

B. fcet.mdic.csx.xsd : core model for facet

```

1. <?xml version="1.0" encoding="UTF-8"?>
   <xs:schema
     targetNamespace="http://www.hosp.u-ryukyu.ac.jp/medi/csx/0.91"
     xmlns:csx="http://www.hosp.u-ryukyu.ac.jp/medi/csx/0.91"
     xmlns:xs="http://www.w3.org/2001/XMLSchema"
     elementFormDefault="qualified"
     attributeFormDefault="unqualified"
     finalDefault="#all"
     version="0.91c">
10. <xs:include schemaLocation="fcet.elct.csx.xsd"/>
   <xs:element name="facet"
     type="csx:facet.Type"/>
   <xs:complexType name="facet.Type" />
15. <xs:sequence>
   <xs:element ref="csx:substance" minOccurs="1" maxOccurs="unbounded"/>
   <xs:element ref="csx:relation" minOccurs="0" maxOccurs="unbounded"/>
   <xs:element ref="csx:marginalia" minOccurs="0" maxOccurs="1"/>
   </xs:sequence>
20. <xs:attribute name="version"
     type="xs:token" use="required" fixed="0.91c"/>
   <xs:attribute name="oid"
     type="xs:token" use="optional"/>
   <xs:attribute name="doc:facet.ID"
     type="xs:token" use="optional"/>
   <xs:attribute name="doc:facet.Num"
     type="xs:normalizedString" use="optional"/>
   </xs:complexType>
25. <xs:element name="substance"
     type="csx:substance.Type"/>
   <xs:element name="relation"
     type="csx:relation.Type"/>
   <xs:element name="conjugator"
     type="csx:conjugator.Type"/>
30. <xs:element name="marginalia">
   <xs:complexType>
   <xs:sequence>
   <xs:element name="note"
     type="xs:string" minOccurs="0"/>
   </xs:sequence>

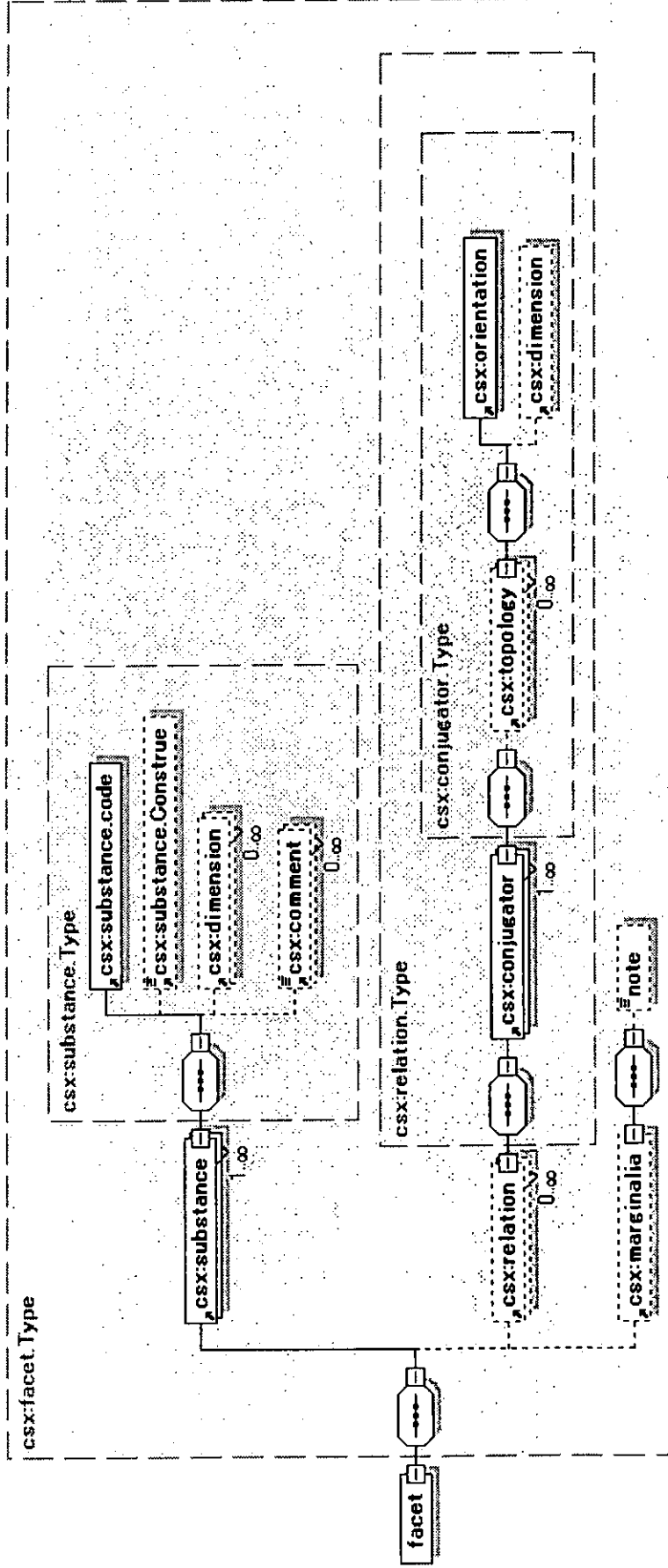
```

35. </xs:complexType>

</xs:element>

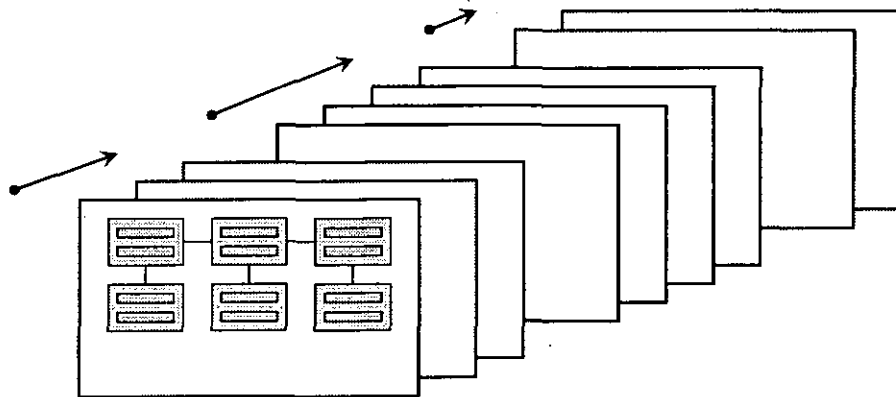
33. </xs:schema>

C. Diagram



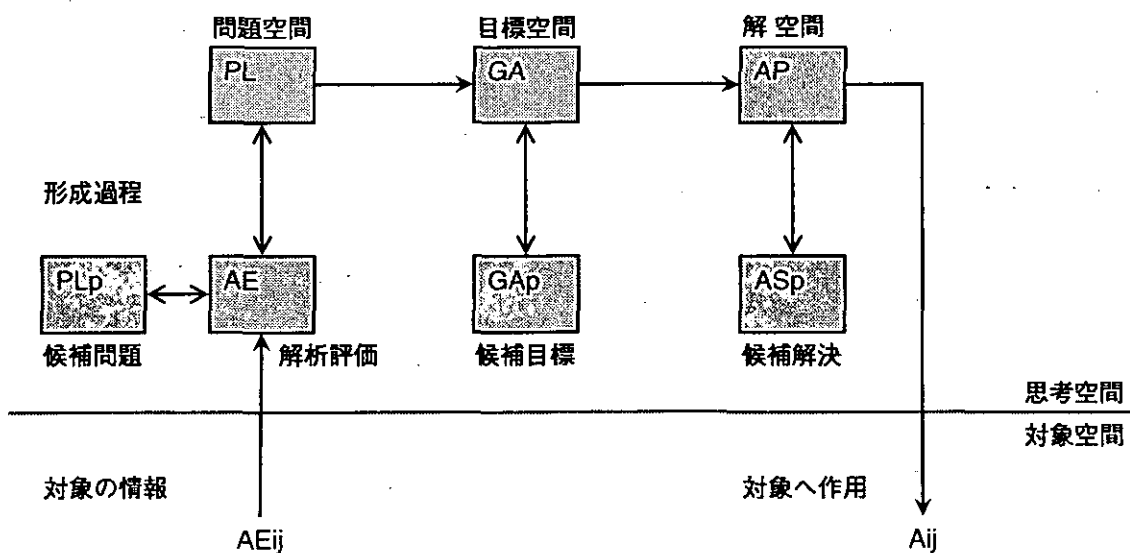
Outline of whole process

Phase: diagnosis, treatment, follow
 Cycle: thinking and making decision
 Block: steps of thinking process
 Cell: elements of the block



