

この資料は日本Mテクノロジー学会員専用です。

この資料を学会員以外がコピーしたり、学会員以外に配布することを禁じます。

Copy right : M Technology Association – Japan

日本Mテクノロジー学会事務局

〒260-8677 千葉市中央区亥鼻 1-8-1

千葉大学医学部附属病院企画情報部内 鈴木隆弘

Tel: 043-226-2346

Fax: 043-226-2373

Email: mta-office@mta.gr.jp



MTA2016

第43回日本Mテクノロジー学会大会 講演論文集

2016年8月26日(金)～27日(土)

キャンパスイノベーションセンター東京（東京都港区）



大会長	インターシステムズジャパン株式会社	植松	裕史
プログラム委員長	愛媛大学医学部附属病院医療情報部	木村	映善
実行委員長	インターシステムズジャパン株式会社	石橋	賢二

巻頭言

第 43 回日本 M テクノロジー学会大会

メインテーマ

IoT 時代のデータマネジメント

大会長

インターシステムズジャパン株式会社

代表取締役社長 植松 裕史



この度、第 43 回日本 M テクノロジー学会大会の大会長を務めさせていただくことになりました。大会に先立ち、皆様にご挨拶を申し上げます。今回は大会テーマを「IoT時代のデータマネジメント」として開催いたします。

Internet of Things (IoT) ということばに代表される「すべてのものがインターネットでつながった社会」により、コンピュータ等のデバイスだけにとどまらず、世の中におけるさまざまな物体（モノ）が通信機能を持ち、相互にデータがやり取りされる時代になりました。自動認識や自動制御、遠隔計測などが可能となり、自動車の位置情報をリアルタイムに集約して渋滞情報を配信するシステムや、検針員に代わって電力メーターが電力会社と通信して電力使用量を申告するスマートメーターなどが開発されています。これまで不可能であったことが、普通に行われる時代となり、今日、その影響は産業界にも医療界にもきわめて広範囲に及んでいます。

そうした時代のデータマネジメントはいかにあるべきか。これは、非常に重要なテーマであるばかりでなく、従来とは異なった視点や考え方が求められます。新しい時代のデータマネジメントを可能にするテクノロジーの活用にも積極的に取組まねばなりません。日本 M テクノロジー学会がその牽引役となるべく、今大会がそのための有意義な場となれば幸いです。

ぜひ多くの皆様のご参加・ご協力を得まして、活発な議論が行われますよう、祈念しております。

第 43 回日本Mテクノロジー学会大会

大会プログラム

第 1 日目

9:30～	受付開始(学会会場)
10:00～10:05	開会挨拶 大会長 植松 裕史(インターシステムズジャパン株式会社)
10:05～11:15	基調講演 司会:土屋 喬義(土屋小児病院) 「ロボット技術とAI」 ～センサー社会がもたらすもの～ 中川 友紀子 先生 (株式会社アールティ)
11:15～11:55	教育講演 司会:川中 普晴(三重大学大学院工学研究科) アメリカのオープンソース電子カルテとIoT Atsushi Inoue 先生 (Eastern Washington University)
11:55～13:00	昼休憩 (日本Mテクノロジー学会評議員会:605 号室)
13:00～14:30	大会長企画 司会:植松 裕史(インターシステムズジャパン株式会社) 「IoT in Industry」(シンポジウム形式) 「IoT/ビッグデータ時代に効く! Dell のビッグデータソリューション」 ～エッジコンピューティングからリアルタイム分析までを実現するエコシステム～ 堀田 鋭二郎(デル株式会社) 「これから実現するIoT が創る新たな世界」 笠倉 英知(インテル株式会社) 「IoT 時代のデータ基盤の要件」 佐藤 比呂志(インターシステムズジャパン株式会社)
14:30～14:45	休憩
14:45～16:45	プログラム委員長企画セッション 司会:島井健一郎(千葉大学医学部附属病院) 「IoT 時代の医療分野の SOA とは」(ワークショップ形式) 山本 康仁(都立広尾病院) 木村 映善(愛媛大学医学部附属病院) 下川 忠弘(京都民医連中央病院) 島川 龍載(広島赤十字・原爆病院) 飯田 征昌(名古屋市立大学病院)
18:30～	懇親会(18:10までに竹芝栈橋集合)

第2日目

9:30~10:00	<p>学会長講演 司会:鈴木 隆弘(千葉大学医学部附属病院)</p> <p>学会長 土屋 喬義(土屋小児病院)</p>
10:00~11:00	<p>一般口演セッション1 司会:嶋 芳成(日本ダイナシステム株式会社)</p> <p>指定席券自動発行システムについて 山本 和子</p> <p>M 言語のデータベースにおける数値解析の方法の利点 高橋 亘(基礎コミュニケーション科学研究所)</p> <p>Node.js による Cache アプリケーションの作成経験 西山 強</p> <p>GT.M ルーチンを Cache¥ルーチンに変換するツール 鈴木 利明(日本ダイナシステム株式会社)</p>
11:00~11:20	休憩
11:20~11:50	<p>大会長講演 司会:石橋 賢二(インターシステムズジャパン株式会社)</p> <p>技術進化と未来 大会長 植松 裕史(インターシステムズジャパン株式会社)</p>
11:50~12:35	<p>一般口演セッション2 司会:土井 俊祐(千葉大学医学部附属病院)</p> <p>医療情報技師認定取得後に資格継続更新を行わない技師の傾向を探る 伊勢田 司(一般社団法人日本医療情報学会 医療情報技師育成部会)</p> <p>DPC データの openEHR/アーキタイプマッピングツールの構築 櫻井 理紗(兵庫県立大学大学院応用情報科学研究科)</p> <p>当院におけるデータベース運用の歴史の現在 渡部 功之(獨協医科大学小児科)</p>
12:35~12:40	閉会挨拶
12:40~13:00	日本Mテクノロジー学会 総会


基調講演
教育講演
大会長企画

基調講演

「ロボット技術とAI」 ～センサー社会がもたらすもの～

株式会社アールティ 代表取締役 中川 友紀子 先生

身近になってきた人工知能やサービスロボット、IoTについて、わかりやすく概要を説明します。また、センサーにはどんな種類があり、どうやってデータ化しているのか、医療介護現場向けにはどんな使い方ができるのかを実例を交えて解説します。これらのセンサーが生み出すビッグデータをつかったロボット、人工知能、IoTが医療介護現場にどのような影響をもたらすのか、現状と将来像についてご紹介します。



中川 友紀子
株式会社アールティ 代表取締役

1971年生まれ
1995年法政大学大学院修士課程修了 工学修士
1995年東京工業大学 大学院総合理工学研究科 助手
1998年科学技術振興機構ERATO北野共生システムプロジェクト研究員
2001年日本科学未来館展示サブリーダー
ロボットベンチャー等を経て
2005年株式会社アールティを創業
2006年初自社製品発売開始
パワーポイントと連動する教育用アームロボット
2007年アールティオフィスを秋葉原電気街に移転
以下、社歴参照 <https://www.facebook.com/RTnetjp/>

著書
日経Roboticsコラム連載 (2015年8月号から)
ロボットのいる暮らし (単行本・共著、2007年、日刊工業新聞社)
入門 Bluetooth(単行本・共著、2000年、技術評論社)

近年のロボット関連の取材でインタビュー、講演等
2015年 日本経済新聞社講演会にて講師
2015年 世界で知るべきロボット業界の女性25人に選出
(ロボハブ シリコンバレーロボティクスのWEBメディア)

専門分野
動画処理、マルチモーダル情景理解、ニューラルネットワーク、ファジィ知識工学、GA、自然言語処理、UNIX系ネットワークスーパーユーザー(基幹ネットワークからサーバー立ち上げまでフルスタックで対応可能)、科学館展示、ロボットを通じた科学技術教育
2015年より日本知能情報ファジィ学会副会長

2015年 日本経済新聞7月17日夕刊2面で紹介
2015年 日経MJでロボットバイオニアフォーラムの紹介記事掲載
(2014年以前の講演、論文、受賞多数のため割愛)

株式会社アールティ ホームページ：<http://www.rt-net.jp/>

Facebook：<https://www.facebook.com/RTnetjp/>

教育講演：アメリカのオープンソース電子カルテと IoT

A Report on Opensource EMR and IoT in the United States

井上敦司

Atsushi Inoue

東ワシントン大学（米国）

Eastern Washington University

Spokane, WA 99202, USA

キーワード: オバマケア、オープンソース、EHR、IoT

1. はじめに

アメリカにおけるオープンソース電子カルテと IoT (Internet of Things) の現状 (2016年8月現在) を既に公開されているデータや既知の事実を基に報告する。

2. アメリカの電子カルテの現状

過去8年間に及ぶオバマケアの推進により、電子カルテは急速に普及したと言える。米国連邦政府機関である ONC の報告によると2011年から2015年の4年間で全米の電子カルテ導入率が28%から84%に躍進した。多くの要因が絡んでいるが、特筆すべきは意味ある使用 (Meaningful Use) というインセンティブ (報酬) を導入した市場活性政策を連邦政府が取ったことである。参考までに、同時期の日本における電子カルテ導入率は50%から70%への躍進であった。

ベンダー別に見ると、EPIC がこのインセンティブプログラムに参加している民間医療機関全体の約25%に導入されており、寡占状態にあることが分かる。一方、少数ながらオープンソース電子カルテの導入も散見される。特筆すべきは PracticeFusion の2.5%である。OpenEMR も3件の導入が報告されている。VistA の導入例も数件報告されている。

主に患者に対する医療情報ポータルサービスを目指したウェブ化や管理労力やコストの軽減を目指したクラウド化が進んでいる。更に、iOS や Android といったモバイル用 OS に対応したアプリケーションも提供されている。それに伴う情報セキュリティ管理、特に個人情報の取り扱いに関する課題が多くなっている。

3. オープンソース電子カルテ

オープンソース電子カルテは多数存在しており、世界中で導入実績が多く報告されている。アメリカにおいては、その全体に占める割合は少ないものの、無視できない導入件数が報告されている。オープンソース技術は現在の IT の骨幹技術の多くを成しており、ベンダー固有の技術の競争相手というよりも、むしろそれをベースに自社独特の製品開発はサービス提供を行う、いわば共存・協力する対象として認識されている。

米国において、このインセンティブを利用した電子カルテの実践導入には導入対象であるソフトウェアに対する公的な承認が必要である。当然、オープンソース・ソフトウェアも例外ではない。残念ながら、この承認を得るための時間と労力は大きいので、オープンソース・ソフトウェア開発コミュニティにとっては大きな負担 (特に財政面) となる。このため、ごく一部を除き、オープンソ

ース電子カルテの導入は不適切となってしまう。承認を受けたオープンソース電子カルテは幾つか存在する。その代表例が OpenEMR と VistA である。

VistA は米国退役軍人病院（即ち、連邦政府機関である）の公認電子カルテであり、会計や退役軍人の病歴管理など、関連必要機能を多く持つ病院情報システムである。オープンソースとして、希望者へは無償で提供される。MUMPS ベースのシステムであることも特筆すべきである。退役軍人病院は連邦政府の外郭機関であり、このインセンティブの対象とならないため、導入数を比較することは適切ではないとされている。また、OpenVistA や WorldVistA など、オープンソースソフトとしての提供や導入をサポートする動きもある。近年、肥大したシステムアーキテクチャのテコ入れや現在の ICT に批准するための改良や、高等教育機関で医療情報技術者を育成するための教材開発を目的とした非営利機関 OSEHRA が設立された。

OpenEMR はドイツでベンダー固有の電子カルテ製品として開発されたソフトウェアがオープンソースとして無償提供されたものがベースとなっている。開発コミュニティが成熟しており、今はそのコミュニティで改良や開発が行われている。米国のインセンティブプログラムのための承認は OEMR という非営利団体を通じて獲得・運営されている。オープンソース電子カルテ導入実績は OpenEMR が世界中で一番大きいとされている。

4. 医療関連 IoT

近年、IoT ブームに乗って、多くの医療関連計測機器の IoT 化が進んでいる。それを目的としたベンチャー設立事例が多く見られる。しかし、その多くは一般向けの健康管理機器（例：fitbit）や教育、研究目的のものが多く。

医療現場では旧態依然とした計測機器が多く用いられており、結果は紙へ印刷され、それを手入

力やスキャンイメージのアップロードといった形で電子カルテへ付与されている。しかし、これは機器買い替えの周期や医療従事者の教育の問題であって、IoT 自身の技術的な問題ではない。

従って、オープンソース医療関連 IoT 技術を語るには早すぎるが、教育や研究開発を目的とした関連技術は特筆に値する。ここではラズベリーパイやアルデュイーノについて報告する。オープンソース技術のみで、電子カルテ、（患者や職員向け）教育管理システム、IoT 計測機器をうまく統合したシステムが容易にかつ手軽（低価格で）に開発できる事例を紹介する。

5. 提案

オープンソース技術が核となる 2 つの提案を行いたい。まず、開発や導入の無理、無駄、ムラを省くための手法（Lean Development）とその推進政策、そして、医療情報技術者育成のための教育である。どちらもオープンソース技術があって初めて成り立ものであり、未来の技術開発やサービス提供のあり方として、真剣に検討されるべきであると考えます。

6. おわりに

アメリカにおける電子カルテ導入の現状とオープンソース電子カルテと医療関連 IoT の現状について報告を行った。そして、未来へ向けて検討すべき事を提案した。

講演者紹介

東ワシントン大学経営・公共管理学部情報システム・ビジネス分析学科教授。専門は人工知能、特にファジィ理論とその応用、及び情報セキュリティ管理。人に優しい情報通信技術を目指して、金融、営業、エンターテイメント、社会インフラセキュリティ管理など多岐にわたる応用分野を手掛ける。近年、オープンソース技術の積極的利用を目指した教育活動やデジタルアントレの推進を

中心に活動中。1999年シンシナティ大学工学
部情報工学科博士課程修了（Ph.D.）。

連絡先：

メール：ainoue@ewu.edu

inoueatsushij@gmail.com

ウェブ：<http://www.inoueatsushi.net/>

<http://inoueatsushi.wordpress.com/>

大会長企画

IoT in Industry

司会：インターシステムズジャパン株式会社 植松 裕史

本大会は Internet of Things (IoT) を大きなキーワードとして取り上げております。大会長企画セッションでは、特に産業界にスポットを当て、グローバル IT 企業 3 社の担当者をお招きし、IoT にかかる先進事例をシンポジウム形式でご紹介いただきます。

【講演 1】

『これから実現する IoT が創る新たな世界』

インテル株式会社

ダイレクト チャネル セールス

リージョナル IoT 事業開発マネージャ

笠倉 英知 様

<概要>

インテルが実現しようとしている IoT ビジネスにおける戦略、そしてパートナー連携の重要性をご説明します。

【講演 2】

「IoT/ビッグデータ時代に効く！Dell のビッグデータソリューション」

～エッジコンピューティングからリアルタイム分析までを実現するエコシステム～

デル株式会社

エンタープライズ・ソリューションズ&アライアンス部

BigData ビジネス開発マネージャー

堀田 鋭二郎 様

<概要>

IoT、ビッグデータを活用するためには、データの流れ、つまりは ① 収集 ② 蓄積 ③ 統合 ④ 加工 ⑤ 管理 ⑥ 処理 ⑦ 分析 のデータの動きを理解する必要があります。デルは そのデータの流れを End to End でサポート可能な豊富な自社製品ポートフォリオを保有している唯一のハードウェア・ソフトウェア会社へと変貌しています。様々な選択肢がある昨今、どのようなアプローチが活用方法として最適解となりうるか？一例を交えながらご説明申し上げます。

【講演3】

『IoT時代のデータ基盤の要件』

インターシステムズジャパン株式会社

ビジネスディベロップメント シニア・マネージャー

佐藤 比呂志 様

<概要>

すべてのものがつながる IoT の世界では処理すべきデータの量の爆発とともに、そのデータ処理の質にも大きな変革の時が来る事が予想されており、対応する IT 基盤、その中でもとりわけデータ基盤の変革が急務です。本講演では IoT に必要なデータ基盤の要件を明らかにしつつ、インターシステムズ社がそのチャレンジにどのような戦略およびどのような製品で対応しようとしているかを先進事例の紹介を交えながら説明していきます。

プログラム委員長 企画セッション

Stream Computing を医療情報システムに導入する試み

Introducing streaming computing to medical information system

木村 映善¹⁾ 蒲生 祥子¹⁾ 石原 謙¹⁾

Eizen Kimura¹⁾ Souko Gamo¹⁾ Ken Ishihara¹⁾

愛媛大学医学部附属病院医療情報部¹⁾

Dept. Medical Informatics of Ehime University Hospital¹⁾

キーワード: Stream Computing、Internet of Things、MQTT、CDSS

1. はじめに

従来、我が国の医療情報システムはオーダリング・電子カルテシステムを中核に、周辺部門システムとメッセージ交換する構成をとってきた。この仕組みは、データ発生源が人間による手入力、部門システムからの結果通知、あるいは部門システムで保持している情報を参照するためのデータ所在等の情報を受け渡すという範囲に限定して、有効に機能している。一方、リアルタイムでデータを蓄積するシステムとして、重症・急性期患者情報システム、分娩監視システム、生体情報モニタ等がある。しかし、これらのシステムは単独に閉じており、また収集したデータをリアルタイムの利活用に供することを想定して設計されていない。すなわち、現状は過去の情報に基づいた分析と予測は可能であるものの、現在進行形で起こっている事柄を把握し、医療従事者の時機を得た介入を促せるようなシステムを実現するためのアーキテクチャを意識した設計にはなっていない。

データの発生源を大きく分類すると、(1)医療機器による撮影・測定、(2)部門システムからのデータ発生 (例:検査結果)、(3)医療従事者や患者の動態、(4)患者のバイタル、(5)医療従事者による入力に分けられる。前述したように、連続したデータは(1)において部分的に実装されているが、残りは発展途上の領域である。人間に可能な限り行動の制約を加えずに連続的な情報収集をする手段と

して、スマートフォン、ウェアラブルコンピュータ、センサーといったテーマに関わる情報処理技術が貢献することが期待されている。しかし、各技術群がリアルタイムのデータ処理を想定したアーキテクチャの構成要素として組み込まれるに至っていない。つまり、医療分野においてリアルタイムで流れるデータを標準的な手法で収集し、管理するアーキテクチャの概念は完成されていない。学術的にリアルタイムに情報収集し、異常・変化の早期検出・介入させることでアウトカムを向上させる試みは行われているが、その成果が実用化されて医療現場に普及しないのは、前述したように情報収集と処理の仕組みが標準化されていないために、実現までの実装コストが高くつくことが大きな要因と思われる。

そこで、一例として以下のようなシステムを想定し、アーキテクチャの要件検討をする。医療従事者や患者の動態・バイタル情報をリアルタイムで収集する。臨床判断支援システムは収集されたバイタル情報を処理し、必要に応じて医療従事者に介入を促す。そのようなシステムにおいて共通して求められる要件を整理し、プロトタイプ構築を試み、その経験をもとに今後我々が取り組むべき課題を提示する。

2. 方法

前述した、リアルタイムにデータを生成する発

生源から、継続的にデータ分析して臨床判断支援システムに連携するための汎用的なアーキテクチャについて考察する。本アーキテクチャに必要な構成要素は、機器・システムが通信するための仕組み、それらの通信を信頼性のあるものにする仕組み、連続的なデータ分析をする仕組みである。そしてそれらの仕組みは汎用的なものであり、様々なシステム構築をする際においても基本的な構成関係の変更が生じないものであることが望ましい。

(1)機器・システムが通信するための仕組み

Machine to Machine (M2M)は個別に稼働している機器同士が、自律的に生成されたデータをリアルタイムにネットワーク上でデータ交換、処理するシステムとしての概念である。そして、Internet of Things (IoT)は、比較的閉域的環境での運用を念頭においていたM2Mの概念を、インターネットという舞台上にあらゆる物にコンピュータが組み込まれた状態にまで世界観を広げたものである。IoTは来るべき高度情報化社会を表現する重要なキーワードとして扱われており、法律の文言でも引用されている。2016年4月には、国立研究開発法人情報通信研究機構法及び特定通信・放送開発事業実施円滑化法の一部を改正する等の法律（平成28年法律第32号）において、国立研究開発法人情報通信研究機の一業務として新技術開発施設供用事業を定めている。その対象として「インターネット・オブ・シングスの実現（インターネットに多様かつ多数の物が接続され、及びそれらの物から送信され、又はそれらの物に送信される大量の情報の円滑な流通が国民生活及び経済活動の基盤となる社会の実現をいう。）が定義されている。先述したとおり、あらゆる機器に組み込まれデータを通信する世界を想定するのであれば、(1)多くの機器を運用するために非力な計算処理環境でも稼働できること。省エネルギーの稼働ができ、設定が比較的容易であること、(2)他機器やシステム連携

