

HyperDemo

HyperDemo2

病院情報システム・支援システム・地域連携

2017年11月22日(水) 14:45 ~ 16:00 K会場（Hyper Demo）（3F イベントホールB・C・D・E）

[2-K-1-HD2-3] 院内データリアルタイム見える化サーバの構築による病院業務改善の試み

堀内 亮佑¹, 寺島 健史¹, 阿部 勝也², 山岸 大輔³（1.新潟大学地域医療教育センター 魚沼基幹病院, 2.株式会社BSNアイネット, 3.株式会社富士通新潟システムズ）

【目的】入退院・空床情報等のデータを電子カルテ端末からほぼリアルタイムで閲覧可能とするシステムを構築し、システム部門・一般職員の業務改善の効果を検証する。

【方法】 HISネットワーク内に Pythonが動作する専用サーバを設置し、サーバにインストールした富士通 Symfoware clientから JDBC接続した富士通データウェアハウス（DWH）データを院内770台の HIS端末及び500台のスマートフォンの Webブラウザから閲覧可能とした。表示データは空床情報、病床利用率、外来患者数、入退院患者数、再入院率、紹介逆紹介率、外来収入・単価、院内死亡率、褥瘡発生率、転倒率、B型肝炎ウイルス（HBV）再活性化対策アラート対象患者一覧等で、Webブラウザの非同期通信機能により数分毎に最新データに自動更新される。

【結果】見える化サーバの構築により、これまで日報等にて日々作成・配信していた各種データがリアルタイムで病院職員に閲覧可能となり、システム部門業務が大きく軽減された。加えて、1) 空床情報がリアルタイム表示されることでベッドコントロール業務がより効率的に確実に実行できるようになり平均病床稼働率の上昇に貢献した 2) 診療科別紹介率・逆紹介率のリアルタイム表示により逆紹介率上昇に貢献した 3) HBV再活性化対策アラートを従来の2週間毎から1日毎の自動出力が可能となり高リスク患者へのアラート迅速化に貢献した、等の効果が確認できた。

【考察】院内データ見える化サーバの構築は、システム部門の日常業務軽減に加えて、誰もが電子カルテ端末からいつでも最新のデータにアクセスできるようになったことで、院内データがより身近に感じられるようになり、経営陣のみならず一般職員の業務改善への意識向上をもたらした。今回のデータ以外にも様々な指標が病院には存在する。経営管理部門や診療部門と連携してより効果的な「見える化」を推進していきたい。

院内データリアルタイム見える化サーバの構築による病院業務改善の試み

堀内 亮佑^{*1}、寺島 健史^{*1}、
阿部 勝也^{*2}、山岸 大輔^{*3}

*1 新潟大学地域医療教育センター・魚沼基幹病院 診療情報管理室、*2 株式会社 BSN アイネット、
*3 株式会社富士通新潟システムズ

Data visualization with real-time update for the improvement of hospital business operation

Ryosuke Horiuchi^{*1}, Kenshi Terajima^{*1}, Katsuya Abe^{*2}

*1 Health Information Management Office for Uonuma Kikan Hospital, *2 BSN INET Co., Ltd.,
*3 FUJITSU NIIGATA SYSTEMS LIMITED

[Background and purpose] When the number of unoccupied hospital beds is decreasing, the bed management section shows the message that tells the distress situation of the remained beds to urge and prompt a discharge towards hospital staffs. However, there are no efficient ways to monitor bed occupancy rate, which leads to the increasing demand for a hospital real-time data visualizing system free of charge.

[Methods] After having concluded the non-disclosure agreement (NDA) with the electronic medical record (EMR) system vendor (Fujitsu), we started to construct the system that ODBC connects to the healthcare data warehouse (Healthcare DWH), thereby showing unoccupied beds data as well as other important hospital indices such as average length of the hospital stay in real-time on web browsers of the EMR clients. We introduced the Windows server into the EMR network. The server has Symfoware ODBC driver and Python environment. We created Flask app that renders the HTML file in which ChartJS graphs are plotted and deployed the app with IIS hosting service.

