

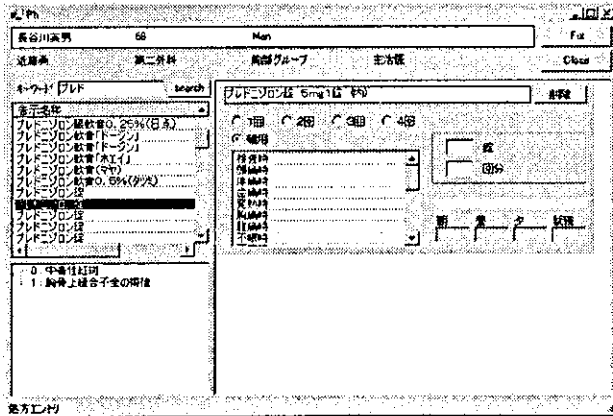


### C. 15. 5 処方 OrderTool

画面例を列挙する (実装は大嶺).

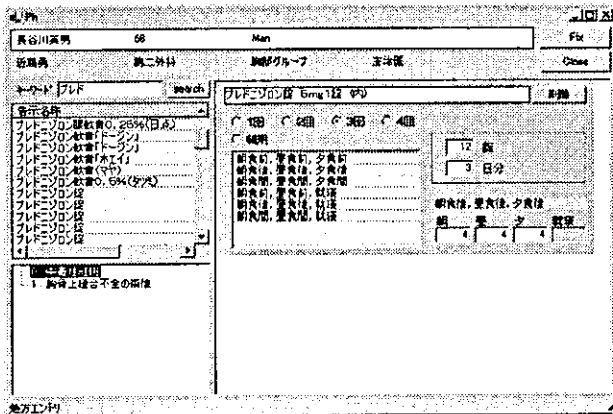
#### C.15.5.1 Search

画面左上方で検索し, 検索結果をリスト表示させ, その中から選択する.



#### C.15.5.2 Instruction

画面右手で, 定時/頓用を選び, 用量等を決定する.



#### C.15.5.3 Targeting

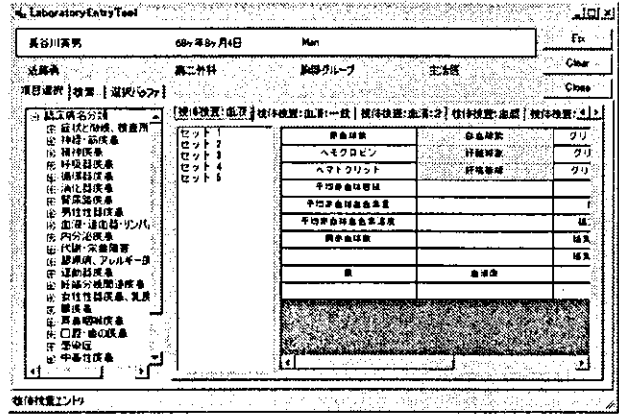
その後, 画面左下方で, この処方を格納するべき診療ブロックを選択する.

### C. 15. 6 検査 OrderTool

画面例を列挙する (実装は大嶺).

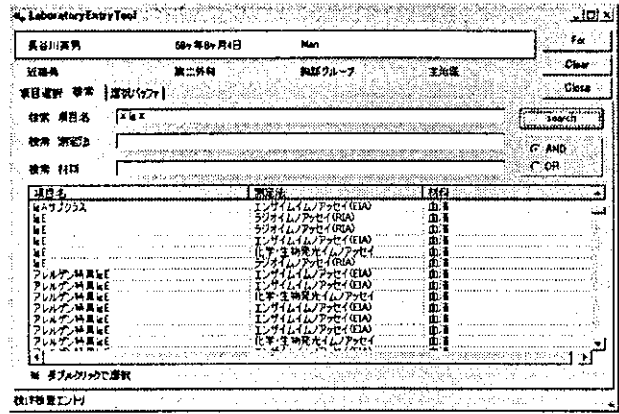
#### C.15.6.1 Menu

メニュー画面は, (C.15.2.3) 仕様によって作製した.

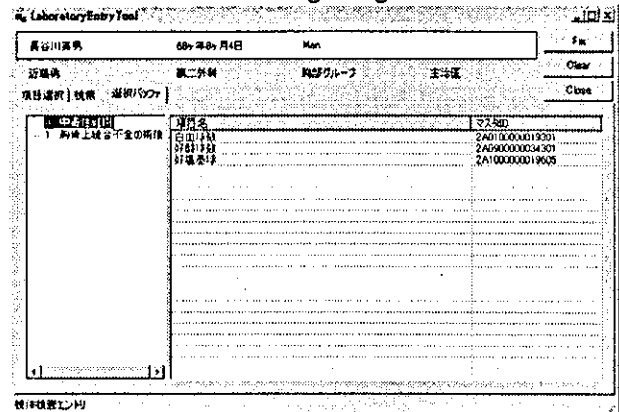


#### C.15.6.2 Search

メニューに存在しない項目は, 3軸の and/or にて検索可能である.



