

としての,

- ・ 関連する定性的な評価項目 (セット)
- ・ 関連する定量的な評価項目 (セット)

である。日常の臨床現場では、より具体的な Secondary EP あるいは Surrogate をにらみながら診療に当たることが多いであろう。

#### Surrogate

- ・ Primary EP の予測因子となりうる (と考えられる) 定性的 and/or 定量的な評価項目
- ・ Secondary EP として用いることも可能とされる

#### C. 5. 3. 4 帰還と時間と焦点と

このように見てくると意図実現過程においては、思考世界なる意図と現実世界での事象事態とが相互作用することに伴って意図実現過程における計画あるいは意図が修正変更されるのであるから、Thread と pGA/dGA/Surrogate/EP とが深く関わり合うことは、むしろ当然である。そのような過程的な時間、その序列系列における計画単位において feedback が掛かる、すなわち帰還なる再帰的関与が為されて計画単位が評価され、必要に応じて計画単位は修正変更されるからである。

最終的には計画単位が修正変更されるものの、その動力源たる意図あるいは目標が第一標的として扱われる必要がある。というのも意図の作用していない計画は、すでに計画ではなくなっ てしまい、単なる闇雲な行為に墮するからである。ゆえに Thread は、AP ではなくて pGA/dGA/Surrogate/EP と相応しながら制御されねばならなかったのである。

ところで GA とは、そもそも intervention を要する・幾つかの primitive な PRi を内包した PR に対する GA/GAi であり、然るが故に PL/

PRi に応じて GA/GAi が生起せしめられたのであった。それでは PL/PRi とはそもそも何であったのであろうか? 目標 (GA) に対比するモノとして応えらば、それは一個の人間という総体においての、或る「重要」な病態概念 (を指し示す語)、もしくはその存在を示唆する兆候または指標を指し示す語) であった。

いずれも当の総体の将来のありかたを左右する事態と「思われる」ため、つまり予後に関する高い蓋然性を「認定」するからこそ、dGA や pGA を発生させる始原的な動力となっていた。ここ、この思考過程において「重要」性が「認定」されていた。この「判断」は <C.5.3.1 要約> に記したように、事態の「認識」と「信念」とに拠って初めて「成立」している。したがって (a) 当の総体において在ると「認識」されえた事態のうち、どのような事態を重要と「認定」するのか、そして (b) 事態が輻輳している際にはいずれの事態に焦点や重点を置く、つまり優先性を「認定」し「判断」するのかは、引き続いて生じる処理ヒエラルキーすなわち計画とその実施に深く影響することとなる。というよりむしろ、計画と計画された行為の実施に対しての支配性を有している。

繰り返すがそのような支配性が成立する条件とは、(i) 事態の「認識」を成立させるために要する「信念 (知識)」, (ii) 事態の「認識」, (iii) 事態を「認識」したという「信念」, (iii) 事態の重要性を「認定」するための「信念」, (iv) 事態の重要性の「認定」, (v) 事態の重要性を「認定」したという「信念」, (vi) 個々の事態の間に優先性を「認定」するための「信念」, (vii) 事態の優先性の「認定」, (viii) 事態の優先性を「認定」したという「信念」, さらには、(ix) 引き続いて生じる処理においても優先性を「認定」した当の事態に対する処理を優先して

よいと「認定」するための「信念」, などに拠っている。

この複雑な機構によって成立している問題と目標との関係, そして引き続く行為の計画とその実施, すなわち Thread の成立は, それでも仮定的で試行的な側面を免れえず, 時間あるいは順序性という基底によって支えられた行為の列という過程性を免れえない。別の言い方をすればメタなる順序性に全体が支配されている。

したがって意図実現過程が妥当性を保持し続けるためには, 結果として正であれ負であれ, 帰還なる再帰的関与という事態については, 確然とせざるをえないのである。と同時に, 近位目標 (pGA) と遠位目標 (dGA) との弁別は必ずしも明瞭ではないけれどもそれが現実であり, そしてその弁別自体に上述した以外の・实际的で特別な意義を認めることは困難である。そして, 情報の扱いのために情報の扱いがあるのではなく, 現実を扱うために情報の扱いがある。

### C. 5. 3. 5 Thread 遷移の判別

Thread の遷移と消長を捉えること, それを表現したり解釈したりする処理と, そのような処理環境を与える表現を用意しておくことは, 診療経過という文脈 context を理解するうえで重要である。意図実現過程なる「臨床思考過程と診療経過」における Thread の変遷と消長は, まさに <C.2.3.2 意味的構造> に記した誤謬と失敗に関する追跡点 (tracing point; TP) に相当するからである。

意図実現過程についても会話構造における TP と同様に, 「失敗」すなわち「期待や予測 (GA/EP) と現実 (AE) との比較における相違または合致」のみならず, そのような期待や予測 (GA/EP) とそれに基づく計画 (AP) と行為 (MA) を生み出すに至った様々な情報組織体へのポイン

タが確保されている必要がある。そして先取りしていうならば, 本研究ではそれを提供しており <C.5.4 参照実装モデルの定式化>, また表現枠組 CSX がその実現を下支えしている。

さて Thread は pGA/dGA/Surrogate/EP と絡みながら遷移し消長することから, <C.5.3.2 Thread 消長の概要> のみを見ると, Thread 管理を (半) 自動的に処理することも, あながち困難ではないように見えるかもしれない。しかしその予測は optimistic であろう。というのも <C.5.3.4 帰還と時間と焦点と> に敢えて執拗に列挙したように, 様々な「認識」や「認定」と, それらを下支えしている「信念」に拠っているからである。このような「知の運用」は通常は暗黙的であり, また経験的でもある。これらは情報工学においては扱いづらい対象ではある。とはいえ医療は信託を伴う準委任契約とされており結果が強く期待される貢献であるが, しかし結果だけで全てが評価されるわけではない。社会的な信頼はもとより法廷係争においても, 実のところ「医療のプロセス」すなわち臨床思考過程と診療経過における「<その時点>での<可能性と妥当性>」こそ, 文脈と結果の全体に対する総体的な評価の拠り所とされている。この事情はまさに <C.2 意図と経験知の表出化> で見てきた。そして暗黙知や経験知の表出化方策を模索することも本研究主題の一つとしていた。

したがって Thread 遷移に関わる事項をより掘り下げて, Thread 遷移の判別可能性に関する若干を挙げておく:

- ・ PL, AE, pGA/dGA/Surrogate/EP が関わることは既に前述した通りである。なお <C.5.3.4 帰還と時間と焦点と> にも記している通り PL においては個々の PRi の priority ( の変化 ) にも関わっている。

- ・ PL/PRI, AE/AEi, pGA/dGA/Surrogate/EP とそれらの各要素 pGAI/dGAI/Surrogate(i)/EPI については, その生滅や変化の枚挙処理自体は単純で容易である. ただし, それらのうち幾つかが多少変化したのみで Thread が遷移消長するわけではない. むしろ,
- ・ PL,AE,GA あるいは AP の変化の「大きさ」が Thread の遷移消長を決定づける. これを判別するには少なくとも二つの手法が考えられる: 一つは, (A) 各要素事項の抽象的な「距離」あるいは「位相性」が信念(知識)として記述されており, それを参照できること. そのような世界知識は多軸であって, 語が指し示す概念間の「距離」のみならず, 行為の負荷やリスク, 行為の実現に投入すべき様々な資源ほか, 社会的な要素なども含まれることが期待される. 今一つは, (B) 事由 AE/rGA/rAP の各要素 AEi/rGAI/rAPI の生滅や意味的な変化を検知することであろう. この枚挙も, これ枚挙処理自体としては単純で容易である. ただし,
- ・ 事由 AE/rGA/rAP の生滅や変化を精確に検知するためには, 事由候補 rAEP/rGAP/rAPP の各要素の生滅や変化も全て検知する必要が生じる. というのも事由候補 rAEP/rGAP/rAPP を取捨選択するという思考行為のなかに, たとえば相反する事由のうち的一方のみを事由 AE/rGA/rAP としたのが立ち現れることがあり, 結果として, その時点において「重きにおいて焦点したその事由」の一端が顕れることがあるからである. ただし, これらを検知できるためには相応する環境を要する.
- ・ このような事情の変化は, 「節にも辺にも彩色されたグラフの差異」として顕れることが期待される. と同時に, あるいは構造自体が変化して Thread 遷移前後では構造間の同型性が失われるかもしれない. そのような記録と検知とを可能とする環境である.

