
サービスロジックを基盤とした医療情報 SOA の
知的構築に関する研究

2017年3月

北海道科学大学大学院

廣田 健一

目次

第1章 序論

1-1 本研究の目的	4
------------	---

第2章 医療情報システムの役割とクラウドコンピューティングの可能性

2-1 医療情報システムの変遷と役割	6
2-1-2 医療情報システムと情報共有	12
2-1-3 企業情報システムと医療情報システムの比較	14
2-1-4 病院間連携におけるシステム連携	15
2-1-5 医療情報システムと医療安全	16
2-1-6 各種医療機器の接続と紙文書の電子スキャン	20
2-1-7 医療情報と医療の質	25
2-2 医療情報システムの新たな利用形態	26
2-2-2 診療情報の整理	27
2-2-3 診療カルテ記載情報におけるデータアプローチ	29
2-2-4 B型肝炎再活性化防止システム	33
2-3 クラウドコンピューティングの活用	38
2-3-2 クラウドコンピューティングにおける SOA の展開	40

第3章 医療情報システムの複雑化と医療政策における統合化への対応

3-1 医療情報システムの複雑化	41
3-1-2 従前のソフトウェア開発プロセスと改善の兆し	44
3-2 医療情報システムの統合化への動き	48
3-2-2 統合化における課題と論点	53

第4章 医療行為の標準化とクリティカルパスウェイ

4-1 クリティカルパスウェイによる医療行為の標準化	55
4-1-2 クリティカルパスウェイ支援システム	57
4-1-3 クリティカルパスウェイの作成・検索・適用への支援	59

4-1-4	クリティカルパスウェイ作成支援システム (オーダ情報統合テーブルの利用)	60
4-1-5	クリティカルパスウェイ条件検索システム	62
4-1-6	クリティカルパスウェイ情報蓄積システム	64
4-1-7	ロボット支援腹腔鏡下前立腺全摘除術のクリティカルパスウェイ	65
4-1-8	クリティカルパスウェイ支援システムの効果と課題	67
第5章 クラウドコンピューティングサービスにおける サービスロジックの位置づけ		
5-1	クラウドコンピューティングサービスの活用	68
5-1-2	ICT におけるサービスの定義	70
5-1-3	API 連携によるサービスの創出	72
5-1-4	医療サービスの創出	74
5-1-5	サービス連携におけるサービスロジックの位置づけ	75
第6章 サービスロジックを基盤とした SOA の知的構築		
6-1	SOA 基盤へのサービスロジックの実装	76
6-1-2	SOA におけるプラットフォーム概念モデル	78
6-1-3	Web サービスと Java コントロール	80
6-1-4	サービスロジックを基盤とした SOA	81
6-1-5	クラウドコンピューティングにおける SOA の展開	84
6-1-6	医療 SOA におけるクリティカルパスウェイサービス例	85
6-2	医療情報の共有における現状と課題	88
6-2-2	ニューロロジカル・レベルによる医療情報の知的表現	89
6-3	ランドルト環方式による理解	92
6-3-2	ランドルト環方式 e ラーニングによる情報共有	93
6-4	XML 言語の利用によるプログラミング強化	95
6-4-2	XML 言語の概要	97
6-4-3	XML 言語におけるタグ拡張機能	99
6-4-4	XML 言語におけるオブジェクト拡張機能	101
6-4-5	XML 言語におけるデータベース接続	102
6-4-6	XML データベース	103

6-4-7 XML 言語における Prolog タグの利用	106
6-4-8 XML 言語における推論ロジック	108
第 7 章 クリティカルパスウェイ指向の医療情報 SOA	
7-1 ロジック記載	116
7-2 シチュエーショナル・アクティブ・API によるサービス	117
7-2-2 クリティカルパスウェイサービス創出の仕組み	119
7-2-3 サービスロジックを基盤とした シチュエーショナル・アクティブ・API	121
第 8 章 結語	122
参考文献・参考 URL	123
用語説明	129
謝辞	135

第1章 序論

1-1 本研究の目的

サービス・サイエンスはSSME(Services Sciences Management and Engineering)の簡略した表記であるが、情報科学・工学にマネジメントを融合させた新しい学問分野であり、サービスを汎化し科学することでICT(情報技術)との相乗効果による体系化した高い品質のサービスを作り出すための基礎学問として位置づけられている。

サービスロジックはこの中核をなすものであり、高いサービス品質を保証することになる情報の品質を論理的に定義する基盤となるものである。しかし、情報とは何かという点で品質を明確化できているとは言えない。特に、医療における情報の品質を考えてみると、そこから単純に医療(サービス)の質の向上につながり、生死を左右する責任を負うことを考えなければならない。よって、医療における情報の品質を高めることは、医療の質の向上に大きな影響を与えることは明白である。

わが国においては、厚生労働省が大規模病院における入院時の診療報酬の計算方法として診断群分類包括評価(DPC)を義務付けており、毎月提出される調査データ(EFDファイルなど)の内容をもとに医療行為の標準化、さらには医療の質の向上までを目指した取り組みを行っている。しかし、医療の質の評価は、構造・過程・結果の3つの側面で計測されることが必要であり、1995年に日本医療機能評価機構が設立されていることから日本において医療の質への大きな関心が集まっていることは間違いないが、その中で現在重要視されているのは構造(施設基準、医療体制等)に対する評価である。医療の質は構造・過程・結果の3つの側面による評価を行う必要があり、過程・結果の評価方法の確立が必要である。そのため、臨床現場においては、医療の質の向上に向けた様々な取り組みが実施されており、その中のひとつとして、クリティカルパスウェイは、診療ガイドラインに沿った標準的な医療行為を実施する上で効果的であることから、各病院において導入されている。

そのような状況下、厚生労働省は、同省内に大臣を本部長とした「データヘルス改革推進本部」を設置し、2017年1月12日にICTを活用した次世代型保健医療システムの実現に向けた施策について検討会を開催しており、その中で、わが国の健康・医療・介護施策におけるICTの利活用は、様々な縦割り構造を背景に、その前提となるデータが分散し、相互につがらない形で取り組みが進められてきた結果、一体的に機能していないと指摘した上で、膨大な健

健康・医療・介護のデータを整理し、それらを収集・分析して、これからの健康・医療・介護分野のICT利活用が「供給者目線」から「患者・国民、利用者目線」になるようにICTインフラを作り変え、健康・医療・介護施策のパラダイムシフトを実現していかなければならない、そして「ビッグデータのプラットフォームを構築する必要がある」と提言しており、2020年度から本格稼働を目指すという壮大な提言を行った。

しかし、これらシステムの構築には、わが国のIT史上でもまれに見る大規模な環境整備が必要であり、実現するためには、新たな手順や仕組み（システム仕様の確定、標準化の推進、各種システムの連携、ソフトウェア開発手法、ハードウェア基盤選定など）が必要となる。しかし、これらについては述べられてないため、仕組みを考えることは必須であり、急務である。

このような統合化されたICTプラットフォーム基盤における医療サービスの構築は、医療の適切性、正確性を支援するためのものであり、形式化された知識の組み込み素材、方式が重点課題となる。そのためには、医療と情報科学・工学を組み合わせることが必須であり、組み合わせによって発見された、医療知識の形式化やそれらを用いた医療の質の向上を実現するためのシステム方式を提示することは、学際的な医療情報における応用研究として非常に価値が高いと考えている。

本研究では、医療情報の品質を高め、医療の質の向上を目的として、そのために必要な次世代の医療情報システムを実現するための仕組みについて述べる。

概観としては、クラウドコンピューティング上でSOA(Service Oriented Architecture)を用いたSOA on Cloudによるサービス機能ネットワークを示し、XMLを用いた知識ベース（推論機能をもつデータベース）、医療サービス、既存システムなどをJavaコントロール技術により連携させ、ITサービスを創出するソフトウェア・コンポーネントが利用されたサービスロジックを基盤とした、医療情報SOAプラットフォームについて述べる。さらに、リアルタイムな医療情報の反映にはニューロロジカル・レベルによる医療情報の整理とランドルト環方式による理解度に応じた情報共有を応用した知的構築について考えた。

また、言語として機能するXML言語とその拡張機能であるタグ拡張とオブジェクト拡張によるプログラミング強化を行い、SQLタグやPrologタグを用いてXML-DB(XMLドキュメントベース)にアクセスし、推論機構をもつデータベースとして扱えることを示し、これらの技術が医療情報SOAプラットフォームの構築に応用することの有用性について述べるとともに、サービスロジックを実現させるための技術としてシチュエーション・アクティブ・APIの必要性について論じる。

第2章 医療情報システムの役割と

クラウドコンピューティングの可能性

2-1 医療情報システムの変遷と役割

情報システムとは、組織体（または社会・個人）の活動に必要な情報の収集・蓄積・処理・伝達・利用にかかわる仕組みである。広義の情報システムは、人的・機械的なものを含み、コンピュータを中心とする機械的なものを指す場合は狭義の情報システムと呼ばれる。コンピュータによる情報システムは、主に企業経営の分野で多く用いられ、トランザクション処理、情報報告、オフィスオートメーション、意思決定支援、戦略情報のために利用されている。

一方、1970年代以降、医学の分野においても情報システムが導入され、医療情報システムの研究が進められてきた（図1参照）。医療情報システムは主に二つの視点から発展してきた。一つ目の視点は、事務的作業の軽減を目的とした医療情報システムであり、1970年代以降の第一世代の医療情報システムを部門システム導入期と呼び、各部門に情報システムが導入され、医療事務会計、レセプト作成、臨床検査などの電子化が行われた。1985年代以降の第二世代はオーダーリング導入期と呼ばれ、オーダ伝票、部門システムなどの電子化が行われた。2000年代以降の第三世代は電子カルテ導入期と呼ばれ、カルテの記載や各種医療機器との接続などの電子化が進められてきた。二つ目の視点として、診療の診断支援を目的とした医療情報システムがある。これは医療情報システムを通して診断に必要な情報を提供し、意思決定や診断支援を目指したシステムである。この二つ目の視点から進められてきた医療情報システムはその難易度の高さ、実現性の難しさから進められてこなかったが、ここ数年はA I（Artificial Intelligence）を活用した診断支援を目指した動きが出てきた。しかし、それらはA Iと同様に「診断支援とは何か」という定義が曖昧なまま進められてきたため、実際には一部の大学病院にて「診断支援らしき動き」を医療情報システムにて実施しているに過ぎない。また、医療情報システムも大手ベンダーによって提供されるパッケージソフトを活用することが大学病院においては普通となり、スクラッチ&ビルドタイプの開発をしている病院は皆無に等しい。今後の方向性としては、基本部分をパッケージソフトにより実現し、そこに各種チェック機能、各種医療機器との接続、紙媒体のさらなる電子化、そしてA Iを活用した診断支援を加えた形のハイブリッド型の医療情報システムとなることが考えられる。

国内の医療機関におけるIT化の変遷

これまでの医療機関におけるIT化は大きく4世代に分けられる。

医事会計計算処理の高速化からはじまり、オーダーエントリーシステムの導入による診療現場からの各種オーダー入力、電子カルテシステムの導入によるペーパーレス・フィルムレス化、そして現在は中核病院と周辺の診療所等が連携することで患者さんに効率的で適切な医療を提供する地域医療ネットワークの導入がすすんでいる。

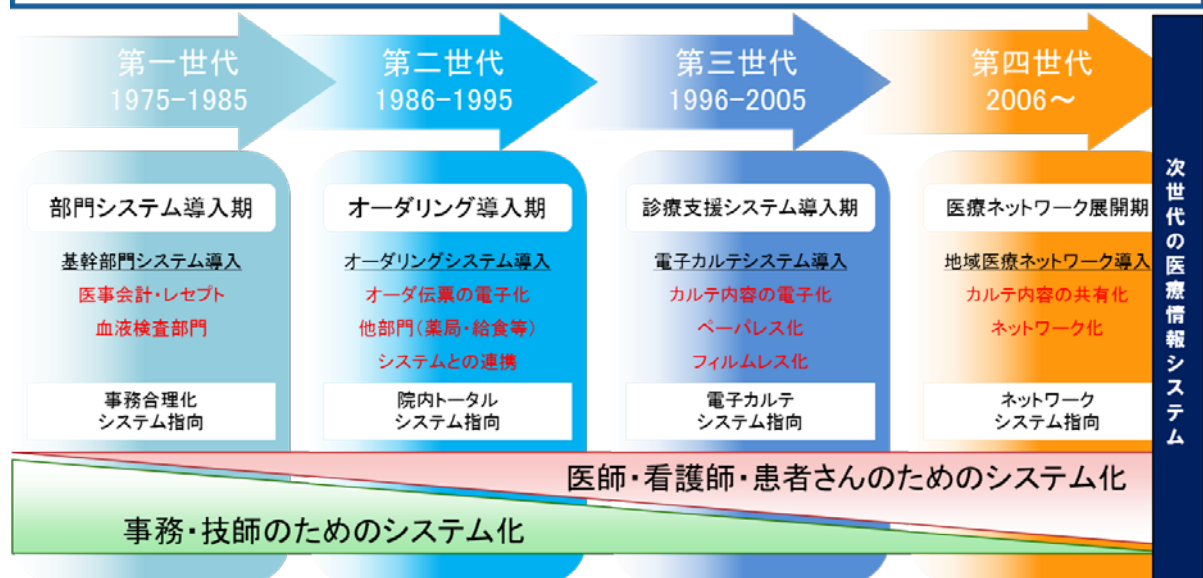


図1 医療情報システムの変遷

医療情報システムの機能範囲は広く、その全体像を把握することが難しくなっているが、個々のサブシステムを抽象的に捉えたと、掌握可能な数のコンポーネントが組み合わさったシステムとして見ることができる。医師は、診療現場から診療記録を入力し、処方や検査等の依頼（オーダー）を発行する。オーダー情報は、実施部門に伝達されると同時に電子カルテシステムに保存される。実施部門では、実施した行為や使用材料が入力され、これらが電子カルテシステムに記録されると同時に医事会計部門に伝達され、会計計算のために処理される。検査部門等の部門システムでは、レポートや画像が生成され依頼医に伝達される。使用した材料の情報は、物流管理システムにも伝達され物の補充に利用される。物流管理システムは、物の在庫を把握し、消費したものを購入し在庫数を把握する。経営管理部門は、収入データを医事システムから、費用データをオーダーの実施記録と物流管理システムから把握する。

病棟においては、医師は薬剤部に処方情報を伝達すると同時に看護師に対して服薬、注射の指示を出す。その他、直接看護に関わる事項について指示だしを行う。看護師は、医師からの指示やオーダー情報を収集して、看護計画を立て、日々実施した看護、観察の記録を電子カルテに入力する。

また、医療情報システムは、各部門システムと部門を結ぶシステムに大きく

わかる。部門を結ぶシステムの代表例は、オーダエントリシステムである（図 2 参照）。その他に、予約システム、診療録情報の管理機能（狭義の電子カルテシステム）、看護システムなどがある。部門を結ぶシステムは、複数の部門が一つのデータベースを操作する構成となる。つまり、診療録管理システムのデータベースを介して部門と情報を伝達している。

部門システムは、それぞれの部門の業務を支援するシステムである。検査部システム、放射線部システム、薬剤部システムなどがその代表である。医事会計システムも、一つの部門システムと捉えることができる。部門システムは、部門にデータベースがあり、その入出力を部門の職員が実行する。部門でシステムが担うべき業務が複雑で多岐にわたる場合は、部門システムを構築して対応することが良い。部門システムは、医療情報システムとは別のベンダーであることもめずらしくない。その中でオーダエントリシステムは部門を結ぶシステムの中心である。外来、病棟からオーダが発行され、データベースに書き込まれる。この情報が部門に伝達され、部門システムがこれを受けて処理する。また、部門側から実施情報をデータベースに送信する。医事会計システムは、会計計算が必要となったタイミングで、データベースから実施済みで未会計のオーダ情報を受け取り、医事会計計算を行い、オーダサーバに会計済みのフラグを書き込む。全てのシステムがこのような形の連携をするわけではないが、基本的にはこのような動作となる。

近年では、すべての部門においてシステムを導入し、業務の効率化を行っている。薬剤部、検査部、放射線部などでは、古くから部門システムが導入されてきた。これらに加えて、手術部、リハビリ部、病理部、内視鏡センターなど、病院のあらゆる部門で、それぞれに対応する部門のシステムが開発され、活用されており、導入されていない部門は存在しない。

診療録（電子カルテ）に記録されるべき情報には、(1) 症状や病歴、身体所見、診断や計画、入院期間などのサマリなど、患者の診療に携わる医療従事者が、診療する過程で発生する記録部分や、(2) 処方や検査、注射、処置等のオーダと連動した実施記録部分、(3) 検査報告書などの中央診療部門で作成される報告書が存在する。これらの情報を、データベースに記録し閲覧するシステムであり、さらにバックアップするなど保存性が高いシステムが求められている。

医療現場では、放射線画像や超音波画像、内視鏡画像など、多くの画像情報が扱われている。これらは、画像データベースで管理されており、画像データの場合、必ず画像発生機器からのデータの受け取りになるが、データベースで管理する場合、患者 ID、画像の種類、発生日などの属性情報が必要になる。画像発生機器に、ユーザが直接これらの情報を入力する運用には手間がかかり間違いが発生しやすいため、医療情報システムから画像発生機器側に属性情報を

送り、この情報を付加した形で画像データが送り返されることとなる。これら画像データについては、標準規格として DICOM 規格が広く受け入れられている。

また、病棟の看護師が操作するシステムは、看護過程をコントロールするシステムに加え、医師との連携部分が多い。処方、注射、処置の指示を受け、投与実施を登録する部分においては、オーダエントリシステムの一機能を担うことになる（指示簿システム）。看護診断から看護計画を立てる部分は、看護システム特有の機能であるが、バイタルを入力し温度板を作成する部分は、電子カルテのひとつの機能と言っても良い。このように、看護システムは多くの機能が複雑に混合しており、さらに、看護業務として連続性が求められるため、最も開発が難しいものの一つであり、各ベンダーによってシステム内容が大きく異なることが多いシステムである。つまり、ベンダーを変更する場合、データ移行が難しいシステムの一つである。

部門システムで代表的なものであるのが、放射線部門システムである。放射線部門は、画像診断部門、核医学部門、治療部門に分かれており、診断部門、核医学部門のシステムは基本的に外来、病棟からの検査依頼に対して画像とレポートを作成、返信し、医事に会計情報を伝達する。核医学部門は、検査前にラジオアイソトープの購入やその注射が必要であり、管理区域の出入管理、アイソトープの管理が必要となる。

放射線検査のオーダでは、内科、外科の主治医が検査依頼をすることになる。検査依頼の内容は、患者が検査室に行って検査を受けるため、様々な情報を伝達する必要があり、また、検査室、検査機器を一定時間占拠するために予約が必要な検査が多い。外来の診療予約のように単位時間当たりの検査枠を用意して主治医が自由に予約枠を取る方法（オープン予約）と、血管造影のように、放射線科医が検査依頼内容を見て決める場合には検査枠を公開しない場合（クローズ予約）がある。

単純撮影では予約なし検査で運用するケースが多いが、オープン予約枠に設定し、制限を 999 人などにして運用することが多い。外来の診察前採血と同様に単純 X 線撮影後に画像を見て外来診察とする場合もあり、外来診察予約に検体検査の採血や放射線検査が連携される必要がある。

検査依頼情報については、(1) 依頼者情報：検査依頼内容についての問い合わせなどで必要、(2) 患者基本情報：患者 ID さえわかればデータベースから取得できるが、放射線部門で保存が要求されている照射線では患者住所などが必須である。保険種別も保険請求時に必要である。外来/入院の別、病棟名も患者呼び出し等に必要である。(3) 患者診療基本情報：造影剤のアレルギーの有無あるいは過去の造影剤使用歴、腎機能障害の有無、妊娠の可能性の有無（妊娠初期が X 線被爆の影響が大きい）、搬送の状況、障害の有無などが要求される。(4)

診断名/病歴/検査目的：診断名はレセプト請求を考え、登録病名と連携する必要がある。病歴、検査目的は検査方法の選択時と読影時に必要である。(5)検査名称は腰椎6方向、胸部単純4方向撮影などセットの検査名があり、撮影時に分解して施行される。検査によっては「右側臥位で正面像」など細かい指示がコメント欄あるいは病歴欄などに記載される。(6)検査室説明、同意書の発行が同時に行われる。画像検査オーダは、画像データ、レポートと連携する必要があるが、多くの場合、オーダ番号がHIS、RIS、検査機器、PACS、レポートシステムに伝達されてキーになることが多い。しかし、RISで発生する検査番号もDICOMでは重複を許さない番号（Accession Number）を採番している。これらオーダ番号あるいは検査番号により、検査依頼と画像とレポートは連携して表示できるような仕組みとなっている。

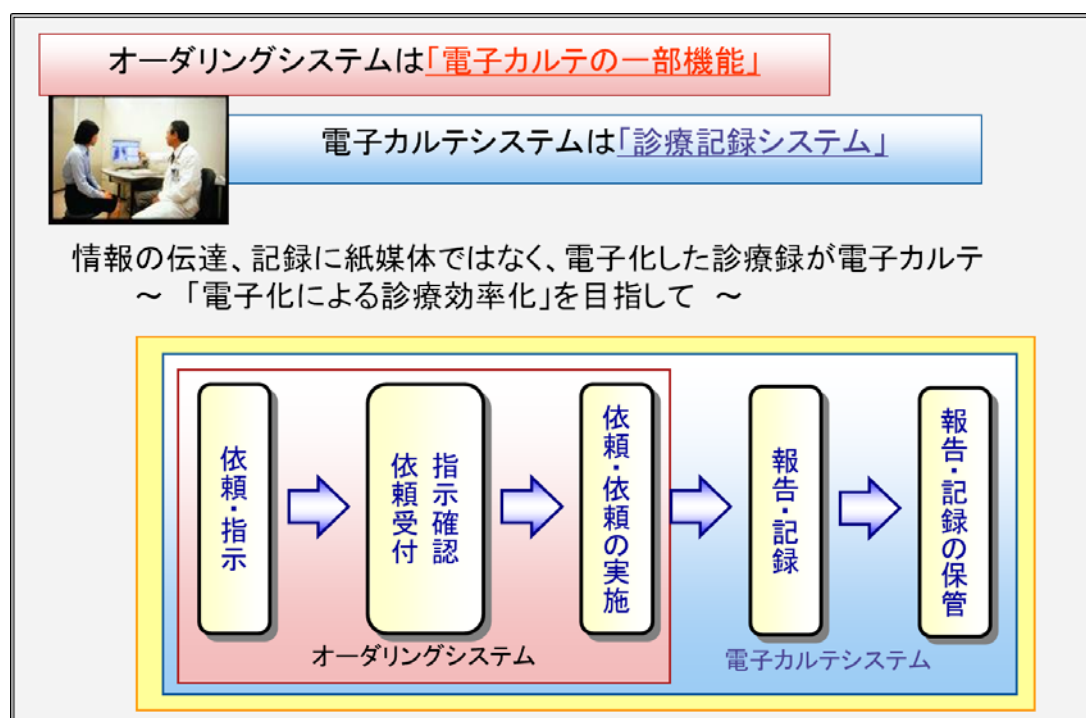


図2 電子カルテとオーダーリングシステムの違い

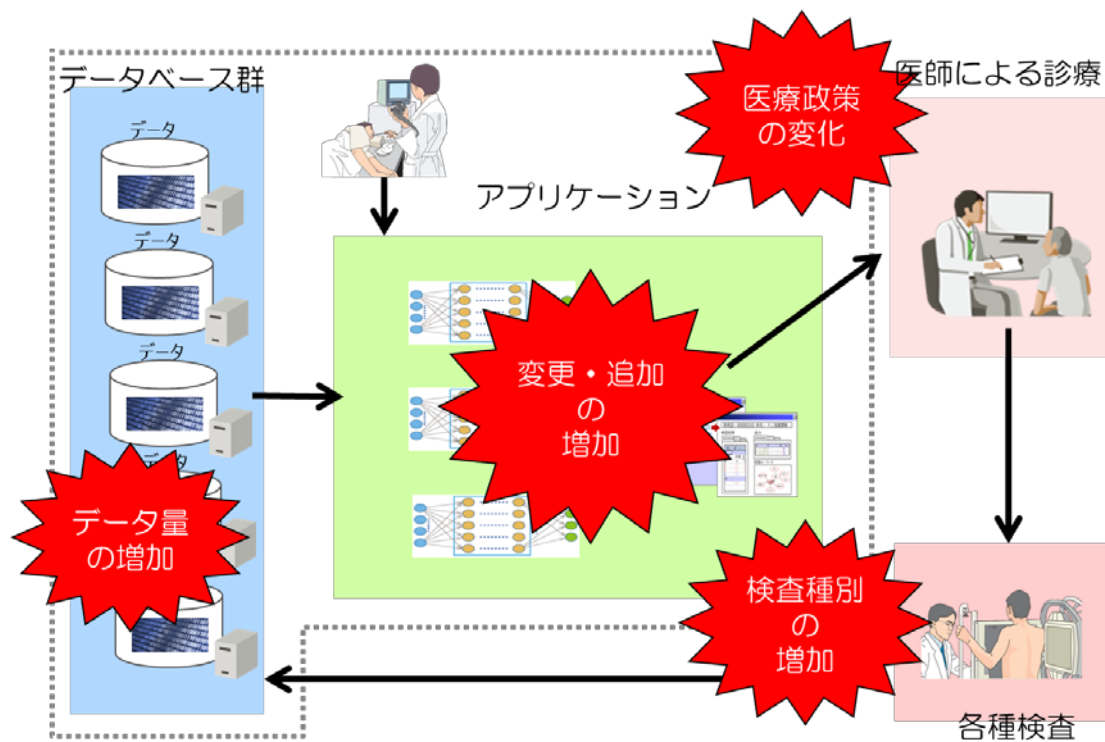


図 2-1 医療情報システムを取り巻く環境

2-1-2 医療情報システムと情報共有

1970年代以降、医学の分野においても情報システムが導入され、医療情報システムの研究が進められてきた。医療情報システムは主に二つの視点から発展してきた。一つ目の視点は、病院内における事務的作業の軽減を目的とした医療情報システムであり、1970年代以降の第一世代の医療情報システムを部門システム導入期と呼び、各部門に情報システムが導入され、医療事務会計、レセプト作成、臨床検査などの電子化が行われた。1985年代以降の第二世代はオーダーリング導入期と呼ばれ、オーダ伝票、部門システムなどの電子化が行われた。診療報酬上の加算もあり、放射線画像情報システムの導入が広がったのもこの時期である。2000年代以降の第三世代は電子カルテ導入期と呼ばれ、カルテの記載、自科検査、手術動画保存などの電子化が進められてきた（図3参照）。二つ目の視点として、診療の診断支援を目的とした医療情報システムがある（表1参照）。これは医療情報システムを通して医師へ診断に必要な情報を提供し、意思決定や判断支援を目指したシステムである。近年における医療情報システムは一つ目の視点である事務的作業の軽減を目的として進化しているが、クリティカルパスウェイなどの様々な医療者が関係すべき医療を進めるためには今以上の情報共有が必要であり、病院をまたいだグローバル・クリティカルパスウェイを実施する場合、その情報共有は病院間においても実施しなければならない。しかし、情報共有を行う場合は、データの標準化や情報共有ツールも必要となり、医療現場はあまり情報共有が整備された場所とは言えない面がある。



図3 患者の導線に係わる医療情報システム

表1 代表的な医療情報システム化の範囲

No.	システム機能
1	診療情報管理
2	外来診療業務
3	入院診療業務
4	オーダ発行
5	電子診療録
6	看護支援
7	自動精算機
8	医療事故防止
9	医事会計
10	物流管理
11	データウェアハウス
12	経営管理
13	病床管理
14	病歴管理
15	給食・栄養
16	検体検査
17	細菌検査
18	生理（生体）検査
19	放射線診断及び放射線治療
20	薬剤
21	輸血
22	病理
23	手術
24	リハビリ
25	地域連携
26	自科検査
27	チーム医療

2-1-3 企業情報システムと医療情報システムの比較

企業における情報システムと医療情報システムの違いについて考えると、システム・アーキテクチャとしては、同じと考えても問題はない。なぜなら、現在においては企業システムと医療情報システムともにパッケージソフトによる実現が多いためである。2000 年以前までに日本において多くみられたスクラッチ&ビルド形式による、カスタムメイド型情報システムはパッケージソフトにおける価格、性能、安定の優位性により大幅にシェアを落としており、パッケージに必要機能をアドオンする形による情報システムが大半を占めている。それは企業情報システム、医療情報システムの両者に言えることである（図4参照）。また、欧米においてはパッケージソフトに業務を合わせる事が通常であることから、日本においてもこれらの流れはより広まることは間違いない。よって、企業における情報システムにおいては標準的に行われている。ビジネスプロセスの全体最適を考えたサプライチェーンマネジメント（SCM: Supply Chain Management）の考え方を医療情報システムの連携においても活用されると思われ、そのための情報の共有方法、共有内容、これらを実現するために必要な事項の整理を行う必要がある。例えば、リレーショナル・データベース（RDBMS）においては、正規形と呼ばれる形式にすることにより、データの一貫性の維持と効率的なデータアクセスが可能になる。つまり、正規化を行うことにより、データの冗長性と不整合を防ぐために整理を行っている。

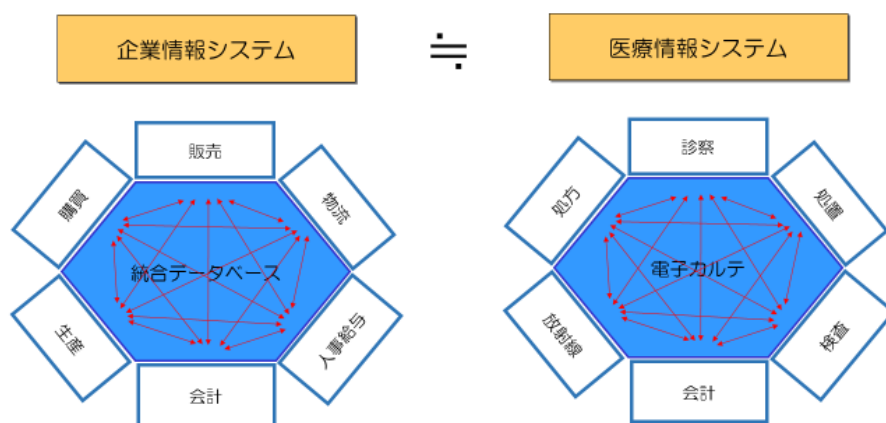


図4 企業情報システムと医療情報システム

2-1-4 病院間連携におけるシステム連携

病院内だけではなく病院間の連携を考えるにあたり問題となってくるのが、各病院において電子カルテシステムのパッケージの違いから情報の品質、形式が違う点が上げられる。例えば、病院間の連携における事例として、地域連携クリティカルパスウェイが存在する。地域連携クリティカルパスウェイとは、急性期病院から回復期病院を経て早期に自宅に帰れるような診療計画、治療結果を作成し、治療を受ける全ての医療機関で共有して用いるものであり、この連携の際には各病院間で電話連絡、メールにより患者情報の連携を行っているが、患者情報の連携をお互いの電子カルテシステムにて行うことができれば、より正確に地域連携を進めることが可能であると考えられる。しかし、病院における情報システムにも平成 18 年度より標準化ストレージの規格として SS-MIX2 がすべての医療機関を対象として医療情報の交換・共有による医療の質の向上を目的とした「厚生労働省電子的診療情報交換推進事業」を開始しているが、医療機関により、診療上の特性や規模、経済状況等の事情から導入に強制力はない。そのため、一部の病院、診療所しか対応していないのが現状である。また、SS-MIX2 の形式では電子カルテ上の記載情報は規格の対象となっていないため、カルテ記載情報を含めた医療情報の品質を整える必要があると考えられる。診療録へのコーディングの統一、規約作成など、診療録の整理が重要である。

2-1-5 医療情報システムと医療安全

電子カルテを「正診療録」として運用している大学病院内は、電子カルテに加えて様々な部門システムも稼働しており、これら医療情報システムにおける潜在的な危険を未然に、早急に解決することが事故の防止に大変必要である。医療安全の確立には、状況にもとづく適切な警告、危機事象情報の収集と分析、分析結果にもとづく適切な教育の3つが必要とされている。

人間は間違えるため、システムのなカバーができる、より安全な体制・システムを作ることが重要であり、そのためにエラーチェック機能が現在では重点的に改修されている。また、医療情報システムが関係した情報によるインシデントは3割近くになるとも言われている。これらはすべての医療行為に医療情報システムが関与することになり、利用範囲が拡大したことと、さらに医療情報の品質が医療安全では重要なファクターとなっていることを物語っている。さらに医療におけるインシデント（図5参照）が発生した場合、インシデント・アクシデントレポートシステムにより、発生した事象を現場で記録し、その分析が行われているが、これらは医療情報システムの一部であることから、医療機器と同様に病院運営において、必要不可欠の存在となっていることは間違いない。しかし、これら情報化によって、かえって医療安全が脅かされる場合も散見されるのは事実である。つまり、医療情報システムにおける情報の品質が、医療の質に関係していることは明確であり、医療情報における最重要な課題と考える。

以下にインシデントの発生を未然に防いだ例の一部を記載するが、ベンダー標準パッケージでは、これらを防ぐための施策は考えられていない。医療情報技師が医療情報システム、医療安全という両面から医療機関が色々な場面を想定しながら進めているというのが実態である。

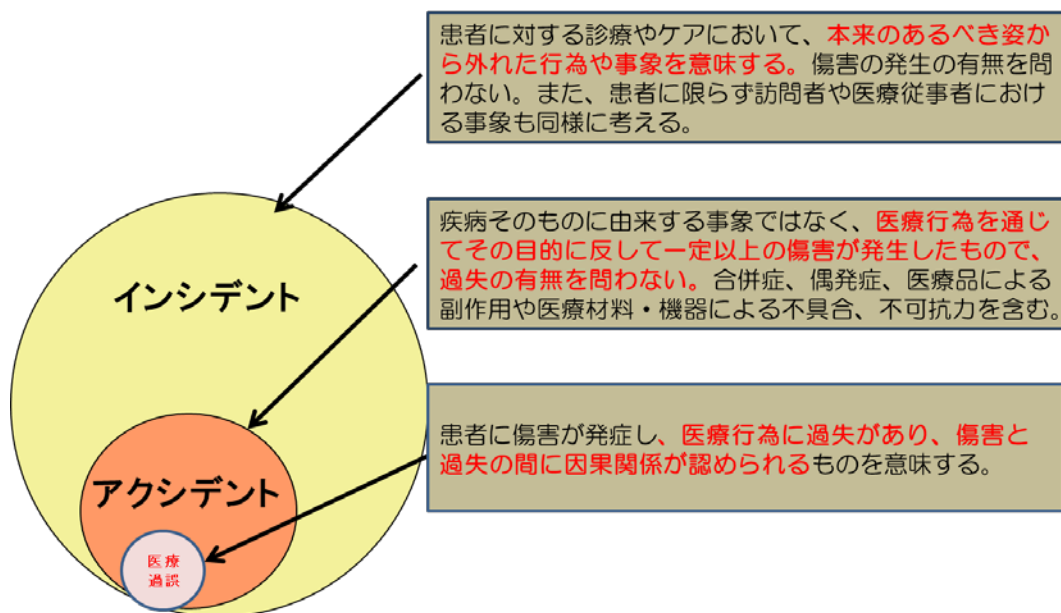


図5 インシデントと医療過誤

〈イシンデント発生予測の例①〉

体内植込み型の除細動器（ICD）、CVポートなどがあるにも関わらず、CT/MRを撮影する場合。なお、自己申告、問診を行わない場合。

■ 体内埋込アイコンの表示と入力フィールドの作成

Before	After
<p>患者バー</p> <p>リハ 030 リハ 030 75歳6ヶ月</p> <p>リハ 030 リハ 030 75歳6ヶ月</p> <p>【インプラント】 ステント</p> <p>※アイコンにマウスカーソル触れると、登録されている内容がポップアップ表示されます。</p>	<p>患者バー</p> <p>リハ 030 リハ 030 75歳6ヶ月</p> <p>リハ 030 リハ 030 75歳6ヶ月</p> <p>【体内埋込み】 ステント 人工弁 ペースメーカー・ICD</p> <p>※アイコンにマウスカーソル触れると、登録されている内容がポップアップ表示されます。</p>

