

ける記述物として表明されているが、この観あるいは相においては、どちらの状況であるにせよ、意図や目標や計画の instance object であることに変わりはない。

また (iv) 欲求または目標の実現もしくは達成とは、個々の小さな行為を単に発現するコトで終えられるのではなくて、通常はそれらの系列の集積によって、大目標が達成されることを意味している。したがって「実現もしくは達成」の記述」それ以前に、三点が必要である：(a) 意図または目標の記述 (b) 現実世界における事象の記述 (c) 意図または目標と現実世界における事象との比較。現実世界の事象は、それ自体すでに何らかの class の instance なのであるから、その事象の記述をしてその事象の instantiation であると扱うことは出来ようもない。

そもそも、この (b) という過程は、現実世界における事象を、当のヒトが当のヒトにおける観察すなわち認識によって検知した事象であった。このような事態を鑑みると、先とは逆に、現実世界から思考世界へと (変形や修飾や逸脱を伴うかもしれない状態で) 移す (写す) ことが為されている。したがってこの過程は、そしてこれらの object の関係は、detectOf とでも呼ばれる意味役割となる。このように detectOf されて思考世界に取り込まれた当該事象が、引き続いて何らかの思考素材として活用される際には、広義には referTo であろうし、狭義には様々な意味役割が割り当てられることになる (初年度までの成果と報告書を参照願いたい)。

意図実現過程の形式的な記述は、とくに類種を意識する、あるいは class や instance という考え方を適用しようとするとき、却って混乱しやすいように思える。しかし上述に見てきたように、instance の instance を想定する必要もな

ければ、なにかしら中間的な階層を想定して中間的なクラスを想定する必要もない。なお属性と属性値で表現するという手法も考えられるしそれでも処理は可能であるが、ただ UML class diagram の身上の一つである理解のしやすさは損なうことになるように思われる。思考世界と現実世界とのギャップは大きく、かつダイナミックであるにも関わらず、そのような状況を絵図として明瞭に表明しないことになるからである。

### C. 3. 5 点検

前節の要件と昨年度までの成果とを対比して点検する。

#### C. 3. 5. 1 記述順序など

<C.3.3 要件定義の諸準備> において見たように、表現枠組において、構造 (syntax) と意味 (semantics) とは表裏を為して不可分であり、また (Kant に云わせれば先験的な) 思考枠組がこれらを支えている。思考枠組においては、(i) 根底的な表象形式と (ii) 判断枠組 (あるいは形式) そして (iii) 存在論的な認識枠組とが不可分な構造として構想される。

したがって表現枠組はこのような広義の認識枠組 (すなわち思考枠組) に影響を与えることとなり、意味的構造の表現可能範囲を自ら規定することとなる。この事情によれば形式的構造を先に論述せざるをえないことにもなるのだろう。

表現枠組とはすなわち形式的構造である。その仔細は <C.1.1 表現枠組 CSX> にも述べたが、<C.3.5.2 形式的構造> の事項も看過することはできない。

意味的構造には二つの側面がある：

- ・ 認識の形式
- ・ 関係の種類ならびに関係の間の関係 (体系)

前者は表現枠組の応用手法によって示される。本研究主題への応用は後述する <C.4 表現枠組 CSX> にて述べることとするものの、その一部については下記に続ける <C.3.5.3 意味的關係 (認識の形式)> との重複は不可避であろう。一方、後者の一部は本研究成果の一つである fcet.atst.csx.xsd あるいは他の語彙体系などによって示され、また別の一部は、表現枠組の応用手法によって示される。下記に続ける <C.3.5.4 意味的關係 (諸般と範疇)> では CSX の構造とともに fcet.atst.csx.xsd の概要を点検することになる。

### C.3.5.2 形式的構造

形式的構造については <C.1.1 表現枠組 CSX> と <C.3.4.2.1 基底構造> ならびに <C.3.4.2.2 空相> も参照願いたい。

現有のメタ表現枠組 CSX は、<B.1 経緯参照> および <C.1 これまでの成果> に掲げた対象概念を担う節N, 対象間に関係を結ばせる節S, その関係において個々の節Nが担うべき個別の関連意義役割を担う辺A, 要素P, 要素Q, 要素Tを有しており、かつそれぞれの機能や構成は <C.3.4.2 形式的構造> に提示した要件と相応している。したがって現有のメタ表現枠組 CSX は、形式的構造の大概として要件を充たしている。

また節N, 節S, 辺Aの具体の意味の「位置づけ」は、各々の属性である catagory (圏) と family (族) とによって表現されている。なおこれらの属性は class (類) や kind (種) とは呼んでいない。Catagory や family に格納されるべき値は、そしてそれらによって表現されるべき事項は、「意味關係の關係」であり・あるいはまた「事物事象間の關係」である。すなわち将に此処において「観」が示されることとなる。

表現枠組 CSX の基底モデルは基底的であるがゆえに、それ自体は、そのままでは何も表さない。単に「根底的な形式構造」としてのメタメタ構造があるのみである。具体の認識枠組は、すなわち「観」は、これら catagory (圏) と family (族) とが具体の値によって埋められることで、具体の枠組としてたち出でて来ることになるのである。メタ表現枠組 CSX は、そのように設計されている。

観は多様である。また多様であって然るべきである。前述した <C.3.3 要件定義の諸準備> を一瞥しただけでも、そのアプローチには哲学、言語、論理、計算があり、そのうえ具体の適用に当たっては領域特異性も加わることにあり、さらに形式的には大域局所問題も加味せざるをえないこととなる。これら全てを同根に帰しうるような包括的な体系の構築を試みることは、挑戦的に過ぎるように思われる。

現実的で可能なことは、「根底的な形式構造」としてのメタメタ構造に対して、メタ構造たる「意味關係の關係」あるいは「事物事象間の關係」を観 (catagory 圏や family 族) として賦与しながら・モデル (class 類や kind 種) を生成し、そして個々の class を現実化する (instanciation), ということである。ただし <C.3.4.2.4 意図と具現と記述> に記した諸關係と Instanciation とを混同してはならない。

本研究、ならびに厚生労働科学研究 (H15- 医療 -050, H17- 医療 -043) が提案主張する核心とは、将にこの図式である。この図式において、前述に掲げられている求められるべき表現可能性が確保されている。

なお要素P, 要素Q, 要素Tの幾つかの属性も、必要に応じて coding schema を与えることもできる。また class と instance は同定子の附番規則によって判別しうる。構造の仔細について

は <C.1.1 表現枠組 CSX> にて述べた。

### C.3.5.3 意味的關係（認識の形式）

対象性，すなわち実体概念と属性概念との弁別・詰まるところ属性關係と関連關係との弁別は，哲学的においても情報構造設計や記述枠組設計においても困難な問題である。後者については主辞賓辞に関わる考察によるのみでは判別を得られないこともある。そのようななか，two-level modeling approach における archetype の活用という手法も提案されているが，これは <C.3.3.3.2 境界問題> において，境界問題を生じせしめることを見た。

メタ表現枠組 CSX では，基底構造によって再帰構成しながら種々を生成し・その結合は観によって統合する戦略なので，境界問題を回避しつつ属性關係と関連關係（実体概念と属性概念）を扱う。ゆえに archetype も，この過程における中間構造と捉えうようになっている。