種々の立場が担うコストの適切性そのほか勘案

すべき事項は多々あるが, いちおう可能性として記した. なお「彩色」の判別にも (A) のような世界知識のうち少なくとも subset が必要となることは無論である. しかしそれでも (A) と (B) とには明らかに差異がある.

純然たる (A) の手法においては以下の限界もある: (i) 語(の指し示す概念)間の距離は多変量空間における距離としての扱いに還元されており, ときには空間変換さえ為されているために, 語や概念の差異は検知しえても, 認知可能な意味の差としては帰還させられないことがある. と同時に, (ii) 各変量を担う軸性の重みは, 判断の目的や判断する状況において変容するにも関わらず, それらへの追従は通常は困難であるし, また (iii) 階を上位にとって「大まか」な比較をするということにも向いていない. 加えて, (iv) 語は文脈構造のなかに埋め込まれている語なのであり, とくに意図実現過程においては来歴性を無視しえないところであるが, 文脈非依存の, 意味差異というよりは抽象距離のみを検出するものであることから, そのような差が文脈構造のなかににおいて如何ほどの意義を持ちうるのかは俄には判断できない.

一方 (B) については, 部分グラフの同型性を検知すること, その彩色性を加味した「相似」を検知することに本質がある. 前者については意図実現過程なる文脈という「地」における部分グラフであることにも留意するならば, むしろグラフとして想定せずに多面体として想定することもできよう. いずれと想定するかは計算量と計算時間とに眼目をおいて決すればよいことである. 後者には, 少なくとも (A) と比較すれば, 必要な変量は少なく済む可能性が高い. というのも将に当の語が埋め込まれている (embedded) グラフ構造に多くの意味が含まれているからである. そして処理の対象は「構造の意味」では

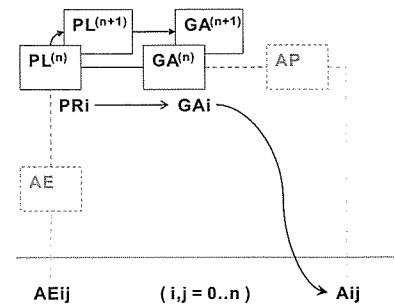
なくて「構造」を対象とすることが出来ているからである。

そのうえ「判断」や「決断」に重要なのは、単一の語の意味の差異ではなくて、語と語との結合が如何ようであるか、すなわち事由を述べ立てる言明であった。

### C. 5. 4 参照モデルの定式化

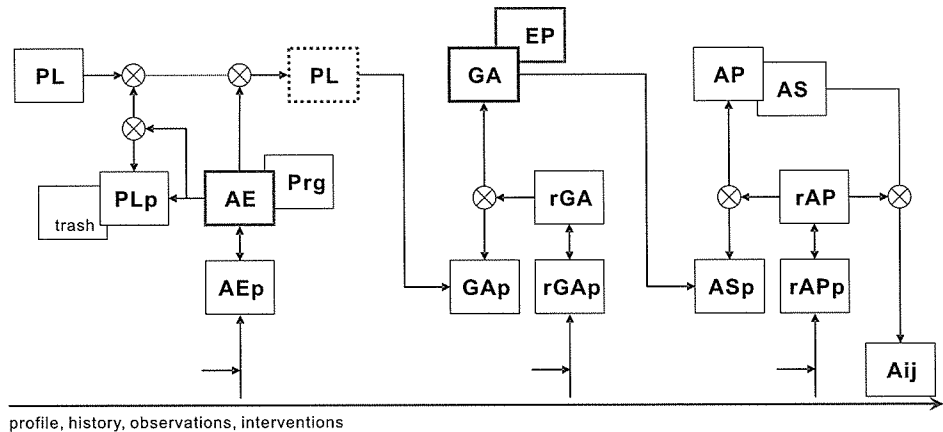
#### C. 5. 4. 1 構造の俯瞰

主任研究者は前研究プロジェクトまでに、思考過程の概念モデルにおける PL/PRI と Aij との短絡を参照モデルとして定式化したうえで試作実装を完了している <B.1.5 概念モデルの短絡化>。本研究においては GA と AP、そして rGA と rAP の一部をも含んだ構造を参照モデルとして定式化することになる。また GA には EP も含める。



なお前プロジェクトでは、その研究主題等の事由から AE/GA/AP などを PL の一部すなわち問題空間のほうに組み込んでしまっていたが、今回これらを正しく分離することとなる。ただし DxProblem に対して来歴性を伴って修飾的に作用する要素として設けてあった Intervention と Stage については、DxProblem の一部として組み込んだままとする。

臨床思考過程モデルと診療経過モデルとを融合した意図実現過程モデルは複雑な構造となる。よって先ず概略的あるいは概念的な図式を概観しておく。流れの説明と略号については <C.1.2 病名変遷と



profile, history, observations, interventions 病名診療行為連関 > や <C.5.2 概念モデル> を参照願いたい。図中左上の二つの PL は PL の変遷を表現している。GA は pGA/dGA/EP を含む。PL /GA/AP は当該 Cycle における決断あるいは結論とも云いうる。これらに関する候補列挙と取捨選択の事由をも挙げうるモデルとして設計している。ただし上図には重大な省略が為されている。というのも、まさに事由の言明を担うべき関連を表明していないからである。

実際のところ意図実現過程モデルは複雑である。構造が複雑となる原因は、深層認知における <C.2.3.1 形式的構造> に求めることができよう。すなわち或る局面においてはほぼ階層的に表現する意図構造と、現実の過程的な流れにおいて様々なモノやコトが相互に因果関係を結ぶ網目状構造の、二つの異なる構造が錯綜して一つの構造とされるからである。したがってこのような構造的複雑さは、元来、不可避なのである。

とはいえそれら全ての「関係の線」を、たとえば上図のなかに描き込むことは不可能ではないにせよ極めて判読性を低下させるという意味においては不適切であろう。しかし意図実現過程の定式化とはそれに近いことを為していくことを意味している。もっとも整理のための創意や手法もないわけではない。