#### C.15.6.3 Targeting

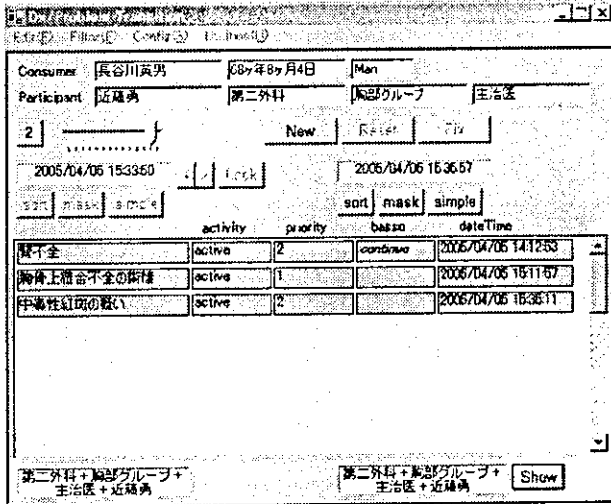


### C. 15. 7 病名変遷 Editor

画面例を列挙する (実装は尾藤, 与那嶺; 基幹クラスは山田).

#### C.15.7.1 Single pane

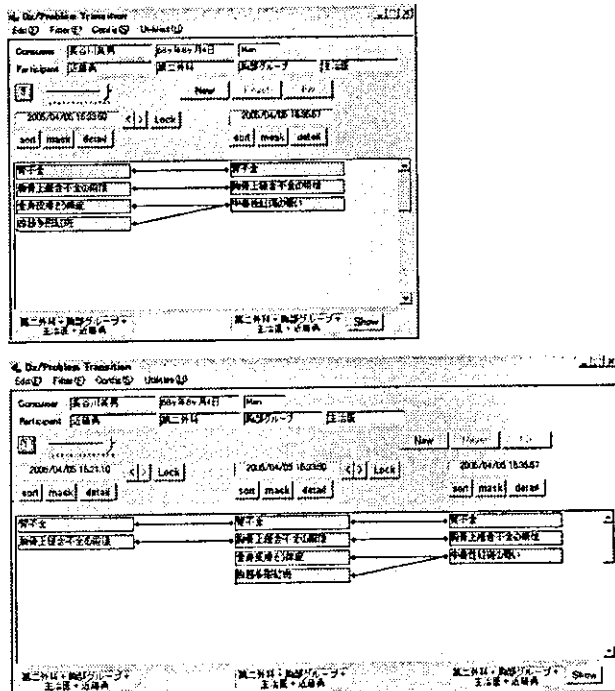
通常の 1 号様式保険傷病名欄と大差ない.



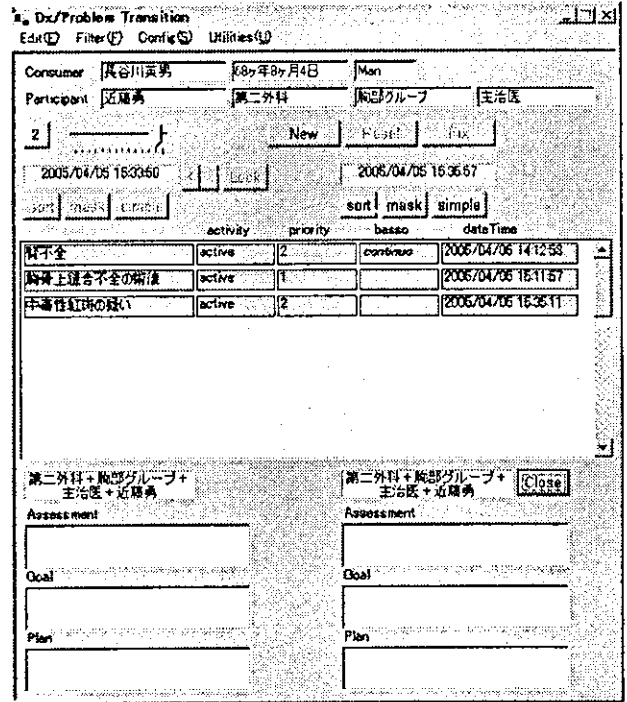
ただ activity, priority, basso を設定できるようにになっている.

#### C.15.7.2 Multi pane

Multi-pane にすると, 病名変遷が閲覧編集可能となる.



