

H17-医療-043
厚生労働科学研究費補助金
医療技術評価総合研究事業

診療の方向性に基づいた監査や追跡性に
資する
電子カルテの記述モデルに関する研究

平成 17 年度 総括研究報告書

主任研究者 廣瀬 康行
平成 18(2006)年 3 月

ISBN 4-902408-12-0

主任研究者:

廣瀬 康行 琉球大学 医学部附属病院

分担研究者:

山本 隆一 東京大学 大学院情報学環
植田 真一郎 琉球大学 大学院医学研究科

研究協力者:

乾 健太郎 奈良先端科学技術大学院大学 情報科学研究科
山下 芳範 福井大学 医学部附属病院
原田 亜季 琉球大学 医学部附属病院
矢嶋 研一 矢嶋歯科医院
神田 貢 神田歯科クリニック
阿部 隆明 阿部内科医院
山田 清一 株式会社 テクセル
与那嶺 辰也 株式会社 創和ビジネス・マシNZ
村上 英 東芝住電医療情報システムズ 株式会社
山本 聡 株式会社 ソリトンシステムズ

委託開発者:

株式会社 創和ビジネス・マシNZ
株式会社 テクセル

協賛協力:

インターシステムズ ジャパン 株式会社

目 次

I 総括研究報告		
診療の方向性に基づいた監査や追跡性に資する 電子カルテの記述モデルに関する研究	-----	1
廣瀬 康行		
II 主任/分担研究報告		
1. 上記モデルに関する計算可能な意味構造の保持の ありかたに関する研究	-----	一括
山本 隆一		
2. 上記モデルに関する要求要件ならびに 臨床意義に関する研究	-----	3 1
植田 真一郎		
III 研究成果の刊行に関する一覧表	-----	4 1
IV 研究成果の刊行物・別刷	-----	4 3

本研究の性格上、特に山本隆一分担研究者とは密接に協力して課題解決にあたる必要があり、成果内容も深く絡んでいる。さらに研究経費も一括処理していることから、山本分は総括研究報告書にとりまとめることとする。

H17-医療-043
厚生労働科学研究費補助金
医療技術評価総合研究事業

診療の方向性に基づいた監査や追跡性に
資する
電子カルテの記述モデルに関する研究

平成 17 年度 総括研究報告書

主任研究者 廣瀬 康行
平成 18 年 3 月

目 次

A. 研究目的.....	1
A. 1 必要性.....	1
A. 2 範囲と方向.....	2
A. 3 分掌.....	2
B. 研究方法.....	3
B. 1 情報モデル.....	3
B. 2 概念モデル.....	5
B. 3 倫理面への配慮.....	5
C. 研究結果.....	7
C. 1 思考過程の概念モデル.....	7
C. 2 Perspective.....	12
C. 3 CSX.....	14
C. 4 思考過程の記述と語彙.....	21
C. 5 診療文書の記述と語彙.....	24
C. 6 OntoScope.....	25
D. 考察.....	27
D. 1 意義.....	27
D. 2 CSX.....	27
D. 3 語彙と述語.....	27
D. 4 マッピング可能性.....	27
D. 5 応用可能性.....	27
D. 6 今後の展開.....	28
E. 初年度における暫定結論.....	28
F. 健康危険情報.....	28
G. 研究発表.....	28
H. 知的財産権の出願登録状況.....	28

厚生労働科学研究助成 医療技術評価総合研究事業 (H17-医療-043) 総括研究報告

診療の方向性に基づいた監査や追跡性に資する 電子カルテの記述モデルに関する研究

主任研究者 廣瀬 康行 琉球大学医学部附属病院 教授

研究要旨 : 今後の保健医療福祉を支える医療情報システムには、診療行為の論拠性と効率化、そして診療の品質維持に資する一次情報を精確に集積する情報構造を持つことが求められている。この課題を解決するためには、動的要素の大きい臨床現場においても原因や事由に基づいた行為の連続としての成果を記録する医療情報システムの構築に資すべき情報モデルと記述構造とが必須である。これらの構築には情報技術のみならず臨床思考過程への洞察をも必須とするものであり、従来のようなシステム機構の延長線上に在りうるものではない。

目標を一挙に達成するのは困難ゆえ、前年度までの研究では次の三つに焦点して完遂してきた :

(1) 病名やプロブレムの変遷状況を捉えること、(2) 病名やプロブレムと診療行為とを関連付けること、(3) その関連性は論拠や事由として意味付けられること。よって本研究はその知見や枠組みを発展的に活用していくこととした。

さて診療とは病名やプロブレムを論拠とした治療目標やエンドポイントに対する介入なので、診療成果は治療目標やエンドポイントと比較されて初めて意義付けされうるものだが、現況システムは文脈保持力を欠いている。より端的に云うならば、治療目標やエンドポイントと診療成果とを比較する意図を感じられない情報構造である。

したがって上述した目標を達成するためには、両者を合理的に含めうるような新しい枠組みが必要となる。研究者らは、まさに臨床思考過程にその情報構造を求めることとし、これをモデル化し、規定していった。その記述形式には H15-医療-050 の成果である ontology CSX を活用した。そして種々の関係の視座と意義、それらに関わる述語ほか語彙を検討しながら、思考過程の構造を明示化した。

分担研究者

山本隆一 東京大学大学院情報学環助教授
植田真一郎 琉球大学大学院医学研究科薬物
作用制御学教授

なかにあって診療行為の論拠性と効率化と品質維持、あるいは行政施策の立案に必要な一次情報を精確に収集分析するには、医療情報システム自体の品質の向上が不可欠となっている。

本研究の性格上、特に山本隆一分担研究者とは密接に協力して課題解決にあたる必要があり、成果内容も深く絡んでいる。さらに研究経費も一括処理していることから、山本分は総括研究報告書にとりまとめることとする。

また本邦のみならず OECD 諸国では医療の効率化と同時にその安全性が問われ、ために科学的根拠に基づく診療の実施、そして比較と分析が求められているところである。

A. 研究目的

A. 1 必要性

A. 1. 1 社会的な諸要請

本邦の保健医療福祉制度は過渡期をむかえており各種の改革が推し進められているが、その

診療とは「原因や理由に基づいた病名やプロブレム」に対する「目標」や「エンドポイント」を設定したうえでの医療介入の「連なり」であり、またその結果としての「成果」である。

にも関わらず、現状のほとんど全ての電子カルテシステムは基本的にトランザクション管理を主としているなどの事由から「文脈の保持力」に欠けており、上述の目的に合致する構造設計となっているとは言い難い状況である。

換言して極論すれば単なる伝票処理と料金計算のための縦割りシステムであって、「原因や事由に基づいた考察と行為の実施」の連続結果としての成果を記録しながら、その「成果」や「品質」については、当初の「目標との比較による評価」を支援するような構造設計となっていない。さらには情報源の確認に基づいて当該情報の信頼性を評価する、などということも、為しえなかったのである。

そのため、一次情報抽出の際にも単なる項目の羅列が得られるのみであり、その整理は多大な人手を介するか、でなければ、ノイズや「診療の方向性(診療ベクトル)」の不整合を含んだままに解析処理を開始せねばならない現況であるし、そのような状況は長きに亘って続いていた。

診療の「事由」と「方向性」の記述が為されて初めて個々の診療行為の内容の妥当性の検証すなわち監査や、診療成果の評価をしようにも関わらず、この点が過小評価または看過されてきたからである。

よって大規模な EBM の研究結果にも自ずとその適用限界はあるし、場合によっては、その適用範囲が誤解誤用されることさえありうる現況となっている。

本研究はこれの解決に資するべく、診療文脈における「論拠性の記述」を確保しつつ、診療の方向性に基づいた監査や追跡性に資する電子カルテの記述モデルに関して研究するものである。

