

障害調整生存年 (Disability-adjusted Life Year; DALY) の関数の特徴

分担研究者 杉 森 裕 樹  
 聖マリアンナ医科大学予防医学教室

研究要旨：健康負担を定量的に把握可能できる障害調整生存年 (DALY) の関数の特徴・問題点を整理した。単なる死亡者数や死亡率のみならず、死亡年齢と障害度をも加味した新しい複合指標であるが、本研究班の DALY による健康負担推定のシステム化、データベース化にあたっては、信頼できる統計データの確保、および算出ロジックの特徴をふまえた健康負担の評価が重要と考え

A. 研究目的

健康日本21は市町村または企業において、対象疾患を選定し、その健康負担を定量化して把握した上で、各対象疾患に対する改善目標を数値で提示していく健康施策である。本研究では、健康負担を死亡損失および障害損失として定量的に把握する指標として、近年注目されている障害調整生存年 (Disability-adjusted Life Year; DALY) を取り上げ、それを算出するために必要なDALYの関数の特徴・問題点を整理することを目的とした。

B. 研究方法

DALY の開発者である Murray CJL らの文献などを中心に検索し、DALY の関数の特徴と問題点を整理した。

C. 研究結果

DALY は、1992 年より世界銀行の要請をうけた Harvard 大学の Murray が、WHO と共同で行った Global Burden of Disease (GBD) study の中で開発した健康指標である。Quality adjusted life year (QALY)の方法論を拡大する形で、「損失生存年数 (死亡損失) (Years of Life Lost due to premature death: YLL) と「障害共存年数 (障害損失) (Years of Life lived with a Disability : YLD) から構成され、「死亡」と「障害」を1つの単位で測定した新しい複合健康指標である。国際比較可能であり、現在開発途上国をはじめとして多くの国々で政策立案のツールとして利用されつつある。GBD や NBD (National Burden of Disease)などは、世界または当該国における DALY の総和を表したものである。

DALY は、【1】似ている障害の健康負担は同等である ("like as like"の定理)、【2】年齢・性別以外に個人属性を健康負担量算定において考慮しない、の二つの基本原則が底辺にある。その上で(1)死亡損失、(2)生存年数の社会的価値 (年齢による重み付け)、(3)時間選好(time preference)、(4)障害 (非致命的健康度) による重み付けを手順として計算される。DALY の定積分関数の数式は

$$\int_{x=a}^{x=a+L} DCx \exp(-\beta x) \exp\{-r(x-a)\} dx$$

で示される。各項の意味するところは、 $Cx \exp(-\beta)$  は年齢による重み付けを表す。Murray らは  $\beta=0.04$  すなわち 25(=1/0.04)歳での1年間の価値が最も高いとしている。(図1)  $\exp\{-r(x-a)\}$ は時間選好を

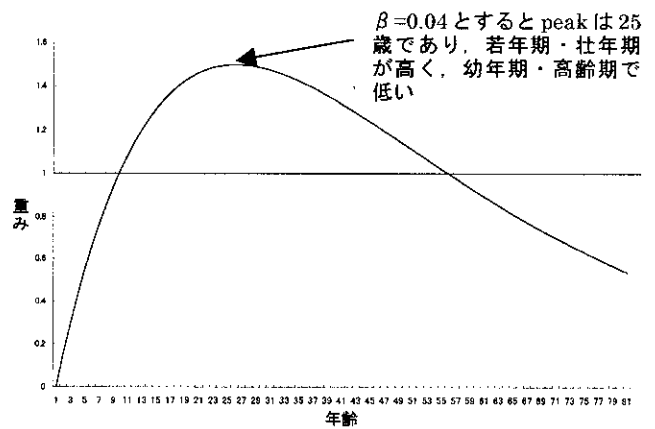


図1 DALY における年齢の重み付け

示すが、Murray らは 3%を用いている。a は障害発生時点または死亡時点、Lは障害の持続時間または死亡時点での期待余命を表す。Dは対象疾病による障害度による重み付けである。完全な健康状態D=0、死亡の場合D=1とし、評価系は専門家集団における数回の意見集約型アンケート方式(デルファイ法)に依拠する独自のものである。具体的には Person trade-off (PTO)法が GBD 第5班に採用されている。(表1)