[Results] Having constructed the hospital data visualizing system makes it possible for hospital staffs to see various data in real time, which have been created and distributed by a daily report, and reduced the business volume of system administrators. Showing the data of unoccupied beds in real time makes the bed controlling business more efficient and more accurate, which dedicates to the increasing bed occupancy rate.

[Conclusion] Hospital staffs, not only the senior managements, but also the general staffs, have become more aspired to improve their own business operation, being more familiar with the hospital data through the hospital data visualizing system which makes them able to access the latest data whenever they want. A hospital has various indices. We aim to promote the effective visualization in cooperation with the management departments and medical departments.

Keywords: Data visualization, Symfoware, ODBC, Python

1. 諸論

空床数が減少してくると、看護部管理室より退院を促進するメッセージを電子カルテトップページに表示することで医師をはじめとする病院スタッフに病床数の窮迫状況を周知し、退院促進をすることにしてはいたが、根拠となるデータの適切な表示方法が無く、院内データ見える化システムの需要が高まっていた。

2. 開発目的

電子カルテベンダ(富士通)と秘密保持契約を締結し、診療データウェアハウス(以下、「診療 DWH」という)に ODBC 接続することが可能となったことを機に電子カルテクライアント端末の Web ブラウザに空床情報をリアルタイム表示させる取り組みを始めた。理論上、ODBC 接続が可能となるデータは全て Web ブラウザに表示させることができることから、空床情報以外の重要な指標も合わせて表示させた。

3. システム概要

独自に用意したサーバから Symfoware ODBC 接続し、診療 DWH のデータをリアルタイム表示する Web アプリケーションを作成した。

3.1 サーバの導入と Symfoware ODBC ドライバのインストール

Web サーバとして使用するサーバ1台(Windows Server 2012 R2)を電子カルテネットワーク内に導入し、同サーバに Symfoware ODBC ドライバ(32 bit 版)を富士通に依頼してインストールしてもらった。

3.2 Python その他の実行環境を整備

サーバを電子カルテネットワークに接続する前に予め Python3(64 bit 版)の実行環境と Python, JavaScript の各種ライブラリ及びデータベースとして使用する PostgreSQL 等をインストールしておいた。

3.3 Excel による外部データ接続と自動更新

当初は Symfoware ODBC ドライバを Python の ODBC API で操作するつもりだったが、Symfoware ODBC ドライバが 32 bit であるのに対し、インストールした Python3 は 64 bit 版であったため、アーキテクチャ不整合により断念せざるを得なかった。代替案として、Microsoft Excel にて ODBC 接続による外部データ接続を行い、その Excel ファイルを Python で読み込むこととした。外部データ接続のデータ更新は Windows タスクスケジューラに乗せた Python のプログラムで行うこととした。タ

タスクスケジューラは1分ごとに Python のプログラムを走らせるように設定し、Excel ファイルが1分ごとにデータ更新されるようにした。これによりデータのリアルタイム表示が可能になる。

3.4 Flask による Web アプリケーションの作成

Web アプリケーション作成用の Python のライブラリである Flask を使用し、読み込んだ Excel のデータを HTML ファイル上にプロットすると同時に、サーバ上に Web アプリケーションをデプロイした。

3.5 ChartJS によるグラフ表示

読み込んだ Excel のデータを HTML ファイル上にプロットする際、HTML5 の Canvas 機能を利用して動的なグラフを表示させるための JavaScript のライブラリ ChartJS を利用し Web ページ上に任意のグラフを表示させるようにした。なお当院の電子カルテ端末の Web ブラウザは Internet Explorer 9 であるが、ChartJS のグラフは問題なく表示された。

3.6 IIS によるホスティング

Flask によるデプロイは開発用のものであり、安定性の面からプロダクトベースの Web ページレンダリングには適していない。そこで Windows Server 2012 R2 標準の Web サーバ IIS を利用し Web ページのホスティングを行なった。これにより院内の電子カルテ端末、内線スマートフォン、タブレット等の Web ブラウザからページを持続的・安定的に閲覧することが可能となった。