A. 1. 2 EBM の危うさ

また (i) evidence を獲得するための臨床試験では診療ベクトルを無視した過度の単純化によって診療アウトカムを評価してしまったり、(ii) そのような evidence を活用する際にも、当該 evidence を獲得した介入試験が前提した診療ベクトルと対象症例の診療ベクトルとの一致性を無視して試験実施したにも関わらず、それを evidence としてしまったりする事例も散見されるようである。

A. 2 範囲と方向

二カ年計画の初年度である本年度は、臨床思考過程の情報構造を定式化するとともに、種々の情報塊の関係についての視座や述語等を整理する。

言うなれば思考作業それ自体の意味構造を明らかにして、状況における決断事由や・文脈における目標設定とその達成への指向性を強く意識した情報構造を有するインテリジェントコンテンツの枠組を策定する。

その様なセマンティック・アノテーションには前提とされうる抽象的な「枠」が必要となるが、意味関係は全て、ある観点 (perspective) において (1) 要素が存在するべき領域や支領域を特定しながら (2) 個々の情報塊または要素間を意義付けしたリンクで関連づけることで、意味世界の多次元構造を明示することとする。

そのうえで、診療目標 (GOAL) や・目標達成の停止条件であるエンドポイント (EndPoint) を位置づけ、それらと他の情報要素の関係構造を規定する。

A. 3 分掌

廣瀬康行

診療の事由と方向性を明記して追跡性を確保する情報モデル

山本隆一

上記のモデルに関する計算可能な意味構造の保持のありかた

植田真一郎

上記のモデルに関する要求要件ならびに臨床意義

B. 研究方法

B. 1 情報モデル

本研究主題の実現には、複雑な関係様相を簡潔かつ自在に表現できる情報モデルが求められているが目的に合う情報モデルは稀有である。

ただ研究者は厚生労働科学研究 (H12-医療-009 および H15-医療-050) を実施した。その成果として ontology に基づく meta meta 表現枠組を開発すると共に、その XML Schema による直列化形式を定義した。

その情報モデルの namespace prefix は *csx* なので、今後これには *CSX* を冠して呼ぶこととする。

CSX meta modelling framework は関係の様相を仔細かつ正確に表現する能力を有しながらも簡明で小さい。よって情報モデルに CSX を採用し、これを発展的に改変していくこととした。

B. 1. 1 本質

CSX は ontology のうち情報モデルである。

CSX は meta modelling framework 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。

よって、meta-modelling framework 自体には何も含んでないとも言えるが、逆に全ての構造と要素とを生み出す力を内包している。

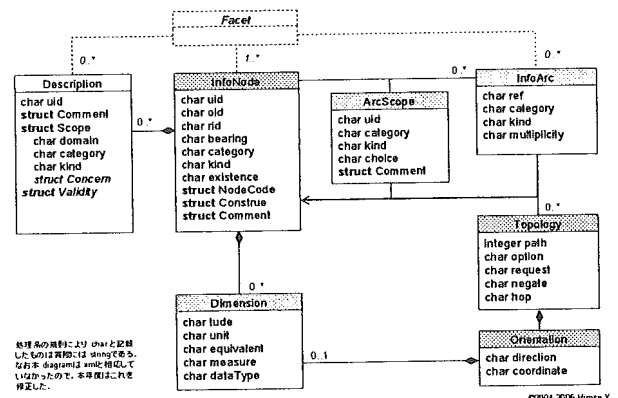
なお 構造の生成と解釈には 多大なコストを要すると思われがちだが、再帰性と同型対応によって現実時間内、しかも実業務時間内に処理可能である。実際、それらは参照実装に拠って証明されている。

B. 1. 2 要素ダイアグラム

この情報モデルは根源的であるがゆえの柔軟性と汎用性を有しており、とくに事物の関係を記述する能力に長けている。

というのも CSX は、(A) 具体のみならず抽象も対象とでき、かつ事物が属する domain または

subdomain を表す属性を有する、(B) 関係要素は対象領域固有の業務上の関連性のみならず、深層格や修飾補語という根源的な関係も表現できる、(C) 事物要素や関係要素の各諸属性に格納される値は種々のコード体系のコードであり code schema に則った階層構造に定位するコードの使用を前提している、(D) 細粒度から大粒度までの情報塊を再帰的に構成できるような枠組を提供している、からである [医療情報学 33 (1) :33-43, 2003] [CSX M 02:2003 v0.90] [CSX M 01:2003 v0.90 rev1] [CSX S 01: 2003 v0.90 rev1]。



要素は meta meta-information object のため抽象度が高く、それそのままでは具体の object は何も表現していない。しかし逆に、汎用度は極めて高く、通常のデルでは明示しにくいメタ情報も表現できる能力を有している。

B. 1. 3 特徴

B.1.3.1 Framework

CSX model とは、ontology でもあり、と同時に meta-modeling 手法 (meta meta-expression 環境) でもある (ただし CASE で云うところのそれとは異なることに御留意願いたい)。

CSX とは ontology を礎として、meta-modeling から modeling, concretizing までの各段階で扱うべき情報客体を全て取り扱うことを可能とした framework を与えている：

- Meta meta-information object
 - Meta meta-entity
 - Meta meta-relation
 - Scope of meta meta-relation
- Generic (meta-information object)
 - generic Class
 - generic Attribute
 - generic Association

- Constraint
- Information object (user defined)
 - Class, Attribute, and Instance
 - Horizontal and vertical relation
- Scope of relation

B.1.3.2 Elements

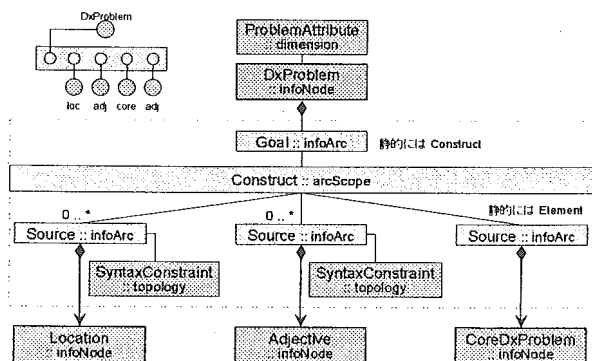
仕掛けはモデルデザインに埋め込まれている：

- Ontology に即しつつ極めて抽象度の高い中核要素を提供：情報素 (infoNode)，関係素 (infoArc)，関係視座素 (arcScope)
- 補足的に、極めて抽象度の高い意味での計量や性状を表現する計量性状素 (quale) を提供
- 補足的に“抽象”空間での“位置”関係を表現する位相素 (topology) を提供
- 各要素が表現する内容は要素属性の値に拠り、その値はコード体系に定座するコードであることを前提すると規定
- ある情報素が他の情報素と関係を結ぶ際は、その関係を前提する視座意義や視野範囲を必須とするように規定
- 具体事象を表現する情報塊は、この枠組の内て再帰的に構成していくよう規定

B.1.3.3 Concretizing

各 meta meta-information objects は、各々の範疇子 @category や @kind にコード体系とくに code schema (taxonomy が明示的なコード体系) 内のコードを格納して具体化され、さらに情報素 infoNode では子要素 nodeCode で詳細な事象を規定されて class (model) とされていき、最終的には @uid が附番された時点で instance とされる。

大粒度または大域粒度の情報塊を構成したり文脈などを表現したりする際は、細粒度の情報素を順次組み合わせながら、つまり再帰的に、構成していくことになる (下図は病名の例)。



このとき各々の情報素は互いに関係づけられ

ることになるが無秩序な結合は許されず、視座意義や視野範囲が限定されて、関係が結ばれることになる。

これらによってデータ・ハンドリングにおける解釈の発散や混淆を防ぐとともに、処理効率を損なわないよう留意されている。

B.1.3.4 Single model architecture

よって、meta object から user object に至るまで単一の情報モデルで表現できるのである。

これは、特定の対象ドメインに関する具体の大規模な情報モデルを構築する際も、細粒度から大域粒度に至るまで情報塊を扱いつつ要素数 (Class 数) は増加しないことを意味している。これは CSX model の重要な側面を示している：