DALY は理想的な平均寿命(男：80歳、女 82.5歳、Model life-table West Level 26:UN population Division)を用いており、理想寿命 (余命) からの質的乖離年数 (障害ウェイト×年齢価値×時間選好を持続時間の長さで積分) と言うこともできる。また DALY は QALY と異なり、大きいほど評価が悪く

表1. 健康状況・障害度による重みづけのD値の例

障害度	重み付け	該当する健康状態の例
1	0.00-0.02	顔面の白斑、低身長(2SD未満)
2	0.02-0.12	水溶性下痢、重度咽喉痛、重度貧血
3	0.12-0.24	橈骨骨折によるギプス固定、不妊、勃起不全、関節リウマチ
4	0.24-0.36	下肢切断(膝より下)、聴力完全喪失
5	0.36-0.50	直腸癌、軽度精神発達遅滞
6	0.50-0.70	うつ病性障害、視力完全喪失、大麻癖
7	0.70-1.00	活動性精神病、痴呆、重度脳障害、四肢麻痺

なるマイナスの価値である点が特徴である。

#### D. 考察

関数式に示されるように、DALYの測定には疾病の発生数、死亡数、発生または死亡の年齢、罹病期間、疾病障害の重さによって規定される。DALY測定において最も困難な点は、疾病の負担料すなわち疾病の発生数や罹病期間の推定である。我が国は統計データが比較的整備されているが、YLDの厳密な算定に耐えうる統計データが全ての疾病について存在するわけではない。本研究班のDALYによる健康負担推定のシステム化、データベース化にあたっては、まず代表的な疾病に対し、性別・年齢階級別（5歳幅など）で平均的な疾病負担量を推定していくプロセスが現実的である。また我が国でも簡便法を用いたDALYの推計が既に試みられており（池田ら1990、福田ら1993）、DALY算定のアルゴリズムを開発する上で参考になる。

統計データによってDALY結果は変動するため、データベース作成あたって、統計データ（死亡診断）の信頼性を確保する必要がある。人口動態統計の脳卒中死亡例は研究者の脳卒中登録例と3倍程度の開きがあることが知られている。また複合死因の取り扱いも難しい。脳卒中後遺症死亡における肺炎の寄与などがあげられるが、疾病間の比較を行う場合には無視できない例も生じる。

次にDALYの算出のロジックには、議論の余地が残されているので考察する。まず年齢による重み付け(Age-weighting function)は、図1に示したように、生涯を通じて一定ではなく、人の持っている年齢価値観が、上に凸の山なりの形であるとする。すなわち人が完全に健康で1年間生きることの価値は年齢によって異なることを意味する。経済性ではなく、生理的な能力を反映したものと解釈すべきであるが、年齢差別(ageism)ではないかという批判が考えうる。

YLLについては、死亡しない場合、あと何年生存すると期待できたかという平均余命を元に計算する。年齢を横軸、死亡損失を縦軸にとったグラフは幼児期が僅かに右上がりになるのを除くと基本的に右下がりの形をする。これは平均余命をもとにして死亡損失を算定する以上、右下がりになる。このことは高齢者の死亡に比べて若年者の死亡の方が損失が大きく、高齢者の死亡を防ぐより、若年者の死亡を防ぐ方が価値があることを意味する。倫理的に問題はないのであろうか。

倫理的には個人属性（性別、人種、宗教、学歴、社会経済性など）によって個人の価値（基本的人権）が異なるのは、差別や公平性の観点から許されない。しかしながらこの年齢という属性項目は性質が他の属性とやや異なる側面を持っている。人は生きていく限り順番にあらゆる年齢を経験する。現在若くても、いずれ将来は同じ条件が自分にも当てはまる。従って、この問題は単なる年齢による差別とはいえない。同時に、この論理はAge-weighting function

の批判にも援用できる。

一方、YLDは、DALY関数上ではYLLと同じ土俵にのっており、同じ一つの単位(DALY)で算定される。障害が死亡に換算してどのくらいの価値の損失に相当するかを判断する必要がある。(表1)これは、同じ年齢ならば障害度の低い人の死亡を阻止する方が障害度の大きい人を救命するよりも価値があることを意味する。これも倫理的に問題はないだろうか。この場合、障害の重みづけは、障害を持った当該者でない者が回答者となりうるもので、誰もが当該者となりうる上述の「年齢属性の特殊性」で説明するには無理がある。