＜イシメント発生予測の例②＞

遠方から来られた患者に対して、再診予約日と採血日を同一日にするのを忘れてしまった。そのため、採血を行わなかった場合。

- 再診予約オーダの予約日時を検体検査オーダの次回診察日とする

After

エディタに展開されている再診予約オーダの予約日時を、検体検査オーダの次回診察日として引き継ぎます。

【例】一般検体検査

エディタ(新規) - テスト 077(9999000772)

【プログレスノート】 2011/03/04(金) 09:55 外科 外未
01時: 2011/03/04(金) 09:55 医師: 医師 (ほか)
作成: 2011/03/04(金) 09:55 作成者: 医師: 医師 (ほか)

採取日付 2011/03/25 次回診察日 時刻 10:00

再診予約の再診日の日付(2011/03/25)を引き継ぎました。

外注: 腫瘍マーカー 外注: 膠原病 外注: 感染症・ウイルス 外注: 細菌

「次回診察日」をクリックすると、エディタに展開されている予約日時を引き継ぎます。引継いだ場合、背景色が変わります。

日付にカーソルをあわせると、ツールチップが表示されます。

【再診予約・検体検査】 2011/03/25(金) 10:00-11:00
依頼 (01時: 2011/03/25(金) 10:55 医師: 医師 (ほか)
作成: 2011/03/25(金) 10:55 作成者: 医師: 医師 (ほか)
診療グループ 外科
予約時名称 医師 (ほか)

＜イシメント発生予測の例③＞

患者掲示板が1年で消去されることを知らずに、記載を行っていた。他科、他職種への連絡事項、患者情報が消えてしまう場合。

- 患者掲示板の記載画面に掲載期間のコメントを追加する

○患者掲示板の記載入力(新規)画面に掲載期間についてコメントを追加します。

Before

記載入力(新規)

記録 2015/07/14 09:18 システム管 本番システム 富士通

受付一覧への表示
☒ 医師への伝達
☒ 看護師への伝達
☐ 院長への伝達
☒ メッセージ有効期限 ▼ 2015/07/28

記録 2015/07/14 12:04 システム管 本番システム 富士通

病棟マップへの表示
☒ 医師への伝達
☒ 看護師への伝達
☐ コメディカルへの伝達
☒ 受付一覧への表示
☒ 医師への伝達
☒ 看護師への伝達
☐ 院長への伝達
☒ メッセージ有効期限 ▼ 2015/07/28

表示欄へ追加 表示欄とカルテ両方へ追加 閉じる

After

記載入力(新規)

記録 2015/07/14 11:58 医師 第二内科 てすと医師

受付一覧への表示
☒ 医師への伝達
☒ 看護師への伝達
☐ 院長への伝達
☒ メッセージ有効期限 ▼ 2015/07/28

てすと医師
への伝達
中 大 小 色

※「メッセージ有効期限」を設定しない場合、有効期限は1年間に設定されます。
 表示欄へ追加 表示欄とカルテ両方へ追加 閉じる

※「メッセージ有効期限」を設定しない場合、有効期限は1年間に設定されます。
 表示欄へ追加 表示欄とカルテ両方へ追加 閉じる

また、最近では患者情報がサマライズされている患者プロフィールに入力されている情報を「正」として、オーダ時に入力された情報がそれら患者プロフィール情報と齟齬がある場合は、チェックロジックが働き、オーダ発行を中止させるという動きになっている（図6、7参照）。いかに医療情報システムが医療安全として重要なものであるかを再認識する動きとして注目している。

- CT/MR(造影)検査依頼オーダを確定する際に、造影剤アレルギーが入力されており、かつオーダ内「造影剤の副作用歴」の「無」が選択された場合、オーダ発行不可のメッセージを出力する。



図6 造影剤アレルギーチェックの動き

- MR(単純/造影)検査依頼オーダを確定する際に、体内埋め込みが入力されており、かつオーダ内「体内金属」の「無」が選択された場合、オーダ発行不可のメッセージを出力する。

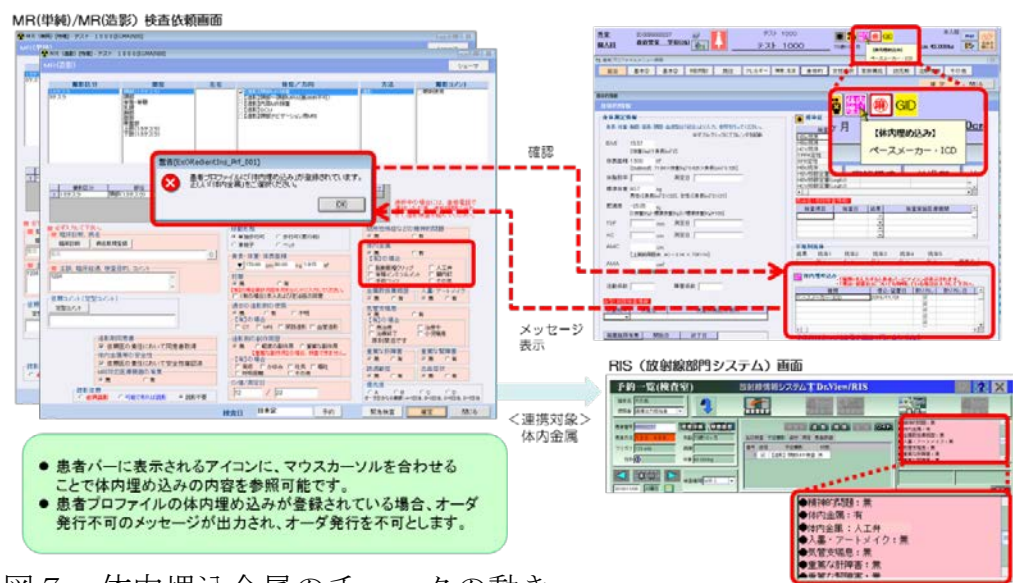


図7 体内埋込金属のチェックの動き

2-1-6 各種医療機器の接続と紙文書の電子スキャン

医療情報システムの中心をつかさどるのは電子カルテであるが、医師はそこへ診療情報を記載しており、各種オーダ情報、文書情報も存在する。よって、電子カルテ化が進むということはそこに診療情報が集約しなければならず、別に存在している、同意書、紹介状、検査レポートなどについては電子化する要望が多く存在した。そのため、紙文書のスキャンングを行い、電子カルテへ貼り付けるためのシステム運用を構築した。また、紙文書と同様に近年では急増している他病院から持ち込まれた CT、MR などの画像情報を取り込む試みを行った。他院から持ち込まれた画像は CD に入っていることが多くあることから、それらをスキャンセンターにて診察前に読み込み、その結果を電子カルテへ設定（詳細に言えば PACS であるが）することとした（図 8 参照）。

さらに、中央検査部ではなく、各診療科においても医療機器を保持しているが、従前は、超音波装置や内視鏡装置の結果は紙によりプリントした結果を紙カルテへ貼り付けるなどしていた。しかし、入院、外来ともに電子カルテが「正」の診療録となっていることから、電子化を進める動きが活発となり、各診療科における自科検査機器と医療情報システムを接続し、検査結果画像の取り込みを行った。

例えば、各診療科で保有している超音波装置、内視鏡装置、赤外線眼振記録装置、顕微鏡装置による検査結果画像などがあるが、それ専用のサーバに保存しており、他科からも医療情報システムを通じて検査画像結果の閲覧を可能とした（図 8-2）。

このように、医療情報システムの利用範囲は急激に広まっていることから、さらなる電子化を進める動きが活発になることは間違いない。

以下にスキャンセンターにおける紙文書、他病院から持ち込まれた画像の取り込みフローを示す（図 9～12 参照）。また、外来におけるカルテ電子化までの道のり概念図、自科検査取込み例を以下に示す（図 8～8-2 参照）。

(参考) 外来電子カルテ化までの道のり概念図

➤ 外来電子カルテ化までの道のり概念図を以下に示す。

道のり概念図

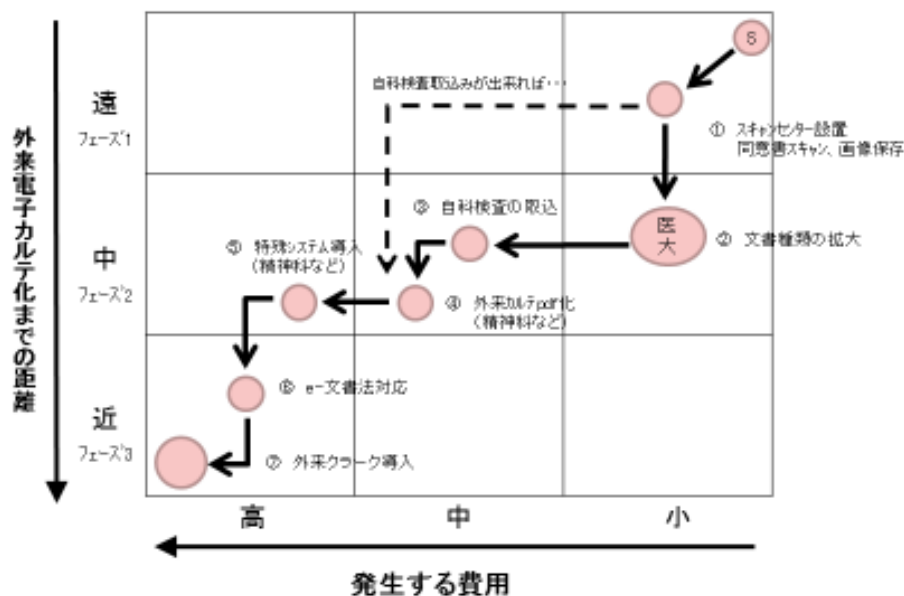


図8 外来カルテ電子化までの概念図

診療現場において、電子カルテ画面から検査結果取込画面を起動し、検査結果を取込みます。

超音波診断装置・内視鏡検査装置の検査結果の取込み例



図8-2 現場負担を最小限にした自科検査結果取込み例

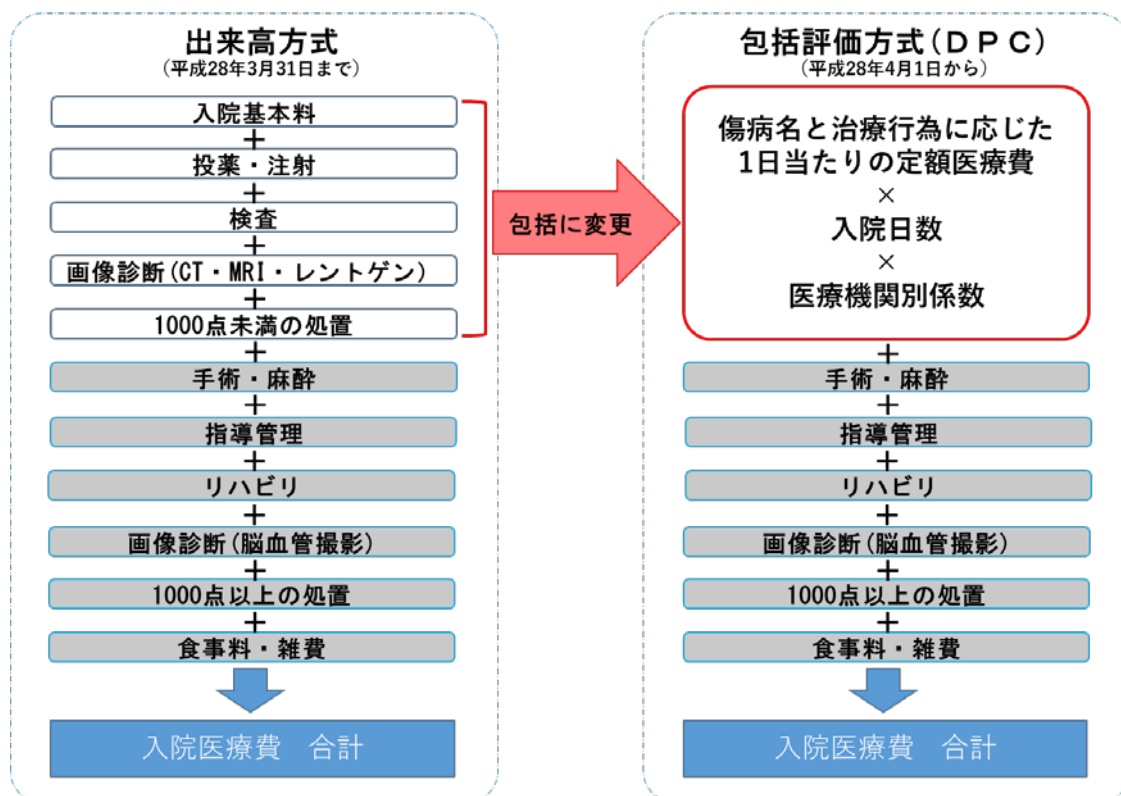


図8-3 DPC（入院医療費の包括評価）の計算

<同意書スキヤンの運用フロー>

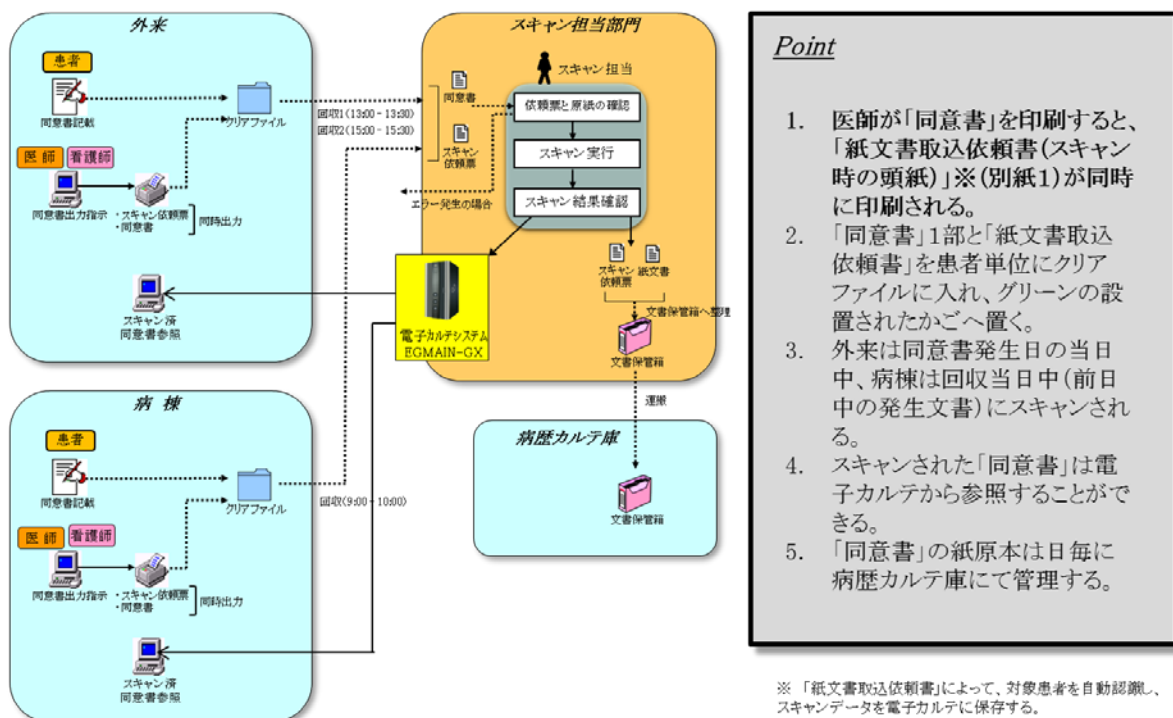
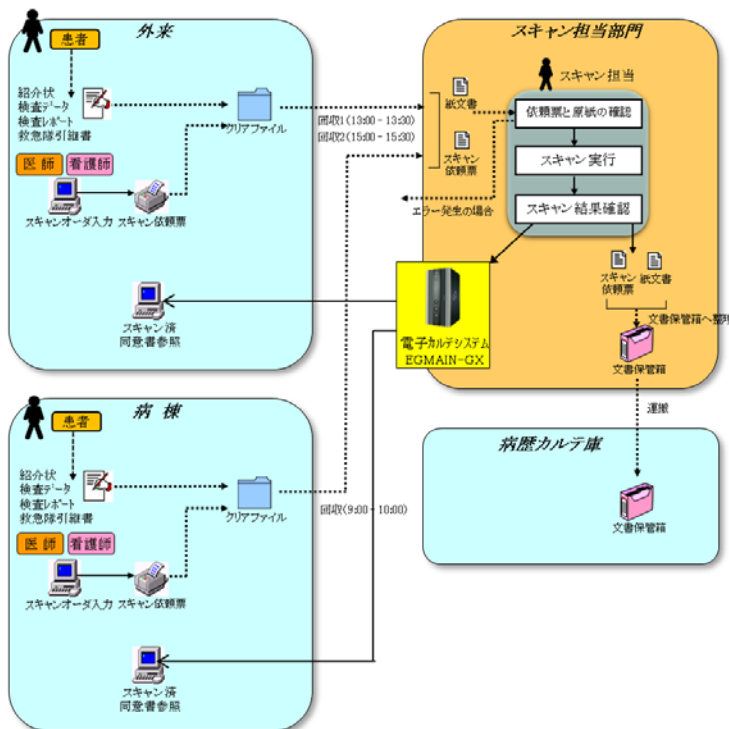


図9 同意書スキヤンの運用フロー

<持ち込み文書スキャンの運用フロー>



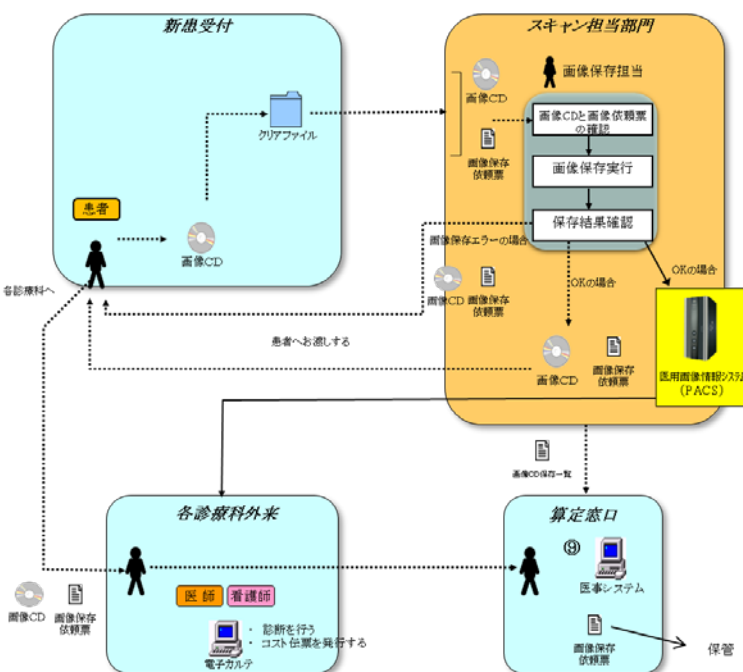
Point

1. 「持ち込み文書」が発生した場合、文書の受け取り者がスキャンオーダーを発行する※。
2. プリンタから「紙文書取込依頼書」が印刷される。
3. 「持ち込み文書」と「紙文書取込依頼書」を患者単位にクリアファイルに入れ、設置されたグリーンのかごへ置く。
4. 外来はスキャンオーダー発生日の当日中、病棟は回収当日中(前日中の発生文書)にスキャンされる。
5. スキャンされた「持ち込み文書」は電子カルテから参照することができる。
6. 「持ち込み文書」紙原本は毎日に病歴カルテ庫にて管理する。

※ スキャンオーダー発行者: 医師、看護師、クラークとする。

図 1 0 持ち込み文書スキャンの運用フロー

<画像CD保存の運用フロー>



Point

1. 画像CDの保存可否結果は、「画像保存依頼票」※(別紙2)に記載される。
2. 保存が成功した場合は、電子カルテのロールブラウザから画像を参照できる。
3. 保存が失敗した場合は、「画像保存依頼票」に理由が記載され、医師は診療科の画像参照用端末にて画像CDを参照し、診断を行う。
4. 画像診断を行った後は、電子カルテの他院画像診断検査タブよりコスト伝票を発行する。
5. 「画像保存依頼票」は患者様にお渡しし、計算センターへ提出してもらう。

図 1 1 画像 CD 保存の運用フロー

スキャン依頼オーダー操作方法

1) 同意書作成によるスキャン依頼オーダー

同意書作成時の保存画面に「スキャン依頼オーダー同時発行」がチェック状態で表示されます。

文書保存画面を閉じると自動的にスキャン依頼オーダーが展開されます。

スキャン依頼オーダー操作方法

2) 他院持ち込み文書に対するスキャン依頼オーダー

持込文書のスキャン依頼はオーダーする必要があります。
ナビゲーションマップ【共通】→スキャン依頼 から スキャン依頼（患者選択）を起動します。

文書名を選択し「追加」ボタンを押下します。
続けて他の文書を追加できます。
フリーコメントは省略可能です。

決定ボタンを押下するとスキャン依頼オーダーが展開されます。

スキャン依頼オーダー操作方法

3) 紙文書取込依頼書の印刷・再印刷

スキャン依頼オーダーを発行すると紙文書取込依頼書が印刷されます。

スキャンしたい文書の上に重ねて患者番号にクリアファイルに入れ、各診療科・病棟に用意してある、グリーンのかごに入れてください。

スキャン依頼書の再印刷

患者のカルテを開き
スキャン依頼オーダーの右クリックメニューから「印刷」を選択します。

他院からの持ち込み画像について

1) 他院持ち込み画像の参照方法

患者のカルテを開くと、「画像入出力」というオーダーが表示されています。
「関連文書あり」をクリックすると画像ビューワが立ち上がり取り込んだ画像を参照することができます。

※他院からの持ち込まれた画像について、うまく読み込めない場合があります。
その際は、これまで同様高品質にありまます画像参照用端末にて画像を参照いただきますようお願い致します。

画像ビューワの検索一覧にて、モダリティ種別が「F」が表示されている検査が他院持ち込み画像となります。

図 1 2 スキャン依頼オーダー発行

2-1-7 医療情報と医療の質

わが国においては、厚生労働省が大規模病院における入院時の診療報酬の計算方法として診断群分類包括評価（DPC）を義務付けており、毎月提出される調査データ（EFDファイルなど）の内容をもとに医療行為の標準化、さらには医療の質の向上までを目指した取り組みを行っている。

良質な医療を効率的に提供していくという観点から、これまで出来高方式で計算していた入院基本料＋投薬・注射＋CT、MRI、レントゲン等の検査＋1000点未満の処置を、各病院機能に応じて3つの医療機関群に分類し、医療機関別係数（＝基礎係数＋暫定調整係数＋機能評価係数Ⅰ＋機能評価係数Ⅱ）を割り出し、傷病名と治療行為に応じた1日あたりの定額医療費×入院日数×医療機関別係数で求めることとした。DPCの導入により、傷病ごとに必要な投薬、検査、処置が平均化され、客観的なデータにより医療の標準化、透明化を進めようとしている。

また、DPCデータは構造化されたデータであるため、全国規模のデータベース構築にも有用であると考えられている（図8-3参照）。

医療情報を利用する際に最も重視すべき点は医療情報の品質維持が重要である。医療情報の品質低下がそのまま医療の質の低下になりうる。一般的に、医療品質の評価は、構造・過程・結果の3つの側面で計測されていることが多く散見される。1995年に日本医療機能評価機構が設立されるなど、日本においても医療の質というものに大きな関心が集まっており、そこで重視されているのは構造に対する評価である。しかし、医療の過程、結果について評価をする方法については、世界的にも難しい問題となっているが、安全かつ効果的な医療を提供するためには、医療の質の向上を図る必要があり、臨床現場においては様々な試みが行われている。その中でクリティカルパスウェイは、診療ガイドラインに沿った標準的な医療行為を実践する上で効果的であり、さらに医療の質の過程、結果の評価をすることが可能となるツールである。高い医療の質を保つためには、専門性の高い医療情報を暗黙知ではなく、形式知として定義し、それを表現する必要がある。多忙な医師が簡易、かつ質の高いクリティカルパスウェイを作成することがキーポイントと考えている。

2-2 医療情報システムの新たな利用形態

医療情報の利用を考えると、大きくは1次利用と2次利用にわけることができる。1次利用とは、収集された各種医療情報を本来の収集目的のために利用することで、患者に直接還元があるものを言い、それに対して、2次利用とはその医療情報を収集する本来の目的以外のために利用することであり、公益のために活用する場合を言う。1次利用の目的は、患者が自ら望んだ診療のためであり、その利用については基本的には患者の同意が得られているものとみなせるが、2次利用については、例外を除き原則として患者の同意については注意が必要である。

1次利用を具体的に言うならば、例えば病院を受診した患者について各種検査を実施、それらの結果から診断を行うこと、それにもとづいて薬剤を投与するなどの治療を施すこと、投薬結果にもとづく検査データの変化から患者の治療度合いを判定することなどである。診療、治療に直結したデータであり、診療目的の利用が行われる。それに対して2次利用では、病院経営のため、各種機関への情報提供、国や自治体の医療政策の立案のため、治験をはじめとする医学研究のため、さらには医学教育のためなど、非常に幅広く利用される。2次利用では、場合によっては全国規模のデータベース構築や、複雑な検索が必要になることもあり、データ格納方式、データの標準化、アーキテクチャ、データ処理技術などが求められ、1次利用以上に標準的なものはないため、今後の大きな課題となっている。

2-2-2 診療情報の整理

医療従事者が医療行為をする場合、患者の電子カルテを開き、電子カルテに記載された医療行為を一つずつ確認しながら新規作成する必要がある。その際、電子カルテに保存された大量の医療行為からクリティカルパスウェイに必要な要素、例えば「処方」「注射」「処置」「手術」等の各オーダ、必要となる期間、タイミングを見つけ出すことは、紙カルテは勿論のこと、電子カルテにおいても非常に時間がかかり、医療従事者にとって大きな負担となっている。

そこで、大学病院における医療情報統合システムを構成する電子カルテシステム、各部門システムなどから、SQLServer2012（マイクロソフト社）をデータベースとして利用し、すべてのオーダ情報、医事会計情報を ETL 経由にて整理統合した「オーダ情報統合テーブル」を作成した。このオーダ情報統合テーブルは、テーブル構成が明確に提示されているため、開発ベンダーが保存内容を把握しているだけではなく、テーブルレイアウトを把握可能とし、そこからレポートやテーブルを作成可能であることがポイントとなっている。

SQLServer2012 を採用した理由として、Reporting Services 機能を利用することで、オーダ情報統合テーブルから容易に Web レポートの作成が可能であることである。医療情報統合システムにおいて、膨大に蓄積された各オーダと医事会計などの情報から作成されたオーダ情報統合テーブルは、クリティカルパスウェイ作成時に必要となる医療行為を全病院横串視点での参照を可能とした。

これらから、病名単位、術式単位に必要な医療行為を抽出するわけだが、まずは共通項、すなわち複数人に実施されている医療行為、期間、タイミングなどを抽出することが重要と考えている。それらの医療行為こそが、医療の標準化、または医療の過程・結果の評価につながると考えている。今回の参照可能なレポートとしては、「病名一覧」「術式一覧」の 2 つのレポートを用意した。

病名一覧では、病名を検索キーとして、前方一致により同一の病名の患者を一覧として表示させることが可能であり、発行された各オーダを入院日から 20 日分表示することを可能としている。また、術式一覧では、術式を検索キーとして、前方一致により同一の手術を行った患者に対して手術 3 日前から 20 日分のオーダを表示することを可能としている。更に患者 ID をクリックすることにより情報をドリルダウンすることが可能であり、該当患者に対するオーダの詳細内容を確認、解析することが可能となる。病名・術式単位に「処方」「注射」「処置」「手術」等の各オーダ、手術の術式、検査項目等を表示することにより、過去に診断した同じ症例の医療情報を参考にすることが可能となる。各オーダから該当する疾患に関してのオーダにおける「共通ルール」を導くことが可能となり、それらの結果をもとにクリティカルパスウェイを作成することが可能となるため、医療従事者、主に医師のクリティカルパスウェイ作成支援、

または医師以外の医療情報担当者がクリティカルパスウェイを作成する際に非常に有益な情報を保持したクリティカルパスウェイ作成支援システムとして利用されている。

2-2-3 診療カルテ記載情報におけるデータアプローチ

昨今のコンピュータシステム利用における大きな潮流である「ビッグデータ」の活用・分析などの重要性は、医療の領域にも徐々に浸透し始めている。電子カルテに代表される医療情報システムを導入している医療機関では、診療に係わる電子データが削除されることなく、日々のデータ量は増加している。これらの膨大な電子データを有益に利用するためには、医療情報システムにおいて電子データを整理した上で保存することが重要である。

医療品質の向上につながるような情報を玉石混淆のビッグデータと呼ばれるレベルの診療データの中から見つけ出していくことは、医療情報担当者（医療情報技師等）が今後取り組むべき最重要な仕事であると考えている。大学病院は数十万人分の膨大な診療データを保持している。それに加えて電子カルテには、医療従事者の記載情報が膨大に存在している。それらを横断的に検索し、見ることが出来れば、記載情報からの「共通ルール」が導き出されるのではない、つまりデータ分析を行い、帰納的アプローチにより法則性が見つかり、その法則性は専門性の高い医療における暗黙知であるかもしれない。しかし、これだけの診療データを検索可能にするにはコンピュータシステムとして考えても技術的に容易ではない。例えば、リレーショナルデータベースに格納されたデータを全文検索するには数日を要することになるであろう。そこで、ビッグデータを効率的に分散処理が行えるオープンソースのソフトウェア基盤である Hadoop を採用し、分散処理を行うことを考えた。Hadoop による分散処理機能を利用することにより、非常に複雑なプログラミングやハードウェア基盤が必要となる分散処理システムを構築しなくとも、所有する約 1 億 3000 万以上のカルテ記載データ（文書）を秒単位に全文検索することを可能とするカルテ記載キーワード検索システムの実現が可能となった（図 1 3 参照）。

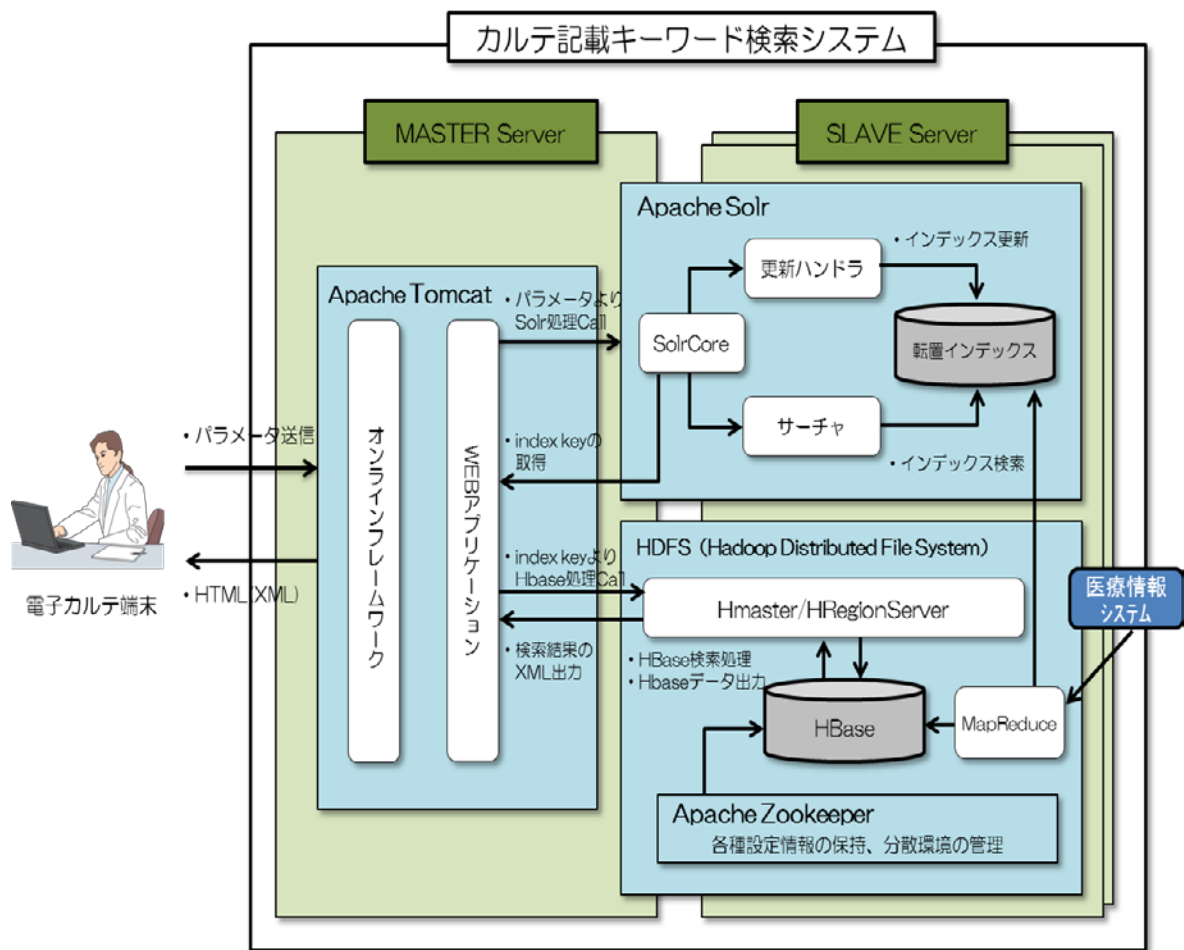


図 1 3 カルテ記載キーワード検索システム

Hadoop を活用した分散処理システムを採用した大きな理由としては、コスト的なメリットである。データ処理速度の大幅な向上のためには、サーバのスケールアップを行うことが必要であり、メインフレーム単位のサーバを用意する必要がある。すなわち億単位の費用が必要となり、容易に実装することは難しい。そこで、オープンソースソフトウェアである Hadoop を中心としたシステム方式の検討を進めた。分散処理システムの主な構成としては、Hadoop-core である HDFS (Hadoop Distributed File System) と MapReduce に加えて、検索エンジンに Apache Solr、各種設定情報の保持、分散環境の同期に Apache ZooKeeper、データベースに Apache HBase を採用し、複合的にオープンソースを利用した。それぞれのツールについての技術的な詳細説明はここでは割愛するが、各種オープンソースを上手く組み合わせることによって、非常に安価に分散処理検索システムを実現することが可能となった。このシステムは、電子カルテシステムのネットワーク環境に配置されているため、既存の電子カルテ