3.7 フロントエンドでの自動更新

タスクスケジューラによる1分ごとの Excel ファイルの更新は飽くまでサーバサイドの更新であるため、クライアント端末で Web ページを表示させた後は、ページのリロードをしないとサーバサイドの更新内容が反映しないことになる。そこで JavaScript の非同期通信処理によってページをリロードすることなくサーバサイドの更新内容を Web ページ上に反映されるようにした。60 秒に1回非同期通信処理が走るようにした。これによりページを開いたまま別の作業をしていたとしても常に最新のデータが反映されることになる。

3.8 表示させるデータおよびページレイアウト

当初の第一目標である、空床情報については、空床数に加えて予定入院数(転入含む)、予定退院数(転出含む)、稼働率、「重症度、医療・看護必要度」患者該当割合、DPC 入院期間 II 超え割合、平均在院日数、病床回転数を表示(図1、図2)させた。患者情報保護の観点から図としての提示はできないが、診断群分類番号およびDPC入院期間II最終日を付加した入院患者リストのデータテーブルも表示させた。トップページにはタブを用意し、空床情報タブをデフォルトで表示させる。タブには主要臨床的インディケータを優先的に割り当てる。またサイドバーも作成し、プライオリティが比較的低いが特定の部署にとっては重要な情報を表示する専用ページに飛ぶリンクを貼り付けておく。またユーザからの問い合わせを受け付ける問い合わせフォームも作成し、サイドバーのリンクから飛べるようにしておく。今後は診療科ごとの24時間以内再入院率や疾病ごとの死亡率のグラフ化、病棟部門以外の分野への拡大を検討したい。また別途取り組んでいるHBV再活性化対策については現状2週間に1回のペースで実施しているアラート対象患者一覧出力を、同サーバを利用して日次処理化することで、よりきめ細かい注意喚起が行えるようにしていきたい。

3.9 日次再起動と日次バックアップ

タスクスケジューラにより、1日1回サーバ再起動を行うようにした。再起動開始時間は現場の意見を参考に今後決定する。また1日1回午前1時にサーバ全体のバックアップをサーバに接続した外部ハードディスク内に取りるようにした。

4. システム評価

Excel ファイルを介したデータ連携や使用する Web ブラウザによる制約から数々の改善点が認められる。

4.1 重たいページ

表示する情報が増えるにつれてページの初回ロード時間が増大していった。一度開いてその後は開いたままにしておけば問題にならないが、ユーザフレンドリーではない。

4.2 1分ごとの Excel ファイルの更新

タスクスケジューラで Excel ファイルを1分ごとに更新するようになったが、このためにサーバでの Excel ファイルの手動編集作業が難しくなった。タスクスケジューラが動かす Python のプログラムには、win32com という Windows のモジュールを使って更新対象の Excel ファイルを開き、すべて更新ボタンを押した後ファイルを閉じる、という一連の動作を GUI のウィンドウを開くことなく実行する命令文を記載したが、ファイルを閉じるという動作のところで開いている別の Excel ファイルも道連れに閉ざされてしまう。

4.3 最大同時アクセス数

残念ながらテストを行えていないが、Symfoware データベースと Web ページの間に立つ Excel ファイルが文字通りボトルネックになりはしないか危惧している。

4.4 双方向性

グラフ表示部分は ChartJS の標準機能により、ユーザが不要な項目をクリックして除外し、見たい項目だけ残すと、それに応じてグラフが変化し、残した項目だけのグラフが表示されるなど、対話的な操作は可能であるが、データテーブルについては、ユーザがソート順を変更したり、表示項目を絞り込んだりすることはできない。また絞り込み結果に基づいてグラフとデータテーブルの両者が瞬時に変化するような仕組みも実装されていない。その辺りの動作を手軽に実装できるモダンな JavaScript ライブラリが最近出始めてきているが、Internet Explorer 9 に対応しているものは少ないため、対応に窮している。