の障害が相互に影響しないような疎結合を実現出来ること、(3)重要な情報は可及的に確実に伝えられる仕組みをもつことが求められる。

(2)通信を信頼性のあるものにする仕組み

機器から連続してデータを受信し、データの蓄積（監査証跡、履歴記録、診療録への反映）、データ可視化（モニタリング）、データ分析（データウェアハウス）、臨床判断支援システムに分配する仕組みが必要である。多数の機器とのデータ送受信を確実にやり、かつ即時性のある処理を実現するために、メッセージ処理のスループットは大きく、かつ低遅延で処理されることが望まれる。また、あるシステムが障害を起こしても、可能な限り他の機器やシステムの稼働に影響を与えないような疎結合的連携を実現する仕組みが必要である。例えば、2者間の連携において、メッセージを受け取る側が稼働を停止した場合でも、もう片方はその影響を受けずに継続してメッセージを送信することが可能である。逆にメッセージを送信する側が停止しても、メッセージを受け取る側も影響を受けない。このような状態を2者間に限らず多対多の関係でも実現する必要がある。即時性かつ低遅延である通信手段の一つとしてソケット通信が使われるが、2者間が正常に稼働し続けていることが前提である。システム障害に対応するためには双方に障害を想定したメッセージの一時的待避、再送処理のロジックを組み込む必要がある。組み込み機器での処理の負担や、多対多における通信の仕様調整を検討すると現実的ではなくなる。機器・システム間が直接通信する通信プロトコルを採用するのではなく、システム間の通信を介在し、信頼性がありかつお互いのシステムの状態に依存しない通信環境を提供するミドルウェアの存在が不可欠である。

(3)連続的なデータ分析をする仕組み

時系列的に連続して流れるデータをストリーム

データと呼び、ストリームデータをリアルタイムに分析・利用する処理をストリーミングコンピューティング(以下、SC: Streaming Computing)という。SCにより、現実世界で発生した事象をリアルタイムに認識し、迅速に対応することが可能になる。従来のデータ処理は、過去に蓄積されたデータを分析したり、大量のデータを効率的に処理したりするためにバッチ的処理を分散処理するアプローチが主流であった。一方、SCでは、リアルタイムに起こるイベントを処理、分析するために連続的な処理を行う必要がある。擬似的なリアルタイム処理として、これまでのアプローチの処理を細かく短時間に頻回実行することでも実現可能であるが、ストリームデータの異常、中断への対応や、複数の処理を連結させるための標準的な仕組みがないとシステム構築の工数が増加する。すなわち、流入するデータの捕捉、データを処理するロジック、処理結果の出力を標準的な枠組で提供し、その上で実装されるデータ処理のロジックがデータ流入元、出力先のシステムの状態やデータの利用可能性を意識することなく、与えられたデータの処理のみに専念できる環境が必要である。

(4)上記の3つの仕組みが医療分野に必要な背景

入院中の患者は病床、病棟、手術室、ICUと様々な場所を移動しており、患者に関する情報を収集する方式も一通りではない。バイタルサインで例示するならば、病床における看護師によるRFID搭載機器(体温計、SpO2)を用いた測定、ウェアラブルデバイスやモニタによる連続的測定がある。様々なデバイスが患者に付帯されうるが、そのデバイスによって測定される情報がどこに送信されるかを医療従事者に意識させないことが望ましい。システムごとに個別の送信設定を医療従事者にさせることは業務上の負担増や間違った設定によるインシデント増の可能性があるのである。一方でシステム側から見て、入力源がなんであろうと、同じ患者の同種の計測結果は同一

のインタフェースで情報収集されることが望ましい。例えば体温の遷移を、看護師のデータ入力、RFID対応機器からの入力、ウェアラブルセンサー、モニタの区別なく、同一の体温データとして一本化して受け入れて、時系列的分析に集中できる仕組みが求められる。また、システム全体を俯瞰した場合、このような多様性に富む機器群と各システムの多対多の連携を、医療従事者やシステムインテグレータが複雑な手続きなく実現できる仕組みが望まれる。そして、長時間にわたる入院があり、クリティカルケアの局面もあるため、長時間の安定稼働と信頼性のある通信が必要である。そして、臨床判断支援システムのロジック等が、医療機関を越えてポータビリティを確保できるように、標準医療情報モデルとの親和性を意識しつつ、標準規格を組み合わせたアーキテクチャの設計が望ましい。

3. 結果

(1)MQTT

M2Mの通信プロトコルとしてMessage Queue Telemetry Transport (MQTT)を採用した。MQTTはISO/IEC PRF 20922で策定されている、TCP/IPプロトコル上に実装された軽量なメッセージプロトコルである。非力なデバイスや省エネルギーのためにメッセージ構造や通信が簡素なものに設計されている。HTTPとMQTTの比較では、MQTTの方が双方向通信、一方向通信ともにより省エネルギーで可動し、単位時間に交換可能なメッセージ数も多いことが確認されている。TopicベースでのPublish/Subscribe (Pub/Sub)型データ配信のミドルウェアと併用することで、1対1、1対N、N対Nの即時性のあるメッセージ配布が実現可能である。また、メッセージ配送のQuality of Service (QoS)として3段階(0:配送保証なし、1:配送保証するが重複可能性あり、2:配送及び無重複を保証)が選択可能であり、QoS1あるいはQoS2のグレードが医療アプリケーションの品質に合う。

(2) Message Queuing System

MQTT メッセージを仲介するミドルウェアとして、VerneMQ を採用した。(1)オープンソースであり、研究、実運用に無償利用可能であること、(2)QoS1,2 をサポートしていること、(3)WebSocket をサポートし、HTML5 対応 Web ブラウザにも対応できること、(4)クラスタ構成を組み、冗長性、パフォーマンス上のスケーラビリティを確保できることを選定条件とし、2016 年時点で VerneMQ、eMQTT が条件に合致していた。Pub/Sub 型のメッセージ交換モデルでは、メッセージを送信する側を Publisher (Pub)、受信する側を Subscriber (Sub) と定義し、Pub は特定の宛先ではなく「トピック」を指定してメッセージ送信する。Sub はメッセージ受信したいトピックを指定する。この Pub、Sub の関係はそれぞれにおいて多対多の関係を定義可能である。2-(4)節で言及したように、同一の測定項目を複数機器から採取して患者ごとに同一のストリームに集約するには、複数の機器が Pub として参加し、例えば「患者 ID/測定項目」の階層化されて命名されたトピックに Publish する。そして臨床判断支援システム等、データを受信するシステムは監視対象となる患者のトピックを Subscribe する形で実現する。

(3) Apache Spark

Apache Spark は Hadoop 等の Map/Reduce ベース分散処理システムの弱点であった、秒単位での低レイテンシかつ連続的な処理や反復処理への対応を可能にした処理系である。Resilient Distributed Datasets (RDD) というデータセット単位で繰り返し処理を行うフレームワークを提供する。さらに Spark の機能を利用してストリーム処理をするためのライブラリとして Spark Streaming が提供されている。また、コミュニティによって機械学習アルゴリズムを適用できるようにする MLlib 等のライブラリが提供されており、今後の機械学習を用いた研究への布石として採用した。

(4) 検証用システム

当院の既存のバイタルサイン収集の仕組み（青色）に加えて、今回のアーキテクチャ検討のためのプロトタイプを追加した（橙色）。現在、ネイティブにデータを MQTT プロトコル経由で出力できる医療機器やモニタが存在しないため、既存のバイタルゲートウェイと電子カルテシステム間を取り持つ連携システムに手を加えた。すなわちバイタルゲートウェイから電子カルテに送信される、バイタル情報が格納された XML ファイルから JSON メッセージに変換し、MQTT プロトコルで送信する仕組みを開発した。将来、医療機器からネイティブに MQTT 送信するようになって、今回の仕組みを改変することなく統合可能であり、過渡的な運用を容易にする。MQTT ミドルウェアにおける Pub/Sub の設計は、バイタルサインのデータソースを表すトピック、topic://vital/spotcheck に publish し、そのデータを必要とするシステムから当該トピックを subscribe させる。今回はストリーム処理をする処理系である Apache Spark とバイタルサインのログを保管するための fluentd が subscribe している。Apache Spark Streaming はデータソースとして MQTT をサポートしており、MQTT 経由でバイタルデータを読み込み、閾値以上血糖値が検出された場合は警告を表示する簡易なロジックを実装した。

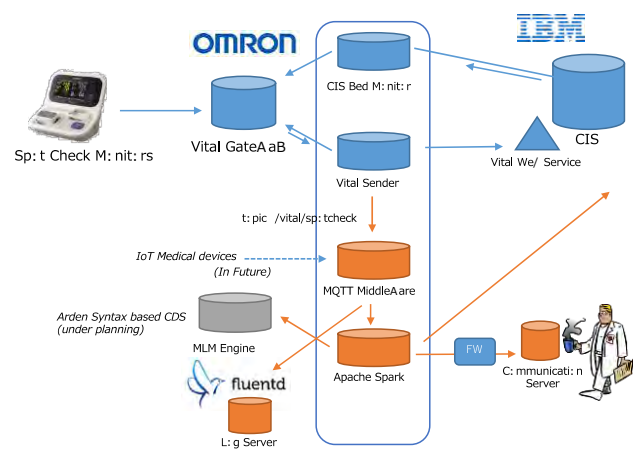


図1 検証環境

4. 考察

(1)トピック名のセマンティクス

今回は基礎的調査のために既存のシステムのデータを SC 環境に直結させる設計をした。しかし、以下に述べる理由で汎用性がない。(1)観測対象と観測者それぞれの主体と相互の関係の定義が厳密に考慮されていない。通常、観測対象と観測者の関係は多対多となる。SC は同時に1つの対象にたいして継続的処理を行うため、ある患者に関する観察対象群のなかから対象となるものを選び、1つの SC 処理を対応付ける必要がある。例えば、「1人の患者」から継続的に得られる計測データは心拍、呼吸、体温、体動等がある。それに対して、呼吸の継続的監視、転倒リスクの早期検出、感染症の早期兆候検出といった様々な監視アルゴリズムが考えられるが、それぞれのデータとアルゴリズムを処理する SC を紐付ける標準的な手法を提唱していない。(2)トピックのパス命名規則の標準化がなされていない。今回はバイタルサインデータのデータソースを表す意図でトピック名を"/#{patient_id}/vitalsign/spotcheck"とした。これは、今後、バイタルサインのデータが取れる機器が増え、例えば、RFID 搭載温度計(thermometer)、ベッドサイドモニタ(bedside monitor)等が接続されるようになった場合、それぞれのシステムからは、"/#{patient_id}/vitalsign/thermometer、"/#{patient_id}/vitalsign/bedside_monitor の各トピックに publish する。バイタルサインのデータ収集側は、"/#{patient_id}/vitalsign/#というワイルドカード指定で、全てのトピックを同時に subscribe することにより、あらゆるバイタルサインのデータを一つのトピックのように扱うことができるということを想定していた。この想定では、トピック名中の”vitalsign”が、「バイタルサイン」を扱うトピックを指定するものであると概念的に合意されている必要がある。一方で、「バイタルサイン」は今のところ便宜的につけた観測項目の集合名にすぎない。トピック名に観測項目の集合体の意味を持たせるのなら、どの項目がどの集合体に所属するのかを標準化

しなければならない。また、観測対象および観測者が新たにその定義に従って実装しなければならない、実装コストは大きなものとなる。それよりは、後述するようにメッセージが包括的に医療情報を記載できるものとし、どの内容を受け取るべきかを観測者が取捨選択する方法を取る方が望ましいと思われる。その場合は、トピックス名は患者ごとに、そのメッセージを規定する情報モデルの種類に即した命名をすることが望ましいと思われる。例えば、生理検査の連続的データは、HL7 メッセージであればメッセージタイプによる分類をし、"/#{patient_id}/ORU^R01 というトピック名となり、openEHR の archetype ベースであれば、"/#{patient_id}/openEHR-EHR-OBSERVATION.pulse.v1 というトピック名を作成する。同時に複数項目を観測したい場合は、観測に必要な分のトピックを同時に subscribe する。

(2)医療情報モデル

連続してデータを送信し、また受信してデータ処理をするには、可能な限り双方にデータ変換やマッピングを要求しないように標準情報モデルや標準プロファイルを採用することが望ましい。IHE Patient Care Device Technical Framework (IHE PCD TF)における PCD (Patient Care Device)において、患者ケアデバイス(Patient Care Device)とシステム(Enterprise)との通信(Communication)を定義する Device Enterprise Communication Profile が定義されている。本プロファイルでは、PCD の情報モデルと専門用語の定義は ISO/IEEE 11073-10201 と ISO/IEEE 11073-10101 に準拠することとし、これらの規格で定義されている DIM (Domain Information Model)を HL7 v2.6 にマッピングすることと定めている。XML を用いた通信でも、XML 本文に HL7 メッセージを原形のまま埋め込んだ様式を提案している従って、観測対象をモニタリングするセンサーは HL7 v2.6 以降のメッセージを生成して送信するか、ゲートウェイを通して HL7 形式に変換するような設計を意図するべきであろう。また、PCD プロ

ファイルでは PCD と Enterprise 二者間のスポットでの通信のみを想定しているようであり、多対多の通信、連続的な通信については触れられていない。この部分の調和を如何にとっていくかが今後の課題である。

(3)通信プロトコル

2016年7月時点での IHE PCD TF 公開版では、Minimal Lower Layer Protocol (MLLP)、Web Service 技術群 (WS*)、そして警告の送信に Wireless Communication Transfer Protocol (WCTP)の3種類のプロトコルが定められている。いずれも中核となる情報モデルは HL7 メッセージであり、それぞれソケット、Web サービス、HTTP(S)が使われている。MQTT 等のメッセージキューイングシステム等の記述はない。信頼性があり、かつ軽量なプロトコルの追加・採用を検討することが望ましい。あるいは、PCD で規定された従来の通信プロトコルを採用し、メッセージを受け取る側の Enterprise において、本稿で提案しているアーキテクチャに合致するように MQTT プロトコルに変換して送信するという運用も考えられる。

5. 総括

リアルタイムで医療情報を処理するシステムのアーキテクチャを検討した。本アーキテクチャの構成要素には、機器・システムが通信するための仕組

み、それらの通信を信頼性のあるものにする仕組み、連続的なデータ分析をする仕組みが含まれる。M2M/IoT 時代向けに低処理性能な機器や省エネルギーに特化した通信プロトコルである MQTT を選択した。システムの状態に依存しない安定した通信、機器と情報システムの多対多の実装するために Pub/Sub モデルの通信形態を提案した。ストリーム処理の要件を確認するために具体的な実装として Apache Spark Streaming を利用した。結果、ストリーム処理は同時に一つの観察対象の特定項目群を連続的に監視するような作り込みをするものであることを確認した。同じ観測項目でありながら様々なデバイスから収集されるものを多対一で集約し、その観察対象を指定するためのセマンティクスが必要であることを確認した。その観察対象を指定するための手段として、標準情報モデルのメッセージタイプやモデル名を指定することを提案した。一方で、IHE PCD プロファイルにおいては多対多の通信は想定されていないため、本システムのようなアーキテクチャと PCD プロファイルとを併せた運用について検討を詰めていく必要がある。

6. 謝辞

本研究にあたり、オムロンコーリン社にバイタルサイン連携サーバに関する仕様開示及び技術的協力を頂いた。

IoT 時代の医療の変化

SOA のための ESB の拡張

Internet of Things Era to Rapidly Change Healthcare services

Extended Enterprise Service Bus for SOA

山本 康仁¹⁾

Yasuhito Yamamoto¹⁾

東京都立広尾病院¹⁾

Tokyo metropolitan Hiroo Hospital¹⁾

キーワード: SOA, ESB, EAI, 電子カルテ、DWH、位置情報

1. 背景

電子カルテパッケージを俯瞰すると、構成によっても異なるが、生理検査、検体検査、内視鏡、病理、手術実施、放射線実施、画像管理、リハビリ、給食など多くのパッケージソフトが放射状に接続されたハブ&スポーク型の EAI (Enterprise Application Integration) を構成している。接続方法はソケット通信で、半固定長のベンダー独自技術によるものが多く、ベンダーロックインの弊害や、集中処理による可用性のボトルネックが問題になっている。接続部分の変更には考えた以上の手間と費用を要し、EAI 本体と外部システムの更新時期が異なると、インターフェースフォーマットの変更が出来ず、機能向上を断念することもある。

これに対して、バス型のアーキテクチャをとる

ESB(Enterprise Service Bus)は、ルーティングやデータ変換などを分散処理し、疎結合でサービスの連携が可能で、XML など標準的で可塑性の高い技術を基本としているため、変更が強く、低コストで更新のズレを吸収できるなどのメリットがある。

では実際 ESB を導入できるのかというと、そこにはハードルが存在する。現状の日本型電子カルテパッケージは EAI を構成し、ESB を二階建てで導入する費用対効果面での説得性に欠ける。大手 EAI

の場合、接続インターフェースがデファクトスタンダード化しており、逆に標準的接続を仕様に盛り込むと価格が上昇してしまうことがある。最大の問題は SOA 的視点から大手電子カルテパッケージを俯瞰すると、パッケージ内部のツールの細分化や再利用が出来ておらず、まして外部から一部分を疎結合で利用することが出来ない。たとえば、放射線実施システムで入力された造影剤の副作用情報は電子カルテ本体側に反映するすべは無く、電子カルテの副作用入力部分を実施システム側で切り取って使うことも出来ない。内視鏡実施システムでの使用薬剤の入力は、電子カルテのそれとは異なり、データ連係は一方通行でアレルギーチェックロジックも適応されないばかりか、後発医薬品マスターの反映すら手作業となっている。

しかし、データ活用にはシステム全体を網羅するものが必要であり、本邦でも近年やっと電子カルテ以外にも拡張された DWH が入手できるようになった。当院では、病院全体を俯瞰する目的で 10 年前からリアルタイム DWH を導入、開発を続けているが、これは必要に迫られて複数のシステムの情報を収集するに至った。拡張の結果、ESB に近い状況にあるので報告する。

2. 目的と方法

急速な高齢化など人口動態の変化に病院経営の環境は年々厳しさを増し、前年同月比を各科ごとに比較するような従来の評価では不十分で、病床機能報告や DPC の調整係数などに対応するためには、俯瞰的かつ緻密な病院機能の評価と舵取りが必要とされている。医療制度によって大きく変化する医事会計処理後のデータではなく、数年にわたって同じ基準で評価できる発生源をソースとした、病院横断的な情報の評価が求められる。

診療判断支援システムとして端を発した当院の HiPER2.0 は、それがもつリアルタイム DWH 機能とメッセージプロセス機能を生かして病院経営判断に利用されている。情報収集範囲は電子カルテだけでなく、生理実施、病理レポート、内視鏡実施、重症系生体モニター、電話交換機、無線位置情報装置群を網羅している。接続はリアルタイムであり、電子カルテと重症形生体モニターは TCP/IP socket、生理実施画像システムとは XML によるファイル交換、各々の電子カルテクライアントとは http リクエストによる連携、病理と内視鏡レポートシステムとは web scraping、電話交換機とはシリアル通信などと、多彩な方法を用いている。

開発プラットフォームは FileMaker 社製の FileMaker Pro 11 と Apache Tomcat、MSIS 社の JAVA ベースミドルウェア、エーアイ社の音声合成および IVR システムで構成され、VMware 社の仮想化技術を用いて、4 台のブレードサーバー内に 13 インスタンスの Windows Server 2008R2 を用意、並列分散処理を行っている。本邦において FileMaker Pro はコンシューマ用小規模データベースソフトウェアとして認識されているが、そのデータベースエンジンの内部構造は一般の RDB と異なりツリー型で、事前に想定されるクエリに対しての索引を自動的に生成することで、比較的高速に結果を得ることが出来る。また 1 フィールド長を事前に定義すること

なく最大 2 GB まで利用でき、オブジェクトも格納できるが、ツリー構造であるためガーベジコレクションなどのメンテナンスが自動化されていて管理がしやすいなどの特色をもつ。データベースエンジンと処理ロジックが一体化しているため、演算処理時のキャッシュ取得が最適化しているというメリットがあり、プログラマがキャッシュの取得や解放を考慮しなくても自動的に行われる。同時に強力なキャッシュコヒーレンスを持つことから、複数の処理ロジックを並列処理させたときに処理効率が向上し、処理レイテンシが短縮できる。こうした利点を生かしつつ、FileMaker Pro のスクリプト言語を使い、文字列操作を中心にしたメッセージ処理系を構築した。(図 1)

この処理系を用い、電子カルテはじめとする病院全体の複数システムの情報を取得し、最長でも数秒以内の遅延でビジネスロジックを完了することが出来た。自然文解析や処方を変更した場合の病態的原因の類推、GIS マップ処理のための住所のクレンジング、過去に観測した 250 万台のモバイルデバイスと院内に持ち込まれた千台以上のデバイスの照合など、様々な演算を遅延無くリアルタイム処理している。