- CSX model は小さくロバストでドメイン独立である。
- CSX model において、ドメイン特異性は、代入される属性値、関係視座や関係視野のとりかた、および具体の制約内容によって示される。
- CSX model はドメイン記述において single architecture model を支援するので、界面にて発生する汚染や混淆を回避する。

また CSX model は視座意義や視野範囲において述語と深層格あるいは修辞関係を扱うことを想定している。したがって静的な関係様相のみならず動的な関係様相も記述可能としている。

B.1.3.5 Axis and Taxonomy

CSX は軸性を持たない。正確には、a priori の軸性は規定しておらず、その構築環境のみ提供している。よって如何なる軸性も必要に応じて生成することができるのである。

なお軸性は arcScope/infoArc に階層を与え、infoNode [@category/@kind] に perspective を与えて形成されることになる。ある軸性の階層構造つまり taxonomy における関係を vertical relation と呼ぶなら、horizontal relation については別個の関係 scope によって形成される。こうして多層多重グラフ構造を構築しうる。

したがって infoNode に対する infoArc とは、背後に scope を持っている SLOT であると解釈することも可能である。Quale を値 SLOT と解釈するならば。

B. 2 概念モデル

B. 2. 1 病名プロブレム変遷

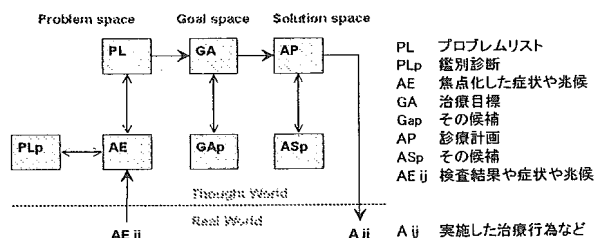
病名/プロブレムの変遷に関する情報モデルについての報告は少ない。ただ、プロブレム変遷記述言語を主題とした研究が発表されており、これは、病名/プロブレムの変遷の記述に必要な述語群と、個々の病名/プロブレムの詳細を記すべき修飾節を提案している [医療情報学連合大会論文集 17: 60-61, 1997]。

その述語群と修飾節はそれぞれ『変遷関係』と『要素属性』とに置換されうる。よって、この病名/プロブレム変遷モデルを採用することとした。

なお上図は厚生労働科学研究 (H15-医療-050) において rank を Basso Continuo と呼び換えて概念を明確化したものである。

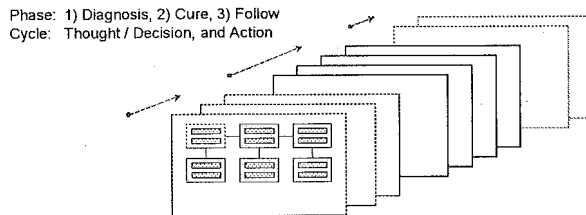
B. 2. 2 思考過程モデル

臨床での医師の思考過程あるいは決断過程のモデル化を扱った研究は極めて稀である。ただ認知科学的な概念モデルが報告されている [医療情報学 17 (3) S: 185-192, 1997]。



この思考過程モデルでは、まず現実世界と思考世界 (思考空間または問題解決空間) を分離し、次に、思考空間を三つの subspace すなわち、問題定義空間、目標決定空間、計画策定空間に分割している。そして、思考運動は現実世界を観察して獲た情報を思考空間に取り込み、知識などを参照して種々を考察策定し、計画実施に際しては、計画された個々の加療行為を、思考空間から現実へ写像する、としている。そして、この思考ならびに現実実施の一区切りを診療セッションまたはサイクル (Cycle) と呼んでいる。

医療ドメインにおいては、初診から転帰に至るまでの一般的な診療過程は三つの相フェーズ (Phase) に弁別できよう：診断相、加療相、継随相。



Phase は Cycle の連なりから構成される。医療ドメインの場合にはプロブレムリスト (PL) を「綴じ」あるいは回転の軸とした螺旋構造を成す、としている (診療スパイラル)。

そして AE : Assessment and Evaluation (評価) と GA : Goal (目標) が補足的に機能して、診療ベクトルの修正、あるいは設定されていた停止条件 = 計画の終点 (EP : EndPoint) との合致性比較によって然るべき対処が為される (はず・または・べきである)、としている。

この発表に引き続いてさらに幾つかの報告が為されており、それらは、熟考過程のみならず決断過程、heuristic な思考過程の考察、思考素材の扱い、などが主題とされている [医療情報学連合大会論文集 15: 569-570, 1995] [医療情報学連合大会論文集 16: 834-835, 1996] [Proc M Tech Assoc J 24:90-94, 1997]。

本研究は目標決定空間の内部構造を定式化し、そのうえで問題定義空間・目標決定空間・計画策定空間から現実世界へと写像された要素ととの一部についての連関形成を定式化するものである。なお End Point 設定は、Goal 設定と共に目標決定空間に存することになる。

よってこの概念モデルを採用することとした。

B. 3 倫理面への配慮

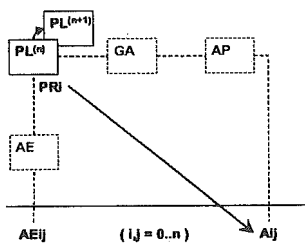
本研究は臨床実データを用いないので倫理的な問題は生じないので特段の配慮を要しない。

【附】 前研究までの成果

本研究結果の理解を円滑とするために前年度までの成果概略を附記しておくこととする。

附. 1 概念モデルの短絡化

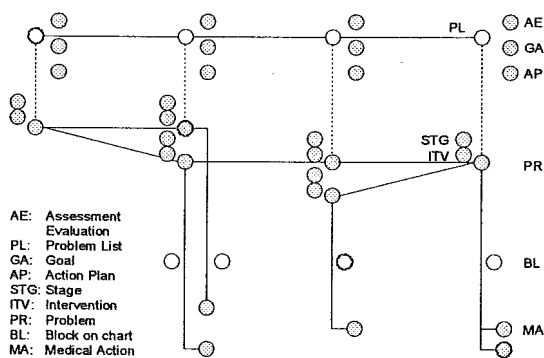
病名/プロブレムの変遷と、病名/プロブレムと診療行為を関連づけるモデルを策定するため、B. 2. 1 の概念モデルを以下のように書き換えた。



すなわち (i) 病名/プロブレムと診療行為とを短絡したうえでその関係と、さらに (ii) 病名/プロブレム変遷の関係を記述した。

附. 2 診療過程のグラフ化

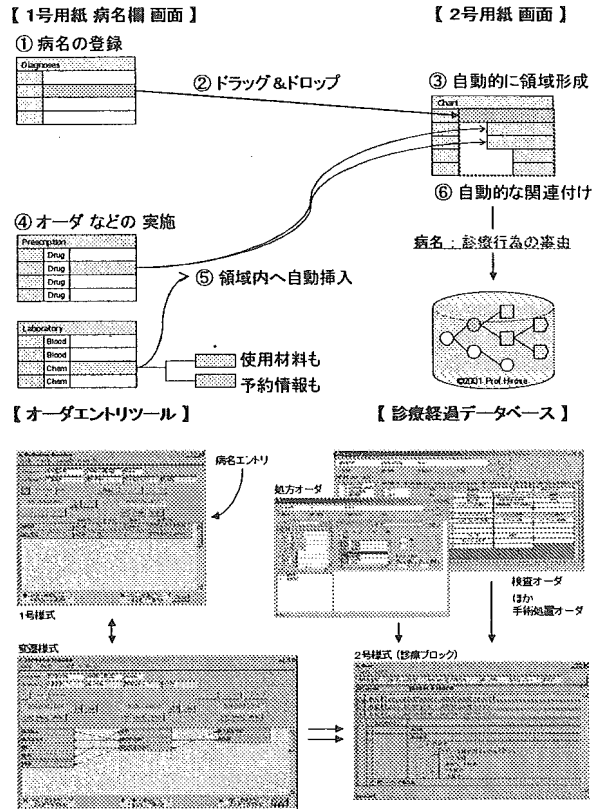
その結果、診療歴の病名や診療行為等を示す infoNode は、下図の如くグラフ構造を形成することとなる。



