

# 粒子線治療装置における ソフトウェア開発での 共通フレームワーク開発

馬淵 英之

三菱電機マイコン機器ソフトウェア（株）

E-mail:hmabuchi@mms.co.jp

## はじめに

粒子線治療装置とは、がん治療装置の1つである。がん治療は人命にかかわるので、その治療装置には高い安定性と信頼性が求められる。粒子線治療装置は、今後幾多の医療機関に販売されるが、そのときの保守は少数の人員で行えることが求められている。さらに、納入後に医師の要望、機器の変更、治療方法の変更などにより、ソフトウェアの機能拡張や変更を行う必要性が出てくることが予想される。したがって開発時に容易に機能拡張を行えるようにしておく必要がある。

近年、ソフトウェア業界では、オブジェクト指向が1つのキーワードになっており、その中で、保守性、生産性、再利用性を高める技術として、ソフトウェアのコンポーネント化、フレームワーク化が注目されるようになってきた。今回開発する粒子線治療装置ソフトウェア開発は2世代目であるが、第1世代目の装置は、図-1に述べる計算機単位で個別に開発を行う従来手法に基づいたものであったので、共通的な機能の追加といったシステムの拡張性の面で、インタフェースの共通化や共通的な内部仕様の関連性をいかに高めるかといったところに改善、工夫の課題があった。今回の開発では、計算機間でソフトウェアのアーキテクチャを共通化し、似て非なるソフトウェアの作成をできるだ

け避け、不具合の分散化の阻止と開発後の保守性、拡張性を高めるために、オブジェクト指向技術を用いて共通のフレームワーク（ソフトウェア再利用のための枠組み）を構築してきた。

本稿は、今後、オブジェクト指向を用いてフレームワークを構築しようとしている方の参考になればと考え、粒子線治療装置のソフトウェア開発では、どのようなフレームワークを開発したかを述べる。なお、表記法は、UML (Unified Modeling Language) <sup>1)</sup> を使用している。

## 粒子線治療装置のソフトウェアの概要

粒子線治療装置は、加速器を用いて粒子を加速し、その粒子線を用いてがん細胞を破壊する装置である。そのソフトウェアの主な役割は、照射管理、治療スケジュール管理、粒子線を治療室まで導き入れるための機器の制御とその管理、治療台の位置の決定である。このシステムは、HP-UX10.20<sup>☆1</sup>（一部Windows NT<sup>☆2</sup>）上で動作するシステムであり、図-1で述べるような計算機で構成され、ネットワークを介して接続される。また、