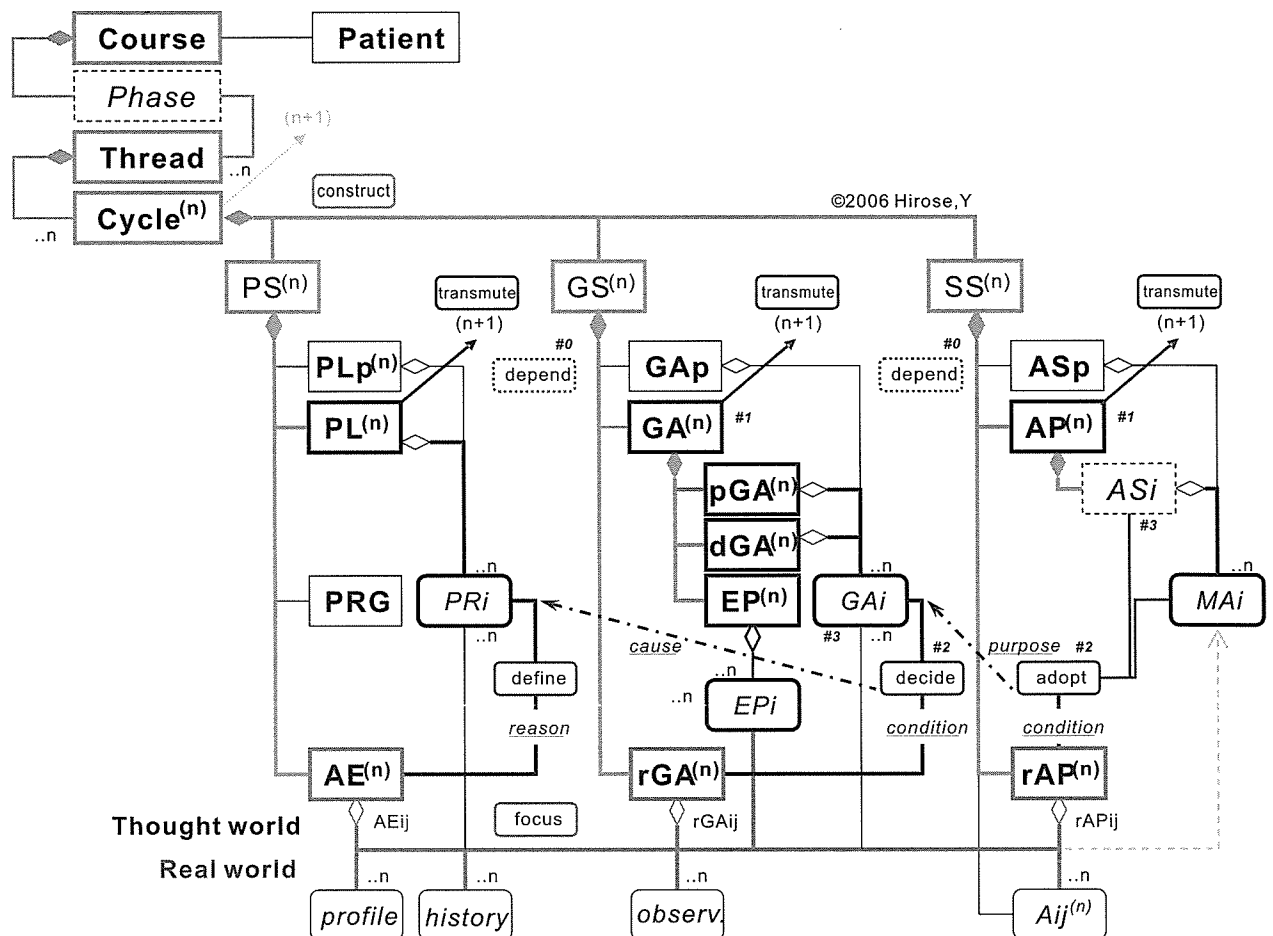
### C. 5. 4. 2 関連

図を順次に説明していく。まず Course から PS/GS/SS を経て、それぞれ PL<sub>p</sub>/PL/PR/AE, GAp/GA/pGA/dGA/EP/rGA, ASp/AP/AS/rAP に至る経路と要素とそれらの構造は、そういう構造を構成する arcScope によって結合され形成される。この構造は先の図と相応している。過程構造であるから時の流れに応じて、というよりも診療経過に応じて、この構造自体が重ねられていくが、定型的であって比較的には扱いやすい構造だろう。

一方, profile/history/obserc./Aij と AE/rGA/rAP とを結ぶ辺と要素とそれらの構造は、現在ならびに過去の ( 広義の ) 現実世界に横たわっている ( と認識された ) 事象や事態を只今の思考世界へと取り込む関係であり、そういう構造を構成する arcScope によって結合され形成される。

AE と PL<sub>p</sub> と PR と PL, rGA と GAp と GA と pGA/dGA, rAP と MA と ASp と AP を、それぞれに結ぶ辺と要素とそれらの構造は、なんらかの意味での決断とその根拠を表す関係であり、そのような構造を構成する arcScope によって結合され形成される。ただし下図は全ての判断のうち一部しか表明していない。

灰地の長方形は container infoNode, 灰地の角取り長方形は Cell infoNode, 白地の小さな角取り長方形は arcScope, 下線文字は arcScope のもとに結ばれる意味役割で、その表現は infoArc が担う。



#### C.5.4.2.1 目的的関連とその扱い

##### ◆ 図中 #0

GA subspace や AP subspace は各々先行する subspace に依拠しながら決定されるが、先行 subspace から直接的に影響される container とその内容は GAp と ASp のみ、つまり候補のみである。最終的な GA や AP とその内容は、他の様々な条件や経緯などが勘案されたのちに決断される。

##### ◆ 図中 #1

GA や AP は各々後続する Cycle における当該 container に影響するものの、その内容つまり GAI や ASI (MAI) は、必ずしも PRi (n) と PRi (n+1) とのような関係を形作るとは限らない。GA や AP は同一の Course において、先行する同様の PL からできても異なった内容となることもありうる。したがって、GAI (n) ~ GAI (n+1) 関係や ASI (n) ~ ASI (n+1) 関係は、PL のような変遷関係になるとは限らないゆえに、そのような関係は記述しない。

##### ◆ 図中 #2

GAI あるいは ASI (MAI) は、各々当該 Cycle において先行する複数の PRi や GAI に依存することがある。GAI はときに複数の PRi と対応し、API はときに複数の GAI と対応することがある。よって decide あるいは adopt なる arcScope においては、そのような Cell をも参照する必要がある。

##### ◆ 図中 #3

AEi, GAI, ASI あるいは MAI の内容は、PRi のようにコード化することが困難な場合も多いと考えられる、言い換えれば言まや文章で表現されることがある。そのような内容を格納する候補としては要素 construe または要素 quale であるが、前者を選択することとする。

#### C.5.4.2.2 因果的関連とその扱い

意図実現過程においては、目的的関連と構造の以外にも、因果的関連と構造が想定される。

##### ◆ 信念など

まず PLp/GAp/ASp は多くの場合には医学的知識なる「信念」から引きだされうる、すなわち既に表出されている知を活用することが可能である。言い換えれば知識ベースを用意しておいて半自動的に援用することも可能である。とはいえ経験知などの暗黙知が候補されることも否めない。このような場合、それを参照引用することになる。

##### ◆ 事態など

次に AE/GAp/APp は、医学的な所見等のみならず、より広汎な範囲に、事由とされるべき事象や事態を求める。実際のところ古今東西を問わず医療が社会的な枠組の内にしかありえないことにも想いを馳せると、医学知識のみという埒を越えた様々な事物事象事態が、とくに GA や AP を選択する決断の事由 (の候補) もしくは「勘案すべきこと」として挙げられることがある。しかしそのような全てを rGA や rAP に挙げて整理とはほど遠く見通しが悪くなるばかりでなく、また「事由として採択した」という事実でもない。その一方では、背反するような事由候補が同時に存在することも頻繁である。それらの事由候補は、GA や AP を決断するための候補であるのみであって未だ事由そのものではない。

## ◆ AEp/rGAp/rAPp

これらの扱いに関する形式的な解として、事由等の候補を容れる container として AEp/rGAp/rAPp を用意した。この container は <C.5.4.2 > 図には描出していないが <C.5.4.1> には明示している。AEp/rGAp/rAPp は全ての事由候補を参照し、そして自身のうちに蒐集することを役割とする。

すなわち、システムを操作する者が AEp/rGAp/rAPp の内いずれかの事項を事由として焦点し (focus), そして AE/GAp/APp のうちに採り上げる (take) ことを反映する枠組とする。このような記述枠組を想定すると PL そして GA や AP の採否に関する「根拠の根拠」を解釈したり、決断の際に勘案した「視野範囲」を検証したりすることができるようになる。