端末上から Web アプリケーションとして起動する事が可能となっている。よって、患者単位ではなく、患者横串にて電子カルテに記載されている内容を検索する事が可能である。

これらのシステムを活用してカルテ記載への全文検索を使ったカルテ記載データの 2 次利用を徐々に始めているが、その中で検索キーとするキーワードの選び方がポイントとなることが判明してきた。医師のカルテへの記載内容によっては、医師ごとにニュアンスの違う言葉が使われていることも少なくないため、漏れなく拾い上げるためには、結果が重複してもいくつかの言葉で検索するか、ある程度の用語の統一が必要になる。さらに、患者からの訴えを拾い上げられるようになることで、新しいカルテ記載データの利用方法の可能性があると考える。例えば、医師の診断においては患者の“胸がぎゅっとする”という表現を“前胸部絞扼感”と医療用語に置き換えて診断推論しているが、カルテ記載をキーワード検索できれば、患者の擬音表現と病気の相関が見出せるだろうし、これはカルテ記載からの新しいアプローチ方法と考える。また、患者の訴えや状況をさらに細かく記載した看護記録も検索範囲としているため、詳細な医療情報の母集団が大きくなれば診断や治療に有用な情報をより高い確度で掘り出すことができ、将来的に他大学病院も含めて全国規模でデータを集めた解析が可能となれば、地域特性がないデータとなり医療全体へのさらなる貢献が期待できる。さらに薬剤単位の副作用検索など、カルテ記載の全文検索を用いたデータの活用法は多岐にわたる。薬剤の副作用については、新たな取り組みとして、ドラッグ・インフォメーション室との連携を進めており、各医薬品の添付文書に記載されている副作用以外の“副作用らしき記載”がないかを当システムを用いて検索を行っている。カルテの記載内容から、新たな副作用の兆しを見出すことで、より質の高い安全な医療を目指している。診療情報ビッグデータの有効活用による臨床、研究へのデータアプローチから「新しい知見」を得ることの実現が、医療における暗黙知を形式知にするために必要な次世代医療情報解析の足がかりになることは間違いない流れとして考えている（図 1 4、1 5 参照）。

記載データの取り扱い

「患者毎」の電子カルテ情報を蓄積し、現場が使いやすいように検索を行う仕組みを提供する。

- ⇒ 過去の記載データについてあまり活用されていない。
- ⇒ カルテ記載がフリーテキストにて入力されているので、データの分析が困難。
- ⇒ SOAP、看護記録、レポート記載、各サマリ等の文字データが多い。

現在考えている記載データの取り扱い

蓄積された文字データを活用し、**新たな価値・発見**を導くことで、現場だけでなく臨床研究で利用できるデータを作成する仕組みを提供できるか？

- ⇒ テキストマイニング技術を利用して、「**患者横串**」で**キーワード検索**を行う等の活用を行う？
- ⇒ 過去のデータを解析し、「**新たな知見**」の**発掘**を行う？

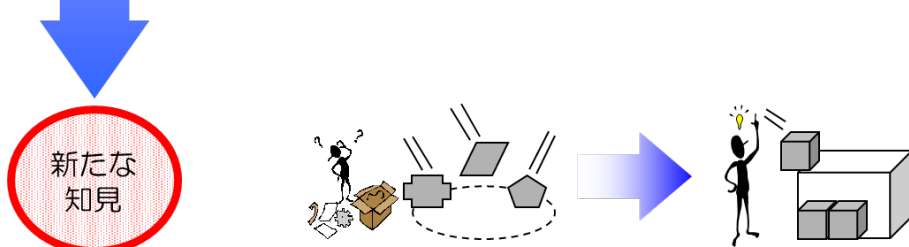
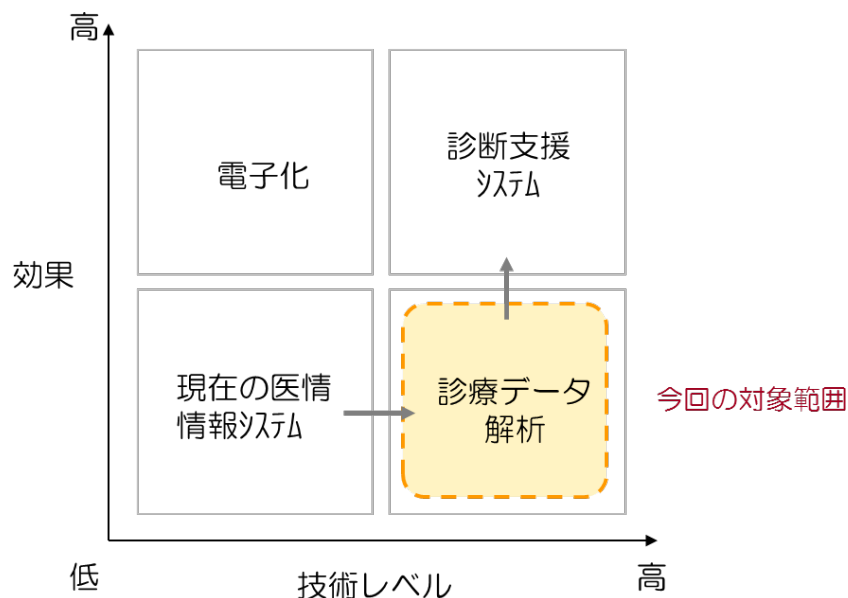


図 1 4 記載データの取り扱い



蓄積された医療者の文字データを解析することにより新たな知識を発見し、それらを診療・研究へ利用する。さらには、診断支援に活用することを構想

図 1 5 診療データ解析の位置づけ

2-2-4 B型肝炎再活性化防止システム

昨今では、HBV キャリアだけではなく、HBV 既往感染者においても、化学療法や免疫抑制療法を行うことによって HBV が再活性化し、劇症化することが知られている。そのため、リスクが想定される患者に対する確実なスクリーニング検査を実施するよう、啓蒙活動および医療情報システムの有効な活用と仕組み作りが重要であると認識している。劇症化を示唆されている薬剤が処方された患者へのスクリーニング検査状況についてのデータ抽出を行い、その結果から大学病院における取り組み状況を調査した。

B型肝炎再活性化防止に関するスクリーニング検査状況を調査した結果、B型肝炎対策ガイドライン（日本肝臓学会）に沿った形での検査が実施されていないケースが散見されたため、医療者への啓蒙活動を進めると共に、医療情報システムへ再活性化防止システムを開発した。主な機能としては、マスタ登録された薬剤をオーダする際に B型肝炎に関連する検査の結果を自動判定し、専用モジュールにより患者の状況に応じた検査オーダの発行が可能な仕組み、さらにはそれらの検査結果から電子カルテへの各種アイコン表示、検査結果緊急通知などを実現する機能を構築した（図 16～23 参照）。また、カルテ記載全文検索システムにより診療録記載から必要情報を収集している。

B型肝炎再活性化防止システムは、B型肝炎の劇症化防止、スクリーニング検査の実施率向上に非常に有用であった。さらにカルテ記載全文検索システムを利用することによって、診療録記載からの独自病名、病態、検査実施等についての精緻な疾患調査を行えている。これらにより申請書類等の作成サポートが可能となっており、さらには医療の質を向上させるための新たな取り組みとして実践している。

今回の B型肝炎再活性化防止システムは、第三世代 AI 技術における知識埋め込み型 AI と言える。ガイドラインが確りと定まっている疾病、処置などに関しては同様にガイドライン(知識)をコンピュータへ埋め込むことによって、CDSS(Clinical Decision Support System)と呼ぶことができ、本システムは全国的にも先進的な事例として取り上げられている。主な機能を以下に示すが、「患者の状況」を患者プロフィールとし、常にここを監視することにより、メッセージやスクリーニング検査項目が可変することは、本研究に深く関わっており、これらが医療の質の向上に貢献出来ていると考えている。

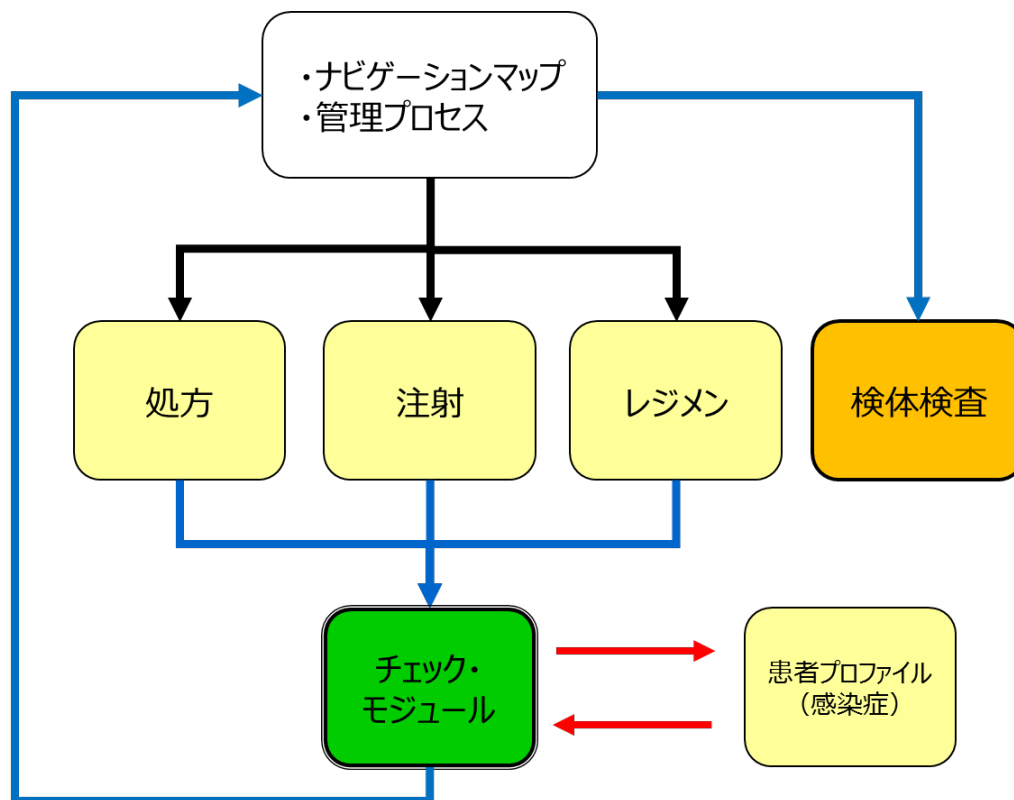


図 1 6 処方オーダーと検体検査オーダーのダイナミックリンクの仕組み

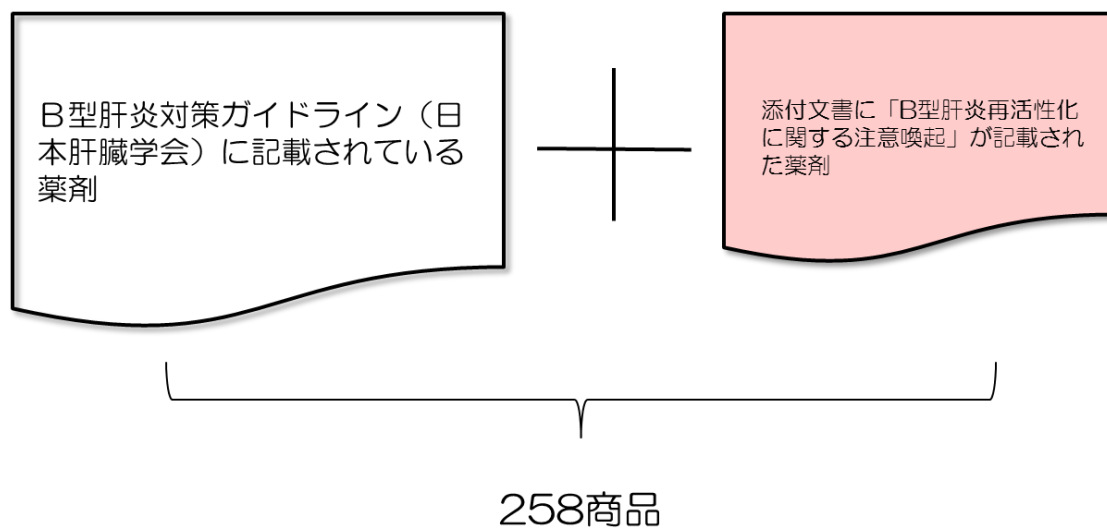


図 1 7 指定薬剤の設定

〈患者プロフィール - 身体的 - 感染症〉

感染症

検査項目	検査日	結果
HBs抗原	2016/05/12	+
HBs抗体	2016/05/12	
HCV抗体	2016/05/12	
TPPA定性	2016/05/12	
RPR定性	2016/05/12	
HBc抗体	2016/05/12	
HBV核酸定量(Log copies)	2016/05/12	
HBV核酸定量(Log IU)	2016/05/12	
HCV核酸定量(Log IU)	2016/05/12	

感染症(他院検査情報)

患者バー

41歳9ヶ月

B肝疾患

【肝疾患陽性反応】
先日の肝炎検査(HBs抗原)にて陽性となっています。速やかに以下について実施してください。
① HBs抗体、HBc抗体の測定を行ってください。
② B型肝炎活性化予防のため、核酸アナログ製剤を投与してください。
③ 日本肝臓学会専門医へコンサルトを行い、指示に従ってください。

肝炎検査結果「HBs抗原」の結果が「+」の場合、患者バーへ「B肝疾患」アイコンを表示します。

マウスポインタをアイコンに近づけると、左記の吹き出しが表示されます。

図 2 0 肝疾患陽性反応患者へのアイコン表示

- HBV核酸定量検査値が2.1以上である患者について通知（オーダ発行医師）を行います。
- 緊急通知受信画面にて、対象の患者ID、氏名、検査依頼日、検査結果値を確認することができます。

HBV核酸定量検査結果が届きました

メッセージアイコンを表示します。

検査結果値が閾値を超えた旨のメッセージと患者ID、患者氏名、検査依頼日、検査結果値を表示します。

緊急通知受信

HBV核酸定量検査値が2.1Log copies/mL (1.3Log IU/mL)以上になりました。肝臓外来を受診して下さい。

患者ID: 09001217
患者氏名: テスト
検査依頼日: 2016/06/20
HBV核酸定量検査値(Log copies/mL): 2.1
HBV核酸定量検査値(Log IU/mL): 1.3

図 2 1 HBV 核酸定量検査の結果通知機能

- 〈HBV核酸定量検査患者一覧〉

図 2 2 HBV 核酸定量検査患者一覧

図 2 3 レセプトコメントへの対応

2-3 クラウドコンピューティングの活用

近年、クラウドコンピューティングを利用したサービスの提供が注目されており、サービス利用者はインターネットを経由して様々なサービスを利用することが可能になり、それにより開発コストの削減、サーバ維持管理が不要、アクセスが容易などの利点があげられる。クラウドコンピューティングのサービスは3つに分類される：SaaS (Software as a Service)、PaaS (Platform as a Service)、IaaS (Infrastructure as a Service)。

一つ目の SaaS はエンドユーザを対象とし、サービスとしてのソフトウェアを提供する。例えば、メールや文書作成、表計算ソフト等のオフィスソフトを提供するアプリケーション・ソフトウェアの利用では、パソコンにソフトをインストールすることなしに、インターネットにアクセスするだけで利用することができる。よって、作成したデータを自身のパソコンに保存する必要はなくなり、インターネットに接続されているどのパソコンからでもデータにアクセス可能になり、データの保守管理を行う必要はなくなるというメリットが存在する。また、SaaS はユーザインターフェースだけではなく、サービス連携のための API (Application Program Interface) を用意しており、様々な他サービスとの連携を行うことを可能としている。技術的なサービス連携方式としては REST (Representational State Transfer) が存在する。REST は Web の標準通信プロトコルである HTTP を利用してサービスにアクセスするため、ネットワーク通信をサポートしている多くのプログラミング言語にて利用できることがメリットなる。また、そのデータ形式には XML が利用されており、REST を標準技術としてサービス連携が進むことになるだろう。

二つ目の PaaS はアプリケーション開発者を対象とし、サービスとして開発者向けのプラットフォームを提供する。アプリケーションの開発・実行環境を提供することにより、アプリケーション・ソフトウェアを開発することができる。具体的には、ソフトウェア開発におけるプログラム言語の設定やデータベース等を提供する。利用できるプログラミング言語は PaaS ごとに異なるが、例えば Java であればオープンソースの様々なライブラリを利用することができる。Java 言語を使う場合、代表的な開発ツールは Eclipse であり、PaaS で利用する際は Eclipse にプラグインをインストールして活用するケースが多い。

三つ目の IaaS はインフラ開発者を対象とし、サービスとしてのインフラストラクチャーを提供する。ソフトウェア開発を行うための基盤を提供するもので、物理的な CPU、メモリ、ハードディスクなどのハードウェアや OS 等を提供している。ポイントとしては必要な時に必要なリソースを使うことができるため効率性が非常に高いことである。オンプレミスと決定的に違う点はここにある。

クラウドコンピューティング以前においては ASP (Application Service Provider) によって、ネットワーク経由でサービスを提供している事業者のサーバ内にあるアプリケーションを利用することができた。個々のアプリケーションを自社のコンピュータにインストールする必要が無く、ソフトの維持・管理に関わるコスト削減が可能な点においては、クラウドコンピューティングも同様のことが言えるが、両者の大きな違いは、サービスの連携を視野に入れているかどうかである。前述した SaaS の説明にて述べているが、ASP ではサービス・プロバイダーが提供しているアプリケーションをユーザが利用することだけを主眼としているが、クラウドコンピューティングでは、必要なサービスだけをクラウドから選択し、ユーザ自身がクラウドで提供されているサービスを組み合わせることにより、ユーザ自身がサービスの提供者となることを可能としている。考えようによっては ASP がクラウドコンピューティングの登場により SaaS と呼び方を変更しただけとも考えられるが、サービス連携を全く考えていなかった ASP サービスとサービス連携 API を準備している SaaS とでは全くの別物と考えている。

2-3-2 クラウドコンピューティングにおける SOA の展開

クラウドコンピューティングは、サービスを提供するための技術基盤、すなわちプラットフォームである。一方、SOA(Service Oriented Architecture)はソフトウェア構築手法であり、SOA のサービスを提供するプラットフォームをクラウドに移行することにより、企業や個人は様々なサービスを利用することが可能になる。そしてクラウド上の様々なサービスを利用することにより、新たなサービスのニーズが生まれる。サービスを提供する側においては、クラウド上では様々な企業がサービスを提供することが可能であることから、サービス同士の連携を視野に入れたサービスの構築が必要であり、また、新たな顧客ニーズが発生したときに、瞬時に新たなサービスを作り出せるような環境作りが必要である。クラウド環境で提供される SOA を「SOA on Cloud」と呼び、そこで提供されるサービス構築を考える（図 2 4 参照）。

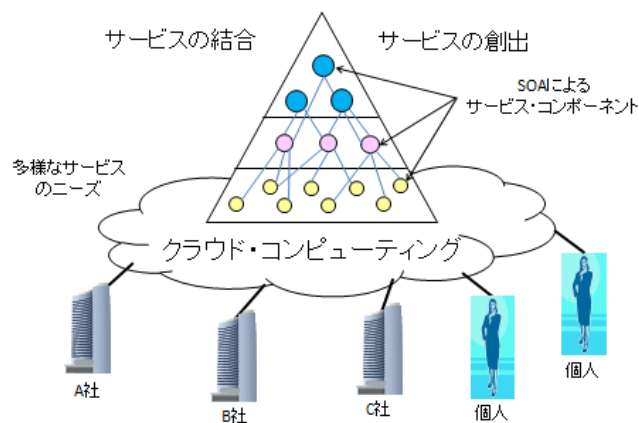


図 2 4 SOA on Cloud によるサービスの構築

第3章 医療情報システムの複雑化と

医療政策における統合化への対応

3-1 医療情報システムの複雑化

現在、医療情報システムは常に改修をすることが必要になっている。例えば、診療報酬改定対応（2年に一度大規模、毎年3月に小規模改修が必要）による改修は必要不可欠であるものだが、前述したように医療安全関係、利便性向上、各種施設基準の変化、診療科など組織改編、チーム医療の増加、医療機器の増加と性能向上、各種医療機器との接続、外部連携、治験関係など、その可変要素は他業界システムと比較しても非常に高いものとなっている。

また、その改修された医療情報システムは、実際の医療現場で、患者の命に関わる要素でもあり、改修によるプログラム・運用の間違い（バグ）や遅延は許されない。このような状況の中、医療情報システムの改修、新機能追加の現状について大手ベンダー各社は理解しているが、根本的な解決、有効なソフトウェア開発手法についてなんら策を出していない状況である。現に開発手法は従前どおりのウォーターフォール・モデルであり、そのため、システム仕様について医療現場との齟齬が度々発生している。

これら様々なソフトウェア改修を行っているための問題も発生している。システムの根幹であるデータベースの正規化、レスポンス、これらを含めた全体のアーキテクチャの整理が行われず、機能を追加することはコンピュータシステムを構築する上で非常に問題である。

わが国における保険医療は社会システムのひとつであり、その情報化は人間の健康と生命に関わることから、きわめて重要である。そのため、以下のことを踏まえた設計が必要である。

(1) インターフェース

多くの職種が関わることとなる医療情報システムでは、特にインターフェース機能が重要となる。画面設計には人間工学をいかした発想を活かしつつ、使いやすさ、チェック機能、紙カルテにはない電子としての利便性を享受されることが必要となる。しかし、現時点の医療情報システムは紙媒体を電子媒体へ単純に変換したと言えるものが多く、電子化であることのメリットを享受できていない。また、各ベンダーによってインターフェースが異なるという点も問題である。

(2) 安定した稼働

よく使われる「24 時間 365 日」の稼働が必須である。医療機関は業務の性質上、24 時間休みなく稼働している。よって、医療情報システムも稼働を休む時間はない。バージョンアップなどの際に 1 時間程度停止することがあるが、現場も医療情報担当も大変になる。よって、ノンストップに稼働することが必須であり、電源設備、空調設備とともにバックアップ設備が必要である。当然、事業継続の観点で医療情報システムのバックアップも重要である。

(3) 電子保存の 3 原則

医療情報システムにおいて、データを電子保存するための条件は、平成 11 年の厚生労働省の通知「診療録等の電子媒体による保存について」にてはじめて示された。その中の 3 原則と言われている、真正性、見読性、保存性についてのシステムの対応が必要となる（図 2 5、2 6 参照）。真正性に関しては、改ざん防止のために履歴管理を詳細に行うなど、要求難度が高い。

(4) データの管理、保存形式

電子カルテになってから、紙カルテや画像フィルムの倉庫管理は行わなくてよくなったが、電子媒体の場合、数十年後に見読性、保存性を保持することが可能であるか検討する必要がある。近い将来、クラウドコンピューティングによる医療情報システムの実現が始まるとは考えているが、部門システムを含めた形での開始は今しばらく時間がかかるものとする。

(5) 標準化

これまでは、ひとつの病院内におけるコンピュータシステムであった医療情報システムは、医療連携などにより他方面との連携が必要になる。よって、柔軟にインターフェースを構築しなければならない。そのために通信規格、データ連携規格、データ格納規格などの様々な形式が定義されており、ここから合致したものを選択する必要がある（図 2 7 参照）。よって、継続性を重視した蓄積データの利用を考えた設計が必要である。

上記 5 つに代表されるように、医療情報システムに求められる設計要件は多く、従来のソフトウェア開発でこれらの要件を満たすことが可能なのか非常に難しい問題である。

これらの厳しい条件を満たすためには、以下の課題が存在する。

(1) データベース設計の基本的問題

データベース構築に関して、データの利用目的に応じて最適なモデルを選択することになるが、数年後も導入当初と同じような運用であることは少ない。多くの定型業務を効率よく処理しつつ、運用によってはシステム変更が発生する。多くの医療情報システムでは、望むような診療情報をなかなか得られないという批判は、データベース設計を起因としたものが多い。データベース設計は一度設計し、構築した後に列項目を挿入することは非常に難しく、さらにはレスポンスを考えると正規化も考慮しなければならない。このように流動的な医療現場では、リレーショナルデータベースではなく、変更しやすいデータベースが望ましい。

(2) 標準化の乱立

疾病名をはじめとした標準マスタや画像規格の DICOM (Digital Imaging and Communication in Medicine) などについては進展があったが、その他のデータ規格、連携規格などについては標準化が進んでいない。現在では、複数の標準規格団体による規格があり、どの規格で統一されるのかそれが検討されている最中である。

(3) 複雑化した医療情報システムのメンテナンス

前述したとおり、部門システムも統合化されている医療情報システムのシステム構造は複雑化しており、すべての機能を把握している技術者は少ない。まず、新たな画面、帳票がほしい場合もデータベースになればデータベースへの追加、変更処理が必要になり、アプリケーションも入れ子処理が多くなり、複雑化している。さらにセッション管理やレスポンス管理をしなければ医療現場にて使える医療情報システムとはならない。

そのため、2年に1回発生する診療報酬改定対応や度重なるシステム改修に対応するためには、大きな労力がかかっている。また、電子化が進んでいるなか、万が一システムトラブルが発生した場合の影響は、範囲も影響度も非常に深刻である。

これらの問題から、データベース、アプリケーションが柔軟に改修できるように根本的な設計方法、改修方法の進化が望まれる。さらに、これらの問題を解決しなければ、医療情報の品質の低下、すなわち医療の質へ悪影響があることがわかる。

3-1-2 従前のソフトウェア開発プロセスと改善の兆し

ICT サービスは、サービスを運用・管理するためのコンピュータ、すなわちコンピュータを動かすためのソフトウェア群から構成され、それらを連携させて最終的に消費者へサービスとして提供されている。ソフトウェア開発プロセスには、様々な方法がこれまで提案されている。

その中で代表的なソフトウェア開発プロセスである、ウォーターフォール・モデルでは、システム開発の全体作業を、要求定義、外部設計、内部設計、プログラミング、テスト、運用・保守、というフェーズ（局面）に分け、この一連の流れに沿ってシステム開発が進められている。このモデルの特徴としては、要求定義から始まる上流工程から運用・保守までの下流工程へと作業を進め、後戻りしないことがあげられる。そのため、ウォーターフォール・モデルで最も重要なのは最初の「要求定義（仕様）」である。この段階で顧客の要求を明確に把握しないと、その後の設計、プログラミングに大きな影響を与えることになるため、最初の要求定義において、顧客とシステム開発者との間で、現状の問題点を調査・分析し、システムに対する要求事項を定義する必要がある。

システム開発の要求定義において、顧客の要望が確定していない場合は、プロトタイプモデルによりシステムの大まかな内容をモデル化し、それをもとにシステム開発を進めていく。このモデルの特徴としては、顧客にシステム開発への参画意識をもたせることにより、要求定義のミスを未然に防ぐことができるが、顧客とのスケジュール調整の困難や開発コストの増大があげられる。

1990年代後半よりシステム開発を取り巻く環境が、大規模システムの開発から、システムの多様化、中小規模のシステムを低コスト、短期間で開発するといった特徴をもつようになってきたことにより、ソフトウェア開発に対して俊敏さが求められるようになってきた。このような状況に対応するものとしてアジャイル開発があり、そのための多様な開発プロセスが提案されている。アジャイル開発の特徴として、システムに対する要求を初期に設定するのではなく、動作しているソフトウェアをもとに要求を明確にしていく、検証は開発の最後に行うのではなく開発と並行して行う、計画を立ててからその通りにプロジェクトを進めるのではなく、プロジェクトを進めながら計画を精密化していく、大勢のメンバが一緒に作業するのではなく、少数のメンバが効率良く作業することを優先することであると述べている。

このようにソフトウェア開発プロセスはシステム開発の規模やユーザがどの程度システムの目的を明確化しているかによって開発手法も変わる。そして、変化の多い医療サービスの開発にはウォーターフォール・モデルが合致しないということは明らかな事実であるが、システム開発会社は引き続きこのモデルを基本として開発を行っている。新しい動きとしては、画面系の開発を最初に

終わらせ、画面構成、画面遷移を顧客と同意することによって開発期間の短縮や俊敏なシステム変更を行えるように試していることが見受けられる。このようにインターフェースを重視した動きも見られる。これらは、システムの提供ベンダー毎に操作性が異なっていることが多いが、医師は異動が多いため、移動した直後ではコンピュータの操作が分からないために診療に集中でいない問題が生じてしまう。また、臨時で外部の医師が診療するような場合にも、コンピュータ操作が分からないための問題が生じやすい。このような画面インターフェースが重要視されており、各ベンダーともに試行錯誤をしている段階にある。

1999年4月22日 厚生省(当時)から都道府県知事への通知

1999年4月22日付けで、厚生省(当時)健康政策局長、医薬安全局長、保険局長の連名による「診療録等の電子媒体による保存について」という文書が各都道府県知事宛に出された。ポイントは下記技術的3条件の基準を満たせば、法律で保存が義務づけられている診療録や診療諸記録を電子媒体により保存してもよいという主旨の通知である。

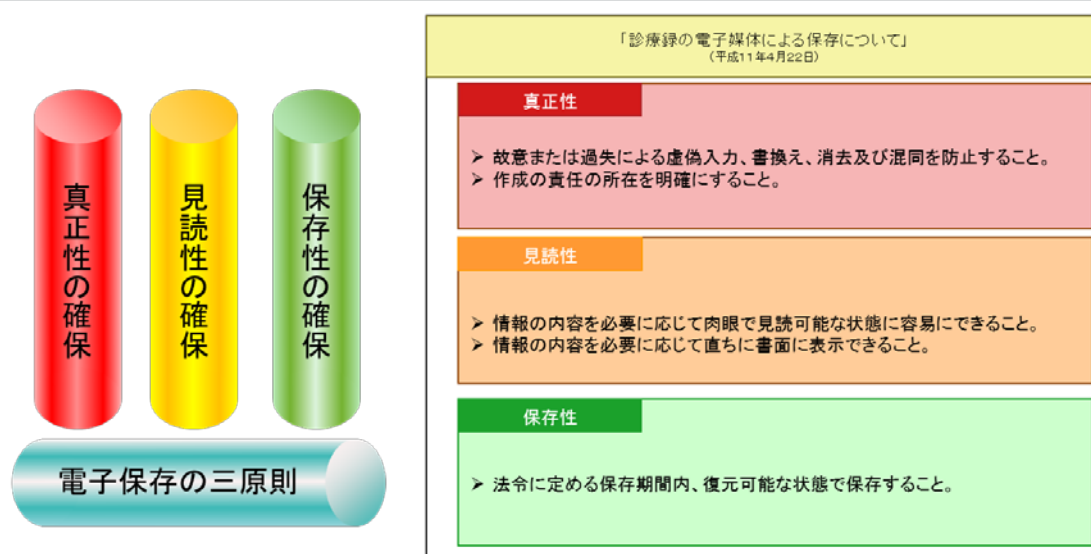


図 2 5 電子カルテの三原則

セキュリティに関する技術的対策と人的対策

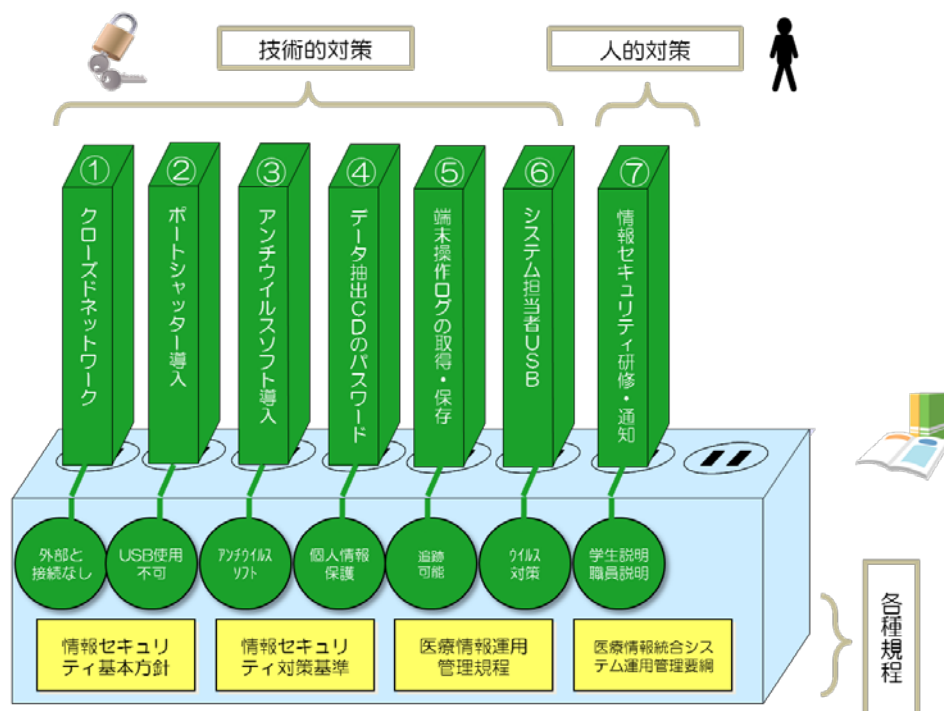


図 2 6 セキュリティに関する対策

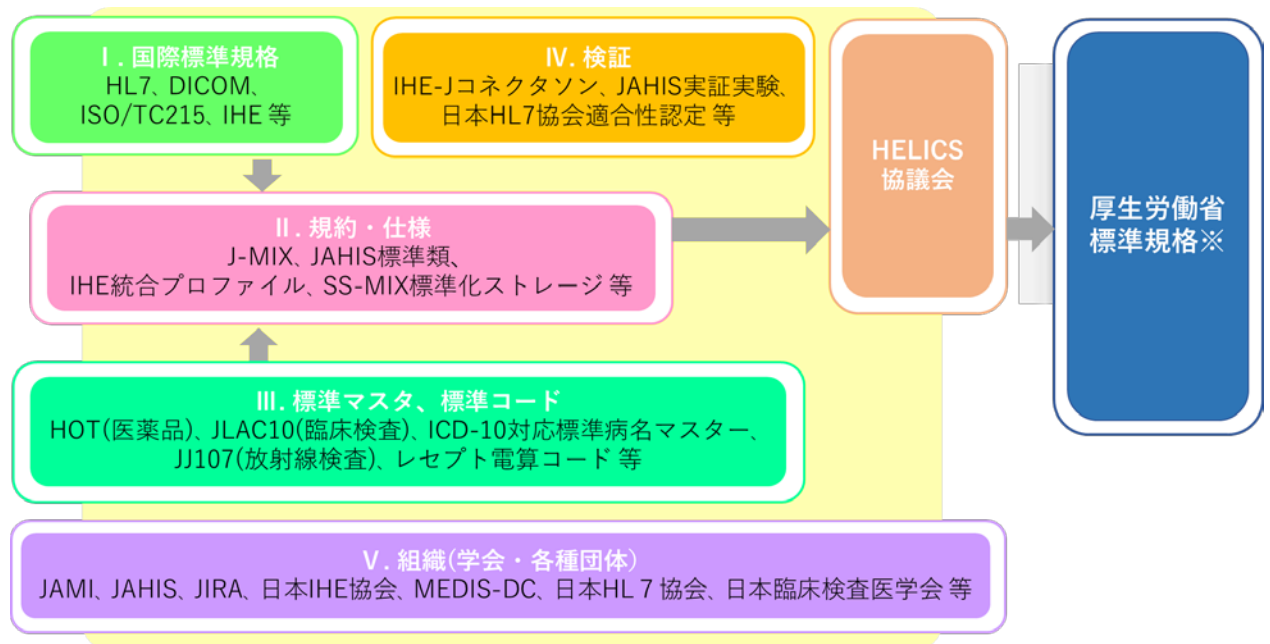


図 2 7 わが国における医療情報システム標準化に向けての活動
(出所：保険医療福祉情報システム工業会)