☆1 HP-UX10.20は、米Hewlett Packard社の登録商標です。

☆2 Windows NTは、米Microsoft Corporationの登録商標です。

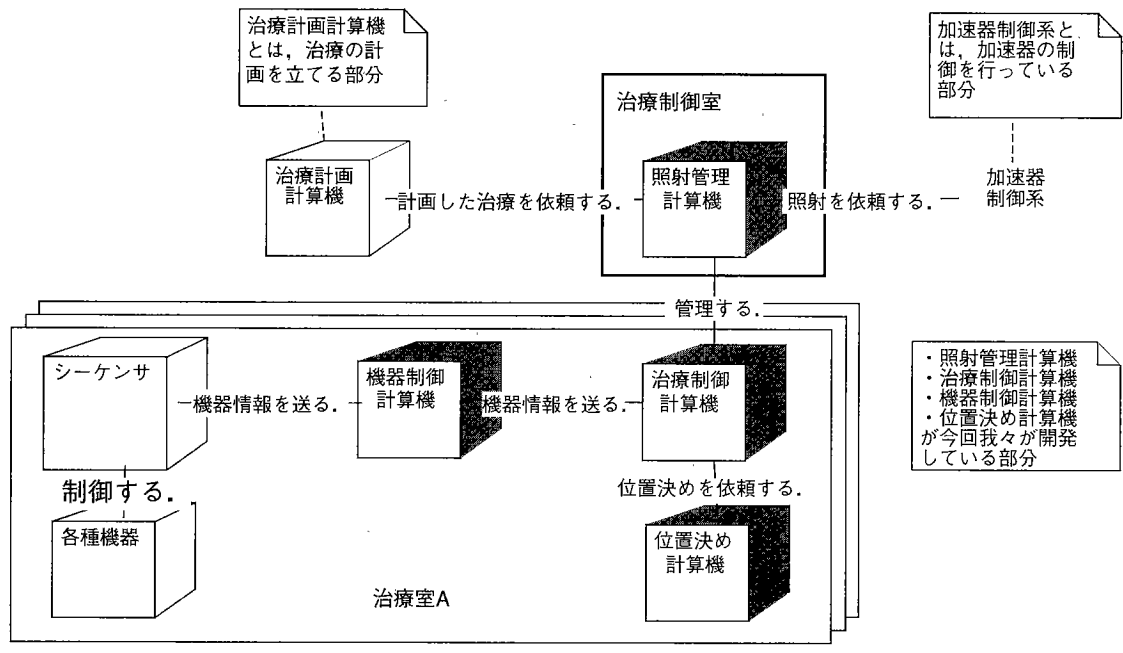


図-1 粒子線治療装置のシステム構成図

	照射管理計算機	治療制御計算機	機器制御計算機	位置決め計算機
OS	HP-UX10.20	HP-UX10.20	Windows NT4.0	HP-UX10.20
言語	HP C++ <sup>☆3</sup>	HP C++	Visual C++ <sup>☆4</sup>	HP C++
画面構築ツール	UIM/X Ver3.0 <sup>☆5</sup>	UIM/X Ver3.0	Visual C++	UIM/X Ver3.0
分析・設計ツール	Rational Rose/C++ Ver4.0J <sup>☆6</sup>	Rational Rose/C++ Ver4.0J	Rational Rose/C++ Ver4.0J	Rational Rose/C++ Ver4.0J

表-1 開発環境

それぞれの計算機は、GUI (Graphical User Interface) を持っている。

治療患部がセッティングされるように治療台位置を計算する。

### システム構成

システムは、次のような計算機で構成される (図-1参照)。

照射管理計算機：システムに1つだけあり、粒子線の照射管理と治療スケジュール管理を行う。

治療制御計算機：治療室ごとにあり、治療の際に機器の制御を行う。

機器制御計算機：ビームラインごとにあり、実際の機器につながっているシーケンサとの間で機器のデータのやり取りを行う。

位置決め計算機：治療室ごとにあり、治療計画通りに

### 開発環境

開発環境を表-1に示す。

## 粒子線共通フレームワーク概要

粒子線治療装置のソフトウェア開発では、3層以上の継承構造をとるフレームワークを構築した。継承構造が複雑にならないように単継承とし、第1層では、実行時にクラスの型情報を識別するために識別子とオブジェクト複製のためのメソッドを用意し、第2層では、画面、モデル、通信を分離したアーキテクチャを採用した (図-2参照)。そして第3層では、粒子線治療装置のソフトウェアで共通に使う可能性のある機能をフレー

☆3 HP C++ は、米Hewlett Packard社の登録商標です。  
 ☆4 Visual C++は、米Microsoft Corporationの登録商標です。  
 ☆5 UIM/XはVisual Edge Software社の登録商標です。  
 ☆6 Rational Rose/C++ Ver4.0Jは、米国Rational Software Corporationの登録商標です。

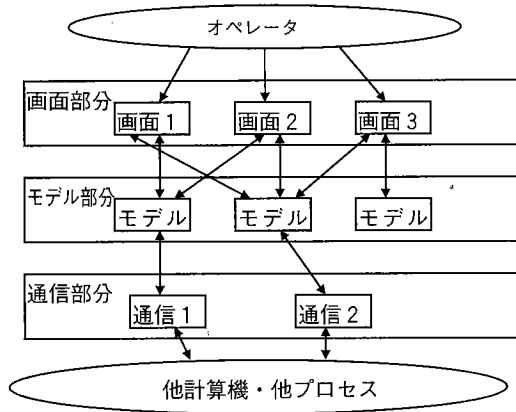


図-2 粒子線治療装置のソフトウェアの3階層アーキテクチャ概念図

計算機	機能
照射管理計算機	<ul style="list-style-type: none"> <li>全体の機器の管理と機器の制御値表示</li> <li>それぞれの計算機の管理</li> <li>患者の一覧表示</li> <li>医師・技師の一覧表示</li> </ul>
治療制御計算機	<ul style="list-style-type: none"> <li>機器の制御とその制御値表示</li> <li>機器の状態一覧表示</li> <li>位置決め制御</li> <li>照射ビームの制御</li> <li>機器制御との間の機器情報の通信</li> <li>画面上で矩形表示したオブジェクトに対してマウスを用いて動かすことにより、機器のデータ設定を行う</li> </ul>
機器制御計算機	<ul style="list-style-type: none"> <li>シーケンサとの機器のデータのやり取り</li> <li>治療制御との機器データのやり取り</li> <li>機器の値表示</li> </ul>
位置決め計算機	<ul style="list-style-type: none"> <li>画像処理を用いた治療台位置決め</li> <li>患者の画像データを選択するための一覧表示</li> <li>治療台位置の決定とその値の治療制御への通信</li> </ul>