## C.5.4.3 格納

主要な項についての格納関係を記す。

## ◆ Course

Course は複数の Thread を、Thread は複数の Cycle を格納する。Cycle は ProblemSpace, GoalSpace, ActionPlanSpace を 0..1 個ほど格納する。Problem Space は少なくとも PL (PRi) と AE (AEi) を格納する。Goal Space は GA (GAi) と rGA (rGAij) を格納する。GA は pGA/dGA/EP に細分化されることがある。ActionPlanSpace は AP (APi) と rAP (rAPij) とを格納する。

なお () 内は container に格納される Cell infoNode であって (sub)space とは直接には結びつかない。これらの Cell は来歴のなかに埋め込まれて (embedded されて) いるので infoArc による参照 infoNode となる。Phase ならびに ProblemSpace/GoalSpace/ActionPlanSpace は割愛してよい。

## ◆ PL

Container infoNode である PL は要素 PRi を格納する。要素 PRi は各々要素 AEi と関係づけられながら PL に格納される。

AE の要素 AEi は、対応する PRi が <define> されるときにその <reason> を与える。AE は PRi を <define> するための reason となるような、過去現在未来の文脈に embed された infoNode を <focus> し、AEij として格納する (参照として)。その対象は Cell の場合も container の場合もある。後者の場合には謂わば chunk された情報組織体を参照することになる。

## ◆ GA

Container infoNode である GA は要素 GAi を格納する。要素 GAi の値実体は free text の場合がある。その free text は infoNode/construe に格納する。

要素 GAi は各々要素 rGAi と関係づけられながら GA に格納される。このとき、対応する (1個以上の) PRi を <cause> として参照する。GA は PRi の解決の方向性を与える。rGA の要素 rGAi は、対応する GAi の <decide> において <condition> を与える。rGA は GAi を <decide> するための condition となるような、過去現在未来の文脈に embed された infoNode を <focus> し、rGAij として格納する (参照として)。その対象は Cell の場合も container の場合もある。後者の場合には謂わば chunk された情報組織体を参照することになる。

## ◆ AP



Container infoNode である AP は APi すなわち要素 ASi または要素 MAi を格納する。要素 ASi は container でもあり、要素 ASi または要素 MAi を格納する。要素 ASi または要素 MAi の値実体は free text の場合がある。その free text は infoNode/construe に格納する。

要素 ASi または要素 MAi は各々、要素 rAPi とも関係づけられながら AP に格納される。このとき、対応する (1個以上の) GAi を <cause> として参照する。AP は GAi を実現するための具体的な方策を与える。rAP の要素 rAPi は、対応する ASi または MAi の <adopt> において <condition> を与える。rAP は ASi または MAi を <adopt> するための <condition> となるような、過去現在未来の文脈に embed されたセル infoNode を <focus> し、rAPi として (参照として)。その対象は Cell の場合も container の場合もある。後者の場合には謂わば chunk された情報組織体を参照することになる。

#### ◆ GA ならびに AP の遷移

GA(n) は GA(n+1) に、AP(n) は AP(n+1) に、各々関連付けられる。これらも遷移関係である。

ただし、GAi あるいは APi レベルでは変遷関係を記述しない。その事由は <C.5.4.2.1 目的的関連とその扱い> の 図中 #1 の通りである。

#### C.5.4.4 採用する category/family

本研究を遂行するにあたって、CSX において category/family に必要となる値を挙げる。なお下記の列挙に完璧性を主張するものではなく、主に当面の実装に要する事項を枚挙したものである。加えて、試作実装で用いる予定の uid テンプレートも附記しておく。

#### ◆ infoNode

· uid	ORI(.TRD.##.CYC.##).[blockName]	
· category	ClinicalCourse	Course
	Problem (subSpace)	Problem
	Goal (subSpace)	Goal
	Solution (subSpace)	Solution
	Document	(2号様式で)
· family	ClinicalCourse	Course
	Phase	Phase
	ThreadOfClinicalCourse	Thread
	CycleOfClinicalCourse	Cycle
	ProblemList	PL
	Problem (Item)	PR or PRI
	AssessmentAndEvaluation	AE
	AssessmentAndEvaluation (Item)	AEi
	elements of AEi	AEij
	Goal (List)	GA
	Goal (Item)	GAi
	Goal.proximal	pGA
	Goal.proximal (Item)	pGAi
	Goal.distal	dGA
	Goal.distal (Item)	dGAi
	EndPoint	EP
	EndPoint (Item)	EPi

ReasonOfGoal	rGA	
ReasonOfGoal (Item)	rGAi	
elements of rGAi	rGAij	
ActionPlan (List)	AP	
ActionSet	AS	or
ReasonOfActionPlan	rAP	
ReasonOfActionPlan (Item)	rAPi	
elements of rAPi	rAPij	
Action or Cure or Intervention	MA	
Action or Cure or Intervention (Item)	MAi	
Session	( 2号様式で )	
	( 仔細は nodeCode@code で )	
Operation	( 2号様式の処置手術で )	
Order	( 2号様式のオーダーで )	
Order.Entry	( 2号様式のオーダーで )	
Order.Item	( 2号様式のオーダーで )	
Prescription	( 況2号様式の処方で )	
Instruction	( 2号様式の処方で )	

◆ description/perspective

· uid	ORI(.TRD.##(.CYC.##)).DS	
· uid	ORI(.TRD.##.CYC.##.[blockName]).DS	
· category	Document	
	Definition	
	Knowledge	
	Space	
	Process/Context/Transaction	
	ThinkingProcess (Decision)	
· family	Chart.Form_1.jp	診療録 . 一号様式
	Chart.Form_2.jp	診療録 . 二号様式
	ClinicalCourse	Course.Clinical
	ClinicalCourseStructure	Course.Structur.Clinical
	Problem	( 必須ではないが視野の限定には便利 )
	Goal	( 必須ではないが視野の限定には便利 )
	Solution	( 必須ではないが視野の限定には便利 )

◆ arcScope

· uid	RL.ORI(.TRD.##.CYC.##.[blockName])
· category	Construct
	Define
	Decide
	Adopt
	Focus
	Review
	Transmute
	Group
	UnGroup

	Insert	
	Release	
	Set	
	Bearing	
	Cure	(2号様式の HD と BD で)
	Examine	
	Compare	
	Vary	
	Order	
	Prescribe	
	Cure ..	
・ family	Container	
◆ infoArc		
・ category	Nominative	
	Genitive	
	Object	(2号様式の HD と BD で)
	Source	(プロブレム変遷で)
	Goal	(プロブレム変遷で)
	Reason	(2号様式の HD と BD で)
	Theme	(2号様式の HD と BD で)
	Condition	
	Location	
・ family	Proceeded	(プロブレム変遷で)
	Converged	
	Diverged	
	Promoted	
	Denoted	
	Terminated	
	Whole	
	Portion	
	Purpose	
	Reason	(2号様式の HD と BD で)
	Cause	
	Condition	
	Construct	
	Element	
	Session	
	Preface	
	Block	
	Directory	

## C. 5. 4. 5 直列化の概形

### C. 5. 4. 5. 1 試作実装における uid の扱い

前研究プロジェクトまでにおいて試作実装時の目視確認などの目的と開発過程の経緯事情などから、当面のこととして、CSX 要素の uid には階層状の附番規則によって附番していた。本研究でも、この手法を踏襲するので、前研究プロジェクトで用いた規則の概要とサンプルを掲げておく。