附. 3 診療記録への写像

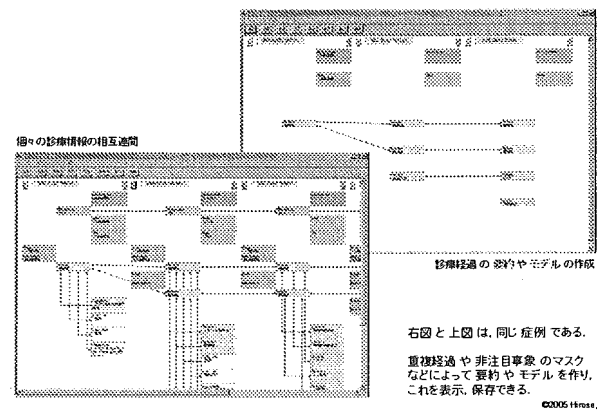
この情報塊を診療録に写像する、つまり診療録としての情報塊へと構造を調えるには幾つか手法があろうが、前研究では human interface から database までなんら変換することなく、ユーザの意図した意味構造を capture する枠組みとした。

その結果、通常の一号様式や二号様式のようにリアルタイム表示することもできるし、さらに次項で述べる利点も併せて得ることができた。



附. 4 グラフ化による成果

下記は、長大な診療履歴情報を自在に要約して供覧したり共有したりする tool の例示である。



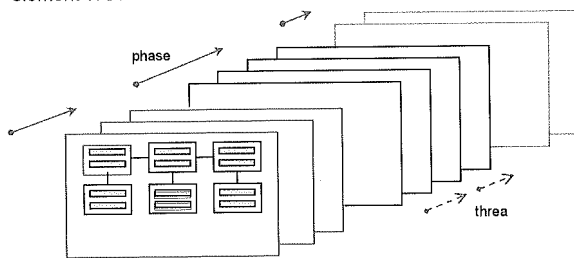
このほか、レセプトや DPC における病名と診療行為の対応付けや、精確な経営分析にも資することは勿論である。

C. 研究結果

C. 1 思考過程の概念モデル

C. 1. 1 プロセスの俯瞰

Course: whole phases of clinical process
 Phase: chain of Threads
 Thread: relatively short chain of Cycles
 Cycle: decision process at an encounter
 Space: portion of a Cycle
 Block: step of decision process in a Cycle
 Cell: element of Block



まず診療プロセス全体を俯瞰しながら、各構成要素を定義命名していく。

ある関わりのなかでの (actField: H15-医療-050) ある患者の診療プロセスを Course と呼ぶ。

Course は幾つかの相 (Phase) に分けることもできる。医療介入を行う医師は、通常、診断相、加療相、継随相を意識するであろう。ただしこれらの三相は明確に分離されるわけではなく、また後付けとして認知可能となることも少なくない。

個々の Phase は多くの診察から成っているが、医師の心が、ある患者の状態に焦点してその問題を考察(再考)し・その解の目標を決断し・目標達成のための戦術を策定する・一連の・かつ単位的な思考の流れを Cycle と呼ぶ。一回の診察診療を想定している。

外来診療では一日に一回の Cycle が発生し、数日または数ヶ月を経た後に、再び Cycle が発生することになる。ICU などでは、一日のうちに幾度か Cycle が発生することもある。いずれにせよ Phase は複数の Cycle の集まりとなるが、それらは雑然と並んでいるのではない。

むしろ診療経過のなかでの意味ある纏まりとしての Cycle の sequence を見てとることができる。これを Thread と呼ぶ。Thread は通常、目標やエンドポイントを尺度とする自己評価的に動作する、とも言える (後述する)。

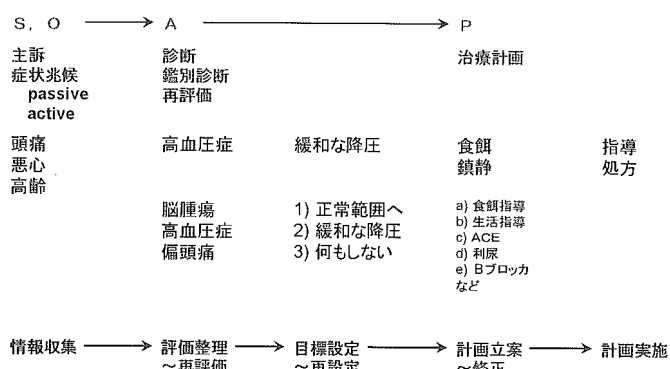
個々の診療 Cycle は幾つかの構造的な思考段階を内包している。これを Space と呼ぶ。一般的には Plan-Do-Check/Study-Act cycle として定式されることがあるが、少なくとも医療あるいは医師の思考過程を必要かつ充分なようにモデリングする際には、そのような定式は不十分もしくは不明確である。そもそも、いったい何のための Plan だったのだろうか？

よって本研究では Space を問題定義空間、目標決定空間、計画策定空間の三つとして捉える。次に Space をさらに細分化して認識する。すなわち個々の Space には幾つかの思考段階がある、として、それらを Block と呼ぶ (後述)。

そして各 Block には、その Block の挙動に応じて、病名やプロブレム、症状や兆候、診療の目標や計画、個々の医療行為 (介入) 等という事項が格納される。それらの事項を総称して Cell と呼ぶ。

Course, Phase, Thread, Cycle, Space, Block, Cell は階層構造を成し、また互いに意味関係で結ばれることになる。また後述するように、時間経過の向きに基づきながら、個々の要素は互いに影響しあうことになる。

C. 1. 2 診療経過の例



具体例を想定すると理解の助けになると思われるので、左図を挙げておく。

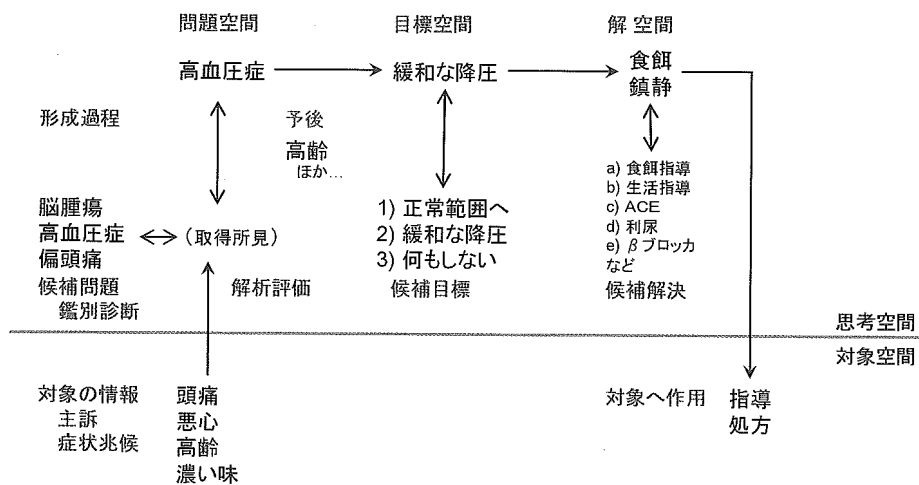
Weed の POMR (Problem Oriented Medical Record) に則った整理として描いており、Progress Note に焦点し、Problem List は左図からは割愛している。