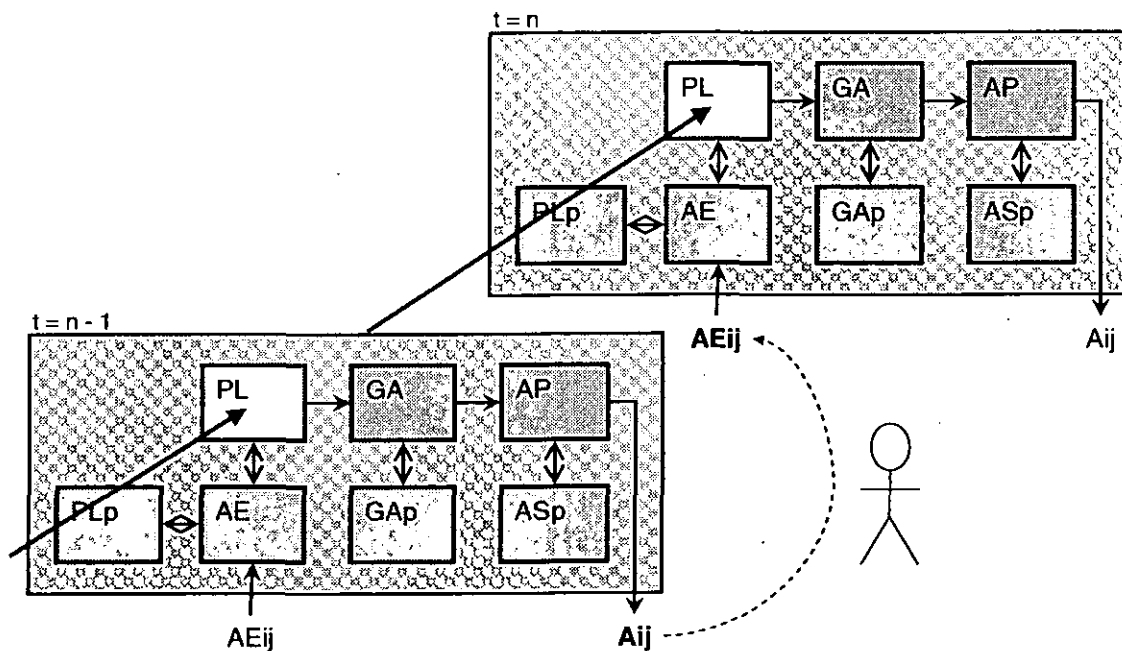
©1997-2004 Yasuyuki Hirose

Outline of the model



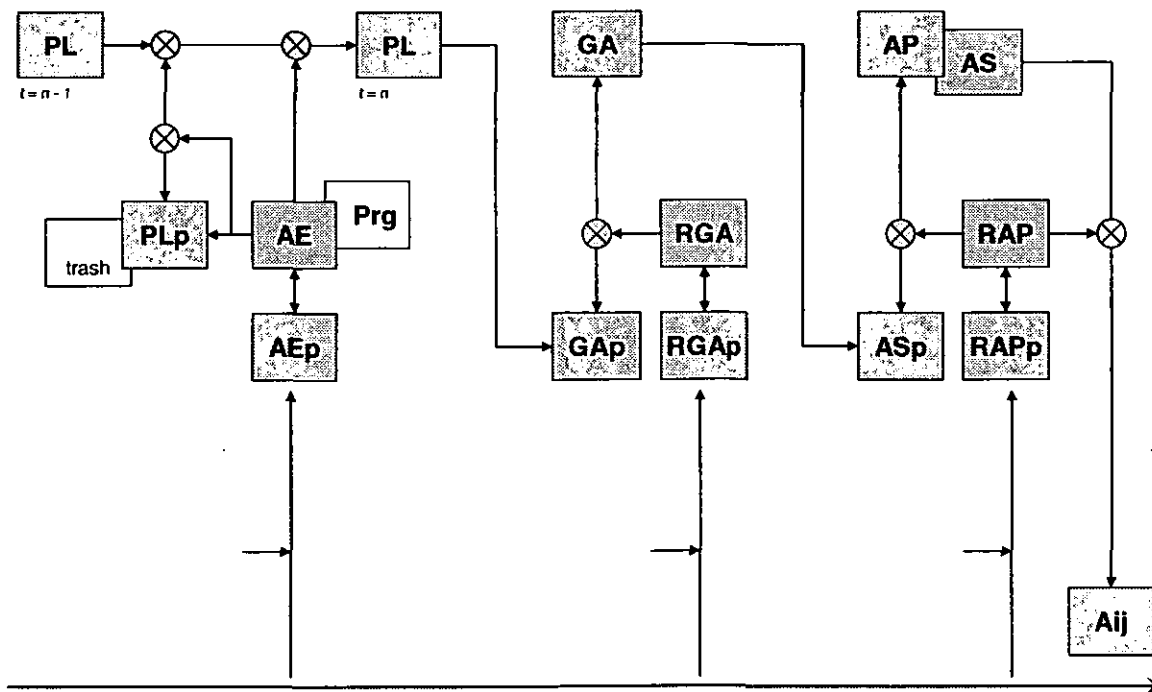
©1997-2004 Yasuyuki Hirose

Spiral structure



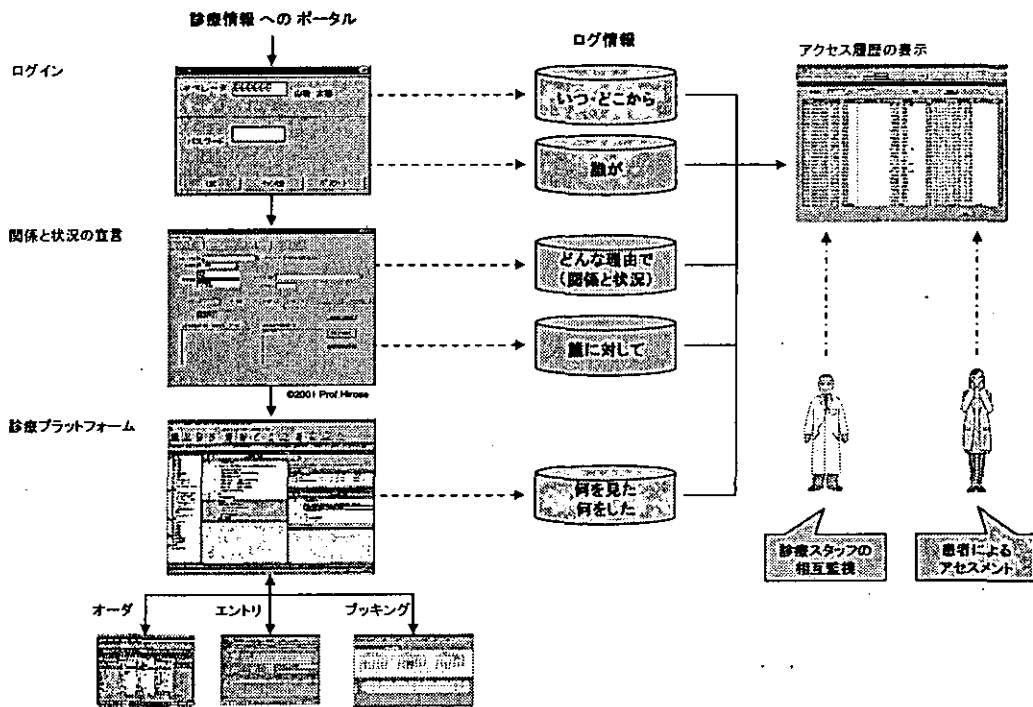
©1997-2004 Yasuyuki Hirose

Whole model



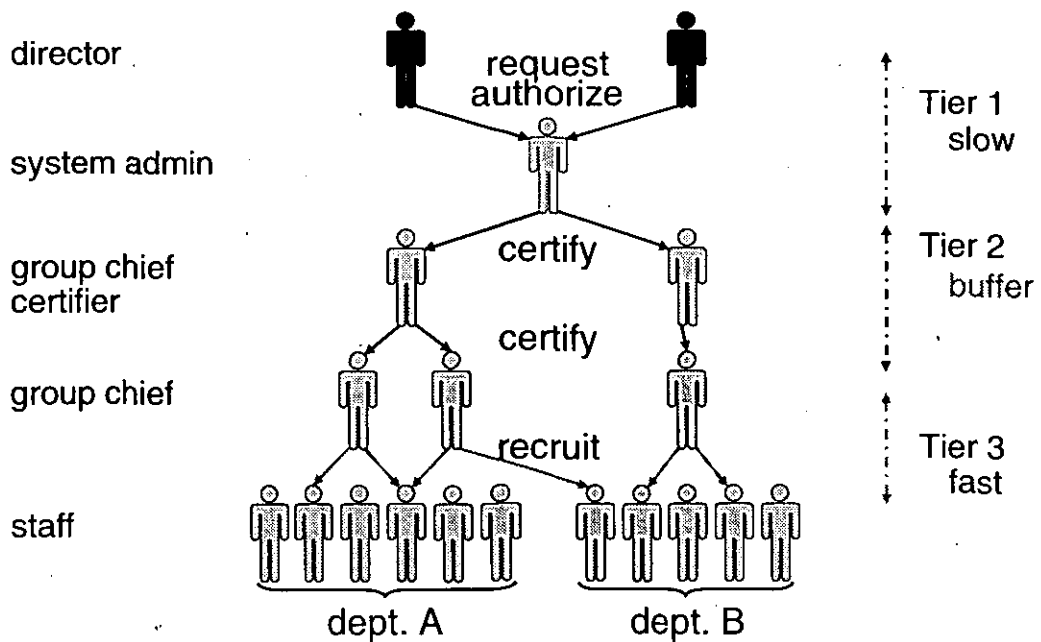
©1997-2004 Yasuyuki Hirose

アクセスの事由と履歴 : 相互監視と監査



©1996-2004 Yasuyuki Hirose

診療グループの形成 : 権限委譲とその現実的な利点



©1996-2004 Yasuyuki Hirose

診療プラットフォームのヒューマンインターフェイス

～モード、コンテナ、ツール～

○廣瀬康行*§, 佐々木好幸**§, 木下淳博***§, 水口俊介^§, 大林尚人^^§, 竹田淳志^^^
 東京医科歯科大学・歯科麻酔学講座* 予防歯科学講座** 第二保存学講座***
 高齢者歯科学講座^ 歯科放射線学講座^^ 第三保存科学講座^^^
 医学部附属病院医療情報部システム開発室 §

