

電子カルテシステム 「MegaOakHR」

並川 寛和・宮川 力
佐藤 雄亮・高島 浩二

要旨

電子カルテシステムソリューションの中心となるMegaOakHRは、オーダ指示・カルテ記述・情報参照など、医療情報の記録・共有を行います。本稿では、MegaOakHRの開発に至った背景をふまえ、その新機能と内部構造の変更を紹介します。

キーワード

●電子カルテ ●.NET ●.NET Framework

1. はじめに

NECは、電子カルテシステムソリューションを担うパッケージとして、2000年にMegaOak-NEMRをリリースしました。その後、様々な研究、議論、経験により、電子カルテに求められる機能は日々拡張され、その期待も大きくなっています。

これらの背景に加え、技術的な進歩もふまえ、利用者から見た価値の強化およびシステム構造の見直しを行ったMegaOakHRを2006年10月にリリースしました。

本稿では、その後の機能強化も含めて、MegaOakHRについて紹介します。

2. MegaOak-NEMR

1999年4月22日に、当時の厚生省から、「診療録等の電子媒体による保存について」という行政文書が出され、電子カルテシステムとして3つの原則（真正性、見読性、保存性）が提示されました。これにより電子カルテの導入は本格的に始まり、診察記事の入力、紹介状、サマリなどの文書管理機能、表形式でカルテ情報を鳥瞰するFlowSheetなどの機能をリリースしてきました。

電子カルテの特徴としては、1つのデータに対する表現形式が様々あることが挙げられます。1つの情報が、2号紙イメージになったり、表形式（FlowSheet）になったり、経過表（歎計表）になったりと様々な切り口で検索され、加工して表示されます。通常の業務システムに加え、Data Warehouse的な

要素が含まれるためDBMSとしてObject Databaseなども検討しましたが、既存のオーダリングシステムで採用していたOracleを、パッケージとしての継続性を重視して採用しました。

一方、ハード的には、設計当初のサーバはCPUが高々400MHz程度と非力であり、1,000台を超えるクライアントからの多次元の検索処理と表示イメージ構築をサーバ側で行うことは困難と考え、クライアントで処理を行う、いわゆるファット・クライアントの構成としました。さらに、高速化と可用性、安定性の向上をめざし、クライアントのディスクにキャッシュデータを持ち、メッセージキューイングにより非同期処理を行うシステムとしました。

3. 電子カルテシステムへの期待

電子カルテに求められる機能、期待される役割は、日々、高まってきています。「三原則」が通達される以前から、日本医療情報学会などに代表される学会や研究会では、電子カルテの姿に関する研究、議論が盛んになされていましたが、さらに経験が積まれることと社会的背景の変遷により、その形も徐々に変化し、より具体的になってきました。これまでできていたことは当たり前であり、次のステップへ飛躍する段階となったと言えるでしょう。

新機能に対する主な要望としては、以下の3点があります。

- 1) 病院業務の質を上げ、より効率よく行うための機能
- 2) インシデントを防止する等、安心・安全につながる機能
- 3) 個人情報保護法等、セキュリティ要件に関する機能

病院情報システム 電子カルテシステム「MegaOakHR」

4. MegaOakHRの新機能

ここでは、前述した要望を受け、開発を行ったMegaOakHRの代表的な変更点を記述します。

4.1 ユーザビリティの向上

人間工学的に、人の視線は左から右に、上から下に流れるのが自然な動きであると言われています。MegaOakHRでは、最初に見るべき情報、操作すべきボタンを左上に配置し、最後に操作するボタンを右下に配置しました。視線の動きは大きくなりますが、全体を鳥瞰し、見落としを防ぐことが重要な医療の世界では、この視線の動きが有効となります。

ボタンの押し間違えを無くすため、ボタンの色、配置も変更しています。「閉じる」、「やめる」系のボタンを左下に緑色で、「登録」、「確定」系のボタンを右下に黄色で、その他のボタンを中央に青色で配置、配色しました（図1）。