3-2 医療情報システムの統合化への動き

厚生労働省は、同省内に「データヘルス改革推進本部」を設置し、2017年1月12日に、ICTを活用した次世代型保健医療システムの実現に向けた施策について検討会を開催した。

わが国の健康・医療・介護施策におけるICTの利活用は、様々な縦割り構造を背景に、その前提となるデータが分散し、相互につがらない形で取り組みが進められてきた結果、一体的に機能していないと指摘した上で、膨大な健康・医療・介護のデータを整理し、徹底的に収集・分析して、これからの健康・医療・介護分野のICT利活用が「供給者目線」から「患者・国民、利用者目線」になるようにICTインフラを作り変え、健康・医療・介護施策のパラダイムシフトを実現していかなければならないと提言している。すなわち、「ビッグデータのプラットフォームを構築する」必要があるとしている。

これにかかるシステム構築は、わが国のIT史上でもまれに見る大規模な整備環境であり、健康・医療・介護の分野を有機的に連結したICTインフラを2020年度から本格稼働を目指している（図27-2）。

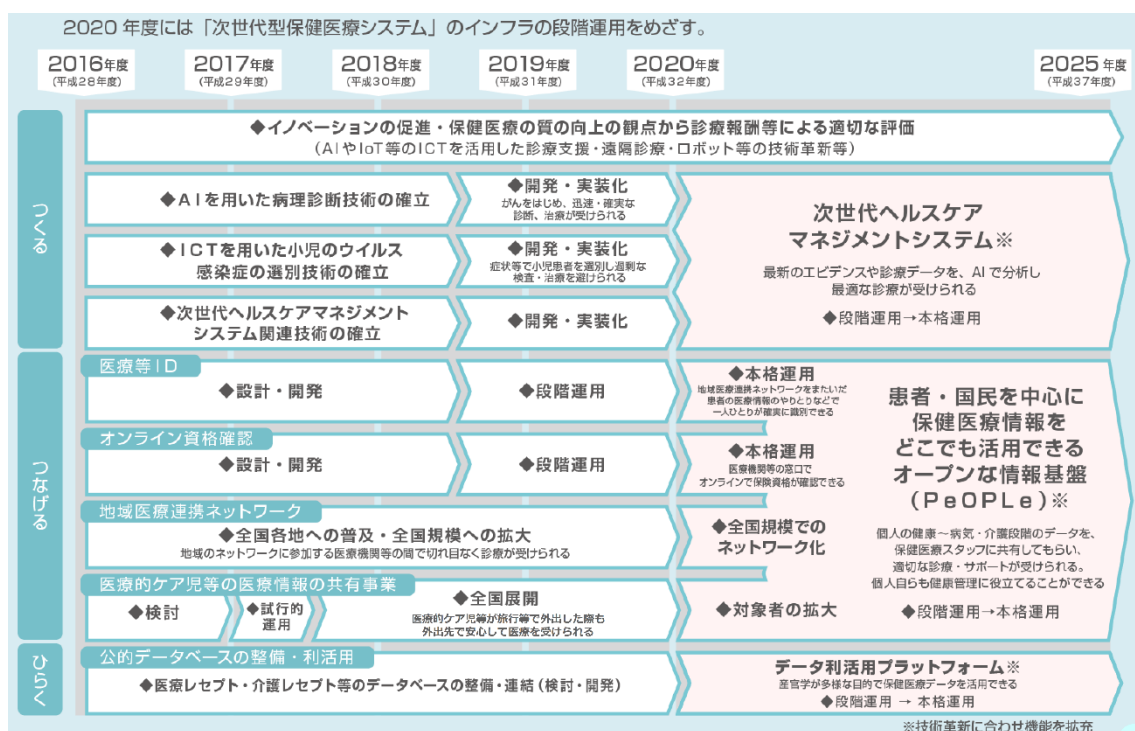


図27-2 「次世代型保健医療システム」の構築へ向けた主なアクション・工程表（出所：厚生労働省）

わが国における医療情報システム標準化に向けての活動を図27に示したように、厚生労働省は、保険医療分野の適切な情報化を進めることを目的として標準規格を制定している。その背景には、ICTを活用した「次世代型保健医療システム」の構築を目指す国策が存在する（図27-2）。

以下、厚生労働省が提言しているICTを活用した「次世代型保健医療システム」の構築に向けた3つのパラダイムシフトとインフラについて示し（図27-3）、項目ごとに説明を加える。

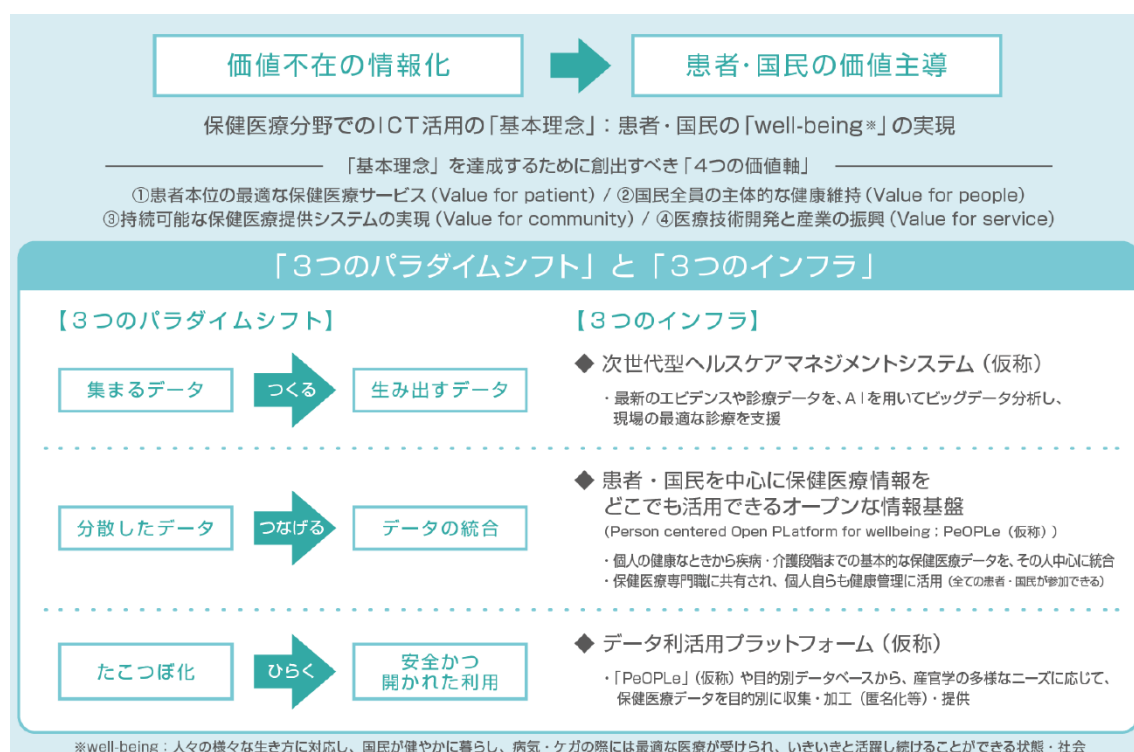


図27-3 ICTを活用した「次世代型保健医療システム」の考え方
（出所：厚生労働省）

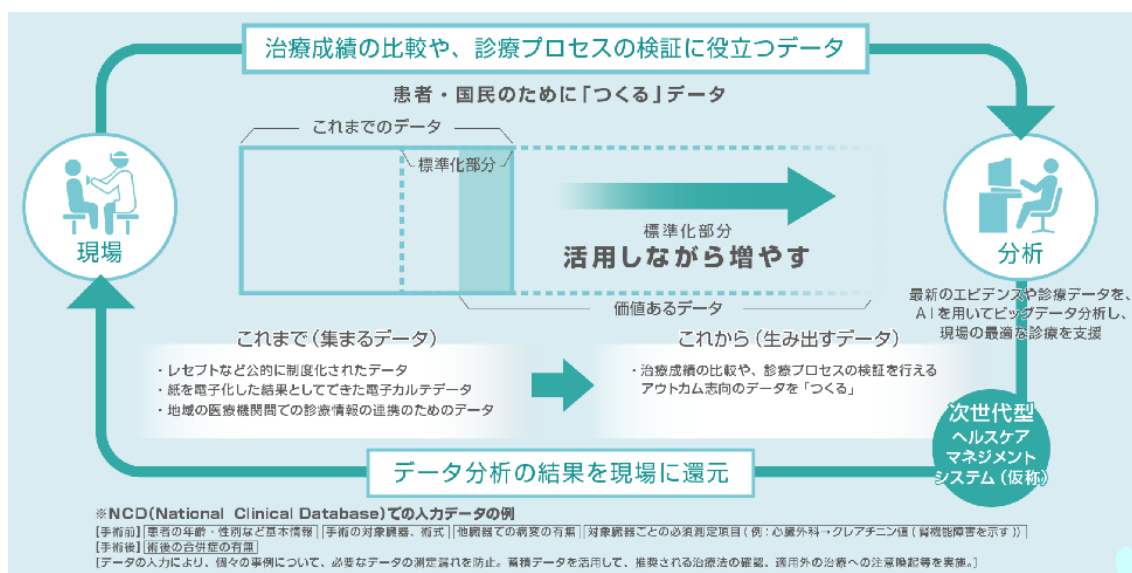
(1) 「つくる」

「集まるデータ」から「生み出すデータ」として活用する。

- データ収集段階から、データの収集・分析・活用（出口）で使えるアウトカム志向のデータを「つくる」。
- データ分析の結果を現場へ還元・活用しながら、現場主導でデータを見直し、増やして行く自律・自走のサイクル（エコシステム）を構築する。

「次世代型ヘルスケアマネジメントシステム」（仮称）を整備する。

- 最新のエビデンスや診療データを、A I を用いてビッグデータ分析し、現場の最適な診療を支援するインフラを整備する。



(出所：厚生労働省)

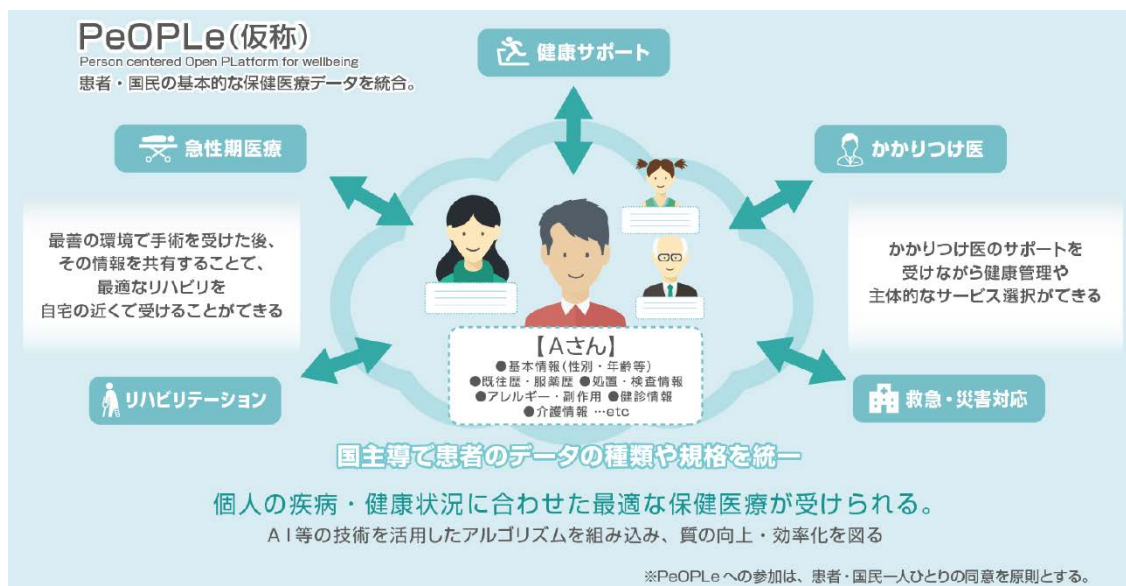
(2) 「つなげる」

「分散したデータ」から「データの統合」を行う。

○個人の健康なときから疾病・介護段階までの基本的な保健医療データを、その人中心に統合する。

「患者・国民を中心に保健医療情報をどこでも活用できるオープンな情報基盤 (PeOPLE)」(仮称)を整備する。

○個々人の保健医療データが保健医療専門職に共有され、個人自らも健康管理に役立てられる、すべての患者・国民が参加できるようインフラ整備する。



(出所：厚生労働省)

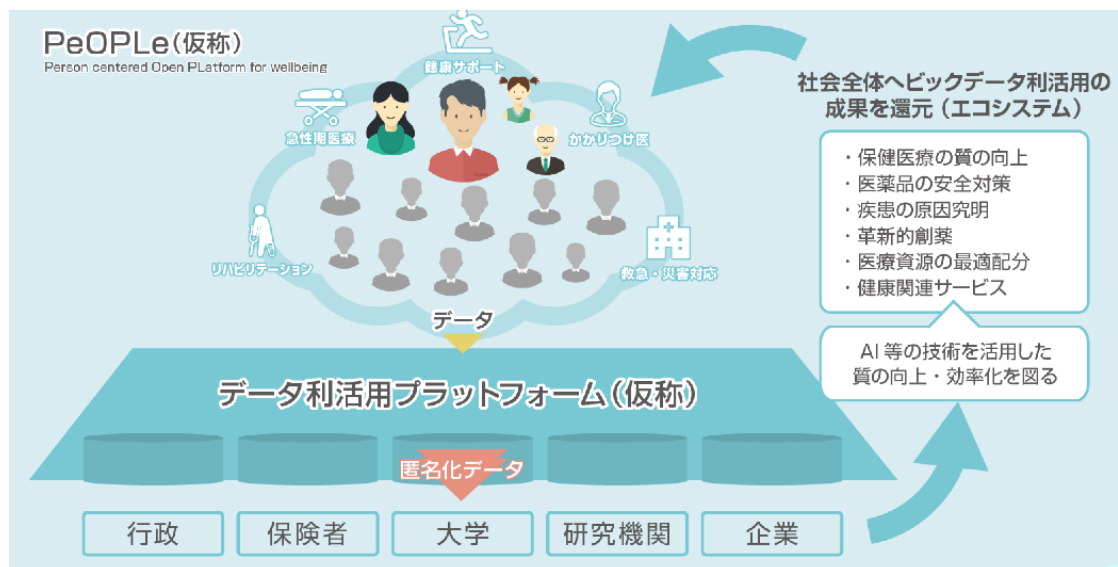
(3) 「ひらく」

「たこつぼ型」から「安全かつ開かれた利用」を行う。

○保健医療分野のデータを「ひらく」ことで、産官学の様々なアクターがデータにアクセスして、医療・介護等の保健医療データをビッグデータとして活用する。

「データ利活用プラットフォーム」（仮称）を整備する。

○「PeOPLe」（仮称）や目的別のデータベースから、産官学の多様なニーズに応じて、保健医療データを目的別に収集・加工（匿名化など）提供できる「データ利活用プラットフォーム」（仮称）をインフラ整備する。



（出所：厚生労働省）

3-2-2 統合化における課題と論点

前述のとおり、厚生労働省は、集められたエビデンスや診療データから、AIを用いてビッグデータを分析し、価値のあるデータを「つくる」こと、個人の疾病、健康状況、介護段階までの基本的な保健医療データを「つなげる」こと、産官学の様々なアクターが、医療・介護などの保健医療データをビッグデータとして「ひらく」ことの3つのキーワードを基に、患者本位の最適な保健医療サービスの提供、国民全体の主体的な健康維持、持続可能な保健医療情報の提供システムの実現、医療技術開発と産業の振興を価値軸とした「次世代型保健医療システム」を構想している（図2-7-4参照）。

しかし、一方では、保健医療の現場、研究機関、産業界等とともに保健医療システムを共創し、その成果が循環するICT施策ガバナンスが必要であることを認識してはいるものの、これまで個々が主体となりオンプレミスでICTシステムを構築してきたため、アクター間の連携方式に互換性がないばかりか、各種データが分散管理されているため、統合化された保健医療システムが確立できていないのが現状である。

保健医療分野においては、全国共通の診療報酬明細書（電子レセプト）があり、年間約20億件のデータ集積があるため、これを足掛かりとしてガバナンス強化を図るのが現実的である。しかし審査基準や審査プロセスは統一されておらず、データ活用プラットフォームの整備には時間を要するであろう。

言い換えれば、わが国における医療情報の統合化に向けたアウトライン構想は提示されているものの、具体的な方法論については見出せていないということが事実であり、大きな課題である。

本研究では、医療情報の共有と統合、診療支援と医療安全、そして医療行為の標準化と医療サービスの平準化を説いており、これらはわが国が見据えている次世代型医療情報システムと同じベクトルである点においても、非常に有用な研究であると言える。

さらに、統合化された医療情報システムはどのような連携方式、データ形式が最適であるかということを考える必要がある。プラットフォーム基盤としては、クラウドコンピューティング上に医療情報を展開することになると考えられるが、その場合、災害時などにおけるBCP（事業継続性計画）も考えなければならない。よって、冗長性を確保したクラウドコンピューティングシステムが必要になる。

これらの次世代型医療情報システムを実現するための方法論について、第4章以降において述べることとする。

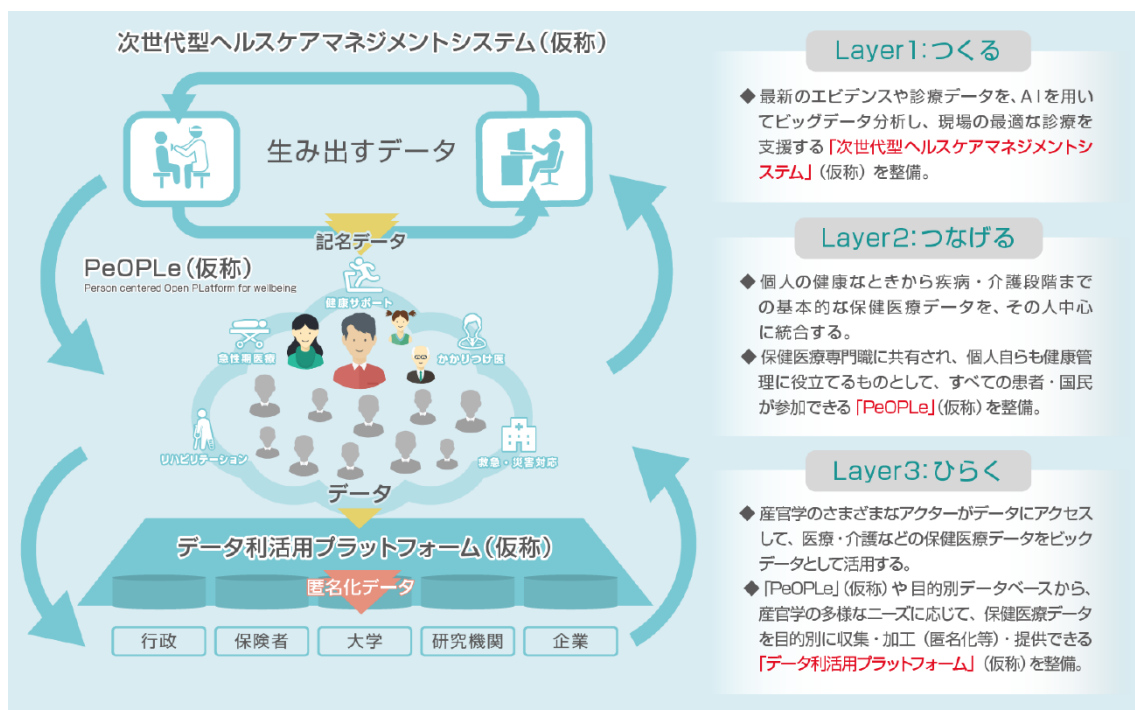


図 2 7-4 ICT を活用した「次世代型保健医療システム」の全体イメージ
(出所：厚生労働省)

第4章 医療行為の標準化とクリティカルパスウェイ

4-1 クリティカルパスウェイによる医療行為の標準化

クリティカルパスウェイとは、疾病単位に治療プロセスや各種検査、医療者のケア、薬剤師・栄養士による指導などについての標準的な経過を定義する、説明するためにまとめたものを示す。クリティカルパスウェイの種類としては、オーバービューパス、オールインワンパス、日めくりパス、スタッフ用パス、患者用パスなどが存在し、これらは適切な場面において利用されている。その中で、オーバービューパスは基本的な形式とされており、横軸に時間軸を取り、縦軸にアウトカムと医療項目、観察、栄養、指導、カンファレンス、チーム医療などといった医師、看護師をはじめとする医療者に関する項目が表記される。本研究にてクリティカルパスウェイと表記されたものは粗方、このオーバービューパスを示している。時間軸は、治療進行によってフェーズ単位に記載されるようになっており、アウトカム（達成されていなければならない患者状態）も日々記載されることが通常である。これらアウトカムが予定通り順調に達成されているかをチェックするが、達成されていない場合はバリエアンスとなる。カレン・ザンダーは、「アウトカムかケア介入で、実際に計画した行動での欠損」と定義している（カレン・ザンダー 1992）が、アウトカムが達成されなかった事象と理解することもできる。このバリエアンス情報も重要なポイントであり、つまり、予測した結果と異なる結果になった場合であるため、予実管理となり、疾患単位に改善すべき点が見えてくることが多いため、この検討結果はクリティカルパスウェイ改訂に反映されることが多い。さらにバリエアンス分析の結果として、クリティカルパスウェイ改訂だけではなく、医療情報システムの改善を行うこともあり、これは医療の質の向上に寄与することになっていると考えられる。

紙ベースのクリティカルパスウェイを使用している医療機関も存在しているが、観察事項や医師記録をオーバービュー形式に盛り込むにはスペース的な問題が多くあるため、大学病院においてはほとんどが電子カルテやオーダリングシステムの中に電子化されたクリティカルパスウェイ機能が搭載されている。電子化されたクリティカルパスウェイは、縦軸を患者単位にカスタムメイドすることが可能であり、そこで表記された医療項目はすなわち各オーダ項目となり、実質的な医療プロセスとして機能している。例えば、疾患単位にクリティカルパスウェイは作成される場合が多く、その際に標準的な治療プロセス、必須となる治療行為を取り込んでおり、これらを適合されと思われる患者へ適

用する。適用した後、その患者ごとのプロフィール情報（ここでは年齢、検査結果など）や疾病の進行度によってオーダー項目を追加、変更、削除を行う。これらは医師が主に実施しており、インフォームド・コンセントは実施されるにしても、その基準は明確にはされていない場合が多く、まさに担当医師の経験、独自の知恵、カンに頼っている部分が多い。

上述のクリティカルパスウェイは医療者用について記載しているが、患者用も存在している。患者用とは、病院の標準的な治療計画を表示したものであり、インフォームド・コンセントに使用することを目的としている。患者が医療を受けるために知っておくべきゴール、アウトカム、治療プロセスが記載されており、患者が治療、ケア計画を理解できるようにした上で、医療者用と同じフォーマットを用いることが望ましい。

このようにクリティカルパスウェイは今後の医療に必須であり、医療の質の向上を考える上で重要なポイントとなっている。

4-1-2 クリティカルパスウェイ支援システム

電子カルテに代表される医療情報システムを導入している医療機関では、診療に係わる電子データが削除されることなく、日々のデータ量は増加の一途をたどっている。これらの膨大な電子データを有益に利用するためには、医療情報システムにおいて電子データを整理した上で保存することが重要である。さらに、整理する方法によっては医療情報の品質維持に大きな影響を与え、医療情報の品質低下は、直接医療の質の低下につながると考える。

一般的に医療の質の評価は、構造・過程・結果の3つの側面で計測されていることが多く散見されており、1995年に日本医療機能評価機構が設立されていることから日本における医療の質への大きな関心が集まっていることは間違いないが、その中で現在重要視されているのは構造（施設基準、医療体制等）に対する評価である。しかし、医療の質は構造・過程・結果の3つの側面による評価を行う必要があり、過程・結果の評価方法の確立が必要である。そのため、臨床現場においては、医療の質の向上に向けた様々な取り組みが実施されており、その中のひとつとして、クリティカルパスウェイの導入が行われている。

大学病院ではクリティカルパスウェイを作成し、実際に患者へ適用することが推奨されているが、電子的クリティカルパスウェイの作成率、適用率ともに非常に高い状況ではない。クリティカルパスウェイは、診療ガイドラインに沿った標準的な医療行為を実践する上で効果的であり、医療の質の向上に貢献が可能なツールである。高い医療の質を保つためには、専門性の高い医療情報を暗黙知ではなく、形式知として定義し、それを適切に表現する必要がある。クリティカルパスウェイの導入とともに医師によって異なる思考や手技を言語化、可視化することによって、医療の標準化がさらに進化すると考えている。そのため、質の高いクリティカルパスウェイの導入をさらに促進させることを目標とし、クリティカルパスウェイにおける作成・検索・適用の各手順を支援するための情報システムを構築している（図28参照）。

これらシステムを利用することにより、実際にクリティカルパスウェイ（ロボット支援腹腔鏡下前立腺全摘除術）の作成を行い、その作成過程からあぶり出された課題を整理するとともに、患者の「状況」を鑑みたかたちのクリティカルパスウェイの作成・検索・適用方法について考えた。

クリティカルパスウェイ作成・検索・適用への支援

- クリティカルパスは作成・検索・適用という手順にて実施されるが、それぞれを支援するためのシステムを構築した。

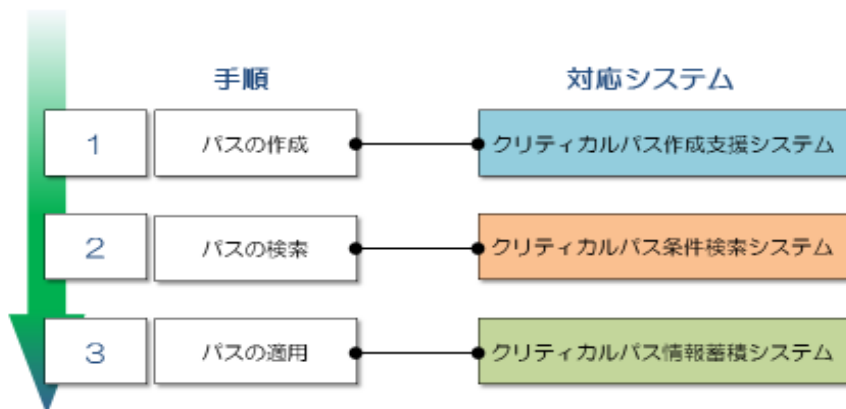


図 2 8 クリティカルパスウェイ作成・検索・適用の全体

4-1-3 クリティカルパスウェイの作成・検索・適用への支援

クリティカルパスウェイの基本的な形式である、オーバービューパスを患者へ利用するためには、医療従事者（主に医師を中心とした医療チーム）がクリティカルパスウェイの作成・検索・適用という手順に沿って進める必要があるが、これらは忙しい医療現場において、じっくりと時間をかけて取り組む余裕がないこと、医療従事者間の手技や診断基準などの暗黙知を形式知として伝達、共有出来ていないことが電子的クリティカルパスウェイの適用率を低下させている大きな要素であることは間違いのないところであると考えている。これらを少しでも解消するため、各手順の電子的支援とはどのようなシステムが必要、効果的であるか、そのシステムを導入することによって、医療品質がより向上することが可能であるか、という点について考えた。

まずは各疾患に合致したクリティカルパスウェイを作成することが必要であるが、このクリティカルパスウェイの作成は、まさに医療従事者の暗黙知を形式知にしなければならず、その疾患に対する実際に治療した経験値、医療知識、それらの具現化が必要であり、最も難易度が高く、しかし最重要なステップである。前述のとおり作成されたクリティカルパスウェイが医療情報システムに保存されていた場合、他医療従事者が探し当てることがクリティカルパスウェイの検索である。ここでは、いかに担当患者の疾患に合致したクリティカルパスウェイを検索することが可能であるか、という検索方法、手順に工夫が必要である。検索されたクリティカルパスウェイを患者へ実際に活用する行為がクリティカルパスウェイを適用することである。このクリティカルパスウェイの適用時にこそ、疾患単位に必要な医学的要素などを見つけることが可能となる瞬間であると考えた。

本システム開発においては、クリティカルパスウェイの作成を支援する「クリティカルパスウェイ作成支援システム」、患者に適合したクリティカルパスウェイの検索条件の設定、検索支援を行う「クリティカルパスウェイ条件検索システム」、クリティカルパスウェイを適用した理由、適用した患者情報を蓄積するための「クリティカルパスウェイ情報蓄積システム」が存在する（図28参照）。これら3つのシステムより、クリティカルパスウェイへの支援が可能となり、さらには患者の「状況」を鑑みたクリティカルパスウェイを検索・適用することが可能になると考えた。

4-1-4 クリティカルパスウェイ作成支援システム (オーダ情報統合テーブルの利用)

医療従事者がクリティカルパスウェイを作成する場合、電子カルテに記載された医療行為を一つずつ確認しながら新規作成する必要がある。その際、クリティカルパスウェイに必要となる要素、例えば「処方」「注射」「処置」「手術」等の各オーダ、必要となるタイミングなどを見つけ出すことは、紙カルテは勿論のこと、電子カルテにおいても非常に時間がかかり、医療従事者にとって大きな負担となっている。そこで、大学病院における医療情報統合システムを構成する電子カルテシステム、各部門システムなどから、SQLServer2012 をデータベースとして利用し、すべてのオーダ情報、医事会計情報を ETL (Extract/Transform/Load) 経由にて整理統合した「オーダ情報統合テーブル」を作成した。このオーダ情報統合テーブルは、テーブル構成が明確に提示されているため、開発ベンダーが保存内容を把握しているだけではなく、病院においてもテーブルレイアウトを把握しており、そこからレポートやテーブルを作成可能であることがポイントとなっている。これらから、病名単位、術式単位に必要となる医療行為を抽出するわけだが、まずは共通項、すなわち複数人に実施されている医療行為、期間、タイミングなどを抽出することが重要と考えた。それらの医療行為こそが、医療の標準化、または医療の過程・結果の評価につながると考えている。参照可能なレポートとしては、「病名一覧」「術式一覧」の2つのレポートを用意した(図29、30参照)。

病名一覧では、病名を検索キーとして、前方一致により同一の病名の患者を一覧として表示させることが可能であり、発行された各オーダを入院日から20日分表示することを可能としている。また、術式一覧では、術式を検索キーとして、前方一致により同一の手術を行った患者に対して手術3日前から20日分の各オーダを表示することを可能としている。更に患者IDをクリックすることにより情報をドリルダウンすることが可能であり、該当患者に対するオーダの詳細内容を確認、解析することが可能となる。病名・術式単位に「処方」「注射」「処置」「手術」等の各オーダ、手術の術式、検査項目等を表示することにより、過去に診断した同じ症例の医療情報を参考にすることが可能となる。各オーダから該当する疾患に関してのオーダにおける「共通ルール」を導くことが可能となり、それらの結果をもとにクリティカルパスウェイを作成することが可能となるため、医療従事者、主に医師のクリティカルパスウェイ作成支援、または医師以外の医療情報担当者などが質の高いクリティカルパスウェイを作成することが可能と考えた。

【病名一覧】

1) 検索条件
 2) レポート表示ボタン
 3) 入院日情報
 4) DPC情報部
 5) オータ情報部
 6) 患者IDリンク

- 1) 検索条件
 次の検索条件を指定することが可能
 ・入院日_検索開始日
 ・入院日_検索終了日
 ・病名(術式一覧画面では術式)
 ・診療科
- 2) レポート表示ボタン
 検索条件に該当するレポートを表示する。
- 3) 入院日情報
 検索条件で指定した入院日を表示する。(術式一覧画面では手術日)
- 4) DPC情報部
 検索条件に該当するDPCの情報を表示する。
 「患者ID」「患者氏名」「年齢」「術式」「医療資源最投入病名」「診療科」「入院日」「退院時転帰」「主治医」「DPCコード」
- 5) オータ情報部
 入院日から20日分のオータ情報を表示します。(術式一覧の場合は手術日の3日前から20日分のオータ情報を表示します)
- 6) 患者IDリンク
 患者IDをクリックすることで、診療部オータ情報レポートを表示する。

※術式一覧も同様の画面レイアウト

図29 クリティカルパスウェイ作成支援システム全体

【診療別オータ情報】

1) 患者情報
 2) オータ分類
 3) オータ情報

- 1) 患者情報
 患者情報を表示する。
- 2) オータ分類
 オータ情報の分類を表示する。
- 3) オータ情報
 遷移元の病名一覧、術式一覧の各レポートに表示されていたオータ情報をさらに詳細に表示する。表示期間は最大20日分を表示する。

図30 クリティカルパスウェイ作成支援システム詳細

4-1-5 クリティカルパスウェイ条件検索システム

作成されたクリティカルパスウェイは、病院単位、診療科単位、個人単位にて保存・管理されるが、これらは同じような疾患の患者へ適用されることも少なくない。医師はこれら作成済みの病院、診療科単位に保存されているクリティカルパスウェイから活用できるものを探すことになる。他医療従事者が作成したクリティカルパスウェイから自身が担当している患者に適したクリティカルパスウェイを見つけるためには病名、術式などをキーワードとして探す必要になるが、本システムでは、患者の状況に応じたクリティカルパスウェイを見つけ出すための施策として、パス単位に検索条件の設定画面を作成した。この設定画面から各クリティカルパスウェイに対して「年齢」、「性別」、「術式」、「病名」をマスタに設定することにより、検索する際の絞り込み条件の設定を可能としている。本機能は、担当している患者の状況（年齢、性別、術式、病名）から、適用すべきクリティカルパスウェイを検索する際に利用する便利機能である。例えば、冠動脈を術式に、病名を狭心症、年齢を 60 歳としてキーワード検索を行うことが可能であり、他クリティカルパスウェイを検索することができる。これらによって、闇雲にクリティカルパスウェイのファイル名をもとに探すのではなく、絞り込み条件により最適なクリティカルパスウェイが検索可能となるシステムを構築した（図 3 1、3 2 参照）。