### C.15.7.3 AS, GL, PN Assessment, Goal, Plan も入力可能とした.



### C. 15. 8 加療履歴と病名連関

画面例を列挙する（実装は大嶺；基幹クラスは山田）。

#### C.15.8.1 診療ブロックの構成例

前述 (C.15.4~6) の入力結果は以下のように表示される。

診療履歴	CONVERT & COMMENT
→ 診察セッション 2005/04/05 15:03 ~ 2005/04/05 15:06 T-4 CTX-B	
→ 診察セッション 2005/04/05 15:13 ~ 2005/04/05 15:20 T-3 CTX-B	
→ 診察セッション 2005/04/05 15:21 ~ 2005/04/05 15:21 T-2 CTX-B	
→ 診察セッション 2005/04/05 15:36 ~ 2005/04/05 15:40 T-1 CTX-B	
2005/04/05	<ul style="list-style-type: none"> <li>中絶性心房細動           <ul style="list-style-type: none"> <li>→ オータノゾール</li> <li>→ フリドコゾン錠 6mg1錠 (P)</li> <li>内服 5錠</li> <li>術後経過 異常性心室性期外収縮</li> <li>1日量 12錠</li> <li>3日分</li> </ul> </li> <li>→ オータノゾール投与</li> <li>白濁時投与</li> <li>呼吸時投与</li> <li>静脈投与</li> </ul>
	胸骨上縁不全の病歴
	胸骨創傷後遺症
	第二外科 近藤典

これで病名と加療行為が関連付けられた。

#### C.15.8.2 アウトライン化の構成例

必要に応じて、加療行為履歴をアウトライン化できる (C.15.2.4)。

診療履歴	CONVERT & COMMENT
→ 胸骨上縁不全	
→ 胸骨創傷後遺症	
→ 第二外科 近藤典	
→ 診察セッション 2005/04/05 15:13 ~ 2005/04/05 15:20 T-4 CTX-B	
2005/04/05	<ul style="list-style-type: none"> <li>不全</li> <li>胸骨上縁不全の病歴</li> </ul>
	第二外科 近藤典
2005/04/05	<ul style="list-style-type: none"> <li>不全</li> </ul>
	第二外科 近藤典
2005/04/05	<ul style="list-style-type: none"> <li>中絶性心房細動           <ul style="list-style-type: none"> <li>→ オータノゾール</li> <li>→ フリドコゾン錠 6mg1錠 (P)</li> <li>→ オータノゾール投与</li> </ul> </li> <li>胸骨上縁不全の病歴</li> </ul>
	第二外科 近藤典

### C. 16 参照実装 (Client/Server 版)

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

#### C. 16.1 Native XML DB の評価

方法(B.12)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. 16.2 Caché の採用

Caché について前年度からコード・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 の特長等は, 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, 閾が低いとは云えないように感じていた。

しかし前述 (16.1) の状況に拠って, Caché において 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(16.1)の 1) と 2) については前年度に, 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. 16.3 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が要求する諸環境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

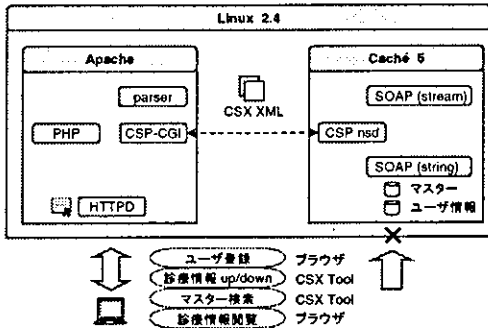
よって 0) により, 慣れない CSP でのシステム構築は避けつつ, 1) 2) は String ベースでの通常の SOAP サービス, 4) は SOAP により Stream をキックすることにした。その際 d) にも配慮することとなったが, その一方で結果的には

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. 16.4 システム構成と機能

システム構成は以下のようにした。



すなわち、Caché の直接アクセスは許可せず、CSP-CGI を介して、CSP::SOAP サービスを利用することとした。Stand-alone 版と同様、マスタ検索は Caché のデータベースを検索し、診療経過ファイルは OS 保管することとした。

図中で「ユーザ情報」と記されている内容は、実際には組織情報も含まれている。つまり (C. 15.1) で提示した C/S 版でのディレクトリとクラス構成のうち crt, csm, stf, org に含まれるファイルを Caché に格納したわけである。