表-2 それぞれの計算機の主な機能

ムワーク化することにした。次節以降で、第3層以下について詳しく述べる。

### 第3層以下

第3層以下では、似て非なるソフトウェアの作成を極力避けるために、共通的に作れる個所を見つけフレームワーク化する必要がある。そこでまず、各計算機で共通的な作りができそうな部分を見つけるために各計算機の大まかな機能を抽出する。抽出した機能は、表-2の通りである。

これらから、粒子線治療装置のソフトウェアでは、グラフィック処理(表-2の下線部)、機器のデータの制御(以下単に機器制御部分とする。表-2の2重下線部)、何らかのデータの一覧表示部分(以下単に一覧表示部分とする。表-2の破線部)が同じようなソフトウェアになる可能性がある。したがって、それらをそれぞれフレームワーク化することにした。また、そのほかに、アプリケーションでの静的なオブジェクトを生成し、関連付けするための構築クラスも共通的な作りのできる(以下単に構築部分とする)フレームワーク化した。以降では、構築部分、グラフィック処理部分、機器制御部分、一覧表示部分の順に、それぞれの抽象クラスについて述べる。

#### (1) 構築部分

アプリケーションには、必ず静的にオブジェクトが存在し、それぞれのオブジェクトは、それらのポインタを用いて関連付けられている。したがって、アプリケーションを構築するにあたって、まず、静的に存在

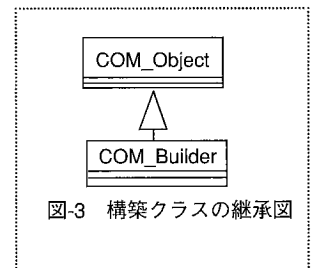


図-3 構築クラスの継承図

するオブジェクトを生成し、次に、そのオブジェクトに関連付ける必要があるオブジェクトのポインタを登録する。このような作業を行う個所を複雑にすると保守の際に分かりにくい。したがって、このような生成と関連付けを行う抽象クラスとして今回COM\_Builderを設けた。その中では、画面、モデル、通信部分を構築するメソッドとそれらに関連付けるメソッドをそれぞれ純粋仮想関数で用意することによって統一的な作りを実現できるようにした(図-3)。

#### (2) グラフィック処理部分

粒子線治療装置のソフトウェア開発で共通的に必要とされるグラフィック処理として画像描画、マウスを使用した図形の描画などの機能があるので、それらの機能を持つグラフィックライブラリを開発した。そのクラス図(代表的なクラスのみを記述)を図-4に示す。実際にアプリケーションにこのライブラリを適用する場合、図-4のクラスを必要に応じて継承し、アプリケーション独自の機能を追加したクラスを作成する。