元来障害による重み付けは、障害程度を死亡換算して、死亡や障害の健康負担を把握することで、医療資源の投入によりどれだけの健康改善が得られるか判断するものである。一つの考え方として、プラスの障害の重み付けを与えられると言うことは、逆に健康改善が可能であり、社会的・政策的価値があると解釈できる。GBD第5版のPTO法ではこの点が反映された内容となっている。

最後にYLDの課題について触れる。まず疾病による障害度Dは、期間経過に関わらず常に一定の健康損失をきたす点、またある障害がもたらす健康損失を死亡損失と比較した相対的な大きさは年齢に関わらず一定である点があげられる。またPTO法の障害度の重み付けに、大きく影響する因子に障害のcoping/adaptation(受容)の問題がある。受容前の重み付けを採用すれば予防とリハビリテーションは費用効果が高く算定されるが、受容後の重み付けを採用すると予防・リハビリは費用効果の観点からは魅力が少なく、一方、生存年数の増加の方が費用効果高い結論となる。YLDの特徴を考慮した総合的評価が重要である。

#### E. 結論

DALYを算出するために必要なDALYの関数の特徴・問題点を整理した。本研究班のDALYによる健康負担推定のシステム化、データベース化にあたっては、信頼できる統計データの確保、およびDALY算出ロジックの特徴をふまえた健康負担の評価が重要と考える。

#### F. 班友

飯田行恭 NTTサイバースペース研究所  
市村 匠 広島市立大学情報科学学部

#### 参考文献

1. Murray CJL, Acharya AK. Understanding DALYs. *J Health Economics*. 1997; 16:703-730.
2. Murray CJL. Quantifying the burden of disease: the technical basis for disability-adjusted life years. *Bulletin of the World Health Organization*. 1994; 72(3): 429-445.
3. Murray CJL, Lopez AD: *The Global Burden of Disease*. Harvard University Press. World

Health Organization, 1996.

4. 池田俊也, 田崎綱也. わが国における障害調整生存年 (DALY) -簡便法による推計の試み-. 医療と社会. 1998 ; 8(3) : 83-99.
5. 福田吉治, 長谷川敏彦, 八谷 寛, 田端航也. 日本の疾病負担と障害調整生存年 (DALY). 厚生指標. 1999 ; 46(4): 28-33.
6. Bobadilla JL, Coweley P. "Designing and implementing packages of essential health services" J Intern Development. 1998; 7: 543-554.

## 「健康日本 21 施策策定支援システムの評価項目とデータベースの必要事項」

研究協力者 飯田 恭行 NTT サイバースペース研究所

### 1. システム評価項目

健康日本 21 施策支援システムに関しては、その利用目的、利用方法等から特に以下の 3 項目が重要な評価項目となる。

- (1) 操作性
- (2) 応答性
- (3) 拡張性

(1) については、ユーザーがシステムに関する知識がなくても GUI 等により簡単に操作ができなければならない。例えば、施策等の選択は、プルダウンメニューやテキストボックス等を用いて簡易に入力できなくてはならない。また、選択した施策の疾病負担軽減効果等はグラフや図を用いて視覚的に表示し、ユーザーがすぐに結果を理解できることが望ましい。

(2) については、本システムはシミュレーションによる策定支援を想定しているため、結果を見て施策を変更するといった対話的な使い方が求められる。このため、結果を迅速に表示することができなければならない。

(3) については、医療統計データ、疾病負担量の計算アルゴリズム等は改良、変更されていくため、それらの変更に対するシステムの改造が必要最小限のプログラム改造で対応できるモジュール構造を持つソフト体系であることが望ましい。

### 2. DALY に基づく疾病負担量の計算とシステムへの実装法

施策支援システムの最も重要な機能は、人口構成、有病率等から各疾病に関する負担量の計算、

および選択した施策による疾病負担の軽減量の計算である。疾病負担量の計算方法として色々な方法が提案されているが、本検討では、詳細な医療経済の研究成果に基づいた DALY による計算方法を取りあげることとした。