東京医科歯科大学歯学部附属病院では既に電子カルテを稼働しているが、次期システムへの移行に伴って現行システムの理念を継承しつつも、これまでの苦い経験や反省点を踏まえつつ、これからのヒューマン・インターフェイス (HI) を再考した。

1 初めに診療録ありき

現行システムではオーダリングシステムからの脱却を主眼にして、病名と診療行為を関連づけながら記録できるような、病歴記録主体の電子カルテを実現した。

電子カルテは「システム内における診療」のためのプラットフォームであり、オーダリングなどに必要な機能は、単なるツールとして捉え直したわけである。

1-1 現行システムの理念

現行システムの特徴は以下の三点に要約できる；

- ・伝票指向型、医事会計指向から「患者病歴指向」へ
- ・Episode と Medical Action との関連性を記録する
- ・参照した画面のログを記録する

診療データは互いの関連性と時間推移におけるその変遷の中にも深い意味と意義が存在する。したがって様々な診療情報は時間経過の中において、関連づけによって統合することを考え、これを実現したのである。なお関連づけされたデータは Medical History File として保存されている。

参照画面の記録には二つの意味がある；

- ・誰が・いつ・何を見たのか
 - ・診療スタッフは何を参照して診療行為を決断したか
- 当然ながら前者はセキュリティとプライバシー上の意義があり、後者については決断過程の足跡の一部を記録したかったのである。なお参照記録は Reference Log Journal に保存されている。

1-2 電子カルテは診療プラットフォーム

上記の事項を実現するに最適なヒューマンインターフェイスとは、「システム内における診療」のためのプラットフォームすなわち電子カルテである。そしてプラットフォーム上に様々なツールを用意してオーダ、病名入力、診療行為入力などを行うようにした。したがって我々は、現行システムを歯科オーダリングシステムと呼ばわれることを嫌っているし、不適切であると考えている。

このような環境では、プラットフォームは仮想患者としてイメージされることを期待できる。というのも、患

者の様々な経過や情報を参照しながら次の診療行為を決定させる環境だからである。このことによって、誤操作によるシステム内での医療「事故」の防止にも役立つし、システム内で「その患者」を見失うこともなくなるのである。

1-3 現行システムのツール

ツールの種類を大別すると、エントリツール、オーダリングツール、ブラウジングツールがある。なお、次期システムではブックイングツールも加えていく。各々のツールに対するアクセス権は各々のツール毎に詳細に管理することができる。

- ・エントリ 病名、歯式/身体部位、処置、所見、指導内容、階層構造メニュー、カテゴリー入力
- ・オーダリング検査、処方、2号用紙画面へ反映
- ・ブラウジング検査結果、2号用紙画面でも閲覧
- ・関連づけ 画面展開にて暗示的・自動的に病名と診療行為との関連づけは、1号用紙から2号用紙への展開によって、暗示的に行っている。

1-4 カテゴリー入力

エントリツールでは、階層概念によりカテゴライズされた入力項目を階層メニューから選択入力できる。このとき、任意の階層レベルでのカテゴリー入力も可能としている。

たとえば、虚血性心疾患をシングルクリックすれば該当項目が選択入力できるが、これをダブルクリックした場合にはさらに下層概念のメニューが提示されて、左壁心筋梗塞などの詳細な病名を入力することができるようになっているのである。

2 次期システムの HI でキーとなる概念

電子カルテの場合、オーダリングシステムとは全く異なった考え方によって操作環境を組み立てる必要のあることをまず確認しておきたい。単に「選択メニューからデータを入力できれば良い」といったものではないのである。

2-1 視座と場と文脈

というのも診療スタッフは各々の診療現場の状況に応じて、様々な視点や観点に立脚して、適切な様式で表現された文脈すなわち病歴という「場」において、考察と決断とを迫られるからである。

一方、入力すべき事項は単なるオーダリングシステム

に較べて各段に増え、これに伴って必要な手続きや手順も増加する。したがって何らかの方策により入力ステップの削減を支援することは、システムにとって本質的でさえある。考えられるキーとしては；

- ・適切なデフォルト
 - ・その入力操作状況（または文脈）に応じた一括処理
 - ・異なる操作ブランチへのジャンプ
- などが挙げられよう。

視座や思考の場にせよ、入力操作の状況または文脈にせよ、その「状況」における目的やアクセス方法の要求に応じた多様性と可塑性、あるいは複合的な機能を提供できる環境であることが必須となるのである。

さらに、大病院では一人の患者に多数のスタッフが関わっている実態も無視できない。したがって診療プラットフォームは、一面、交流の「場」とも成らざるを得ないことにも留意すべきであろう。

2-2 キーワード

2-2-1 視座と文脈

電子カルテの優位性の一つに、表現形式や表現様態を自在に制御選択しうるといふことがある。一方、カルテは記録媒体であるとともに思考の場を提供していることに留意すれば、診療スタッフの視点観点や思考様態にあわせて複数の表現様態（モード）を提供し、その場の要求によって選択できる機能の装備を考えることは、当然の帰結といえよう。

もっとも、紙カルテもしくは紙カルテを模した電子カルテは、形式的構造は有するものの意味的・文脈的記述に適した構造を有していると言えるものではない。ということシステムとしては、診療の流れを事後に解釈／追跡／再現するに適した記録構造とヒューマンインターフェイスを有しているとは言えないことになる。これを解決するために三つの機能考えた；

プラットフォームに、思考過程を直接的に表現できるようなモードも用意しておく。

後述する「引用」によって、診療データを文脈ごと指摘できるようにする。

プラットフォーム（たとえば電子カルテ）とツール（特にブラウザ）とを連動させた、retro-spective/pro-spective なトレース機能を設ける。

繰り返すが、これらの機能は日常的に必要となるわけではない。しかし思考の場を提供するには必須となろう。またこれらで全てが解決できるわけではないのだが、相当程度の改善と進歩は期待できよう。

2-2-2 多様性と可塑性

ヒューマンインターフェイスへの要求は個別のもしくは個人的にならざるをえないことを思うに加えて、前述したように、同一の個人においても様々な状況に応じた要求が発生せざるを得ないことを考えれば、多様性と可塑性は不可欠な機能要素である。

またこれは、ヒューマンインターフェイスの様々なレ

ベルと切り口とで求められることになる。レベルについて言えば、プラットフォームのレベルでは複数のモードが用意されるし、エントリツールのレベルでは、階層化選択メニューによる入力も、総リスト検索による入力も共にできる、ということである。さらに選択メニューは、診療科や医師によって個別に調整できる必要がある。

これとは異なった側面で言えば、たとえば、患者と医師との「関係」と「状況」ならびにそれらの変化に応じて、プラットフォームにおいてアクセスできるツールやコンテナ（後述）が制御される、ということである。

なお、多様性と可塑性とを追求しすぎると発散するのではないかと、という危惧もあろう。しかし反対に、良いツールが何種類か提供されることにより、むしろ結果として収束する可能性があるという考え方もできよう。いずれにせよ rigid な部分と variable な部分、primitive な部分と additional な部分は、いずれも存在するわけであって、多様性と可塑性は variable / additional な部分に求められるべきことは言うまでもない。

蛇足ながら、アプリケーションの標準化を考える際には、多様性を求める・人間の・自然であり根元的な欲求を無視するような愚を冒してはならないようにも思われる。

2-2-3 複合性と多価性

ここでは、ツール機能やデータ型の混在を許容するような環境の必要性を述べる。

冒頭でも述べたが、これまでのシステムは部門の業務形態に深く根ざした伝票入力指向型のシステムであった。しかし診療現場における入力や参照は必ずしも単一の部門業務のみに対応しているわけではなく、むしろ通常は複数の機能を組み合わせて操作している。これを能率的に行うためには、操作ブランチの昇降やジャンプを繰返すのではなく、単一機能のツールを組合わせた複合的なツールによる一括処理を可能とする環境を提供すべきである。

複合ツールはもちろん、「そのとき」の操作状況や思考過程に自動的に対応できるような知的かつ動的なものが望まれはする。しかし静的で目的指向性の強いものであっても、相当の有用性は認められよう。なお複合ツールが動的であれ静的であれ、いずれにせよ複合ツールの作成編集ユーティリティは提供されるべきである。