思考空間の区分化，物理空間内における位相關係，ならびに意図実現過程における因果關係と概念關係については，<C.2 意図と経験知の表出化> の <C.2.5 点検> において見た通りである。

時間断片における事象の事態認定と四次元連続体としての事象列については，プロブレム病名変遷を表現する形式が将にそのようになっている。個々の時間断片においては PL や GA において事態が語られ，また一つのプロブレムや病名の変遷は・その変遷を包摂している四次元連続体としての「観」が与えられている。すなわち，そのようにして同一性を認定することによって・プロブレムや病名の変遷を捉える認識形式を反映している構造となっている。

そして時制としての独立的な時刻列については，事物事象の意味的關係とは無關係に，システ

ム内において唯一つの順序關係を表現する節Nを設けておいて，そのなかに・発生した全ての事物事象を表す節Nを格納する（実際には各々の節Nへのポインタを節Aとして纏める）ことを想定している。

また過程におけるコトの部分全体關係や同時性・共起性・順序性などの先後關係については，要素Tとその配下としての要素Qによって表現可能としている。

これらのことから，したがって現有のメタ表現枠組 CSX は，認識の形式に関わる意味的關係を表現するにあたって必要となる形式的構造 <C.3.4.1.2 認識の形式> の要件を充たしている。

### C.3.5.4 意味的關係（諸般と範疇）

前項までにおいて，形式的構造の妥当性ならびに形式的構造が認識の形式に関わる意味的關係の表現を支援することを確認したが，これらに具体性が与えられるためには，節Nや節Sあるいは辺Aなどへ，<C.3.4.1.1 關係と範疇> にて示したような，謂わば「彩色」が施される必要がある。その一部については fcet.atst.csx.xsd で定義している。

ただし表現枠組 CSX は，それのみではなくて他の体系をも参照できる構造としている。なお本節は <C.1.1 表現枠組 CSX> を参照しつつ卒読願いたい。

#### C.3.5.4.1 分析判断的記述

まず対応せねばならないのは存在論的範疇論であって，すなわち量数と性質，場所と時間，そして關係である。実体は基本的には節Nで表現される。これが主辞となりうるのか賓辞における名詞句とのみなりうるのか，これを換言するなら或いはクラスと扱うか属性と扱うかは，必要に

応じて判別できる形式が具備されれば、存在論的には宜しかろう。CSX は、これらの表現可能性を満たしている。

なお属性の種類や範疇を表現する概念で・かつ他の概念を値として持ち・賓辞として使われるものは、要素Qを用いた便法で表現できるようになっている。

関係については、CSX では意味関係と意味役割とを分離分割し、前者を節Sにて後者を辺Aにて表現している。一つの実体もしくは概念は、他の様々な実体もしくは概念との間に意味関係を結んでおり、一つの纏まった意味関係においては一つ以上の意味役割を担っている。この状況を正確に表現するために、CSX は、そのような形式とされているのである。

時間とは諸実体 ( 諸事物 ) 間の関係のありかたを示す認識形式の一つであるため、根源的には実体としては扱わない。とはいえ、事象を表現する実体においては、すなわち節Nにおいては、時刻を表現したいこともある。このような便宜的な要求を満たすために、節Nの要素Qによる量数的な表現も許容している。

節N	実体 ( コト, モノ )
要素Q	量数, 性質, 場所 ( 狭義 )
節S	関係 : 一つ関係全体ほか
辺A	関係 : 意味役割
要素T	系列的または位相的な関係

Aristotle の云う能動と受動および姿勢と様態や状態と所持については、述語動詞をも含めた広義の関係を構成する要素、すなわち節Sと考えることができる。姿勢や様態は多様であるが、CSX は少なくともそのうちの実体間の空間的な位置関係に加えて時間や事象などの抽象的な「位置」関係をも表現することができる。これは、或る関係 ( 節S ) において・その関係における役割 ( 辺A ) に関する「位置づけ」を要素Tで

表現することによって実現している。

表現枠組 CSX は再帰的な構成員ならびに範疇子 (category/family) によって多様な事物事象を表現できる形式構造となっており、これらのことと上述したことの応用展開が降に掲げた事項となっている。

#### C. 3. 5. 4. 2 総合判断的記述

次に対応すべきは思惟的範疇論である。あるいは、Kant による言葉を借りるなら、すなわち分析判断とは対比的な判断としての、総合判断における表現に関する諸範疇である。それらは量、質、関係、様相という順で、その名辞が列挙されるが、モデル化枠組という観で眺めるとき、最初に関係 : 定言を挙げるべきだろう。

##### 定言

これは分析判断の骨格を成すものであって、は対象化であり概念化であり実体化、あるいは class 化や instance 化と、それらの ( 否定や依存や様相を伴わない ) 結合である。ここで結合とは言明と同値である。またこのような言明には当然ながら supertype の述語付け ( 賓辞化 ) も含まれている。

そして概念化あるいは class 定義とは固有性 ( もしくは別々性 ) の表明や主張であることから、それと表裏されて偶有性も同時に明らかになる ( ことが多い )。なお、偶有性としての属性も固有性としての属性も、量質という概念範疇のみに限定されているわけではなく、実体もそのような属性である意味役割を果たすことができる ( たとえば上述した supertype など )。

CSX では、事物事象 ( モノやコト ) を現す情報塊を節Nによって表現している。なお class と instance との別は、oid や uid に依っている。また述定については節Sが、その述定なる「関

係」における各項の個々の意味役割については辺Aが担うこととしている。

#### 関係 (Relation)

定言 (AはBである)

- － 実体性 (Subsistenz)
- と偶有性 (Inhaerenz)

#### 範囲限定

全称, 特称, 単称の別が謳われることの意である。CSX では, 全称と特称は節Nの限量子属性の値において明示される。なお単称については, その属性には何も格納されないことによって暗示される。

#### 量 (Quantitaet)

全称 (全のAはBである)

- － 単一性 (Einheit)

特称 (或るAはBである)

- － 数多性 (Vielheit)

単称 (此のAはBである)

- － 全体的 (Allheit)

#### 極性範囲

ここで極性とは肯定否定の謂いであり, よって否定は肯定極性の反転と捉えることができる。また範囲とは, 極性反転が作用する範囲, 否定辞が作用する範囲の意味である。

認識という機構において留意すべきは, 述定動詞における否定と名辞における極性反転とでは, その想定される表象が本質的に異なることである。前者は「その関係における関係の分断」であり, 後者は「その関係において結合された対象において・その対象限定を外すこと」である。なお閉世界を限定したうえで二値のみを採用した場合には, その時に限って, 名辞の極性反転と述定動詞の極性反転との意味は相応合致するが, これはむしろ特殊である。

CSX では節Nの属性において極性反転子すなわち否定辞を取りうるが, しかしこれは現仕様に

おいては先に述べた限量子が格納されるべき属性と同一としている現況である。また CSX の現仕様においては, 述定動詞を表現する節Sには, 極性反転子を格納する属性を有していない。

#### 質 (Qualitaet)

肯定 (AはBである)

- － 実在性 (Realitaet)

否定 (AはBでない)

- － 否定性 (Negation)

無限 (Aは非Bである)

- － 制限性 (Limitation)

#### 依存関係

依存関係には, 仮言ならびに選言が挙げられている。仮言とは依存関係あるいは因果関係であり, 選言とは排他的選言すなわち相互依存を指している。

ここで云う依存関係は思惟的範疇論のそれであるからメタあるいは上階であって, ゆえに具体について, 仮言は個々の言語の文法において種々に展開されることが前提されている。