DALY 関数は、障害度  $D$ 、年齢の重み付け関数  $W$ 、時間割引関数  $T$  の積で表現される関数であり、この関数を使った疾病負担量の計算は、例えば、年齢  $a$  歳のとき、ある疾病に罹患し、 $L$  年間、障害度  $D$  で過ごした患者の疾病負担量は、以下ようになる。

$$\int_a^{a+L} D \cdot W(x) \cdot T(x-a) dx$$

施策支援システムにおいてはこの関数を直接用いて疾病負担量を計算するのではなく、予め、疾病毎に上記の関数により計算された疾病負担量のデータベースとして実装するのが現実的な解であろう。即ち、図 1 のように性、年齢階級別に分類された患者の疾病負担量の平均値をデータベースとして持ち、疾病負担量の計算に用いればよい。

また、図 1 のような構造の表を使って疾病負担量を計算する方式は、早死損失等 DALY 以外の方法による疾病負担計算に対しても、表の中の数値を入れ替えるだけでまったく同様の処理で疾病負担量を計算できるので汎用性もある。

図1. 疾病負担表

性別	疾病	年齢 <sub>1</sub>	年齢 <sub>2</sub>	・	・	年齢 <sub>n</sub>
男性	高血圧	$d_{11}$	$d_{12}$	・	・	$d_{1n}$
	糖尿病	$d_{21}$	$d_{22}$	・	・	$d_{2n}$
	心疾患	$d_{31}$	$d_{32}$	・	・	$d_{3n}$
	・	・	・	・	・	・
	・	・	・	・	・	・
女性	高血圧	$d'_{11}$	$d'_{12}$	・	・	$d'_{1n}$
	糖尿病	$d'_{21}$	$d'_{22}$	・	・	$d'_{2n}$
	心疾患	$d'_{31}$	$d'_{32}$	・	・	$d'_{3n}$
	・	・	・	・	・	・
	・	・	・	・	・	・