予めこのように分割しておいたことに拠って、Client 側での改変は入出力「向き」変更と理解することが容易となった。

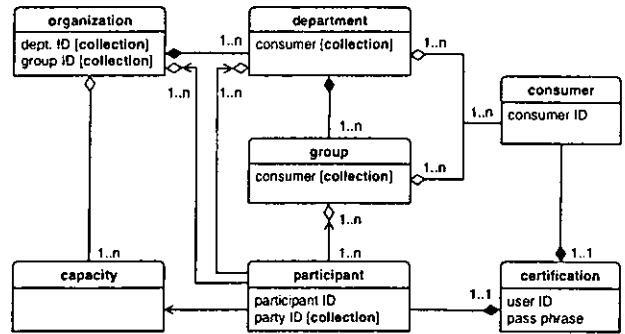
なお (C. 15.1) で挙げた ext は、今年度の C/S 版実装では対象外とした。というのもサーバに対して upload されたバイナリファイルを無防備に扱うリスクは、回避すべきだからである。

また transaction 管理については本研究主題を外れるので説明を割愛する。

### C. 16.5 場の形成 (簡易)

役柄配役立場モデルの部分実装方針 (C. 13.4) (C. 13.5) (C. 13.6) に従い、次図の Class 構成とした。

なお Class への collection の持たせかたは、参照実装「診療システム」での絞り込み効率に配慮しての方策である (C15.2.1)。



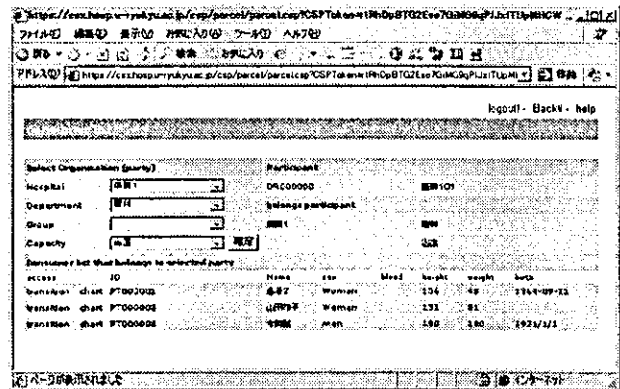
### C. 16.6 結果参照画面例

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。

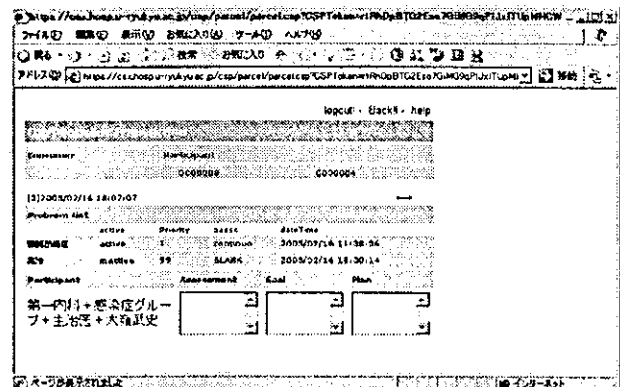
よって画面の構成や展開 ならびに 各操作については、Stand-alone 版と変わらない。

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。



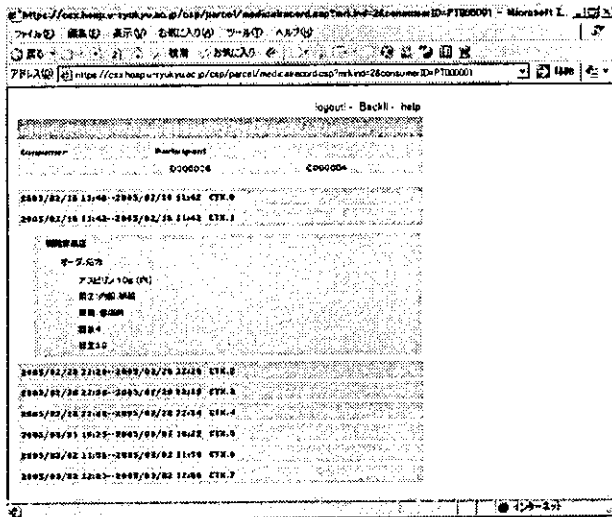
下図は、最新の診療セッションの病名リストを表示した状態である。



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. 16.7 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となってしまふからである。

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/login.csp>

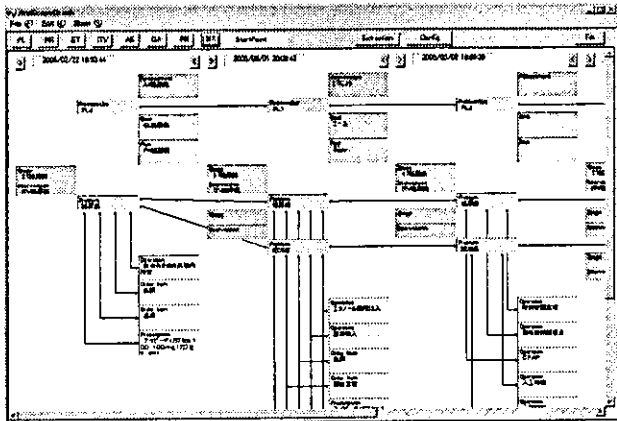
### C. 17 関連情報の抽出と要約

TheTOOL の基本性能は (C.14) 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の Dx/Problem Composer の Graph pane に拠っている。