さて、診療データならびに関連情報には様々なデータ型が存在している。カルテの記述は種々のデータや情報を参照しながら為されるわけであるが、その際、データ型がテキストであれコードであれ、ひとたび入力された後には同一内容を再入力するのではなく、簡単な操作によって活用できるべきである。しかも必要ならば、個々のデータが背景に持っている文脈関係も保持しながら参照できることが最善である。

この要求から導かれる診療データのハンドリングに

関する機能は以下の通りである；

1. 診療データの表示入力領域には、様々なデータ型が混在できること。
 2. ある表示入力領域から他の表示入力領域へ、引用や複製（後述）ができること。
 3. ひとつのデータは、引用も複製もできること。
 4. 引用の場合は、引用元の文脈も保持していること。
- ここに述べた複合性と多価性を提供できる環境によって、入力や記述に伴う煩雑さを解消させつつ、病歴記録の詳細さ厳密さを増すことも可能となるのである。

2-2-4 明示性と暗示性

多機能さは利便性を高め正確さを増す一方で、複雑さ煩雑さや解り難さを増大させてしまう側面を持っている。このような負の側面を軽減するためには、状況に応じて、あるいは選択によって、多機能さや複雑さを隠蔽してシンプルに見せる必要が生じる。

しかしただ単に隠蔽すれば良いというものではなく、隠蔽された機能は暗黙の了解によって適切に自動処理されなければならないのである。

たとえば、現行システムでは病名と行為との関連づけ（リンク）は、1号用紙画面から2号用紙画面への展開の中に隠蔽した。これは操作を単純化した反面、画面の親子関係を強固にせざるを得えなかつたり一意対応にしか適用できないという欠点を持つものである。逆にこれらの束縛を除けば自由度が増すとともに多機能・詳細かつ正確な指定が可能となるが、一方では、ハンドリングは複雑になってしまう。

そしてヒューマンインターフェイスには、これらの双方をサポートすることが求められているのである。

2-2-5 交流の場

誤解を生じやすい表現であるが、要は、大病院では一人の患者に対して多数のスタッフがケアを行っている業務実態から考えて、異なる職種の異なるスタッフが入力したデータや情報の相互参照を必要以上に束縛しない配慮と、それを支える機能が必要であるということである。

そのような方策の一つに、診療行為の入力だけではなく診療データの閲覧の際には、スタッフと患者との「関係」と「状況」の宣言を必須とし、閲覧者や入力者のリストを開示できる機能が挙げられる。

リストの開示が院内もしくは患者自身に対して行えるなら、関係法規によって言動を規制されている診療スタッフの管理を最小のコストで実現することが期待できるからである。もっとも、このような方策のみで充分であると主張しているわけではない。

一方、診療データは様々なスタッフや機器あるいはアプリケーションに組み込まれた関数などから入力・収集されることになる。データの信頼性の評価は情報ソースに依るところも大きいので、情報ソースの記録と明示は必須の機能である。

2-2-6 その他

・自動転記／連動／連携

自動処理による負荷の軽減のためにプラットフォーム～ツール間などの、このような機能は不可欠である。

・近いことと遠いこと

ツールなどの機能呼出にせよ入力事項にせよ、「その状況」における近いことと遠いことが区別できるなら、ヒューマンインターフェイスの操作性は各段に改善されるはずである。しかし実現のためには、システムが知識を有していなければならない、などの条件が必要である。したがって現時点では解を有していない。

3 次期システムの HI での具体化

3-1 プラットフォームとモード

診療プラットフォームと、ナースング・プラットフォームの二種を用意する。

診療プラットフォームには電子カルテ、電子カードックス、診療論理ワークベンチという三種のモードを、またナースングプラットフォームには、通常の看護記録モードと看護計画モードを用意する。そして状況に応じた適切な表現様態によって、病歴という文脈を表現することができるようにする。

3-2 コンテナ

コンテナはプラットフォームに付随する容れ物であり、関連する情報をひとまとめにして整理する領域である。コンテナの例は以下のとおり；

基本情報、

各種一覧（感染症、担当医、入力者、閲覧者）

予診、プロフィール、プロブレム、SOAP

各種管理番号、各種サマリ、各種診療文書

看護区分、看護サマリ

3-3 ツールとフォーム

ツールの種類については既に述べたとおりであるが、これに加えてフォームも提供する。フォームとは、病態別疾患別、あるいは一塊の定型的な診療行為に応じた複合ツールのことである。またフォームの作成編集ユーティリティも同時に提供する。

3-4 反映／自動転記／連動

・反映

プラットフォーム、ブラウジング・ツール、エントリ・ツール、コンテナそして部門のサブシステム間では、当然ながらデータは相互に反映され、参照や引用も可能とする。すなわち、一つの情報を様々な観点や考え方で見ること／書くことができる環境を提供する。

・自動転記

自動転記とは、オーダリング・ツールの呼出に伴って、プラットフォームやコンテナとツール間において、必要事項情報の自動的な受渡しをする機能のこと。

・連動

プラットフォームとブラウザの連動とは、いわば、適正診療デバッグをステップワイズに走らせて、診療結果レジスタを見るような機能のことである。

3-5 複製と引用

入力では、複数のアクセス方法を用意するとともに、複製によってツールの起動そのものを省略できるようにする。なお引用によって、既入力事項を文脈や因果関係とともに指摘参照しながら、新しい文脈を作ることができるようにする。

・複製による入力

コード化された入力項目のうち、既に入力され表示されている項目を「複製」して、新規の入力事項とすることができることである。「複製」は、エントリ・ツールを使っての入力処理と同等とみなされる。このときの入力ならびに編集バッファは、プラットフォーム（電子カルテなど）そのものである。

・引用

プラットフォーム、コンテナ、ツール間でのドラッグ&ドロップ等による「引用」も可能とする。

「引用」は「複製」とは異なっており、背景としては「引用」した情報に基づいて決断が為される、という前提が存在している。すなわち；

「引用」は文脈や因果関係を含んだ参照操作であり、「複製」は単に入力項目の複製作成ならびに仮入力処理に過ぎない。

なおコードとテキストの混在も可能で、コード化された項目は「引用」可能とする。

3-6 バッファリング/診療行為ブロック

入力事項のバッファリング&エディットを、それ専用の領域で行うことは自然とは言えない。そこで入力事項のバッファリングと編集を、プラットフォームにおいて行えるようにする。つまり、たとえば2号用紙における操作環境では、電子カルテの新規入力欄が、そのままバッファリングと編集のための領域となっているのである。

・診療行為ブロック

この際、診療行為ブロックと呼ぶ複数の領域を設けることができるようにする。

「2号用紙」を確定するまでの間は、入力された種々の項目は、ドラッグ&ドロップ等の簡単な操作で、診療行為ブロック内の移動・追加・削除や、診療行為ブロック間の移動などの編集ができる。

診療行為ブロック内の診療行為は、対応する病名等と関連づけられている。保険点数等は診療行為ブロック毎に自動計算され、表示される。

・関連づけ

診療行為ブロックによってエピソードとアクションの関連づけを行うとともに、自由な入力順序にも対応することができるようになるのである。

3-7 入力順序の非束縛性

保険傷病名入力は、処置等の診療行為に応じて傷病名を選択できる。診療行為を入力する前に保険傷病名を入力してもよいし、診療行為を入力した後に保険傷病名を入力してもよい。

この機能は、前述のバッファリング&エディット/診療行為ブロックという機能が前提条件となって実現できるものである。

なお、この機能によって法的な問題を生じる可能性は無い。というのも、確定処理前の編集段階においてのみ、項目入力に関して時間的な前後があるのみであり、診療行為項目の確定入力処理後に保険傷病名を後づけするわけではないからである。

3-8 アウトライン表示

2号用紙における閲覧機能である。我々は、院内の事情によって、各診療科ごとの2号用紙「シート」や、診療日ごとの「シート」を考えていない。したがって2号用紙の表示様態は事実と評価の羅列になるわけであるが、このままでは読みにくいばかりでなく理解もしにくい。

そこで、オペレータの多様な要求に合わせて表示できるような、自由度の高いアウトライン表示機能の必要性を感じた。

アウトライン表示機能における検索項目は、以下の通りである。これらを個別もしくは複合して検索し、焦点となっている事項のみをアウトライン表示するものである；

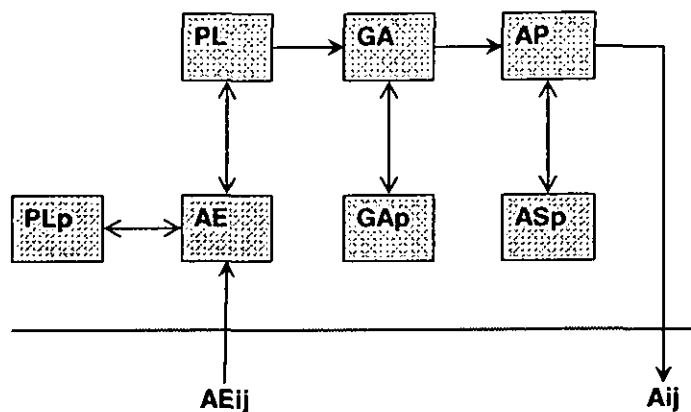
部位名、病名もしくは保険傷病名、プロブレム名
診療行為名（処置項目、検査項目、薬剤等）、
技工指示書、診療科部局、担当医名、
診療グループ名、診療行為者名