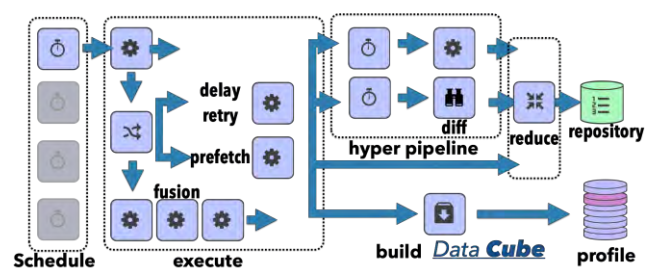


図 1 メッセージ処理部分の模式図

2. 成果

情報粒度やフォーマットはそれぞれ異なるが、トランザクションは一日約 13 万件あり、それとは別に 1000 台以上の無線デバイスを 1 分おきに追跡している。処理は多岐にわたるが、基本的に患者、

医療者、医療行為の視点で情報追跡ができるように電子カルテは別に情報を蓄え、単純な情報の交換や変換にとどまらず、ビジネスロジックを処理できる。特に分散処理と遅延処理、パイプライン処理を得意としており、数秒から数ヶ月におよぶ遅延処理を可能としているだけでなく、処理ログ解析や可視化を備え、ロジックの追跡が可能となっている。(図 2 および 3)



図2 ビジネスロジックの追跡画面

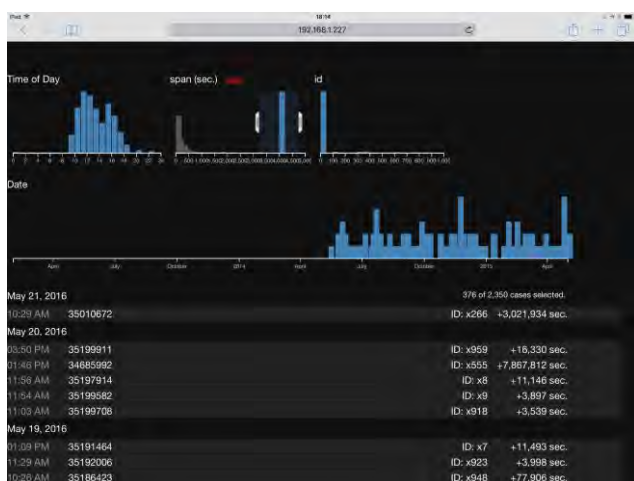


図3 頻度解析画面

4. 考察

SOA の観点から ESB が持つべき機能として、アプリケーションサーバーやデータベースを流れるサービスをモニタリングする仕組みが必要だ。ESB の技術的要件として、ポリシーに従いロードバランスを行うであるとか、フェールオーバーを指示するなどあるが、サービスという括りで考えるのなら、

次元の異なる処理が必要になるだろう。当院で実際に HiPER を運用したとき、システム単位では整合性があるが、システムを跨いだサービス単位でみると、情報の欠落や矛盾といった、運用に端を発した不整合に遭遇する。すなわち、病院という特殊な現場では、システムへの入力よりも医療行為が重視され、自動入力も乏しい。操作が省略され、順番が入れ替わるなどが普通に発生する。とくに局所最適化しやすい部門専用システムでは見かけ上業務が完結したように見えるが、情報処理的には破綻している。患者個人の時系列も、患者横断的な一覧の両方の閲覧性の低いパッケージソフトウェアでは、破綻が気づかれぬまま放置されることも経験する。

このような事象を解決するには、ESB にトランザクションを監視するだけでなく、次の 2 点の機能が必要であると考え。一つは矛盾や欠損を発見し、補完するビジネスロジックと、IoT を代表とするセンサー技術を応用した情報収集の自動化と監視技術である。

そこで、当院では HiPER を用いて業務プロセスを監視している。診療判断支援機能を流用し、情報収集の漏れをリアルタイムで連絡、運用を正常に保つよう医療者に介入する仕組み用意した。また複数の情報から欠落データを合成し、補完するなどの対策を講じている。それらを実現するには、そもそも欠落や矛盾に気づく必要があり、情報を収集して俯瞰して見せることが必要である。一般の ESB にあるようなトランザクションの可視化では不十分で、医療データを中心としたトランザクションのみではなく、プロセスの主体である医療者の状況をも追跡し、可視化することに注目した。その手法として、初期より PHS 電話機の登録基地局情報を活用した医師の位置情報追跡と電子カルテ端末のリアルタイムトランザクションを利用した医療者追跡機能を搭載している。

また複数のソースを統合した各種実施情報の追跡の精度を上げてプロセス監視している。たとえば手術予約情報と手術実施情報（会計情報）の突き合わせだけではなく、手術記録記載と入室認証の複数ポイント結合や、医師記載と実施情報による受付漏れ補完などある。

業務追跡は最終的に積み上げられ、日時レベルでの診療実績情報として可視化され、チェックサム形式で業務追跡の精度の管理を行っている。プロセス監視は、一つの実施行為のタイムスタディの粒度から、最終的には入退院レベルでの年報の粒度までの多段階の可視化を行えるように拡張した。

しかし、こうした管理手法はあくまで、一部能動的に取得することを除き、プロキシベースである。そのため、業務端末がサーバーエンドクライアント方式だと、保存タイミングまではプロセス介入や監視をすることが出来ない。保存後初めてトランザクションが発生し、それを解析し即座に対応したとしても、医療者がすでに離席していることを考慮する必要があった。そのために、音声と電話を利用したプロセス介入手法を開発運用している。しかし本来スマートな介入は、シームレスな端末レベルでの連携である。

そこで、技術試験ではあるが業務端末にエージェントを組み込み、ESB側からリモートでログイン状況や患者選択状況を監視、プッシュで電子カルテアプリケーションのモジュールを操作できる環境を作成した。具体的には特定の薬剤指示に対して web フォームを起動する、あるいは web フォームの選択で電子カルテオーダーツールを起動するなど、SOA 的な運用をシームレスに実現出来た。ただし、セキュリティポリシーを理由に現時点、運用には至っていない。

2点目として IoT などセンサー技術を利用した情

報収集の自動化の試みについて触れたい。

前述の通り業務プロセスに医療者の思考や状態が深く関わることに注目し、より精密なリアルタイム情報を取得するために無線デバイスの位置情報を試験的に取得し可視化している。現時点では観測開始時刻や滞在時間、移動状況や過去の出現状態から、匿名情報のまま患者持ち込みデバイスと職員デバイスの切り分けと、エリアごとの増減を演算、業務システムからえた外来患者推定位置との比較を行っている。（図4）さらに、業務端末のリアルタイム解析による職員位置情報をリファレンスに、モバイルデバイスの mac アドレスを推定、さらには推定情報をもとに、電子カルテ位置マップの自動拡充を試している。将来的にはオプトイン形式で職員位置情報を業務支援に活用できるだろう。

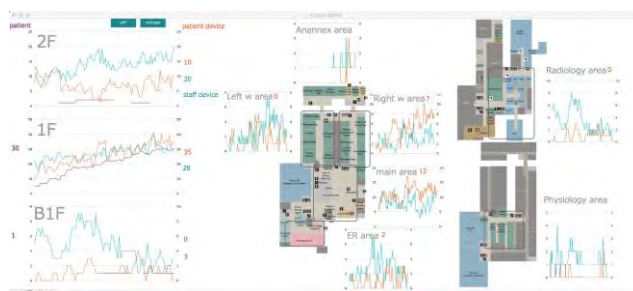


図4 エリア別患者分布数推定

電子カルテをはじめとする業務パッケージソフトは、安定性やシステム更新時のダウンタイムを短くする観点からも、極力カスタマイズを抑制して運用することが望ましい。一方で SOA を踏まえ、パッケージソフトにもサービス単位で機能呼び出しで利用できるような工夫を要求仕様として盛り込みつつ、俯瞰的かつ複雑な診療判断支援や経営判断支援は業務パッケージ外の ESB に実装し、パッケージ群をサービス単位で再構築しつつ、IoT 技術を代表とする最新技術の恩恵を逐次取り込むような、発展できる医療情報システムが、今後の医療を取り巻く環境変化を乗り越えるために必要だろう。

地域包括ケアシステムを支える ESB/SOA 基盤

院内連携基盤と地域連携基盤との融合に関する一提案

ESB/SOA Infrastructure Supported Integrated Community Care System A Challenge of Collaboration Platform Harmonization between Inside and Outside

下川 忠弘¹⁾

Tadahiro SHIMOGAWA¹⁾

公益社団法人京都保健会 京都民医連中央病院¹⁾

Public Interest Incorporated Association Kyoto Hokenkai, Kyoto Miniren Chuo Hospital¹⁾

キーワード: 地域包括ケアシステム,

1. 背景

団塊の世代が後期高齢者となる 2025 年に向けた医療・介護提供体制の見直しの一環として、地域包括ケアシステムの構築が求められている。そして、その構築に寄与するような、ICT を活用した医療機関間や医療機関と介護事業所との間の情報共有が行われるようになることが求められている。それら情報共有のために医療情報連携ネットワークを構築することとなるが、厚生労働省発出の文書及び保健医療情報分野の標準規格（厚生労働省標準規格）では、情報共有の仕組みや情報の取得方法、及び基本的な情報流通の標準規格に言及しているものの、全ての関係事業者が所有しているシステムがそれら標準規格に対応しているとは限らない。また、それらシステム間をどのように接続するかについての言及がない。そのため、現実的には全てのステークホルダーが当該ネットワークへシステムを接続し、情報流通を行うことのハードルは高いと思慮される。

2. 目的

本発表では、全てのステークホルダーが医療情報連携ネットワークへ接続するための手法として、院内連携基盤と地域連携基盤とを融合させる

ことを提案する。具体的には、個々のシステムを、サービスを提供する要素として定義し、それらのサービス間の情報流通を、院内・院外を区別することなく定義する（SOA/ESB 基盤の定義）。そして、それぞれの連携基盤に必要となる技術的要素を定義し、それら要素と標準規格等との関連性を明確にする。

そして、これらの定義を基にした、ある法人における地域包括ケアシステムに対応した院内及び連携基盤の実装イメージを提案する。

3. 期待される成果

本発表による定義は、保健医療分野における標準化技術を用いた実装イメージを明確化し、かつ複数の標準化技術をフレームワーク化することによるシステム化を推進することに寄与すると考えられる。

参考文献

1. 厚生労働省(2014)「健康・医療・介護分野における ICT 化の推進について」, <http://www.mhlw.go.jp/file/06-Seisakujouhou-12600000-Seisakutoukatsukan/0000042495.pdf> (2016 年 8 月 5 日アクセス)
2. 厚生労働省(2016)「医療分野の情報化の推進について」, http://www.mhlw.go.jp/stf/seisakunitsuite/bunya/kenkou_iryuu/johoka/ (2016 年 8 月 5 日アクセス)

ビジネスルールアプローチによる抗菌薬使用届出の 事前チェック機能の開発

Development of a preliminary checking function of
antibacterial medicine use notification by business rule approach

島川 龍載¹⁾, 今井 将洋²⁾

Tatsunori Shimakawa¹⁾, Masahiro Imai²⁾

広島赤十字・原爆病院¹⁾, 京セラ丸善システムインテグレーション株式会社²⁾

Hiroshima Red Cross Hospital & Atomic-bomb Survivors Hospital¹⁾

KYOCERA MARUZEN System Integration Co.,Ltd.²⁾

キーワード: BPM, BRA, SOA, ESB, 抗菌薬使用届出

1. 背景

医療と情報技術のそれぞれの進歩に伴い、医療機関における病院情報システムの発展が急速に進んできている。しかし、病院運用の要求事項に応えるために、システムの個別開発や部門システム導入などにより部分最適がなされてきたため、複雑なシステム構成となっている。システム全体を俯瞰した管理が十分にできないことで、医療サービスレベルを満たしているとは評価しづらい状況と言える。

病院経営の改善や医療の質及び安全性の向上に向けた中長期的な戦略において、システムや機能により提供されるサービスを業務フロー

(Business Process Management : BPM) の視点からコンポーネント化し、データとプロセスを全体最適化した統合プラットフォーム (Service Oriented Architecture : SOA) を構築することで、病院情報システムのフローを可視化した健全な運用を実現することが期待できる。

SOA プラットフォームの利用において、戦略的に経営、行動、実務、手順の観点でロジックや誓約をルール化したビジネスルールアプローチ

(Business Rule Applies : BRA) の手法を取り入れることで、具体的なガイダンスによるシステム

アーキテクチャを実現し、プロセスの合理化と効率化を図ることが可能となる。

2. 目的

バンコマイシン等の抗 MRSA 薬及び広域抗菌薬等の使用における耐性菌治療に際して、耐性菌の出現を防止するための抗菌薬適正使用として、届出制または許可制を取り、投与量、投与期間などの把握を行い、臨床判断を行うことが求められている。

許可制とは、第三者に使用を許可してもらうことであり、届出制は医師自身が抗菌薬の使用を許可することと言える。広島赤十字・原爆病院では、患者の状態を把握している主治医が重篤度や緊急性などを判断することが自然であるという結論から、事前届出 100% を目標に抗菌薬使用届出書の作成有無にかかるチェックシステムの仕組みを考えることとなった。

処方や注射オーダと抗菌薬使用届出状況のチェックには、異なるシステム間の連携が必要となることから、SOA プラットフォームの ESB (Enterprise Service Bus) を活用し、BRA の 4 つの有用性を見据えたチェックシステムの開発を行うこととした。

① ルールの分離

ルールを他のシステム要件から明確に分離することでルールを再利用したり、他のシステム要件から独立して変更したりできる。

② ルールのトレース

ルールに対して、ミッション、コール、目的、戦略、戦術、ポリシーなどをトレースし、ルールの正当性の保証とルールを変更した場合の影響度をわかりやすくする。

③ ルールの外部化

ルールを誰にでもわかるフォーマットで公開し、ルールの見える化を行う。

④ ルールの配置

ルールを容易に変更できるような環境におくことで環境の変化に対応する。

対象の抗菌薬に対する抗菌薬使用届出書の作成有無のチェックをESBで外部化する。その上で処方及び注射オーダーの問い合わせをSOAPによるWebサービスとし、抗菌薬使用届出書の作成状況と対象抗菌薬のデータベースへの格納をREST (JSON) によるWebサービスとした。これにより、各システムからは、パラメータを所定のフォーマットで渡すだけでESBから返ってきた結果を受け取り、利用者が操作性の違和感なく、適切な処理を行うことができるように構築している。

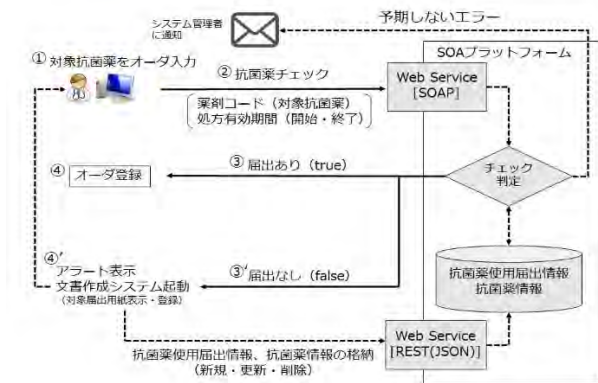


図1 抗菌薬使用届出チェックの概要

3. 成果

抗菌薬使用届出書の事前作成率が100%となり、抗菌薬使用届出書が薬剤部に速やかに届出されるようになった。また、対象抗菌薬の届出日とオーダーの実施予定日の有効期間チェックにより、同薬剤の2回目以降の継続オーダーに対しても適切にチェックされるようになった。これらのプロセスと結果は、SOAプラットフォームを介して、可視化されていることで、柔軟に環境変化に対応できるようになっている。

4. 考察

現実のシステム化の動機と実装との間には、多くのギャップがあり、それを埋めるためには意志と努力が必要となる。BRAでは、業務仕様をルールで表現することによって、このギャップを軽減し、比較的容易に動機と実装との間のトレースができるようになる。

さらには、組織や医療者の思考や意思決定などの具体化や表出化がしやすく、情報ライフサイクルの可視化により、中長期的なビジョンを病院がコントロールして形成しやすいと言える。

今後は、パラメータの自動更新へのアプローチを行うことで、ルールの自動化によるディープラーニングなどの機械学習を実装し、経営改善や医療の質や安全性の向上に向けた予測と傾向の分析支援に対しての有用性を見出していきたい。

参考文献

1. 酒井義朗, 井上光鋭, 有馬千代子, 久保裕子, 霞田美恵子, 指定抗菌薬使用届出制度の導入効果, 感染管理 Vol.23no.1, 2008;
2. 飯田征昌, 電子カルテシステムにおける汎用的なルールベースチェック機構の構築とその応用, 第20回日本医療情報学会春季学術大会, 2016
3. ロナルド・G・ロス, 宋雅彦, 渡部洋子, アジャイル経営のためのビジネスルールマネジメント入門, 2013
4. Barbara von Halle, BusinessRulesApplied:Building Better Systems Using the Business Rules Approach, 2001

診療記録における IoT デバイス活用に向けた SOA 基盤・電子カルテシステムにおける連携手法とその実践

Approach and practice of the inter-system linkage by the SOA infrastructure and electronic medical records system for applying the IoT devices to medical records

飯田 征昌

Masayoshi Iida

名古屋市立大学 医学部管理部病院医事課病院情報システム係

Division of Hospital Affairs, Nagoya City University Hospital

キーワード：電子カルテ，IoT，臨床判断支援，ルールベースシステム，XML

1. 背景

昨今、IT 業界のみならず一般的にも浸透しつつある用語かつ、その概念を用いた製品・サービスの展開が急速な勢いにて広がりを見せている IoT: Internet on Thing であるが、様々な活用事例において特に注目すべきは、製造業における革新的概念と取り組みであろう。

その代表格ともいべきドイツ連邦政府の提唱する Industrie 4.0 は、IoT を実現するデバイスによるセンシング・アクチュエーションを端点に、クラウドコンピューティング、ビッグデータ分析、人工知能といった最新の IT システムと FA: Factory Automation との統合によって、製造プロセスのみならずそのマネジメントの統合へと反映させるべく、アーキテクチャやプロトコルの標準化・オープン化施策を推進している。

スマートファクトリーの実現から、数々の企業で構成されたサプライチェーンにおける生産マネジメントの統合まで、スマート・エコシステムの実現とその効用によって、少子高齢化や国際競争の激化といった先進諸国の社会環境の変化に追従すべく、商品付加価値の自在な創出と生産効率向上の両立を果たすことが可能な、産業の持続発展において効果的なモデリングを提示する先端事例と言える。

翻って、医療分野に着目した場合、高度化の一途

を辿る医療環境の目まぐるしい変化に対する迅速な対応や、リスク管理を中心に高度かつ集約的なマネジメントが求められ続ける一方で、投入可能なリソースの縮小傾向といった厳しい課題に持続して取り組まなければならない状況である。

よって、先述した製造業における革新的なアプローチは、今後の医療における IT が目指すべき方向性とも合一する概念であり、このモデルに範を取りながら、医療における IoT の実現をはじめとした、IT 統合アーキテクチャの体系的かつ合理的な構築と実践の取り組みは、今後の医療情報分野の発展においても重要な要素となるであろう。

2. 目的

医療分野における IoT 活用によって期待される基本的な効果として、多様なセンシングデバイスが測定したデータをリアルタイムにネットワークを介して伝達し、診療記録として電子カルテシステム（以下、EMR: Electronic Medical Records）に格納することでの業務効率化やデータ転記に係るリスクの排除などが挙げられる。更にその発展形として、測定データの種類・頻度・精度の向上と関連しながら、ビッグデータ分析や人工知能技術の組み合わせることでの、高度な臨床判断支援への応用なども現実味を帯びつつある状況である。

そこで、医療における IoT 活用に向けての必須条件として先ず考慮すべき観点として、センシングデバイスが測定したデータに対し、患者や測定者の属性の付与や連結を、極力簡易な方法にて可能であるかという点が挙げられる。

しかしながら、体温計、血圧計、体重計など、最も基本的かつ利用頻度の高いシンプルなセンシングデバイスである程に、短時間で様々な属性(患者)を計測・記録するケースが殆どであり、これら比較的安価な部類に入るデバイスに対して、患者属性の付与や TCP/IP 通信などの高度な機能を実装する手法を取ることは現状困難である。

更に、現状の医療現場に導入されている通信可能なセンシングデバイスの大半は、独自プロトコルやデータフォーマットを用い、専用システムや個別対応のアプリケーションを介さなければ、EMR との連携は不可能であることから、スケールメリット等を見出さない限り、非常に高コストとなり易い。

この様な状況下にあって、センシングデバイス連携実現のニーズは医療安全や業務効率化の観点から日を迫る毎に高まりつつあることから、IoT とは程遠い様々な制約を持ったセンシングデバイスを、極力合理的な手法にて IoT デバイスとして位置付けられるアプローチを見出し、本院における IT 統合基盤であるサービス指向アーキテクチャを実現するプラットフォームを用いたシステム間連携基盤(以下、SOA 連携基盤)との組み合わせによって、センシングデバイスと EMR の効率的な連携機能構築の実現を試みた。