一方, 排他的選言とは, 選言肢間には排他的な選択状況のあることの言明されていることから, それら選言肢は総体として一を成すことが含意されている。ゆえに相互排他的とは, 逆から見るならば, 相互依存した存在関係であることとなるわけである。具体においては, たとえば論理演算子の一つとして扱われる。

両者とも節Sにおいて意味関係が示され, 辺Aにおいては個々の節Nが担うべき意味役割が示されることになる。具体の仮言は, 基盤的な前件と共に付随的な前件あるいは諸条件が相俟って後件が成立することがある。このようなとき複数の項すなわち節Nが, 各々の意味役割を担うことになる。

なお定言と同様に仮言も選言も確固たる言明であって, すなわち主張もしくは信念であって, こ

の点が蓋然や確然との決定的な差異である。両範疇の混同は当然ながら必ず回避されねばならない。

#### 関係 (Relation)

仮言 (AであればBである)

- － 原因性 (Kausalitaet)
- と依存性 (Dependenz)

選言 (AであるかBである)

- － 相互性 (Gemeinschaft)
- 能受の間の相互作用

#### 様相

すなわち様相論理に関わる関係である。可能世界意味論において、実然とは現に到達していることであり、確然とは到達することを要するという判断であり、蓋然とは可視であるという判断である。

具体、すなわち個々の言語の文法においては、法・相・時制が相絡まって表層表現様式が現出されることとなると共に、また個々の言語に応じて、感情的な「色彩」の薄いような単なる到達可能性のみならず、願望や後悔などの話者の微妙な心的陰翳が加えられて法助動詞などが用いられることとなる。

#### 様相 (Modalitaet)

蓋然 (AはBでありうる)

- － 可能性 (Moeglichkeit)

実然 (AはBである)

- － 存在性 (Dasein)

確然 (AはBであるを要す)

- － 必然性 (Notwendigkeit)

#### C.3.5.4.3 包含と継承

包含関係あるいは集合間の関係も他の事項の表現と同様に、節Sにおいて包含関係という意味関係が示され、辺Aにおいては個々の節Nが担うべき意味役割としての上位性あるいは包摂性と下位性あるいは被包摂性などが示されることになる。

ただし現況の fcet.atst.csx.xsd では、集合における包含関係を直接的に表現するための、節Sや辺Aの属性値は定義していない。むしろ文書構造や意図実現過程構造などの比較的大きな情報塊構造において、節Nが container とであるとか、container に含まれる cell であるとか、そういう表現として定義している。この事由は、単に本研究主題に即したということのみである。継承の扱いは、表現における課題というよりもむしろ、アプリケーションにおける処理上の課題として捉えられるように思えるし、実際のところ表現枠組としては書き記すことのみで終わってしまう(ことが多い)。したがって表現枠組 CSX においては、たとえば継承を表現するための属性を節Nに追加したなら、少なくとも表現としては事足りることになる。しかし継承は、どのような場合に継承として明示的に表現する必要が生じるのであろうか？

それは、個々の小宇宙的なモノとして分離されている多数のモノすなわち class があって、その中においてソレを継承する、と宣言したい場合だろう。逆にそのような状況が想定されていないか、あるいはそのような表現を回避して代わりに階層構造(の全体)を必ず表現するなどの方針がある場合には、継承を表現する属性を取って用意する必要も無くなる。すなわち必須ではなくなる。加えて、継承には厄介な問題が常につきまとう。よく引き合いに出される例としては、鳥の翼と飛ぶことであろう。あるいは犬やネコと、それらに対するものとしての AIBO などの「ペット性」であろう。いずれの例の場合にも、通常は後者を interface として継承することが UML 的であろう。

さらには、大きな階層があるとき、または構築しようとするとき、特に機能的な継承の場合、中位においてはいったん機能継承しな

い種類があつて、しかし更なる下位では再び機能継承する種類がある、というような階層しか構築しえない、という場合も想定される。この場合も interface を活用すれば、その機能を表現する、という意味においては解決できる。たとえば、そのように「表現された鳥」が(仮想世界のなかで)飛ぶことができるという意味においては全く問題を生じない。しかしながら、もしそのような階層があつて、そのような階層のもとに「鳥」を理解しなければ知識を得たことにならない、としたら、知識表現という目的においては、UML 的な interface 継承はどのように位置づけられるべきであろうか。

たしかに業務アプリケーションをコーディングする際には、この手法で解決できる事項もあるし、この手法が無ければ不適切なコーディング(というよりモデリング)となってしまうこともある。しかしながら interface として継承することで何が解決したのか、そもそも解決したその問題とは何に拠って生じていたのか、と云えば、一つに現実世界の多様な関係があつたことと、一つに処理上の問題などを回避するために多重継承を禁止したからであつた。そして後者は処理上の、つまり実アプリケーション構築上のみにおける問題である。

これらのことから、現況の CSX においては継承を表現するための属性は用意していない。

#### C.3.5.4.4 領域特異性

領域に特異的な関係は種々あつて、たとえば医学領域においては発現場所や原因など、枚挙に暇がない。本研究では、研究主題に必要な事項ならびに過去の研究経緯において既に登録していた事項のみが fcet.atst.csx.xsd に記されている。

#### C.3.5.4.5 役割と範疇

修辭関係すなわち主題役割または意味役割や格とは、或る関係が構成されている際に当該関係内において個々の句が担うべき意味的な役割のことであつたが、これは標識などによって識別される。標識とは、屈折語の場合には形態屈折であるし、加えて前置詞(句)が併用されることもある。膠着語の場合には格助詞などが標識となる。

いずれにせよ関係を構成する主役となる単位、たとえば述定動詞があつて、標識があつて、そのうえで個々の句が結合されていた。これらを抽象化して結合子：結合意味：結合対象と考えるならば、CSX では節S：辺A：節Nと相応することになる。それぞれの要素の属性に値を格納して、関係と個々の意味役割とを表現することになる。

前述した <C.3.3.2.3 言語と認知> に掲げてある殆どの事項は節Sや辺Aの属性値として fcet.atst.csx.xsd の定義に反映されている。ただし、経路(Path)や方向(Direction)については辺Aに加えて要素Tも用いながら表現することになる。

なお属性値の定義済み云々に関わらず、動作主(Agent)と行為者(Actor)とを弁別するか否かとか、被動作主(Patient or Beneficiary)や対行為者(Counter-Agent)や受領者(Recipient)を認定するか否かといった点は、表現枠組の課題ではなくて、その利用者側の選択として利用者に委ねられている。

次に(上位の)概念範疇あるいは概念の型について取り上げる。現況の fcet.atst.csx.xsd には、それらは定義されていない。というも(上位の)存在論的範疇(ontological categories)を策定することは、本研究の主題ではないからであ

る。むしろ <C.3.5.4.3 包含と継承> と同様に、文書構造や意図実現過程構造などのサブ領域における具体の範疇化を、しかも試作実装に必要な分だけを定義した。

とはいえ fcet.atst.csx.xsd に存在論的範疇を定義する(登録する)こと自体は、当然ながら容易ではある。もちろん存在論的範疇を策定することには相応する考察と労力が伴うこととなろうが、なお概念範疇もしくは概念の型とともに概念関数を整理することは重要だが、その際には相当の困難が伴うものと推測される。Jackendoff が挙げた事項のみでは不足すると思われるし、述語の特性を鑑みて整理することは、述語の複雑で錯綜した modality を十二分に検討する必要があるからである。

