

C. 1.3 参照実装 (Web 対応)

Native XML DB の検証は廣瀬, Caché の検証は山本 (ツト) と廣瀬の協同, そして実装は山本 (ツト) と大嶺が行った。

C. 1.3.1 Native XML DB の評価

方法(B.14)に記した協賛協力を得て EsTerra を試した。その内容は以下である：

- 1) XML の定義
- 2) XML の取込 (取込時間, データサイズ)
- 3) COM での検索 (速度, 安定性)
- 4) SOAP での検索 (速度, 安定性, 遠隔処理)

EsTerra は well-formed XML に対応している。ゆえに 1) は不要であり, 楽であった。

次に, 2) については, 数千件程度の record 数ならば取込時間も許容範囲内であって, むしろ native XML DBMS としては秀逸である, と評価できた。ただ数万件を超えると, field 数にも拠るものの, 取込には長時間を要した。また, データサイズも source と比較して相当大きくなった。

COM での検索は, EsTerra として optimize 可能な限りは, native XML DB として極めて高速であった。よって 3) は, 多いに満足できた。

SOAP の実現については, Java, Apache, TOMCAT, AXIS 等を要求されるが, これ自体は問題なく, MS Windows 環境においても容易に整えることができた。ただ, 速度と安定性については少々難があったと云わざるをえない。SOAP 自体が, いまだ進化を続けている現況下, これは已むをえないことなのであろう。そのうえ現状の EsTerra では, いわゆる remote procedure を登録することはできない状況である。

これら二点から, C/S を前提としたシステムを構築するためには, 結局のところ, サーバ側に多くの作り込みが必要となることが判明した。さらに前述した如く, 種々, 大きな資源を要求する現況である。

よって, 本研究を推進するにあたっては, 開発に要する時間とコストから, EsTerra の採用は見合わせる事が妥当であろうと判断した。

C. 1.3.2 Caché の採用

Caché について初年度からコード・マスタ用 DBMS として採用している。その performance

や, 必要資源の極小化や最適化については特筆すべきであろう。Caché, または M language の特長は, 主任研究者も大規模病院情報システムにて経験済みである。

ただ Caché は M Language ベースゆえ, これは初心の者にとっての開発については, 関が低いとは云えないように感じていた。

しかし Caché SOAP を試してみたところ極めて容易に SOAP を確立することができ, かつ高速性と安定性とを確認した。

ただ幾つかの制限事項をクリアするための諸工夫が, 後になって必要となったが。

なお(C.13.1)の 1) と 2) については初年度に, 既に確認済みである: 前者は通常の Class 定義または RDBMS table schema 定義と同等である。後者については, それなりに index 定義しても source と大差ないデータサイズにて格納される。

また 3) については, COM ではなく ODBC で通信するが, この速度と安定性についても確認済みである。よって C/S 版参照実装の DBMS には Caché を採用することとした。

C. 1.3.3 SOAP

Caché は高速かつ柔軟だが現状の版での XML や SOAP の扱いについては, 幾つか制限事項または仕様がある:

- a) XML は W3C XML Schema ベース
- b) XML 授受における namespace の未反映
- c) String 送受の 32KB 長制限
- d) Base64 multi-byte char 未対応 (回避策あり)
- e) Encoding 時間問題 (不可避)
- f) Session 管理問題 (現状不可避)

一方, 本研究での参照実装において求められた諸条件の一部は以下の通りである:

- 0) 短期間かつ低コストでの開発
 - 1) ログイン時の情報授受 (32KB 内)
 - 2) マスタ・コード検索 (通常 32KB 内)
 - 3) SQL command の実施
 - 4) 診療履歴 xml ファイルの送受 (32KB 超)
 - 5) 少なくとも簡易な lock 機能

よって 0) により, 慣れない CSP でのシステム構築は避けつつ, 1) 2) は String ベースでの通常の SOAP サービス, 4) は SOAP により Stream をキックすることにした。その際 d) にも配慮することとなったが, その一方で結果的には b) を回避することとなった。

なお 3) はサーバ側モジュールでフィルタし、
また 5) は C/S の双方に簡易版を作り込んだ。

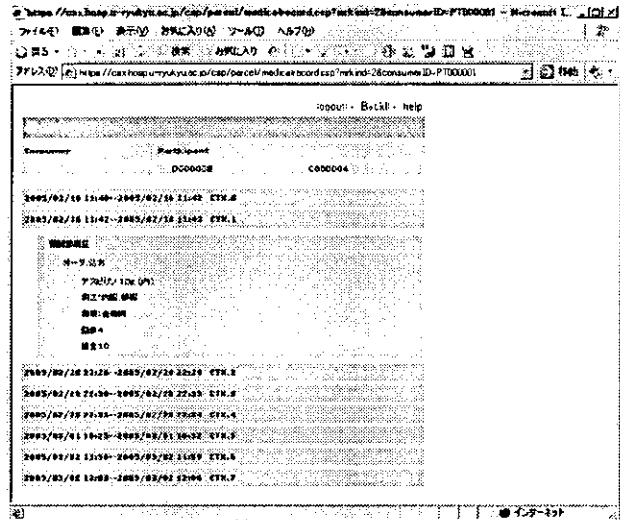
C. 13.4 サーバ参照画面例

C/S 版とは、データ管理等をサーバ側で行い、
操作自体は Client 側で行うものである。

よって画面の構成や展開 ならびに 各操作に
ついては、Stand-alone 版と変わらない。

とはいえ、サーバに格納されたことの確認と、
今後の web 版開発に足掛かりを得ておくために、
web の診療履歴参照画面を作成したので、その
表示例を以下に掲げておく。

下図は、所属と立場を宣言して患者リストを表
示した状態である。

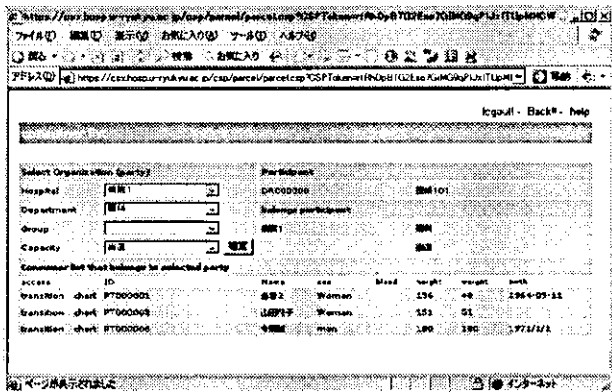


C. 13.5 公開と登録など

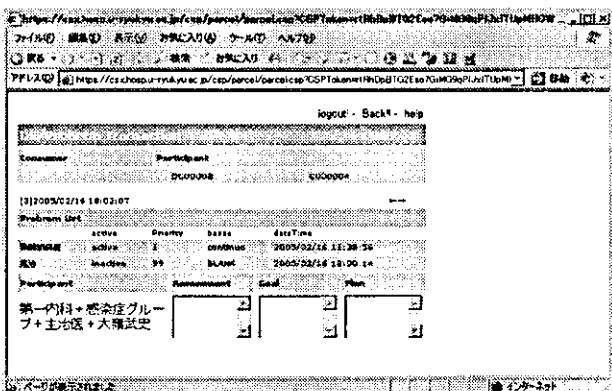
C/S 版での登録等の管理フローほかについては
記述割愛する。というのも、これを詳述すると
操作マニュアルになってしまうからである。