3. 方法

診療記録における IoT デバイス活用に向けた検討課題として、以下の事項を設定する。

- ① デバイスの通信方式
 - ② 収集可能データの分類・モデリングと集積手法
 - ③ デバイスデータと患者属性の紐付け手法
 - ④ 各デバイス特性に対応した EMR 連携手法
- まず、①は IP 通信機能の有無によって区分すべ

き課題となり、Wi-Fi 等による IP 通信機能を有しない場合、NFC、RFID、Bluetooth、ZigBee などの近距離無線通信や RS-232、USB などのシリアル通信となるであろう。よって、デバイスデータ受信と IP 通信変換を行うゲートウェイ装置の要否に至る。

②については、センシングデバイスで測定・出力可能なデータの特性や、属性データ(測定日時、デバイス識別子、患者情報など)がどのように付帯されるかに応じて、最終的なデータ格納先までに経由するプロセスおよびデータフォーマットなどのデザインに影響すると共に、③の実施可否にも及ぶことが想定される。

③は①、②のそれぞれの特性に影響される要素として考えられ、紐付け処理のサーバー側もしくはゲートウェイ装置側での実施選択や、紐付け可能な属性をデバイス側が持たない場合において、ユーザーインターフェースを有するアプリケーションの利用に至ることなどが考えられる。

④については、最終的なデータ格納先である EMR におけるデータ取り扱いや、これに至る格納プロセスの定義が対象となることから、接続デバイス毎の個別対応範囲を極力減らし、効率的な EMR 連携プロセスの実現こそが、診療記録における IoT デバイス活用の重要なポイントとなる。よって、既存の SOA 連携基盤上の様々な EMR 連携プロセスの流用による構築を前提とすることが、最も合理的なアプローチと成り得る。

以上の課題設定に基づき、IoT デバイスと EMR の効率的な連携の実現に向け、①～③に基づく接続デバイスの特性に応じた手法の定義および、SOA 連携基盤における④の方法について以下に示す。

(1) 通信可能なセンシングデバイスの特性に応じた IoT 化アプローチ

ア. デバイス直接接続連携方式

データに患者属性の付与が可能なデバイスかつ、IP 通信が可能である場合に、SOA 連携基盤のサービスに直接接続し EMR 連携を行う方式とする。た

だし、SOA 連携基盤にて実装可能なプロトコルに限られることから、汎用的な通信プロトコルを実装したデバイスまたはゲートウェイ装置でなければ実現は困難である。

イ. デバイス直接接続・患者紐付け管理データ連携方式

データに患者属性を持たないが、デバイス識別子やタイムスタンプを有し、IP 通信が可能であるデバイスを対象に、患者とデバイス識別子との関係を経時的に示すデータを外部データソースなどに有することを前提条件とする。よって、デバイスデータ送信時の SOA 連携基盤上の処理において、患者紐付けが可能となることでの EMR 連携方式とする。

また、本方式の発展形として、NFC、RFID、AeroScout、Beacon 等を用いて、時系列の患者ロケーションやデバイスロケーションと、デバイス識別子との紐付け方式も考えられる。

ウ. デバイス接続アプリケーション連携方式

データに患者属性を持たず IP 通信不可能なデバイスを対象に、アプリケーションとの通信によるデータ連携を前提とする。アプリケーションでの患者選択操作によって、取得データの患者紐付けが可能となり、アプリケーションから SOA 連携基盤のサービスに接続し連携を行う方式とする。

エ. デバイス直接接続・アプリケーション連携併用方式

データに患者属性を持たないが、デバイス識別子やタイムスタンプを有し、IP 通信が可能であるデバイスのデータを先にリポジトリに格納することを前提条件とする。その後、アプリケーションはリポジトリに格納されたデバイスデータの検索(デバイス識別子やタイムスタンプの条件指定)と選択によってデバイスデータを取得し、以降はウと同様にアプリケーション経由の EMR 連携を行う方式とする。

(2) SOA 連携基盤での EMR 連携アプローチ

本院の SOA 連携基盤の基本機能について、以下に示す。

- ① 外部システムより様々な方式 (TCP/IP, SOAP, HL7, File, SQL, POP3 等) にて受信・取得したデータをメッセージとして格納
 - ② メッセージを EMR データとして生成・更新が可能な形式に変換 (必要に応じて EMR をはじめとする外部データアクセスによる不足要素の補完など)
 - ③ 変換後のメッセージを XML 出力し、SOA 連携基盤上の EMR 更新モジュール (VB.NET) に送信することで、EMR データの生成・更新処理を直接実施
 - ④ 必要に応じて、処理結果・要求データを外部システムに対して返信
 - ⑤ 上記、①、②、④の組み合わせにより、外部から EMR 等のデータ参照サービスの構成も可能
- 以上の構成・方法により、メッセージ単位でトランザクショナルに EMR データの参照・生成・更新が可能な連携基盤となっており、カルテ記事、指示・オーダー、検査結果、熱型表、患者プロフィール、メール、ファイリング画像など、EMR の主要データの大半を網羅する連携を既存機能として実現している。

よって、既設の EMR 連携プロセスの流用・活用を前提とした構築手法が可能な環境であることから、(1)ア～エに示した IoT 化アプローチモデルのいずれにおいても、基本的には、デバイス側が接続可能なプロトコルその後の連携プロセスに対応したデータ変換・メッセージ格納を行うサービスアダプター (①に該当) の対応のみにて EMR 連携が可能となる。

4. 結果

- (1) NFC 通信機能付き血糖測定器・体温計への適用
本院で既に導入済みであった血糖測定器や体温

計（テルモ社製）において、NFC 通信機能の搭載が判明したことを契機として、スライディングスケールに基づいたインスリン投与量確定の際に、正確な血糖値の入力が必要となることから、誤入力リスクの軽減策として連携機能の実装を行った。

既存の電子カルテ Web アプリケーション（以下、Web EMR）を利用するスマートフォン端末（Android 4.2）において、NFC リーダーが搭載されていることから、該当デバイスの NFC ヘルスケアライブラリ（http://www.sony.co.jp/Products/felica/business/products/NFC_healthcare.html）を組み込んだ Android アプリケーションを新規に構築し、Web EMR の熱型表入力画面機能に NFC 連携起動ボタンと測定値のインターフェースを設け、URL スキームにて両アプリケーションの連携を行う、デバイス接続アプリケーション連携方式での対応とした。

よって、EMR 連携プロセスには一切手を加えない形にてデバイス連携を実現する簡易なモデルとなった一方、測定値が手入力・デバイス入力のいずれで行われたかの識別不能な点が課題として浮上したことから、EMR 記録内容にデバイス連携を考慮した項目追加の実装を、Web EMR および SOA 連携基盤において行う予定である。

(2) 無線通信機能付き身長・体重計への適用

身長・体重の誤入力データに伴う、抗癌剤投与量の誤計算が問題となったことを契機に、データ通信機能を有する身長・体重計の選定を行い、ZigBee を用いた専用規格とこれに対応した通信モジュールおよびインターフェースアプリケーション

（Windows のみ）を組み合わせた形式にて通信機能を提供する製品（seca 社製）の採用を照準に定め実装を行った。

先述の NFC 通信デバイスと異なり、据付型のデバイスを不特定の患者・測定者が利用するシーンにおける操作効率を重視し、1 台の安価な Windows タブレットを身長・体重計とセットする方式にて、これを擬似的に患者属性の付与と IP 通信が可能な

デバイスとして見立てる形によるデバイス直接接続連携方式を採用した。

その方法として、デバイス側が提供するインターフェースアプリケーションと IP 変換・患者属性付与を行う独自アプリケーション（VB.NET）間を、仮想 COM ポートを用いた RS-232 シリアル伝送方式にて結ぶことで、ZegBee 経由で受信したデバイス測定データのリアルタイム反映を実現している。また、バーコードリーダー等からの患者 ID・利用者 ID の読込時に、SOA 連携基盤に患者・利用者情報の照会を行い、入力した ID のチェックを実施した上で、患者・測定者の紐付けを行い SOA 連携基盤に伝送データを行う。

SOA 連携基盤側の対応については、患者・利用者情報の照会および測定データの更新処理が可能な SOAP 形式の Web API サービスアダプターを新設し、シンプルかつ汎用的なレイアウトの XML メッセージ交換を採用している。

その後の連携プロセスにおいては、生体モニタリングシステムの熱型表連携において構築した、SOA 連携基盤上の計測データジャーナルを介しての非同期連携プロセスを流用した。デバイス出力の身長・体重データは、生体モニターのバイタルデータと同様に、項目毎の時系列データの一要素として扱い、排他制御の影響を回避した連携が可能となるメリットを採用し、デバイスの独立性を高めている。

5. 考察

診療記録における IoT デバイス活用に向け、4 種類のアプローチモデルを今回定義したが、連携対象となるデバイスやシステムのデータ構造、運用場面の違いなど様々な要素を整理し、連携プロセスやパターンモデリングを検討しながら構築したことにより、既存デバイスの IoT 化アプローチと、EMR 連携実現への実装手法の明確化に繋がり、既存の Web EMR 機能や SOA 連携基盤上のプロセスそれぞれのメリットを活かした上での効率的な実装に繋がった。

また、今回の身長・体重計連携における実装を通じて、安価かつ小型化が進んだ PC と一般的なソフトウェア開発手法を組み合わせた手法は、SOA 連携基盤の効用によって非常にシンプルかつリーズナブルに構築できたことから、今後もシリアル通信が可能なデバイスを中心に、更なる応用・展開が期待できる。

また、この他にも無線 LAN 搭載 SD メモリカード (FlashAir) とこのアプリケーションを用いた実装も今後安価な IoT 実現手法として注目される。例えば、デジタルカメラのワイヤレス自動画像転送の実現と、今回試みることが出来なかったデバイス直接接続・アプリケーション連携併用方式を組み合わせる形で、セキュリティ・利便性の向上を試みることが可能である。また、ファイル出力可能なデバイスへの組み込みも安価な実現手法として期待できる。

今回の SOA 連携基盤の活用による IoT 化アプローチの取り組みを契機に、今後より様々な展開が期待される IoT デバイスの活用を視野に入れ、その実

装においても処理目的やデータ内容に依存しないサービスアダプターの汎用化や、ルールベースでのプロセス制御・管理の実現を試みるアプローチを取り入れるなどしてきた。

更なる課題としては、よりシンプルなプロトコルである REST や MQTT への対応などが考えられるが、今回の実装手法を足掛かりにすることで、導入デバイスの状況に応じながら迅速に対応することが可能な準備は整いつつあると言えるだろう。

今後も、その構築手法を洗練させながら、システムやデバイスの連携プロセスの集約と可視化を通じた、効率的な IT 統合の実践による HIS 全体のスマート化・エコシステム化の推進により、様々なサービスを低コストにて創出する持続的な発展基盤を目指していきたい。

参考文献

1. 杉崎真弘, Embedded Technology 2015/IoT Technology 2015 IPA セミナー【第7部】最新システム技術の海外動向～乱立する IoT 標準化の今を読み解く～, <http://www.ipa.go.jp/files/000049604.pdf>, (参照 2016-08-01)

一般口演セッション

指定席券自動発行システム

－審査員が存在する場合－

Reserved seat automatic publication system when plural judges exist

山本 和子

Kazuko Yamamoto

キーワード：指定席券，審査員，座席自動発行，入試

1. はじめに

昨今、話題になっている地域保健医療情報ネットワークシステムはどうあるべきか？ 新しいユニークなシステムは？ 等を、広く一般から意見や企画案を公募し、優秀な意見や企画案に対して試行的開発を求める場合、応募者が多くても、狭い会場で効率よく1日で審査できるように、指定席券自動発行システムを開発したので報告する。

2. 前提条件

指定席券自動発行システムを開発するに際して、下記の基本機能と追加機能を考慮している。

A. 基本機能

例えば映画館を1日借り切って、2時間の映画を1日に4回繰り返し(班と称す)放映することとし、申込者を公募する。映画館の座席は600席。2時間毎に同じ映画を4回放映するから計2400人の座席がある。内訳は座席の a.行(I)=30、b.列(J)=20、c.班(H)=4、合計座席(I x J x H)の二次元配列(図1)である。会場が扇形である場合は、事前に座席の無い部分を削除しておく。

公募の結果、N=2300人の応募があり、応募者の希望を加味した上で、N人の指定席券を自動的に発行する。これが基本機能である。

B. 追加機能

意見や企画案を公募する場合は審査員が存在す

る。複数の審査員に対して応募者は1名である。審査は座席順に行う。1日に審査できる応募者数 b.列Jは時間的に固定されるから、応募者数Nが決まれば審査員の a.組数Iが決まる。応募者の待ち時間を少なくするために c.班の概念が必要となる。会場は複数の地区に用意したい。会場からの距離が遠い者ほど早く帰れるようにしたい。応募者は過去の審査員と当たらないようにしたい等、基本機能 abcに加えて、これらの追加機能を用意しておく。即ち、d.応募者一覧(住所、氏名等)に過去の応募歴を加え、e.審査員の過去の審査組歴を照合しながら、f.審査会場と応募者の住所の距離が遠い者から順番に座席を振り分ける方法である(図2, 3)。

3. システムの概要

使用機種はノートPC(Windows10)、CacheのCSPで開発している。基本データベース(DB)は、

a. 応募者DB: 当年度応募の受付番号、氏名、住所(都道府県)に、過去の応募年度とその年度の審査組番号を付加。過去は10年間遡ること可能。

b. 審査員DB: 過去の年度別審査時組番号。

c. 都道府県DB: 審査会場からの距離。

の3種である。

上記に加えて、1日の審査件数。審査組数、過去の記録年数を設定すればシステムは稼働する。

映画鑑賞

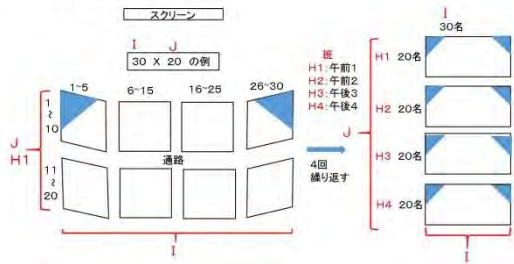


図1 基本機能

審査員が存在する 地区 1箇所の例

地区	地区 I		地区 II	
	11 X J11 の例	(各3名)	110 X J11 の例	(各3名)
H1	K1-1-1 K1-1-2 K1-1-3	K2-1-1 K2-1-2 K2-1-3	K3-1-1 K3-1-2 K3-1-3	K4-1-1 K4-1-2 K4-1-3
H2	K1-2-4 K1-2-5 K1-2-6	K2-2-4 K2-2-5 K2-2-6	K3-2-4 K3-2-5 K3-2-6	K4-2-4 K4-2-5 K4-2-6
H3	K1-3-7 K1-3-8 K1-3-9	K2-3-7 K2-3-8 K2-3-9	K3-3-7 K3-3-8 K3-3-9	K4-3-7 K4-3-8 K4-3-9
H4	K1-4-10 K1-4-11	K2-4-10 K2-4-11	K3-4-10 K3-4-11	K4-4-10 K4-4-11

図2 追加機能 (例1)

審査員が存在する 地区 2箇所の例

地区	地区 I		地区 II	
	12 X J11 の例	(各3名)	16 X J11 の例	(各3名)
H1	K1-1-1 K1-1-2 K1-1-3	K2-1-1 K2-1-2 K2-1-3	K3-1-1 K3-1-2 K3-1-3	K4-1-1 K4-1-2 K4-1-3
H2	K1-2-4 K1-2-5 K1-2-6	K2-2-4 K2-2-5 K2-2-6	K3-2-4 K3-2-5 K3-2-6	K4-2-4 K4-2-5 K4-2-6
H3	K1-3-7 K1-3-8 K1-3-9	K2-3-7 K2-3-8 K2-3-9	K3-3-7 K3-3-8 K3-3-9	K4-3-7 K4-3-8 K4-3-9
H4	K1-4-10 K1-4-11	K2-4-10 K2-4-11	K3-4-10 K3-4-11	K4-4-10 K4-4-11

図3 追加機能 (例2)

4. システムの検証：審査員が存在する場合

検証のためのテストデータは、総て架空である。2015年に公募した結果、応募者は36名。1日に審査できるのは12名。従って組は3組。審査員は1組3名として合計9名である。再応募者には過去5年間の審査記録がある。

①応募者36名は審査会場からの距離が遠い順に並べ変えて受付番号を附番する。過去2010, 2011, 2012, 2013, 2014年間の応募時の組番号を追加。再応募者の組別内訳は図4に示した通りで、全応募者36名中、再応募者24名(66.7%)で新応募者12名(33.3%)より多い。

②審査員9名を選出。審査員DBより過去5年間

の審査時の組番号を記載する(表1)。再応募者が過去の審査員と当たらないようにするため。

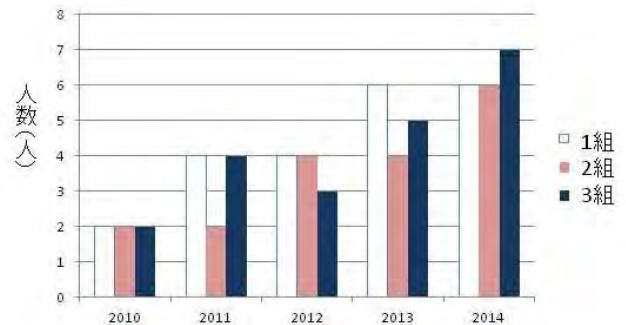


図4 組別過去5年間の応募者数

表1 審査員別審査年度別組番号一覧

審査員番号	審査年度					
	2010	2011	2012	2013	2014	2015
1	1	1	3	2	1	3
2	1	1	3	2	1	3
3	1	2	2	3	1	3
4	2	3	2	1	2	1
5	2	2	2	1	3	1
6	2	3	3	1	2	1
7	3	2	1	3	2	2
8	3	1	1	3	3	2
9	3	3	1	2	3	2

5. 成績

5-1. 実行

①審査員9名の履歴設定。

②再応募者について受付番号順に過去の応募時の組とその年の審査員の組を比較し、2015年度に審査不可の組を列記する。

③逆に審査可能の組を列記する。審査可能の組が全く存在しなかった12名に、警告1が発せられた(図5)。

④再応募者について審査可能の組列から1つの組を選定していく。最後に、選定された組別の人数が示され、どの組の審査員を入れ替えればよいか、警告2が発せられた(図6)。

以上により、2組の審査員を新審査員と入れ替え、①②③④で記録した応募者の組に関するデータを消去する。



図5 警告1の表示例

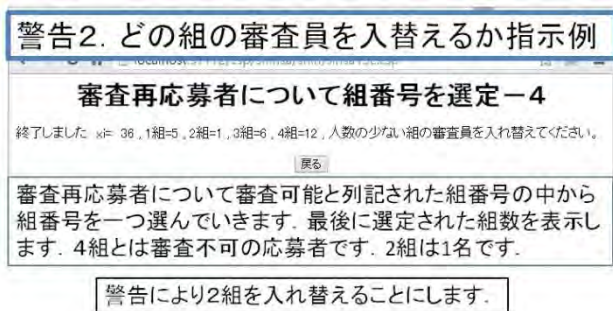


図6 警告2の表示例

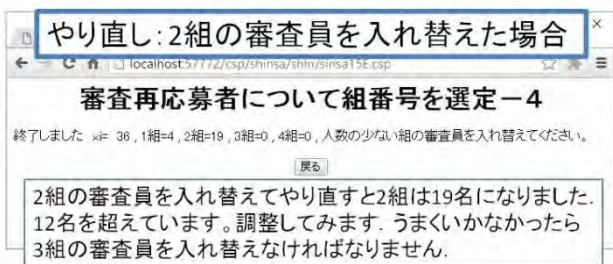


図7 審査員を入れ替えた場合

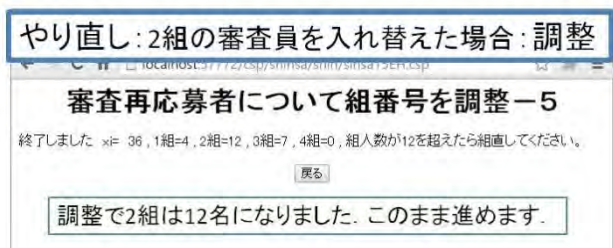


図8 2回目の調整

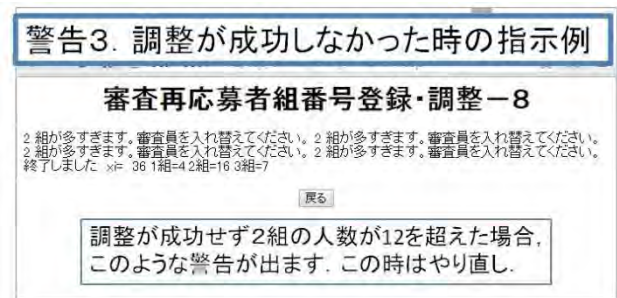


図9 警告3の表示例

5-2. やり直し

①②③④まで再実施する。警告1と2は出ない。分布は1組4, 2組19, 3組0となり、2組は12を超えている(図7)。

⑤2組のデータを他の組に振り分けできないか調整すると2組は12名に収まる(図8)。

⑥⑦で36名の指定座席(組・順・班)を用意。

⑧再応募者の組を確定。警告3は出ない。順調に組み分けされたと判断する。組み分けが不成功なら警告3が出る(図9)。

⑨で再調整。

⑩新応募者の組を振り分ける。

⑪応募者全員の組・順・班を決定。完成。

警告に従い、2組の審査員3名全員を入れ替えて組み分けし完成させた最終的座席表一覧を表2に示す。

表2 最終的な座席一覧表

1組			2組			3組					
組	順	班	受付番号	組	順	班	受付番号	組	順	班	受付番号
1	1	1	A03	2	1	1	A05	3	1	1	A01
1	2	1	A15	2	2	1	A06	3	2	1	A04
1	3	1	A16	2	3	1	A08	3	3	1	A07
1	4	2	A17	2	4	2	A11	3	4	2	A09
1	5	2	A02	2	5	2	A12	3	5	2	A10
1	6	2	A13	2	6	2	A22	3	6	2	A32
1	7	3	A14	2	7	3	A23	3	7	3	A35
1	8	3	A18	2	8	3	A24	3	8	3	A19
1	9	3	A20	2	9	3	A26	3	9	3	A21
1	10	4	A25	2	10	4	A27	3	10	4	A28
1	11	4	A29	2	11	4	A30	3	11	4	A33
1	12	4	A34	2	12	4	A31	3	12	4	A36