#### C.3.5.5 述定動詞の関係

これは体系としてのオントロジーを構築するにあたっては明らかにしておくべき事項である。しかしながら本研究主題の範囲外である。なお、その整理には <C.3.5.4.5 役割と範疇> に記した留意が必要であろう。

#### C.3.5.6 関係の関係

これは体系としてのオントロジーを構築するにあたっては明らかにしておくべき事項である。しかしながら本研究主題の範囲外である。



## C. 4 表現枠組 CSX

### C. 4. 1 更新

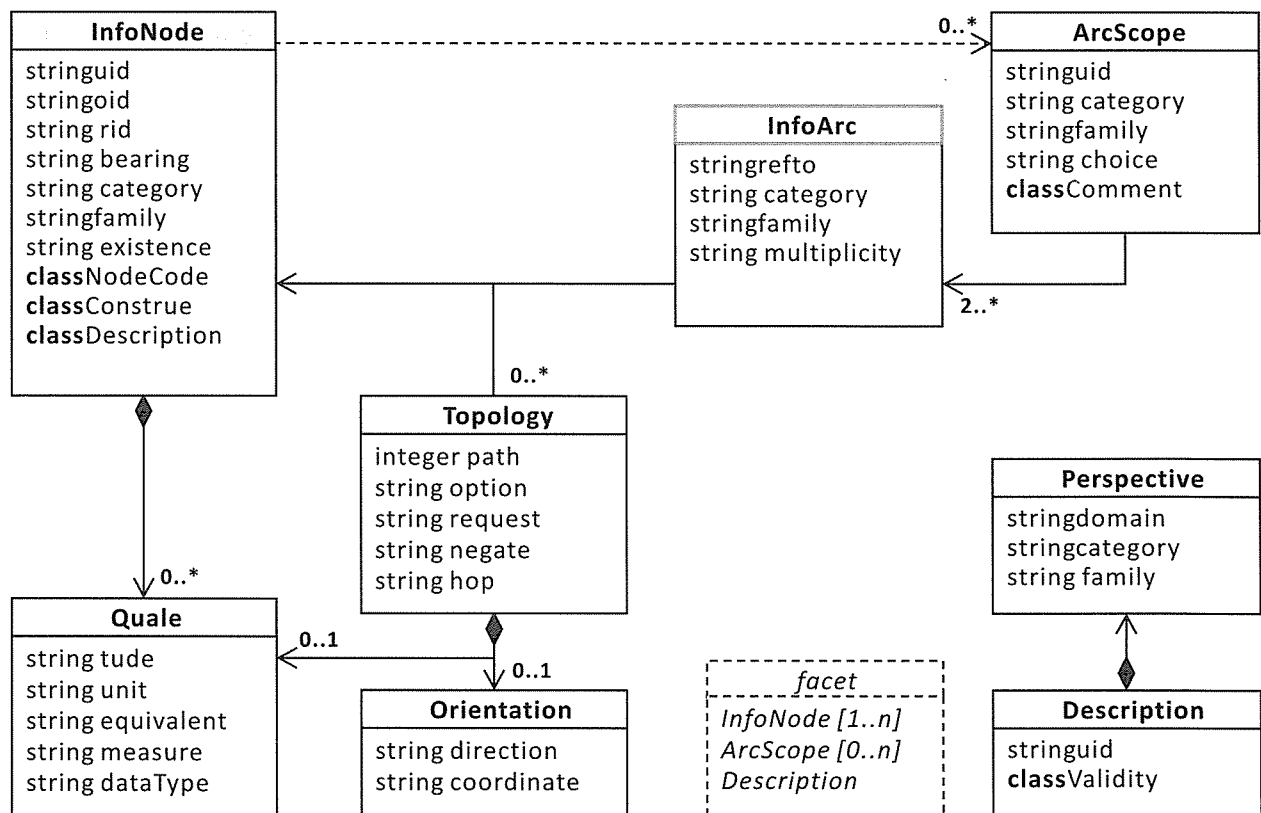
前述した <C.2 意図と経験知の表出化 > ならびに <C.3 オントロジー > から、観 (perspective) あるいは相 (aspect) や場 (scope) を明示的に扱うことの重要性は明らかとなっている。三者の差異は <C.3.3.3.5 観と相と場と > に掲げてはいるものの、しかし (i) 表現枠組に多くの属性を用意することは各々の用途について徒に混乱させうる要因ともなるし、また (ii) CSX は基底枠組を再帰的に用いて細粒度から巨大粒度の情報まで扱うものであるし、加えて (iii) 静的な体系のみならず処理の場などの動的な分野も表現対象としうる。これらのことから、表現枠組における属性名としては scope ではなくて perspective ただ一つとすることにした。すなわち description@scope を description@perspective と呼び替えることとする。

また CSX は「観」の記述を重視しており結果として、結合子 arcScope は多様な機能役割を担う要素として位置づけている。そして結合対象たる infoNode は atom としても container としても機能し、或る具体の infoNode( の存在 ) が想定されている空間あるいは領域を [@category @kind] で表現していた。しかしこのように見てくるとき、もはや類 / 種 (class/kind) というよりも圏 / 族 (category/family) と呼び替えたほうがより適しているので、以降は [@category @family] と呼ぶこととする。

次に dimension については量質の表現を担うにも関わらず、その名称から計量的なニュアンスを伴い易いことから quale と呼び替えることとした。

@kind	=>	@family
@scope	=>	@perspective
@ref	=>	@refto
dimension=>	=>	quale

なお、その他の事項については <C.2.5 点検 >

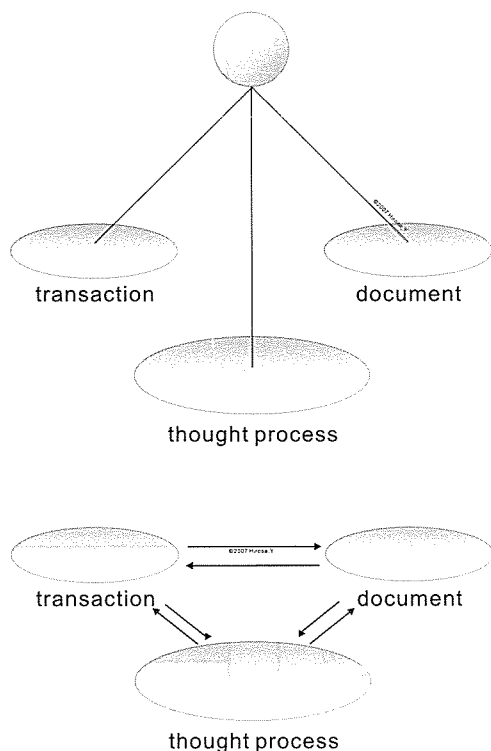


ならびに <C.3.5 点検> に挙げた通りである。

## C.4.2 観

主任研究者は以前より観・相・場を強く意識していた。だからこそ CSX のみならず、役割配役立場モデル (Cast-Character-Capacity model : 3C model) においても、actField (行為場) や actPoint (行為点) などを定式化したのである。これらは主任研究者には当然の発想であったため強くは発表してこなかった。しかし次の事由から、本件を本報告書でも取り上げることとした：(i) category/family の位置づけの明確化と理解を助ける説明、(ii) 情報の再利用や再構成、(iii) CSX の特徴の明確化、(iv) IT ontology において看過されがちであること。