RL: Relation or Scope

PPT: Participant

C1F: ChartFirstForm

CTX: Context

PL: ProblemList

AS: Assessment

GL: Goal

PR: Dignosis or Problem

STG: Stage

ITV: Intervention

TRS: Problem Transition

D: Directory

C2F: ChartSecondForm

P: Preface

B: Block

B: Block

HD: BlockHeader

BD: Block Body

CTN: Container

ITM: Item/ItemSequence

#### ◆ \_PrDX

病名変遷を記録する xml ファイル。

```
<csx:infoNode uid="PL.0" category="Problem" family="ProblemList"
bearing="RL.PL.0 TRS.PL.0.1">
  <csx:description uid="PL.0.DS">
    <csx:perspective domain="Medicine" category="Document"
family=" 診療録 . 一号様式 " />
  <csx:description uid="PL.DS">
    <csx:perspective domain="Medicine" category="Document"
family=" 診療録 . 一号様式 " />
```

#### ◆ \_PrMa

2号様式を記録する xml ファイル。試作実装としてオーダエントリ情報なども格納する。

```
<csx:infoNode uid="CTX.0" category="Document" family="Session"
bearing="RL.C2F.CTX.0">
  <csx:description uid="CTX.0.DS">
    <csx:perspective domain="Medicine" category="Document"
family=" 診療録 . 二号様式 " />
  <csx:description uid="CTX.DS">
    <csx:perspective domain="Medicine" category="Document"
family=" 診療録 . 二号様式 " />
```

#### ◆ \_PMLk

上記の \_PrDX.xml と \_PrMa.xml との同期と関連事項のリンクを取るためのファイル。

```
<csx:description uid="CTX.DS">
  <csx:perspective domain=" 医療 " category="Process"
family="ClinicalCourse" />
```

診療セッションを同定する CTX とそれに伴っている連番によって同期する。

#### C. 5. 4. 5. 2 本研究での全体像

本研究における直列化設計としては `_PrDX` と `_PrMa` とを融合して診療経過構造とし、これとは別に思考過程構造を用意して `_Vector` とし、両者の同期は Cycle 番号を用いることを想定した。ただし、実際の試作実装については委託業者の構成に拠ることとした。なお両 xml ファイルを纏める際の xml 構造テンプレートとして以下を用意した。その際の uid 附番の冒頭には ORI を用いる。

```
<csx:facet ..>
  <csx:infoNode uid="ORI.****" ... > ...
  <csx:ArcScope uid="ORI.****" ... > ...
  <csx:description uid="ORI.DS">
    <csx:perspective domain="Medicine"    category="ThinkingProcess"
                                          family="ClinicalCourse" />
</csx:facet>
```

#### C. 5. 4. 5. 3 診療経過構造 (Course structure)

診療経過としての構成に関わる主要素の記述の概形を以下に掲げる。各要素の category/family ならびに uid には、<C.5.4.4 採用する category/family> および <C.5.4.5.1 試作実装における uid の扱い> に拠る。なお ## は連番である。

##### ◆ Course

```
<csx:infoNode uid="ORI"                                category="Course" family="Course"
bearing="****">
  <csx:description uid="ORI.DS">
    <csx:perspective domain="Medicine"    category="Process"
                                          family="ClinicalCourseStructure" />
```

##### ◆ Thread

```
<csx:infoNode uid="ORI.TRD.##"                        category="Course" family="Thread"
bearing="****">
  <csx:description uid="ORI.TRD.##.DS">
    <csx:perspective domain="Medicine"    category="Process"
                                          family="ClinicalCourseStructure" />
```

##### ◆ Cycle

```
<csx:infoNode uid="ORI.TRD.##.CYC.##"                category="Course" family="Cycle"
bearing="****">
  <csx:description uid="ORI.TRD.##.CYC.##.DS">
    <csx:perspective domain="Medicine"    category="Process"
                                          family="ClinicalCourseStructure" />
```

##### ◆ 階層的な格納

```
<csx:arcScope uid="RL.ORI"                            category="Construct" family="Container">
  <csx:infoArc ref="ORI"                              category="Nominative"
family="Construct" />
  <csx:infoArc ref="ORI.TRD.##"                       category="Genitive"  family="Element" /> ...
</csx:arcScope>
```

```

<csx:arcScope uid="RL.ORI.TRD"
  <csx:infoArc ref="ORI"
    category="Construct" family="Container">
    category="Nominative"
    family="Construct" />
    <csx:infoArc ref="ORI.TRD.##"
      category="Genitive" family="Element" /> ...
  </csx:arcScope>

```

```

<csx:arcScope uid="RL.ORI.TRD.##"
  <csx:infoArc ref="ORI"
    category="Construct" family="Container">
    category="Nominative"
    family="Construct" />
    <csx:infoArc ref="ORI.TRD.##.CYC.##"
      category="Genitive" family="Element" />
  ...

```

...

なお診療経過構造としての Cycle 配下については、\_PrDX や \_PrMa と同等である。

#### C. 5. 4. 5. 4 思考過程構造 (ThoughtProcess structure)

思考過程としての構成に関わる主要素の記述の概形を以下に掲げる。各要素の category/family ならびに uid には、<C.5.4.4 採用する category/family> および <C.5.4.5.1 試作実装における uid の扱い> に拠る。なお ## は連番である。

##### ◆ 概要

ここで @family="[subSpaceName] は視野の限定には利便であるが、割愛可能である。Thread についても同様である。

```

<csx:infoNode uid="ORI.TRD.##.CYC.##.[blockName]"
  category="[subSpaceName]"
  family="[blockName]"
  bearing="****">
  <csx:description uid = "ORI.TRD.##.CYC.##.[blockName].DS">
  <csx:perspective domain = "Medicine"
    category = "ThinkingProcess"
    family="[subSpaceName]" />

```

##### ◆ Cycle の構成

ここでは主要要素についての纏めあげのみを記しておく。

```

<csx:arcScope uid="RL.ORI.TRD.CYC.##"
  <csx:infoArc ref="ORI.TRD.##.CYC.##"
    category="Construct"
    family="Container">
    <csx:infoArc ref="ORI.TRD.##.CYC.##"
      category="Nominative"
      family="Construct" />
    <csx:infoArc ref="ORI.TRD.##.CYC.##.[blockName]"
      category="ThinkingProcess"
      family="[subSpaceName]" /> ...
  </csx:arcScope>

```

```

<csx:arcScope uid="RL.ORI.TRD.CYC.##.PRi"
  <csx:infoArc ref="ORI.TRD.##.CYC.##.PRi"
    category="Define" >
    category="Object" />
  <csx:infoArc ref="ORI.TRD.##.CYC.##.AEi"
    category="Reason" />
  </csx:arcScope>

```

```

<csx:arcScope uid="RL.ORI.TRD.CYC.##.Gai"
  <csx:infoArc ref="ORI.TRD.##.CYC.##.Gai"
    category="Decide" >
    category="Object" />
  <csx:infoArc ref="ORI.TRD.##.CYC.##.pGA|dGA|EP"
    category="Location" />

```

```

<csx:infoArc ref="ORI.TRD.##.CYC.##.rGAi"           category="Condition" />
<csx:infoArc ref="ORI.TRD.##.CYC.##.PRi"           family="Cause" /> ...
</csx:arcScope>

<csx:arcScope uid="RL.ORI.TRD.CYC.##.ASi|MAi"       category="Adopt" >
  <csx:infoArc ref="ORI.TRD.##.CYC.##.ASi|MAi"       category="Object" />
  <csx:infoArc ref="ORI.TRD.##.CYC.##.rAPi"         category="Condition" />
  <csx:infoArc ref="ORI.TRD.##.CYC.##. GAi"         category="Purpose" /> ...
</csx:arcScope>

```

なおこの書式の規定には前提事項があつて、PRi と AEi, GAi と rGAi, APi と rAPi はそれぞれ一意対応することを求めている。すなわち種々の個別の事由の chunking を求めている。

#### ◆ 事由とされるべき項の収集もしくは焦点化

また上述において arcScope が担っている predicate のイメージは「focus して take する」ほどのものである。したがって本来は predicate Focus は case Object のみを取るべきだが、ここでは predicate Take のイメージに引きづられて case Goal と case Source とを取っている。