### 3. データベースの要件

#### (1) データベースの種類と内容

施策策定支援システムとして以下の種類のデータベースと情報が必要である。

① 市町村データベース：性別、年齢階級別人口構成、有病率、死亡率

② 疾病負担データベース：性別、年齢階級別、疾病・事故別の疾病・障害負担量

③ 施策データベース：性別、年齢階級別、疾病・事故別に関する負担量減少等の施策毎の効果

疾病負担量計算フローとこれらのデータベースの関係を図2に示す。

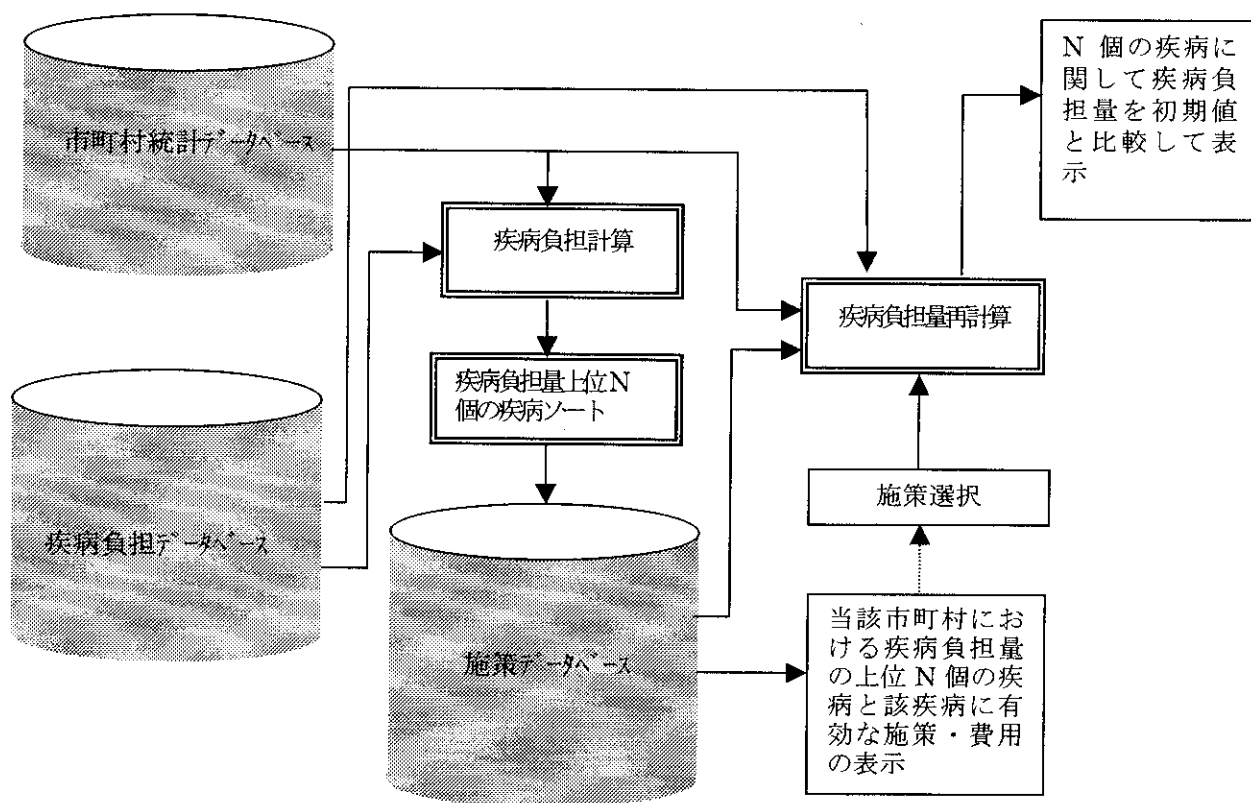


図2. 疾病負担量計算フローとデータベースの関係

## (2) データベースの構造

上記の各データベースは、統一的な構造を持ち、かつデータの変更、負担量の計算アルゴリズムの変更にも容易に対応できる汎用性、拡張性を持った構造でなければならない。具体的には、2で述べた、図1のような表形式が考えられる。各データベースの年齢階級は同一にしておくことが望ましく、具体的には5歳か10歳幅の年齢階級とするのが現実的であると思われる。

## (3) データベースのセキュリティ機能

権限を有しない者によってデータベースが変更されないようデータベースの保護機能を持たねばならない。即ち、システムのデータ管理メニューにはパスワード等を設定できるようにしておき、予め登録された名前とパスワードが一致しなければその操作を行えないようにする必要がある。

## Web-DB システムの構成

研究協力者 市村 匠 広島県立大学情報科学部 講師

### 1. システム構成図

システムの構成図は、図1のようになる。

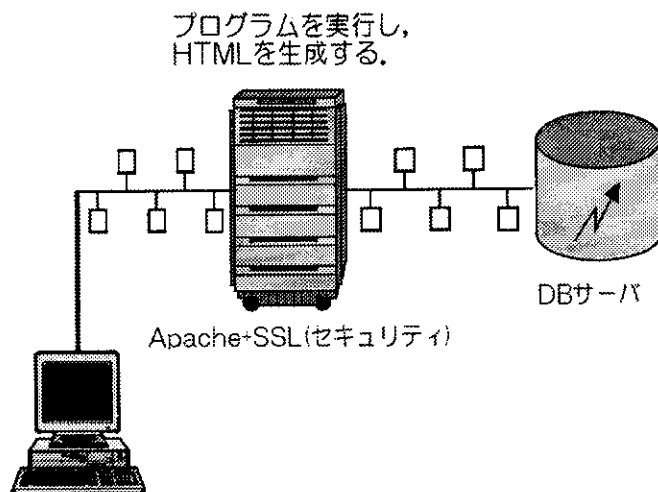


図1 システム概略図

システムは、サーバとクライアントに分かれる。クライアントとサーバ間は、LAN による通信が可能な状態にあるので、インターネットを通じた通信を行うことも可能で

ンを用意し、データベースにアクセスがある度にデータを更新するのではなく、キャッシュを作成し、過去数回アクセスがあったものを一時的にそこにデータを蓄えてお

表1 データベースソフト選定に関する事例

構築事例	オペレーティングシステム	データベースソフト
1)	Unix(Workstation)	Oracle
2)	Windows NT Server	SQL Server
◎3)	PC-UNIX	PostgreSQL