なお以下を読み進む際は <C.3.3.3.5 観と相と場と> ほかに <C.3 オントロジー> のみならず、<A.5 これまでの研究経緯> <B.7 オントロジー> <B.8 記法> <C.1.1 表現枠組 CSX> をも参照願いたい。



### C.4.2.1 多様性

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う。

しかしその一方では、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、そのようにして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.4.2.2 圏と族

ある domain に 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。その下の層にはさらに subdomain を categorize できるだろう。

CSX の infoNode, arcScope, infoArc はそれぞれ属性に category と family を持つ。ある domain への perspective は infoNode@category に担わせることができる。その各層は、例えば値の書式にも容易に表現可能であるし、もちろん他の妥当な方策も採りうる。

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@family で表す。そして category も family も hierarchical な表現を許すので、family の値もまた taxonomical に

構成することができる。

一方, arcScope と infoArc の category や family を用いると, (巨視的な) 関係や仔細の意味役割について, それらの範疇的階層体系を構成することができる。

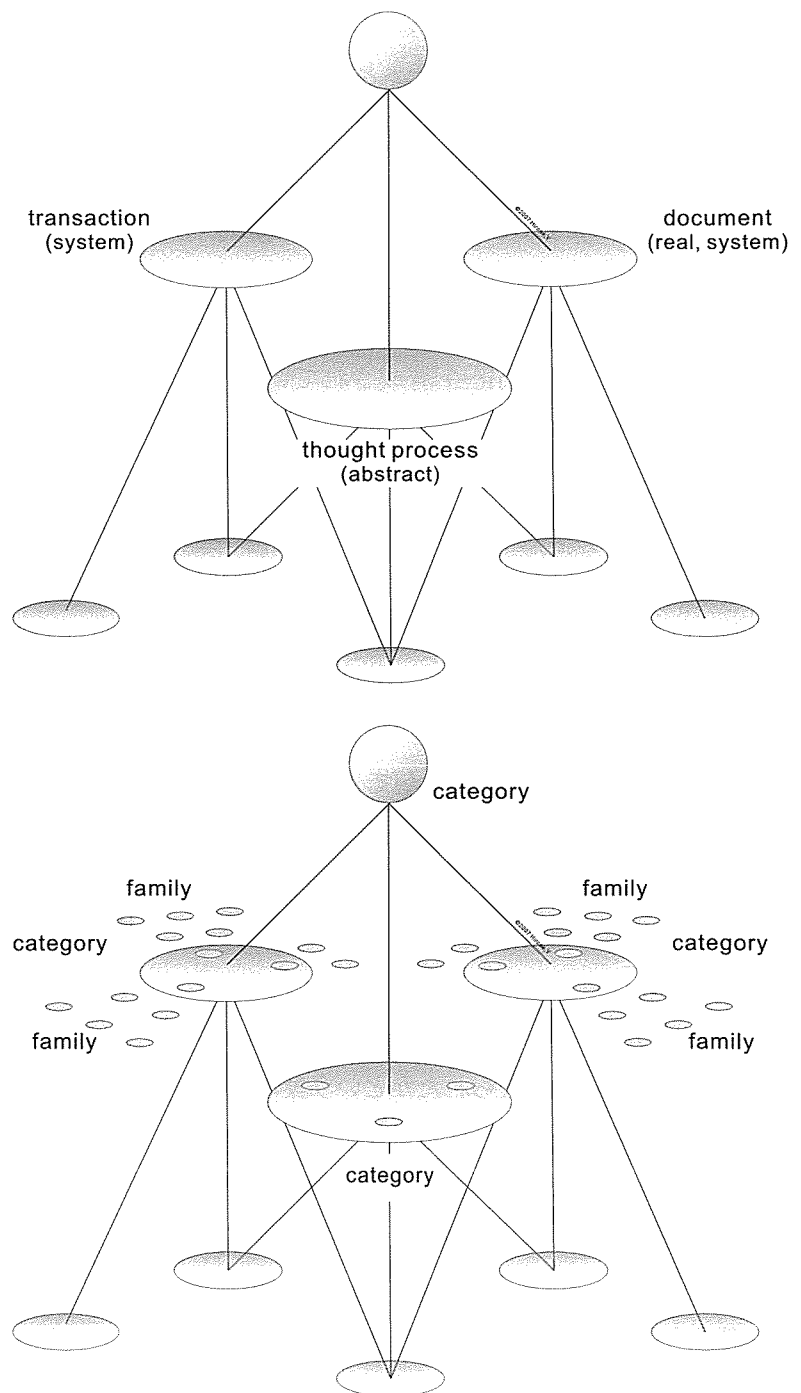
なお arcScope や infoArc を domain specific な関係にも活用しようとするとき, これら二者の category/family は infoNode のそれらと似通ることがある。

さて前述した Cycle の形式化で暗示しているように単位的な情報塊 (Cell) は, ある subdomain に配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。

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。

では money なる Cell はどうだろう。まず経営管理なる perspective で

は上位, もしくは変化に注目しやすいような位置に配座せられるであろう。しかし臨床思考過程という perspective では, たとえば Problem Space に現れるだろうか。通常は現れそうにはない。ただし例えば coin を aspiration したなどの症例の場合には, money.coin という形で PL の PR として現れてくるかもしれない。



### C. 4. 2. 3 自在性

となると個々の infoNode において、その意味内容を指し示す方策は如何にあるべきだろうか。コード体系はその目的からして視点を固定して整理されるが、それにも関わらず事物事象それ自体は様々な見方や扱い方をされることが常である。したがってコード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、硬直的な範囲でしか事物事象を表現できなくなってしまう。それでは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」することは容易ではなくなる。

一方 CSX は infoNode 属性である category と family とに perspective の表現を担わせている。属性 category も family も、格納する値それ自体が hierarchical 構成されていることも許容し、その値の root が単一であるとは想定してしない。よって表面上は category と family の二変量に見えても実際には多軸構成を可能としている。そして container として機能する infoNode は、container の内容要素として他の infoNode を格納する、つまり embed することで、個々の事象が如何なる観において捉えられたのかを表現することができる。そして内容要素 infoNode が指し示す意味は、その infoNode/nodeCode において、他の体系に存在している語や概念を指し示すことで表現するのである。

すなわち infoNode 自体が何物かという表現と、それ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 の変換は容易であり、かつ曖昧さや解釈の齟齬を回避しうる枠組とす

ることが出来ているのである。

そもそも対象とは、その対象がそのような対象として観られたような、その「認識枠組」と「(環境)構造」ならびに「(対象)範囲」とで見て取られているのであり、この事情は既に <C.3 オントロジー> にて仔細に見てきた。これは語や概念などの体系を構築する際にも、あるいは事象の同一性を認定することで変化を感知する際にも重要である。両者はその切り口が異なるに過ぎないのみで、認識なる基底的な機構については同等なのである。そして観こそが関係も粒度も定めていた。

したがってまた、観あるいは相は、巨視的にも微視的にもなりうる。だからこそ例えば、種々の異なる情報塊、しかも属する範疇が全く異なる情報塊を一括する文書を作成することもできるし、そのような会話を円滑に交わすこともできるのである。