このような predicate の混淆が生じた事情は、AE/rGA/rAP の生成と活用において、<C.5.4.2.2 因果的関連とその扱い> に記した AEp/rGAp/rAPp の機能役割を活用しなかったことに起因している。これは単に本研究主題の範囲ならびに試作実装における事情によっており、情報構造ではなくて画面設計において、AEp/rGAp/rAPp の編集環境を提供する良好な操作環境を構想し設計し実装していくことは、本研究範囲を超えるからである。この割愛した点については、今後「診療論理ワークベンチ」について研究する際に扱うこととする。

```

<csx:arcScope uid="RL.ORI.TRD.CYC.##.AEi"           category="Focus" >
  <csx:infoArc ref="ORI.TRD.##.CYC.##.AEi"           category="Goal" />
  <csx:infoArc ref="ORI.TRD.##.CYC.##.***"          category="Source" /> ...
</csx:arcScope>

<csx:arcScope uid="RL.ORI.TRD.CYC.##.rGAi"         category="Focus" >
  <csx:infoArc ref="ORI.TRD.##.CYC.##.rGAi"         category="Goal" />
  <csx:infoArc ref="ORI.TRD.##.CYC.##.***"          category="Source" /> ...
</csx:arcScope>

<csx:arcScope uid="RL.ORI.TRD.CYC.##.rAPi"         category="Focus" >
  <csx:infoArc ref="ORI.TRD.##.CYC.##.rAPi"         category="Goal" />
  <csx:infoArc ref="ORI.TRD.##.CYC.##.***"          category="Source" /> ...
</csx:arcScope>

```

なお \*\*\* は、対象とされた過去現在未来に embed された意味ある実要素である。そしてそれらが、各々 AEi/rGAi/rAPi の要素 AEij/rGAij/rAPij とされる。

## C. 6 試作実装

### C. 6.1 軌軸性

一般にデータが transactional な様相を含んでいる場合、つまりなんらかの過程の記述の際、経過の軌軸を確保しておく必要がある。軌軸は時間軸のみで事足りる場合もあれば、現象や事態に求めることもある。本研究では主題によって第一に PL, 第二に GA とする。Thread では後者が当の局面における駆動性を強く発揮するからである。

### C. 6.2 OOP への展開

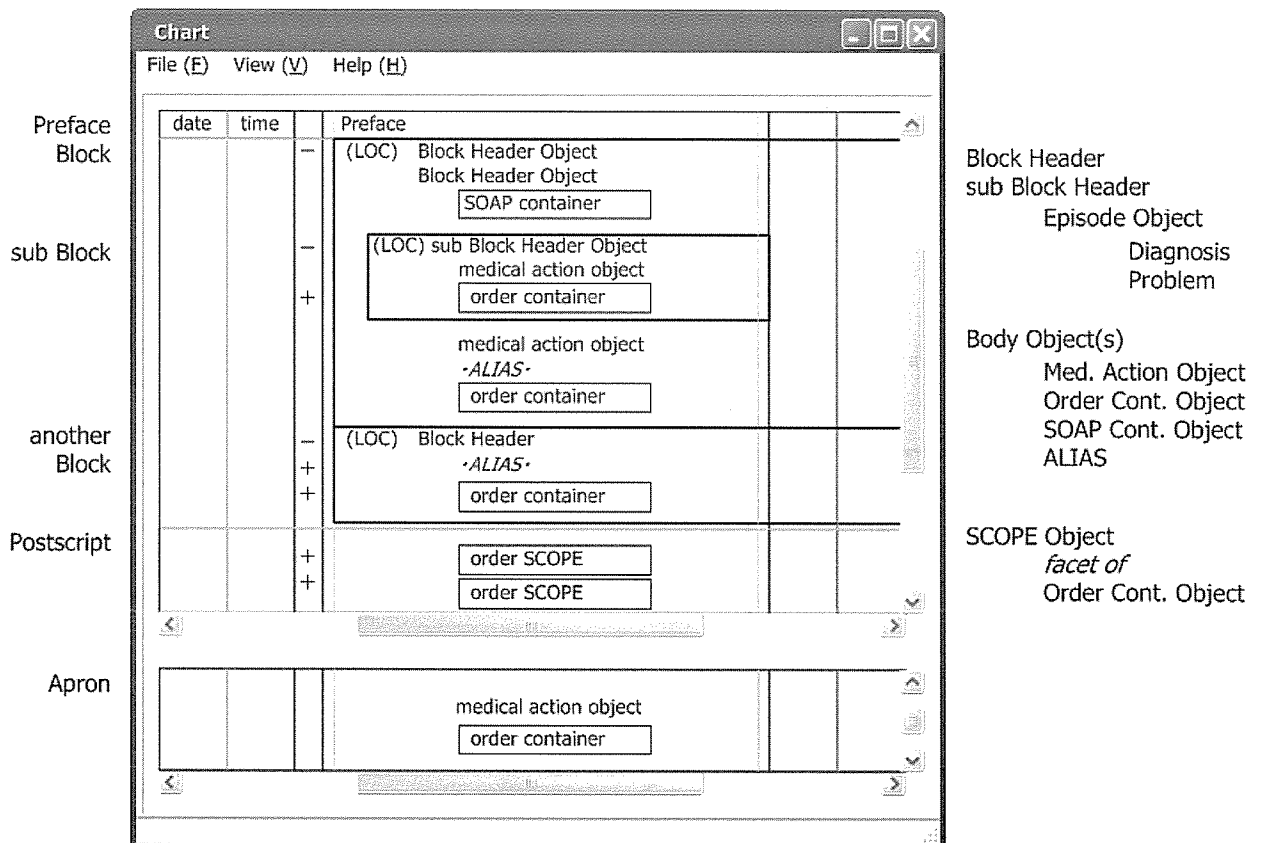
前項を直接に OOP するならば各 infoNode をクラス化していくこととなる。しかしそれは UML 環境について検討したように、必ずしも合理的でなく、むしろ CSX の特性を活かして interface として定義しておき generic な abstract を生

成し、そのうえで対象とする環境や業務に合致させながらコーディングするほうが、理論的には整っているように思われる。となると前項のクラスは全て一つで賄いことになる。そして polymorphism を活用するべく delegate を用意して展開していくこととなる。

なお Block や Cell は、その属する Cycle 内で他の Block や Cell と関係を形成できることは勿論、その Cycle を超えながら他の要素と関係を形成することも可能とする必要がある。この自由度は思考空間における複雑な事象の表現を支えることとなる。その自由度を明示的に制御する機能役割を、arcScope の扱いに絡むモジュール等に担わせることになる。