#### C. 17. 1 一覧性

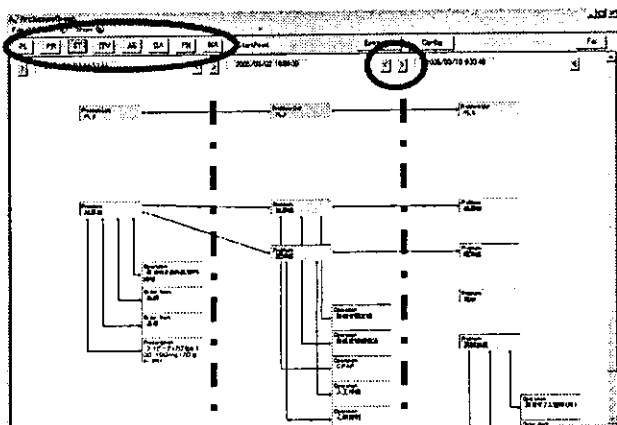
診療システムで記録された診療経過は、病名やプロブレムの変遷とともに、それらと加療行為とを結び付けている。このようなグラフ構造に配された情報塊の諸関係は複雑なので、視覚的に捉えたいものである。

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、も実装した。



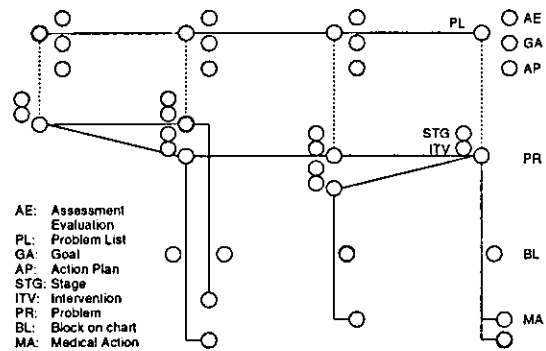
これで通常の意味での「要約」が可能となった。

#### C. 17. 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。

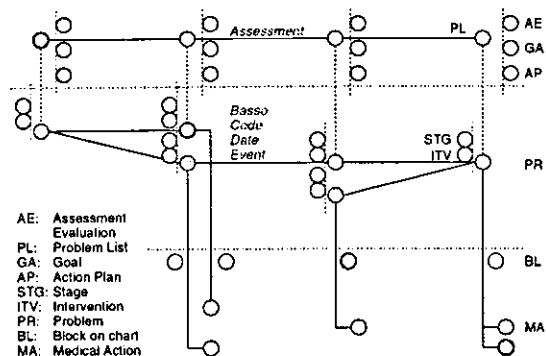
ただし下図の如く、全ての infoNode は相互に連結している。



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：

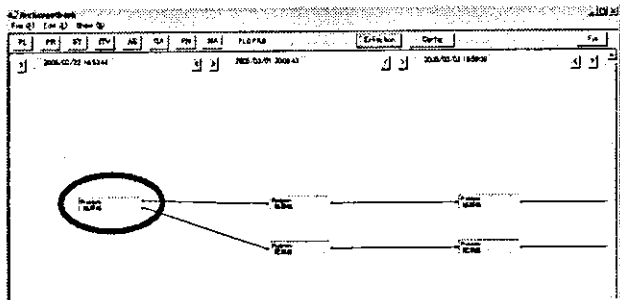
- a) 包絡する infoNode の種類の限定
- b) 日付による限定
- c) 診療イベントによる限定
- d) 根幹病名コードが変化した場合の処理
- e) 病名修飾コードが変化した場合の処理
- f) DxProblem の属性 Basso に依存した処理
- g) Assessment や Goal の変化に依存した処理
- h) そのほかの変化に依存した処理

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、今年度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してみると、以下の結果が得られることになる。