バス詳細情報登録

バスコード バージョン 治療開始日までの日数 標準適用日数

99 日 10 日

名称 ※全角25文字(半角50文字)まで入力可能

バス難型

コメント ※全角50文字(半角100文字)まで入力可能

有効期間

登録病名 病名選択

コード	病名
1	
2	
3	
4	

プロセスチャート

[A-1:00
デフォルトプロ
適用
10 日間]

検索条件指定する

バス検索条件

年齢 ~ 性別

術式 術式選択

Kコード	手術名称
1	
2	
3	

今回追加したクリティカルバスにヒットする条件

検索条件として病名を指定

横軸編集 確定 閉じる

図 3 1 クリティカルパスウェイ検索条件登録画面

バス・レジメン

バス/マップ/レジメン適用

バス・レジメン一覧 - テスト 1 1 0 0 2 (0000011002)

バス・レジメン選択 新規適用

バス・レジメン選択 リハ 1 5 9 (0090003608)

検索条件 バス レジメン

検索/最新 件数: 0 件 (133)

バス・レジメン名

バス・レジメン選択

検索条件 バス/マップ レジメン 適用除外

共通 科別 分類 病名 検索

科別

- 消化器内科
- 循環器内科
- 第三内科
- 第四内科
- 第一外科
- 第二外科
- 整形外科
- 脳神経外科
- 産科周産期科
- 婦人科
- 小児科
- 眼科
- 皮膚科
- 泌尿器科
- 耳鼻咽喉科

検索条件

年齢 36 性別 男性

術式(手術名称またはKコードのキーワード検索)

前方一致 部分一致 完全一致

冠動脈

病名 病名選択

コード 病名

1	0209	狭心症
---	------	-----

バス・レジメン選択 リハ 1 5 9 (0090003608)

バス・レジメン選択

検索条件 バス/マップ レジメン

共通 科別 分類 病名 検索 閉じる

11111111:テストバス[1]

図 3 2 クリティカルパスウェイ検索画面

4-1-6 クリティカルパスウェイ情報蓄積システム

検索されたクリティカルパスウェイを患者へ適用した際に、その適用した理由、効果、改善点などを記載するための「適用理由入力画面」を構築した（図33参照）。この画面からの入力項目として「年齢」、「性別」、「臨床病名」、「術式」、「フリーコメント」を適用理由として入力可能としている。この機能により、適用したクリティカルパスウェイの採用理由を形式知として明確にすることによって、クリティカルパスウェイ作成における暗黙知の形式知化へ使える情報が医療情報システムへ蓄積されている。入力された情報はロールブラウザへカルテ記載として記録され、システム内部的にはXML形式としてデータベースに格納される。これらは実際の医療現場における貴重な情報であり、適用事例の蓄積を継続的に行うことで、クリティカルパスウェイを構成する知識のデータベースとすることができる。また、カルテ記載として登録されることで、前述しているカルテ記載キーワード検索システムによって検索することが可能となり、クリティカルパスウェイ作成、検索、適用、蓄積のPDCAサイクルを回すことを実現している。

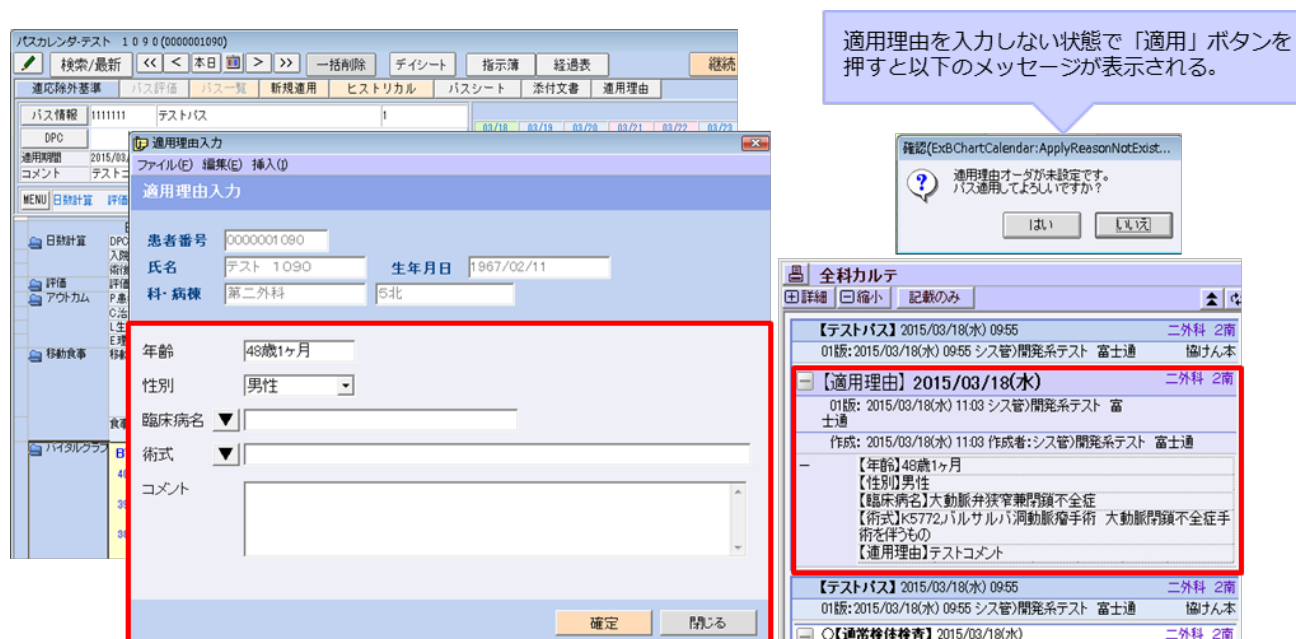


図33 クリティカルパスウェイ適用利用入力画面

4-1-7 ロボット支援腹腔鏡下前立腺全摘除術の クリティカルパスウェイ

これらのクリティカルパスウェイ作成支援システムを活用し、ロボット支援腹腔鏡下前立腺全摘除術に関するクリティカルパスウェイ作成を実施した。

クリティカルパスウェイ作成支援システム（オーダ情報統合テーブル）の術式検索画面を利用し、術式に「ロボット支援腹腔鏡下前立腺全摘除術」、オーダ種に全オーダ、診療科に「泌尿器科」、手術日に「3ヶ月」を設定してレポートを出力した。その結果、前立腺がんの治療の術式に「ロボット支援腹腔鏡下前立腺全摘除術」を行っている数人の患者の全オーダ情報が出力された。しかし、各患者における術前・術後のオーダは処方、処置、注射、検体検査、生理検査、放射線検査といった様々なオーダ種があり、さらに、その数は膨大なため、同類項として考えられるオーダ情報の抽出を行うことは人力では不可能であった。そのため、オーダ種に注射のみを設定し、再度レポートを出力した。その結果、注射ではヴィーンF輸液、バンスポリン静注、アクチット輸液、ロピオン静注が多用されていることが視覚的に理解出来たため、これらがクリティカルパスウェイを作成する上で必要な注射であること、すなわち同類項を把握することが可能となった。つまり、クリティカルパスウェイの作成時に、患者カルテを1つずつ開かなくとも複数のオーダ情報を一覧表示で確認することが可能となり、スクロールしながら概要が確認できる。さらに詳細が見たければ1クリックで具体的に「どのタイミングで」「どのような処方、注射を実施したか」を把握できるようになったことは、これまでクリティカルパスウェイ作成のために多くの時間を費やす必要があった医療従事者の負荷軽減に寄与すると考えている。課題としては前立腺がんの進行度（ステージ）によって術前の注射に大きな差異があり、また、がんの進行度により適用する「注射」の種類も大きく違うことがこれらの情報から把握することができた。

さらに、クリティカルパスウェイ条件検索システムの病名に「前立腺がん」、術式に「ロボット支援腹腔鏡下前立腺全摘除術」を設定し、クリティカルパスウェイの検索を実施した。着目したポイントとして数値情報である年齢があげられる。当初、抽出した患者の平均年齢を検索条件マスタ画面に設定していたが、平均値では適切ではないことがすぐに判明した。そのため、データのばらつき度合いを考えて、標準偏差や分散の考えを用いた年齢を設定した。つまり、平均値としては25歳ではあるが、内容を見るとデータ群Xは平均から5歳のばらつきで、データ群Yは12歳のばらつきが存在した。そのため、95%以上の精度について考えると「平均±標準偏差×2」からの数値データが平均値からそれほど散らばって存在しているかを把握し、その年齢の条件設定を行った。

クリティカルパスウェイ適用画面に「適用理由」項目を追加しているが、こ

これらの情報はクリティカルパスウェイを作成する、検索条件を設定する際の貴重な判断材料として有効となる。

今後、さらに適用理由の情報が蓄積され、貴重なデータベースとなることに注目している。

4-1-8 クリティカルパスウェイ支援システムの効果と課題

非常に暗黙知であることが多い医療行為について、できる限り形式知となっている情報を集約し整理することで院内の医療行為を横断的に参照し、それらをもとにクリティカルパスウェイの作成、検索を支援することが可能な、新しい医療情報システムを構築した。

これらシステムを構築する際のコンセプトは、患者への各種オーダ情報を俯瞰的に見ることにより、帰納法的な推論が可能になることである。また、入力した情報を利活用する仕組みであることから、PDCA サイクルをまわしていることになり、これらは医療の質の向上につながっていると考えている。しかし、医療の現場は常に変化しており、クリティカルパスウェイも例外ではなく、変化への対応が求められるが、現状の仕組みではこれらに対応はできない。今後求められる機能としては、あらかじめ想定された状況に合わせて作られたクリティカルパスウェイだけではなく、「患者の病態が変化した際に、それにあったクリティカルパスウェイを瞬時に提供することが可能な仕組み」が必要になると考えている。

次世代のクリティカルパスウェイ支援システムを考える上で必要不可欠な技術と考えるため、第5章以降にてそのために必要な技術要件について記載する。

第5章 クラウドコンピューティングサービスにおける

サービスロジックの位置づけ

5-1 クラウドコンピューティングサービスの活用

近年、クラウドコンピューティングを利用したサービスの提供が注目されており、サービス利用者はインターネットを経由して様々なサービスを利用することが可能になり、それにより開発コストの削減、サーバ維持管理が不要、アクセスが容易などの利点があげられる。クラウドコンピューティングのサービスは3つに分類される：SaaS (Software as a Service)、PaaS (Platform as a Service)、IaaS (Infrastructure as a Service)。

一つ目の SaaS はエンドユーザを対象とし、サービスとしてのソフトウェアを提供する。例えば、メールや文書作成、表計算ソフト等のオフィスソフトを提供するアプリケーション・ソフトウェアの利用では、パソコンにソフトをインストールすることなしに、インターネットにアクセスするだけで利用することができる。よって、作成したデータを自身のパソコンに保存する必要はなくなり、インターネットに接続されているどのパソコンからでもデータにアクセス可能になり、データの保守管理を行う必要はなくなるというメリットが存在する。また、SaaS はユーザインターフェースだけではなく、サービス連携のための API (Application Program Interface) を用意しており、様々な他サービスとの連携を行うことを可能としている。技術的なサービス連携方式としては REST (Representational State Transfer) が存在する。REST は Web の標準通信プロトコルである HTTP を利用してサービスにアクセスするため、ネットワーク通信をサポートしている多くのプログラミング言語にて利用できることがメリットなる。また、そのデータ形式には XML が利用されており、REST を標準技術としてサービス連携が進むことになるだろう。

二つ目の PaaS はアプリケーション開発者を対象とし、サービスとして開発者向けのプラットフォームを提供する。アプリケーションの開発・実行環境を提供することにより、アプリケーション・ソフトウェアを開発することができる。具体的には、ソフトウェア開発におけるプログラム言語の設定やデータベース等を提供する。利用できるプログラミング言語は PaaS ごとに異なるが、例えば Java であればオープンソースの様々なライブラリを利用することができる。Java 言語を使う場合、代表的な開発ツールは Eclipse であり、PaaS で利用する際は Eclipse にプラグインをインストールして活用するケースが多い。

三つ目の IaaS はインフラ開発者を対象とし、サービスとしてのインフラストラクチャーを提供する。ソフトウェア開発を行うための基盤を提供するもので、物理的な CPU、メモリ、ハードディスクなどのハードウェアや OS 等を提供している。ポイントとしては必要な時に必要なリソースを使うことができるため効率性が非常に高いことである。オンプレミスと決定的に違う点はここにある。

クラウドコンピューティング以前においては ASP (Application Service Provider) によって、ネットワーク経由でサービスを提供している事業者のサーバ内にあるアプリケーションを利用することができた。個々のアプリケーションを自社のコンピュータにインストールする必要が無く、ソフトの維持・管理に関わるコスト削減が可能な点においては、クラウドコンピューティングも同様のことが言えるが、両者の大きな違いは、サービスの連携を視野に入れているかどうかである。前述した SaaS の説明にて述べているが、ASP ではサービス・プロバイダーが提供しているアプリケーションをユーザが利用することだけを主眼としているが、クラウドコンピューティングでは、必要なサービスだけをクラウドから選択し、ユーザ自身がクラウドで提供されているサービスを組み合わせることにより、ユーザ自身がサービスの提供者となることを可能としている。考えようによっては ASP がクラウドコンピューティングの登場により SaaS と呼び方を変更しただけとも考えられるが、サービス連携を全く考えていなかった ASP サービスとサービス連携 API を準備している SaaS とでは全くの別物と考えている。

5-1-2 ICT におけるサービスの定義

今日では、ICT の利用は必要不可欠なものとなっており、ICT を利用した様々なサービスが我々に提供されている。一般的に言われているサービスとは、米国マーケティング協会によると「販売のために提供される、もしくは、財の販売と結びついて提供される諸活動、便益、満足」であると定義されている。また、フィリップ・コトラーは「一方の側が他方の側に提供することができる、本質的に無形であり、何らかの所有権をももたらさない何らかの活動あるいは利益である」と述べている。

一方、ITIL (IT Infrastructure Library) においては、「顧客に対して価値を提供するための手段の 1 つであり、顧客が特定のコストやリスクを負わずに、期待する成果の実現を促進すること」であると定義されている。さらに、「顧客のビジネスを支援するために、顧客と合意した品質水準にて ICT の機能を提供すること」であるとされている。ICT を使ったサービスという概念は定義されており、通常概念におけるサービスではなく、ICT におけるサービスとは、「1 つの業務を実現するために自立している 1 つの情報システムである」と言うことができる。

ICT によるサービスが取り巻く環境の変化を取りあげると、言うまでもなくインターネットは「無保証」のベスト・エフォート型サービスであるため、従来であれば、社会や企業の基盤に近い部分におけるシステムには、専用システムと専用ネットワークが用いられるのが当然と考えられてきたが、現在では医療の分野においてもインターネット対応、オープン化、ウェブ化がものすごいスピードで進行している。また、インターネットを使ったサービスは、サプライ・サイドのインフラの合理化が先行していましたが、現在、IT への関心はサプライ・サイドからデマンド・サイドのシステムに明らかにシフトしており、デマンド・サイドのデータをオープンなネットワークを介してサプライ・サイドにフィードバックすることが重視されている。大きな技術的な変化は、ウェブから「ウェブ・サービス」への変化となる。ウェブとは基本的には情報のリンクとすることができるが、「ウェブ・サービス」とは、アプリケーションの機能（あるいは部品）を自由に組み合わせて柔軟にサービスを生み出す構造となっており、言うなれば、ウェブ上の分散アプリケーション技術と考えても良い。ウェブ・サービスは、XML を中心に SOAP (Simple Object Access Protocol) や UDDI (Universal Description, Discovery and Integration for Business on the Web)、WSDL (Web Service Description Language) といった技術を使って複数のアプリケーション機能（部品）を自由に組み合わせて使えるようにするものであり、ユーザが「サービス・リクエスター」に希望を投げた時点で、リクエスターのサーバが UDDI レジストリーからアプリケーション機能（部品）を

自動的に探しだし、サービス提供者が保有している複数のアプリケーション機能を組み合わせ個々のユーザに最適なパーソナルサービスをつくり出す。これらのすべてのデータ層は XML がベースになっており、XML ベース技術はますます重要になる。

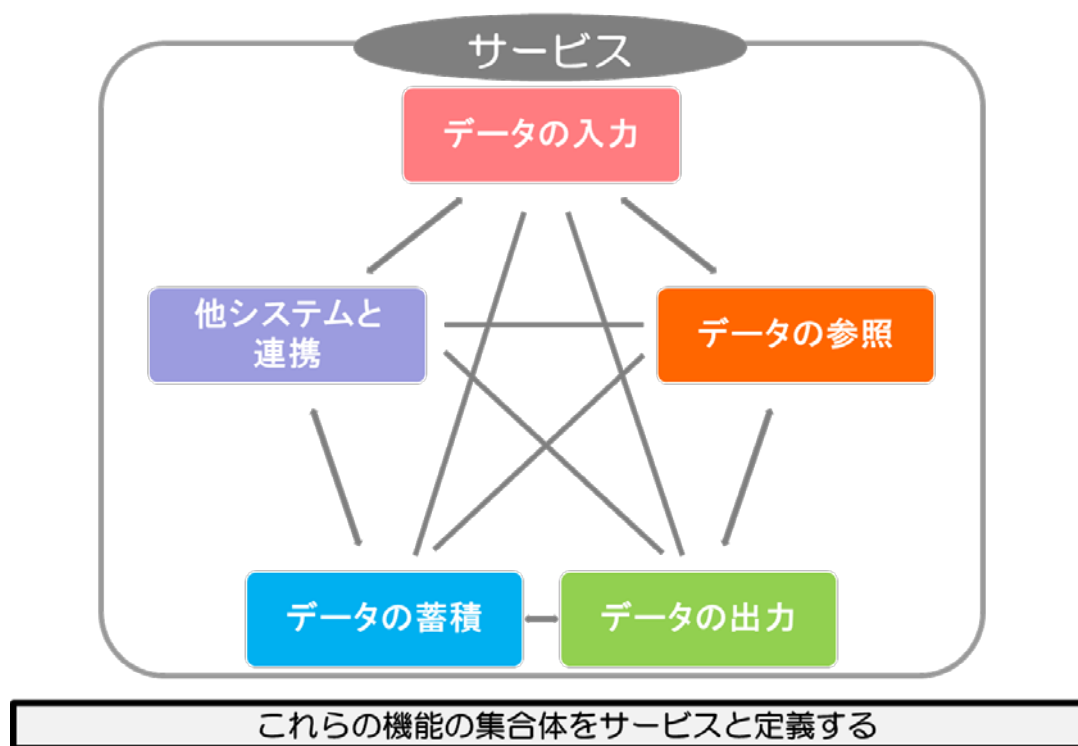


図 3 3-1 サービスの定義

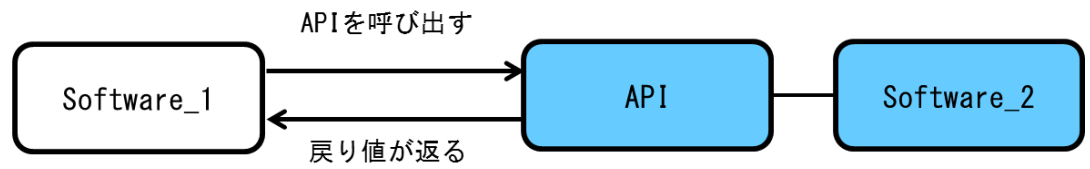
5-1-3 API 連携によるサービスの創出

ソフトウェア構築におけるコンピュータ・プログラムの記述は、ソフトウェア開発者がソフトウェアの持つすべての機能をプログラミングするのではなく、現在ではAPI（Application Program Interface）によって多くのソフトウェアが共通して利用する機能をあらかじめ定義し、それらを必要な時に呼び出して利用することが通常となっている。そのためソフトウェア開発の効率化・コスト削減における一つの重要な手法となっている。

API には様々な種類があるが、例えば Web サイトを開発する時に利用する Web API がある。個人でウェブサイトを作成する場合、データベース機能や検索エンジン、言語解析などを構築するには高度な知識が必要になるが、Web API を利用することにより、必要な機能を Web サイトに取り入れることが可能になる。また、複数の API の組み合わせ、例えば、飲食店の情報を検索できる API と地図情報を表示する API を組み合わせることにより、新たなサービスを提供することが可能になる。複数の Web API を組み合わせて新しいサービスを生み出す開発技法を「マッシュアップ（Mash Up）」と言い、短期間で高機能な Web サイトを構築することができる。多くの API は無償で提供され誰でも自由に使うことができ、当初の API 開発者が想定していなかった新たなサービスが生まれる可能性がある。現在では、スマートフォンのアプリ開発において Web API の利用は必須となっており、これら API 連携による各種サービスの提供により、「API エコノミー」という概念を生み出すという現象にまで発展している。API エコノミーとは、企業や組織が保持している API をオープン化し、それらを活用し合うことによって従前の「ソフトウェアから別のソフトウェア機能を読み出す」という意味から、「ある企業や組織のビジネスから別の企業や組織のビジネスを読み出す」という意味で使われはじめている。呼び出し側から見た場合は、API の先にあるものはソフトウェア機能ではなく、ビジネスそのものであり、ビジネス連携、つまりサービス連携である（図 3 4）。

今後、API エコノミーが普及するためには各組織が保持しているサービスを API 化して、外部に公開（オープン API 化）することが必要となる。しかし、外部に公開するとなると、セキュリティ対策、認証管理、トラフィック管理、課金形態、性能管理などの整備が必要となり課題は多い。また、他技術と同様に連携には仕様の統一が必要である。国際規格 ISO20022 が存在しており、REST をベースとした業界標準策定が実施されている。

従来のAPI



APIエコノミー

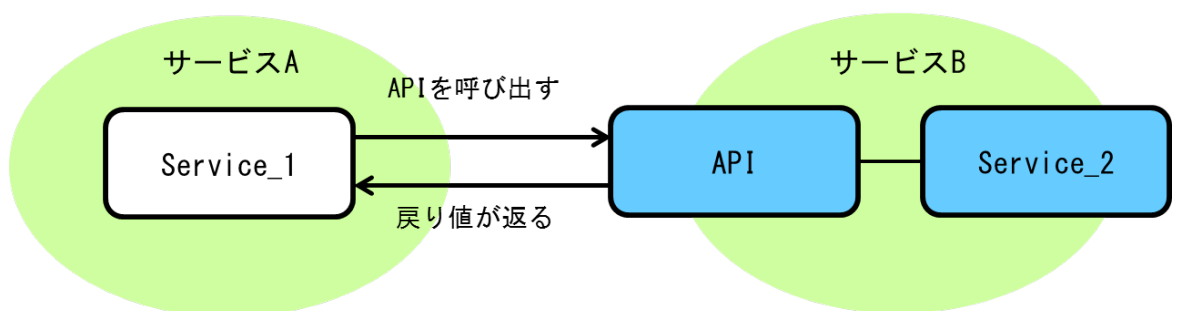


図 3 4 API エコノミー（出所：野村総合研究所）

5-1-4 医療サービスの創出

医療情報システムを構成する各種医療サービスの構成と、それらのサービスのもとになる電子カルテにおける各オーダ、部門システムのあり方を考える必要がある。医療情報システムにおいては、患者単位の様々な医療サービスの連携、その可視化が必須であり、連携される各医療サービス（看護サービス、放射線サービス、検査サービス、病理サービス等）から最終的に主治医が病名診断、治療診断、治療方法を決定する。しかし、各サービスの連携度が進むにつれ、それらの構成要素である医療情報の品質は低下していると考えられる。つまり、医療現場においてインシデント、アクシデントなどの発生原因には、医療情報の品質低下との相関関係があると考ええる。各医療サービスは医療情報によって形作られることから、どのような医療情報を取り揃えるかによってサービスの品質に影響を与え、さらには、扱われる医療情報の品質についても考える必要がある。米国医療施設評価合同委員会（JCAHO）によるインシデントの原因分析によるとコミュニケーション不足、医療トレーニング不足が 50%を占めているという結果がある。よって、患者状況を鑑みた「理解」を伴う情報共有による医療情報の品質の向上も考える必要がある。

5-1-5 サービス連携におけるサービスロジックの位置づけ

複数の Web API を組み合わせて新しいサービスを生み出す「マッシュアップ (Mash Up)」やサービス連携を実現させるためにはシームレスに API、サービスを連携させることが重要となる。そのために様々なアプリケーション統合技術が考案されており、EAI (Enterprise Application Integration) などが存在していた。しかし、これらの統合技術は既存の業務システム間の連携であり、新規システムなどを柔軟にかつ汎用的に統合可能な技術ではなかった。これらは部分最適にはなったが、全体最適となるアーキテクチャではない。特に医療情報システムは、患者の状況、つまり、常に変化する症状に合わせて情報システムを柔軟に対応させることが可能なアーキテクチャを必要とする。本研究ではこれらの要求に合わせて「サービスロジック」を提案している。

サービスロジック基盤は各部門システムをサービス化することで柔軟にサービスを拡張可能とし、それらサービスの組み合わせや各種プロセスから患者の状況に合致したサービスモデル（治療プロセス）を創造し、その場面で最適な医療行為を提供することが可能となるアーキテクチャである。これらの技術はクラウドコンピューティング上において SOA (Service Oriented Architecture) により構築されたサービス同士が連携し、新しいサービスを創造することが可能なアーキテクチャとなる。

第 6 章 サービスロジックを基盤とした SOA の知的構築

6-1 SOA 基盤へのサービスロジックの実装

元来、アーキテクチャとは、建築様式や構造という意味で用いられてきた。SOA(Service Oriented Architecture)とはサービス指向アーキテクチャの略である。ソフトウェア分野におけるアーキテクチャとは、ソフトウェアを構築するための「システム設計手法」を示す。サービス指向アーキテクチャとはビジネスプロセス（例えば、調達、製造、配送、販売等）の構成単位を「サービス」と見なし、それらを組み合わせてソフトウェアを構築する手法である。ここでのサービスとは、例えば注文する、在庫を照会する、といった自立した機能を持つ一つの単位を指す。そして、「製造状況を確認し、配送処理を行う」という一連の業務プロセスをサービス同士の連携により実現する。つまり、それぞれの情報システムをサービス間における連携で自動化する考え方、システム設計手法が SOA と言えるのである。

ソフトウェア構築において、サービスという単位で部品化（コンポーネント化）し、標準的なインターフェースを持たせてネットワーク上に公開（オープン化）することにより、目的に応じてサービスを組み合わせ、システムを柔軟に構築することを可能とし、変化や状況に素早く対応することが可能になると考えている（図 3 5 参照）。よって、大学病院における医療情報システムにおいては、このサービスロジック基盤は各部門システムをサービス化することで、それらのサービスの組み合わせや治療プロセスについて、患者の変化、状況に合致したサービスモデル（治療プロセス）を創造することを可能とし、その場面で最適な医療行為を提供することが可能となるアーキテクチャである。これらの技術はクラウドコンピューティング上において SOA により構築されたサービス同士が連携し、新しいサービスを創造することが可能なアーキテクチャにより実現される。SOA 化された医療情報システムでは、次のようなことが実現可能となり得る。

- あるシステムにおける変更が与える影響を極小化する
- 既存システムと新規システムとの連携を容易にする
- 治療プロセスやルールの変化に柔軟に対応する
- システムや治療プロセスの変更を容易にする

また、SOA におけるサービスの定義としては本稿第 5 章にて述べたサービス

定義に加えて言うならば、既存資産である医療情報システムの自立したある機能を標準的なインターフェースによりアクセスできるソフトウェアモジュールであり、サービスへは、標準的インターフェースを介してのみアクセス可能で、そのシステムがどのような言語、開発環境であるかは意識する必要はない。つまり、サービスはプラットフォームに依存しない定義されたインターフェースにより通信可能なネットワークサービスとなる。この SOA 基盤における既存システム、サービスの連携によって、サービスロジックが実装され、このサービスロジックは状況の変化に対応可能なサービスでもある。

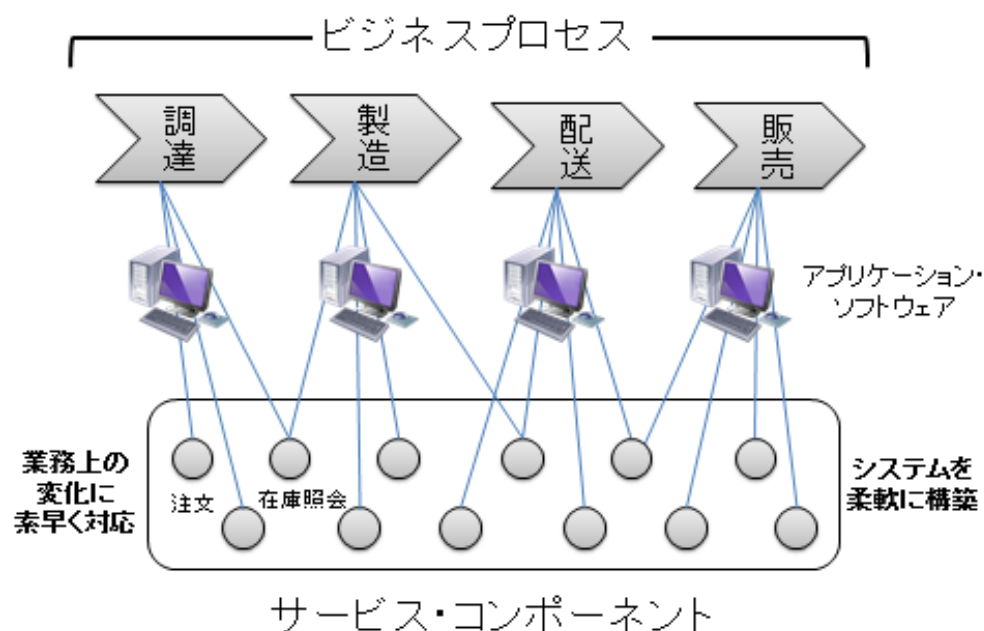


図 3 5 SOA によるソフトウェア開発

6-1-2 SOA におけるプラットフォーム概念モデル

SOA をサービス基盤として実践するために必要な概念モデルを以下のように定義した（表 2 参照）。

- (1) データアクセスレイア
- (2) 医療サービスレイア
- (3) サービスロジックレイア
- (4) プレゼンテーションレイア
- (5) 共通機能レイア
- (6) 開発環境レイア

これらの階層は、すべて Java コントロールおよび Web サービスにてインターフェースを行う。よって、すべての下位層で提供されるサービスは、その先の実装技術がどのような環境で構築されていても、共通のインターフェースを用いてサービスを提供することが可能になると考えている。SOA による実装をするのであれば、共通のインターフェース技術を容易に適用可能なかたちに設計することが必要である。通常、既存システムに接続するためには、接続先のシステムにて公開された API を利用しなければならない。さらに API は、Java 言語、C 言語、VB 言語やネットワークプロトコルなどを利用しなければならない。例えば、Java でデータベースにアクセスするためには JDBC API、IBM MQ に接続するためには MQ API などの専用 API を利用する必要がある。しかし、Java コントロール技術により各 API をラッピングすることにより共通のインターフェースを開発することが可能となり、Web サービスにおいても Java コントロール技術にてラッピングすることにより SOA 共通基盤では Web サービスを自由に連携することが可能となる。つまり、これらの技術によりデータアクセスレイアにおける既存システムや既存サービスへの接続を容易に実現できる。また、プラットフォームの基盤として J2EE(Java 2 Enterprise Edition)アプリケーションを利用する。

SOA に基づいたサービスを実現するには既存システムを連携させるだけでなく、新しいロジックなども SOA 共通基盤上に実装する必要がある。J2EE はこれらを実装する際に各種コンポーネント技術を利用することを可能とする。中心となるサービスロジックレイアにおいては、BPM(Business Process Management)を基本的な考え、仕組みとしている。前述した通り、サービスを実現するために Java コントロール技術を利用するが、BPM ではこれら Java コントロールを実際の医療プロセス（医療情報 SOA）に合わせるかたちで組み合わせることが可能となる。このように医療 SOA を構成する単位を Java コントロ

ールに統一することにより、サービスとして連携させることを可能とする。

表 2 医療情報 SOA 技術レイア

技術レイア	説明	利用テクノロジ	インターフェース
データアクセスレイア	バックエンドシステムと接続するためのアダプタやドライバなどを用いたラッパーサービスを提供する。	JDBC、Javaライブラリ	Javaコントロール Webサービス
医療サービスレイア	医療サービスを提供する。必要となる機能は下位サービスとの連携により実現する。	JDBC、Javaクラスライブラリ	Javaコントロール Webサービス
サービスロジックレイア	サービスロジック（ルールエンジン）をサービスの組み合わせで定義、実行するための実行環境を提供する。	BPM(Business Process Management)	Javaコントロール Webサービス
プレゼンテーションレイア	ポータルサービス中心のWebデスクトップサービスを提供する。画面部品レベルのサービスを保持する。	ポータル	HTTP、Webサービス、WSRP (Web Services for Remote Portlets)
共通機能レイア	ロギング、セキュリティ、例外処理、印刷などの共通サービスを提供する。	Apache XMLBeans、 Apache JavaBeans、 Javaクラスライブラリ	Javaコントロール Webサービス
開発環境レイア	統合開発環境を提供する。	Apache Beehive、 Eclipse	Java、XML

6-1-3 Web サービスと Java コントロール

Web サービスは、XML 形式でデータ（メッセージ）をやり取りし、Web サービスを提供しているサーバに対して、クライアントが関数呼び出し（オペレーション）を行うかたちにてメッセージのやり取りが行われる。そのメッセージにおける標準規格を SOAP プロトコルと呼ぶ。Web サービスには、SOAP のほかにもいくつかの標準仕様が存在し、代表的なものとして、WSDL、UDDI などがある。クライアントアプリケーションは、HTTP を介して、SOAP メッセージをサーバに送信します。メッセージを受け取った Servlet は、実行されるべきオペレーションを識別し、パラメータを Java クラスや EJB に渡してロジックを実行する。最後にクライアントアプリケーションへ応答 SOAP メッセージを返す。その際、Web サービスで受け渡しされるメッセージは XML のため、Java が理解できるように変換または逆変換する必要がある。この変換処理のことを「デシリアライズ」、「シリアライズ」と呼ぶ。

また、Java コントロールは、トランザクション処理、非同期処理、セキュリティなどの様々なサービスを提供可能なコンテナ用のコンポーネントである。Java コントロールとは、その名の通り、リモコンのような存在であり、例えば、EJB コントロールならば、EJB の規約に従った呼び出し方法を知らなくとも EJB コントロールさえ作成すれば対象の EJB にアクセスすることが可能となる。

6-1-4 サービスロジックを基盤とした SOA

「サービスロジック」においては、BPM(Business Process Management)における考えを根幹技術としている。そもそも BPM とは、業務管理手法のひとつで、業務の流れ（プロセス）を単位ごとに分析・整理することによって、問題点を見出し、最適な作業の流れを模索する、という管理手法を言うことが多い。一般的に、業務プロセスとは「一連の細かな業務の流れ（プロセス）が関わりあって形成している流れのこと」であるといえる。例えば、薬剤の調達から製薬、払い出し、患者への使用といった個々の作業がプロセスに当たる。BPM は、この個々のプロセスについて、効率化や統廃合といった改善を実施することとなる。