ただし今後、以下で公開する予定である：

<http://www.hosp.u-ryukyu.ac.jp/medi/csx/>
[https://csx.hosp.u-ryukyu.ac.jp/csp/parcel/](https://csx.hosp.u-ryukyu.ac.jp/csp/parcel/login.csp)
[login.csp](https://csx.hosp.u-ryukyu.ac.jp/csp/parcel/login.csp)



下図は、最新の診療セッションの病名リストを
表示した状態である。



右上方の矢印 (←→) をクリックすると前後の
隣接する診療セッションの病名リストが表示
される。

そして、ある診療セッションにおける加療行為
の履歴を表示した状態は、次の図に示している。
Stand-alone 版の“container”と、ほぼ同様の
効果を持つ Class を PHP で作成、制御して表示
している。

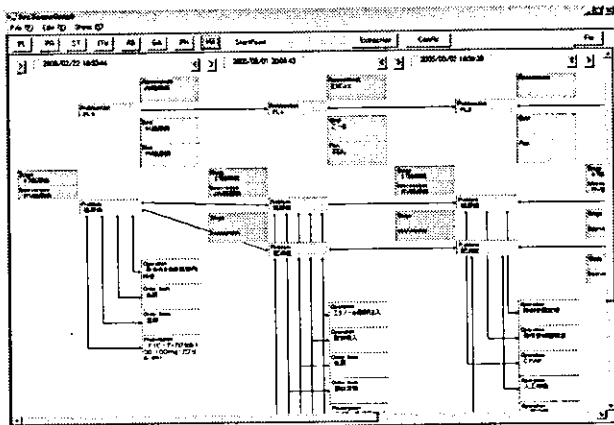
C. 1.4 参照実装 (抽出と要約)

TheTOOL の基本性能は (C.11) に記した。本節では機能や機構を説明する。TheTOOL の実装は与那嶺と山田が行った。なお ArcScopeGraph の原型は、尾藤と山田が製作した Dx/Problem Composer の Graph pane に拠っている。

C. 1.4.1 一覧性

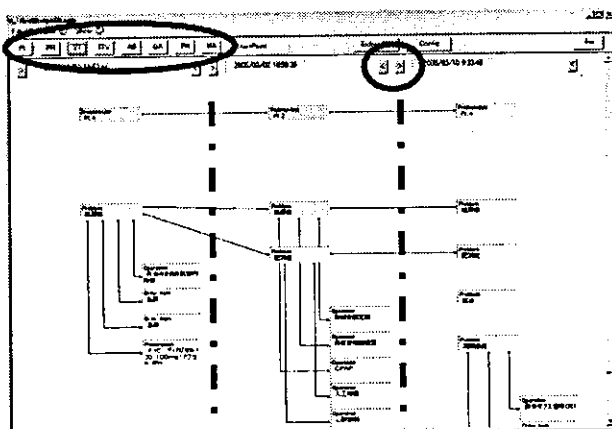
診療システムで記録された診療経過は、病名やプロブレムの変遷とともに、それらと加療行為とを結び付けている。このようなグラフ構造に配された情報塊の諸関係は複雑なので、視覚的に捉えたいものである。

ゆえに、これを表示する画面 ArcScopeGraph を設けた。



これだけでも診療の過程を考察検討するには有益であろう。

とはいえ、考察すべき点を焦点した際、視野をその範囲に絞り込みたいこともある。そこで、(1) infoNode の種類ごとの mask 機能、(2) 一括 mask 機能に関わらず個別の infoNode を常に表示させる lock 機能、(3) 診療セッションごとの「送り」機能をも実装した。



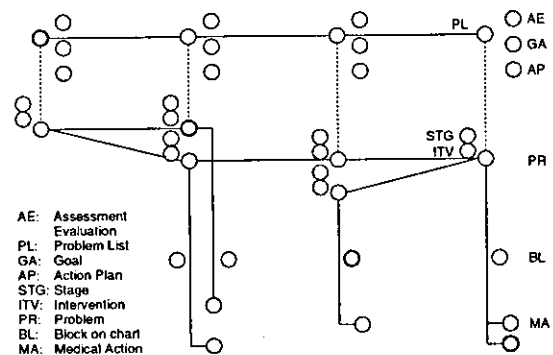
これで通常の意味での「要約」が可能となった。

C. 1.4.2 抽出 (流れ)

ただ病名やプロブレムの変遷を検討する場合には注目した「その DxProblem」の「流れ」を知りたいものである。

よって (4) 特定 infoNode を起点とし、それと隣接する infoNode を順次追跡探索する機能を設けた。

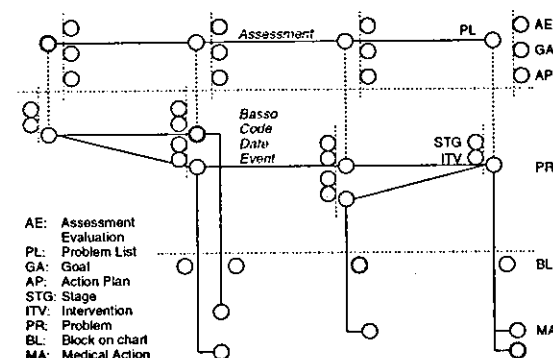
ただし下図の如く、全ての infoNode は相互に連結している。



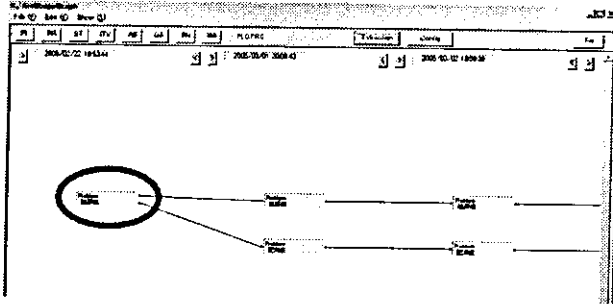
そこで、(5) 探索停止条件の設定が必要となる。その種類は 8 種ある：

- 包絡する infoNode の種類の限定
- 日付による限定
- 診療イベントによる限定
- 根幹病名コードが変化した際の処理
- 病名修飾コードが変化した際の処理
- DxProblem の属性 Basso に依存した処理
- Assessment や Goal の変化に依存した処理
- そのほかの変化に依存した処理

なお、a~c は包絡範囲を限定し、d~h は脈絡条件を指定することになる。ただし c, g, h は対象範囲外とした。



これを実装して、先の例と同じ症例を用いて、実際に抽出処理を実施すると、次の結果が得られることになる。



そしてさらに (C.14.1) に掲げた (1)～(3) の機能を適用したなら、十分に満足できる抽出と要約を得られることになる。

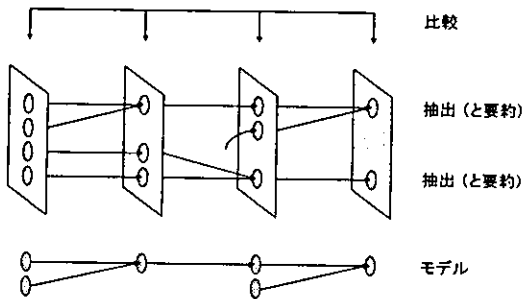
これらの機能を実現できる最大の理由は、将に、病名/プロブレムの変遷と・病名/プロブレムと加療行為とを連関させた・診療経過記録を作成した