また、患者取り違え防止を考慮し、カルテ画面に性別と年齢に応じたイラストを表示しています（図2）。

4.2 チーム医療の推進

病院内では、様々な職種のスタッフが一人の患者に携わります。その際に情報を正確に伝達することは、インシデントを防止する重要な要素になります。

図1 ボタン配置、配色の変更

現在、リリースされているのは「医師ToDo」（図3）、「看護師ToDo」、「病棟ToDo」、「患者ToDo」と呼んでいるもので、職種をまたがった指示、伝達をもれなく伝える仕組みを開発しました。指示を受けた側の備忘録としても使え、指示に疑問があるときには疑義問合せ ToDoリスト上で行うことが可能です（ToDo：やらなければいけないこと、忘れてはいけないことなどをメモしたもの）。

4.3 その他の機能強化

クリニック・パスの機能強化としては、疾患や患者の状態に合わせて診療プロセスを臨機応変に組み合わせることがで



図2 メイン画面のイラスト表示



図3 医師ToDoの例



図4 ユニットパスの画面例

きるユニットパスを開発し、きめ細かい診療支援をめざしています（図4）。

サマリや紹介状を記述する文書管理機能としては、過去データの引用機能を充実させ、文書の作成をより省力化する仕組みを提供しています。

さらに、個人情報保護法に代表されるセキュリティ機能強化に対する要望は大きく、不正アクセスを抑制、監視するためのアクセスログの機能や、担当患者以外の情報を参照できなくなるアクセス制御機能を開発しています。

5. MegaOakHRの内部構造

5.1 .NET Frameworkの採用

MegaOak-NEMRの開発以降、ハードウェア環境も大きく変化しました。MegaOakHRでは、ハードウェアのリソースを有効に活用するため、モジュールの再配置を行いました。しかし、開発期間、影響度の観点から、すべてのソースコードを記述し直すわけにはいかないため、旧来のモジュールを残し、互換性を保ちながら新しいモジュールを同居させる必要がありました。また、先ほど、「Data Warehouse的な要素が含まれる」と記述しましたが、これを実現するためには、物理的なDatabaseスキーマをベースに、柔軟にViewを定義できること

が必要となります。サーバ上にViewを定義することも技術的には可能ですが、それでは逆にサーバ負荷が大きくなりすぎたため、クライアントのメモリ上でViewを構成する必要がありました。

この時期にはMicrosoft社の.NET Frameworkが実用的な段階となっており、.NETの技術であるDataSetが要求にもっとも適合する技術と考え、Visual Basic.NET 2005を開発環境として採用しました。従来のモジュールはVisual Basic6.0、Visual C++6.0を使用して開発されていますが、ラッパーを経由してシームレスにやり取りしているため、ユーザから見ればどこが6.0で、どこが.NETかは分からなくなっています。

5.2 保守性の高いパッケージをめざして

MegaOak-NEMRは1997年に開発されたPC-Ordering97とその後継のPC-Ordering2000、PC-Ordering/ADに電子カルテ機能を追加したものであり、10年以上の歳月を経たソースコードが残っています。このソースコードはその後の機能強化、カスタマイズにより複雑化し、保守性の悪いものになっていました。この状態では開発効率が悪く、お客様が望む機能を実装するのに必要以上の工数がかかってしまう状況となりました。

このため、MegaOakHRでは、一部のソースコードを.NETで記述し直しています。ただ単に記述言語をコンバートするのではなく、プログラムの内部構造を見直して冗長性を排除するリファクタリングを行うことにより、将来の仕様変更にも柔軟に対応できるパッケージをめざしました。実際に、.NETでリファクタリングしたソースコードは従来のソースコードに比べ、規模が1/5～1/3に抑えられ、可視性も上がっています。

シンプルな構造になっているかどうかを判断する1つの基準として、複雑度を定量評価するメトリクスという概念を導入しています。保守性の悪いソースコードは、行数が多くったり、分岐が多くたりすることが知られています。我々は、メトリクスを測定するためのツールを作成しており、1メソッド当たりの行数、分岐数などが測定できるようにしています。このツールを活用することにより、複雑で保守性の悪いソースコードを容易に見つけることができ、シンプルさを維持するリファクタリングの実践に役立てています（図5）。