4.5 非同期通信処理と分析

非同期通信により、60秒に1回グラフやデータテーブルの部分のみがリロードされるが、上述の双方向機能を利用してデータを確認したり分析したりするためにユーザが画面を操作していても、このリロードが発生すると初期画面に戻ってしまう。WebSocket 技術を使えば非同期通信処理を利用することなく、サーバサイドの更新内容をフロントエンドにリアルタイムに表示させることが可能と思われるが WebSocket は Internet Explorer 9 に対応していない。ただし Internet Explorer 9 でも WebSocket と同様な動作を保証する SockJS-client という JavaScript のライブラリが存在しているようなので採用できるか検討したい。当面は非同期通信処理の発生間隔をもう少し長くするか、非同期通信処理を停止させるボタンをページ上に付加するなどの対応を検討したい。

4.6 JDBC 接続

ODBC 接続では Python の ODBC API から Symfoware

ODBCドライバを直接操作しようとする、先述の通り bit 数の違いによるアーキテクチャ不整合のエラーが発生したが、JDBC 接続であればそのような不整合エラーの問題は起こり得ない可能性がある。Symfoware には JDBC ドライバも用意されており、Python には JDBC API がある。JDBC 接続が可能となれば Excel 編集不能の問題が解消されるばかりでなく、同時アクセスのボトルネック問題が解消され、より強固で安定した Web アプリケーションを構築できる。ただし JDBC 接続を行うためには、接続によるリスクを洗い出した上で、リスク回避のための制限事項をシステムベンダ(富士通)と確認し了解を得る必要がある。

5. 考察

院内データ見える化サーバの構築は、システム部門の日常業務軽減に加えて、誰もが電子カルテ端末からいつでも最新のデータにアクセスできるようになったことで、院内データがより身近に感じられるようになり、経営陣のみならず一般職員の業務改善への意識向上をもたらした。病院には様々な指標が存在する。経営管理部門や診療部門と連携して効果的な「見える化」を推進していきたい。

6. 結論

院内データ見える化システムの構築により、これまで日報等にて日々作成・配信していた各種データがリアルタイムで病院職員に閲覧可能となり、システム部門業務が大きく軽減された。空床情報がリアルタイム表示されることでベッドコントロール業務がより効率的に確実に実行できるようになり平均病床稼働率の上昇に貢献した。

参考文献

- 1) 田中誠大. 当院における情報の「見える化」ツール構築による効果について. 第 36 回医療情報学連合大会論文集 2016.
- 2) Welcom | Flask (A Python Microframework).
[<http://www.mhlw.go.jp/stf/seisakunitsuite/bunya/0000039920.html>].
- 3) Flask on IIS - James Polera.
[<http://netdot.co/2015/03/09/flask-on-iis/>].
- 4) Chart.js | Open source HTML5 Charts for your website.
[<http://www.mhlw.go.jp/stf/seisakunitsuite/bunya/0000039920.html>].
- 5) pandas.DataFrame - pandas 0.20.3 documentation.
[<http://www.mhlw.go.jp/stf/seisakunitsuite/bunya/0000039920.html>].
- 6) Choosing Between ODBC and JDBC Drivers - MATLAB & Simulink.
[<https://jp.mathworks.com/help/database/ug/choosing-between-odbc-and-jdbc-drivers.html?requestedDomain=www.mathworks.com>].
- 7) Java アプリケーション開発ガイド入門編 - Fujitsu.
[<https://jp.mathworks.com/help/database/ug/choosing-between-odbc-and-jdbc-drivers.html?requestedDomain=www.mathworks.com>].

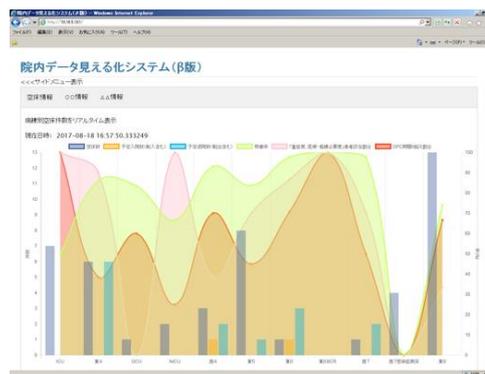


図1 トップページ1



図2 トップページ2