ことに拠っている。この抽出要約の結果を xml instance として保存すれば交換は容易であり、臨床経験に基づいた「知識」が共有できることとなった。

なお (B.2) (B.3) ならびに (D.3.2) も併せて御参照願いたい。

C. 1 4 . 3 抽出 (組み)

さらに「流れ」のみならず「組み」を抽出比較したいこともある (下図は (B.4) に既出)。



つまり、相似相同パターンを抽出し、これを並列表示して比較したい、ことがある。ただ、この相似相同パターン並列表示機能は現状の TheTOOL には搭載していない。

C. 1 4 . 4 モデル化へ

C.14.4.1 さらなる「流れ」編集

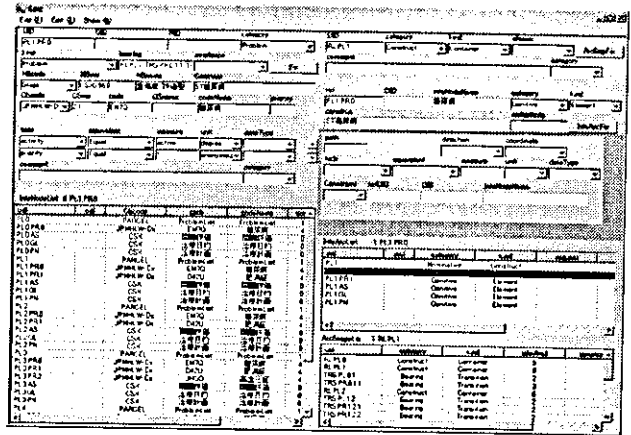
ところで、臨床経過から得られた事実は、必ずしも典型例とは限らない。

たとえば、一つの ProblemList(i) に三つの DxProblem(j) が存在したとしよう。そして、病態生理から考えると、そのうちの一つの

DxProblem は、本来、ProblemList(i-1) に存在しなければならない、としよう。

このような場合、現状の ArcScopeGraph 画面のなかでは、グラフィカルな操作で infoNode を位置変更したり、新たな infoNode を追加したりすることは、現状では、できない。

とはいえ現状においても、EDIT 画面で必要な infoNode を生成することは出来る。また、当然ながら infoArc や arcScope も生成することが出来るようになっている。



C.14.4.2 その他の処理

時刻変更や時刻関係の相対化などについても、上記の EDIT 画面で行うことができる。

ただTheTOOLの入出力はxml documentなので、その様な処理、あるいは匿名化などについても、テキストエディタ等を用いて容易に実施することも可能である。

C. 15 試用アンケート調査

評価を依頼した点は情報モデルと、実システムへの応用可能性、それらから得られるツールと成果（への期待）であった。

これらについては、概ね良好な反応を得ることができた。

ただ、試作したアプリケーションは、あくまで、本研究成果としての情報モデルの応用可能性および有用性を占う目的のためのものであること、従ってこれを現状のまま臨床現場に使用するわけではないことについて、誤解した回答も無くはなかった。その事由は以下に拠るものと思われる：

- ・試作アプリとしては出来過ぎていたため、逆に実業務用の電子カルテと混同されたこと
- ・MS Windows と MacOS の human interface の差による違和感
- ・臨床思考の困難さと、human interface の使い勝手との混同
- ・臨床現場で使用する際に、全ての症例の全診療セッションにおいて、常に熟考を求めるような human interface の使用を強要されるかもしれない、という誤解
- ・TheTOOL を試用に供さなかったこと

などが考えられる。詳細については、分担研究報告書（植田）を御参照願いたい。

一方、臨床実習生に粗方の説明を施したうえで TheTOOL を見せたところ、即座に、その理念と目的を理解して、興味深いと答えた者が大半であった。

加えて、勉強になるツールなので臨床実習での難症例の検討の際に使いたいとの回答を得た。その一方で、診療システムを臨床現場にて使いこなすのは難しそうである、と答えていた。

これらの回答に関する“乖離”の原因としては二点を推測することが可能であろう：

- ・通常の HIS の human interface とは異なってどちらかといえば素っ気無い画面構成であること。
- ・POMR や SOAP の記述は熟考を要求する機会が多いが、それと同様に、病名の変遷や病名診療行為の連関を正確に記述入力することもまた熟考を要する作業であること。

ただ、このような回答は、質問する前から既に予測していた事項に過ぎない。

前者は“rich”な human interface のほうが使い易いように感じられる錯覚、試作実装のコスト制約、そして好みの問題などであろう。

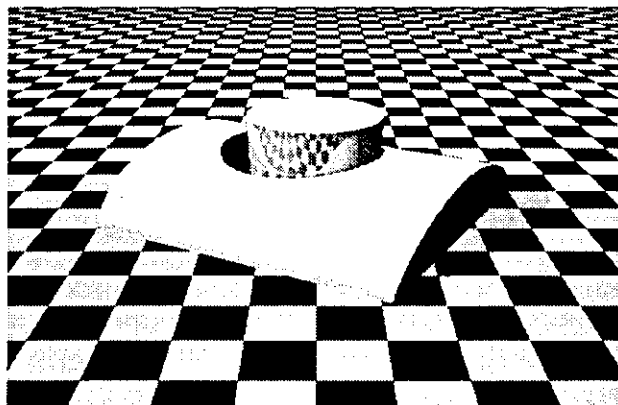
後者は、逆に云えば、現状が如何なるものかを示唆しているようにも思えるところではある。

C. 16 他の試作実装とその意義

C. 16.1 Web アプリと視覚化

神田は RDBMS を利用した Web アプリケーションを開発し、さらには蓄積された診療データを抽出して三次元画像化した。概要は以下の通りである：

- ・Apache と PHP で生成したフォームに入力された診療データは、PXBASE によって PostgreSQL に CSX ontological XML Schema 準拠の xml 形式で格納する。
- ・蓄積データを CSX XML Schema に準拠した xml ファイルとして抽出する。
- ・一方、光源位置、カメラ位置、そして客体の三次元 rotation に関するパラメータ緒元も CSX ontological XML Schema で記述する。
- ・これを 3D graphics framework を利用して視覚化する。

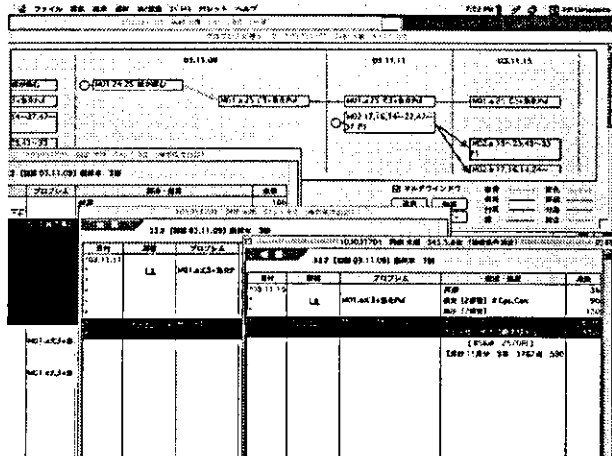


この試作システムは、本研究主題とは趣を異にするものの、本研究の発展性を立証するものとして意義深い：

- ・POSIX 環境における実装可能性の立証
- ・廉価なシステム環境における実装可能性の立証
- ・診療情報の二次利用における CSX XML Schema 形式の有用性の立証（視覚化を含む）