Weed は所見 (observation) を Subjective, Objective に弁別するよう示唆し、それに意義を見出したようである。これは当時の検査や診断技術に依るものと推測される。

惜しむらくは、診療目標やエンドポイントが明確に設定されておらず、評価 (Assessment) の一部として押し込められていたこと、ではある。

主任研究者は 1990 年代前半よりこれを意識しており必要となる概念モデルを提唱していた。ただ当時は (医療 IT 業界での一般的な) 実装に結びつけるための適切な定式化手法を見出せず (もちろん Smalltalk や Lisp や Prolog あるいは ASN.1 や LOTOS はあったし、UML も徐々に広まりつつあったが)。本研究では過去の研究を礎としながら、このあたりを明確化していく。

C. 1. 3 サイクル (Cycle)



上述した仮想症例について初診時 (と引き続く診断相) の思考過程の構造化を試みる。

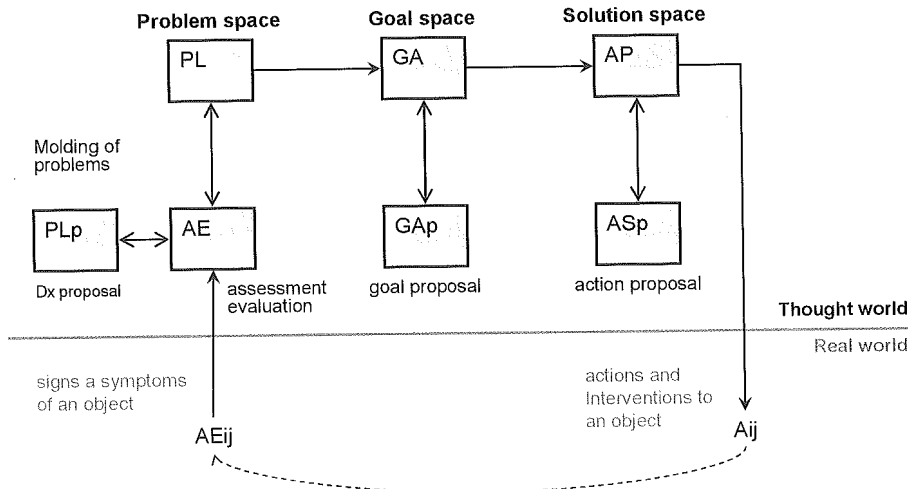
まず現実世界と思考世界 (問題解決空間) とを分離しておく。

現実世界は、確かに現実世界として存在しているであろう。とはいえ思考空間にては『現実・である』と認識されうる事象とは、思考が、外界を認知する際の初段に、その対

象事象を (正しくか否かは別として) 取り込んだ場合のみに過ぎない、ことに留意しておく必要がある。

その後、臨床では、鑑別診断の検討など幾つかの複雑な思考作業が実施され、病名やプロブレムが挙げられていく。その問題に対する解決の目標が決断されるが、その際にも複数の候補が挙げられ、状況に応じながら一つの目標が選ばれる、つまり目標が決断される。

次にその目標を達成するために戦略や戦術が練られ、かつ具体的な手法を、やはり候補のなかから採択していくこととなる。これが計画策定の過程である。そして Cycle の最後に、それらの具体の手法が現実世界へ戻される、すなわち何らかの介入が実施される = 現実化されることになる。



この状況構造を Block 化して左図が得られる。

AEij は現実世界にある (あった) 種々の事象である。事象なので、過去の介入なども含むこととなる。AEij とは事項ゆえ Cell として AE に含まれる。

AE (Assessment and Evaluation) は Block である。これ、または此处は、問題定義していく際の思考作業の場として、主要な役割を果たしている。また PLp

は病名 (診断名) の候補をプールする Block である。

GA (Goal) は目標を格納する Block であり、GAp は目標候補をプールする Block である。

AP (ActionPlan) は計画を格納する Block であり、ASp は個々の単体の介入行為事項や特定の診療プロトコルすなわち一連あるいは組みの介入行為事項のうち、目標に即する群をプールする Block である。

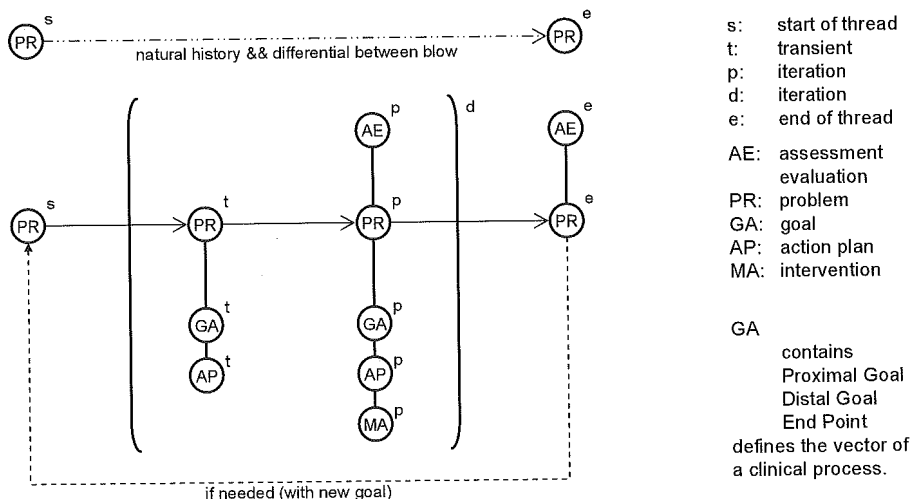
個々の Block や Cell の関係など仔細については後述するが Cycle の概念構造の概略は以上である。医療ドメインにおいては 初めに問題の定義ありき なのである。

これは医療ドメインの思考過程を特徴づける重要な点であり、また問題定義こそ、最も困難な思考作業の一つとなっている。そのうえ問題は、その種類や性質はもとより、優先度や重みも随時変化していくので、問題を解決するには、それらの変化変遷に追従していかねばならない。このような変化変遷の認識とそれへの追従もまた併せて、診療 (のための思考) の困難さを増している。

したがって、診療支援システムなどという語を真に使おうとするならば、当然ながら思考の流れに即した human interface その他の機能を用意するに留まらず、上記のような負担を軽減することが期待されるのである。

C. 1. 4 スレッド (Thread)

前述したように個々の Cycle は Phase を成すというよりも先ずは Thread を成している。



Thread は Goal によってその消長が決定される。

Goal は三つの portion から成る。近位目標 (pGA: proximal Goal), 遠位目標 (dGA: distal Goal) そしてエンドポイント (EP: EndPoint) である。

Thread は通常、以前の前提状況を継承しつつ開始され、固有の dGA と

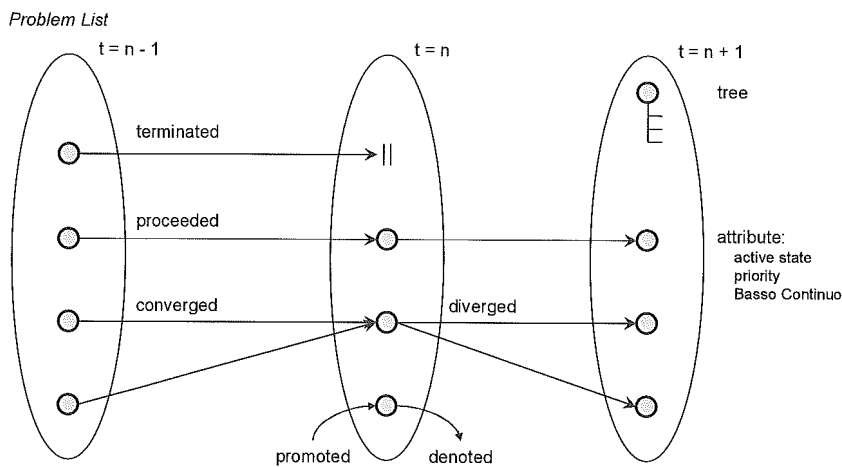
EP とを有している。

よって Thread は目的的な連なりであって、以下のように消長することになる：

- ・連続しうる Thread が存在しない場合に発生する。
- ・直前の Thread が終わった場合に発生する。
- ・dGA や EP に影響するように PL が変化しない限り同等の dGA と EP を保持する。
- ・pGA は変化しうる。よって pGA に関わる PL も変化しうる。
- ・dGA が達成された場合に終わる。
- ・EP つまりは停止条件と合致した場合に終わる。
- ・dGA を目指しつつも pGA にも対処するよう Cycle を繰り返す。