病院情報システム 電子カルテシステム「MegaOakHR」

モジュール名	ファイル数	実行行			分岐数			空白行			コメント行	
		合計	1メソッド最大	1メソッド平均	合計	1メソッド最大	1メソッド平均	合計	1メソッド平均	合計	1メソッド平均	合計
モジュールA	4,118	69,851	544	170	3,363	15	0.8	6,924	1.7	40,736	9.9	
モジュールB	634	15,310	378	241	616	8	1.0	1,864	2.9	5,773	9.1	
モジュールC	122	1,196	50	9.8	44	4	0.4	146	1.2	162	1.3	
モジュールD	1,602	24,908	353	15.5	1,370	15	0.9	2,432	1.5	4,805	3.0	
モジュールE	151	1,144	36	7.6	27	3	0.2	248	1.6	41	0.3	
モジュールF	2,523	22,258	348	8.8	1,030	15	0.4	3,802	1.5	2,943	1.2	
モジュールG	9,891	215,747	1,420	21.8	10,609	36	1.1	26,818	2.7	48,408	4.9	
モジュールH	13	169	53	13.0	9	2	0.7	83	6.4	63	4.8	
モジュールI	394	3,739	57	9.5	207	4	0.5	1,139	2.9	358	0.9	
モジュールJ	607	6,953	232	11.5	268	9	0.4	1,192	2.0	768	1.3	
モジュールK	952	22,545	304	23.7	1,270	20	1.3	6,699	7.0	8,725	9.2	
モジュールL	558	6,655	199	11.9	317	33	0.6	1,622	2.9	1,311	2.3	
モジュールM	176	3,219	461	18.3	97	11	0.6	581	3.3	1,286	7.3	
モジュールN	5	58	28	11.6	4	2	0.8	21	4.2	22	4.4	
モジュールO	871	29,935	489	34.4	1,376	18	1.6	5,582	6.4	11,632	13.4	
モジュールP	821	11,106	210	13.5	557	15	0.7	2,532	3.1	1,825	2.2	
モジュールQ	117	2,415	116	20.6	139	7	1.2	632	5.4	302	2.6	
モジュールR	693	7,167	126	10.3	229	6	0.3	907	1.3	958	1.4	
モジュールS	629	6,815	163	10.8	289	9	0.5	2,253	3.6	1,117	1.8	
モジュールT	23,735	400,713	469	16.9	20,316	26	0.9	69,407	2.9	147,512	6.2	
モジュールU	589	5,549	59	9.4	204	5	0.3	1,617	2.7	725	1.2	
モジュールV	53	575	52	10.8	36	6	0.7	49	0.9	38	0.7	

図5 メトリクスツールの出力例

6. おわりに

本稿では、MegaOakHRの開発に至った背景をふまえ、その新機能と内部構造の変更を紹介しました。

電子カルテシステムに寄せられる要求、期待は日々高まっています。今後もさらなる機能強化、レスポンスの改善と、それを容易に行うためのリファクタリングを継続し、病院業務の効率化と安心・安全の実現に取り組んでいく所存です。

* Microsoft、Visual Studio、Visual C++、Visual Basicおよび.NET Frameworkは、米国Microsoft Corporation の米国およびその他の国における登録商標または商標です。

執筆者プロフィール

並川 寛和

公共・医療ソリューション事業本部
医療システム事業部
電子カルテ第二開発グループ
プロジェクトマネージャー

宮川 力

公共・医療ソリューション事業本部
医療システム事業部
電子カルテ第一開発グループ
主任

佐藤 雄亮

公共・医療ソリューション事業本部
医療システム事業部
電子カルテ第二開発グループ
主任

高島 浩二

公共・医療ソリューション事業本部
医療システム事業部
電子カルテ第二開発グループ
主任

参考文献

- 1) 並川ほか；「電子カルテシステムにおける基本情報単位“Medical Event”的モデルリング」、全国大会講演論文集 第57回平成10年後期(4)、251-252、19981005 (社団法人情報処理学会)
- 2) 並川ほか；「電子カルテシステムの基本情報単位“Medical Event”を格納する非構造化データベースの実現」、全国大会講演論文集 第59回平成11年後期(4)、“4-181” - “4-182”、19990928 (社団法人情報処理学会)
- 3) 高橋康ほか；「カルテ庫サーバとした電子カルテシステム」、全国大会講演論文集 第59回平成11年後期(4)、“4-179” - “4-180”、19990928(社団法人情報処理学会)
- 4) 高橋康；電子カルテシステム「MegaOak-NEMR」、NEC 技報、Vol. 53、No.9、pp.3~6、2000-9.