、起動させる機能も保持しています。

たとえば、業務Aから業務Bに通知を行いたい場合、下記の手順で実行されます（図2参照）。

- 1) 通信元業務Aから、業務管理モジュールへ通知（宛先：業務B、内容：X）
- 2) 業務管理モジュールは、通信先業務Bが起動していない場合、新規起動
- 3) 業務管理モジュールは、共有メモリに通知情報を格納（宛先：業務B、内容：X）
- 4) 業務管理モジュールは、業務Bに通知
- 5) 通信先業務Bが通知を受信し、業務管理モジュールから内容取得

3.2 MegaOakIBARS II

(1) 業務管理と画面制御

MegaOakIBARS IIでは、各業務画面をメイン画面内に表示する、MDI (Multiple Document Interface) 形式となりました。このため、現行業務の画面をメイン画面内に表示する必要があります。

また、現行業務はVisual Basic (R) 6で作成されている単体 EXEであり、MegaOakIBARS IIでは、.NET Frameworkで作成されています。

メイン画面は.NETで作成されているため、そのままではMDI形式として表示することができません。

そこで、現行業務の画面を次的方式を用いて、MDI化を行いました。

- 1) MDIの子画面となる.NET製のWindowsFormクラスを用意（以下、プロキシフォームと記述）
- 2) 1) のプロキシフォームは、1つの現行業務プロセスの起動・表示・終了を管理

プロキシフォームでは、下記の情報を管理・監視し、整序を行う必要があります。

- ・ 現行業務の業務画面 (WindowHandle)
- ・ 現行業務のプロセス状態
- ・ 現行業務の表示状態

これらの方式を用いて、MegaOakIBARS IIのメイン画面は、MegaOakIBARSの現行業務を、新規に作成された業務画面と同様に扱うことを可能としました。

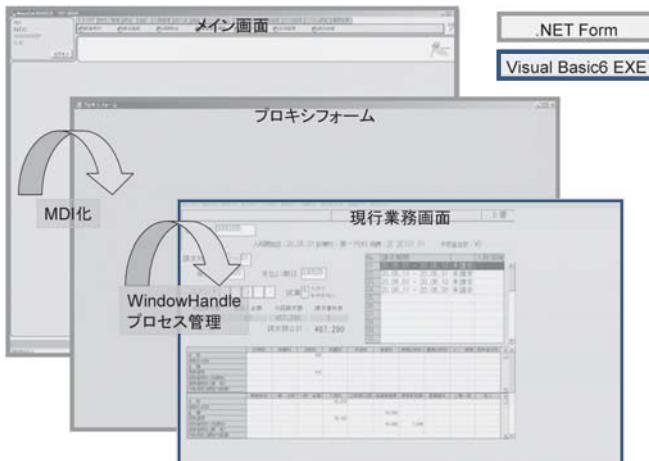


図3 MegaOakBARS IIにおけるMDI表示方式

のことにより、現行業務の修正を最小限に抑え、MDI表示が可能となります（図3参照）。

（2）業務間通信

新業務と現行業務が共存している場合、新業務から現行業務、現行業務から新業務への業務間通信を行う方法が必要となります。

そこで、新業務を管理しているメイン画面と、現行業務を管理している業務管理モジュールの両方を使用し、下記の手順で業務間通信を実装しました。

・新業務から現行業務への通信手順（図4参照）

- 1) 新業務からメイン画面を通じて、現行業務Aのプロキシフォームに通知（宛先：現行業務A、内容：X）
- 2) プロキシフォームは担当する現行業務プロセスを宛先として、業務管理モジュールへ通知（宛先：現行業務A、内容：X）
- 3) 業務管理モジュールは、共有メモリに通知情報を格納（宛先：現行業務A、内容：X）
- 4) 業務管理モジュールは現行業務Aへ通知
- 5) 現行業務Bが通知を受信し、業務管理モジュールから内容取得

・現行業務から新業務への通信手順（図5参照）

- 1) 現行業務から業務管理モジュールへ通知（宛先：新業務A、内容：X）
- 2) 業務管理モジュールは、共有メモリに通知情報を格納（宛先：新業務A、内容：X）

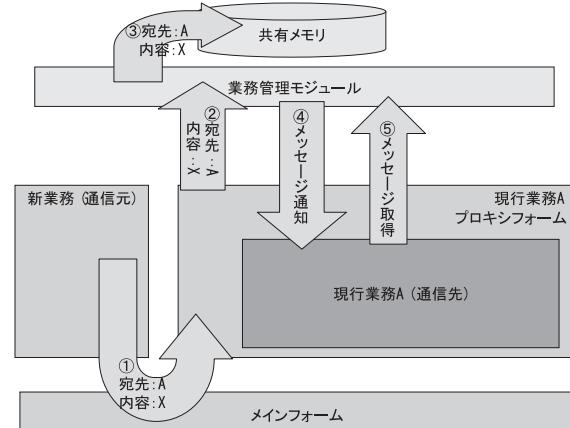


図4 新業務から現行業務への通信手順

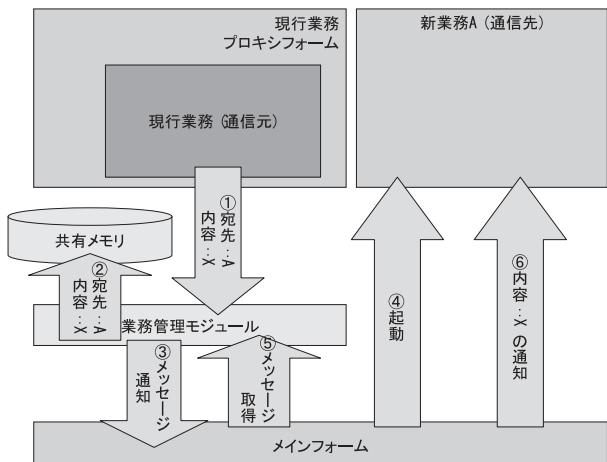


図5 現行業務から新業務への通信手順

- 3) 業務管理モジュールは、メイン画面へ通知
 - 4) メイン画面は、宛先となる新業務が起動していない場合、新規起動
 - 5) メイン画面は、新業務Aの通知内容を、共有メモリから取得
 - 6) メイン画面は、新業務へ通知（内容：X）
- これらの方を用いることで、メイン画面と業務管理モジュールのみの変更で新業務・現行業務の相互通信を可能としました。

4. むすび

以上に述べたように、MegaOakIBARSⅡでは新旧システムのハイブリッド化を図ることにより、旧システムの機能を引き継いだ新システムを開発することができました。

この方式により、現在旧システムを導入している全国約800ユーザのスムーズな新システムへの移行を実現できると考えます。

*Visual Basicは、米国Microsoft Corporationの米国及びその他の国における商標または登録商標です。

執筆者プロフィール

真野 誠
公共・医療ソリューション事業本部
医療システム事業部
ニアマネージャー

菊地 浩人
公共・医療ソリューション事業本部
医療システム事業部
医事パッケージグループ
グループマネージャー

柳原 圭佑
公共・医療ソリューション事業本部
医療システム事業部
医事パッケージグループ
主任

倉持 敦
公共・医療ソリューション事業本部
医療システム事業部
医事パッケージグループ
主任