### C. 6.3 ライブラリ

必要な各種ライブラリの設計と開発については株式会社 TechCell に委託した。





### C. 6. 4 画面設計

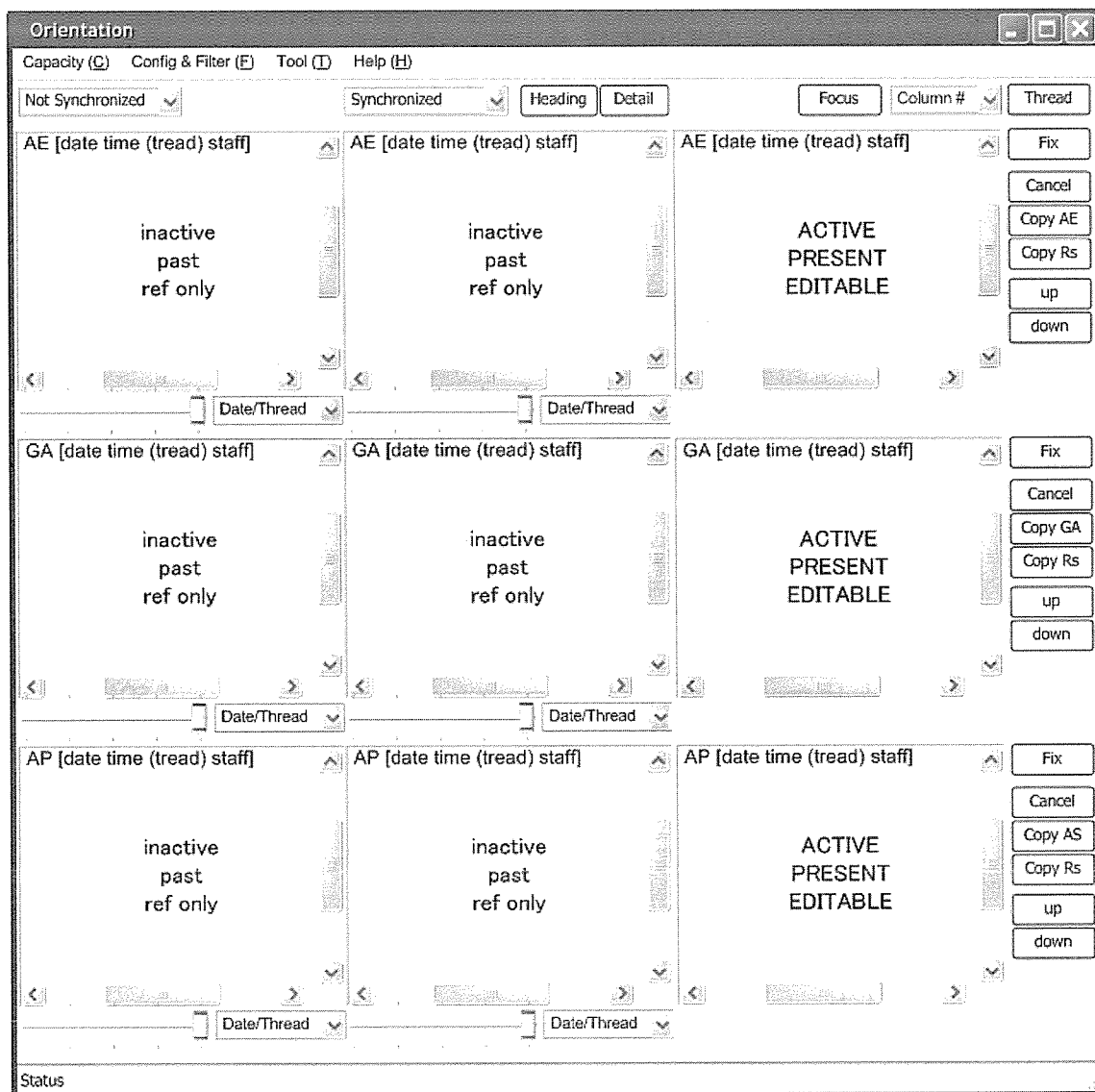
本研究の目的は「診療の方向性」に基づいて監査可能性を与える電子カルテの記述モデルを構築することであった。これは、意図実現過程モデルにおける GA を記録することを意味する。勿論それに先立って病名やプロブレムの変遷が記録されており、加えて病名やプロブレムと診療行為が連関されている必要がある。この二点については前研究プロジェクト (H15- 医療 -050) での成果を利用し、たとえば2号様式画面については前ページの設計を用いた。その環境は、病名やプロブレムと相応する介入を連関させる機能を有している。

そのうえで、GA を入力・編集・記録・参照できる画面環境を提供する必要がある。これについては下図の設計を考案し、これを用いた。

なお本研究主題は「記述モデル」についてであってシステム画面設計の最適化ではないことから、こと「画面」については、本研究で考案策定した記述モデルの意義や応用可能性などを評価するために必要な最低限の機能を実装することとした。

#### C. 6. 4. 1 PR/GA/AP

全体の構成としては、縦方向に PR/GA/AP を並べ、横方向には時間経過に従って (正確には



診療セッション Cycle) に従って三面を並べた。なお AR の pane のみは特殊で、むしろ AR 用 pane ではなくて PL 用 pane に見えてしまうが、これは一つには AE-PR 連関を目視しやすくするため、一つには AE-PR 連関と残り rGA-GA/rAP-AP 連関の表示を統一するため、そして今ひとつに PR/GA/AP 用の各 pane 機能の統一を図ったためである。したがって AE pane では「病名プロブレムエントリツール」を呼び出せるようになっている。

なお機能ボタンとしては前の Cycle から PR/GA/AP をコピーする機能、確定機能などを用意する設計とした。

### C. 6. 4. 2 AE-PR/rGA-GA/rAP-AP

各 pane では、各々 AE-PR/rGA-GA/rAP-AP

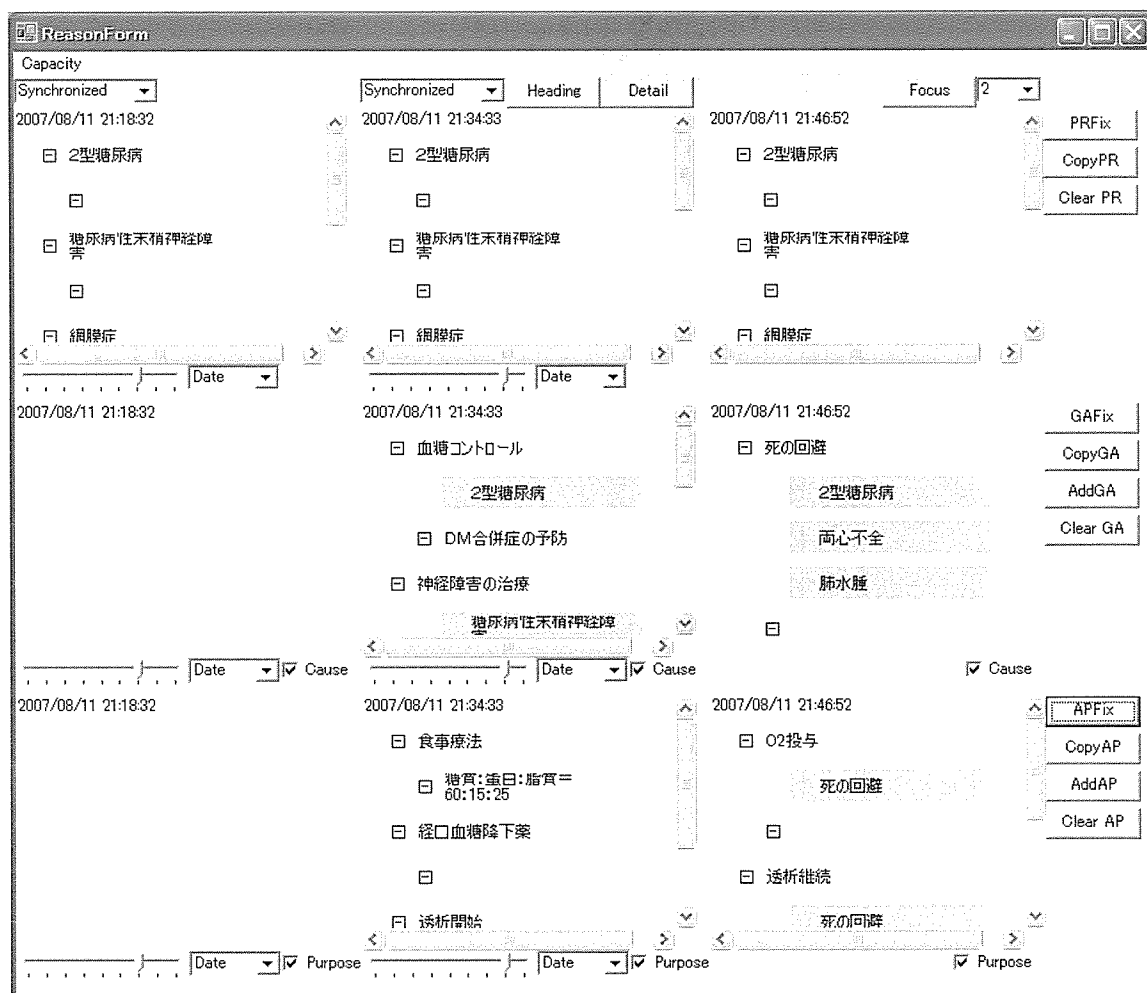
連関を設定できるようにしている。この機能とその連関の明示によって、何故に当の PR/GA/AP を選択したのか、その事由と因果を明示し、編集し、記録できるようにした。