したがって Thread レベルという局所プロセスでは、GA の達成と GA と現実 AE_{ij} との比較に眼目が置かれることになる。ただし後述するように PL の軸性が失われるわけではない。

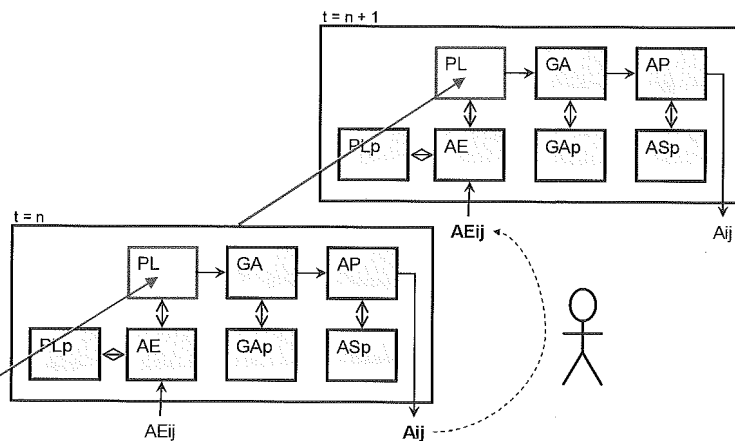
C. 1. 5 スパイラル



Weed は POMR において個々の PR (Problem) の変化や変遷を意識するよう求めた。

変遷パターンには左図の類型がある。それぞれは前研究にて既に述語化している。

Cycle モデルで見たように、Cycle は問題定義から始まり、また Tread は PL つまり PL に格納される PR_i および状況によって決定される GA と EP によって「駆動」されている。



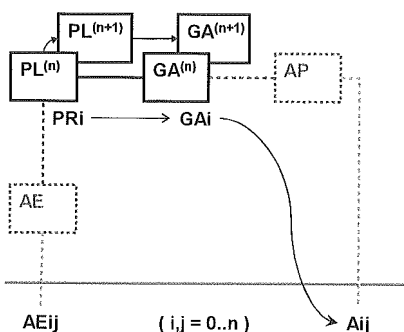
したがって Cycle と Cycle とを繋ぎ Thread として Phase や Course を成す「綴じ」は PL である、ということになる。

現実世界に『うつされた』介入事項 A_{ij} またはその結果は患者を介して新たな事象 AE_{ij} として立ち現れて来るのであるから、結局 Course 全体としては螺旋構造を為している、と理解される。これを診療スパイラルと呼ぶ。

C. 1. 6 Goal の意識

B [附] で述べたように前研究では PL と A_{ij} との短絡を実現することで種々の成果を得た。

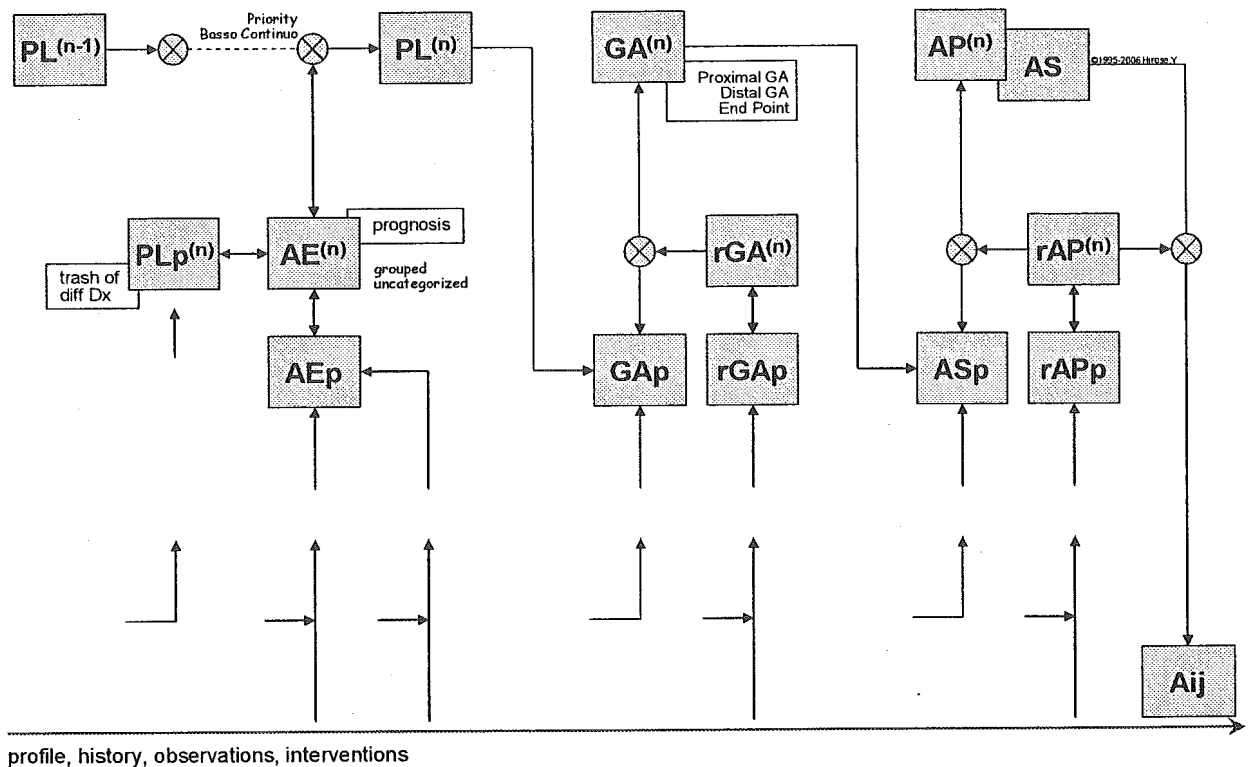
本研究はこれを発展させ、左図のように GA を絡めた定式とする必要がある。考慮すべき要素は 2 を超えているので相当に複雑であり、安易な扱いは危険である。よって思考空間全体を扱って、そのなかに配座することが妥当である。



C. 1. 7 Cycle の形式化

Cycle 全体を眺めながら未定義部分を定義していく。

下図における命名記法は、語幹は大文字 1 字または 2 字, prefix なる小文字 r は「理由づけの場合」であることを示し, postfix なる小文字 p は候補事項となる Cell を保持するプールであることを示し, suffix なる小文字 i や ij が付された information object は事項 Cell であることを示し, superfix なる (n) 等は何れの序の Cycle に属しているのかを表している。



上図は Cycle⁽ⁿ⁾ の状況を表示している。したがって PL⁽ⁿ⁻¹⁾ は直前の Cycle⁽ⁿ⁻¹⁾ に属する PL である。そして PL⁽ⁿ⁾ へと変遷させられるのである。

また AE_p は現実世界から採取された AE_{ij} を格納している。その全てが AE において活用されるわけではないので、AE と AE_p とを分離している。