そしてさらに、(C.17.1)に掲げた (1)~(3)の機能を適用したなら、充分に満足できる抽出と要約を得られることになる。

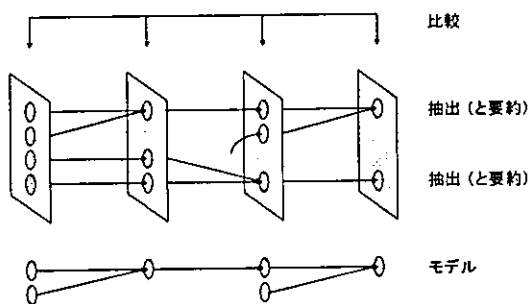
これらの機能を実現できる最大の理由は、将に、病名/プロブレムの変遷と・病名/プロブレムと加療行為とを連関させた・診療経過記録を作成した

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.1) (B.2) ならびに (D.3.2) も併せて御参照願いたい。

### C. 17. 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.5) 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. 17. 4 モデル化へ

#### C.17.4.1 さらなる「流れ」編集

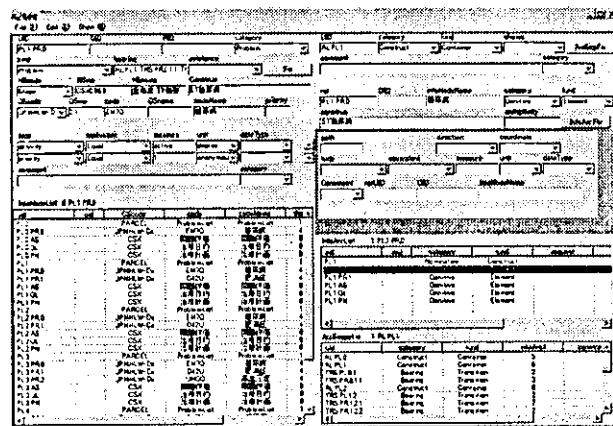
ところで、臨床経過から得られた事実は、必ずしも典型例とは限らない。

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できない。

しかし EDIT 画面においては必要な infoNode を自在に生成することができる。また当然ながら infoArc や arcScope を生成することができるようになっている。



#### C.17.4.2 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. 18 臨床家等からの評価

本研究の主題に関しては、概ね、良好な反応を得ることができた。

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。これらについては一定の評価を得られたように思われる。

ただ今年度の参照実装は、あくまで、本研究の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に試作された。それをこのままのカタチで臨床現場での使用を強要するものではない。

この辺りに関する誤解と思われる回答もなきにしも非ずであったが、その事由は以下に拠るものと思われる：

- ・ MS Windows と MacOS の human interface の差
- ・ 試作アプリとしては出来過ぎのために、むしろ試作アプリとして、ではなく、実業務用の電子カルテと目されたこと
- ・ 試作アプリに過ぎないことと、臨床現場で使用するべき業務アプリとの混同
- ・ 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、という誤解
- ・ TheTOOL の提供が遅れたこと
- ・ 臨床思考の困難さと、human interface の使い勝手との混同

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、と答えていた。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。

前者の human interface は、好みの問題、ゴテゴテした human interface のほうが使い易そうに感じられる錯覚、試作アプリの実装ではコスト管理上“凝る”わけにいかないこと、等であろう。

後者については、逆に云えば、現状が如何なるものであるかを示唆しているようにも思えるところではある。

## D. 考察

### D. 1 CSX model design

#### D. 1. 1 経緯

本研究主題の源流は (B) に記したように単一ではない。

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した。

それは、情報モデルとして ontology と仕上げられたことから、ontology として洗練させつつ、思考空間 (B.1) (B.2) (B.3) に適用することとした。他に選ぶべき候補が見当たらなかったためである (D.2)。

#### D. 1. 2 特徴

幾つも挙げられる (C.6.2) が、主要点を要約する：

(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) 情報モデルは完全に domain 非依存であり、domain 依存部分は全て meta-attribute 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) モデル表現形式として提供する枠組が提供する object は meta meta-information object だが、必要な属性を埋め込むことで、それらは、そのまま instance としても機能する。つまり (vi) 微細粒度から大域粒度まで境界なく再帰的に単一情報モデルにて、多くの事物事象を表現できることで、複数モデルを適用した際の界面に発生する汚染と混淆を回避できる。(vii) 先験的な軸性を情報モデル自体が暗黙的に規定せずに、むしろ自在に設定する環境を提供する。(viii) これらの特徴は全て、ontology に即しつつ meta meta-information object を用意し、かつ、必要に応じて抽象空間に位相構造を与え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、そのような箍 (あるいは規律) が二つある。

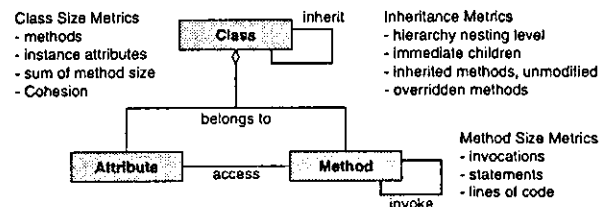
その一つは arcScope によって嵌められる視座意義や視野範囲、今一つは各 object の属性に格納される値は、各々、code schema 内に定位する code の適用を想定しているからである。