以上のようなヒューマンインターフェイスによって、次期システムを構築していく予定である。

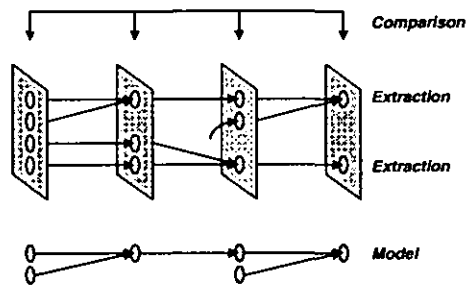
参考資料

- 1) 平成3年度国立大学医療情報部門連絡会議. 1991.
- 2) 第12回医療情報学連合大会論文集. 815-816. 1992.
- 3) 第12回医療情報学連合大会論文集. 333-334. 1992.
- 4) 医療とコンピュータ 7(2) 128-141. 1995.
- 5) 日経ヘルスケア 72(10) 103-109. 1995.
- 6) 第15回医療情報学連合大会論文集 569-570. 1995.
- 7) 平成7年度国立大学医療情報部門連絡会議 63-64. 1996.
- 8) 平成7年度国立大学医療情報部門連絡会議 65-68. 1996.
- 9) 電子カルテってどんなもの?. 中山書店. 1996.

思考過程の概念モデル



PL	プロブレムリスト
PLp	鑑別診断
AE	症状や兆候 (充分な吟味の評価対象となった重要な)
GA	治療目標
Gap	その候補
AP	診療計画
ASp	その候補
Aeij	検査結果や症状や兆候
Aij	実施した治療行為など



思考と診療の過程

臨床思考の過程そして診療過程の全体を process, その process なかでのある区切りを stage または phase と呼ぶ。これらは幾つかの局面 (cycle または facet) からなり、各局面はプロブレムリストの変遷を軸としつつ、時間経過に即して連なることになる。

プロブレムとその変遷は、個々のプロブレムと・ある時点のプロブレムリストとで構成される 閉空間にて成立するものではなく、以下の入出力がありうる:

- ・ 吟味の対象となった 重要な症状兆候 (AE)
- ・ 吟味対象の 検査結果または各種報告等 (AE)
- ・ 鑑別診断 (PLp)

文献 [1, 2, 3A, 3B]

変遷の記述と変遷モデルの意義

プロブレムおよびプロブレムリストの変遷を記述すること、並びにその変遷をモデル化することは、知識経験の蓄積とその活用において極めて重要である。

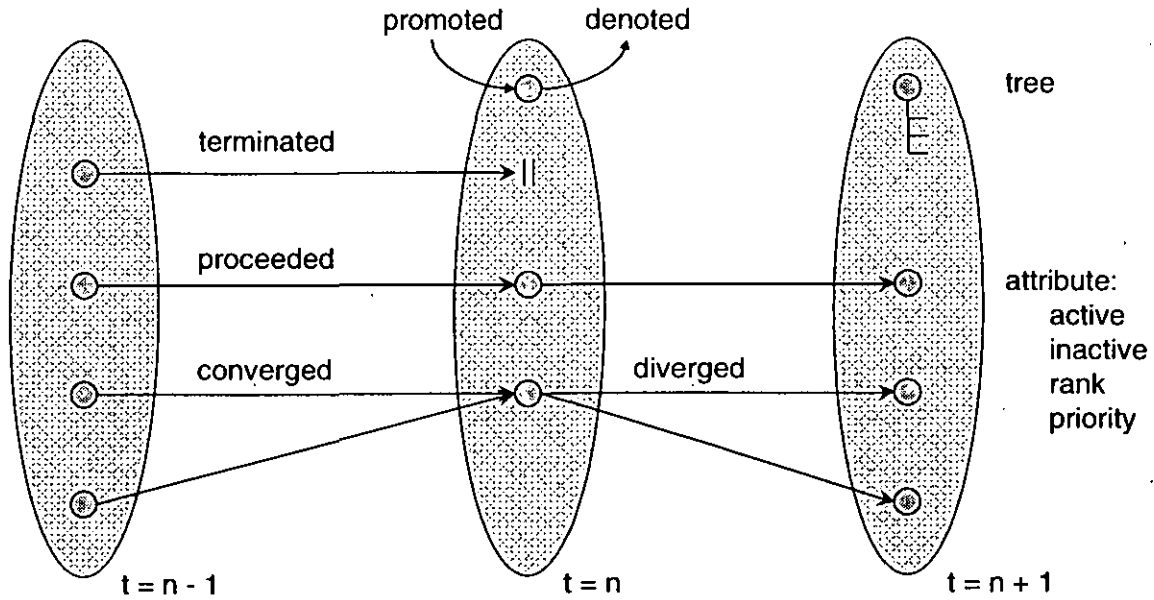
たとえば以下のようなことが自在にできるならば、それは、研究、教育、そして医療経済での適切な制御にも資するところ大であろう:

- ・ comparison (プロブレムリスト全体の変遷)
- ・ extraction (焦点したプロブレムのみの変遷)
- ・ model (プロブレム変遷のモデル)

文献 [4]

プロブレムの変遷

Problem List



変遷記述用 syntax の述語

- ・昇格 is promoted
- ・降格 is denoted
- ・停止(消滅) terminated
- ・進展 proceeded
- ・収束 converged
- ・発散 diverged

変遷記述用 syntax の修飾句

- ・ in activation/inactivation
- ・ in rank [##]
- ・ in priority [##]
- ・ on [date] / at [time and date]

プロブレムリストの構造(方針)

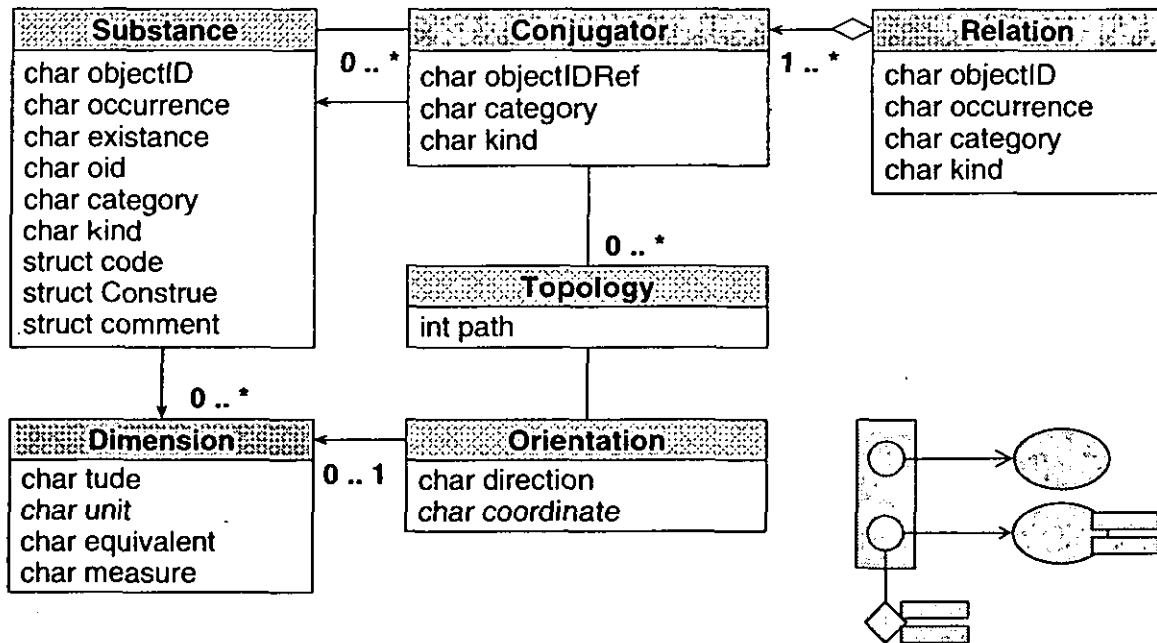
- ・ 0個以上のプロブレムを格納する
- ・ 異なる時刻のプロブレムリスト間の元は (形式的に) 重複しない(させない)
たとえ同一プロブレムであっても、異時刻においては形式的にはおのおの別 instance として扱う。

プロブレムの構造

- ・ 活性非活性状態を表わす属性を有する
- ・ 重み付けを表わす属性を有する
- ・ 優先順位を表わす属性を有する
- ・ プロブレムは階層性を有することができる

文献 [4]

オントロジーとダイアグラム



ただ、思考過程モデルもプロブレム変遷記述形式も、これまでは確たる方策が得られていなかった。

ONTOLOGY and MODEL

しかし著者は厚生科研(H12-医療-009)の研究過程でontologyに即したUML modelを構築しXML Schemaによる記述形式を開発した。

これは元々は歯科診療情報を合理的かつ網羅的に記述するために開発されたが、その応用可能範囲は下記の事由により、歯科領域に留まるものではない：

- ontology
- not-predefined receptacle
- representation capabilities of case/adjective
- separation domain-info from meta-structure

文献 [5, 6]