C. 16.2 診療行為連関

矢嶋は CSX model の旧版を交換書式として用い、病名/プロブレム変遷や病名-診療行為連関に関する情報交換を実現する開発を行って、実用段階にまで至った。



この試作実装は、既に市販されているシステムへの搭載に関しても十分に可能であることが立証された、という点で意義深い。

D. 考察

D. 1 CSX model デザイン

D. 1. 1 経緯

本研究の源流は (B) に記したように単一ではないが、その本流は (B.3) である。

基とした情報モデルは、もともとは物理空間に存在する実体や仮想体あるいは構成体の詳細記述ならびに、それらの間の関係の仔細を表現するために構築された (B.8)。

それは ontology (C.1.1) として仕上げられたことから、これを洗練させつつ変遷モデル (B.2) と思考空間 (B.3) とに適用した。他に選ぶべき候補が無かったためである (D.2)。

D. 1. 2 特徴

主要点を要約する (C.2.2) (D.2.8) :

(i) 情報モデルとして極めて小さい。(ii) よって頑健である。(iii) また domain 非依存であり、domain 依存部分は全て meta-attribute の値枠組に分離している。(iv) 関係と関係視座〔視野〕を明示している。(v) 提供している meta meta-information object が格納する属性値に応じて、それらは、そのまま instance としても機能する。そして (vi) 再帰的な構成によって微細粒度から大域粒度まで境界なく単一情報モデルにて事物事象を表現できる。したがって、(vii) 複数モデルに拠るモデリングの際に界面に発生する汚染と混淆を回避できる。(viii) 先験的な軸性を情報モデル自体が暗黙的に規定せずに、むしろ自在に設定しうる環境を提供する。これらの特徴は全て、(x) ontology に即し・かつ・(xi) 抽象空間に対して必要に応じて位相構造を与えうる枠組を提供していること、に由来している。

D. 1. 3 自由と規律

CSX model は高い表現自由度を有しているが、これのみをもって直ちに有利と断ずることはできない。

むしろ適度な制限制約は、健全な記述や表現を保つためには必須でさえある。CSX model では、そのような籓 (あるいは規律) が二つある。

その一つは arcScope によって嵌められる視座意義や視野範囲、一つは要素属性に格納される

値は code schema 内に定位する code の適用を想定している、ことである。

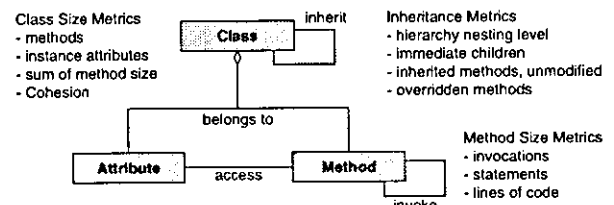
模式的に言い換えるなら、多重グラフなる宇宙で、踊り子は奔放に関係を結んで踊りまわられるように見えるものの、実は arcScope で「関係範囲や関係の意義」なる籓を嵌められ、meta-attribute にて「code schema 内に定位」させられ足元を縛られ、結局 “正しいあり方” のみが有効となる。

すなわち、意味の無い、あるいは妥当ではない情報塊は構成できない、または廃棄されうる。

しかしやはり自由度は高いのである。踊り子が、ではなくて、宇宙のなかに、自在に舞台空間を作りあげ、そこに様々な踊り子を配置することにおいて。

D. 1. 4 処理負荷予測

計算機処理の負荷は、モデル構造とその大きさに強く依存している。処理負荷予測に関する model metric のうち、計測的な項については、CSX model は、極めて小さい。

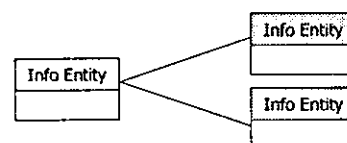


外部参照については属性値探索が該当するが、code を前提するならば処理時間は予測容易であり、また (C.8) (C.10) に記した如く局面によって既に focus されている。

結合度については arcScope の周りで結合度が高くなるが、infoArc は単一の infoNode しか指示しないし、infoNode は自身を規定する arcScope@uid を infoNode@bearing に格納して明示し、また (C.8.3) の事情もある。よって CSX model が計算に多大な悪影響を及ぼすとは思われない。またそれは実証された。

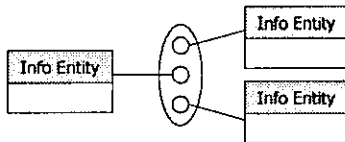
D. 1. 5 関係の結びかた

ある Class が数種の Class と関係を結ぶことは頻繁である。UML では次図のように表現される。



ただ此処には、そのような関係が結ばれた視野範囲や視座意義は明言されてはいないことに留意すべきである。

一方 CSX model では (C.2) およびそれ以降に示したように関係を結ぶ。なお直列化に際しては以下の diagram で示されるように記述されることになる (C.3.4)。



楢井は arcScope を、小井は infoArc を表現しており、entity は四角のみである。

このような表現形式を不自然と感じる向きもあるかもしれない。ただ仔細に検討してみると別段の問題を生じないように思われる。

- ・二項関係（三項組み）であれば、つまり Class が 2 個であれば、問題はない。
- ・多項関係（修飾）であっても、被修飾体（と修飾体）の関係であれば問題ない。
- ・多項関係のとき、親子関係であれば問題ない。
- ・多項関係のとき、親子関係であり、かつ子の間に順序性が存在していても、何ら問題はない（topology で表現する）。
- ・多項関係のとき、構成体（と構成要素）の関係であれば問題ない。
- ・多項関係のとき、移動や変遷のように源泉（と対象または目標）の関係であれば問題ない。
- ・多項関係のとき、arcScope が述語を表現し、infoArc が深層格関係表現しているのなら問題なく、むしろ自然でさえ、ある。
- ・多重グラフにおいて（抽象的な）経路設定や制約を記述する際、そのような場を表現する infoNode が存在するならば問題ない。

つまり考察と例証を繰り返してみたが、反例を発見することは無かった。

このことは、視野範囲や視座意義が限定されたうえで Class 間の関係が結ばれる際、その focus は、或る scope のもと唯一つの Class に絞って考えざるをえない、その様な思考のあり方を示唆しているように思われた。

そして arcScope は、その focus を、@category と @kind に格納している。よって、この直列化手法を却下せず、採択のままとした。

D. 1. 6 n-arity

以前の研究過程では、n-arity について非常に

気に懸けており、di-arity（二項関係：三項組み）を基本とするよう意識していた。しかし、そのような態度はやめることとした。

というのも、比較(表現)や共参加者がある場合には、node 対 node という二項関係の「累積」のみでは、正確には表現しきれない事例もあるからである。

現実に存在する、そのような意味関係を有する意味構造を安易に無視するのは妥当ではないし、また極度の捨象や単純化には慎重な態度であらねばなるまい。

というのも CSX model は現実世界を写しとろうとしている以上、その表面上の形式が抽象的であったとしても、個々の情報素や関係素そして関係視座素が宿しうる多様な色彩にも留意せざるをえないからである（例えば D.1.5）。