6. 考察

審査員が存在する場合の組み分けは、応募者が毎年増加している場合は、審査員も毎年追加されていくので比較的簡単に実施できるが、応募者が減少している場合や、応募者が少なくて組数が少なくなる場合は、応募者が過去の審査員に遭遇する確率は高くなり、組み分けは難しい。表3は3組で組内は審査員3名の例である。審査員が前年度と全く同じ組であった場合は、審査可の応募者は6名、審査不可は3名である。同じ組の審査員の中で1名のみ残し他を入れ替えた場合は、審査不可になる。

今回の検証で用いたデータは組数が少なく、表1に示したように、審査員は過去5年間同じ9名である。同時に応募者も過去5年間の再応募者が多く組み合わせ困難な例である。はたして組み合わせ可能かどうか疑問であった。警告2で2組の審査員を入れ替えるようにとの指示があった(図6)ので、組2の審査員を初めての審査員と入れ替えた結果、組み分けが成功し、表2の座席一覧を得ることができた。数分で実現可能である。

7. おわりに

当初は組み分けすることのみが目的で、順・班は手書きしていた。応募者にキャンセルがあると組・順・班を修正しなければならない。手書きではパンチミスの可能性があり不安である。

今回、発想を逆転させ、始めに応募者全員の座席を用意し、その座席に応募者を振り分けるという方法を考案した。組のみでなく組・順・班が一気に確定でき、警告に従い、わずか数分で何回でも繰り返しができるのは便利である。今後は組の入れ替えを警告するだけでなく、審査員の誰を入れ替えれば良いか、細かく指示できるようにしたいと考えている。

表3 前年度と今年度の審査員組別審査不可数

前年度の組				
	1組	2組	3組	
	1,2,3	4,5,6	7,8,9	←審査員番号
今年度の組				
組例	1組	2組	3組	審査可の数
1例	1,4,7	2,5,8	3,6,9	←審査員番号
	1x	1x	1x	←応募者組
	2x	2x	2x	審査不可
	3x	3x	3x	○の数=0
2例	1,5,9	4,8,3	7,2,6	←審査員番号
	1x	1x	1x	←応募者組
	2x	2x	2x	審査不可
	3x	3x	3x	○の数=0
3例	1,4,7	3,5,8	2,6,9	←審査員番号
	1x	1x	1x	←応募者組
	2x	2x	2x	審査不可
	3x	3x	3x	○の数=0
4例	1,2,8	3,5,7	4,6,9	←審査員番号
	1x	1x	1o	←応募者組
	2o	2x	2x	
	3x	3x	3x	○の数=2

表3 続き

5例	1,2,6	3,4,5	7,8,9	←審査員番号
	1x	1x	1o	←応募者組
	2x	2x	2o	
	3o	3o	3x	○の数=4
6例	1,2,9	4,5,6	3,7,8	←審査員番号
	1x	1o	1x	←応募者組
	2o	2x	2o	
	3x	3o	3x	○の数=4
7例	1,2,⑩	4,5,6	3,7,8	注1)審査員 1名入替
	1x	1o	1x	
	2o	2x	2o	
	3●	3o	3x	○の数=5
8例	1,2,⑩	4,5,6	7,8,⑪	注2)審査員 2名入替
	1x	1o	1●	
	2o	2x	2o	
	3●	3o	3x	○の数=6
9例	1,2,3	4,5,6	7,8,9	←審査員番号
	1x	1o	1o	←応募者組
	2o	2x	2o	
	3o	3o	3x	○の数=6

注1)審査員番号3を10に入替した場合

注2)審査員番号3と9を10と11に入替した場合

M 言語のデータベースにおける 数値解析の方法の利点

— 「関数の数値解析」 —

Advantages of Numerical Analysis with M Language Database

--- Numerical Analysis of the Gamma Function ---

高橋 亘

Wataru Takahasi

基礎コミュニケーション科学研究所

Research Institute of Fundamental Communication Science

キーワード: M 言語, 数値解析, 「関数」, 2 倍精度, 有効数字 17 桁

1. はじめに

M 言語はデータベース運用のための言語であり, 通常, 数値解析には適していないのではないかと考えられている. そのような判断が下される要因に, M 言語では通常の文字列と数値が同じタイプの変数で取り扱われるため, 数値のフォーマットが通常のプログラミング言語とかなり違っていることがあげられる. そのため M 言語によって物理学・工学の数値計算がなされることは非常に珍しいと言える.

しかしながら, M 言語は数値の取り扱いについて, 他の言語にない強みを持っている. それは, 有効数字の桁数である. 通常 M 言語では数値の有効数字 19 桁と考えてよい. 円周率は,

`$ZPI=3.141592653589793238`

オイラー数は,

`$ZEXP(1)=2.718281828459045235`

通常の数値計算では,

`1/3=0.33333333333333333333`

のように計算されるからである.

有効数字 19 桁は通常の言語の `long double` に相当する精度である. 有効数字が 19 桁あることは数値計算の結果が, 数値解析法が正当な方法で

あれば, 容易に 2 倍精度が確保されることを意味する. (多くの数値計算では, 計算法による誤差が 2~3 桁程度は避けられない)

以上のような点に着目して, 我々は M 言語による高精度の数値解析に挑戦することを試みた. 当発表では, 例として Γ 関数と Γ 関数を含む統計式の数値解析を例に挙げることにしたい. Γ 関数が流体力学などの基礎的な関係式によく現れるからである.

2. Γ 関数の性質と数値解析の方向性

最初に Γ 関数の定義と基本的な性質をまとめると, 次のようになる

Γ 関数は Laplace 積分の一例として定義される.^{1,2)}

$$\Gamma(x) = \int_0^{\infty} e^{-u} u^{x-1} du. \quad (2.1)$$

定義式で考えられている変数は通常 $x > 1$ の実数, もしくは式 (2.1) によって複素平面内に解析接続される領域の複素数が考えられる. 変数を正の整数に限定すると階乗計算に同値である.

$$\Gamma(n+1) = n! \quad (2.2)$$

また、定義式 (2.1) を部分積分して得られる式

$$\Gamma(x+1) = x\Gamma(x) \quad (2.3)$$

$$\Gamma(x) = e^{-x} x^x \sqrt{\frac{2\pi}{x}} \left(1 + \frac{1}{12x} + \frac{1}{288x^2} - \frac{139}{51840x^3} - \frac{571}{2488320x^4} + \frac{819395}{1045094400x^5} + \frac{5246819}{75246796800x^6} + \dots \right) \quad (2.4)$$

のように漸近展開される。³⁾ このような漸近展開は、通常、Stirling の漸近展開と呼ばれている。この展開は変数 x が十分大きな実数に対して成り立つのであるから Γ 関数の引数を十分大きなところに持って行く必要がある。関数の引数を十分大きなところに持って行く方法として関係式 (2.3) を活用することが出来る。式 (2.3) を逐次代入的に適用すると次のような式が得られる。

$$\Gamma(x) = \frac{\Gamma(x+n)}{x(x+1)(x+2)\cdots(x+n-1)} \quad (2.5)$$

$$\log(\Gamma(x)) = \log(\Gamma(x+n)) - \sum_{s=0}^{n-1} \log(x+s) \quad (2.6)$$

右辺の初項が漸近展開の対象である。対数を取ることが問題になるので $\log(\Gamma(x))$ の漸近展開を求

$$\begin{aligned} \log(\Gamma(x)) &= \log \sqrt{2\pi} - x + \left(x - \frac{1}{2}\right) \log x + \sum_{j=1}^{\infty} \frac{B_{2j}}{(2j-1)(2j)x^{2j-1}} \\ &= \log \sqrt{2\pi} - x + \left(x - \frac{1}{2}\right) \log x \\ &\quad + \frac{1}{12x} - \frac{1}{360x^3} + \frac{1}{1260x^5} - \frac{1}{1680x^7} \\ &\quad + \frac{1}{1188x^9} - \frac{691}{360360x^{11}} + \frac{1}{156x^{13}} + O\left(\frac{1}{x^{15}}\right) \end{aligned} \quad (2.7)$$

ここでベルヌーイ数 B_i は次のような関係式で規定される $B_0 = 1$ を初項とする数列 B_0, B_1, B_2, \dots である。

$$\sum_{k=1}^{n-1} \binom{n}{k} B_k \quad (n = 2, 3, \dots) \quad (2.8)$$

ベルヌーイ数は $i \geq 2$ で偶数項のみがゼロでな

は定義域の解析拡大の上で重要な式である。この式によると実数軸上の定義域が $x > 0$ の実数に解析拡大されることがよくわかる。

数値解析上重要な要素は Γ 関数の漸近展開である。変数 x が十分大きな実数に対して Γ 関数は

式 (2.5) が $\Gamma(x)$ を $\Gamma(x+n)$ で定義していると考えれば、十分大きな n に対して $\Gamma(x+n)$ を漸近展開することによって、精度の高い $\Gamma(x)$ の定義が得られることになる。

ここで問題になるのは十分大きな n に対して式 (2.5) の右辺の分母が非常に大きくなるという点である。こうした極端に大きくなる要素による割り算は精度を損なう原因になる。このような場合極端に大きくなる要素を抑制する効果のある方法は両辺の対数を取ることである。

(2.5) 式の両辺の対数をとると次のようになる。

めてみると、数学的に面白い結果が出る。結果はベルヌーイ数 B_i を用いて完全に解ける。³⁾

いので式 (2.7) では有効な項のみを記述するように表現されている。

こうして数値計算的に (2.6) の $\log(\Gamma(x+n))$ を与える式は二種類あることになる。(2.4) を活用するものと (2.7) を活用するものである。順にリストすると、

(2.4) を用いた展開は

$$\begin{aligned} \log(\Gamma(x+n)) &= \log(\sqrt{2\pi}) - (x+n) + (x+n - \frac{1}{2})\log(x+n) \\ &+ \log\left(1 + \frac{1}{12(x+n)} + \frac{1}{288(x+n)^2} - \frac{139}{51840(x+n)^3} - \frac{571}{2488320(x+n)^4} \right. \\ &\left. + \frac{819395}{1045094400(x+n)^5} + \frac{5246819}{75246796800(x+n)^6} + \dots\right) \end{aligned} \quad (2.9)$$

(2.7) を用いた展開は

$$\begin{aligned} \log(\Gamma(x+n)) &= \log(\sqrt{2\pi}) - (x+n) + (x+n - \frac{1}{2})\log(x+n) \\ &+ \frac{1}{12(x+n)} - \frac{1}{360(x+n)^3} + \frac{1}{1260(x+n)^5} - \frac{1}{1680(x+n)^7} + \frac{1}{1188(x+n)^9} \\ &\dots \end{aligned} \quad (2.10)$$

のようになる。二者は数学的には同等であるが、数値計算的には微妙な差異があり、実際にプログラム化して有効な方を採用するしか判別の仕様はない。いずれにせよこの段階で $\log(\Gamma(x))$ の数値計算式は確定したことになる。

3. 数値計算のパラメータの確定

実際の数値計算上、精度の高い結果を出すためには二つの問題がある。一つは数値計算に用いられる変数の精度、もう一つは前節の (2.6) 式の n をいくらに設定するかという漸近展開の臨界値の問題である。

数値計算に用いられる変数の精度が 2 倍精度 (有効数字 15 桁) である場合 Γ 関数の漸近展開は (2.9) を用いた場合に x^{-4} のオーダー、(2.10) を用いた場合に x^{-5} のオーダーしか有効でない。この点は、M 言語の場合、文字列変数が有効数字 19 桁の浮動小数点型の変数のように働くので (2.9) を用いた場合に x^{-6} のオーダー、(2.10) を用いた場合に x^{-7} のオーダーまでが有効になる。

漸近展開の臨界値 n をいくらにするのかという問題はプログラムを実際に組んで検証するしか方法はないのであるが、判定に使うのは Γ 関数の

性質の一つで $\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$ という理論値である。

$\log(\Gamma(x))$ が $x = \frac{1}{2}$ で

$$\log\left(\Gamma\left(\frac{1}{2}\right)\right) = 0.5723649429247000867$$

となることを目指せばよい。

Γ 関数の漸近展開 (2.9) で x^{-6} のオーダーまでを組み込んだプログラムでは $n = 100$ のときに最も精度が保証され、関数値として $x = \frac{1}{2}$ で

$$0.5723649429247000882$$

が得られた。

Γ 関数の漸近展開 (2.10) で x^{-7} のオーダーまでを組み込んだプログラムでは $n = 90$ のときに最も精度が保証され、関数値として $x = \frac{1}{2}$ で

$$0.5723649429247000852$$

が得られた。

結果として漸近展開 (2.9) と漸近展開 (2.10) に有意な差は無く、いずれの場合でも有効数字 17 桁が確保されたことになる。

以上により関数 $\log(\Gamma(x))$ の数値計算法は確定されることになるが、有効桁数を考慮して計算結果がより適切に表示されるためには我々はフォーマット関数をプログラムする必要がある。M 言語

で標準的に用意されているフォーマット関数は \$Justify のみであり、この関数は第 3 引数により小数点以下の桁数を指定できるが実数としての有効桁数を決めるものではない。我々は整数部分と小数部分の各桁数の合計を固定して有効桁数を確定するフォーマット関数 NFormat(X,Pat) を独自の構成した。この関数では、第 2 引数 Pat で “E” に引き続く数値を与えると第 1 引数 X の実数としての有効数字が返され、Pat で “F” に引き続く数値を与えると X の小数部分の桁数が限定された値が返されるように設計した。

フォーマット関数 NFormat を組み込んで構成された $\log(\Gamma(x))$ の M 言語版を LogGamma(X) と命名し、その指数関数を取ったもの、つまり

$$\$ZEXP(\$ \$^{\text{LogGamma(X)}}$$

を一つの関数にしたものを Gamma(X) と命名した。(Appendix 参照) $\$ \$^{\Gamma(1/2)}$ の値が

$$1.77245385090551603$$

\$ZPOWER(\$ZPI,1/2) の値が

$$1.772453850905516027$$

Appendix LogGamma(X) のルーチン

```
LogGamma(X) ;
N (X)
; Programmed by W. Takahasi; AM 11:14, Wed, 23 Mar, 2016
; 用法: S Y=$ $^LogGamma(X)
S N=$J(100-X,2,0)
; NO=100 で .5723649429247000882 For X=0.5 ; 参考値 .5723649429247000867 For X=0.5
S Y=$ZLN(2*$ZPI)/2+((X+N-0.5)*$ZLN(X+N))-(X+N)
For K=0:1:N-1 {
  S Y=Y-$ZLN(X+K)
}
S D1=((1/12)/(X+N))
S D2=((1/288)/((X+N)*(X+N)))
S D3=-((139/51840)/((X+N)*(X+N)*(X+N)))
S D4=-((571/2488320)/((X+N)*(X+N)*(X+N)*(X+N))),S4=1
S D5=((819395/1045094400)/((X+N)*(X+N)*(X+N)*(X+N)*(X+N))),S5=1
S D6=((5246819/75246796800)/((X+N)*(X+N)*(X+N)*(X+N)*(X+N)*(X+N))),S6=1
S D=$ZLN(1+((1/12)/(X+N))+((1/288)/((X+N)*(X+N))))+D3+(D4*S4)+(D5*S5)+(D6*S6)
S Y=Y+D
If Y>=1 {
  S Z=$ $^NFormat(Y,"E18")
} Else {
  S Z=$ $^NFormat(Y,"E23")
}
Q Z
;
```

となることから関数 Gamma(X) の $0 < X < 3$ 近傍の精度が分かり、Gamma(17) の値が

$$20922789888000 = 16!$$

となることから関数 Gamma(X) の整数値付近の精度が分かる。

4. まとめ

M 言語の数値計算に対する合理性を再確認し、 Γ 関数の高精度 (2 倍精度越え) の数値解析に成功した。関数 Gamma(X) は、この関数を含む多くの統計関数に活用することが可能である。M 言語による工学・統計学のための数値計算が期待される。

参考文献

1. 今村 勤, 『物理と関数論』(岩波全書), 岩波書店 (1981, 2 月, 東京).
2. 高橋 亘, “Laplace の方法と Γ 関数” 『基礎科学の方法』 No. 1, 基礎コミュニケーション科学研究所 (2016).
3. 高橋 亘, “ Γ 関数の漸近展開と数値計算” 『基礎科学の方法』 No. 2, 基礎コミュニケーション科学研究所 (2016).