そして perspective の生成も採りかたも自由であった。すなわち観自在である。

### C. 4. 3 直列化の一般的方法

CSX における直列化の一般的な方法については、厚労科研 (H15- 医療 -050) 平成 16 年度総括研究報告書 (ISBN 4-902408-09-0) の <C.10 Unique Identifier の設計>、<C.11 構成要素の直列化> および <資料 2> を参照願いたい。

## C. 5 意図実現過程モデル

### C. 5. 1 準備

#### C. 5. 1. 1 シナリオ

形式的なモデルに進む前に今一度改めて意図実現過程の 概要 を文言として整理しておく。

大前提と小前提と結論は演繹推論のそれである。なお問題空間・目標空間・解空間のそれぞれの局面において、三者は「概念としての疾病や病態」、「患者の疾病や病態」または「介入方針」または「概念としての具体の介入」、「症状や兆候」などに読み替える必要がある。

なお、あくまで「概要」ゆえ過不足はある。

現象や事態が提供される

現象や事態を感知する

現象や事態を認識して言葉にする

現象や事態を蒐集して考慮範囲に組み込む

Abduction を行う

大前提の候補を列挙する

大前提の候補を入れ替えてみる

結論を検証する

結論を再蒐集する

結論の強弱を勘案する

結論の遠近を勘案する

そのための試験的介入をする

小前提を獲得する

解決すべき課題を整理する

解決すべき課題に順位を定める

解決すべき課題の軽重を定める

解決すべき課題の解の方向を定める

deduction を行う

小前提の候補を列挙する

小前提を再検討する

小前提を再蒐集する

結論との不整合を検証する

解決すべき課題の解の方向に順位を定める

解決する際の具体的な方法を定める

deduction を行う

小前提の候補を列挙する

小前提を再検討する

小前提を再蒐集する

結論との不整合を検証する

解決する際の具体的な方法に順位を定める

解決する際の具体的な方法に順序を定める

結論との不整合を検証する

解決の具体的な方法を現実世界に発現する

結論との不整合を再検証する

小前提を再蒐集する

現象や事態に気がつく

現象や事態の変化を認識して言葉にする

現象や事態の変化を蒐集して考慮に組み込む

現象や事態の変化を認定する

解決すべき課題の変遷を認定する

解決すべき課題を再整理する

解決すべき課題に改めて順位を定める

解決すべき課題に改めて軽重を定める

適用方法と期待した結果との差異を検証する

小前提を再蒐集する

フィードフォワードするかフィードバックするか

あるいは

解決する際の具体的な方法を変更するか

解決すべき課題の解の方向を変更する

必要なだけ繰り返す

解決すべき課題の解の方向が達成されて介

入処理を終える

解決すべき課題の解の方向が不達成にて介

入処理を終える

### C.5.1.2 留意

感知や認識や認定は、<C.3 オントロジー> に留意する必要がある。

観測者かつ介入者は系のなかに組み込まれていること。主題においては、いわゆる物理的な客観（客体）もしくは物理的測定量のみが対象とされるわけではない。むしろ概念や思考という非物理的な対象こそが重要な役割を演じる系を「考察対象」しているそのなかに観測者そして介入者として存在していること。

## C. 5. 2 概念モデル

臨床思考過程モデルと診療経過モデルとの融合は、診療における意図実現過程モデルを形成することとなる。

### C. 5. 2. 1 プロセスの俯瞰

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当の診療現場の当の時点においては必ずしも明確に分離されうるわけではなく、むしろ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 だったのか? という意図とその扱いついては天賦されているようである。

よって本研究では Space を問題定義空間 (Problem Space), 目標決定空間 (Goal Space), 計画策定空間 (Solution Space) の三つとして捉う。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観てとり、それらの思考過程の各段階を Block と呼ぶ。

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) 等という事態や事象あるいは拠り所とする知識、計画の記述、そして発現された現実の行為などが格納される。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. 5. 2. 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図を挙げておく。これは 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まれていたこと、ではある。

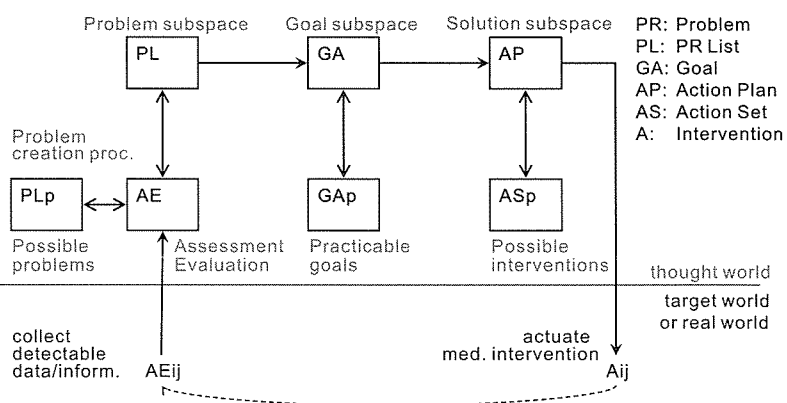
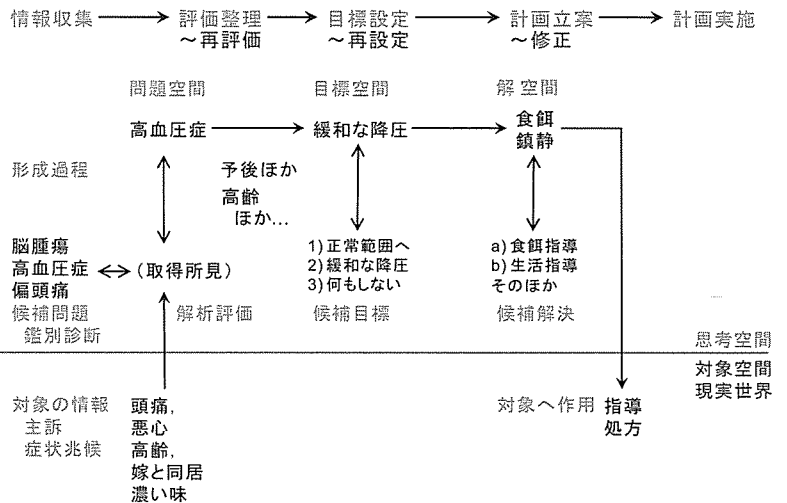
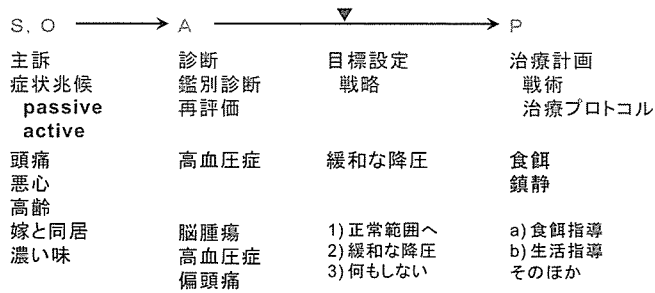
### C.5.2.3 サイクル (Cycle)

上述した仮想症例について初診時 (と引き続き診断相) 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のであろう。とはいえ思考空間において「現実・である」と認識される事象とは、思考が外界を認知すること、その対象事象を (正しい否かは別として) 思考世界のなかに取り込んだ場合のみに「存在」する過ぎない、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 (PR*i*) が挙げられ、プロブレムリスト (PL) が形成されていく。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 = 現実世界に発現され



ることになる。これらの状況構造を形式化するわけである。

AE*ij* 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。AE*ij* 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、または此処は、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 PL*p* 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一組として扱われるべき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. 各々の Block に格納される事項は, それぞれ  $GA_i$ ,  $GAP_i$ ,  $AS_i$ ,  $ASp_i$  などと表現し, 以下同様である. なお AP には,  $AS_i$  もしくは個々の介入行為である  $MA_{ij}$  が格納され,  $AS_i$  はさらに  $AS_i$  または  $MA_{ij}$  を格納するものとする.