この事情を模式的に言い換えると、CSX model 自体は多重グラフの枠組の中で自由度の高い表現力を発揮して、種々の情報塊を奔放に組合せ・あるいは躍らせることも可能のように見えるものの、実のところ arcScope から「可能な関係構成範囲や関係の意義」なる箍を嵌められ、そのうえ拠って立つ足元は全て、code schema 内に明確に定位された事項できつく縛られているので、結果的には“不自然な姿勢”をとることは出来ない、のである

すなわち、意味の無い、あるいは妥当ではない多重グラフ構造を構成することは、実質的には困難となっている。

#### 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.12.3) (C.15.2) (C.15.3) に記した如く局面によって既に focus されている。

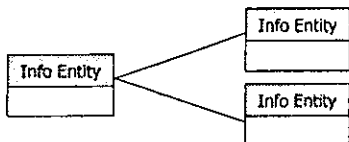
結合度については arcScope の周りで結合度が高くなるが、infoArc は単一の infoNode しか指示していないし、また infoNode は、自らを規定する arcScope を @bearing に格納し明示しており、さらに (C.12.3) の事情もある。

これらのことから CSX model のような一般的ではないモデル化手法が計算に多大な悪影響を及ぼすとは思われない。実際に、試作した参照実装は妥当な速度で応答した。

#### D. 1. 5 関係の結びかた

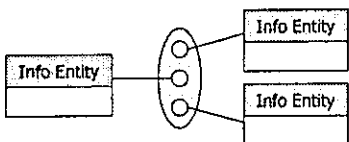
ある Class が数種の Class と関係を結ぶことはよく見受けられる。

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いないことに留意すべきである。

一方 CSX model では (C.6) およびそれ以降に例示したようにして関係を結ぶ。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.11)。



楕円は arcScope を、小円は 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 が深層格関係を表しているのならば問題なく、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なかった。

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、唯一つの Class に絞って考えざるをえないことを示唆しているように思える。

そして arcScope は、その focus を、@category と @kind に格納している。

よって今年度、この直列化手法を採用したままとした。なお今後反例が発見された場合には、上に列挙したような関係のみを対象とする旨、明言しつつ対処することとする。

#### 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」では正確に表現しきれない場合があるからである。つまりそのような意味関係を有する意味構造が、現実存在しているからである。

#### D. 1. 7 制約と輻輳と隔離と

制約表現の付与の際には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に関する制約表現の困難さは、arc での有向性の有無に関わらず、(1) node での多入力・多出力すなわち輻輳と、(2) 隔離 node 間における依存と排他の管理とに拠る。

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) のような制約表現とした。

## 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.5.1) 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え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**  
まだ線引きされていない広がりを区切る。  
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、比較等を記していく。

### 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の努力が為され続けている。複数の階層系 (ツリー) を持っている。

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。言い換えれば世界における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。

いわゆる kind-of は、part-of や caused-by などから注意深く分離され、各 concept は attribute によって結合される。その際の SELECTOR は、興味深い機構を示す工夫となっている。

ただ attribute 種は極めて膨大ではある。なお収録されている概念数は概ね 7,000 である。

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。なお、その中で Frame を構成して、Slot を用いつつ horizontal relation (association) を形成することも、可能であるとしている。

しかし、one tree の世界の構成に過ぎず、multi-taxonomy のなかで関係を構成することはできない (ただし multi-super class には一応は対応している)。

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」枠組み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 ほかに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方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 も知識 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。

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、という枠組みにおけるメタメタ表現環境である。

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

これらからのことから、CSX model とは目的も異なり、また枠組みも異なっていることが直ちに判明している。

さらには (D.2.6) でも述べるが、少なくとも情報モデルの構築においては、two model methodology や 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つことになる。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みを支援するために 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 が参照した枠組みは RDF (Resource Description Framework), OWL (Web Ontology Language), OCL (Object Constraint Language) であって、CSX model 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、という問題を完全に自動的に回避解消するのは不可能だからである。

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 をも記述することができる。

#### 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。一方、“Metathesaurus”と“Semantic Network”については ontology 的であるし人に拠っては、そう考えるだろう。

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てならない。

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みを提供している。それらは Generic (通常 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、枠組みである。

#### 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.6.2) (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ば次であろう：

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。つまり抽象空間に位相構造を定義しう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/プロブレムと診療行為との連関の記述は勿論, これを具現する参照実装にも成功した. したがって, これらに関する CSX model の有用性検証は完遂した.