◎：推奨する組み合わせ

ある。サーバには、WWW サーバとデータベースサーバの2種類が存在する。マシンの処理速度の向上のために、タスクを分散し、サーバを2台のマシンで個々に動作させることが望ましいが、1台のマシンで2種類のサーバを動作することも可能である。設置すべき台数の基準は、データベースサーバで、操作するデータの容量による。莫大なデータを扱うのであれば、2台のマシ

いてもよいという考え方もある。今回は、データベースの性質上、様々なパラメタによりデータベース検索を行うアクセスが多いので、キャッシュを特に設定しない。

### 2. システムの選定

#### 2-1. データベースソフトの観点からの選定

ここでは、表1のように、3つの事例を

提案し、それらの費用、セキュリティ、システム構築の容易性などの観点から比較検討を行う。

事例1)では、ハードウェアおよびソフトウェアの購入費用が高価になってしまう。ここで言う、ハードウェアとは、Sun Enterprise Server のようなものを指す。(なお、Oracle 製品の値上げが検討されており、データベースソフト費用が5倍近

くに価格改訂されたという問題もある。)

事例2)では、ハードウェア、ソフトウェアともに Microsoft 社製品により構成されている。Microsoft 社製品、特にデータベースエンジンやネットワークセキュリティの上で、欠陥がいくつも報告されている。また、仕様で不明確な点が多く、プログラム開発は困難になる。構築に必要な費用の面からそれぞれの製品は単品では安価に思えるが、これ以外にも Internet Server な

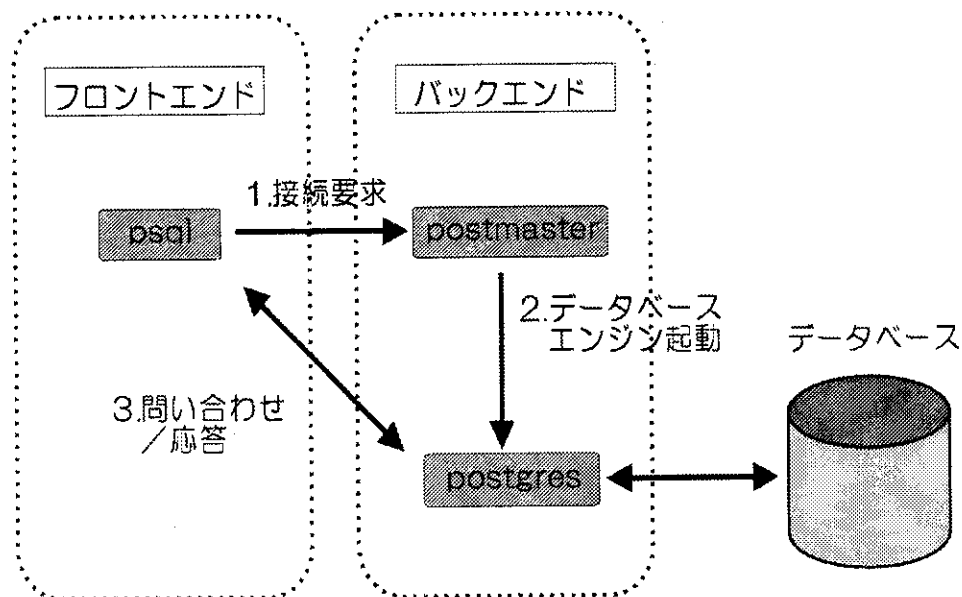


図2 PostgreSQLを構成するプロセス

どのソフトが必要となることも考慮すると、事例1)程ではないが、高価になるという問題が生じる。

事例3)では、ハードウェアの価格は、事例2)と変わらないが、ソフトウェアは基本的に無料となる。オペレーティングシステムはもちろんのこと、データベースソフトも無料で配布されている。セキュリティに関する観点からも問題がクリアされており、新しいセキュリティホールが見つかったとしても、その対応が非常に早い。また、昨今のコンピュータ技術の進歩に伴い、ハードウェアの性能を比較しても、パソコンとワークステーションにあまり差は見ら

れなくなっている。特に、浮動小数点を含む計算を行わない限り、速度の違いは見られない。また、耐久性の問題として、ワークステーションの部品がパソコンに使用されているものを用いていることもあるので、あまり違いは見られない。