そこで新たな厚生科研(H15-医療-050)において、これを応用しつつ本主題を研究開発することとした。

DIAGRAMS

XML Schema document type definition は割愛する。なお relation diagram は上記のとおりである。

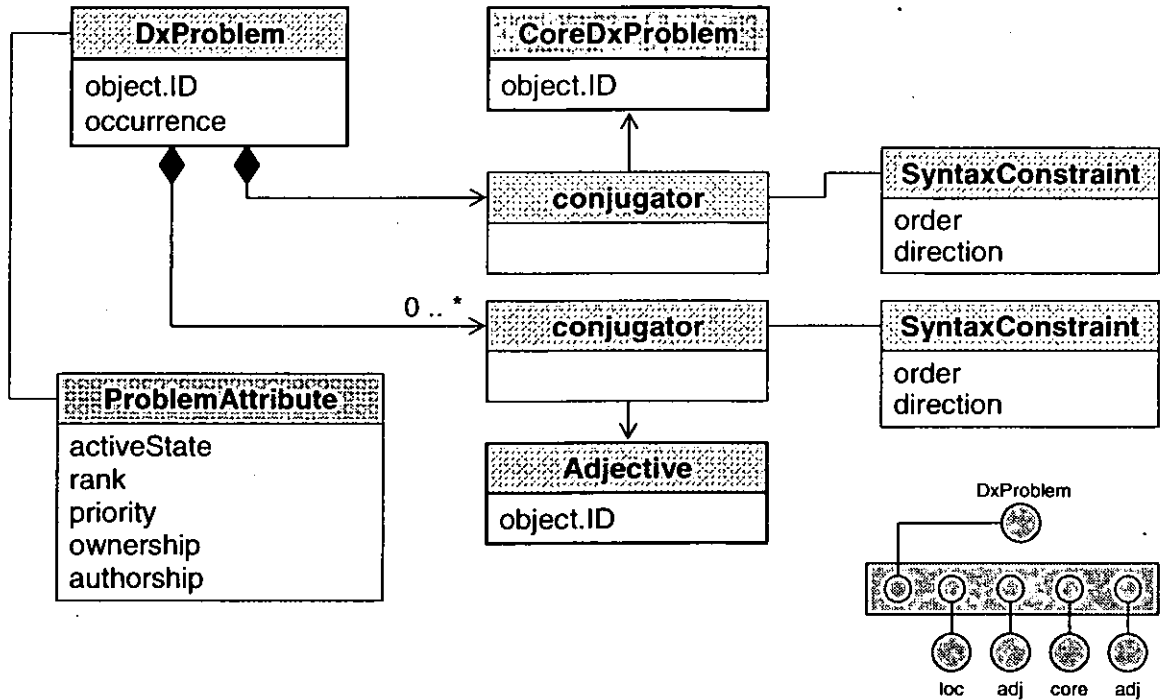
ここで、楕円は substance, 縦長の大きな長方形は relation, 小円は conjugator, また菱形は topology, 横長の小さな長方形は修飾クラスである dimension や orientation を表わしている。

なお本発表で対象外とした事項は以下の通り：

- AE や PLp との間の入出力
- 関与者(stakeholder)とシステム時刻の管理

文献 [7 - 12]

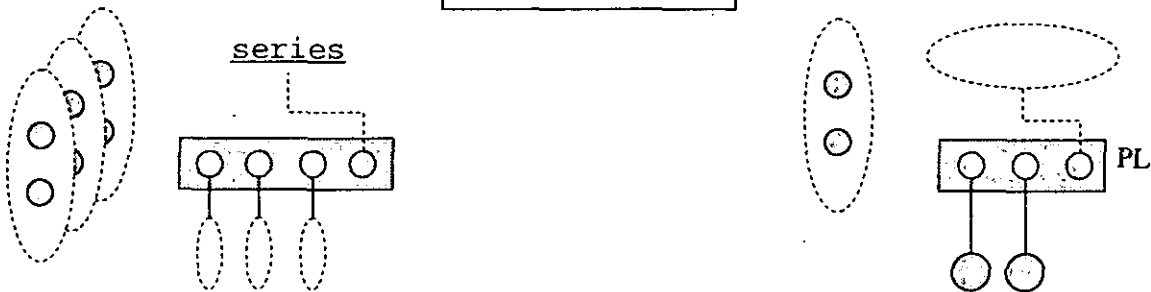
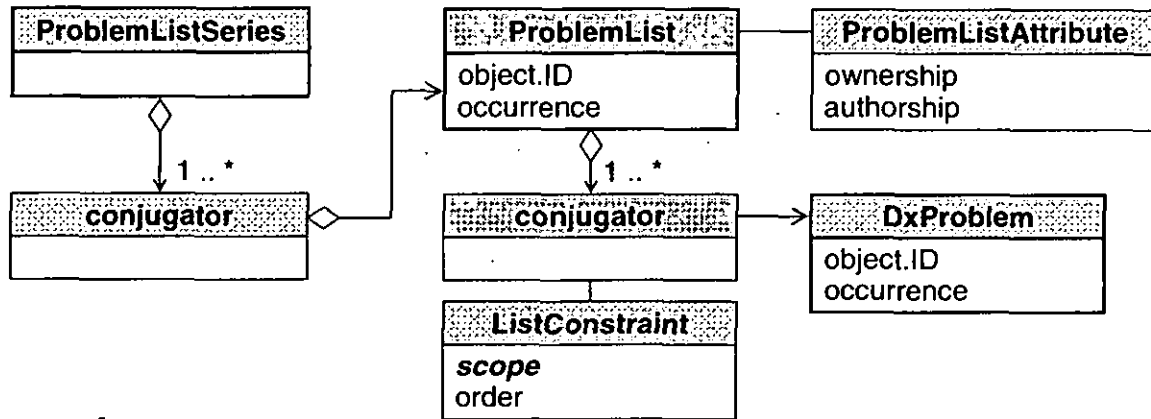
病名の組立て



```

1. <substance object.ID="DxPrblm.A" occurrence="根" substance.category="病名" substance.kind="構成体">
2.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="左尺骨陳旧性骨折疑い" code="???" />
3. </substance>
4. <substance object.ID="DxPrblm.A.loc" occurrence="葉" substance.category="解剖">
5.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="左尺腕骨" code="???" />
6. </substance>
7. <substance object.ID="DxPrblm.A.adj.pre" occurrence="葉" substance.category="病名" substance.kind="病名.修飾">
8.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="陳旧性" code="???" />
9. </substance>
10. <substance object.ID="DxPrblm.A.cor" occurrence="葉" substance.category="病名" substance.kind="病名.コア">
11.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="骨折" code="???" />
12. </substance>
13. <substance object.ID="DxPrblm.A.adj.pst" occurrence="葉" substance.category="病名" substance.kind="病名.修飾">
14.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="疑い" code="???" />
15. </substance>
16. <relation object.ID="DxPrblm.A.REL" occurrences="根" relation.category="木" relation.kind="構成">
17.   <relatedObject object.refID="DxPrblm.A" relatedObject.category="親" relatedObject.kind="構成" />
18.   <relatedObject object.refID="DxPrblm.A.loc" relatedObject.category="子" relatedObject.kind="要素" />
19.   <relatedObject object.refID="DxPrblm.A.adj.pre" relatedObject.category="子" relatedObject.kind="要素" />
20.   <relatedObject object.refID="DxPrblm.A.cor" relatedObject.category="子" relatedObject.kind="要素" />
21.   <relatedObject object.refID="DxPrblm.A.adj.pst" relatedObject.category="子" relatedObject.kind="要素" />
22. </relation>
    
```