PL に基づいて GAp が生成され、その素材 Cell を元に GA の事項 Cell (内容) が決断される。その決断には当然ながら事由が伴うが、その事由とは現実世界あるいは思考世界の他の格納構造 Block 内の或る事項 Cell に依っている。その属する Cycle の序は n とは限らない。それら候補は rGAp に、そして実際に事由として採ったものは rGA に格納される。

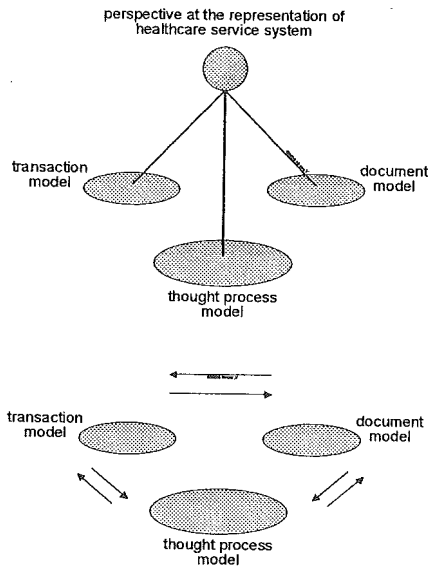
GA に基づいて AS_p が生成され、以下同様にして AP が組み立てられる。事由の参照と構成も同様であり、それぞれ rAP_p と rAP に格納される。なお AP⁽ⁿ⁾ に格納されている個々の intervention は、早晚、現実世界に〔うつされて〕実施されるだろうが、その際、最終的な検証が行われる。rAP はその役割も担っている。

さて GA には三つの portion があり pGA, dGA, EP であることは前述した。pGA と dGA は理解し易いことと思われる (例: 血糖値をコントロールする, 重要臓器の合併症を防止する)。EP の定義は Thread の dGA を変容せしめる状況 (のうち dGA を除いたもの) となり、Thread の定義と表裏一体となる。通常の診療では暗黙的、あるいは dGA の否定的実現であることが多いだろう。ただし臨床試験においては通常明示される。

C. 2 Perspective

幾つかの事由から本節を考察し、また本書でも取り上げることにした：(i) 特に infoNode の属性 category と kind の位置づけの明確化および理解を助ける説明の用意、(ii) 実装業務システムにおける情報の再利用や再構成、(iii) それらによる CSX の特徴の明確化。

C. 2. 1 ドメインへの観



世には様々な domain を想定することができるが、それらの domain に対する観点もまた同様である。

診療情報システム構築というタスクにおいても、文書という観もあれば、ITシステム、そして思考過程という観もあり、さらに経営管理という観もあるだろ。

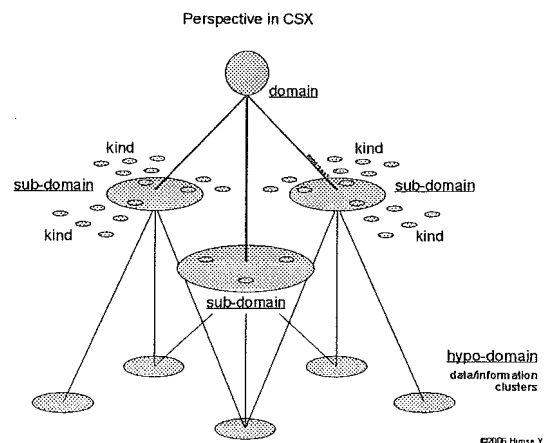
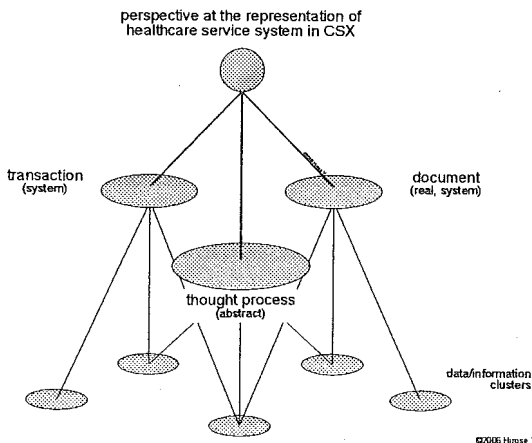
情報システムを導入する際、特に業務システムの場合、通常一つの業務に適する様相で構築される。言い換えるなら一つの観に基づいて設計された情報が蓄積されるということである。

とはいえ蓄積された情報は様々な切り口、つまり異なる観で眺めてみたいものである。このようなニーズを満たすには通常 data warehouse などが利用されるが、実現するためには様々なコストを支払う必要があり、それでさえも限られた観しか反映できないことが多い現況だろう。

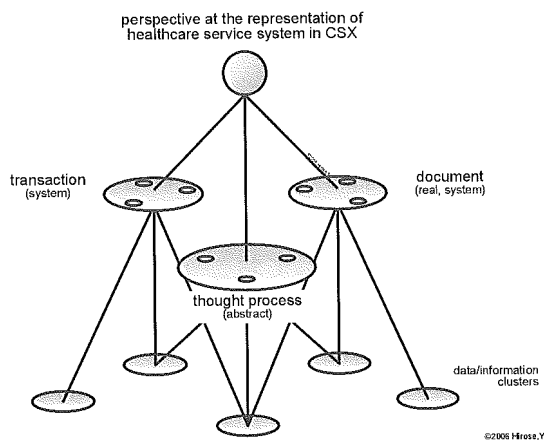
C. 2. 2 Perspective と category/kind

ある domain に対して taxonomical な perspective を設定するとしよう。その際の最上位の層には幾つかの perspective を categorize することができるだろう(例:文書, ITシステム, 思考過程)。その下位の層にはさらに subdomain を categorize できるだろう(例:問題, 目標, 計画)。

CSX の infoNode, arcScope, infoArc はそれぞれ属性に category と kind を持つ。ある domain への perspective は infoNode@category に担わせることができ、上位下位関係の表現は arcScope と infoArc とが担わせることができる。ある (sub) domain とは圏 (category) と考えることができる。そこに幾つかの族 (family) を想定するとき infoNode@kind で表す。そして category も kind も hierarchical な表現を許すので、kind の値もまた taxonomical に構成することができる。



座されている特定の container infoNode (Block) に格納されている。そして、個々の Cell がどの subdomain のどの Block に格納されるのかは、まさに perspective に依存している。



思考過程という perspective においては、PRi や dGai は重要な Cell なので、上位の subdomain に配座されたり、変化に注目しやすいよう配座されたりするであろう。

一方、経営管理という perspective においては、たとえば resource - stock_flow - money という階層のなかで、PRi や dGai は末端の Block に格納されるか、もしかしたらどの subdomain にも現れさえしないかもしれない。

これは当然であって立場や興味が異なれば視点も異なり、視点が異なれば遠近感も異なるだろうし、場合によっては隠れてしまうものもあるだろう。

それでは、money なる Cell はどうであろう。まず経営管理 perspective では、上位または変化に注目しやすいよう配座せられるであろう。思考過程 perspective では、たとえば Problem Space に現れるだろうか。通常は現れそうにないように思えるものの、ただ例えば aspiration では money.coin という形で現れてくるかもしれない。

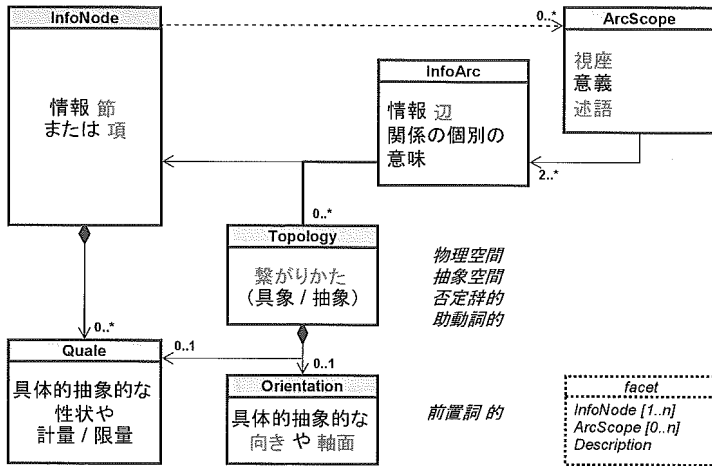
となると個々の Cell となる infoNode において、その内容を示す code は如何にあるべきだろうか。Code 体系は視点を固定して整理されるが、事物は様々な見方や扱い方をされることが常である。よって code 体系自体の視点に依存してしまうような記述表現枠組ならば硬直的であるため、先に述べた業務システムと同様、蓄積情報を様々な切り口で眺められるよう“変換”することも、容易ではなからう。