以上のように、製品購入にかかる費用とセキュリティの問題、またシステム構築が容易な点から見ても、事例3)のハードウェアとデータベースソフトの選択が現時点では最も適している。

PostgreSQL サーバは、具体的に次のような特徴を示す。



図2は、PostgreSQL を構成するプロセスを示している。クライアントであるアプリケーションプログラムは、データベースサーバに接続し、ネットワークを通じて問い合わせを発行して、その結果を受け取る。PostgreSQL では、クライアント側を「フロントエンド」、サーバ側を「バックエンド」と言う。バックエンド側では、2つのプロセスに分かれており、一つは postmaster と呼ばれ、これは起動されると終了することなく、動き続けるデーモンプログラムである。postmaster は、「フロントエンドからの接続要求を受け付ける」、「要求を受け付けた後、データベースエンジンプロセスを起動する」、「データベースエンジン間で共有されるメモリ資源を管理する」の3つの働きをする。

図2において、psql は起動されるとホスト名、ポート番号などをパラメータとして postmaster に接続する。次に、postmaster は psql の指定したデータベースを引数に

して、データベースエンジンである potgres のプロセスを生成する。postgres が正常に起動されると、psql は直接 postgres と通信を行う。この時点で、postmaster は再び他のフロントエンドからのリクエストを待つ状態となる。

PostgreSQL では、共有メモリを使って複数の postgres プロセスが共有するバッファ領域を効率的に管理している。仮に、postgres プロセスが異常終了したとしても、postmaster が異常を検知し、バッファ領域をチェックし、必要なら初期化を行う。

さらに、PostgreSQL のフロントエンドでは、様々なアプリケーションとの通信が行われており、このプロトコルを libpq プロトコルと呼ぶ。PostgreSQL では、libpq プロトコルを意識せず、容易にアプリケーションを実装できるように、それぞれのライブラリが提供されている。

これら構造は、データベースを構築するための容易な設計と言えよう。

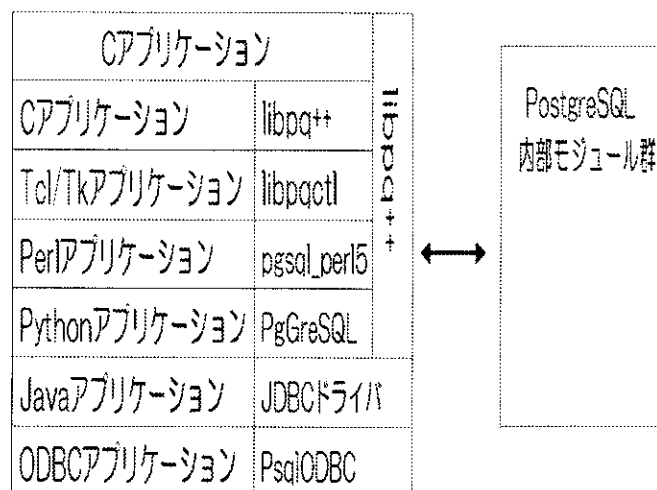


図3 PostgreSQL の通信

## 2-2. WWW Baed Database システムの選定

WWW Baed Database システムの構築手段を検討する際に、クライアントインタ

フェースの形態を考えなければならない。現時点では、HTML と Java アプレットが大きな選択肢として考えられる。

HTML 言語は、文書を広く公開するた

めに開発された言語である。しかも、単なるテキストファイルに限らず、文字の装飾やイメージを文章中に採り込んだり、さらに他の文書へのリンクも容易にしている。

また、Java も、HTML と同様に OS の違いに影響されることなく、同じイメージで実行できることを最大の目的としたプログラミング言語となっている。つまり、クライアント側で実行されるグラフィカルなアプリケーションを Java アプレットとして実行可能とし、サーバ側で実行されるアプリケーションは、これはクライアント側で実行するものと同じものであるが、Servlet などの形で実行可能としている。

両者の違いは、HTML が表現力をもった文書作成言語であるのに対し、Java はプログラムを作成する言語であることである。従って、Java を使ったシステムの方が、HTML を使うよりも、作成に要する時間を必要とし、システム全体として複雑