Node.js による Cache アプリケーション

西山 強¹⁾, 土屋 喬義^{2,3)}

i K H コンサルティング¹⁾, 土屋小児病院²⁾, 獨協医科大学小児科学教室³⁾

1. はじめに

MSM で作成された NICU システムを、GT.M と NodeM を使って、Node.js ベースの Web アプリケーションである EWD.js に移行したところ、機能および性能には問題なかったが、実運用環境へ導入する際に困難があった。

そこでデータベースを GT.M から Cache に、Node.js インタフェースを NodeM から cache.node に切り替えて、運用環境の改善とさらなるレスポンス改善を実現した。

2. GT.M の運用環境

- ・サーバ：ノート PC
OS：Windows7
仮想化 OS：VMware Player
仮想マシン：ubuntu
DB：GT.M
ミドルウェア：EWD.js
- ・自動起動 コマンド作成
- ・バックアップ コマンド作成
仮想マシンバックアップ
GT.M バックアップ

現場に IT 専任者はいないので、平常時の運転開始とバックアップはコマンド作成で対応した。

しかし、トラブル時の対応に不安が残った。現場のユーザーは、Linux になじみがないので、仮想マシン以下の状況を的確に把握して伝達することが困難である。バックアップからの単純リカバリ操作であっても対応に不安がある。事実上、電

源入れ直しで解決できない場合は、現地対応が必要になりそうであった。

3. Cache の運用環境

- ・サーバ：ノート PC
OS：Windows10
DB：Cache
ミドルウェア：EWD.js
- ・自動起動 サービスとして登録
- ・バックアップ タスクスケジュール

Cache は Windows ネイティブで稼働するので、平常時の運転開始と定時バックアップはサービスとタスクスケジュールへの登録で対応した。

また、Windows スキルを持つ現場ユーザーが多いので、トラブル時の状況報告に不安は無く、単純なリカバリ操作などは、サポートデスク的な支援で対応可能と思われる。

4. Cache.node の設定

- ・ ewd.js の起動パラメータ設定 (ewdStart-cache-win.js)

```
var params = {
  //cwd: 'c:\%ewdjs',
  poolSize: 2,
  httpPort: 8080,
  traceLevel: 1,
  database: {
    type: 'cache',
    path: 'c:\%InterSystems%\Cache%\Mgr',
    username: "_SYSTEM",
    password: "SYS",
    namespace: "NICU",
    charset: "UTF-16"
  }
},
```

図 1 cache.node 起動設定

GT.M 接続から Cache 接続に切り替えるのは、ewd.js の起動パラメータを Cache のインストール環境に応じて設定するのみであった。EWD.js 提供のデフォルトは charset: "UTF-8" であるが、日本語対応には charset: "UTF-16" が必要である。

5. cache.node の性能

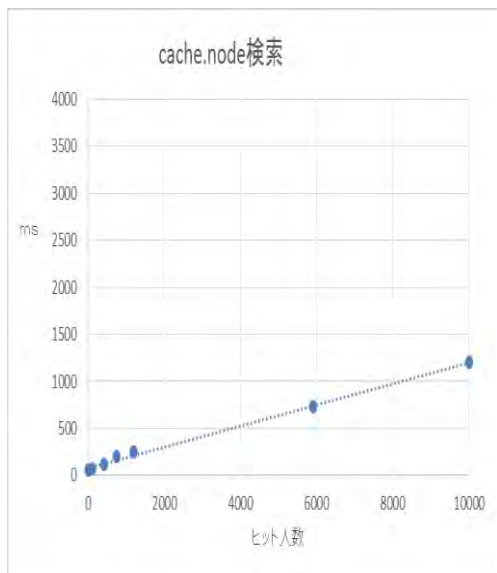


図2 cache.node レスポンス

cache.nodeを使った検索画面のレスポンスを計測したところ、単純な比較はできないが過去のNodeMによるレスポンスより1.5倍程度高速な結果となった。

6. まとめ

EWD.js と cache.node の組み合わせによる Node.js ベースの Web アプリケーションによって、現実的な NICU の運用環境を容易に実現できた。

また、cache.node 経由のデータアクセスは十分早く、レスポンスタイム改善の効果もあった。

参考文献

1. EWD.js リファレンスガイド(bild 0.67) 日本語版 日本 M テクノロジー学会誌

GT.M ルーチンを Cache'ルーチンに変換するツール

Translator tool of GT.M routines to Cache' routines

鈴木 利明¹⁾, 高見 美樹²⁾, 澤田 潔¹⁾, 嶋 芳成¹⁾

Toshiaki Suzuki¹⁾, Miki Takami²⁾, Kiyosi Sawada¹⁾, Yoshinari Shima¹⁾

日本ダイナシステム株式会社¹⁾, 園田学園女子大学人間健康学部人間看護学科²⁾

Japan DynaSystems Inc.¹⁾

Sonoda Women's University, Faculty of Human Health, Department of Human Nursing²⁾

キーワード: GT.M, Cache', ルーチン, コンパイラコンパイラ, 変換ツール

要旨: GT.M と Cache'は共に標準 M 言語仕様の実装である。標準 M 言語仕様および移送基準(現在の言語定義では省略されている)は、実装間でアプリケーションの移送(移植)が行えるように定めたものである。FIPS 用には検定も行われていた。GT.M は標準 M 言語仕様を基本で拡張構文は少ない。Cache'は標準 M 言語仕様を基本としてオブジェクト構文や組み込み SQL 構文などの拡張構文が多い。

標準 M 言語仕様は特に標準の命令と標準の関数は同じ動作をすることを実装者に要求している。しかし、標準 M 言語仕様には実装依存の構文を許しており、それを移行先の実装に適合させなければアプリケーションは正常動作をしない。しかし、手作業では書き換えに手間がかかり漏れが残ることがある。そこで、GT.M と Cache'の構文の差を調べて、GT.M ルーチンプログラムを Cache'ルーチンプログラムに変換するツールを作成したので報告する。変換ツールは、コンパイラコンパイラを使用し、GT.M ルーチンファイルを取り込んで、Cache'用の構文変換にしたルーチンを出力する方法をとった。コンパイラコンパイラのソース(in)として GT.M の BNF 記述を行い、オブジェクト構文(out)として Cache'構文記述をした。関数の適合にはユーザー定義関数(jdsGT.M)を作成した。

はじめに

M 言語の利用者はプログラムを長期にわたって使用することが多い。基幹アプリケーションソフトに使用しており 30 年以上利用しているケースもある。データベースには長期にわたる生データが保存されており、古いデータを捨てることは考えていないからである。

しかし、長期に渡って利用するとハードウェア、OS、実装を変えざる得ない場合が起きる。

M 言語の実装は標準 M 言語仕様および移送基準(現在の言語定義では省略されている)で定義されている。

これらの規格は特に標準の命令と関数は同じ動作をすることを実装者に実装者に実装者に要求している。米国 FIPS(Federal Information Processing Standard)用には検定も行われていた。このため標準命令と標準関数については、移行の為の変換は必要ない。

は文字コードだけである。しかし、標準 M 言語仕様には、装備依存の構文があり(装備者が自由に作成してよい構文(機能)、新機能を追加したり(bitmap 関数等) OS とのインタフェースをしたりしている。

それらのプログラムも移行先の環境に適合させなければ正常動作をしない。GT.M で動作しているプログラム(ルーチン)を Cache'環境に移行させる場合、手作業では書き換えに手間がかかり漏れが残ることがある。

そこで、GT.M と Cache'の構文の差を調べて、変換ツールを作成した。変換ツールは、コンパイラコンパイラを使用し、GT.M ルーチンファイルを取り込んで、Cache'用の構文変換にしたルーチンを出力する方法をとった。コンパイラコンパイラのソース(in)として GT.M の BNF 記述を行い、オブジェクト構文(out)として Cache'記述をした。

関数の適合にはユーザー定義関数(jdsGT.M)を作成した。

標準 M 言語仕様は、ハードウェア、OS 依存と考えられるの

構文調査

標準M言語のBNF(Backus Normal Form または Backus Naur Form)は、標準M言語仕様(1)に定義されている。

標準M言語仕様のBNF一部

See 2.2.2.1 for the definition of `lvn`. See 2.2.2.2 for the definition of `gvn`.

```
expritem ::=
  svn
  function
  exfunc
  exvar
  numlit
  strlit
  ( expr )
  unaryop expratom
```

GTM 構文のBNFを作成し、それに対応するCache'の構文を作成した。また必要があれば動作試験を行った。

変換の対応は下記の方針で行った。

1. 標準命令と標準関数は同じ実行動作をするので、変換が必要ないので等価変換(同じ構文)をする。
ただし命令語、関数語はすべて小文字に変換をした。
2. 命令語、関数語が変わったものは、Cache'用のものに変換する。
3. 変換が必要だが、Cache'に対応する構文がない場合は、Cache'側で等価機能を作成する。関数は等価機能になるユーザー関数を用意しそれに変換する。
4. 複数の命令等に変換が必要なものは、複数に変換する。

変換対応の概略

関数、bit 演算関数等

GTM、Cache'ともにbit演算を利用することができる。対応する関数はあるのだが、実装が少し異なっている。そのため、Cache'側ではユーザー定義関数を作成し対応することにした。

OPEN, USE, CLOSE 命令の引数

OPEN, USE, CLOSE 命令の引数は、実装依存である。多くの機能があるが、対応する機能があったので、それに変換するようにした。特別な対応を必要としたのが `exception` 指定 である。

GTM では、OPEN, USE, CLOSE 命令の引数に

`exception="trap 先"` の指定ができる。便利な構文なので利用頻度は高い。

Cache'にはこの構文はないので、

```
set $ztrap="trap 先"
```

OPEN 命令

のように二つの構文に分けることで対応した。

拡張SET命令

Set 命令に特別な機能が追加されている。

```
Set *fields=$$SPLIT^%MPIECE(@expr@(dataType,i),v)
```

この特別な構文は、`$$SPLIT^%MPIECE` の第一引数の値を、第二引数の値を区切り文字として、分割して `fields` の配列に分割して格納する構文である。

jdsGTM の対応する関数は次のものを作成した。

```
split(x,y,z)  ;s *z=$$SPLIT^%MPIECE(x,y) のシミュレーション
              ;s n=$$split^jdsGTM(x,y,.z)
              new j
              kill z for j=1:1:$l(x,y) s z(j)=$p(x,y,j)
              quit j
```

他のものが見つからなかったため `$$SPLIT^%MPIECE` だけを対象とした。似た構文はDTM(DataTreeMUMPS)にも存在した。

\$zbitfind 関数

`$zbitfind ($bitfind)` 関数の動作が GTM と Cache'は異なっている。

`$find` 関数は、発見した次の位置を返す。

GTM の `$zbitfind` 関数は、発見した次の位置を返す。

Cache'の `$bitfind` 関数は、発見した位置を返す。

各対応表を次に記載する。

対応表の中の `expr`, `intexpr`, `lvn`, `argumentlist` はBNFの記述である。例えば `expr` と記載があると、そこにはM言語の式全ての記述ができることを意味している。`intexpr` は、構文は `expr` であるが最終的に整数評価をした値を使用すること意味している。`lvn` は、ローカル変数定義全てを利用できる。

対応表

OPEN, USE, CLOSE 命令の引数/パラメータの対応

GTM	Cache'	解説
exception=expr	set \$etrap=" expr" を命令直前にトラップ設定をする	
NODELIMITER		無視
FIXED		無視
RECORDSIZE=intexpr	/FIXED:intexpr	
READONLY	/READ	
NOWRAP	/NOXY	
NEWVERSION	/NEW	
APPEND	/APPEND	
DELETE	/DELETE	
NOREADONLY	/NOREADONLY	
RENAME=expr	/RENAME=expr	
DELIMITER		無視

システム特殊変数の対応

GTM	Cache'	解説
\$zlevel	\$stack	
\$zl	\$stack	
\$zstatus	\$\$zstatus^jdsGTM(\$ZE)	
\$zsystem	\$\$zsystem^jdsGTM(\$g(%111zsystem))	
\$zsy	\$\$zsystem^jdsGTM(\$g(%111zsystem))	
\$zs	\$\$zstatus^jdsGTM(\$ZE)	
\$zeof	\$\$zeof^jdsGTM	
\$zdateform	\$\$zdateform^jdsGTM	
\$zchset	\$\$zchset^jdsGTM	日本語版なので UTF-8 を返す
\$zversion	\$\$zversion^jdsGTM	
\$zv	\$\$zversion^jdsGTM	

関数の対応

GTM	Cache'	解説
\$bitnot(expr)	\$\$zbitnot^jdsGTM(expr)	
\$bitand(expr,expr)	\$\$zbitand^jdsGTM(expr,expr)	
\$bitor(expr,expr)	\$\$zbitor^jdsGTM(expr,expr)	

\$bitxor(expr,expr)	\$\$zbitxor^jdsGTM(expr,expr)	
\$bitcount(expr)	\$bitcount(expr,1)	
\$bitfind(expr,tvexpr,intexpr)	\$\$zbitfind^jdsGTM(expr,tvexpr,intexpr)	返り値が異なるので GT.M に合わせている
\$bitfind(expr,tvexpr)	\$\$zbitfind^jdsGTM(expr,tvexpr)	返り値が異なるので GT.M に合わせている
\$bitget(expr,intexpr)	\$bit(expr,intexpr)	
\$bitlen(expr)	\$\$zbitlen^jdsGTM(expr)	
\$bitset(expr,intexpr,tvexpr)	\$\$zbitset^jdsGTM(expr,intexpr,tvexpr)	
\$bitstr(intexpr,tvexpr)	\$\$zbitstr^jdsGTM(intexpr,tvexpr)	
\$bitstr(intexpr)	\$\$zbitstr^jdsGTM(intexpr)	
\$date(expr,expr,expr,expr)	\$\$zdate^jdsGTM(expr,expr,expr,expr)	
\$date(expr,expr,expr)	\$\$zdate^jdsGTM(expr,expr,expr)	
\$date(expr,expr)	\$\$zdate^jdsGTM(expr,expr)	
\$date(expr)	\$\$zdate^jdsGTM(expr)	
\$search(expr)	\$zsearch(expr)	
\$parse(expr,expr,expr,expr,expr)	\$\$zparse^jdsGTM(expr,expr,expr,expr,expr)	第 3 引数、第 4 引数の意味がよく分からなかった たので作成していない
\$parse(expr,expr,expr,expr)	\$\$zparse^jdsGTM(expr,expr,expr,expr)	第 3 引数、第 4 引数の意味がよく分からなかった たので作成していない
\$parse(expr,expr,expr)	\$\$zparse^jdsGTM(expr,expr,expr)	
\$parse(expr,expr)	\$\$zparse^jdsGTM(expr,expr)	
\$parse(expr)	\$\$zparse^jdsGTM(expr)	
\$zlength(expr,expr)	\$\$zlength^jdsGTM(expr,expr)	
\$zlength(expr)	\$\$zlength^jdsGTM(expr)	
\$zascii(expr,intexpr)	\$\$zascii^jdsGTM(expr,intexpr)	
\$zascii(expr)	\$\$zascii^jdsGTM(expr)	
\$zchar(exprlist)	\$\$zchar^jdsGTM(exprlist)	
\$zextract(expr,intexpr,intexpr)	\$\$zextract^jdsGTM(expr,intexpr,intexpr)	
\$zextract(expr,intexpr)	\$\$zextract^jdsGTM(expr,intexpr)	
\$zextract(expr)	\$\$zextract^jdsGTM(expr)	
\$zfind(expr,expr,intexpr)	\$\$zfind^jdsGTM(expr,expr,intexpr)	
\$zfind(expr,expr)	\$\$zfind^jdsGTM(expr,expr)	

命令の対応

GT.M	Cache'	解説
zsystem argumentlist	set %111zsystem=\$zf(-1,argumentlist)	返り値を%111zsystem に返している
set *name=\$\$SPLIT^MPIECE(expr,lvn)	do split^jdsGTM(expr,lvn,name)	例: s *fields=\$\$SPLIT^MPIECE(@expr@(dataType,i),v)

変換ツール

Cache'で動作するコンパイラコンパイラを利用した。

GTM のBNF 構文記述を元に行っている。

通常のコンパイラは、ソースコードとして上位コンピュータ言語ファイルを入力して、オブジェクトコードとしてアセンブラまたはC言語等の下位レベルの言語に変換をする。下位レベルソースは、さらに変換をして実行できるものとなる。今回は、M言語からM言語なので、コンパイラというよりはコンバーターのような変換ツールソフトウェアとなった。

以前から作成してあったコンパイラコンパイラを利用した。手作業の変換では入れ子関数の変換はミスが発生しやすいケースであるが、コンパイラコンパイラを利用した変換ツールは、入れ子になった関数も変換対象とするので、変換の漏れがなくなる。

コンパイラコンパイラのソース一部

(自動的に返り値の先頭に "\$" 記号がつく作りになっているので、見た目\$がひとつ足りないところがある)

expritem 部分

```

expritem ::= strlit      : *1
          | numlit      : *1
          | "$$".exfunc  : "$$" *1
          | "$$".exvar   : "$$" *1
          | "$".function : "$" *1
          | "$".svn      : "$" *1
          | dtmzpiece    : "$g(^zp($j))"
          | (" ".expr " ") : "(" *1 ")"
          | unaryop.expratom : *1 *2
          | "-".expratcom : "$re(" *1 ")" ;

```

拡張システム変数部分

```

svmZ ::= ^L^zlevel^ : ^stack^
      | ^L^zl^       : ^stack^
      | ^L^zstatus^  : ^$zstatus^jdsGTM($ZB)^
      | ^L^zsystem^  : ^$zsystem^jdsGTM($g{%111zsystem})^
      | ^L^zsy^      : ^$zsystem^jdsGTM($g{%111zsystem})^
      | ^L^zs^       : ^$zstatus^jdsGTM($ZB)^
      | ^L^zeof^     : ^$zeof^jdsGTM^
      | ^L^zdateform^ : ^$zdateform^jdsGTM^
      | ^L^zchset^    : ^$zchset^jdsGTM^
      | ^L^za^       : ^za^

```

```

^L^zb^      : ^zb^
^L^zio^     : ^zio^
^L^zversion^ : ^$zversion^jdsGTM^
^L^zv^      : ^$zversion^jdsGTM^
;

```

拡張関数部分

```

functionZ ::= fwordbitnot (" expr ") : "$zbitnot^jdsGTM(" *2 ")"
           | fwordbitand (" expr ", " expr ") : "$zbitand^jdsGTM(" *2 ", " *3 ")"
           | fwordbitor (" expr ", " expr ") : "$zbitor^jdsGTM(" *2 ", " *3 ")"
           | fwordbitxor (" expr ", " expr ") : "$zbitxor^jdsGTM(" *2 ", " *3 ")"
           | fwordbitcount (" expr ") : "bitcount(" *2 ", 1)"
           | fwordbitfind (" expr ", " texpr ", " intexpr ") : "$zbitfind^jdsGTM(" *2 ", " *3 ", "
+4 ", "
           | fwordbitfind (" expr ", " texpr ") : "$zbitfind^jdsGTM(" *2 ", " *3 ")"
           | fwordbitget (" expr ", " intexpr ") : "bit(" *2 ", " *3 ")"
           | fwordbitlen (" expr ") : "$zbitlen^jdsGTM(" *2 ")"
           | fwordbitset (" expr ", " intexpr ", " texpr ") : "$zbitset^jdsGTM(" *2 ", " *3 ", " *4
)"
           | fwordbitstr (" intexpr ", " texpr ") : "$zbitstr^jdsGTM(" *2 ", " *3 ")"
           | fwordbitstr (" intexpr ") : "$zbitstr^jdsGTM(" *2 ")"
           | fworddate (" expr ", " expr ", " expr ", " expr ") : "$zdate^jdsGTM(" *2 ", " *3 ", "
+4 ", " *5 ")
           | fworddate (" expr ", " expr ", " expr ") : "$zdate^jdsGTM(" *2 ", " *3 ", " *4
)"
           | fworddate (" expr ", " expr ") : "$zdate^jdsGTM(" *2 ", " *3 ")"
           | fworddate (" expr ") : "$zdate^jdsGTM(" *2 ")"
           | fwordsearch (" expr ") : "zsearch(" *2 ")"
           | fwordparse (" expr ", " expr ", " expr ", " expr ", " expr ") : "$zparse^jdsGTM(" *2
", " *3 ", " *4 ", " *5 ", " *6 ")
           | fwordparse (" expr ", " expr ", " expr ", " expr ") : "$zparse^jdsGTM(" *2 ", " *3
", " *4 ", " *5 ")
           | fwordparse (" expr ", " expr ", " expr ") : "$zparse^jdsGTM(" *2 ", " *3 ", " *4
)"
           | fwordparse (" expr ", " expr ") : "$zparse^jdsGTM(" *2 ", " *3 ")"
           | fwordparse (" expr ") : "$zparse^jdsGTM(" *2 ")"
           | fwordzlength (" expr ", " expr ") : "$zlength^jdsGTM(" *2 ", " *3 ")"
           | fwordzlength (" expr ") : "$zlength^jdsGTM(" *2 ")"
           | fwordzascii (" expr ", " intexpr ") : "$zascii^jdsGTM(" *2 ", " *3 ")"
           | fwordzascii (" expr ") : "$zascii^jdsGTM(" *2 ")"
           | fwordzchar (" exprlist ") : "$zchar^jdsGTM(" *2 ")"
           | fwordzextract (" expr ", " intexpr ", " intexpr ") : "$zextract^jdsGTM(" *2 ", " *3 ", "
+4 ")
           | fwordzextract (" expr ", " intexpr ") : "$zextract^jdsGTM(" *2 ", " *3 ")"
           | fwordzextract (" expr ") : "$zextract^jdsGTM(" *2 ")"
           | fwordzfind (" expr ", " expr ", " intexpr ") : "$zfind^jdsGTM(" *2 ", " *3 ", " *4 ")
           | fwordzfind (" expr ", " expr ") : "$zfind^jdsGTM(" *2 ", " *3 ")"
;

```

コンパイラコンパイラは、上記のようなBNF定義とそれに対応する出力構文の指定をすると、それを実行するプログラムを生成してくれるツールである。(Linux等では有名なYacc, Lexに相当する)

そのため、BNFを元にした対応定義(これがプログラムに相当する)が作成できれば、ツールとして実行するプログラムは容易に生成することができる。GTM2Cacheには686行を要し、生成されたCache'ルーチンは6925行である。

変換ツールの使い方

このツールは、GTM ルーチンファイルを取り込んで、Cache'用のルーチンファイルを出力するバッチプログラムである。変換ツールの動作環境はCache'なので、Cache'環境でツール(GTM2Cache.rsa と CC.obj)の取り込みをスタジオ等です。ネームスペースは特に固定の指定はしていない。

GTM ルーチンの出力

ルーチン出力プログラム %RO を使用して、ルーチン群をひとつのファイルにまとめ、ファイル出力をする。

コード変換

GTM 処理系の動作環境はLinux なので、ファイルの暗黙的文字コードはShift-JIS ではない。GTM でルーチンを出力すると、文字コードの暗黙値はUTF8 である。Windows 版の Cache'のルーチン取り込みの暗黙文字コードは Shift-JIS である。

今回は外部ソフト nkf を使用して、ルーチンファイルの文字コードを UTF8 から Shift-JIS に変換をした。

DOS 窓で

```
>nkf -Ws --in-place *.ro
```

を使って UTF8 から SJIS 変換をおこなった。

(Cache'は取り込み文字コード UTF8 を指定すれば取り込めるが、コンパイラコンパイラツールが文字コード指定を用意していなかったため文字コード変換を必要とした)

変換ルーチンの実行例

137321 行の GTM ルーチンファイル(c:\temp\Ya.ro)を変換した例

(動作環境 : Intel i5 Core 2.6Ghz (MacBookPro VMware Fusion 上の Windows 7, Memory 6Gbyte)で実行結果)

```
GTM2CACHE>do ^G2C
```

```
GTMのルーチンファイルをパス付で指定してください:c:\temp\Ya.ro
```

```
変換後のファイルをパス付で指定してください:c:\temp\Ya2.rsa
```

```
Intilizing...
```

```
Now Compiling...
```

```
Pass 1/2
```

```
Pass 2/2
```

```
Output of Object code (Generated code)
```

```
Time: 73.53 seconds.
```

Done.