CSX における perspective は、Block なる container infoNode の属性 category と kind が担う。属性 category も kind も格納する値の hierarchical 構成を許容し、その root も単一とはしない。ために表面上は二変数であっても実際には多軸構成を可能としている。そして container である Block は Cell なる事項 infoNode を格納し、つまり embed し、事象が如何なる観において捉えられたかを表現する。Cell の具体内容はその nodeCode にて示される。

したがって Cell 自体が何物かという表現と、それらが如何なる観で捉えられたかという表現は、互いに独立させている。しかも単純に独立なのではなく観は情報構造の内部に明示的に埋め込んでおり処理系に依存させていない。これら二点から、perspective の変換は容易かつ曖昧さや解釈の齟齬を回避しうる枠組となっている。そして perspective の生成も自由であった。観自在である。

C. 3 CSX

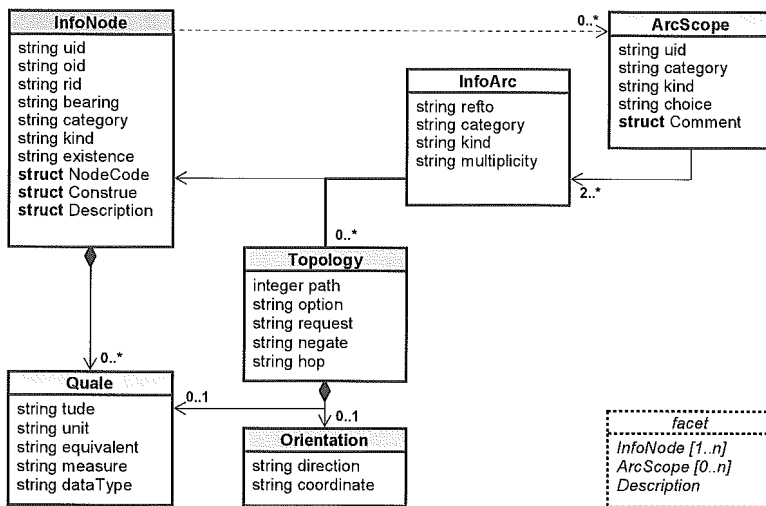
C. 3. 1 メタモデリング枠組



CSX は meta-modelling framework と云えるが、通常のモデリング機能も当然ながら有している。

その要素概略と記述機能の分担を左図に示しておこう。

表現すべき内容が単一でないのは、要素の抽象性に依っている。属性に格納される値によってその具体の役割が決定される。



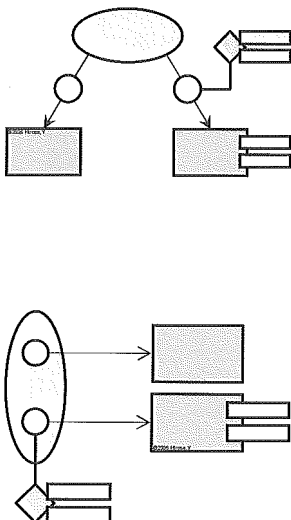
左図は CSX を UML で表現している。

Class diagram の描出は、処理系によって関係クラスを表現できないこともあるので、左図のように記述形式を変えて見せている。

なお命名を一部変更した。

- InfoArc@ref を InfoArc@ref to に
- Dimension を Quale に

また Topology と Quale との関連に表現誤記があったのでこれを訂正した。



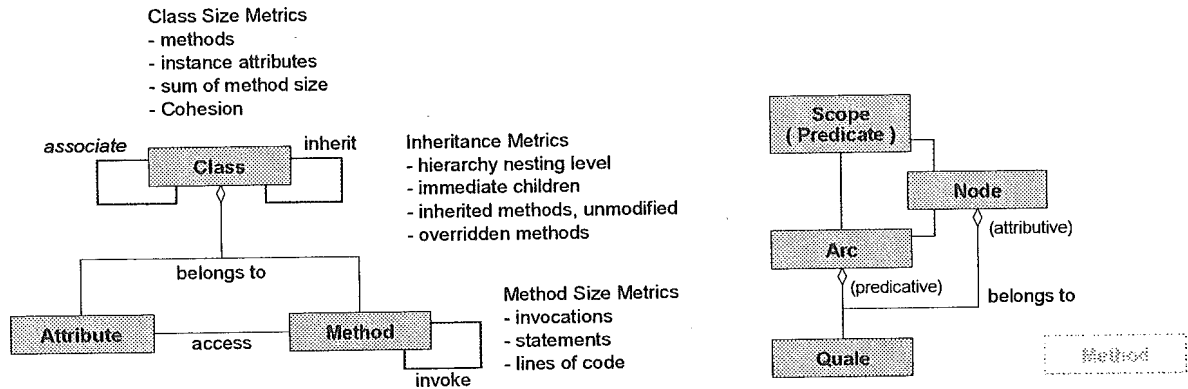
- Information object: infoNode means class, property/attribute.
- Information object: arcScope represents the scope of associations, or predicate/function.
- Information object: infoArc expresses association or meta-association between infoNodes, and the role of them within the arcScope, so if arcScope means predicate, infoArc may represent (deep) case.
- Information object: topology, orientation means topological expression, includes abstract, so also constraint.
- Information object: quale represents each quality or quantity the infoNode has, essentially same as infoNode but is able to be utilized as an attribute in object oriented modeling.

When generating association, the arcScope for the association must always manage it with infoArc. For other associations those are semantically grouped, another arcScope must manage them with other infoArcs within the latter arcScope. Topological expression must be accompanied by an infoArc.

左図は UML には依らない CSX の diagram である。

C. 3. 2 OOM との対比

CSX と OOM は枠組の目的が異なるので、これを考慮しない比較考証や批判は意味がない。とはいえ CSX で表現された情報を OOP で構築する系で処理することなどに配慮するために、二つを対比しておくことは有用だろう。



- ・ CSX では実装処理そのものは意識しないので OOM で云うところの method は持たない。

上述の通り、これは批判に当たらない。そもそもモデルに method が必要か否かということ自体が観に依っている。OOM では implementation のために method は必須だが、少なくとも domain expert 側での OOM では interface が要諦であり method は“説明的な description”に過ぎないとも云いうる。にも関わらず多くの OOM では implementation と interface が混同されている。その意味において、むしろ CSX は OOP における実装設計を邪魔することはなからう。

- ・ CSX では OOM で云うところの attribute と association とを厳密には区別しない。

実際 OOM でも根底思想は同様であって、attribute も association も property の類型に過ぎない。Class に attribute を持たせるのは利便のためであり、また scope の限定でもある。同様のことは CSX でも生じているが、多くの場合、CSX では Scope を明示することとなる。これに伴って Arc も明示され、かつ OOM の立場から見ると、言わば昇格的に扱われることとなる。

- ・ CSX では Node を繋ぐ Arc を明示的昇格的に扱うと共にその Scope をも明示する。

特徴的であり、利点は前述した。なお naive に考えると implementation において闇雲に stack を深くしたり無用な transaction を増やしたりするように危惧されそうだが、これは処理環境と実装手法とに依っている。