になってしまう。

しかしながら、HTML 文書で作成したインタフェースは、用意されている GUI(Graphical User Interface)も豊富ではなく、画面デザインの自由度も低くなってしまう、それに比べ Java アプレットであれば、クライアントとサーバ側で、同様のインタフェースを細部に至るまで実現することが可能である。

また、Java の開発は、既に JDK(Java Development Kit)1.1 から 1.2(通称 Java2)に移っている。Java のグラフィック部分の充実を図るため、Java の機能に Swing というコンポーネントが組み込まれている。図 4 に JDK1.2 の中核的パッケージの一部として、JFC(Java Foundation Classes)の概念を示す。

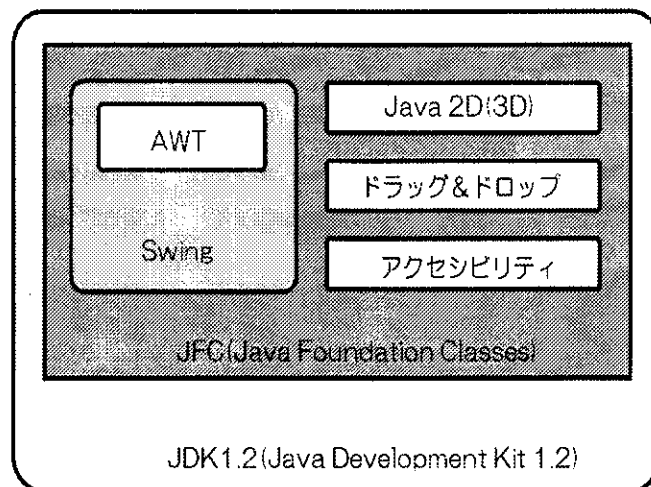


図 4 JFC と Swing

図中の語句は、次のような意味になる。

- ・アクセシビリティ 例え身障者のように、特殊なユーザインタフェースを必要とする場合に、音声による出力、画面の文字の拡大、入力のための音声

認識システムなど必要であり、これらのシステムが AWT のコンポーネントと通信するためのインタフェースを提供する。

- ・AWT(Abstract Window Toolkit) JFC

の重要な礎石であり、中核ライブラリの一部である。

- 2D(3D)グラフィック 高度な 2D もしくは 3D グラフィックと画像処理をサポートしている。
- ドラッグ&ドロップ ネイティブプラットフォームにも対応するドラッグ&ドロップにより、ユーザはネイティブアプリケーションと Java アプリケーションを同時に使いながら、両者の間でドラッグ&ドロップが可能となる。

このような理由により、Java アプレットによるインタフェースの構築を考えるが、これには次節で述べるような問題点もある。

### 3. 現在のシステムの問題点

現在、次の事柄が問題点になっている。

1. 図3において、提供されているライブラリの中で、日本語を考慮しているものは少ない。データベースに日本語を使用することにおいて、問題となるのは、その漢字コードである。一般的にクライアント側はシフト JIS であることが多いが、サーバ側では EUC を用いている。画面への表示は Java を通じて行われているので、使用されている漢字コードは、サーバ側で使用されているもの、すなわち EUC となる。従って、シフト JIS の漢字コードを EUC として表示することになり、いわゆる「文字化け」の状態が起こる。このことは、EUC コードを表示可能としたライブラリを独自に作成したことにより解消したが、クライアントのコンピュータのプログラムから直接サーバにデータを送受信するためには、予めこのライブラリをクライアント

にインストールしておくことが必要となる。ブラウザを用いて、WWW サーバにアクセスした場合には問題なく動作する。

2. Java の Swing のライブラリ(jar ファイル)は、現在のブラウザに登録されていない。Netscapeなどに Java というディレクトリ(フォルダ)が存在し、そこに Java のクラスが保存されているのが分かるが、ここに swing.jar というファイルは見当たらない。そこで、予め swing.jar をインストールしておく必要があるが、これは JDK1.2 には予め含まれているので、JDK1.2 をインストールしておけばよい。さらに、クラス環境変数を設定しなければならないので、例えば Windows9x の場合、autoexec.bat ファイルを変更しなければならない。このように、若干のインストール作業をクライアント側で予め行わなければならないことが問題点の1つである。