D. 1. 7 制約と輻輳と隔離と

制約表現の付与構造を設計する際、依存と排他には深い考察を要した。

多重グラフ構造を成す対象世界における依存と排他とに関する制約表現の困難さは、arc の有向性の有無に関わらず次に拠っている：

- (1) node における多入出力すなわち輻輳
- (2) 隔離 node 間の依存と排他の管理

この事情は一群の node を一つの部分的な界に整理したうえで node 関係を制約したとしても、変るところはない。

このような事由から (C.2.7) の如き制約表現とした。

D. 2 他のモデルとの比較

D. 2.1 用語と概念の整理

CSX と他者との比較をする前に、まずは用語と概念を整理しておく必要がある。

Ontology の意味は (C.1.1) に示したとおりである。

しかし巷間では誤解されているように見受けられることがある。その事由は、以下の各語を弁別していないためと思われるので、ここに整理する。

- ・ **grouping**
散り散りな事物の幾つかを集めて群と為す。
- ・ **category**
広がりを区切る。
CATA- down, against, thoroughly
AGORA marketplace
- ・ **classification**
予め想定した尺度や規範、つまりスキーマに準拠しつつ、品質や品格に即して分ける。
- ・ **systematics**
分類学または体系学の一般的な謂い。
- ・ **taxonomy**
階層性を前提した分類(体系)または分類学。
もともとは生物学から。
- ・ **ontology**
形而上学の一分野。
事物事象 および その 関係の様相 について、
それらの背後にある原理 (それらを上から支配している原理) を考察する学。
- ・ **metaphysics**
形而上学。事物とその性質の 在り方 の探求。
The nature of reality, including the
relationship between mind and matter,
substance and attribute, fact and value.
最初に在る根源的な原理 (または規律や統制)。
先験的思念 (科学的な観察や解析の届かない)。

更に meta-thesaurus や semantic network との差異も押さえておくべきであろうが、言語学や言語処理に関する事項と絡み紙幅が嵩むので、ここでの叙述は割愛する。

以下、本研究成果である情報モデルと関係する事項に焦点しつつ、他との比較を記していく。

D. 2.2 GALEN

GALEN は、言語と推論処理系に非依存のオン

トロジ (building block) を目指して、永年にわたって構築努力が為され続けている。複数の階層系 (ツリー) を持つことを許容している。

言い換えれば各 concept は、いずれかのツリーのどこかに定位されており、その定座位置は一箇所とは限らない。座標系を一つに固定したり、座標を一つに限定したりすることは不自然かつ妥当ではない、という根底理念に基づいているのであろう。

各 concept は attribute によって結合され、kind-of は、part-of や caused-by などから注意深く分離されているが、その種は膨大である。なお SELECTOR は、興味深い機構を示す工夫となっている。

収録されている概念数は概ね 7,000 であり、GALEN はその経験から、複数の親を持つ概念は 40%弱であり、三つ以上の親を持つ概念さえ 8% あることを教えている。

CSX model との差異は、CSX model は SCOPE を記述する要素を持っていること、逆に CSX model は情報モデルの提供であって building block は提供していないこと、である。

D. 2.3 Protégé

Protégé は、基本的には building block を構築するための editing tool の提供、ということである。

Protégé なる editing tool が為しうることは taxonomy の構築が主である。Taxonomy には Frame が配置され、Frame は Slot を用いて horizontal relation (association) を形成することも可能である、としている。

しかし唯一の node tree での世界構成に過ぎず、multi-taxonomy のなかで関係を構成することはできない (但し multi-super class には対応している)。

また horizontal relation の厳密な定式は低く (=無く)、制約の記述も比較的貧弱である。

そのうえ Protégé は、taxonomy における Class (または Frame) の継承関係の支援に強くない。この点について Protégé-2000 の開発提供者側は、ユーザにおける運用での回避を要求している。

また Protégé-2000 は、OWL での出力をサポートしている。しかし逆に OWL の限界に束縛されている。たとえば、現状の OWL では、n-arity は表現できない。よって Protégé-2000 内部で多項

関係を表現できたとしても、OWL へ出力した際には多項関係情報は失われてしまうのである。

知名度が高いことと、真に有意義なこととは、注意深く弁別する必要があるだろう。

CSX modelとの差異は、CSX model は SCOPE を記述する要素を持っていることに加えて、CSX model は multi-taxonomy も n-arity も表現できること、である。

D. 2. 4 OWL/RDF

RDF は、もともと相互運用可能な形で一つの「リソースを記述する」枠組を提供する試みである。利用者が拡張解釈しようとも、RDF の適用範囲とは、本来は其処まで、である（今後の展開によっては変化するだろう）。

RDF は Notation 3 (N3) を礎としているものの、そのサブセットに過ぎない。N3 とは、二項関係（三項組み）である。

RDF は、その語彙を Dublin Core ほか基本的に書誌事項に限っている。近年では、UMLS も登録されたようだが、UMLS は膨大な体系「集」であるにも関わらず、RDF における明確な使用法についての言明は見当たらない。

いずれにせよ現状の RDF にて多彩な多項関係をすぐに自在に構築できるわけではない。

OWL は、RDF を基礎において、用途を限定して構築されたシンプルな ontology である。

OWL を知識 ontology と目する向きもあるとはいえ、当初の目的は web でのリソース特定と、書誌事項の記載であった。つまり web 環境での共有性や相互運用性を主目的としていることが、その設計方針と仕様にも明言されている。

また OWL は、RDF の N3（しかもサブセット）に影響されて、多彩な多項関係を自在に表現することはできない。その改善が試みられているが、NOV/2004 時点でも未だ first draft 段階である。

CSX modelとの差異は、CSX model は SCOPE を記述する要素を持っていること、また CSX model は n-arity を表現できること、である。

D. 2. 5 MOF/CWM, UML/OCL

MOF は、UML によって記述されたモデルをメタメタ表現する試みである (C.1.2)。

その目的とするところは CASE における meta modeling である：言い換えれば MOF を用いて異システム間の双方の要素を記述し、その上で

双方のタプルなどのマッピングを支援する、という枠組である。

CWM は、N3, MOF, XMI, XML, OCL を基にして、同様の目的で用いられる。

これらのことから、**CSX model** とは目的も枠組も異なっていることが直ちに判る。

さらには次の (D.2.6) でも述べるが、少なくともモデル構築においては、multi-model methodology は本質的な脆弱性を内包しているように思える。

CSX model は、最細粒度から最大粒度まで、また事実から規則知識まで、一つの情報モデルで記述することを目標としており、実際に前者は既に証明済みである。根源的な差異である。

D. 2. 6 OpenEHR と Archetype

Archetype

Archetype とは、構成体の要素の、要素原型を意味する。情報システムの場合、システム設計とシステム構築に資する基本 entity の記述と、その entity に関わる種々の制約の記述を云う。

Two model methodology

Information model (IM) あるいは reference information model (RIM) は、システムアーキテクチャを設計する際に極めて有用であるものの、その domain や realm における大概を説明するのみである。

したがって個々の entity および制約属性まで言及するのは困難か、あるいは RIM 内に細かなデータ構造と制約規則とを同時に持たざるをえないことになる。後者の場合、RIM は巨大化するとともに情報モデルと知識表現との分離性が劣化して、維持性が低下することになる危険性を孕む。

そこで IM や RIM に加え archetype を活用してシステム設計する手法 two model methodology が提唱された [Beale; DeepThought, Ocean Informatics, OpenEHR]。

Archetype Definition Language

そのような手法枠組を支援するために ADL (Archetype Definition Language) が開発・提唱されているが、これは XML を敢えて使用していない。その事由は、XML syntax による記述制約の呪縛から逃れるためである、としている [Beale; DeepThought, Ocean Informatics]。

XML も XML schema も、それらが根源的に持たざるをえない syntax に拠る記述制約は小さくないものの、Beale が参照した枠組は RDF, OWL, OCL であって、CSX ではない。

CSX model は single model methodology を前提としている。

というのも、モデル構築において境界を設けた瞬間に、その界面の近傍では、必ず概念汚染を生じうるからである。すなわち、どちら側の model を用いてモデル構築すべきなのか、という問題を完全かつ自動的に回避解消するのは不可能だからである [Hofstadter, Gödel, Escher, Bach -An Eternal Golden Braid]。