BPM の特徴として、管理と改善は、一定のサイクルをもって常に行われるものである、という点が挙げられる。医療現場では、患者の状況、状態が常に変化しており、そこで常に最適な治療を考えて、改善を図って行くシステムとして存在させる。しかし、このような管理、改善を紙ベースにて実施するには医療現場の変化に対応できない。よって、前述したクリティカルパスウェイやセット登録（処方、処置、検査などの短い治療プロセスを医師単位に事前に登録しておく）を多くの病院では電子的な仕組みを構築して実施している。これらの医療現場における一連の作業、治療プロセス、クリティカルパスウェイが単純であれば多くの支援は必要ないが、関連するプロセスの数が多くなってくると、複雑さの度合いが増すため、進捗の監視などをワークフローツールによって行うことになる。

前述した通り、サービスを実現するために Java コントロール技術を利用するが、BPM を用いることによって、Java コントロールを実際の医療プロセスに合わせるかたちで組み合わせることが可能となる。このように医療プロセスを構成する単位を Java コントロールに統一することにより、サービスとして連携させることを可能とする。また、BPM では単にプロセスを実行するだけではなく、サービス利用者への通知やモニタリングなどの拡張機能が必要となる。

これらの技術により「サービスロジック」は、システム連携におけるハブとして医療プロセスという視点から既存システム、各サービス、各データベースを活用することができる（図 3 6 参照）。

その際、Java 以外で構築されたシステムと連携を行うプロセスでは、標準で定義されているデータフォーマットである XML を利用する。しかし、XML を扱う手段として、DOM(Document Object Model)や SAX(Simple API for XML)といった XML 用の API が存在するが、より簡易に扱うために Apache XMLBeans を利用することが有効である。Apache XMLBeans とは、XML 操作の API であり、DOM、SAX と比べて、省メモリで高パフォーマンスである。XML スキーマ定義フ

ファイルを Apache XMLBeans のスキーマコンパイラでコンパイルすると自動的に XML を操作するためのクラスが自動生成される。DOM と似ているが Apache XMLBeans ではツリー構造を一度にメモリに展開せずに、データは char 配列というかたちで保持しており、必要な部分だけをメモリ展開を行うことを可能とする。Apache XMLBeans は基底クラスとして XMLObject を保持している。XML 内容を Parse はできないが、同一の XMLObject として扱うことが可能となる。また、Apache XMLBeans はコメントや改行、空白文字など、XML の要素以外も情報を保持するという特徴がある。DOM などで処理を行うと、いったん解析してメモリに取り込まれた XML はオリジナルフォーマットが失われてしまう。例えば、人間が見やすいように改行やコメントをつけて XML ファイルを作成するが、メモリに取り込まれた時点でそれらの情報は失われてしまう。これに対して、Apache XMLBeans ではメモリに取り込んでも、バックエンドに char 配列としてオリジナルの XML 文字列を保持しているため、再び Write など行いファイルに復元するとオリジナルの情報が保持されているため、XML の利点を生かしつつデータ連携を行うことが可能となる。

これらの各種技術をベースとして「サービスロジック」は実現されるわけであるが、では「サービスロジック」とはそもそも何をつかさどるものであるのか。それは、ソフトウェア開発における仕様書を具現化してものである。「このサービスでは何をするのか」ということが記載されていなければならない。さらに、この「サービスロジック」には既存システムやサービスを柔軟に拡張可能とし、それらサービスの組み合わせやプロセスから患者の状況に合致したサービスモデル（治療プロセス）を創造し、その場面で最適な医療行為を提供することが可能となる仕様書となる必要がある。つまり、内部のロジックは変更することが前提であり、状況に応じて素早く新しいサービスを作ることが求められる。

これらを実現するためには、この「サービスロジック」を構築するために、いかに専門的でありかつリアルタイム性のある情報をロジックへ反映することが可能かについて考え、本研究では、XML 言語によるサービスロジックの生成について論じる。

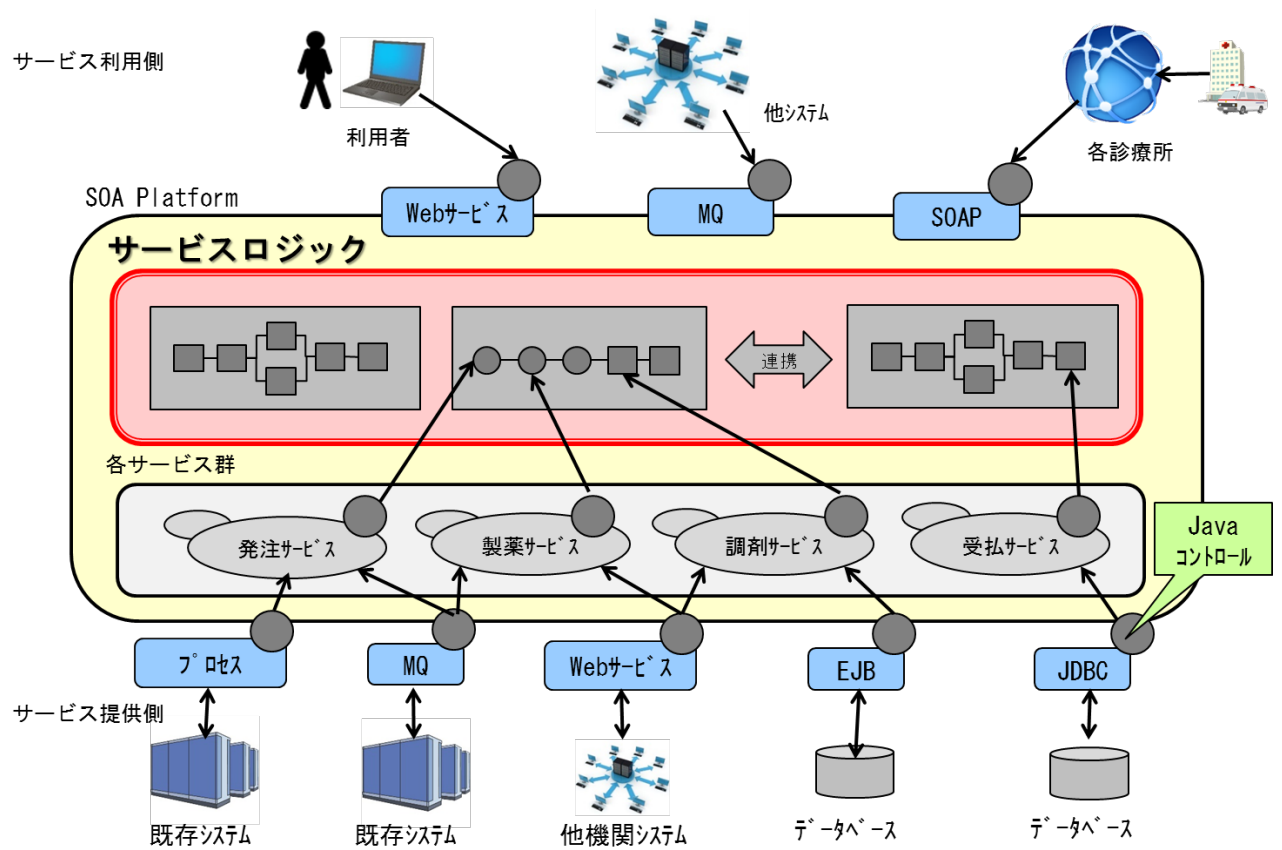


図 3 6 サービスロジックを基盤とした医療情報 SOA

6-1-5 クラウドコンピューティングにおける SOA の展開

クラウドコンピューティングは、サービスを提供するための技術基盤、すなわちプラットフォームである。一方、前述したとおり、SOA(Service Oriented Architecture)はソフトウェア構築手法であり、SOA のサービスを提供するプラットフォームをクラウドに移行することにより、企業や個人は幅広く様々なサービスを利用することが可能になる。そしてクラウド上のサービスを利用することにより、新たなサービスのニーズが生まれる。サービスを提供する側においては、クラウド上では様々な企業、病院がサービスを提供することが可能であることから、サービス同士の連携を視野に入れたサービスの構築が必要であり、また、新たな顧客ニーズが発生したときに、瞬時に新たなサービスを作り出せるような環境作りが必要である。クラウド環境で提供される SOA を「SOA on Cloud」と呼び、そこで提供されるサービスはサービスロジックを基盤とした SOA として展開される（図 3 7 参照）。

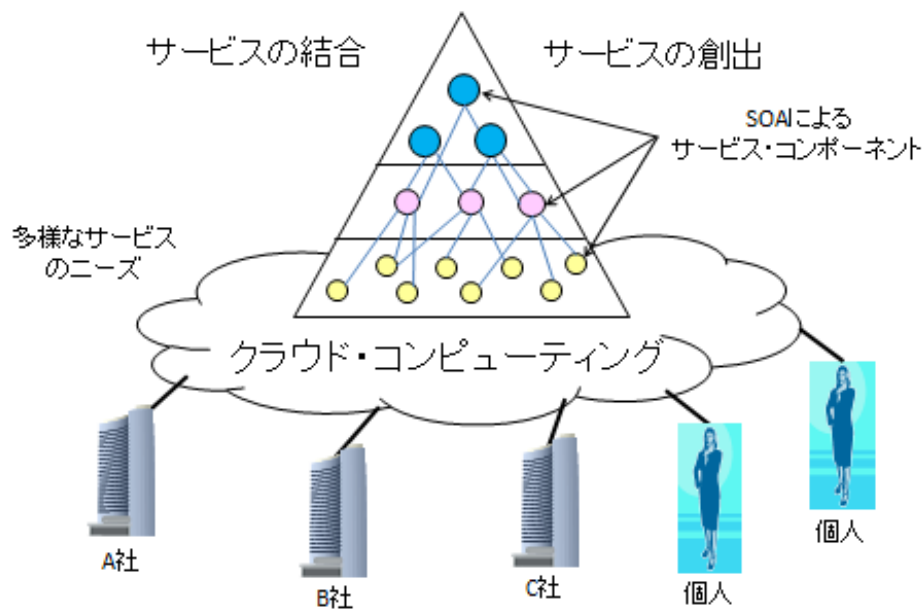


図 3 7 SOA on Cloud によるサービスの構築

6-1-6 医療 SOA におけるクリティカルパスウェイサービス例

サービスロジックを活用した、医療サービスを SOA により構築(医療情報 SOA)するにあたり、一つの例としてクリティカルパスウェイサービスを提示する。医療の現場においてクリティカルパスウェイを導入するには、クリティカルパスウェイの作成、現場での運用、分析、見直しという手順が必要になる。パスの作成においては、まず医療の標準化を図るために医療関係者との間での情報共有が必要不可欠である。そして、パス作成後の成果・目標であるアウトカムを設定する。また、実際に作成したパスをどのような形式で表示するのかというフォーマットについても考える必要がある。医療現場におけるパス運用では、実際に院内チームや他病院との連携されたパスが存在し、かりに達成できなかったアウトカムが発生した場合（バリエアンスという）は、その原因等を記録する必要がある。通常バリエアンスの登録は、あらかじめ予測されるバリエアンス原因を要因別に分類しコード化した、バリエアンス要因コードを使用する。

パスの分析では、パス運用の際に記録されたバリエアンスの内容、バリエアンスが発生したアウトカム、バリエアンス発生数の頻度などを集計し分析を行う。これらの分析結果をもとに、評価のためのチェックシートなどのアセスメントツールを使用して、アウトカムとバリエアンスの評価を行い、パスの見直しを行っていく（図 3 8 参照）。

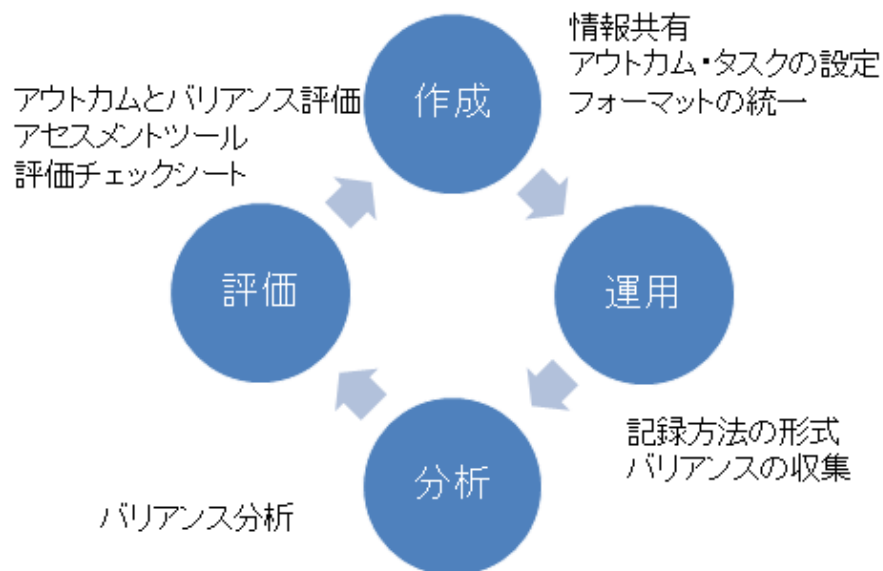


図 3 8 クリティカルパスウェイ導入の流れ

クリティカルパスウェイ作成に必要な情報は、主に病院がこれまで蓄積している過去の情報や、科学的に実証されている文献情報などをもとに作成される。これらの情報を元に作成されるクリティカルパスウェイは、一度作るとその後、変更がないわけではなく、元となる情報に対して新たな知見が加わることにより、情報が変化し、それに伴い作成したクリティカルパスウェイも変更する必要がある。このように、常に情報の変化が起こり、パスの変更が発生するような状況において、クリティカルパスウェイ作成を支援するための「クリティカルパスウェイサービス」について考えた。サービス構築のための基本設計概念としての「クリティカルパスウェイサービス・アーキテクチャ」として、3つの層からなるサービスを設定した（図3.9参照）。

一つ目の層である「クリティカルパスウェイ連携サービス（CP・CaaS: Clinical Pathway Collaboration as a Service）」では、クリティカルパスウェイ作成に必要な情報共有の支援や表示形式の統一、情報の連携を行う際のセキュリティ管理を行うサービスなどが含まれる。二つ目の層は「クリティカルパスウェイ作成サービス（CP・DaaS: Clinical Pathway Development as a Service）」で、実際に様々な情報を元にしたパス作成を支援するためのサービスが含まれる。例えば、パスの達成目標であるアウトカムの標準を定めた BOM（Basic Outcome Master）作成を支援するサービスやパス作成において特に重要な要因であるクリニカル・インディケーターを設定するためのサービスなどが含まれる。三つ目の層は「クリティカルパスウェイ評価サービス（CP・EaaS: Clinical Pathway Evaluation as a Service）」で、作成したパスの評価を支援するサービスである。例えば、達成できなかったアウトカムに関するバリエーションの分析を行うサービスや、評価のツールを提供するサービス、さらには DPC（診断群分類包括評価の導入）から見たパスの設定評価を行うサービスなどが提供される。

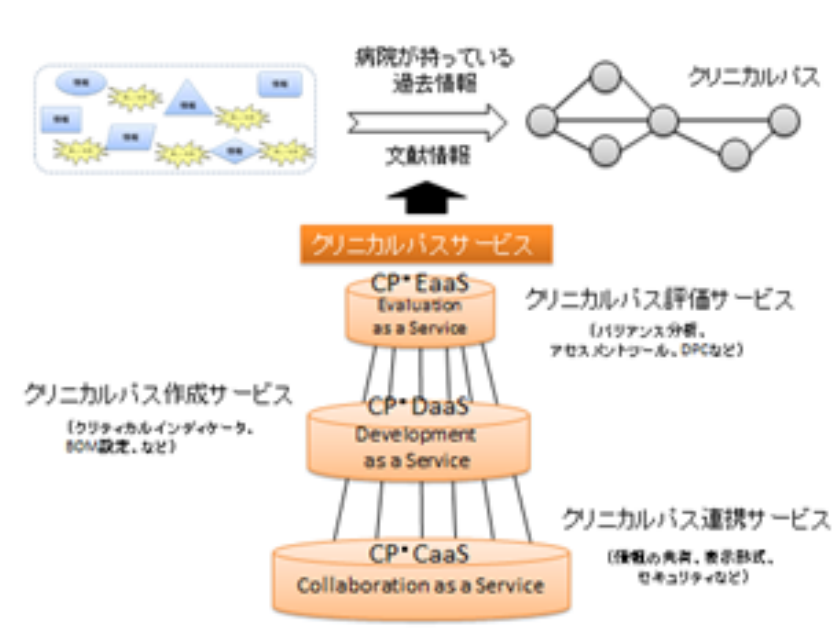


図 3 9 クリティカルパスウェイサービス・アーキテクチャ

6-2 医療情報の共有における現状と課題

医療における診療情報の共有は、その高度なレベルと法的な規制から特異な存在であると言ってもよい。多職種がかかわるチーム医療が当たり前の状況となり、患者に関する情報の共有は医療現場では必須である。これら情報共有は担当医間での共有から医師、看護師といった多職種との共有、さらに技師を交えたチーム内での共有、さらには医療機関などの医療情報システム内のすべての医療スタッフ間での共有へと規模が拡大しているおり、例えば、ある医療機関にて発見されたアレルギー情報の有無などの重要な医療情報が他医療機関においても共有されることは、患者も医療機関も全体的な安全性の向上になる。

一人の患者に数多くの医療スタッフが関与する現在の医療では、それぞれのスタッフ間での情報共有は当たり前となっており、それをどのようなかたちで確実に実行するのかは常に課題とされている。

また、医療情報は専門的知識の集まりであるため、専門的知識であるが故の課題も散見される。既に述べてきた通り、教科書に書いている通り、という知識ではないため、それを形式化して他者へ伝える形にするのは難しい。伝えようとしている医療者の理解度も不明なため、どのレベルで伝えることが良いのか非常に不明瞭であることが多い。つまり、伝える方法は属人的であり、人それぞれによって異なることが多い。そのため、標準化が進められているがその標準化も乱立している状況であり、明確な伝え方はないということが課題である。しかも、「サービスロジック」を構築するためには、リアルタイム性のある患者情報を収集する必要がある、医療情報の共通理解、そして伝えたいことを、その人の理解度に合わせて伝えられる仕組みが必要であると考えた。

6-2-2 ニューロロジカル・レベルによる医療情報の知的表現

これまでの医療情報は、医療機関の各部門に対する医療環境の充実を図るものであるといえる。本研究では、医療行為の支援を行うことに焦点をあてた医療情報システムの構築を目指すこととする(図40参照)。医療行為の支援とは、①ベテラン医師の知識を伝えることができる、②見えないもの(知識)を使えるものにできる、③ベテラン医師の知識の整理ができることの3つを含むものと定義した。



図40 医療環境の充実から医療行為の支援へ

知識の捉え方については様々な見方があるが、本研究では、「状況に応じて情報を取り出すことができる」ものを知識とする。知識の表現方法として、状況に応じて情報を取り出すことを考え、木構造(階層構造)で表現できる知識を扱う。医療の現場においては、医学の専門知識に対する理解をもとに、疾病診断や医療行為が決定される。医療事務作業の現場では、医師、看護師、コメディカルが行った医療行為をもとに、それを診療報酬請求に関わる専門知識と照らし合わせて医療行為を点数化し、最終的に診療報酬として患者に請求される。このように様々な医療に関わる専門知識が存在し、それらの理解が医療行為や医療事務作業に大きく影響を与える。

医療情報システムを利用する医療者の理解に合わせた専門知識を提供するべ

きであるとし、人々の理解の構造について定義しているニューロロジカル・レベルを用いて医療情報の知識表現について考えた。ニューロロジカル・レベルとは、神経言語プログラミング（NLP：Neuro Linguistic Programming）の研究において述べられており、人々が最適な行動を取るための変化と学習のレベルを表している。NLP とは、人々の思考をつかさどる神経回路網が外部からの刺激を受けて、どのように変化しているのか（プログラミング）について述べているものである。そして、その変化のレベルを表したものがニューロロジカル・レベルであると位置づけられる。

ニューロロジカル・レベルは、6つのレベル（環境、行動、能力、信念・価値、自己認識、スピリチュアル）から成り、これらは我々の学習に対する理解の基本的な階層を表していると言える。下位層から上位の層へ移動することにより、感覚的・直感的な思考から論理的な思考レベルへと変化していく。本研究では、ニューロロジカル・レベルにおける「環境レベル」「行動レベル」「能力レベル」を、医療の専門知識に対する医療者の理解度に対応させた。環境レベルは、どこに（Where）、いつ（When）といった外的状況に関わる理解であり、行動レベルは、何を（What）という環境レベルでの具体的な動きについての理解についてである。能力レベルでは、ある環境の中での行動に対して、どのように（How）行動するのかといった、詳しい様子についての理解である。

ニューロロジカル・レベルにおける環境、行動、能力レベルに対して、さらに、それぞれ名詞、動詞、形容詞・副詞レベルを対応させ、医療の専門知識を表現した。図4-1に示すように、医療情報の場合は、病名や薬剤の名前などの名詞レベルに対応させる。行動レベルでは、名詞に対応させる行動として、検査する、処置する、診断するなどの動詞レベルに対応させる。能力レベルでは行動レベルの動詞に対して、例えば迅速に、正確に、ゆっくりと、といった形容詞・副詞レベルに対応させる。これら3つのレベルに対応させることにより、より深い医療情報の理解を図る。医療情報システム利用者の専門知識保有レベルに合わせた情報を的確に提供するため、システム側で医療情報を環境、行動、能力レベルに整理し、それらを表示させることにより、情報の提供側と利用者側のミスマッチを防ぐことができ、利用者にとっても本来欲しい情報が何であるかを把握することができると考える。

これらの医療に関わる専門知識を提供するための概念的仕組みとして、学習者の理解度を測定しながら、理解度を高め、体系的な知識を身に付けさせる仕組みとして、ランドルト環方式（LRM：Landolt Ring Method）を考えた。LRMでの「環」は、視力検査で利用されるアルファベット文字「C」に類似した記号で一部に切れ目を持つ環として表示される。LRMは、学習方法を視力検査になぞらえ、視力検査被験者のランドルト環の切れ目の方向と大きさを「知りたいこ

とのカテゴリ」、「理解の難易度」に関連させて進めていくものである。理解度レベルは、先ほど述べたニューロロジカル・レベル3つのレベルに対応させるものとする。

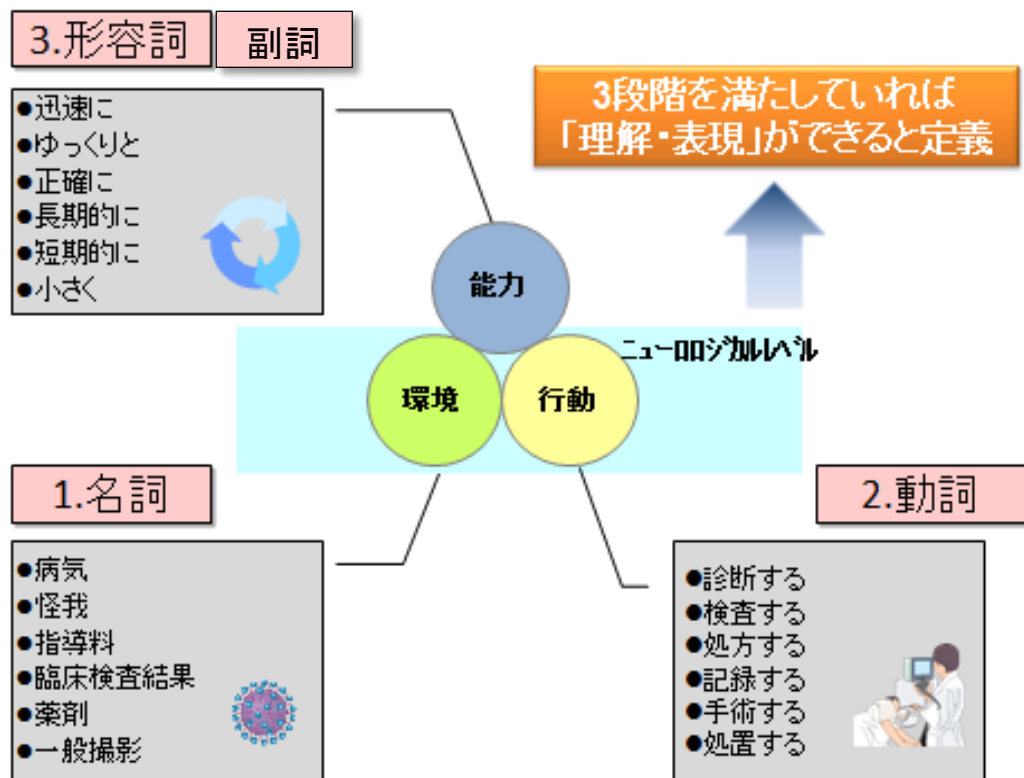


図 4 1 ニューロロジカルを用いた医療情報の整理

6-3 ランドルト環方式による理解

知的資産管理を知識蓄積サイクル、ナレッジマネジメントを知識活用サイクルと位置付け、組織の意思決定力を高めるためにはこれらの連携が重要であり、この連携としてハイパーオーサリングを組み入れた理解度に応じた情報共有を考えた。ハイパーオーサリングの概念は、オーサリングにより体系化された医療情報の集まりに、相互リンクの考えを用いることにより動的にアクセスを行うことができるように構成したものである。このことにより、分散環境において容易に医療情報の相互利用を可能にすることができるのである。即ち、一人一人が持っている医療情報を共有することができ、必要なときに他の人が持っている医療情報を容易に利用することができるようになる。知識ベースに蓄積された医療情報は、e ラーニングとして活用することで高度なレベルの学習ができると考えた。医療における情報共有には医療者の理解度に合わせた e ラーニングをすることが重要である。これを実現する手段として、ランドルト環方式 e ラーニングを考案した。

ランドルト環とはアルファベット文字「C」のような切れ目を持つ環で、視力検査で利用される記号である。ランドルト環方式 e ラーニングは視力検査になぞらえた e ラーニング方式で、視力検査における被験者のランドルト環の見え具合を、e ラーニング、学習者の問題に対する理解度に置き換えて学習を進めていく。

ランドルト環方式 e ラーニングでは、ランドルト環の切れ目の方向を e ラーニングにおける「出題の意図 (カテゴリ)」に見立て、大きさを e ラーニングにおける「問題の難易度」に見立てる。視力検査で最初は適当にランドルト環を指し示し、被験者がそのランドルト環を認識できたら、さらに同程度、もしくは一段階小さいランドルト環を指し示すように、最初は学習者にランダムに問題を出題し、出題された問題を理解できた場合、さらに難易度の高い問題を出題、あるいは本当に理解しているかどうか他の同程度の問題で確認する。出題された問題を理解できない場合、同程度、またはひとつ難易度を下げて問題を出題する。この動きを繰り返し行い、理解度に応じた e ラーニングによる学習をすることができるようになり、適切な「個別学習」ができる (図 4 2、4 3 参照)。

つまり、この理解度に応じた学習が「サービスロジック」へ反映される医療情報となり、これらはリアルタイム性のある患者情報として扱うことができる。

6-3-2 ランドルト環方式 e ラーニングによる情報共有

ランドルト環とは、アルファベット文字「C」のような切れ目をもつ環で、視力を測定する際に用いられる記号である。視力検査では、最初はランダムに選んだいくつかのランドルト環を被験者に示し、ランドルト環の隙間の方向を申告してもらう。そして大体の視力が予測できたら、的を絞って被験者に示していき、視力を決定する。

例えば、最初の方は容易に認識できる大きなランドルト環を示し、切れ目の方向を申告してもらう。そしてだんだんと認識が困難と思われる小さいランドルト環を示していき、同じように切れ目の方向を申告してもらう。ランドルト環の切れ目の方向が認識できない場合、難易度が同程度で切れ目が別の方向を向いたランドルト環を見せ、再度切れ目の方向を申告してもらう。こうした一連の流れを繰り返して視力検査を行い、被験者が申告した結果によって視力を測定する。

ランドルト環方式 e ラーニングシステムは、視力検査になぞらえた e ラーニングの方式である。視力検査被験者のランドルト環の見え具合を、e ラーニング学習者の問題に対する理解度に置き換えて学習をすすめていく。ランドルト環方式 e ラーニングでは、ランドルト環のすき間の方向を e ラーニングにおける「出題の意図（カテゴリ）」に見立て、ランドルト環の大きさを e ラーニングにおける「問題の難易度」に見立てる。

学習者の理解度に合わせて問題を生成・出題し、理解できるように導いていくことによって、受講者のレベルに合った教育をすることができると考えられる。学習者の理解度を「診断」し、診断結果から個人に合致した問題を指し示していくことと、データベースに蓄積された学習履歴をもとに学習者がどの部分を理解していないかを把握し、学習者の進捗状況に合った問題を作成する。最初の問題はランダムに出題し、出題された問題を理解できた場合、さらに難易度の高い問題を出題する。あるいは本当に理解しているかどうか他の同程度の問題で確認する。出題された問題を理解できない場合、同程度、またはひとつ難易度を下げて問題を出題する。以後、これを繰り返す（図 4 3 参照）。

この方式を取り入れることによって個々の理解度に応じた学習をすることができるようになり、適切な「個別学習」が可能となる。

視力検査では、被験者の視力を測定するだけで、それ以上の被験者の視力向上は期待できないが、ランドルト環 e ラーニングでは、学習者の理解できない（＝見えない）問題を少しずつ理解できるように導いていくことができるため、レベルの向上につながると期待できる。

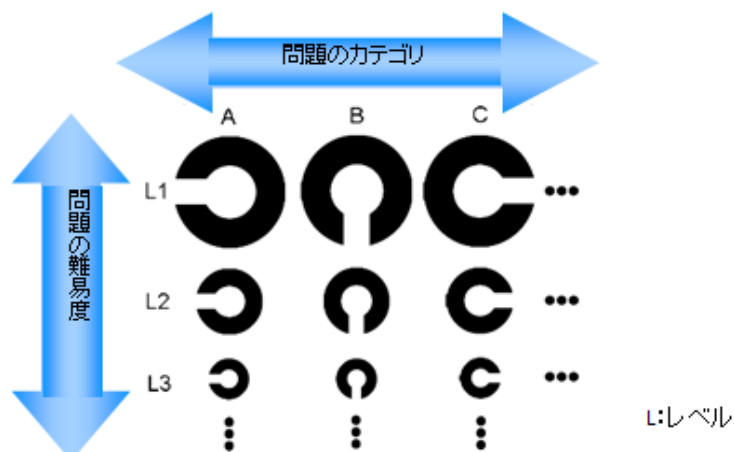


図 4 2 ランドルト環方式

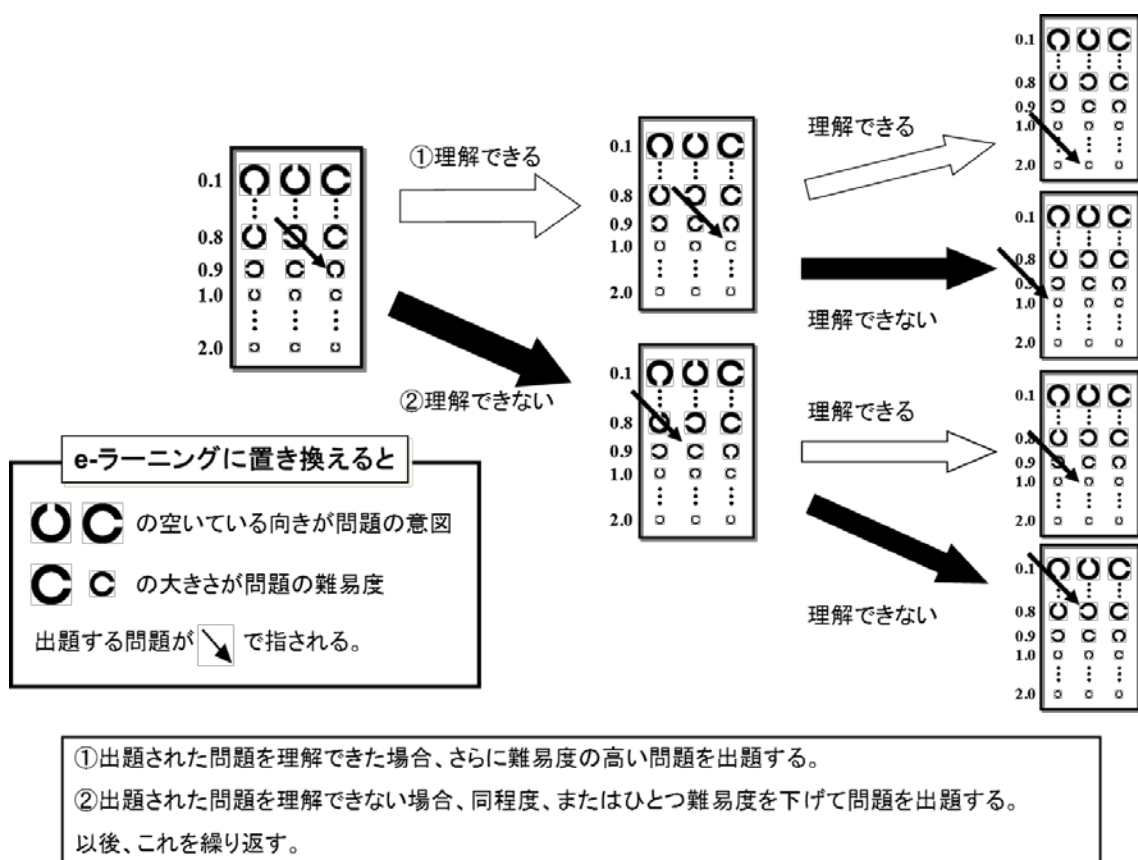


図 4 3 ランドルト環方式による理解度チェック

6-4 XML 言語の利用によるプログラミング強化

本研究では、「状況に応じて情報を取り出し、表示することができる」「状況に応じて情報を取り出すことを考え、木構造（階層構造）で表現できる」もの知識と定義した。各種の医療情報をニューロロジカル・レベルにより階層化を行い、それらを XML 文書で記述した。

データ表現の標準規格である XML (Extensible Markup Language) は、タグの付け方により、データを木構造で表現できる。つまり、XML はマークアップ言語のひとつであり「タグ」と呼ばれる特定の文字列を文書内に配置し、文に意味や構造を持たせることができる。また、現在の電子情報は XML 形式で表現されているものが多い。医療情報を XML によって表現することの利点は、独自のタグを定義することができることから、階層化された医療情報を編集する際に追加や削除がしやすいため、情報の拡張がしやすいこと。さらには、医療情報システムの医師記載内容などは XML にて表現されているため、これらをシームレスに扱うことができる。しかしながら、通常の XML は、データが何であるのか表現することはできても、状況に応じた情報の提供をすることはできない。そこで XML を変換させることができる XSLT を使用することにより、XML で記述した文書に対して、複雑な変換処理を行うことが良いと考えた。XSLT は、XML 文書を別の形式の XML 文書に変換するルールを記述するものであり、本研究ではこれを XML 言語（初歩的な）と呼ぶ。XSLT そのものは、XML 変換処理に関して、基本的なものに限られるため、より高度な変換処理を行うための XML 言語が求められる。そこで筆者が構築している拡張版 XML 言語は一般的なプログラミングロジックの記述、ファイル制御、データベースへのアクセスなどの実行が可能な XML 言語である。さらには XML 言語そのものを拡張するための仕組みを有する。