#### (3) 機器制御部分

粒子線治療装置では、治療に必要な各種の機器の動作制御を行う必要がある。また、複数の計算機で、そ

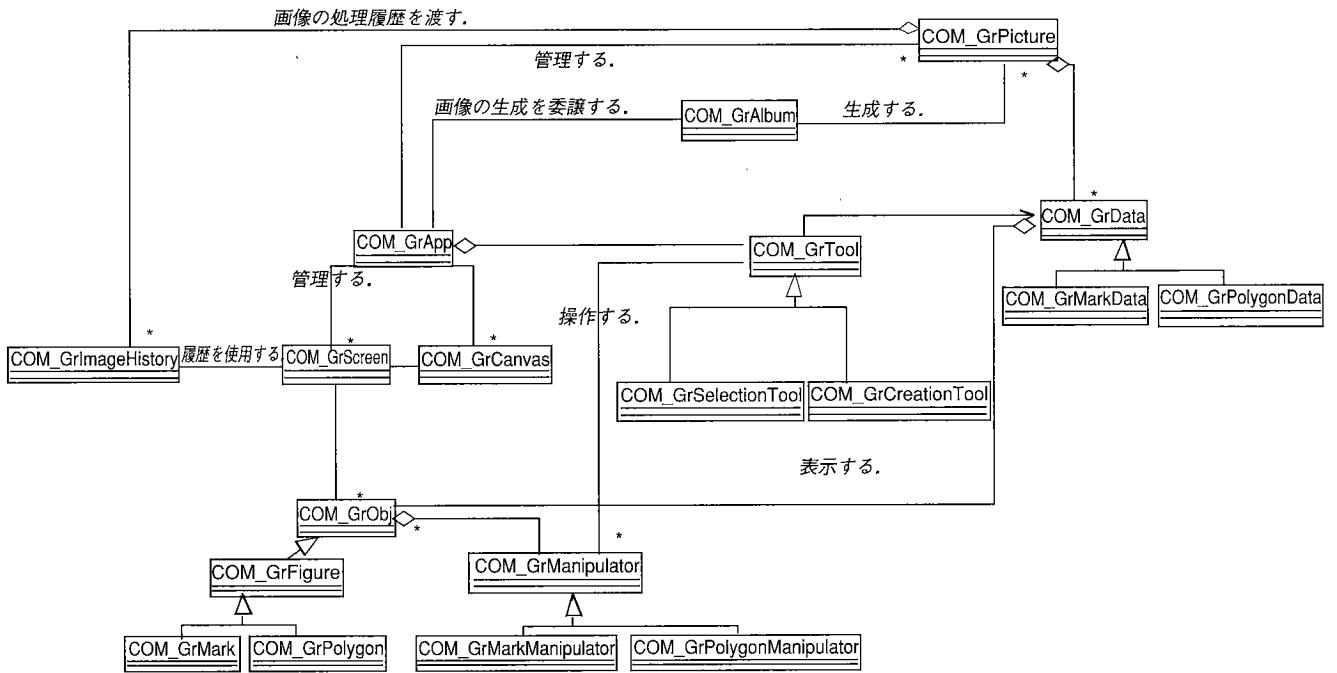


図-4 グラフィックライブラリのクラス図

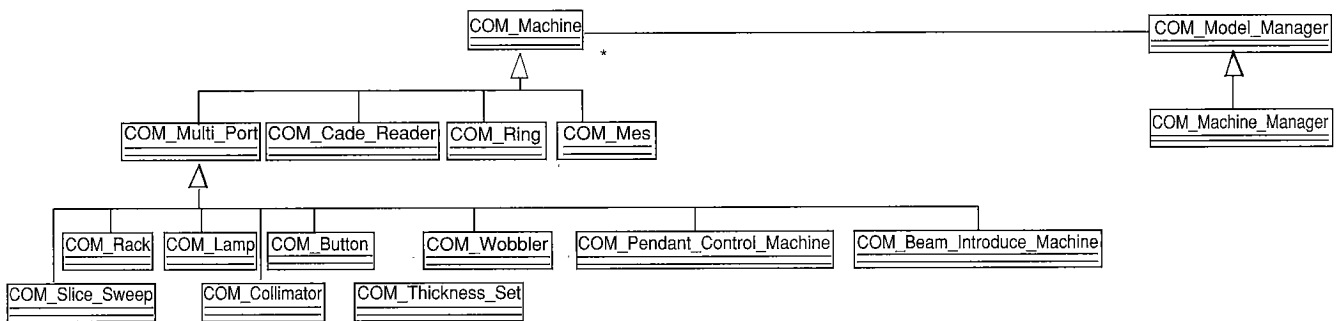


図-5 機器関係のクラス図

の様子をモニタ、または、数値を設定する必要がある。そこで、各機器ごとに、値の監視と値の設定を行うクラスを作成した。しかし、機器ごとに設定するデータの種類が違うので、データごとにそれに対応したメソッドを作ると非常に多くのメソッドを用意しなければならなくなる。よって、共通のメソッドを用いて、引数を変えることで、数値の取得と設定をすることにした。そのために、それらの機器クラスの最上位のクラスとしてCOM\_Machineを設け、このクラスにそのメソッドを用意した。このクラスを用いることで、それぞれの機器に対して、同じ方法で、数値の取得と設定ができる。したがって、粒子線治療装置で使用する機器は、