CSX model は直列化手法に XML を用いることを許容している。

さらに CSX model は通常想定されるモデリングよりも細粒度または大粒度の対象をモデル化する場合、Class と Attribute の弁別は、曖昧または恣意的になる傾向があることを意識しながら構築されている。

そして制約表現能力を獲得した CSX model は、archetype を記述することも可能となった。

D. 2. 7 UMLS

UMLS との比較は容易ではない。

まずは“Resource”とは対比しえないことは、自明であろう。CSX model は (C.1.1) における δ の立場だからである。

一方、“Metathesaurus”と“Semantic Network”については ontology 的であるし、そのように云う向きもあるだろう。ただ NML 自身は、これを ontology と呼ぼうとはしていない。

その事由の詳細は明らかではないものの、賢明なる思慮が含まれているように思える。

CSX model との差は種々あろう。UMLS では言語学または言語処理的な色彩の強い属性ほかを有しており、また通常の意味での Class と Attribute による構成となっている。

CSX model は SCOPE を記述する要素ほか、根源的な要素と枠組を提供している。それらは Generic (通常の Class や Attribute 等) にも instance にも成りえて・そのように機能する、そういう情報モデルであり、枠組である。

D. 2. 8 CSX model 再び

CSX model は ontology のうち情報モデルなので、そこに焦点して他との対比を要約する。

CSX model の特質特徴は既に (C.2.2) (D.1.2) (D.1.3) に述べられている。ここで強調すべき点があるとするならば、次の事項であろう：

CSX model は meta meta-information frame work として全ての粒度性と軸性とを排除しながら meta meta-information object を提供している。粒度性と名目とはコード・マスタから獲得し、軸性は関係視座を与えて構築する。

つまり、個々の抽象要素が自身を具体化する環境と、抽象空間に位相構造を定義する環境のみを提供している。

したがって meta meta-information frame work 自体には何も無い、とも言う。しかし逆に、全ての構造と要素とを生み出す力を内包している。

なお 構造の生成と解釈には 多大なコストを要すると思われがちだが、再帰性と同型対応によって現実時間内、しかも実業務時間内に処理可能としているのである。実際、それらは参照実装に拠って証明された。

D. 3 実装成果と応用可能性

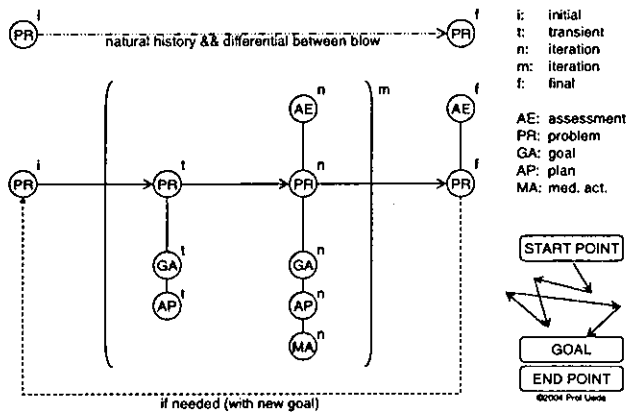
D. 3.1 病名変遷と診療行為連関

病名/プロブレムの変遷, ならびに病名/プロブレム-診療行為連関を具現する参照実装を完遂し, CSX model の有用性を検証し終えた。

病名/プロブレムの変遷や, それらと加療行為との連関に関する汎用的な記述モデルおよび実装による検証は他に無いので, 本研究成果は貴重である。

D. 3.2 診療経過の一覧と要約

上述のような関連づけされた臨床経過を活用すると, (C.11.3.2) (C.14) に示したごとく, 経過一覧やその要約は極めて容易となる。既に症例検討や臨床研究あるいは evidence 収集に有益なことは明らかである。



上図と (C.14.2) に提示した図の相似性を比較して戴きたい。

また (B3) の思考過程概念モデルと比較すれば, この思考過程モデルが有益な礎を与えていることも御理解いただけるだろう。

D. 3.3 臨床教育

このような図式化が可能であるということは, 臨床教育にも有用であることを示している。

たとえば若手の医師にとっては, 成功例と失敗例の差異を検討し, また, 何故に差異が現れることとなったのかの事由を発見するとか, 臨床成果を分かち重要な臨床時期を勘案することなども, 容易となろう。

事実, 臨床実習の学生諸君はその有用性を即座に見抜いている (C.15)。

よって主任研究者の所属する機関で他の基礎講座や臨床講座と協同しながら, 本研究で試作したアプリケーションを活用しつつ統合的で連携した医学教育の方策と方向性を探ろうと計画している。

D. 3.4 診療経過モデルの生成

また (D.3.2) から得られた知見を基にすれば, 診療経過モデルや標準治療計画を策定することも容易となる。

しかもこれらの知識と経験は, 可搬性の高い xml 形式で共有できるのである。

D. 3.5 標準治療計画やパス

制約表現については, その定式化を終えたばかりではあるが, 制約表現を加えたなら, 標準治療計画やパスに関する表現力も豊かになる。

つまりスレッドやバック [Proc JCM197, 504 - 505, 1997] を用いたユーザフレンドリな標準治療計画策定編集画面なども実現できることになる。

もともと, 現時点でも種々のリソースが与えられたなら, 現状モデルで相当程度の実現が可能ではある。ただ制約表現を駆使した実装すれば, より実用的となる。

D. 3.6 知識とルール

そもそも, 視野を明確にしたうえで関係を形成し, その情報の群の意味を捉えることは, 知的活動の基本的なパターンである。よって CSX model は知識の表現と交換に大いに活用でき, また貢献しうるのである。

一方, DPC や点数表における各種の規則や制約条件を記述したなら, 経営管理などに役立てることも容易であろう (A.2)。さらに, 自由度に富み分離性も良いことから, システム・メンテナンス・コストも縮減可能と思われる。

D. 3.7 データマイニング

今迄のデータマイニングとは, 分解されて孤立した atomic なデータに対し確率的な手法を適用することで実施されてきた。

しかしその手法は, 医療に関しては効率的とは云い難い。というのも, 診療現場にて発生するデータや情報は将に現場医師が「関連させつつ考えた出力」にほかならないゆえに, その一部でも関連性をキャプチャできたならば, データ

マイニングの効率も確度も、格段に向上しうるからである。

本研究成果は、そのような資料を蓄積し自在に出力できることを実証した。

ただし本研究が提供するものは、グラフベースである。これを効率的に使うにはレガシな手法ではなく、グラフベースに即した新たな手法が必要となる。

しかし、其の様な手法を憂慮する必要はなく、すでにグラフベース・データマイニング手法が精力的に研究されて応用段階に差し掛かっているのである。

D. 3. 8 自然言語生成

さらに (D.3.2) から得られた知見を基にすれば、infoArc や arcScope は関係に関する種々の意味を含んでいることから、そのような情報も活用しつつ自然言語を生成することさえ、近い将来の実現可能性を秘めている。

医療現場において具体的に何が出来るのかというと、たとえば、完全に事実に基づきつつ、しかも(半)自動的に、診療経過の要約の作成を機械処理しうることを意味する。

このようなパラダイムは (D.3.7) の前半にて示した事情と同様、ヒトの思考と機械処理との「架け橋」において極めて効率的な環境を提供することになりうる

D. 3. 9 役割配役立場モデル

本研究から、以下が明確となった：

- a) 権限管理とは権限根拠の管理のことである。
- b) 権限根拠は、場において都度、生成・獲得されるという観点から事象要素を分離して再統合する必要がある。

このドグマは重要にも関わらず看過され続けてきた。その事由は幾つかあろうが、今後はこの点に留意しつつ、privacy と confidentiality とを保持する機構を考案する必要がある。