変換後のアプリケーションの実行

1. ユーザーは、別途 Cache'用のルーチンファイルを希望ネームスペースにルーチンを取り込む。
2. また、ランタイムルーチンとして、jdsGTM.rsa ファイルをルーチンとして希望スペーススペースに取り込む。

その他 グローバル変数の移行

動作環境を GTM から Cache'に移行するためには、ルーチンだけでなくデータベースであるグローバル変数の移行も必要である。グローバル変数ではルーチンのような変換は必要ない。

GTM 環境で ファイルに書き出しをしコード変換をした。

```
>do ^%GO
```

ルーチン同様に nkf を使用した。

最後に、 Cache'環境でグローバル変数の取り込みを行う。

結果とまとめ

変換ツールを使用すると、GTM のルーチンを Cache'ルーチンに変換できることがわかった。

作成した変換ソフトウエアは、公開して良いプログラムに変更してから澤田氏が管理しているサイト(GTM 研究会 (sites.google.com/site/gtmstudy/))で公開する。

考察

GTM ルーチンを Cache'ルーチンに変換する変換ツールは、実行環境の移行に有効である。コンパイラコンパイラ方は、入れ子になった構文の変換にも利用でき、有効である。

参考文献

1. 標準 M 言語仕様 ANSI/MDC X11.1-1995 Programming Languages - MUMPS
2. JIS X3011-1995 プログラミング言語 MUMPS
3. GTM 研究会(sites.google.com/site/gtmstudy/)の翻訳資料

GTM マニュアル v6.1

4. Cache'オンラインマニュアル

5. M 言語による XML パーサー 第 27 回 MTA 2000

医療情報技師認定取得後に資格継続更新を行わない技師の傾向を探る

失効者からみる資格活用対策

Explore the tendency of the engineer not to qualify continuation updated after medical information technician certification

Qualification leverage measures to see from Revocation

伊勢田 司¹⁾

Tsukasa Iseda¹⁾

一般社団法人日本医療情報学会¹⁾

Japan Association for Medical Informatics¹⁾

キーワード: 医療情報技師, 医療情報技師更新制度, 資格失効者

【背景】

医療情報技師とは「保健医療福祉専門職の一員として、医療の特質をふまえ、最適な情報処理技術にもとづき、医療情報を安全かつ有効に活用・提供することができる知識・技術および資質を有する者」（日本医療情報学会,2004）¹⁾と定義される。この医療情報技師を育成し、認定するために2003年から能力認定制度が一般社団法人日本医療情報学会により開始された。医療情報技師は年間およそ5,000名の受検者の中から3割程度の資格認定者を毎年輩出している。以下にその推移を示す。

医療情報技師能力認定制度では医療情報技師の資質能力が保持されるよう最新の知識技能をもって業務にあたることを目的とし、資格更新制度を2005年から設けた。資格更新制度は生涯学習をポイントとして加算する制度であり5年間の認定期間内に医療情報技師育成部会が認めた医療情報システムに係わる各種学会、勉強会等に参加し、その内容に応じて得点する。合計50ポイント取得することで、認定期間をさらに5年間延長するものである。

また、5年間の認定期間中に医療情報技師育成部会が主催する医療情報技師生涯研修セミナーを最低1回は受講することを必須条件としている。特に医療情報システムに係る業務に就いている場合は5年間で25ポイントとなり、更新ポイントの半分を取得できる。

さらに資格認定期間終了時まで更新手続きを行わない場合は医療情報技師の認定資格を失効し、認定証を利用することができなくなる。

2003年取得、2004年取得者は資格更新制度がない状態で医療情報技師認定試験に合格しており、この2年分については、あらかじめ追加ポイント付与の優遇措置が取られた。2005年取得以降は、試験

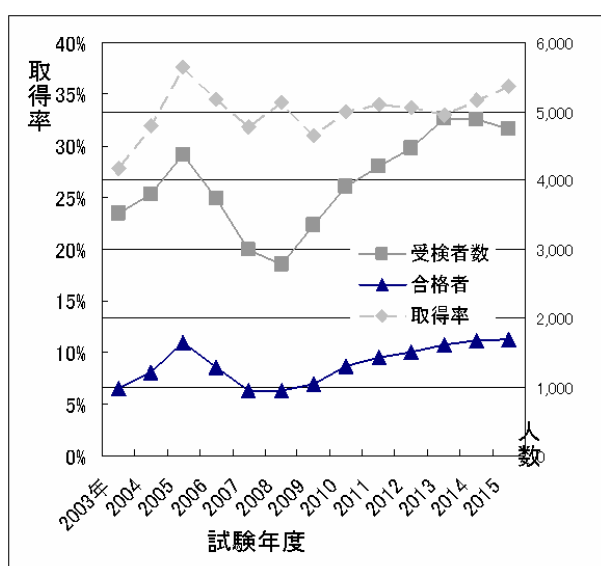


図 1 医療情報技師の受検者と合格者の推移

要綱提示とともに更新制度の提示がなされている。

周知という点では、加えて医療情報技師全員に資格取得の認定証とともに、文書による通達がなされている。他に、現在も更新該当年には、日本医療情報学会から直近に保持している更新ポイントが電子メールの形で通知されている。ただし、文書通知も電子メールも受検時のものである。学生メール、第三者によるとりまとめメール、メールアドレス変更などには対応しておらず、受検以降の医療情報技師の属性情報はあくまで、本人の申告を受けて行っている。更新制度そのものの理解と5年という期間の更新ポイント管理は医療情報技師の自己管理によるものが大きい。

このような更新制度がある一方で、医療情報技師自身の手による自治組織としての全国の医療情報技師育成部会が存在する。この技師会に所属する医療情報技師を対象に行った資格取得の理由を3項目選択するアンケート調査では²⁾、「自己研鑽のため」が7割超、「技術の客観的提示のため」が6割で自己の主観的項目が上位を占める。「仕事に必要」5割、「会社のすすめ」3割半ばはその後に続き、業務上必要で取得した人は多くはないといえる。

【目的】

本研究では、5年間の認定期間終了時に資格更新手続きを「行わなかった」人（以降、失効者という）の属性を調査し、その傾向を探ることとする。調査結果から、医療情報技師の資格を活用するためにはどのような対策が必要かを考察した。

【成果】

1. 調査対象

2003年から2010年に資格取得した医療情報技師のうち、認定失効者を対象とした。上級医療情報技師については対象としない。

表 1 医療情報技師失効者数

試験年度	1回目失効者(人)	2回目失効者(人)
2003年	403	208
2004	612	159
2005	839	170
2006	586	-----
2007	412	-----
2008	492	-----
2009	463	-----
2010	448	-----

2. 調査内容

医療情報技師資格取得時のデータを基に、各属性におけるデータを調査した。

医療情報技師2004年から2010年までの失効者の割合を勤務先、業務内容、年齢ごとに集計した。2003年の受検情報は勤務先および業務内容の分類が不適切で他年度との比較は行わなかった。

医療情報技師の合格者に対する認定失効割合は以下のとおりである。

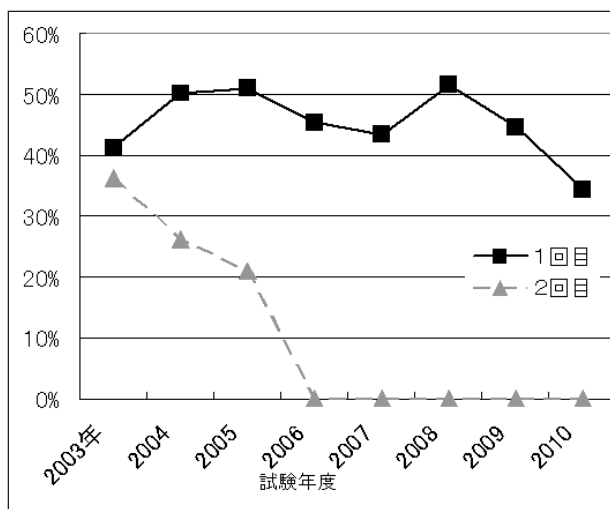


図 2 医療情報技師失効者の年度毎の割合

認定失効者は1回目の更新で2003年取得の失効者は4割であった。その後、5割を前後していたが、2010年取得では3割超まで減少した。また、2003年取得の2回目の認定失効者は前回更新から4割弱であったが、2005年失効者は前回更新の2割に減少した。2006年以降は該当しない。

3. 結果

勤務先別の1回目失効者の割合は以下のとおりであった。

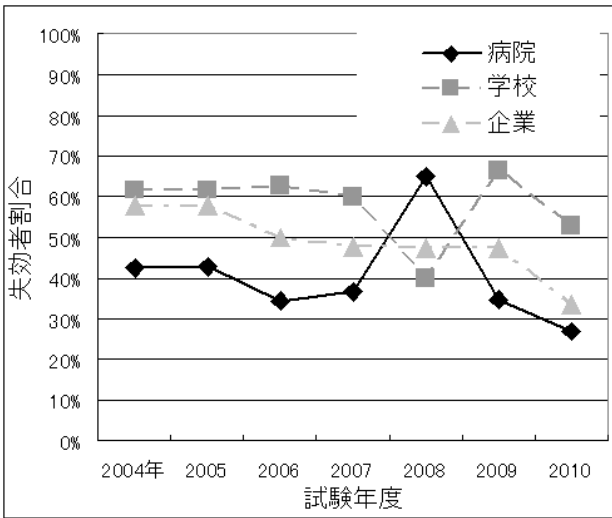


図3 医療情報技師失効者の勤務先別の割合

病院勤務の失効者は概ね4割前後であるが、2008年の失効者は7割弱まで増加し、その後4割から3割と減少した。学校勤務では、6割前後であったが、2008年では4割に減少し、その後7割から5割に変化した。企業勤務では、6割から5割であったが、その後緩やかに3割まで減少した。勤務先別の失効者の割合は学校と企業より病院のほうが少なく、全体的には減少した。

業務内容別の1回目失効者の割合は以下のとおりであった。

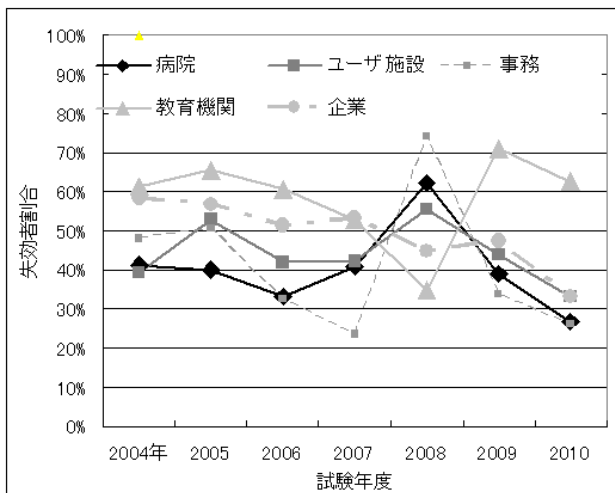


図4 医療情報技師失効者の業務別の割合

なお、図中の病院は病院関連業務（例：医師、診療放射線技師等）、ユーザ施設はユーザ施設の情報システム関連業務、事務は医療事務、教育機関は教員、学生等、企業はSE、営業等とした。

病院は当初4割から2008年に6割まで増加し、以降、3割弱まで減少した。ユーザ施設は、4割、5割であったものが3割弱まで減少した。事務は5割であったものが2007年に2割超に減少し、2008年には7割の半ばまで増加し、3割に戻った。教育機関は6割であったものが、2008年に3割超まで減少し、その後、7割まで増加し、6割に戻った。企業は6割であったものが緩やかに3割まで減少した。

業務別の失効者の割合は、教育、企業、事務、ユーザ施設、病院の順で少ない結果であった。

病院勤務の医療情報技師のうち大半を占める、診療放射線技師と臨床検査技師³⁾について1回目の失効者の割合を調べた結果、以下のとおりであった。

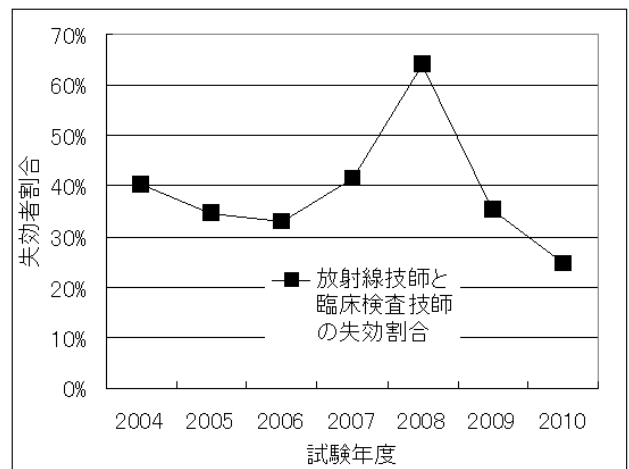


図5 医療情報技師失効者の放射線技師と臨床検査技師の割合

2004年取得では4割であった失効割合は2008年に6割半ばまで増加し、2010年取得では2割半ばまで減少した。

医療情報技師の1回目と2回目の失効者と更新有の別による認定期限時点での年齢は以下のとおりであった。

表 2 医療情報技師の更新時の年齢分布

更新1回目	失効者			更新有		
試験年	n	ave	sd	n	ave	sd
2003年	403	42.99	8.63	530	44.02	7.22
2004年	612	41.72	7.9	554	43.31	7.11
2005年	839	41.05	8.14	763	41.72	7.18
2006年	586	39.76	8	659	41	7.22
2007年	412	41.22	7.58	514	40.77	8.59
2008年	442	40.69	7.74	502	40	9.3
2009年	463	38.74	9.12	561	39.63	7.62
2010年	448	38.54	9.18	780	40.25	7.89
更新2回目	失効者			更新有		
試験年	n	ave	sd	n	ave	sd
2003年	208	50.83	8.15	309	47.98	6.51
2004年	159	50.62	8.58	395	47.53	6.18
2005年	171	48.94	8.16	576	46.10	6.79

失効者と更新有のうち、年齢を比較して若年を灰色で示した。2群の年齢の平均をt検定で分析した結果、1回目では、2004年、2006年、2010年、2回目では2003年、2004年、2005年で有意に差がみられた。

【考察】

2003年取得から2010年取得までの1回目失効者は減少したといえる。2回目の失効者も減少した。これは、ある程度更新制度の周知の効果がみられたといえる。しかしながら、2008年取得者の病院関係者、特に診療放射線技師と臨床検査技師の失効者は一時的に大幅に増加した。このことは2007年および2008年の医療情報技師試験の受検者減少となんらかの関係が疑われるが、診療報酬改定などに理由がみあたらず、継続して調査が必要である。

業務別失効者の割合のうち、教育機関では教師とともに学生を含む。このことにより、学生を多く含むことに連動して教育機関の失効者割合は多くなった。更新制度のうち、医療情報システム関連業務に就くと半分の更新ポイントを付与される。学生で医療情報技師の資格取得を果たしても医療情報関連の職に就かなかつた場合、更新は難しくなる。同様に、医療関連の職から離れた医療情報技師は業務

に連携して更新を行うことが難しい。

医療情報技師の資格取得目的が「自己研鑽のため」である場合には、継続して自己学習を進めることが可能であるが、「技術の客観的提示のため」であった場合、一時的な資格取得のみに留まる可能性がある。「仕事に必要」でない場合、自己の主観的事由で資格取得はしたものの、組織からの客観的評価を得られないと、認定継続に労を費やす可能性は減少する。

また、年齢分布をみると1回目の更新では主に若年者が失効する傾向にある。これは上記同様、学生や転職によるものが考えられる。若年者の職業的立場が大きく変化する中で5年間に方向性が異なり、失効した可能性が考えられる。この1回目の失効者は2007年から2009年にかけて年齢比較の結果、更新有のほうが若年となる現象が生じているが、この理由については、継続的に調査が必要である。

2回目の更新では反対に更新者の年齢が若年となるが、これは定年退職により離職した場合が考えられる。このことから、資格取得直後から更新1回目までの間に、医療情報システムに係る業務に就くこと、または興味を継続できること、日本医療情報学会から連絡がつくことが資格継続の大きなポイントとなると考えられる。

また、医療情報技師会には、同業職能集団として、資格取得直後のフォローアップをすることが資格更新および継続には重要であると考えられる。

参考文献

1. 医療情報技師とは," https://www.jami.jp/hcit/HCIT_SITES/job.php?job=info/annunce02.html (参照 2016-08-10)
2. 伊勢田, 相坂, 大原, 医療情報技師を対象とした心理的要因による内部不正行為防止対策, 医療情報学, 2015, pp1000-1001
3. 業務内容分布表," https://www.jami.jp/hcit/HCIT_SITES/job.php?job=toukei/14gishigyom.html (参照 2016-08-10)

DPC データの openEHR/アーキタイプマッピングツールの構築

Development of mapping tool from DPC data to openEHR/Archetype

櫻井 理紗¹⁾, 竹村 匡正¹⁾, 桑 直人²⁾, 岡本 和也³⁾
 , 黒田 知宏³⁾

Risa Sakurai¹⁾ Tadamasu Takemura¹⁾ Naoto Kume²⁾ Kazuya Okamoto³⁾ Tomohiro Kuroda³⁾

兵庫県立大学大学院 応用情報科学研究科¹⁾,
京都大学大学院医学研究科 EHR 共同研究講座²⁾,
京都大学医学部附属病院 医療情報企画部³⁾

Graduate School of Applied Informatics, University of Hyogo¹⁾

The EHR Research Unit, Graduate School of Medicine, Kyoto University²⁾

Division of Medical Information Technology and Administration Planning, Kyoto University
Hospital³⁾

キーワード: openEHR, アーキタイプ, Web アプリケーション, Cache', DPC データ

1. はじめに

臨床データを医学的に意味のある構造として表現するための手法として openEHR が提案されている。openEHR は EHR におけるデータの標準モデルを定義し、意味的な相互運用性が考慮されている。そのため、ヨーロッパやオセアニアにおける EHR データの標準形式として採用されつつある。一方、我が国では全国共同利用型 EHR 基盤の構築が日本医療研究機構 (AMED) 事業として採択され、日本医療ネットワーク協会によって整備が進められつつある。本事業においては収集された臨床データは openEHR における臨床的な意味モデルを表すアーキタイプによって構造化されることが企図されており、そのため、様々なデータを openEHR/アーキタイプに意味的な構造変換 (=マッピング) することが求められている。これまで、我々の研究チームでは、臨床データとして DPC/PDPS における EF ファイルのデータに対して、openEHR/アーキタイプへのマッピングを試み

てきた。具体的には、Cache' を用いた階層型データベースによって openEHR/アーキタイプのデータ構造を定義し、EF ファイルのデータに対して「データ区分」と「レセ電コード」から意味構造を判定し、openEHR/アーキタイプへのマッピングを試みてきた¹⁾。しかし、マッパーの構築自体は各データの持つメタな意味的情報 (EF ファイルならば区分コードとレセ電コードの組み合わせ) を用いてアーキタイプに対してマッピングルールを定める必要があり、これを臨床データに対して網羅的に構築するのは多大な手間がかかる。

よって、今回はこれまで作成したマッパーに対して、マッピングルールが定義されておらず、分類が未知となる臨床データ (今回は EF ファイルのデータ) を自動的に抽出することで、新しいマッピングルールを作成するためのサポートツールの構築を試みた。

2. 方法

新しいマッピングルールの作成をサポートするツールを Web アプリケーションとして構築する。今回実装する機能として、未知の E,F ファイルのデータに対して、1. マッピングルールが定義されていない EF ファイルのデータの提示、2. 抽出したデータに対して該当すると考えられるアーキタイプの候補とアーキタイプの項目の提示、として本ツールを構築する。データベースとしては Cache' データベースを用い、また Web サーバーとして Apache を用いて、PHP/HTML5/JavaScript で動作する仕組みとする。

2-1. ルールが定義されていないデータの抽出

臨床データとしての EF ファイルについては、データ項目における区分コードとレセ電コードの組み合わせが各アーキタイプの粒度と同等と考えられるため、区分コードとレセ電コードおよびレセ電コード明細名称を用いて、各アーキタイプと対応づけるマッピングルールを定義している。よって、これまでにマッピングを行っていない EF ファイルに対して、これまで出現していない区分コードとレセ電コードの組み合わせを抽出し、提示する。

2-2. アーキタイプの候補と項目の提示

マッピングルールが定義されていないデータの提示がされた後、同様の画面で各レコードの区分コードに基づいたアーキタイプの候補およびその項目を検索可能とする。これは、抽出されたデータの区分コードと同様の区分コードを選択することで、各区分コードと対応するアーキタイプの候補の提示を行う。同時に、その区分コードに基づくアーキタイプの階層構造全体、および項目が確認できるマインドマップの画像を提示させる。これによって、ユーザーはマッピングが不可能であった未知のデータに対してマッピングルールの作成手順と同様に区分コードからアーキタイプの候補を決定し、レ

セ電コードおよびその名称を確認しながら、アーキタイプのどのデータ格納場所に当てはまるかまで確認可能となる。

3. 結果

今回構築したツールの画面の一例を図 1 に示す。マッピングルールが定義されていないデータの提示し、およびそのデータに対してアーキタイプの候補およびその項目を提示することが可能であった。また、本ツールは web アプリケーションとして構築したため、複数でのマッピングルール作成作業が可能になった。



図 1 ツールの画面

4. 考察

本ツールによって、これまでのマッピングルールでは分類不可能であったデータを抽出し、提示することが可能となった。また、そのデータに対してアーキタイプの候補を提示することで、新たなマッピングルールを検討することが可能となった。今後は、本ツールから直接マッピングルールを追加する仕組みを検討する。

参考文献

1) 櫻井理紗, 竹村匡正, 糸直人, 岡本和也, 黒田知宏, DPC データの openEHR/アーキタイプへのマッパー作成の試み, 第 35 回医療情報学連合大会, 2015

当院におけるデータベース運用の歴史と現在

History of practical use and the present situation of a data base in our hospital.