これら XML 言語そのものを拡張する機能を追加することに非常に大きなメリットと意義があるのだが、代表的な拡張機能として、「タグ拡張機能」や「オブジェクト拡張機能」を上げることができる。タグ拡張機能とは、XML 言語内に展開される要素の集合体をライブラリとして管理し、それらを読み出し、活用することによって、単純なタグの記述で複雑な XML 処理を行える仕組みである。例えば、XML 言語のタグ内における要素として SQL 文を埋め込むことにより、XML データベースへアクセスし、格納されているデータの抽出を行い、そして XML 文書へ変換することができる「SQL タグ」などがある。また、オブジェクト拡張機能とは、オブジェクト指向言語の Java のように必要となるクラス（オブジェクト）を増やすことによって機能を拡張する（図 4 4 参照）。

また、それらの呼び出し先となった XML 言語そのものを変化させることも可能となるため、状況に応じた変化が必要となるシステムを構築する際に利用で

きる。つまり、前述しているように医療情報を状況（患者の病態など）に応じ
て、最適な「サービスロジック」の提供を実現させるための仕組み手段とする
ことができる。

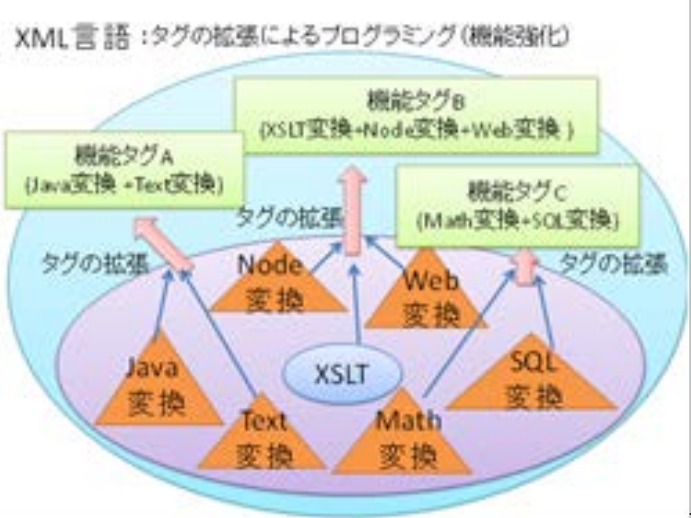


図 4 4 XML 言語の概念図

6-4-2 XML 言語の概要

XML (Extensible Markup Language) は、タグの付け方により、データを木構造で表現することができる。つまり、XML はマークアップ言語のひとつである。マークアップ言語とは、文書の一部を「タグ」と呼ばれる特別な文字列で囲うことにより、文章の構造（見出しやハイパーリンクなど）や、修飾情報（文字の大きさ）を、文章中に記述していくというデータ構造を記述する言語である。マークアップ言語を使用して書かれた文書はテキストファイルになるため、テキストエディタを使って人間が普通に読むことが可能である。代表的なマークアップ言語としては、SGML (Standard Generalized Markup Language)、SGML から発展した HTML (HyperText Markup Language)、XML などが存在している。

XML はマークアップ言語のひとつであるが、タグを使用した囲み方は規定しているが、タグの意味は規定していない。タグの意味を規定するスキーマ言語と組み合わせてマークアップ言語になることができる。このような XML に備わった素材を使い、強化することによって XML 言語はなりたっている。XML 言語は XML 文書を扱うことに特化しており、XML 文書を読み込み、処理し、出力するといったことが可能な言語である。

XML 言語は、XML 形式でスクリプトを記述し、XML 文書を生成するための言語であり、XML 形式でロジックを記述して、その処理結果を XML、HTML、PDF、テキスト形式など、各形式で出力することを可能とする。さらに、単に XML の処理を行うだけでなく、Java ライブラリを呼び出すことが可能となっている。よって、Web アプリケーションで HTML 文書をプログラムから出力する場合、C 言語などでは print 系命令を利用するが、例えば、条件分岐やデータベースアクセスを行う場合、出力の HTML を組み立てるために文字列を連結する処理が必要になり、プログラムの中に C 言語と HTML が混在してしまう。XML 言語はプログラム自体が XML 形式であり、HTML を混在させた場合も違和感はなく利用可能となる。拡張機能を使えば、データベースにも接続可能であり、その出力は XML 文書である。

また、Apache Tomcat などの環境に組み込んで利用することができ、コマンドライン環境から引数を指定して実行することが可能である。

XML 言語は拡張機能を保持しており、1 つめが「タグ拡張機能」である。XML 言語で使用可能なタグセットを増やすことが可能となる。ソースとなる XML 文書内の要素を「関数」と呼び、その関数の集合を「タグライブラリ」としている。タグライブラリの種類を増加させることにより、単純なタグ記述で複雑な XML 処理を行うことが可能となる。2 つめが「オブジェクト拡張機能」である。XML 言語は XPath に似た独自の構文を保持しており、その中でオブジェクト指向にもとづき、オブジェクトの呼び出しを行うことが可能である。オブジェク

トの種類を増やすことで、Java のクラスを増加させるように機能を拡張することが可能となる。

XML 言語の一番の特徴は、「XML 形式でスクリプトを記述し、プログラムそのものが整形形式の XML 文書になっている」という部分である。

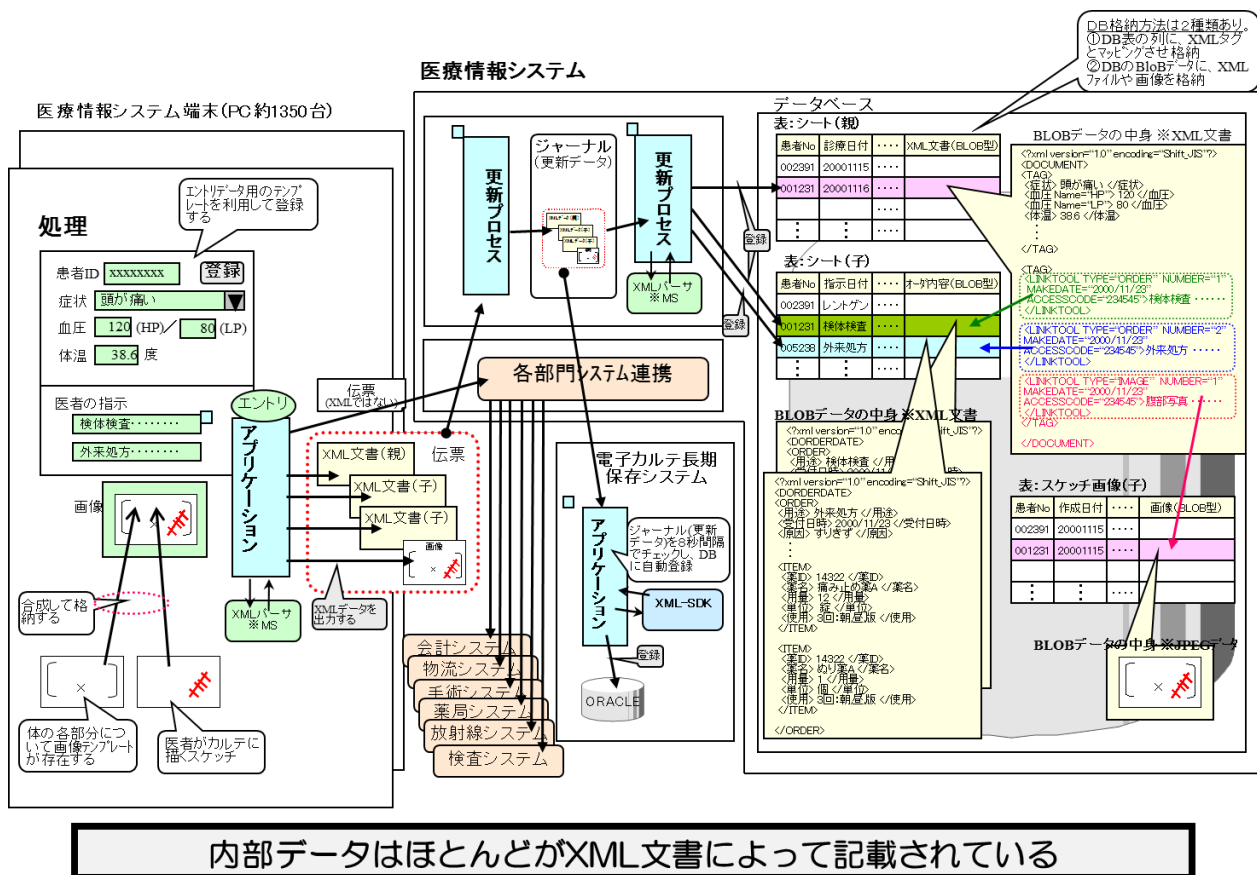


図 4 4-1 XML 文書による内部データの記載

6-4-3 XML 言語におけるタグ拡張機能

タグ拡張機能とは、XML のタグを利用した複数の関数の集まりのことである。これは、XML 形式の拡張であり、主にタグで表現される XML のツリーを生成する、プログラムの実行を制御する機能を追加する際に利用する。以下に代表的なタグライブラリと機能を示す。

(1) XML 言語 : Standard タグ

XML 言語の処理系が標準で実装する関数である。

1 変数を定義する

variable 関数または var 関数を使用する

2 式を評価し、結果のテキスト値を取得する

value-of 関数または eval 関数を使用する

3 式を評価する

exec 関数を使用する

4 変数に値を設定する

set 関数を使用する

5 式を評価し、結果をフラグメントとして取得する

copy-of 関数を使用する

6 テキストノードを生成する

text 関数またはテキスト定数を使用する

7 要素を生成する

element 関数または要素定数を使用する

8 属性を生成する

attribute 関数または、要素定数に指定された属性定数を使用する

9 コメントノードを生成する

comment 関数を使用する

10 処理命令ノードを生成する

processing-instruction 関数を使用する

11 新たなスコープを生成する

block 関数を使用する

11 新たなスコープを生成する

block 関数を使用する

12 条件分岐を行う

if 関数を使用する

13 選択分岐を行う

choose 関数を使用する

14 コレクションの各項目に対し、繰り返し処理を行う

- for-each 関数を使用する
- 15 ソートされた順序で繰り返し処理を行う
 - sort 句を使用する
- 16 条件が成立する間、繰り返し処理を行う
 - while 関数を使用する
- (2) XML 言語:Processor タグ
 - 1 出力形式を指定する
 - output 関数を使用する
 - 2 メッセージを出力する
 - message 関数を使用する
 - 3 別のプログラムを挿入する
 - include 関数を使用する
 - 4 リソースを定義する
 - resource 関数を使用する
- (3) その他のタグライブラリ
 - SQL : XML 言語プログラムに SQL 文を埋め込み、データベースアクセス
 - Mail : XML 言語プログラムからメール送信
 - DB : XML データベースへアクセス
 - Format : 数値や日付のフォーマットを行う
 - Template : XML 言語でテンプレートを定義
 - Spawn : XML 言語から外部コマンドを呼び出す
 - Exception : XML 言語で例外処理を行なう
 - Transform : XML 言語プログラム内で XSLT 変換を行なう

6-4-4 XML 言語におけるオブジェクト拡張機能

オブジェクト拡張機能とは、オブジェクト指向にもとづき、オブジェクトの呼び出しを行うことが可能とする機能である。オブジェクトの種類を増やすことで、Java のクラスを増加させるように機能を拡張することが可能となる。作成したオブジェクトは、インターフェースがきちんと定義された外部環境のオブジェクトにアクセスする場合に有効である。

(1) オブジェクト拡張

- Java-Class : Java クラスにアクセスする
- Bean-Class : JavaBeans にアクセスする
- Math-Class : 算術演算をする
- Calendar-Class : 日付計算を行う
- Text-Class : 簡単な文字列演算を行う
- Text-Class : 簡単な文字列演算を行う
- Node-Class : ノードの直接操作を行う
- Web-Class : Web 環境にアクセスする
- File-Class : ファイルシステムを操作する
- Meta-Class : オブジェクトのメタ情報にアクセスする
- Regex-Class : Perl5 正規表現を使用する
- COM-Class : COM オブジェクトにアクセスする
- WSDL-Class : WSDL ファイルを読み込み、Web サービスにアクセス
- Format-Class : 数値や日付のフォーマットを行う
- Opt-CommandLine : コマンドラインから XML プログラムを起動する
- Opt-Servlet : Web アプリケーションとして XML プログラム動作
- Opt-Cocoon : Cocoon1 のフレームワークの中で動作させるため
- Opt-Cocoon2 : Cocoon2 フレームワークの中で動作させるため
- Opt-Transform : 実行結果のツリーにスタイルシートを適用する

6-4-5 XML 言語におけるデータベース接続

XML 言語から XML データベースに簡単にアクセスするためのタグライブラリである。XML 言語-DB タグを使用すると、XML データベース内に XML 文書の格納、取り出し、検索、更新などを実行することが可能となる。

(1) XML 言語-DB タグ

- 1 コレクションを取得する
collection 関数を使用する
- 2 XML 文書を格納する
save 関数を使用する
- 3 XML 文書を取り出す
load 関数を使用する
- 4 XML 文書の内容を更新する
update 関数を使用する
- 5 XML 文書を検索する
query 関数を使用する
- 6 XML 文書を削除する
remove 関数を使用する
- 7 XML 文書のリストを取得する
list 関数を使用する

6-4-6 XML データベース

Xindice とは、Apache XML Project で開発されているオープンソースの XML ネイティブデータベースのことである。Xindice を基本データベースとしてサービスロジックを格納することを試みた。ここでは、Xindice の起動、終了のバッチファイルを作成したため記載する。

(1) Xindice の起動と終了

Xindice のフォルダにある bin のフォルダにパスを通す必要があり、また、起動するためには、startup コマンドを実行する必要がある。

ここでは、以下のようにバッチファイルを作成する。

Xindice 起動のためのバッチファイル

```
set
XINDICE_HOME=C:\¥bserv_1.1.3¥webapps¥bxi¥xindice
set PATH=%PATH%;%XINDICE_HOME%¥bin
cd %XINDICE%
startup
%SystemRoot%\¥system32¥cmd.exe
```

Xindice 終了のためのバッチファイル

```
set
XINDICE_HOME=C:\¥bserv_1.1.3¥webapps¥bxi¥xindice
set PATH=%PATH%;%XINDICE_HOME%¥bin
cd %XINDICE%
xindiceadmin shutdown -c /db
```

```
C:\WINNT\System32\cmd.exe
C:\%bserv_1.1.3\webapps\%bxi%\xindice

C:\%bserv_1.1.3\webapps\%bxi%\xindice>startup
java -classpath "%config%;C:\%bserv_1.1.3\webapps\%bxi%\xindice\%java%\lib\%ant-1.4.1.jar;C:\%bserv_1.1.3\webapps\%bxi%\xindice\%java%\lib\%examples.jar;C:\%bserv_1.1.3\webapps\%bxi%\xindice\%java%\lib\%infozone-tools.jar;C:\%bserv_1.1.3\webapps\%bxi%\xindice\%java%\lib\%openorb-1.2.0.jar;C:\%bserv_1.1.3\webapps\%bxi%\xindice\%java%\lib\%openorb-tools-1.2.0.jar;C:\%bserv_1.1.3\webapps\%bxi%\xindice\%java%\lib\%xalan-2.0.1.jar;C:\%bserv_1.1.3\webapps\%bxi%\xindice\%java%\lib\%xerces-1.4.3.jar;C:\%bserv_1.1.3\webapps\%bxi%\xindice\%java%\lib\%xindice.jar;C:\%bserv_1.1.3\webapps\%bxi%\xindice\%java%\lib\%xml-apis-1.0.jar;C:\%bserv_1.1.3\webapps\%bxi%\xindice\%java%\lib\%xmldb-sdk.jar;C:\%bserv_1.1.3\webapps\%bxi%\xindice\%java%\lib\%xmldb-xupdate.jar;C:\%bserv_1.1.3\webapps\%bxi%\xindice\%java%\lib\%xmldb.jar;C:\%j2sdk1.4.1%\lib\%tools.jar" -noverify org.apache.xindice.core.server.Xindice C:\%bserv_1.1.3\webapps\%bxi%\xindice\%config%\system.xml

Xindice 1.0 (Birthday)

Database: 'db' initializing
Script: 'GET' added to script storage
Service: 'db' started
Service: 'HTTPServer' started @ http://plasma:4080/
Service: 'APIService' started

Server Running
```

以下に記載しているコレクションとは、XML-DB 中での仮想的なディレクトリに相当し、XML-DB 内の XML 文書は、どこかのコレクションに属することになる。また、コレクションの中にさらにコレクションを作成することも可能である。コレクションを作成するには、Xindice のカレントディレクトリに移してからコマンドを実行する。

■ コレクションの作成(ac) :

例) Xindice のデフォルトのコレクションである、db に mycol という名前の新しいコレクションを作成する。

```
xindiceadmin ac ?c /db ?n mycol
```

というコマンドを実行する

このコレクションの絶対パスは/db/mycol となる。

■ コレクションの削除(dc) :

```
xindiceadmin dc ?c /db ?n mycol
```

アクション	正式名称	動作
ac	add_collection	コレクションの追加
dc	delete_collection	コレクションの削除
lc	list_collections	子コレクションの一覧表示
ad	add_document	ドキュメントの追加
addmultiplied	add_multiple_documents	ドキュメントの一括追加
dd	delete_document	ドキュメントの削除
ld	list_document	コレクション内ドキュメントの一覧表示
rd	retrieve_document	ドキュメントの抽出
ai	add_indexer	インデクサ(インデックス)の追加
di	delete_indexer	インデクサの削除
li	list_indexers	インデクサの一覧表示
import	import	コレクションとドキュメントのインポート
export	export	コレクションとドキュメントのエクスポート
xpath	xpath_query	XPath式によるドキュメントの検索

スイッチ	正式名称	必要となるパラメータ
-c	--collection	コレクション名
-e	--extention	拡張子
-f	--filepath	ファイル名(パス名を含む)
-h	--help	コマンドツールのヘルプ
-n	--nameOf	コレクションもしくはドキュメントの名称
-p	--pattern	インデックスを付すデータを示すためのXPath式
-q	--query	ドキュメント検索のためのXPath式
-s	--namespaces	接頭辞＝名前空間URL; 接頭辞＝名前空間URL;...
-l	--localdb	ローカル環境のデータベースにアクセスする(-dを指定可)
-d	--dbconfig	データベース設定ファイル名称(パス名を含む)(-lと共に指定)
-t	--type	インデックスをつけるデータのデータ型
	--pagesize	インデックスのページサイズ
	--maxkeysize	インデックスのキーの最大サイズ

6-4-7 XML 言語における Prolog タグの利用

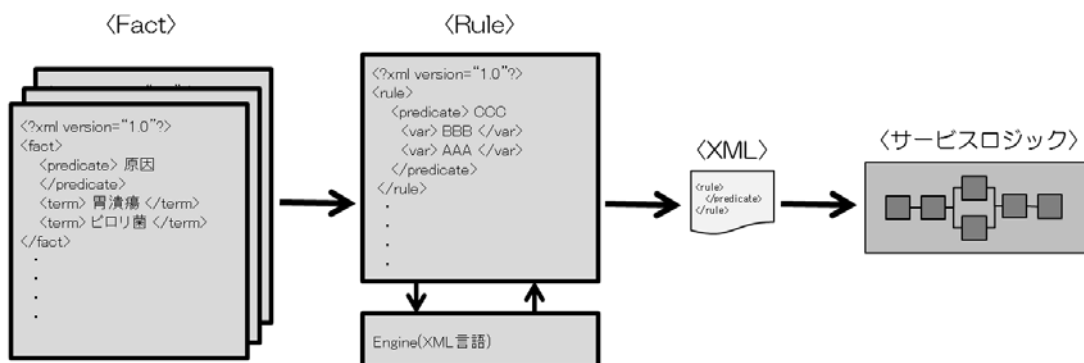
XML 言語に Prolog タグを搭載し、人工知能としてのタグ機能を持たせることにより、XML 言語はインプットされた XML 文書（知識）をもとに推論を行い、既存の XML 文書（知識）を組み合わせる新しい知識を作ることができる。この技術により、医療情報システムにおける患者の状況をもとに作成された XML 文書がインプットになり、その状況（XML 文書）をもとに推論される「サービスロジック」を構築可能とすることを考えている。そのためのコアプログラムを作成した。

```
<?xml version="1.0" encoding="UTF-8"?>
<xi:program
  xmlns:xi="http://www.baykit.org/Xi/1.1"
  xmlns:db="http://www.baykit.org/Xi/db"
  xmlns:pr="http://www.baykit.org/Xi/processor"
  xmlns:prolog="http://www.hit.ac.jp/k-lab/Xi
/prolog">
<pr:resource name="frColl">
<type>XMLDB</type>
<driver>org.apache.xindice.client.xmldb.
DatabaseImpl</driver>
<url>xmlldb:xindice:///db/mycol</url>
</pr:resource>

<prolog:q message="new"/>
  問題探索空間を新しく作成。すでにある場合は削除され作成。
<prolog:fr> . . . </prolog:fr>
  fact と rule を入れる。
<prolog:q message="query"> . . . </prolog:q>
  問い合わせとなり、推論結果は結果として元の位置に出力される。
<prolog:q message="query-again"/>
  再探索を行う。
```

人工知能としてのXML言語

- XML言語にPrologタグを搭載し、人工知能タグ機能を持たせることにより、XML言語はインプットされたXML文書（知識）をもとに推論を行い、既存のXML文書（知識）を組み合わせる新しい知識を作ることができる。
- この技術により、医療情報システムにおける患者の状況をもとに作成されたXML文書がインプットになり、そのXML文書をもとに推論される「サービスロジック」を構築可能とすることを考えている。



上記に人工知能としてのタグ機能を搭載したコアプログラムの構成を記載する。そのインプット (Fact) は XML 文書であり、定義された Rule によって、アウトプットされる XML 言語が可変する。つまり、XML 文書に人工知能プログラムを埋め込み、それらのプログラムが、コアプログラムである XML 言語エンジンと Rule ロジックにより処理が行われ、サービスロジックを構成することが可能となる。ここでは、XML 言語、Fact、Rule による XML 文書の出力が可能なプログラムを作成した。

6-4-8 XML 言語における推論ロジック

XML 言語に fact タグを適用し、推論エンジンに対する事実を設定する。

(1) Fact タグ (事実の設定)

```
<?xml version="1.0" ?>
- <facts xmlns:src="http://xml.apache.org/xindice/Query" src:col="/db/mycol"
  src:key="father-fact.xml">
- <fact>
  <predicate>父</predicate>
  <term>桜</term>
  <term>一太郎</term>
</fact>
- <fact>
  <predicate>父</predicate>
  <term>学</term>
  <term>一太郎</term>
</fact>
- <fact>
  <predicate>父</predicate>
  <term>太郎</term>
  <term>一太郎</term>
</fact>
- <fact>
  <predicate>父</predicate>
  <term>一太郎</term>
  <term>才藏</term>
</fact>
</facts>
```

```
<fact>
  <predicate> 父 </predicate>
  <term> 桜 </term>
  <term> 一太郎 </term>
</fact>
```

<facts>・・・</facts>に fact タグを設定する。

<fact>・・・</fact>に推論エンジンに対する「事実」を設定する。

<predicate> 父 </predicate>には、述語を設定する。ここでは「父である」という事実を設定する。

<term> 桜 </term>は主部を構成する項であり「～は」という部分となる。ここでは「一太郎は桜の」という意味を持たせた構成となる。述語が同じでも、構成が異なると異なる事実となる。

(2) rule タグ (規則の設定)

```
<?xml version="1.0" ?>
- <rules>
- <rule>
  - <predicate>
    祖父
    <var>自分</var>
    <var>x祖父</var>
  </predicate>
  - <predicate>
    父
    <var>自分</var>
    <var>a</var>
  </predicate>
  - <predicate>
    父
    <var>a</var>
    <var>x祖父</var>
  </predicate>
</rule>
</rules>
```

<rules>・・・</rules>に rule タグを設定する。

<rule>・・・</rule>に推論エンジンに対する「規則」を設定する。

<predicate> 祖父 </predicate>は「祖父である」を設定する。よって、そのあとに続く<predicate>で述べられる「規則」となる。

この部分を頭部という。<var>タグは変数を設定し、まだ値は定まっていない状態であり、値を定めるときは<term>タグを利用する。また、同じ変数名は、どこかで値を設定すると、すべてがその値を参照することとする。

(3) Query タグ (問い合わせ)

<a>

```
<prolog:q message="query">
  <query>
    <predicate>
      祖父
    <var> x </var>
    <var> y </var>
  </predicate>
</query>
</prolog:q>
</a>
```

上記の問い合わせ XML に対し下記の XML 結果が得られる。

```
<?xml version="1.0" encoding="UTF-8" ?>
- <a>
- <ans>
  true
  <predicate>祖父</predicate>
  <term>桜</term>
  <term>才蔵</term>
</ans>
</a>
```

<var> x </var>を<term>桜</term>とすると、桜の祖父は誰なのか？という問いになる。また、<var> y </var>も<term>才蔵</term>とすると、桜の祖父は才蔵なのか？の問いになるため、<ans> true </true>のみが返される。

<var> x </var>をそのままにして、<var> y </var>のみを<term>才蔵</term>と表現すると、誰かの祖父は才蔵である。その誰かは？という問いとなる。

```

<a>
  <prolog:q message="query">
    <query>
      <predicate>
        祖父
      <var> x </var>
      <var> y </var>
    </predicate>
  </query>
</prolog:q>
<prolog:q message="query-again"/>
</a>

```

再問い合わせを何度か行くと、<ans> false </ans>が返されるため、この値を利用し、ループさせると以下の結果が導きだされる。

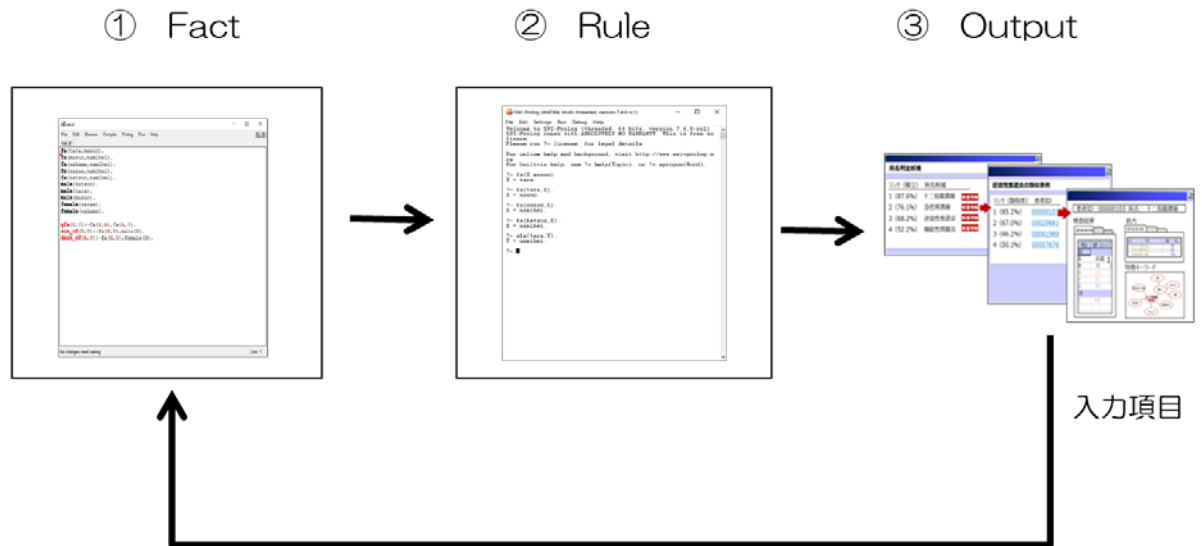
```

<?xml version="1.0" encoding="UTF-8" ?>
- <a>
  - <ans>
    true
    <predicate>祖父</predicate>
    <term>桜</term>
    <term>才蔵</term>
  </ans>
- <ans>
  true
  <predicate>祖父</predicate>
  <term>学</term>
  <term>才蔵</term>
</ans>
</a>

```

以下に Prolog 言語による B 型肝炎再活性化対策のプログラムを記載する。Fact にインプット情報を記載し、B 型肝炎再活性化防止ガイドラインを参考に Rule を Prolog 言語にて記載した。これら Rule 部分はサービスロジックになると考えている。

Prolog言語によるクリティカルパスの表現



Prolog言語によるサービスロジック（Ruleロジック）例

Fact

```

fa(tara,masuo).
fa(masuo,namihei).
fa(wakame,namihei).
fa(sazae,namihei).
fa(katsuo,namihei).
male(katsuo).
male(tara).
male(masuo).
female(sazae).
female(wakame).

```

Rule

```

gfa(X,Y):-fa(X,A),fa(A,Y).
son_of(X,Y):-fa(X,Y),male(X).
daut_of(X,Y):-fa(X,Y),female(X).

```

SWI-Prolog (AMD64, Multi-threaded, version 7.4.0-rc1)

```

File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 7.4.0-rc1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free so
ftware.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.o
rg
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- fa(X,masuo).
X = tara.

?- fa(tara,X).
X = masuo.

?- fa(sazae,X).
X = namihei.

?- fa(katsuo,X).
X = namihei.

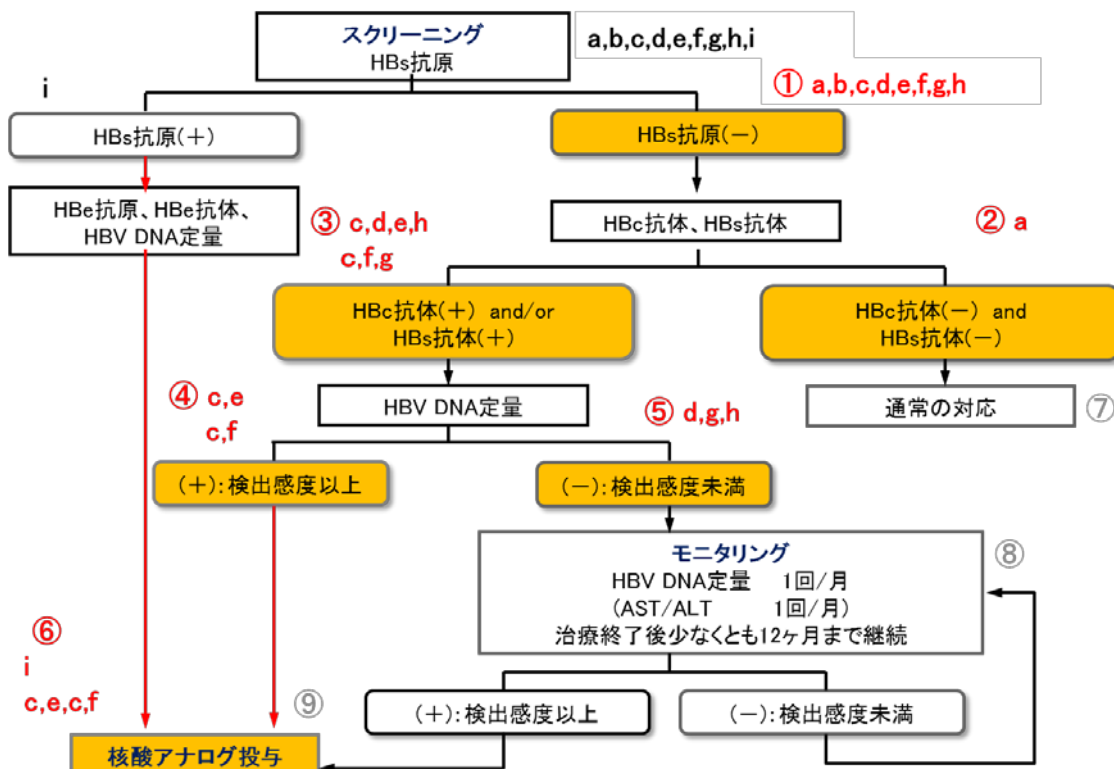
?- gfa(tara,Y).
Y = namihei.

?-

```

“tara”のお祖父
さんは
“namihei”さん

B型肝炎再活性化対策クリティカルパス



Prolog言語によるクリティカルパスの表現

<Fact>

```

b-type_treat2.pl
File Edit Browse Compile Prolog Pce Help
b-type_treat2.pl
/*kannjya*/
patId([a,b,c,d,e,f,g,h,i]).

/*kougen_minus*/
has_HBs_Antigen_M([a,b,c,d,e,f,g,h]).
has_HBc_AntiBody_M([a,b]).
has_HBs_AntiBody_M([a]).

/*kougen_plus*/
/*koutai_plus*/
/*HBV-DNA_plus*/
has_HBs_Antigen_P([i]).
has_HBc_AntiBody_P([c,d,e,h]).
has_HBs_AntiBody_P([c,f,g]).
ge_HBV_DNA_logcopies([c,e,f]).
  
```

Prolog言語によるクリティカルパスの表現

〈Rule〉

```

1. /*1.kougen_minasu(check)*/
   chk_HBs_Antigen_Minus(X):-
     has_HBs_Antigen_M(List),member(X,List).

2. /*2.kougen_minasu(check)*/
   /*2.koutai_minasu(check)*/
   chk_HBcs_AntiBody_Minus(X):-
     chk_HBs_Antigen_Minus(X),
     ( has_HBc_AntiBody_M(List1),member(X,List1)),
     ( has_HBs_AntiBody_M(List2),member(X,List2)).

3. /*3.kougen_minasu(check)*/
   /*3.koutai_plus(check)*/
   chk_HBcs_AntiBody_Plus(X):-
     chk_HBs_Antigen_Minus(X),
     ( has_HBc_AntiBody_P(List1),member(X,List1)),
     ( has_HBs_AntiBody_P(List2),member(X,List2)).

4. /*4.kougen_minasu(check)*/
   /*4.koutai_plus(check)*/
   /*4.HBV-DNA_plus(check)*/
   chk_HBV_DNA_logcopies_GE_plus(X):-
     chk_HBcs_AntiBody_Plus(X),
     ge_HBV_DNA_logcopies(List),member(X,List).

5. /*5.kougen_minasu(check)*/
   /*5.koutai_plus(check)*/
   /*5.HBV-DNA_minus(check)*/
   chk_HBV_DNA_logcopies_GE_minus(X):-
     patId(List),member(X,List),chk_HBcs_AntiBody_Plus(X),
     not(chk_HBV_DNA_logcopies_GE_plus(X)).
  