プロブレムリストの組立て



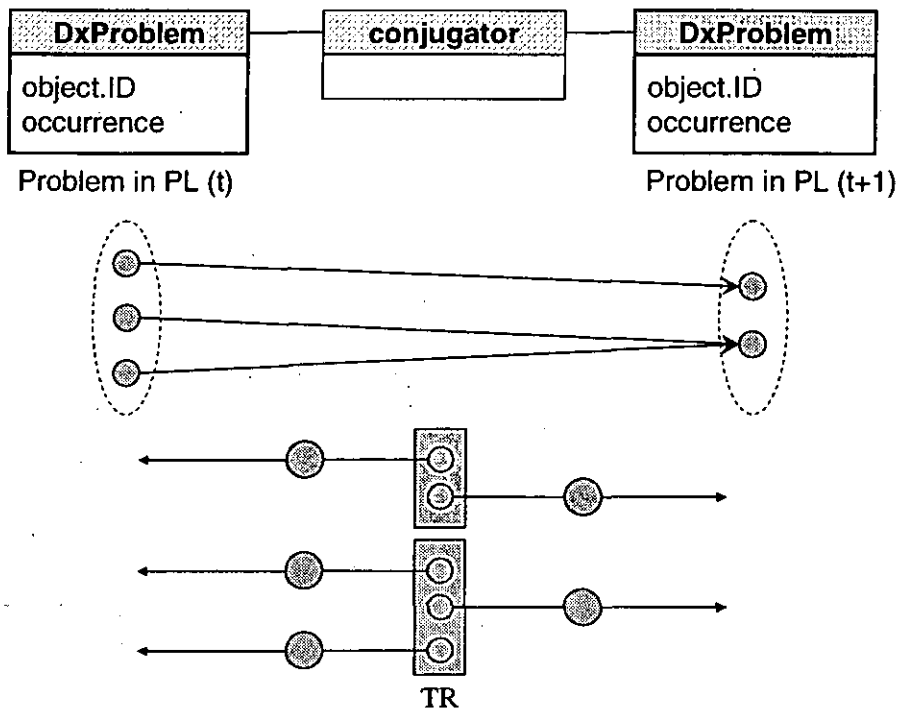
```

1. <substance object.ID="ProblemList.0" substance.category="診断" substance.kind="プロブレム">
2.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="ProblemList(T=t)"/>
3. </substance>
4. <substance object.ID="DxPrblm.0.A" substance.category="病名">
5.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="左尺骨陳旧性骨折疑い" code="???"/>
6. </substance>
7. <substance object.ID="DxPrblm.0.B" substance.category="所見">
8.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="肥満" code="???"/>
9. </substance>
10. <substance object.ID="DxPrblm.0.C" substance.category="所見">
11.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="口渇" code="???"/>
12. </substance>
13. <relation object.ID="ProblemList.0.REL" relation.category="木" relation.kind="構成">
14.   <relatedObject object.refID="ProblemList.0" relatedObject.category="親" relatedObject.kind="構成"/>
15.   <relatedObject object.refID="DxPrblm.0.A" relatedObject.category="子" relatedObject.kind="要素"/>
16.   <relatedObject object.refID="DxPrblm.0.B" relatedObject.category="子" relatedObject.kind="要素"/>
17.   <relatedObject object.refID="DxPrblm.0.C" relatedObject.category="子" relatedObject.kind="要素"/>
18. </relation>
19.

```

[codeSystem.Code および substace.:code については例示省略. 他の例も同様]

プロブレムの変遷



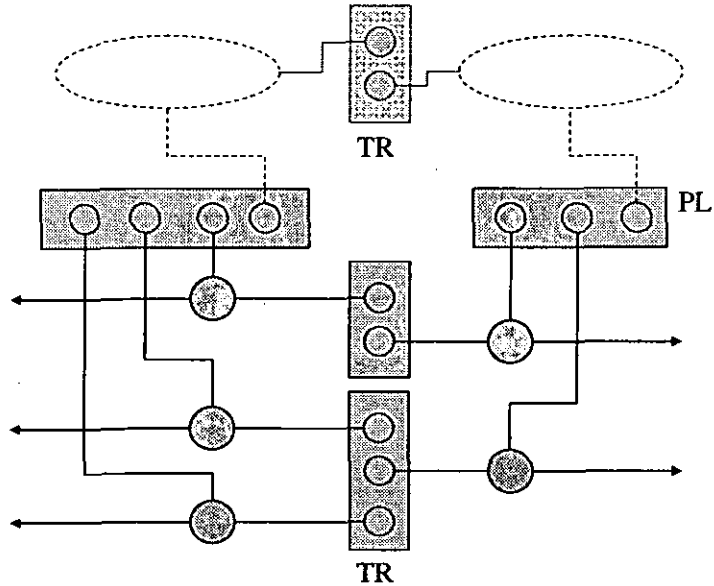
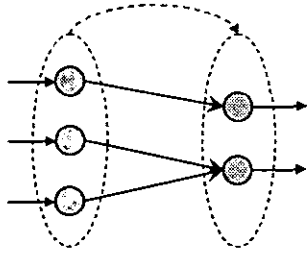
```

1. <substance object.ID="ProblemList.0" substance.category="診断" substance.kind="プロブレム">
2.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="ProblemList(T=t)"/>
3. </substance>
4. <substance object.ID="DxPrblm.0.B" substance.category="所見">
5.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="肥満" code="???" />
6. </substance>
7. <substance object.ID="DxPrblm.0.C" substance.category="所見">
8.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="口渇" code="???" />
9. </substance>
10. <substance object.ID="ProblemList.1" substance.category="診断" substance.kind="プロブレム">
11.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="ProblemList(T=t+1)"/>
12. </substance>
13. <substance object.ID="DxPrblm.1.D" substance.category="病名">
14.   <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="糖尿病疑い" code="???" />
15. </substance>
16. <relation object.ID="ProblemTransition.0.1.REL" relation.category="病" relation.kind="変遷">
17.   <relatedObject object.refID="DxPrblm.0.B" relatedObject.category="源" relatedObject.kind="遷移.合流"/>
18.   <relatedObject object.refID="DxPrblm.0.C" relatedObject.category="源" relatedObject.kind="遷移.合流"/>
19.   <relatedObject object.refID="DxPrblm.1.D" relatedObject.category="対" relatedObject.kind="遷移.合流"/>
20. </relation>
21.

```

【プロブレムリストの構成等は掲載割愛】

プロブレムリストとプロブレム



```

1. <substance object.ID="ProblemList.0" />
2. <substance object.ID="ProblemList.1" />
3. <substance object.ID="DxPrblm.0.A" />
4. <substance object.ID="DxPrblm.0.B" />
5. <substance object.ID="DxPrblm.0.C" />
6. <relation object.ID="ProblemList.0.REL" relation.category="木" relation.kind="構成">
7.   <relatedObject object.refID="ProblemList.0" relatedObject.category="親" relatedObject.kind="構成"/>
8.   <relatedObject object.refID="DxPrblm.0.A" relatedObject.category="子" relatedObject.kind="要素"/>
9.   <relatedObject object.refID="DxPrblm.0.B" relatedObject.category="子" relatedObject.kind="要素"/>
10.  <relatedObject object.refID="DxPrblm.0.C" relatedObject.category="子" relatedObject.kind="要素"/>
11. </relation>
12. <relation object.ID="ProblemList.1.REL" relation.category="木" relation.kind="構成">
13.   <relatedObject object.refID="ProblemList.1" relatedObject.category="親" relatedObject.kind="構成"/>
14.   <relatedObject object.refID="DxPrblm.1.A" relatedObject.category="子" relatedObject.kind="要素"/>
15.   <relatedObject object.refID="DxPrblm.1.D" relatedObject.category="子" relatedObject.kind="要素"/>
16. </relation>
17. <relation object.ID="ProblemTransition.0.1.REL" relation.category="網" relation.kind="変遷">
18.   <relatedObject object.refID="DxPrblm.0.B" relatedObject.category="源" relatedObject.kind="遷移.合流"/>
19.   <relatedObject object.refID="DxPrblm.0.C" relatedObject.category="源" relatedObject.kind="遷移.合流"/>
20.   <relatedObject object.refID="DxPrblm.1.D" relatedObject.category="対" relatedObject.kind="遷移.合流"/>
21. </relation>
22.

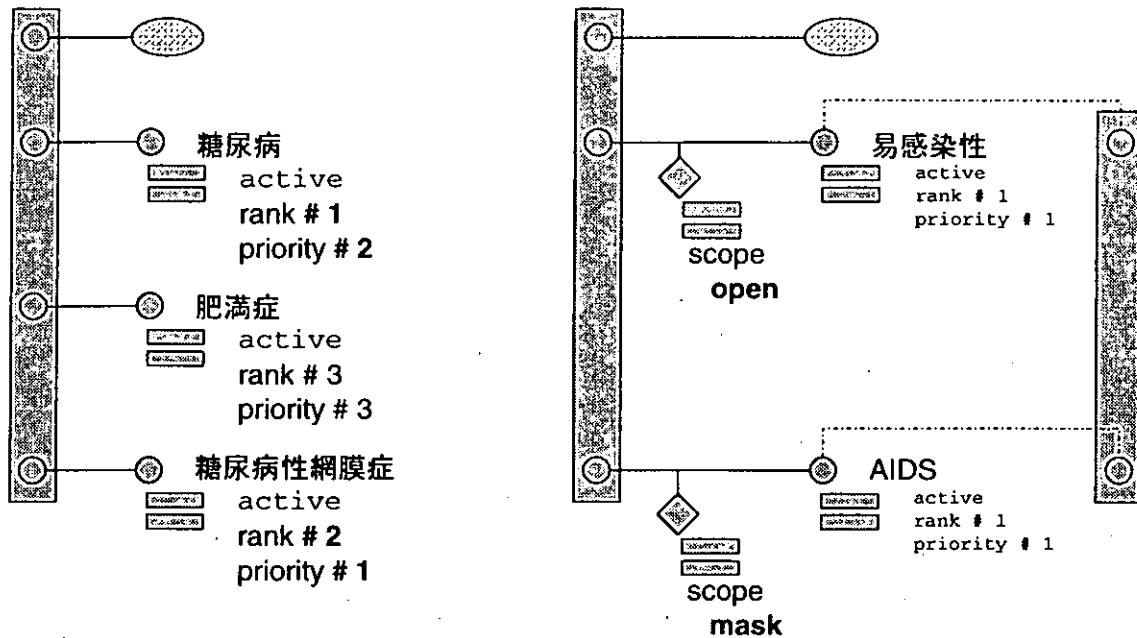
```