渡部功之¹⁾, 田中吾朗^{1,2)}, 土屋喬義^{1,3)}

Yoshiyuki Watabe¹⁾, Gorou Tanaka¹⁾²⁾, Takayoshi Tsuchiya¹⁾³⁾

獨協医科大学 小児科¹⁾, 宇都宮協立診療所²⁾, 土屋小児病院³⁾

Dokkyo University of Medicine, department of Pediatrics¹⁾

Utsunomiya Kyoritsu medical office²⁾

Tsuchiya Children's hospital³⁾

キーワード: M 言語、Java、Caché、データベース、小児科

【はじめに】

1960年代後半に米国マサチューセッツ総合病院のコンピューター科学研究所で開発された、マルチユーザー・マルチタスクのデータシステム。当時は Massachusetts general hospital Utility Multi-Programming System といわれ、MUMPS (=おたふくかぜ・流行性耳下腺炎と同じスペル) と命名されていた。

その後、医療分野を中心に発達したが、柔軟なデータ構造、文字列処理機能、システム開発の高生産性、ネットワーク対応、コストパフォーマンスから一般のデータベース、銀行、図書館、財務管理、物流など幅広く広がった。

1990年から名称を“M”を使うことを承認され、現在では M 言語・M システムと呼ばれている。

当院、獨協医科大学 小児科 周産期母子医療センター 新生児集中治療室 (以下 NICU と略す) では、センター立ち上げの早期より M システムテクノロジーによる患者データベースの運用を行っている。1978年9月から入院した患児から現在に至るまでの約38年の間、の記録をほぼ同様のシステムを現在も運用している。

今回、我々は、今までの情報管理の歴史から現在の管理の解決すべき問題について報告する。

【目的】

当院における過去のデータベースの運用継続の間

題点を把握し、どのように対応したかを明確化する。

【成果】

当院では、1980年前後より M システムによるデータベースの運用を行っている。過去のデータも含めて1978年からの患者データが入力されており、現在も同様のシステムで運用されている。

データベース運用を開始した当時は、現在では広く普及しているオーダリングシステム・電子カルテなどは存在せず、患者情報の管理については、院内での患者 ID と手書き申込書・検査結果・フィルムによる X 線画像・カルテなどのアナログデータのみでしか情報が残せていなかった。そのため、統計情報の収集などは、情報の検索に時間がかかっていた。また、当時のパソコンの性能の問題やデータ容量不足の問題のため、情報のコンパクト化が必要であった。当時の患者管理システムを開発した医師は、言語データの管理面と格納性から M システムテクノロジーが有用だと考え選択した。

M のデータベースは当時から院内の情報管理部門のネットワーク内にあり、そこにアクセスする形で情報管理をしていた。(MSM DDP 接続によるクライアントサーバー接続) アクセスすると“プログラムのメニュー”が立ち上がり、用途に応じて選択する (図 1)。



図 1

主には、患者 ID 登録・集計・検索・産婦人科情報の連携などの項目を使用している 7 桁の ID 番号で管理されている。登録画面からは、6 桁の ID 番号 (00-0000) を入力する。入力ミス为了避免のために 7 桁目は、上の 6 桁からチェックデジットとなっていた。データベースの入力データとしては、名前・ID・治療歴・出身・主治医・手術の有無・診断名・治療内容・予後などの項目を“特殊な略語”を使用して入力する方法で入力されている。(図 2)。2016 年 4 月までの 38 年間に 10396 人の登録がある。検索については、それぞれの

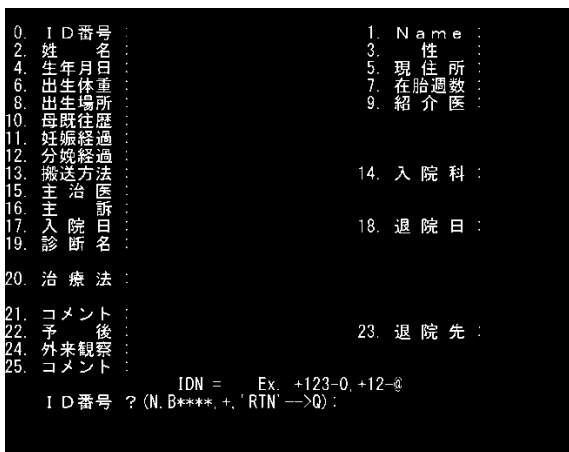


図 2

2000 年以降の問題点として、院内で患者の増加とともに患者に割り当て ID の不足が懸念されていた。20 年の間に、パソコンの普及が進み、治療オ

ーダーの電子化 (2000 年)、部門ごとの電子カルテ化 (2010 年)、病院全体の電子カルテ化 (2016 年) が進んだ。アナログベースの情報は、基本的に取り扱われなくなり、すべて電子媒体として電子カルテに取り込まれるようになった。しかし、電子カルテ化する以前の過去のデータの重要性は高く、また、取り込まれたアナログデータを電子カルテ内からの情報検索することは、困難であり今後もデータベースでの保存が必要であった。すでに当院の M システム開発した医師が引退しており、データベースの入力・取り扱いに精通したい医師でないと患者情報の入力が難しく、世代交代が難しい現状があった。

一番の問題点は、患者 ID の 7 桁から 8 桁の増加であった。ID を 79-9999-9 以降を 100-0000-0 に変更することとなった。患者 ID の桁数が増えたため、桁数が増えた当時は、100-nnnn-n 番台の ID 患者を 80-nnnn-n と入力して当面は対応していたが、長期的な運用には問題があると判断し、既存のデータベースでの運用が不可能となった。しかし、当時のプログラムを開発した医師も一部の改変で過去のデータ管理プログラムの不具合が生じる可能性があるとの指摘があった。また、当時の技術者も、プログラムの開発・改変を繰り返していたため、どのようにプログラムを作ったかが曖昧で、再構築するのは難しいと話していた。

そのため、過去のデータを用いて新しいソフトでの運用に移行することとなった。データベースは、Cache を採用した。M のデータベースは移植したオリジナルを継続して使用した。ID 番号 79-9999-9 以前はプログラムソースの変更を最小限にするため、前 0 (0nn-nnnn-n) を付与する事により対応した。

ユーザーインターフェースは最新の JavaScript ライブラリを使用し、ブラウザ間の通信に node.js を使用した。蓄積されていたデータ部分のみ取り出し、新しく作った M システムにて、HTML 表示を行ってもらった (図 2)。

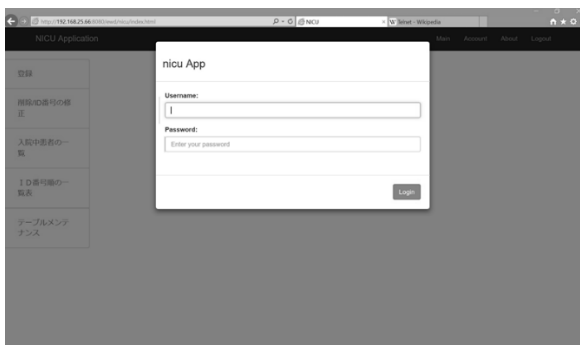


図 3

検索についてもわかりやすい表記を行い、HTML・JavaScript の特性を生かした状態で、図 3)。開発当時のパソコンよりも性能が飛躍的に高くなり、ネット環境の構築により、より使いやすい環境整備ができています。今までのシステムでは、診断名などを“特殊な略語”で入力していた。今回のシステムでは診断名・治療をチェックボックス選択して入力が可能となった。これにより、パソコンの知識がなくても入力が簡単で、入力ミスが少なくなった。それにより、今後は、パソコンに詳しくない人でもインターネットの情報入力と同じように入力可能になっており、長期的な問題は一部解決した。



図 4

【考察】

今回のことで過去の膨大なデータを扱っている

ソフト・ハードが時代とともに変化が必要となってきた現状を痛感した。しかし、開発当事者の退職・高齢化などにより、データベース管理者とそれを引き継ぐ者とのパソコンの知識の差、システム構造の理解の差、などからデータベースの継続に難渋することが分かった。また、時代の変化とともに過去のシステムだけでは対応できない状態が起きていることも事実であり、過去の技術を現在の技術で再構築していくことが必要であると考えられた。そのため、世代交代の問題、情報の引継ぎ、情報交換、技術者の育成などの課題が考えられた。

データベースソフトは、各社からいろいろ販売されており、パソコンの初心者でもある程度チュートリアルに従って簡単にデータベースの構築が可能となってきた。しかし、膨大なデータを扱う上では余計なデータも含まれていることもあり、データ量が多くなることが将来的には問題となる。Mシステムであれば、パソコンの知識がある程度必要であり、一般的な普及が難しいが、大量のデータがコンパクトなファイルサイズで扱える点や検索、処理が高速で使用してストレスが無いなどの利点も大きい。また、すでにネット環境に対応できており、入力がスムーズな点も利点である。この点から、双方での技術協力を行い現在あるデータベースの見直し再構築が今後とも必要と考えられた。

今回のことで、時代の変化とともにデータ管理の常態が変化しつつあり、それに対する対応が大切であることを痛感した。既存のデータベースの扱われているプログラムにも興味をもって理解が必要であることがわかった。今後は、若い世代への指導・教育などが必要であると考えられた。

参考文献

- 1、土屋 喬義、堀井路子、藤井崇則、伊藤章、田中千恵子、臨床現場での電子カルテシステムの開発 第 40 回 M テクノロジー学会大会 講演論文集,2013.
- 2、土屋喬義:マイクロコンピュータによる未熟児データベースの作成:1985年:第12回Mテクノロジー学会論文集

- 3、田中吾朗：未熟児部および産科データベースの構築とその結合：1993年：20回 M テクノロジー学会論文集
- 4、田中吾朗：MSM-MUMPS による周産期センター・データベース構築経験：1996年：23回 M テクノロジー学会論文集
- 5、田中吾朗：Mで構築された周産期センターDATABASE の

Windows NT 版 (MS M4.3.1) への移行経験：2000年：第 27 回 M テクノロジー学会論文集

- 6、MSM アプリの EWD.js への移行：2015年：第 42 回 M テクノロジー学会論文集

「いま」を診る、
「一生」を診つづけていく、
医療のために。

例えば、一分一秒をあらそう状況で
全スタッフが迅速に正確に、一覧性の高いワイド画面で
情報を確認し共有できること。あるいは、
10年20年後に、今日の診療情報をスピーディに検索し、
最良の診断を導きだせること。
病院の規模を問わない、
患者さまの「いま」と「一生」を
同時に見据えた質の高い医療の実現。それはすでに、
彼方に描かれた遠い理想ではありません。
[東芝ブランド]の技術力と
医療支援への情熱を注いだ HAPPY ACTIS。
いま、開発プロジェクトは次のステージへ。

東芝グループの「人のためにつくす」
という想いと総合力から生まれました。



HAPPYACTIS
Heartful system, Advanced system,
Profitable system and Package system for You
Advanced Clinical Total Information System

医療現場の安心安全を支援する新世代電子カルテシステム

- 現場の“思考と必要”を追求したワークフローと操作性
- 患者さまの安全・安心に応えるチェック機能や画面構成
- 医療ビッグデータに最適な高速・高性能データベース採用
- 安定稼働と効率運用を24時間365日体制でサポート

データの欠落はケアの欠落



すべてのデータを集めることで、よりよいケアの提供が可能に

これまでできなかった洞察や連携が可能になれば、患者ケアが大きく変わります。

InterSystems HealthShare®は、患者データを統合しシームレスな連携ケアのためのプラットフォームで、アウトカムを大きく改善することができます。

詳しくはこちらをご覧ください。 <http://www.intersystems.com/jp/healthshare/>



医療用データウェアハウスシステム
CLISTA!

データウェアハウス

医療DWH



病院のあらゆる指標を 数値化・見える化

日々の
診療データ

数値化

検証・改善

臨床

経営

研究



CLISTA!が情報の見える化を強力サポート

病院情報システム

グラフ化・情報共有

統計の自動化・データマート化

データ集約・診療データの論理統合

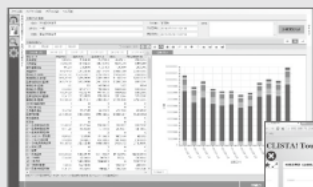
- ボタンクリックだけで様々な統計グラフを出力
- 統計データはWEBで共有
- 日々の管理統計は自動で更新
- 頻繁なデータアクセスを高速化
- 大量な病院情報システムを統合管理
- 様々なデータ形式もまとめて集約

CLISTA!はナレッジ・マネジメントを促進し、
データマイニングの新たな切り口を発掘いたします!!



統計や指標を支えるCLISTA!ツール群

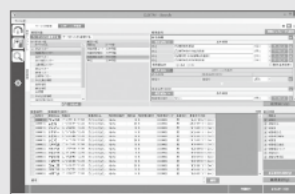
● CLISTA! 統計機能



統計の定型化・自動化/臨床分析
情報共有と情報の可視化

WEB 共有

● CLISTA! 検索機能



あらゆるデータに対する患者や薬剤の
スクリーニングや検索をサポート

● CLISTA! 原価計算



各部門から集約したデータを利用し、
多様な視点での原価計算を実現

● CLISTA! Touchboard

各システムに
またがるデータも
簡単検索

- オプション
- CLISTA! Background (統計分析自動化ツール)
 - CLISTA! Crawl (診療データ集約ツール)
 - CLISTA! View Editor (診療データ論理統合ツール)
 - CLISTA! 原価計算 (患者別原価計算/診療科別原価計算など)
 - CLISTA! DPC (DPC分析モジュール)

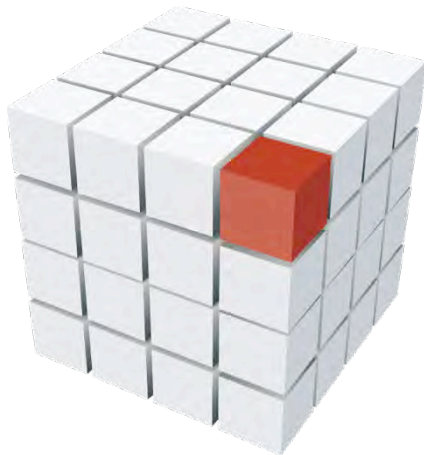


株式会社 医用工学研究所

Medical Engineering Institute, Inc.

<http://www.meiz.co.jp> e-mail: sales@meiz.co.jp

私たちは自由な発想にユニークさを添えて
お客様のニーズを“カタチ”にします。



新しいモノ、発想、テクノロジーで、
お客様と一緒にベストな医療情報システムを
作り上げていく姿勢は、多くのユーザー様に
必ずお役立ちできるものと確信しています。

長年の Caché / Ensemble' 開発で培われた
独自の技術をご提供します。



- ・電子カルテ/オーダーリングシステム (CRESC)
- ・診療情報DWH (med@CUBE)
- ・循環器疾患診療支援システム (med@Heart)
- ・不妊治療支援システム (Warabee)
- ・地域連携システム (med@Link)
- ・調剤支援システム
- ・各種OEM製品(医療ベンダー向け)

データキューブ株式会社

www.datacube.co.jp

〒805-0071

福岡県北九州市八幡東区東田1丁目5番7号

九州ヒューマンメディア創造センター 3F

TEL : 093-663-2931 FAX : 093-663-2933

- | | |
|--------|--|
| 福岡営業所 | 〒812-0011
福岡市博多区博多駅前1丁目14番16号 博多駅前センタービル7階
TEL : 092-433-3366 FAX : 092-433-3376 |
| 沖縄営業所 | 〒900-0033
沖縄県那覇市久米2丁目4番14号 JB-NAHAビル8階
TEL : 098-963-6022 FAX : 098-963-6033 |
| 鹿児島営業所 | 〒890-0053
鹿児島県鹿児島市中央町22-16 アエールプラザ3F
TEL : 099-298-1588 FAX : 099-298-1588 |
| 宮崎営業所 | 〒880-0811
宮崎県宮崎市錦町1番10号 KITENビル5F
TEL : 0985-88-3456 FAX : 0985-88-3457 |
| 東京開発室 | 〒105-0003
東京都港区西新橋1-18-6 クロスオフィス内幸町7階
TEL : 03-6457-9730 |

株式会社プロアスは医療機関様のお役にたてる企業をめざし、日常の運用に係るお困りごとから、経営改善のご支援まで幅広いサービス・製品のご提案を致します。また、有用な情報交換・ご提供を積極的に行い、愛され必要とされる企業として成長を続けます。

医事会計システム

正確に、迅速に、末永く。

中小規模病院様の医療事務現場を支えるシステムです。



データベースに **Cache'**を採用！

データを瞬時に処理でき、システムレスポンスを長期に渡って保てるという特性を持ち合わせています。

豊富なインターフェースを備え、様々なシステム連携機能を搭載したことにより、「電子カルテ」「オーダーリング」「再来受付機」等々の他システムとの連携も可能となっております。

お問合せお待ちしております。



リサイクルトナー

いつも使うトナーだからこそもっと安さと品質を追求しませんか？



価格の安さがリサイクルトナーの魅力！

純正品定価の**約80%OFF**でご提供します。

プリンターメーカー	トナー型番	純正品定価	プロアス価格	差額
Canon	519	8,900円	3,800円	-5,100円
	519 II	23,500円	4,300円	-19,200円
Ricoh	SPTナー 6100	30,000円	4,200円	-25,800円
	SPTナー 6100H	60,000円	4,400円	-55,600円
Fujitsu	LB319A	37,950円	7,200円	-30,750円
	LB319B	50,600円	7,800円	-42,800円

まずは「コスト削減無料診断」をご依頼ください。

- ①プリンタ及びトナー型番、現在の購入価格・月間使用本数をご記載いただき、メール送信 (sales_consul@proas.co.jp)
- ②弊社よりご提案可能なリサイクルトナーの御見積書・コスト削減シミュレーション表のご報告



proas

株式会社プロアス www.proas.co.jp

大阪本社：〒532-0004 大阪市淀川区西宮原1-5-33 新大阪飯田ビル4F TEL.(06)6397-5511(代) FAX.(06)6397-5510
東京支社：〒104-0061 東京都中央区銀座8-15-2 ACN銀座ビル5F TEL.(03)6278-7470(代) FAX.(03)6278-7480

総合医療情報システムの提案から 導入・運用管理までをトータルサポート致します。

1976年（昭和51）設立以来、永年に渡り多くの総合病院へシステムを導入し一貫した保守サポートを提供させて頂いております。この経験・技術を活かし、今後も高品質な総合医療システムをご提案させて頂きます。

「『人』と『コンピュータ』の調和をサポートする」をコンセプトとしてお客様を主人公にすえ、ともに病院情報システムを築き上げていく姿勢は医療に従事される全ての方々のお役に立てるものと確信しております。



医療とITの架け橋となる

山ニシステムサービス株式会社

<http://www.yamani-system.co.jp>

代表取締役社長 西村 紀一郎

代表取締役専務 佐々木 攻

本社：〒010-1427 秋田市仁井田新田2-16-13

TEL 018-839-2881 (代表)

青森営業所：〒030-0946 青森市古館1-12-10

TEL 017-741-1384 (代表)