よって応用可能性は広汎であり、権限管理機構の実装設計は勿論のこと、CP/CPS (Certificate Policy/Certificate Policy Statement) での権限管理など、各種のセキュリティ・ポリシーの策定に活用できるし、またそのように応用すべき概念モデルである。

D. 3. 10 臨床試験と evidence

本研究主題に即した、また本研究成果に即して考察された臨床試験および evidence 獲得への応用可能性の詳細については分担研究報告書(植田)に詳しいので、それを参照願いたい。

ここでは、その範囲事項を列挙するに留める：

- ・患者の抱える問題の記述
- ・それらの問題の変遷
- ・問題の解決に関する記述 (目標や介入)
- ・診療目標と診療成果の比較可能性
- ・前向き(ランダム化)臨床試験に要する記述
 - ・適合基準と除外基準と中止基準
 - ・遠位と近位の診療目標の記述
 - ・一次と二次の加療終点の記述
 - ・診療ベクトルの記述
 - ・診療アウトカムの記述
 - ・診療アウトカムと診療目標との比較
 - ・同意書の取得に関する機能要件
- ・後向き臨床研究の実展
 - ・診療アウトカムからの後向き追跡
 - ・診療品質の評価
 - ・問題/目標/介入/成果の後向き評価
- ・Basso Continuo の記述
- ・経験知識の evidence 化の可能性の付与
- ・コンフィデンシャリティに要する機能要件
- ・診療スタッフ間の情報共有に関する機能要件
- ・地域医療に即した情報枠組

D. 3. 11 Graph pane

これは CSX model に限らず、ontology を下敷きとした様々な情報モデルの視覚化、あるいは、他の用途にも応用しうる。

二十一世紀は知の競争時代と云われるなかで、今後の情報システムは単に業務遂行できれば良いというものではなく、孤立情報の蓄積でもなく、むしろ要素間関係、つまり関係のなかに表出されうる知の集積こそが重要なのである。

このような状況に鑑みても、GraphPane の如きモジュールの応用可能性は広汎であろう。

これらのことから、本研究成果の応用可能性は広汎、かつ今世紀の知の発展の流れに即しつつ種々に貢献できるものと思われる。

またその一部は既に本研究成果において実証済みである。

D. 4 今後の課題

初年度に残された課題は全て解決した。よって、その意味において「遣り残し」は無い。

とはいうものの以下の事項も整えていく必要もある。

なお下記に挙げた事項以外にも主任研究者として理想を求めたい細かな点は幾つもあるが、それらについては仔細に過ぎてしまう嫌いを免れないので割愛することとした。

D. 4. 1 エントリツール

本研究の試作実装は病院や診療所の診療システムとして必要な全ての Entry Tool を揃えておらず、研究主題の成果を立証できる最低限の範囲内に留めている。

よってこれを拡充していく必要がある。

D. 4. 2 編集ステータス

現状の実装では、病名/プロブレムを FIX して加療行為を入力できるようにしている。しかし実際の診療あるいは臨床思考過程は、実はそのようにはなっていない。

主任研究者ほか関係者はこれを意識しているものの、本研究の実装では「行ったり来たり」を支援する機能を割愛した。

というのも、そのような機能を実現するには、非確定 object に依拠した別の object に対する編集操作を許容することを意味しており、そのような状態においても文脈整合性を保持するためには、編集ステータス管理、そして個々の object の@uid の仮附番と確定附番など、実装工数が格段に増加して、予定コストを超過してしまうためであった。

しかし end-user にとっては必須と感ぜられる機能だろう。よって本機能を実現したい。

D. 4. 3 制約表現の実装

制約表現に関する定式化も終えた (C.2) ので、これを実装して検証したい。

題材には二つの候補を挙げるができる：
(i) DPC や点数表が含む制約条件の記述と適用、
(ii) 標準治療計画やパスの記述。

D. 4. 4 3C model の詳細化

3C model (役柄配役立場モデル) については、参照情報モデルとしての詳細なクラス設計は、社会状況からしても急務であると考えている。

よって、これを進めたい。

D. 4. 5 HL7 準拠メッセージ

本研究の成果である CSX model は transaction については対象範囲外としている。それに係る messaging は HL7 に準拠するよう考えているので、このあたりの harmonization も今後必要となろう。

D. 5 今後の展開

D. 5. 1 診療方向性に基づく追跡

本報告書 (C.14) に記したツールはそれだけで意義深い。ただし、その入力を生成し出力する診療システム (C.11)(C.12) は診療ベクトルを記述検証するに十分な機能を備えているとはいえない。

これは human interface の課題というよりも、目標設定と(再)評価における診療プロセスの定式化と実装が不十分だからである。

研究と開発は一步ずつ進めていくべきなので、この点は、本研究主題の対象とはしなかった。しかし今後は、これを進めたい。と同時に試作した診療システムや TheTOOL を洗練させていきたい。

D. 5. 2 述語の扱い

本研究では、述語と深層格による表現を主体に考案を重ねてきた。これはこれで有用であるし、実際に一部の記述には既に応用済みである。

ただ今後は語彙意味論 (Lexical Semantics) またはその派生の応用も併せて検討していく予定である。

D. 5. 3 継承の扱い

この点は情報モデルの設計における「モデル記法」というよりも、むしろアプリケーションへの実装課題として捉えるべき事項であると思われる。

たとえば UML では「△」と記述するだけで表現されたことになってしまう。一方、CSX model では、下位 infoNode は上位 infoNode にて記述された制約を受け継ぐことを前提している (C.2.2)(C.2.7)。

いずれにせよ、継承自体は重要な概念であり、さらに継承の逸脱もまた同様に重要である故、これらの記述についてもさらに検討していきたい。

D. 5. 4 ASN.1 による定式化

本研究では情報モデルの記述には UML や XML を用いてきたが、むしろ ASN.1 を採用したほうが記述しやすい面もあるように思える。よって、これを試みようと考えている。

D. 5. 5 思考過程のモデル化

上記のような努力を続けながら、最終的には、思考過程のモデル化を進めていきたい。

D. 5. 6 ビルディングブロック

上記の事項を踏まえつつ、情報モデルとしての ontology のみならず building block としての ontology を構築したいと思う。

ただしこれには多大の人的コストとともに、長い年月を要する。そのようなリソースを獲得できれば挑戦してみたい。

E. 結論

本研究の成果は汎用的であり、理論あるいは根拠に根ざした合理的な診療情報システム設計から緊急性のある課題の解決まで、広い射程を有している。

E. 1 モデルの意義

臨床思考過程概念モデル, CSX model (ontology と meta-modeling framework) ; 役割配役立場モデルはいずれも、診療現場、臨床研究、臨床教育に意義深く重要有用であることが考察において検証され、実装において実証された。

E. 2 真に有用なる一次データ

診療に関する一次データの品質は、医療施策の策定や経済財務の評価を大きく左右することになる。

— 本研究成果とその方向性は、適切な調査や評価に資するものである。

E. 3 真の evidence の獲得

臨床試験による evidence の獲得は診療品質の維持と向上に必須である。但し診療ベクトルの合致性を検証する研究デザインであり、加えて診療ベクトルの合致性に基づいたデータ解析が為されて、その意義と応用可能性が保証される。