```

これらRuleが「サービスロジック」となりうる

Prolog言語によるクリティカルパスの表現

〈Rule〉

```

6. /*6.kougen_plus(check)*/
   /*6.HBV-DNA_plus(check)*/
   chk_Nucleic_acid_analog(X):-
     has_HBs_Antigen_P(List),member(X,List),
     chk_HBV_DNA_logcopies_GE_plus(X).

7. /*7(kensa_kekka)*/
   treat_Normal(X):-chk_HBcs_AntiBody_Minus(X).

8. /*8(kensa_kekka)*/
   treat_Monitoring(X):-chk_HBV_DNA_logcopies_GE_minus(X).

9. /*9(kensa_kekka)*/
   treat_Nucleic_acid_analog(X):-chk_Nucleic_acid_analog(X).
  
```

これらRuleが「サービスロジック」となりうる

Prolog言語によるクリティカルパスの表現

< Output >

```
SWI-Prolog (AMD64, Multi-threaded, version 7.4.0-rc1)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 7.4.0-rc1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free s
oftware.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.
org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- chk_HBs_Antigen_Minus(X).
X = a ;
X = b ;
X = c ;
X = d ;
X = e ;
X = f ;
X = g ;
X = h.

?- chk_HBs_AntiBody_Minus(X).
X = a ;
false.

?- chk_HBs_AntiBody_Plus(X).
X = c ;
X = d ;
X = e ;
X = h ;
X = c ;
X = f ;
X = g.

?- chk_HBV_DNA_logcopies_GE_plus(X).
X = c ;
X = e ;
X = c ;
X = f ;
false.

?- chk_HBV_DNA_logcopies_GE_minus(X).
X = d ;
X = g ;
X = h ;
false.
```

Prolog言語によるクリティカルパスの表現

< Output >

```
SWI-Prolog (AMD64, Multi-threaded, version 7.4.0-rc1)
File Edit Settings Run Debug Help

?- chk_Nucleic_acid_analog(X).
X = i ;
X = c ;
X = e ;
X = c ;
X = f ;
false.

?- chk_Nucleic_acid_analog(X).
X = i ;
X = c ;
X = e ;
X = c ;
X = f ;
false.

?-
```


第7章 クリティカルパスウェイ指向の医療情報 SOA

7-1 ロジック記載

医療情報システムは、患者の状況、つまり、常に変化する症状に合わせて情報システムを柔軟に対応させることが可能なアーキテクチャを必要とする。本研究ではこれらの要求に合わせて「サービスロジック」を提案している。このサービスロジックを基盤として、そこに医療情報 SOA のもととなる各種医療行為や部門システムなどのサービスを柔軟に拡張可能とし、それらサービスの組み合わせやプロセスから患者の状況に合致したサービスモデル(治療プロセス)を創造し、その場面で最適な医療行為を提供することが可能となるアーキテクチャである。これらの技術はクラウドコンピューティング上において SOA(Service Oriented Architecture)により構築されたサービス同士が連携し、医療情報 SOA という新しいアーキテクチャとなる(第6章参照)。

各種技術をベースとして「サービスロジック」は実現されるわけであるが、では「サービスロジック」とはそもそも何を意味するものであるのか。それは、ソフトウェア開発における仕様書を具現化したものである。「このサービスでは何をするのか」ということが記載されていなければならない。さらに、この「サービスロジック」には既存システムやサービスを柔軟に拡張可能とし、それらサービスの組み合わせやプロセスから患者の状況に合致したサービスモデル(治療プロセス)を創造し、その場面で最適な医療行為を提供することが可能となる仕様書となる必要がある。つまり、内部のロジックは変更することが前提であり、状況に応じて素早く新しいサービスを作ることが求められる。

これらを実現するためには、この「サービスロジック」を構築するために、いかに専門的でありかつリアルタイム性のある情報をロジックへ反映することが可能かについて考えた。

本研究では、「状況に応じて情報を取り出し、表示することができる」ものの知識と定義し、それらを木構造で表現し、さらに各種の医療情報をニューロロジカル・レベルにより階層化を行い、その情報を XML 言語によるリアルタイム性のある、動的な情報反映とすることについて論じた。さらに、シチュエーション・アクティブ・API という技術の必要性、実現性について論じる。

7-2 シチュエーション・アクティブ・API によるサービス

「サービスロジック」を活用した、医療情報 SOA を構築するにあたり、一つの例としてクリティカルパスウェイサービスを提示する。医療の現場においてクリティカルパスウェイを導入するには、クリティカルパスウェイの作成、現場での運用、分析、見直しという手順が必要になる。パスの作成においては、まず医療の標準化を図るために医療関係者との間での情報共有が必要不可欠である。そして、パス作成後の成果・目標であるアウトカムを設定する。また、実際に作成したパスをどのような形式で表示するのかというフォーマットについても考える必要がある。医療現場におけるパス運用では、実際に院内チームや他病院との連携されたパスが存在し、かりに達成できなかったアウトカムが発生した場合（バリエアンスという）は、その原因等を記録する必要がある。通常バリエアンスの登録は、あらかじめ予測されるバリエアンス原因を要因別に分類しコード化した、バリエアンス要因コードを使用する。

パスの分析では、パス運用の際に記録されたバリエアンスの内容、バリエアンスが発生したアウトカム、バリエアンス発生数の頻度などを集計し分析を行う。これらの分析結果をもとに、評価のためのチェックシートなどのアセスメントツールを使用して、アウトカムとバリエアンスの評価を行い、パスの見直しを行っていく（図45参照）。

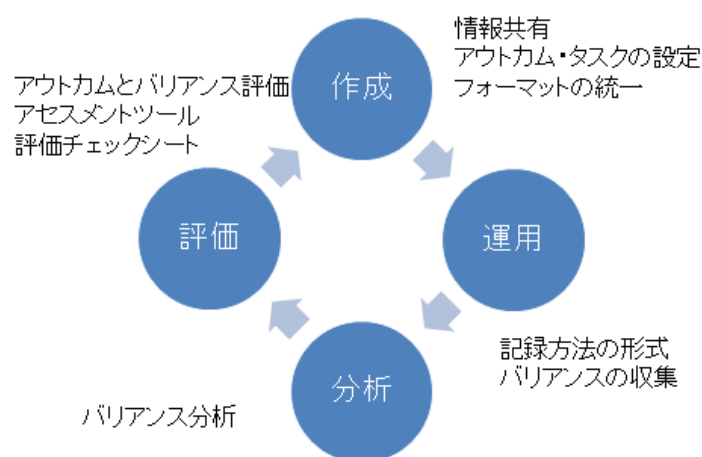


図45 クリティカルパスウェイ導入の流れ

クリティカルパスウェイ作成に必要な情報は、主に病院がこれまで蓄積している過去の情報や、科学的に実証されている文献情報などをもとに作成される。これらの情報を元に作成されるクリティカルパスウェイは、一度作るとその後、変更がないわけではなく、元となる情報に対して新たな知見が加わることに

り、情報が変化し、それに伴い作成したクリティカルパスウェイも変更する必要がある。

このように、常に情報の変化が起こり、パスの変更が発生するような状況において、クリティカルパスウェイ作成を支援するための「クリティカルパスウェイサービス」について考えた。サービス構築のための基本設計概念としての「クリティカルパスウェイサービス・アーキテクチャ」として、3つの層からなるサービスを設定した（図4 6 参照）。

一つ目の層である「クリティカルパスウェイ連携サービス（CP・CaaS: Clinical Pathway Collaboration as a Service）」では、クリティカルパスウェイ作成に必要な情報共有の支援や表示形式の統一、情報の連携を行う際のセキュリティ管理を行うサービスなどが含まれる。二つ目の層は「クリティカルパスウェイ作成サービス（CP・DaaS: Clinical Pathway Development as a Service）」で、実際に様々な情報を元にしたパス作成を支援するためのサービスが含まれる。例えば、パスの達成目標であるアウトカムの標準を定めた BOM（Basic Outcome Master）作成を支援するサービスやパス作成において特に重要な要因であるクリニカル・インディケータを設定するためのサービスなどが含まれる。三つ目の層は「クリティカルパスウェイ評価サービス（CP・EaaS: Clinical Pathway Evaluation as a Service）」にて、作成したパスの評価を支援するサービスである。例えば、達成できなかったアウトカムに関するバリエーションの分析を行うサービスや、評価のツールを提供するサービス、さらには DPC（診断群分類包括評価の導入）から見たパスの設定評価を行うサービスなどが提供される。

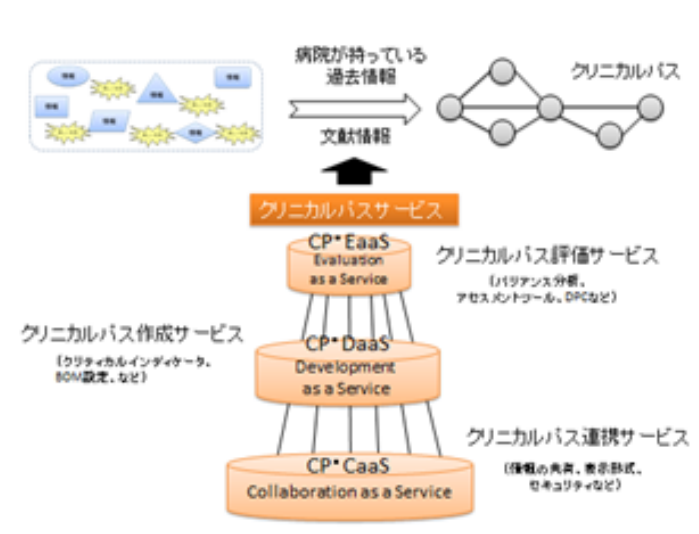


図4 6 クリティカルパスウェイサービス・アーキテクチャ

7-2-2 クリティカルパスウェイサービス創出の仕組み

上記で述べた3層のサービスをもとに「クリティカルパスウェイサービス」を提供していくことを想定している。ここでのクリティカルパスウェイサービスとはITサービスであり、つまり、ソフトウェア群を指す。前述したように、ソフトウェア開発においてはすべての機能をプログラムで記述するのではなく、APIによって共通して利用する機能はあらかじめ定義され、それらを必要な時だけ呼び出して利用している。そこで「クリティカルパスウェイサービス」においても、APIをもとにソフトウェアを構築していくことを想定している。

従来のAPIでは、あらかじめ決められた状況に対応するAPIのみが既に出来上がっており、その状況に対応するAPIのみを呼び出し、ソフトウェアを構築している。「クリティカルパスウェイサービス」創出について考えてみると、情報が常に変化し、それに対応してパスを随時変更していかなければならないことを考えると、事前にこれから起こることを想定することは困難である。よって、あらかじめ想定された状況に合わせて作られたAPIの提供より、状況が変化したときに、それに合致したAPIを瞬時に構築、提供できる仕組み作りが必要になると考える。

本研究では、このようなAPIを「シチュエーション・アクティブ・API」と呼び、シチュエーションで（状況に応じて）、アクティブに（瞬時に）、APIを構築し、組み合わせて使えるようなAPIが今後クリティカルパスウェイサービスを創出する上で必要になると考える。

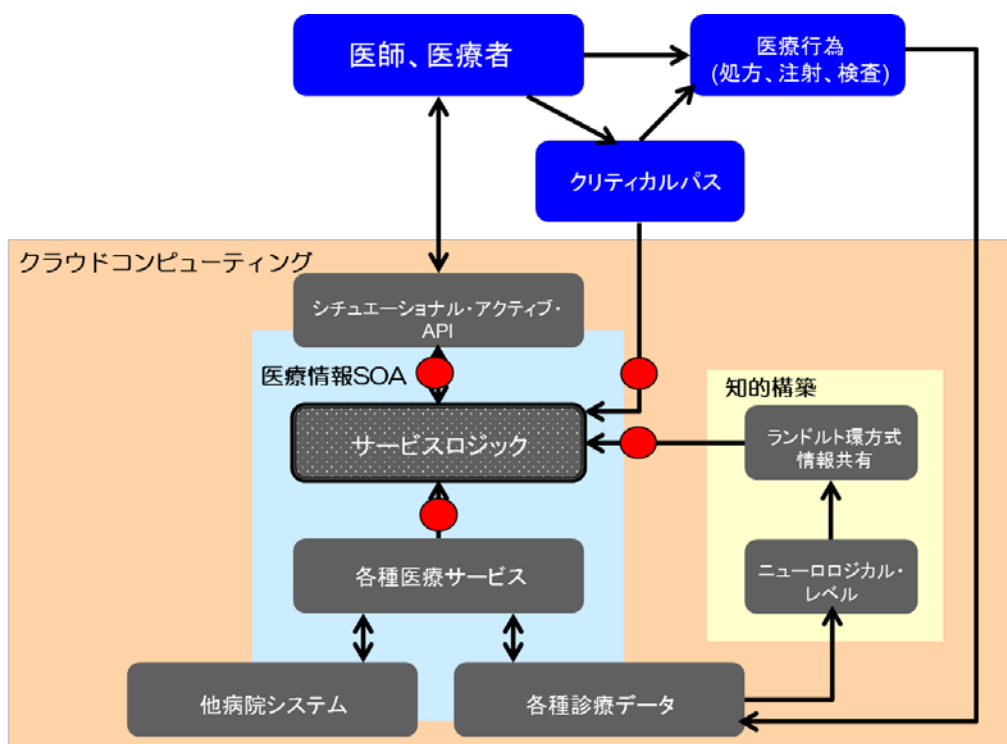


図 4 7 全体俯瞰図

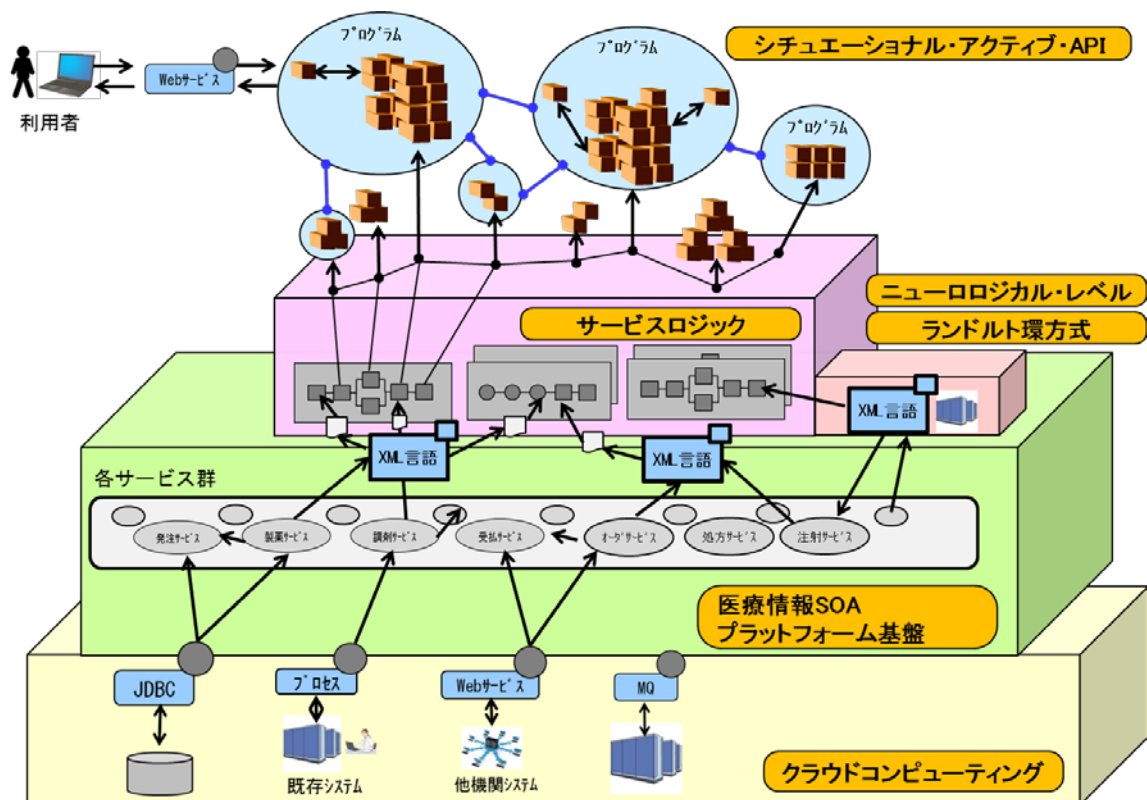


図 4 7 - 1 シチュエーション・アクティブ・API

7-2-3 サービスロジックを基盤とした

シチュエーショナル・アクティブ・API

クリティカルパスウェイサービス、または医療情報 SOA の創出においては、状況の変化に対応するために、シチュエーショナル・アクティブ・API が有効であると考え、これらをシステム開発者側がどのように設計するのか、いわば「シチュエーショナル・アクティブ・API のプログラムの仕様書」に相当するものが必要となる。

クリティカルパスウェイサービスにおけるシチュエーショナル・アクティブ・API の構築について考えてみると、API を利用するユーザは特定のユーザを指すものではなく、ユーザ自身もクリティカルパスウェイの連携先の病院が変更される場合など、状況によって変化することが予想される。また、要求定義を行う場合においても、クリティカルパスウェイは疾病の経過情報によって変化し、それにより、どのようなクリティカルパスウェイサービスが必要になるのか予想することは困難である。したがって、シチュエーショナル・アクティブ・API を構築する際には、従来のソフトウェア開発のように顧客の要求定義が明確であり、それをもとにしたプログラムの仕様書に従ってソフトウェアを構築するのは難しい。

シチュエーショナル・アクティブ・API では、むしろソフトウェアの変更が前提であり、状況に応じて素早く API 構築又は既存の API を結合して新しいサービスを作ることが求められる。

本研究では、シチュエーショナル・アクティブ・API 構築においては、「サービスロジック」をソフトウェア開発に必要なプログラムの仕様書に相当するものとして位置づけており、そのサービスロジックの構築思想、アーキテクチャ、実現方式について述べた。

第 8 章 結語

本研究では、医療機関における医療情報システムの特徴、システムの重要性、ビッグデータやA Iの利用による現況、オンプレミスにおける課題点を明確にしたうえで、状況によって柔軟に作り変えなければならない医療情報システムの構築方法、構成についてのあり方について論じ、それらが医療（サービス）の質に直接つながり、生死を左右する責任を負うことから、医療における情報の品質を高めることによる、医療の質の向上へ貢献するための解決案について述べた。

また、わが国の医療政策として、医療情報システムの統合 ICT プラットフォーム基盤における医療サービスの実現が望まれており、これらを実現する手段として、クラウドコンピューティング上にXML言語を用いた知識ベース（推論機能をもつデータベース）、医療サービス、既存システムなどをJ a v a コントロールにより連携させ、サービスロジックを基盤とした医療情報S O Aプラットフォームの知的構築について考え、リアルタイムな患者情報の反映にはニューロロジカル・レベルによる医療情報の整理とランドルト環方式による理解度に応じた情報共有を応用した知的構築について考えた。

さらには、サービスロジックを実現させるための技術として、XML言語とその拡張機能であるタグ拡張とオブジェクト拡張によるプログラミング強化によってサービスロジックを構築するとともに、このサービスロジックを基にシチュエーション・アクティブ・A P I 構築の必要性、可能性を論じた。

これらは、今後の医療情報システムにおけるソフトウェア開発、システム基盤として必須となる技術である。

参考文献・参考 URL

- [1] 西野正人
「医療の質と病院経営の質の関係についての研究」
商大ビジネスレビュー 第2巻第1号
<http://www.mba.uhyogo.ac.jp/pdf/SBR/2-1/193.pdf> (参照 2016-10-16)

- [2] 富田健司
「医療の質とサービスの質」
同志社商学 第63巻1・2号
<http://doors.doshisha.ac.jp/duar/repository/ir/15477/017063010205.pdf>
(参照 2016-10-16)

- [3] 青山ゆう子、廣田健一、北守一隆
「クリニカルパスサービス・アーキテクチャ構成のためのサービスロジック」日本情報経営学会誌 企画計画 vol.33 No.3 : 78～85、2013

- [4] 松本美穂、松本崇博
「SQL Server 2012 の教科書 開発編」

- [5] 青木浩朗
「Oracle バックアップ・リカバリ実践テクニック」

- [6] 日本医療情報学会
<http://jami.jp/jami/index.html> (参照 2012-03-31).

- [7] 情報システム学会
「情報システムの定義」
<http://www.issj.net/is/02/index.html> (参照 2012-03-31)

- [8] L・マイケル・ホール (橋本敦生監訳、浅田仁子訳)
「NLP ハンドブック 神経言語プログラミングの基本と応用」
春秋社 (2007)

- [9] ロバート・ディルツ
「NLP コーチング」
VOICE (2006)
- [10] 興膳肇、三上行生、北守一隆
「生産管理技術習得のためのランドルト環方式 e ラーニングシステム」
日本生産管理第 31 回全国大会講演論文集 (2010)
- [11] NTT ソフトウェア株式会社 EA コンサルティングセンター
「図解入門よくわかる最新エンタープライズ・アーキテクチャの基本と仕組み」
秀和システム (2005)
- [12] 大森久美子・西原琢夫著、宇治則孝監修
「ずっと受けたかったソフトウェアエンジニアリングの新人研修 開発現場編」
翔泳社 (2012)
- [13] 岡本奏岳
「クリニカルパスの見直しにおける基本的な考え方と手法」
日本クリニカルパス学会誌 第 12 回日本クリニカルパス学会学術集会抄録集』 Vol. 13, No. 4, p. 347 (2011)
- [14] 黒崎寛之
「要点解説 IT サービスマネジメント」
技術評論社 (2010)
- [15] 財団法人 日本情報処理開発協会監修
「情報処理技術者スキル標準対応ー基本情報処理テキスト 2005 年版ー No. 2 システム開発と運用」
株式会社コンピュータ・エージ社 (2005)
- [16] 笹鹿美帆子・菅野由貴子 (2000)
「チームで取り組む クリティカル・パス」
日本看護協会出版会

- [17] 日本クリニカルパス学会学術委員会監修
「基礎から学ぶ クリニカルパス実践テキスト」
医学書院 (2012)
- [18] フィリップ・コトラー著 恩蔵直人訳 (2001)
『コトラーのマーケティング・マネジメントーミレニアム版』
ピアソン・エデュケーション (2001)
- [19] 山田正樹
『実践アジャイル ソフトウェア開発とプロジェクト管理』
ソフト・リサーチ・センター (2005)
- [20] 和田充夫、恩蔵直人、三浦俊彦
『マーケティング戦略』
有斐閣アルマ (2006)
- [21] 廣田健一、青山ゆう子、大森義行、北守一隆
「医療情報におけるシステム構成化と情報品質」
日本情報経営学会 第70回全国大会予稿集【春号】、pp85-88 (2015)
- [22] 厚生労働省
「Ⅱ安心・信頼の医療の確保と予防の重視」
<http://www.mhlw.go.jp/bunya/shakaihoshho/iryouseido01/taikou03.html>
- [23] 地域連携クリティカルパスの普及・推進方策骨子
http://www.phcd.jp/02/kenkyu/chiikihioken/pdf/2008_02_03.pdf
- [24] SS-MIX 普及推進コンソーシアム
<http://www.ss-mix.org/cons/>
- [25] 日本医療情報学会
「SS-MIX2 標準化ストレージ仕様書 Ver. 1.2c データ格納方法およびデータ定義 (2016.04.11版)」

- [26] 廣田健一、青山ゆう子、北守一隆
「イノベーションを創出させるブレイクスルー・コンサルティング」
日本企画計画学会 企画計画、vol. 20, no. 1, pp. 39-44 (2014)
- [27] 室橋高男、廣田健一、保坂良資
「RFID タグによる院内個体管」
電気学会 電子・情報・システム部門論文集 平成 25 年 9 月 4 日発、
pp. 485-487 (2013)
- [28] 廣田健一、福田奈々、渡辺功、北守一隆
「ニューロロジカル・レベルを用いた医療情報システム」
日本企画計画学会 企画計画、vol. 18, no. 1, pp. 36-41 (2012)
- [29] 廣田健一
「診療情報ビッグデータの有効活用による臨床・研究へのデータ指向アプローチ」
月間「新医療」2 月号 pp. 57-603 (DWH を真に活用する病院からの証言論文) (2016)
- [30] 廣田健一
「BI ツールと Hadoop の導入により埋もれたデータを“使えるデータ”にして治療の最適化や医療の質の向上をめざす」
HOPE Vision vol. 22 2015 July, pp4-5 (これからの医療につなぐデータ活用術) (2015)
- [31] 廣田健一
「外来電子化に向けた活動～スキャンセンター設立とその運用」
医事業務 12 月 1 月合併号 No443 (経費削減プロジェクト第 1 部事例 4) (2013)
- [32] 下佐粉昭、野間愛一郎、久保俊彦、高橋賢司
「XML-DB 開発 実技コース」
翔泳社 (2008)
- [33] 日本クリニカルパス学会
「クリニカルパス用語解説集 2012. 3. 30 初版第 2 刷」 (2012)

- [34] 黒田知宏監修 電子情報通信学会編
「現代電子情報通信選書『知識の森』 医療情報システム」
オーム社 (2012)
- [35] 川道亮治、佐藤誠、岸芳三、生沼一公、今井陽平、星野尚、篠原邦武、
中内文、中内瑞樹、安藤利和、山崎毅
「オープンソース XML プロジェクト完全解説 横浜ベイキットオフィ
シャルガイド」
毎日コミュニケーションズ (2003)
- [36] 日本 BEA システムズ株式会社
「SOA 実践のための BEA WebLogic Platform 活用ガイド」
翔泳社 (2005)
- [37] 日本医療情報学会医療情報技師育成部会
「新版 医療情報 医療情報システム編」
篠原出版新社 (2009)
- [38] 大谷純、阿部慎一郎、大須賀稔、北野太郎、鈴木教嗣、平賀一昭著、
株式会社リクルートテクノロジーズ、株式会社ロンウイト監修
「オープンソース全文検索エンジン Apache Solr 入門」
技術評論社 (2014)
- [39] 佐々木達也
「Hadoop ファーストガイド」
秀和システム (2012)
- [40] 三木 大知
「パターンでわかる Hadoop MapReduce -ビッグデータのデータ処理入門」
翔泳社 (2012)
- [41] 保健医療分野における ICT 活用推進懇談会 (平成 28 年 10 月 19 日)
<http://www.mhlw.go.jp/stf/shingi/other-jyouhouseisaku.html?tid=312619>

- [42] データヘルス時代の質の高い医療の実現に向けた有識者検討会（平成29年1月12日）

<http://www.mhlw.go.jp/stf/shingi/other-hoken.html?tid=350947>

用語説明

[1] AI (Artificial Intelligence)

広義としての人工知能のこと。人間の脳が行っている知的な作業をコンピュータで模倣したソフトウェアやシステム。具体的には、人間の使う自然言語を理解したり、論理的な推論を行ったり、経験から学習したりするプログラムなどのことをいうが、これらは強い人工知能とも呼ばれる。

[2] アイソトープ (isotope)

同じ化学元素に属しながら、質量数の異なる原子を互いにアイソトープであるという。同位体、同位元素ともいう。原子記号の左肩に質量数を付けて区別する。例えば水素=H、重水素= ^2H 、三重水素= ^3H と表す。

[3] ラジオアイソトープ (radioisotope)

アイソトープのうち放射性のものをいう。略して RI と称する。核医学部門では放射性医薬品を指し、心筋シンチ検査、骨シンチ検査などのため放射性医薬品をごく微量体内に投与し、放射線医薬品から出る放射線を検出することにより診断や治療に使用する。

[4] SCM (Supply Chain Management)

小売から卸や物流、部品メーカまでネットワークで統合し、販売や生産、在庫などの情報を企業間で共有することで納期短縮や在庫削減につなげ、経営効率を一気に向上させる手法。また、そのための情報システムのこと。

[5] Hadoop (ハドゥープ)

データを複数のサーバに分散し、並列して処理するミドルウェアのこと。テラバイト、ペタバイト級大容量データの分析などを高速処理できるため、「ビッグデータ」活用における主要技術として活用が進んでいる。

[6] HDFS (Hadoop Distributed File System)

分散処理システムの Hadoop が利用している分散ファイルシステムのこと。OS のファイルシステムを代替するものではなく、その上に独自

のファイル管理システムを構築するものであり、通常は OS にマウントして様々なプログラムからアクセスすることはできない。

[7] MapReduce

コンピュータ機器のクラスター上での巨大なデータセットに対する分散コンピューティングを支援する目的で、Google によって 2004 年に導入されたプログラミングモデル。並列処理を Map と Reduce の二つのフェーズに分け、Map がデータを抜き出し、Reduce がデータの加工や集計をする。

[8] Apache Solr

Java ベースのオープンソース高機能全文検索エンジンのこと。

[9] Apache ZooKeeper

オープンソースプロジェクトで、大規模分散システムでよく利用される、設定情報の集中管理や名前付けなどのサービスを提供するソフトウェア。

[10] Apache HBase

HDFS (Hadoop Distributed File System) 上に構築された分散データベース。大量の非常に細かいデータをリアルタイムに読み書き出来るのが特徴。Facebook Message の基盤技術としても使用されている。

[11] CDSS (Clinical Decision Support System)

医療情報システムにおける診療判断支援機能のこと。「免疫抑制・化学療法により発症する B 型肝炎対策ガイドライン」に基づいて開発した本研究「B 型肝炎再活性化防止システム」がこれに相当する。

[12] API (Application Program Interface)

プログラム(ソフトウェア)の機能や管理するデータなどを、外部の他のプログラムから呼び出して利用するための手順やデータ形式などを定めた規約。

[13] REST (Representational State Transfer)

HTTP を使ってデータをやり取りする手法のひとつ。URL を指定すると XML でデータが得られるため、Web ブラウザで簡単に動作確認ができる。XML を使ったデータ交換の技術には SOAP があるが、SOAP より手軽

に扱えるため、Web サービスと通信する手段としてよく利用される。

[14] ASP (Application Service Provider)

インターネットの Web をベースとした特定のアプリケーション処理サービスのこと。「Web サービス」ともいう。

[15] DICOM (Digital Imaging and Communication in Medicine)

米国放射線学会 (ACR) と北米電子機器工業会 (NEMA) が開発した、CT や MRI、CR など撮影した医用画像のフォーマットと、それらの画像を扱う医用画像機器間の通信プロトコルを定義した標準規格のこと。

[16] EAI (Enterprise Application Integration)

アプリケーション統合。もともと独立して設計運用されている企業ビジネスアプリを連携させること。企業内の異なる業務システム間や、企業間のシステムを相互に連携し、データやプロセスを統合するための技術。

[17] マークアップ言語

文書の一部を「タグ」と呼ばれる特別な文字列で囲うことにより、タイトル、ハイパーリンクなどの文章の構造や、文字の大きさや色などの修飾情報を文章中に記述していく記述言語。

[18] XML (Extensible Markup Language)

拡張可能なマーク付け言語。SGM の規格をベースに、インターネットで利用しやすいように決められたもの。インターネットを利用した、ネットワーク型システムでの情報交換を可能にするデータフォーマットの規格として注目されている。

[19] DOM (Document Object Model)

XML 文書 (データ) をアプリケーションから操作する場合の、データモデルとインターフェースを規定したもの。SAX ではできない、XML データへのランダムなアクセスと編集が可能である。

[20] SAX (Simple API for XML)

イベント駆動型で XML 文書の先頭から最後までを、順番に処理する場合のアプリケーションインターフェース (API)。ほとんどが Java で実

装されている。DOM では、XML の全データを展開するが、SAX では先頭から順に処理する。そのため省メモリであり、大きな XML の処理に利用。

[21] XSLT (XSL Transformations)

XML のデータ構造変換を行うための言語規約。XSLT を用いて XML のデータ変換を行うには XSLT のプロセッサが必要。

[22] XPath (XML Path Language)

XML 文書の中の特定の要素を指し示す記述方法を定めた規格。

[23] Apache XMLBeans

Apache ソフトウェア財団の XML プロジェクトの一部である、Java と XML データバインディングとの変換を行うフレームワーク。

[24] XMLObject

省略可能な Web サーバと Web ブラウザ間でやりとりされるデータの中でファイル形式を示す識別子である MIME タイプ、ID、および XML 文章の文字符号化が何で行われているかを示すエンコーディング属性など任意のデータを格納することができる。

[25] Xindice

オープンソースの XML データベースのこと。また、リレーショナルデータベースで簡単に再現できない XML データに固有の機能を提供する。

[26] NLP (Neuro Linguistic Programming)

別名「脳の取り扱い説明書」とも呼ばれる最先端の心理学。最強のコミュニケーションの手法として極めて優れており、同時に目標達成や自己成長にも大きな効果をもたらす。

[27] SGML (Standard Generalized Markup Language)

ISO8879 に準拠した電子文書標準化用の汎用マークアップ（テキストにフォントや文字の大きさ等の指定を書き込むことで、文書構造を記述できる）言語である。

- [28] JavaBeans
Java プログラムから利用できるソフトウェア部品の仕様。個々のソフトウェア部品を Beans と呼ぶ。
- [29] JDBC API (Java Data Base Connectivity Application Program Interface)
Java からデータベースを呼び出すための A P I のこと。
- [30] IBM MQ
多様なアプリケーションやビジネス・データを、複数のプラットフォームで シンプルかつ迅速に統合する IBM のメッセージング・ミドルウェア。
- [31] J2EE (Java 2 Enterprise Edition)
アプリケーションサーバを対象とした Java2 の規格。主要構成要素は、JSP、JavaBeans、Java サブレット。企業の業務システムや E コマース用サーバに必要な機能が実装されている。
- [32] Java Servlet
Web サーバの Java アプリケーションを構築するためのインターフェース。
- [33] EJB (Enterprise JavaBeans)
Java におけるサーバ向けの JavaBeans の仕様のこと。
- [34] コンテナ
プログラミングの分野で、データやオブジェクトなどをまとめて格納するためのデータ構造 やクラスなどの総称。
- [35] コンポーネント
モジュール（関数などの処理単位）が組み合わされているソフトウェア群。
- [36] SOAP (Simple Object Access Protocol)
ネットワーク上のアプリケーションが XML を介して情報を交換し合うための仕様。SOAP は、非集中／分散環境におけるシステム間の構造化さ

れ型付けされた情報交換のための XML を用いた単純で軽量なメカニズムを提供する。

- [37] UDDI (Universal Description, Discovery and Integration for Business on the Web)

XML を使った、インターネット上で提供されている Web サービスを検索するための仕組み。

- [38] WSDL (Web Service Description Language)

Web サービスが提供する機能や必要なパラメータなどを記述するためのマークアップ言語。XML がもとになっている。Web アプリケーションや Web サービスが他の Web サービスを利用するために必要な情報を得るための仕組み。

- [39] Apache Tomcat

Java サーブレットや JSP (JavaServer Pages) を処理する Web コンテナアプリケーションサーバ。

謝辞

本研究に際し、終始懇切なるご指導並びにご高閲を賜りました北海道科学大学大学院工学研究科電気工学専攻 北守一隆教授に心から感謝の意を表しますとともに、終始懇切なるご指導を賜りました北海道科学大学未来デザイン学部メディアデザイン科学 木下正博教授、三田村保教授に心から感謝の意を表します。また、本研究に貴重なご助言を賜りました北海道科学大学保健医療学部義肢装具学科 野坂利也教授に心から感謝の意を表しますとともに、本研究にご協力頂いた北海道科学大学大学院工学研究科電気工学専攻 木下雄太氏にお礼申し上げます。