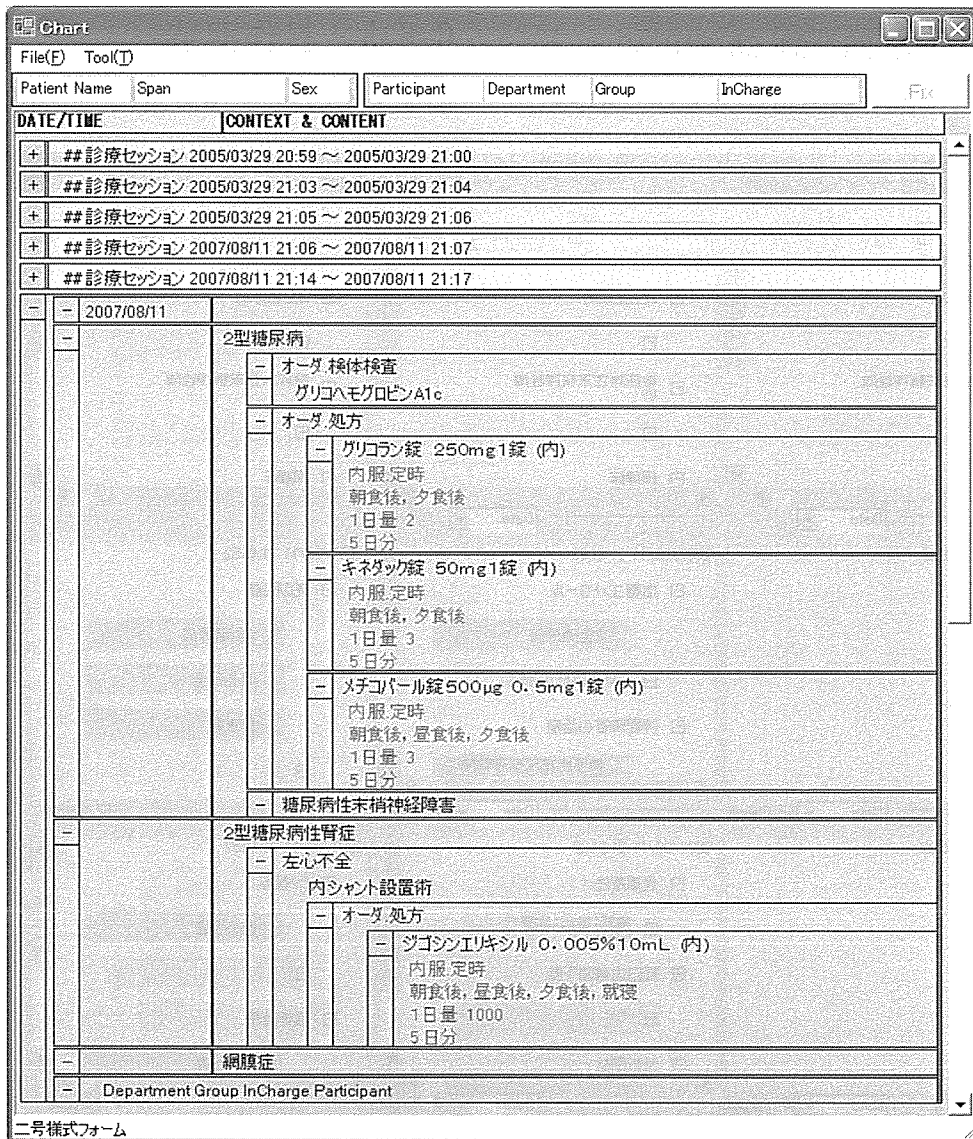
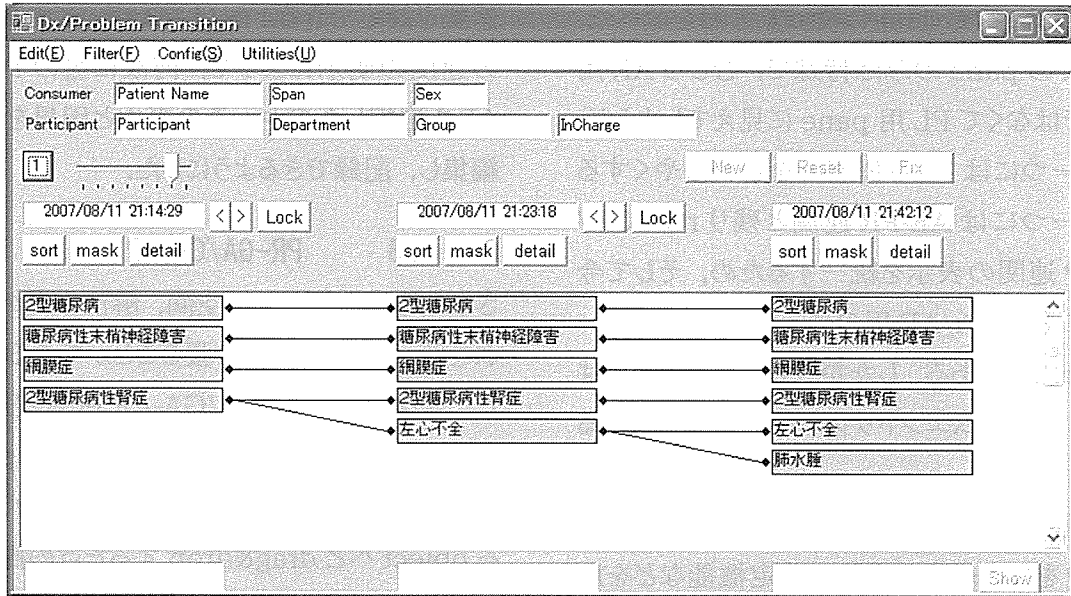
### C. 6. 4. 3 PR-GA/GA-AP

さらに問題空間、目標空間、解決空間のそれぞれを跨った意味リンクを張ることができるようにした。すなわち、PR-GA/GA-AP 連関を実現している。これは、例えば PR pane から GA pane へ跨って、ソースとなる object からターゲットとなる object へと drag&drop するだけの操作で実現できるようにしている。GA-AP 間も同様である。

### C. 6. 5 試作実装

診療システムの参照実装と、その出力する診療





履歴データ利用ツールの開発は株式会社創和ビジネスマシズに委託した。開発環境は C#.NET Framework である。

### C. 6. 5. 1 診療システム

ほぼ設計通りに実装しえた。

なお前述したように、1号様式画面ではなくて、AE/GA/APと事由とを制御する画面で多くのことを編集できるようにしている。1号様式画面では、変遷関係の線の制御と、病名やプロブレムの詳細属性とを行うのみである。

介入の実施と記録のために病名を2号様式へと展開する機能は、前研究プロジェクトと同様に、1号様式において展開操作をすることとなる。

### C. 6. 5. 2 データ利用ツール

さらに、前研究プロジェクト (H15- 医療 -050) と同様のツールを構築した。このツールによる可視化と情報共有の意義は大きいだろう。

これが比較的容易に実装できたことは、やはり表現枠組ならびにモデル設計に拠るところも大であらう。