— 本研究の成果は、その様な努力と資源投入効果の極大化に対して貢献するものである。

E. 4 監査可能性

我国の既存または現状の診療情報システムは、“診療行為”の監査を想定しないものが大半である。

— 本研究成果とその方向性は、監査と証跡に関わる実装に資するものであって、診療品質はもとより、個人情報保護を適切に管理する礎を与える。

E. 5 臨床教育と知識表現

論拠性ある行為の経過を記録記述することは、まさに、経験と知識とを蓄積することと同値であり、今世紀の本邦ほか OECD 諸国の医療行政にも強く求められている。

— 本研究成果は、事実や経験を知識へと昇華させる環境を与えつつそれを促すものである。

よって臨床教育に資するところ大であり、また診療記録形式と知識表現形式を統一的に扱いうる枠組は、知の大競争時代に有用かつ重要である。

F. 健康危険情報

ない。

G. 研究発表

- 1) 廣瀬康行. Ontology 的分析により構築した記述モデルによる病名やプロブレムの変遷の表現可能性. 医療情報学. 23S:962-966, 2003 (2003年11月)
- 2) 廣瀬康行. 関係者と組織との諸関係を記す 役割配役立場モデル. 医療情報学. 23S:504-507, 2003 (2003年11月)
- 3) 矢嶋研一, 廣瀬康行, 森本徳明, 佐々木好幸, 成澤英明, 尾藤茂. 診療履歴情報とプロブレムの ontology 的リンクモデルと電子カルテシステムへの適用例. 医療情報学. 23S:800-801, 2003 (2003年11月)
- 4) Yasuyuki Hirose. Tiny and Compact Meta Meta-information Model. MEDINFO 2004:1640, 2004 (2004年9月)
- 5) 廣瀬康行. 制約類型と CSX Ontological XML Schema による表現. 医療情報学. 24S:816-817, 2004 (2004年11月)

H. 知的財産権の出願登録状況

現時点では、ない。

以上

資料 1

CSX XML Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- created by Yasuyuki Hirose, MedInfo, Ryukyu Univ Hosp on 14-Feb-2003 0.90 -->
<!-- revised by Yasuyuki Hirose, MedInfo, Ryukyu Univ Hosp on 08-Oct-2004 0.96.9 -->
<!-- revised by Yasuyuki Hirose, MedInfo, Ryukyu Univ Hosp on 17-Jan-2005 0.97 -->

<xs:schema
  targetNamespace="http://www.hosp.u-ryukyu.ac.jp/medi/csx/0.97/"
  xmlns:csx="http://www.hosp.u-ryukyu.ac.jp/medi/csx/0.97/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="0.97">
  <xs:include schemaLocation="fcet.atst.csx.xsd"/>

  <xs:element name="facet">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="csx:infoNode"
          minOccurs="1" maxOccurs="unbounded"/>
        <xs:element ref="csx:arcScope"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="csx:description"
          minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
      <xs:attribute name="version"
        type="xs:token" use="required" fixed="0.97"/>
      <xs:attribute name="oid"
        type="xs:NMTOKEN" use="optional"/>
      <xs:attribute name="category"
        type="csx:facet.category.Type" use="optional"/>
      <xs:attribute name="facetID"
        type="xs:NMTOKEN" use="optional"/>
      <xs:attribute name="facetNum"
        type="xs:normalizedString" use="optional"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="infoNode">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="csx:nodeCode"
          minOccurs="1" maxOccurs="unbounded"/>
        <xs:element ref="csx:construe"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="csx:dimension"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="csx:description"
          minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
      <xs:attribute name="uid"
        type="xs:ID" use="required"/>
      <xs:attribute name="oid"
        type="xs:NMTOKEN" use="optional"/>
    </xs:complexType>
  </xs:element>

```

```

<xs:attribute name="rid"
<xs:attribute name="bearing"
<xs:attribute name="category"
<xs:attribute name="kind"
<xs:attribute name="existence"
</xs:complexType>
</xs:element>

<xs:element name="nodeCode">
<xs:complexType>
<xs:attribute name="CSname"
<xs:attribute name="CScode"
<xs:attribute name="CSver"
<xs:attribute name="priority"
<xs:attribute name="codeName"
<xs:attribute name="code"
</xs:complexType>
</xs:element>

<xs:element name="construe">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="NSname"
<xs:attribute name="NScode"
<xs:attribute name="NSver"
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>

<xs:element name="arcScope">
<xs:complexType>
<xs:sequence>
<xs:element ref="csx:infoArc"
<xs:element ref="csx:comment"
</xs:sequence>
<xs:attribute name="uid"
<xs:attribute name="category"
<xs:attribute name="kind"
<xs:attribute name="choice"
</xs:complexType>
</xs:element>

type="xs:anyURI"
use="optional"/>
type="xs:NMTOKENS"
use="optional"/>
type="csx:infoNode.category.Type"
use="required"/>
type="csx:infoNode.kind.Type"
use="optional"/>
type="csx:infoNode.existence.Type"
use="optional"/>

type="xs:normalizedString"
use="optional"/>
type="csx:nodoCode.CScode.Type"
use="required"/>
type="xs:normalizedString"
use="required"/>
type="xs:int"
use="optional"/>
type="xs:normalizedString"
use="optional"/>
type="xs:NMTOKENS"
use="required"/>

type="xs:normalizedString"
use="optional"/>
type="csx:construe.NScode.Type"
use="optional"/>
type="xs:normalizedString"
use="optional"/>

minOccurs="1" maxOccurs="unbounded"/>
minOccurs="0" maxOccurs="unbounded"/>

type="xs:ID"
use="required"/>
type="csx:arcScope.category.Type"
use="required"/>
type="csx:arcScope.kind.Type"
use="optional"/>
type="csx:arcScope.choice.Type"
use="optional"/>

```

```

<xs:element name="InfoArc">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="csx:topology"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="ref"
      type="xs:string" use="required" />
    <xs:attribute name="category"
      type="csx:infoArc.category.Type" use="required" />
    <xs:attribute name="kind"
      type="csx:infoArc.kind.Type" use="optional" />
    <xs:attribute name="multiplicity"
      type="csx:infoArc.multiplicity.Type" use="optional" />
  </xs:complexType>
</xs:element>

<xs:element name="topology">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="csx:orientation"
        minOccurs="0" />
      <xs:element ref="csx:dimension"
        minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="path"
      type="xs:int" use="optional" />
    <xs:attribute name="option"
      type="xs:NMTOKENS" use="optional" />
    <xs:attribute name="request"
      type="xs:NMTOKENS" use="optional" />
    <xs:attribute name="negate"
      type="xs:NMTOKENS" use="optional" />
    <xs:attribute name="hop"
      type="xs:NMTOKENS" use="optional" />
  </xs:complexType>
</xs:element>

<xs:element name="orientation">
  <xs:complexType>
    <xs:attribute name="direction"
      type="csx:orientation.direction.Type" use="required" />
    <xs:attribute name="coordinate"
      type="csx:orientation.coordinate.Type" use="optional" />
  </xs:complexType>

<xs:element name="dimension">
  <xs:complexType>
    <xs:attribute name="tude"
      type="csx:dimension.tude.Type" use="required" />
    <xs:attribute name="unit"
      type="csx:dimension.unit.Type" use="required" />
    <xs:attribute name="equivalent"
      type="csx:dimension.equivalent.Type" use="optional" />
    <xs:attribute name="measure"
      type="xs:normalizedString" use="required" />
    <xs:attribute name="dataType"
      type="csx:dimension.dataType.Type" use="optional" />
  </xs:complexType>

```