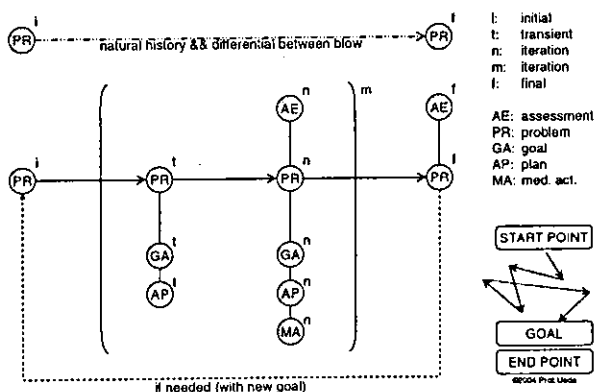
病名/プロブレムの変遷や, それらと加療行為との連関に関する汎用的な記述モデルおよび実装による検証は他に無いので, 本研究成果は貴重である.

#### 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.14) (C.17) 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. 現時点で既に臨床, とくに難症例の解析や臨床研究あるいは evidence 収集等に有益なことは明らかである.

加えて, 臨床教育や症例検討にも有用であろう. 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, 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.

さらに infoArc や arcScope には, 事情や理由等の意味を含めることもできるので, それらを基にして自然言語を生成することさえも, 近い将来に実現可能であろう.



上図と (C.17.2) に提示した図の相似性を比較して載きたい.

さらに (B2) に掲げた思考過程の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 診療経過モデルの生成

また前述 (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. 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. 3.4 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を作ることでもできよう.

もっとも, 現時点でも, それを作製するための種々のリソースが与えられたなら, 現状モデルにて可能である. ただ制約表現を加えたモデルにて実装したほうが, より実用的となる.

#### D. 3.5 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だし, また自由度に富むことから, メンテナンス・コストも縮減可能と思われる.

#### D. 3.6 GraphPane

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 の如きツールの応用可能性は広いと云えよう.

#### D. 3.7 役割配役立場モデル

本研究から, 以下が明確となった:

- 権限管理とは権限根拠の管理のことである.
- 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. 4 今後の課題

前年度報告書 (D.2) (D.3) の課題は、今年度報告書 (B.14) に要約しつつ今年度方針を明確にしていたが、これらは全て解決した。よって、その意味において「遣り残し」は無い。

そのうえでの、次なる目標を掲げておく。

##### D. 4. 1 診療方向性に基づく追跡

本報告書 (C.17) 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.14) (C.15) は診療方向性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、むしろ目標設定と(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。しかし次年度においては、これを進めたい。

と同時に、今年度に試作した診療システムや TheTOOL を、洗練させていきたい。

##### D. 4. 2 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. 3 制約表現の実装

制約表現に関する定式化を終えた(C.2)ので、これを実装して検証したい。

題材としては二つ挙げられる：(i) DPC や点数表が含んでいる制約や条件の記述とその適用、(ii) 標準治療計画やパスの記述、である。

##### D. 4. 4 述語の扱い

今年度は、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. 4. 5 継承の扱い

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.6.2)。

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. 4. 6 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. 4. 7 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. 4. 8 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。

##### D. 4. 9 そのほか

そのほか、普及と発展のためには以下の事項も併せて到達していく必要がある：

###### D.4.9.1 Human interface

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となる全ての Entry Tool を取り揃えてはならず、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.4.9.2 HL7 準拠メッセージング

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。

## E. 結論

### E. 1 真の evidence の獲得と監査可能性

本研究の成果は汎用的であり、理論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。後者について若干を記す：

EBM 研究は診療品質の維持と向上に必須だが、臨床現場との間で“診療ベクトル”が乖離したままに、解析が進められたりデータ収集が続けられたりしても意義は高まらない。

- 本研究成果とその方向性は、その様な努力と資源投入効果の極大化に貢献するものである。

一次データの品質は、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。

我が国の診療情報システムは、システム内での“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り、個人情報等を適切に管理する礎を与える。

これは全て、「原因や事由に基づいた行為」の連続における診療現場の状況や診療の成果を記録するための情報モデルの研究開発や試作実装を目標としたことに根ざしている。

### E. 2 臨床教育ならびに知識の表現と交換

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である。

これは診療品質の監査ばかりでなく、熟練者と初心者との比較を機械処理する可能性を拓くものであり、臨床教育に資するところ大である。

また診療記録形式と知識表現形式を統一的に扱える枠組は知の時代に有用かつ重要である。

### E. 3 進捗と課題

前述 (D) ほかの通りである。

今後は (E.1) (E.2) を意識しつつ、さらなる展開を推進していく。

## 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, 2004 (2004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