【紙幅制限のため substanceの正規の記述は省略】

<substance object.ID="DxPrblm.1.A" />
<substance object.ID="DxPrblm.1.D" />

【紙幅制限のため DxPrblm.?A の遷移については省略】

主病名とエイリアスの扱い



主病名について

```

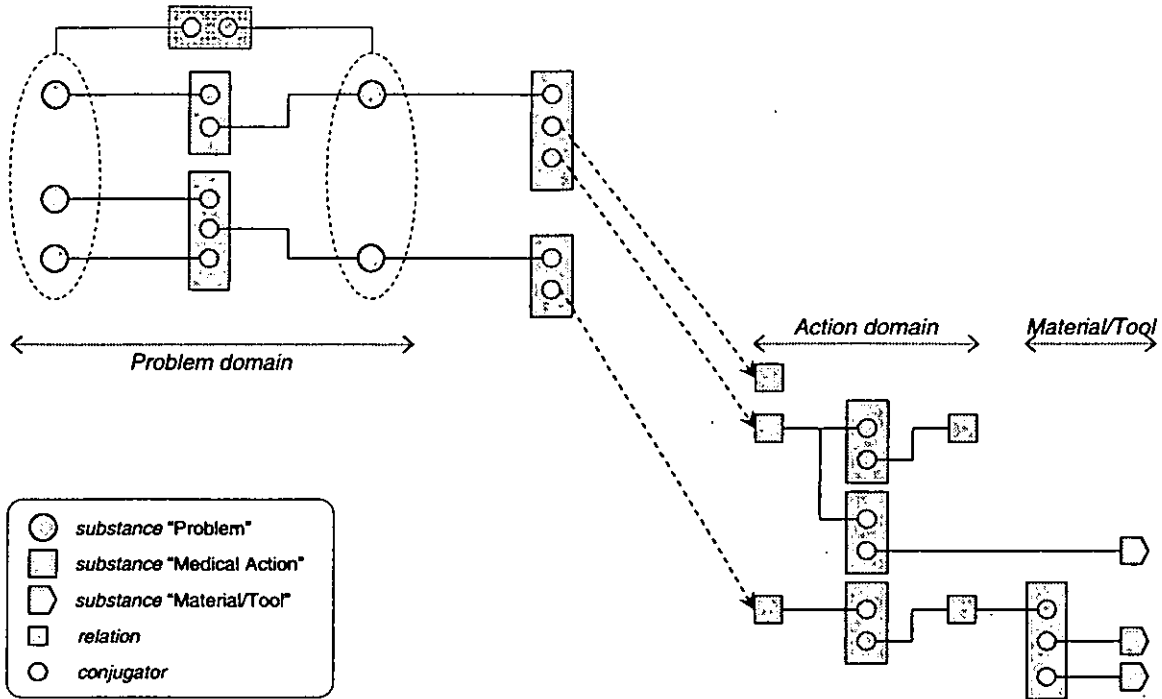
1. <substance object.ID="DxPrblm.3.K" substance.category="病名">
2. <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="糖尿病" code="???" />
3. <dimension tude="rank" unit="anonymous" equivalent="等級" measure="1" />
4. <dimension tude="priority" unit="anonymous" equivalent="等級" measure="2" />
5. </substance>
6. <substance object.ID="DxPrblm.3.L" substance.category="病名">
7. <substance.code codeSystem.Code="xxx" codeSystem.Version="" code.Name="糖尿病性網膜症" code="???" />
8. <dimension tude="rank" unit="anonymous" equivalent="等級" measure="2" />
9. <dimension tude="priority" unit="anonymous" equivalent="等級" measure="1" />
10. </substance>
    
```

エイリアスについて

```

1. <substance object.ID="DxPrblm.x.A" /> <!-- 易感染性 --> 【紙幅制限のため substanceの正規の記述は省略】
2. <substance object.ID="DxPrblm.x.B" /> <!-- AIDS -->
3.
4. 【紙幅制限のため relationの正規の記述は省略】
5. 【紙幅制限のため 「真の」 relationの記述は省略】
6. <relatedObject object.refID="DxPrblm.0.A" relatedObject.category="子" relatedObject.kind="要素" >
7. <topology> <orientation direction="open" coordinate="scope" /> </topology>
8. </relatedObject>
9. <relatedObject object.refID="DxPrblm.0.B" relatedObject.category="子" relatedObject.kind="要素" >
10. <topology> <orientation direction="mask" coordinate="scope" /> </topology>
    
```

メディカルアクションとの関連付けへ



メディカルアクションへ

左記までで病名とプロブレム、そしてプロブレムリストと、それらの変遷が扱えることが判ったので次は診療行為との連関の表現である。その概要は上図に示す。

なおこの実装例については、矢嶋が本学会大会にて発表しているので、そちらを参照願いたい (H-1-6)。

現時点での未完事項としては、関与者と時刻の扱いである、としている。実は記述可能ではあるが、モデルとしては未完である、という立場でいる。

関与者の扱いについては一般的な役割 (role) のみに留まらないモデルが必要になると思われるが、これは口頭発表 (3-A-2-3) を参照されたい。 文献 [13, 14]

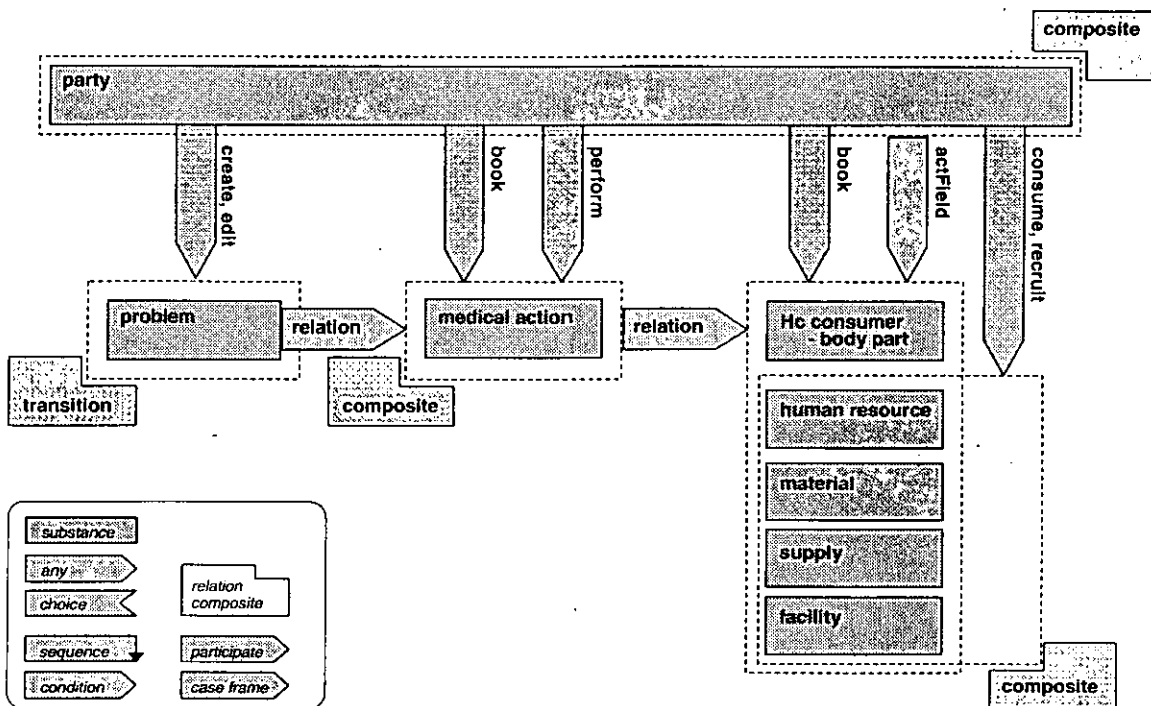
制約の記述の必要性

さて病名の組立てにしても、上記の連なりにしても、substance に附された category や kind, すなわち、各々が属する概念領域や分類等によって結合性、つまり、relation と conjugator による連関の可否が定められうる。

厚生科研 (H12-医療-009) では、単に事実を記録する迄に scope を限定していたので制約の記述は不要であったが、今後は (1) 知識記述のためにも、また (ii) 機械処理のためにも、必要に思われる。

なお上図では PL→PL, PL→MA, MA→MA そして MA→material は許される、などという形式的記述になろう。

制約について



制約の種類

必要な制約には、以下のような事項が考えられる:

- ・ 親和性(連結性)と透過性
- ・ 存否と濃度
- ・ 順序と排他と依存
- ・ 限界値/許容値 または その範囲
- ・ 格フレーム
- ・ 参照可能性と操作可能性

ただ本発表で提示した枠組ではこれら全てを完全に表現することは困難に思われるので、今後若干年モデルを拡張することとなる。

制約の表現

制約記述の方策については、本発表の主題を大きく逸脱してしまうことになるので、ここでは上図のみを挙げておくに留めることとする。

これは、substance 間の連結において適用されるべき制約に関わる表現の一例であり、HL7 v3 CDA 制約記述形式を模倣しつつ、これを若干拡張して記した diagram である。

ただ diagram は視覚的で理解しやすいものの機械処理には役立たないゆえ、左記の制約事項を記述すべき XML Schema の枠組みを考案・整理中である。

文献 [15]