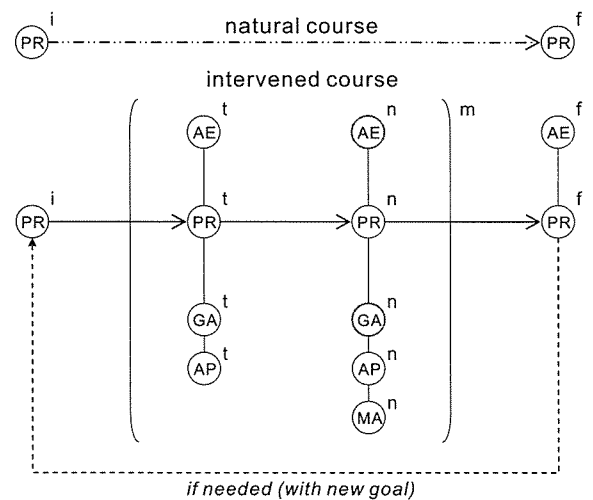
なお場合によっては, 思考空間内もしくは記述された計画としての介入行為は  $MA_{ij}$  と, 一方, 現実世界に「うつされ」て発現された介入行為は  $A_{ij}$  と, 弁別して記すことがある.

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なければならない. 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. 5. 2. 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, とくに pGA と EP については, しばしば当該 Thread に固有な pGA と EP とを持つことがある.

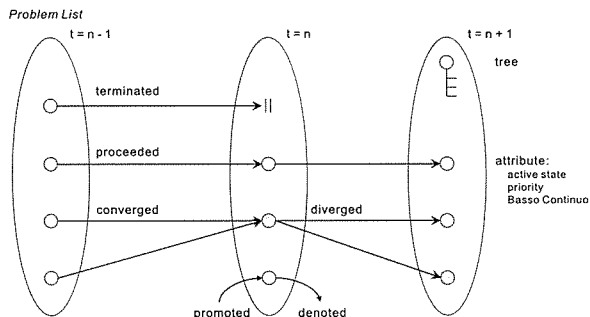


Thread は合目的的な連なり (のはず) なので消長には規則性を見て取ることができる. すなわち Thread レベルという局所プロセスでは, GA の達成と GA と現実 AE<sub>ij</sub> との比較に眼目が置かれることになる. ただし全体的には PL の軸性が失われるわけではない. Thread の消長に関する仔細は <C.5.3 Thread 遷移と意図> にて述べる. いずれにせよ GA と AP とを分離して定式化していたことは意図実現過程の定式化においても意義ある方策であった. GA 特に dGA とは「方針」あるいは「上位の意図」を示しており概して安定的であって, 逆に AP は個別的で具体的な意図である (比較的具体的に

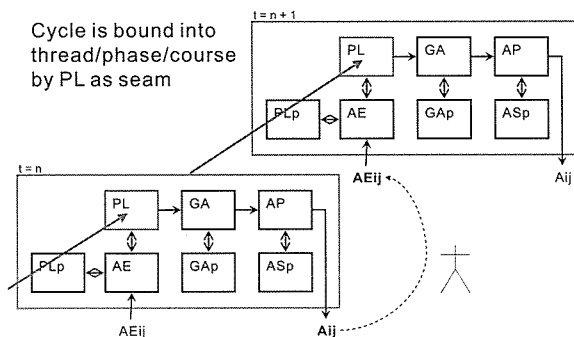
な) 計画だからである. AP は環境と環境の変化によって様々であったり availability が変化し, また動的に変更されることになる.

### C. 5. 2. 5 スパイラル (Spiral)

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. 変遷パターンには幾つかの類型がある.



それぞれは前研究において, 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, 全ての診療過程は基本的には PLこそ契機であり原動力である. とはいえ個々の局面においては, つまり Thread においては PL/Pri ならびに種々の状況によって決定される GA と EP によって主に「駆動」されている, と見なされうる. したがって, いずれにせよ Cycle と Cycle とを繋ぎ Thread そして Phase や Course を成す「綴じ」は PL である, ということになる.

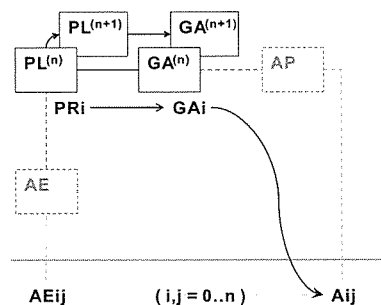


現実世界に「うつされた」介入事項 Aij またはその結果は患者を介して新たな事象 AEij として立ち現れて来るのであるから, 結局 Course 全

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. 5. 2. 6 Goal の意識

前研究プロジェクト H15- 医療 -050 では PL と Aij との短絡を実現することで種々の成果を得た. 本研究ではこれを発展させて GA を絡めた定式とした.

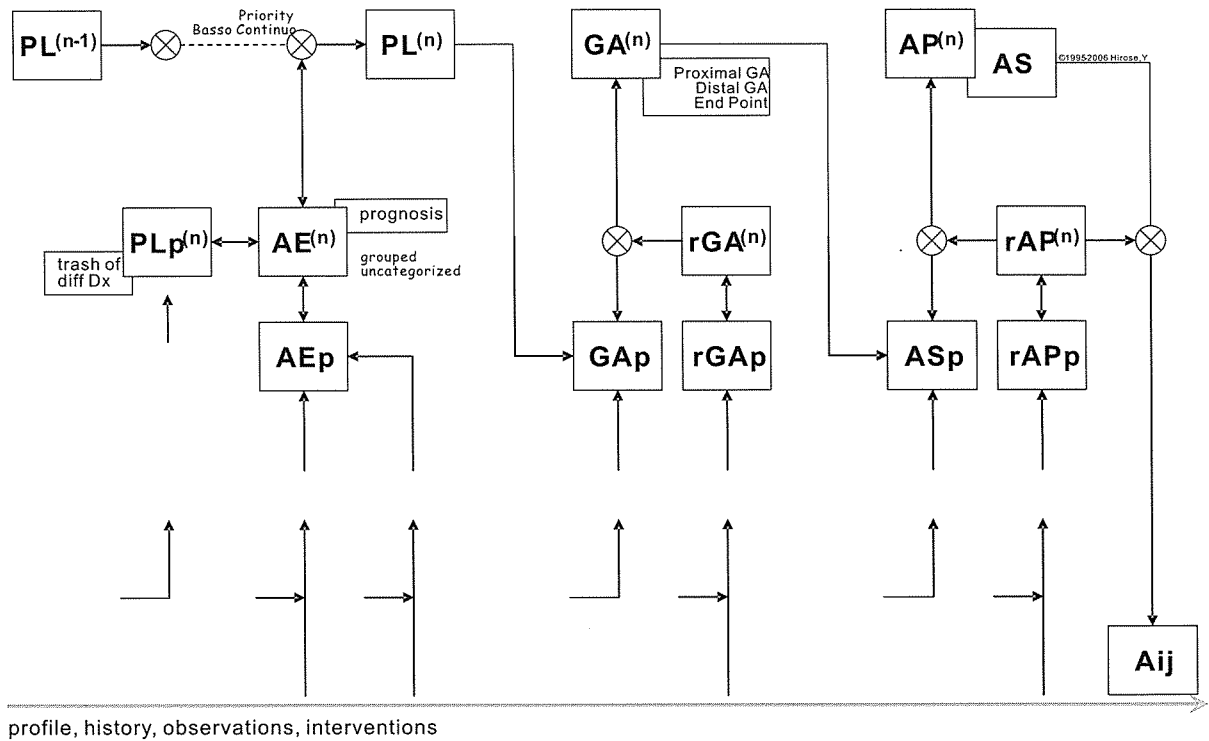


考慮すべき要素数は2を超えているので相当に複雑であり, 安易な扱いは危険である. よって思考空間全体を扱って, そのなかに配座することが妥当である.

### C. 5. 2. 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. 各要素の命名記法は, 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」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, いずれの序の Cycle に属しているのかを表している.

Cycle(n) の遷移において 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 AEp は現実世界から採取された AEij 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 AEp とを分離している.



PLに基づいてGA<sub>p</sub>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rGA<sub>p</sub>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<sub>p</sub>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<sub>p</sub>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. 5. 2. 8 意図実現過程モデル

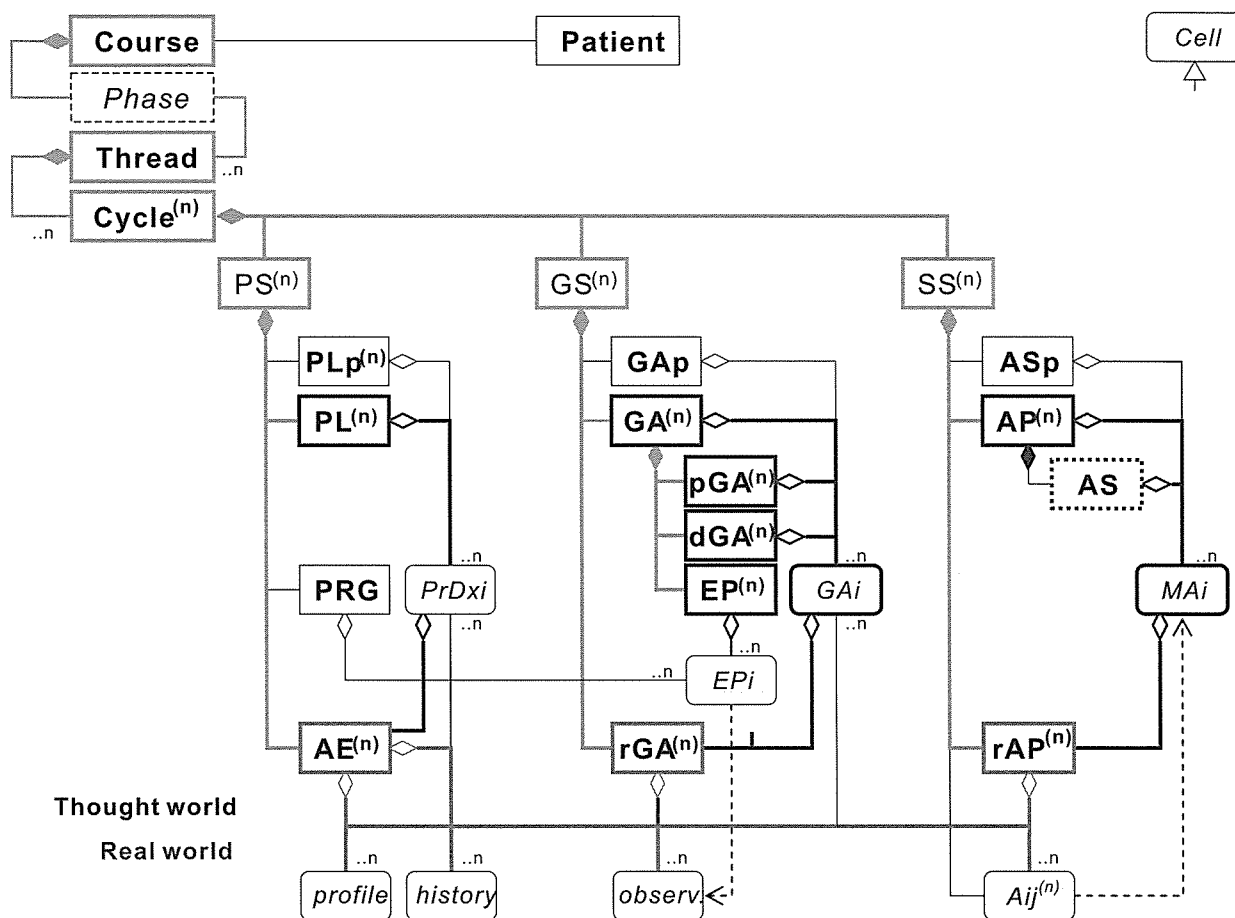
臨床思考過程モデルならびに診療過程モデルを融合して、意図実現過程モデルを得た。

CSXのinfoNode属性categoryやfamilyはThoughtProcessを頂点としてhierarchicalに決せられる:

ThoughtProcess.Course.(Phase).Thread.Cycle.[Space].[Block].[Cell]

つまりそのような観perspectiveを採るといふことでもある。

なお実業務では事前にPhaseを同定または確定することは困難な場合が多いので実装においては割愛することとなろう。Space, Block, Cellは@familyによって族が決定される。先述のCycleの形式化において用いたabbreviationでよかろうが、全名も選択しうる。



©2006 Hirose, Y

これら二つはいずれも container infoNode に適用されることも既に <C.4.2 観> において述べている。Cell infoNode については perspective を担う container infoNode とは独立である。ただ perspective を反映させたほうが良い場合もあるように思われる。

さてそれらの nodeCode に格納すべき値を網羅した医療関係コード体系については、現況、研究者らは聞かない。医学医療以外ならば語彙集を求めることも可能であるが、逆に大きすぎる。よって fcet.atst.csx.xsd に含めることとするが、これは言わば CSX dictionary と同義である。次に検討を要する事項は OOP 環境への反映戦略、そして arcScope と infoArc に関わる名辞の付与である。

なお上図は「理由づけ」という関係を省略して提示している。これは後述する。

### C. 5. 2. 9 関係と語彙

各 container infoNode は perspective を与えられており、かつ他の container infoNode などとの関連を有していることから、syntax は semantics の一端を担っていることになる。とはいえそれだけでは思考の挙動は見えずらい、もしくは不明確である。

CSX ではそのような地の挙動を arcScope を predicate として使うことで表現できる。その際に infoArc は、predicate における case を表現する役割を担うことになる。Case に関する属性値は前年度までに定義しているので arcScope について検討した。

主に container infoNode 間に適用すべき predicate を扱ったが、一部には Block を超えて Cell 間に適用すべき predicate も記している。なお、Block 内の Cell 間において適用