必ずこの機器クラスを継承しなければならない。また、各機器は、それ自身で値の設定、送信ができるものがあるが、他の機器と連動して動作するものもある。しかし、それらの機器同士を直接関連付けてしまうと、機器の独立性が損なわれる。したがって、それらを制御する抽象クラスとしてCOM\_Model\_Managerを導入し、COM\_Machineと抽象クラス同士での関連付けを持たすようにした(図-5)。実際の機器を制御する場合、このCOM\_Model\_Managerを継承してそれぞれの計算機独自の制御方法を実装したクラスを作り、そのクラスに必要な機器のクラスを関連付ける。

(4) 一覧表示部分

アプリケーションにおいては、さまざまなデータをリスト表示してみることが多い。また、そのリスト表示した画面においては、ソート等の機能と選択機能が付いているものが多い。したがって、そのようなリスト表示用のクラスとして、リスト表示するデータを扱う部分とその結果をその後の処理に反映するクラスとして、COM\_TableクラスとCOM\_TableManagerクラスを作成した。また、実際には、粒子線治療装置のソフトウェアにおいては、X Window System<sup>☆7</sup>上で、そのリスト結果を表示する必要があるため、X Window Systemのライブラリを用いて、COM\_Tableが編集したデータをリスト表示するクラスとしてCOM\_TableXを設けた(図-6参照)。

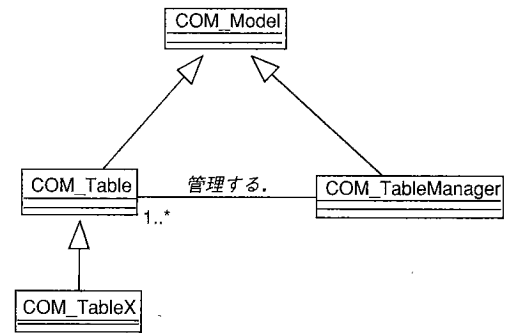


図-6 リスト表示関係のクラス図

### フレームワークの粒子線治療装置特化部分

今回開発したフレームワークで粒子線治療装置に特化した箇所は、機器制御部分と通信の部分である。機器制御部分の特化箇所は、計算機間での機器のデータの通信を行う場合に、医用専用形式のファイルを参照する構造になっている部分である。通信部分については、ソケット通信を用いているのでLANのみを想定している部分である。ただし、上記の特化部分のフレームワークの設計自体は、機器を制御する計算機であれば同じような設計が使えると考える。

## フレームワーク開発について

今回オブジェクト指向を用いたフレームワークの開発を通して得たことを以下に記述する。(1)～(3)までは、フレームワーク全般について述べ、(4)以降は、フレームワークの第3層以下について述べる。

#### (1) ドキュメントについて

フレームワークに関するドキュメントは、オブジェクト図、シーケンス図、各クラスのメソッドとメンバの説明のみではなく次の項目を入れてドキュメント化する必要がある。

- フレームワークの使用方法
- フレームワークの作成の意図
- フレームワークの拡張の方法
- フレームワークのサンプルコード

これらをドキュメントに残すことにより、実際に継承して使う場合、プログラムソースコードを直接追

<sup>☆7</sup> X Window Systemは、The Massachusetts Institute Of Technologyの商標です。

かける箇所が削減できるので、作業効率がよくなる。

#### (2) 継承について

フレームワークを設計すると継承階層が深く(4層以上)なりがちだが、アプリケーションの実装の際に、足かせとなりかねない。また、継承が深いとファイルをまたがった作りになるので、プログラムソースコードを追う箇所が複雑化し、デバッグ効率の低下や潜在的な不具合を作り込む原因となることがある。したがって、継承は、共通アーキテクチャを決める1, 2層のような継承部分を除くと3層程度が妥当である。

#### (3) フレームワークの作成時期

我々は、アプリケーションの開発における、設計の中ほどでフレームワークの構築を始めたが、設計の初期段階から、フレームワーク開発チームを設けて、ホットスポット、フローズンスポットを考慮に入れ、アプリケーション開発より常に先行して構築を始めた方が効率的に開発できたのではないかと考える。また、今回のフレームワークは、数社で開発したが、フレームワークを構築する場合、コードの均一化、メンテナンスの効率化、リファクタリングの容易さを考えると、キーとなる会社1社の数名で開発し他の協力会社は、そのフレームワークを使用するような開発スタイルの方がよいと考える。

#### (4) グラフィック部分

グラフィックフレームワークでは、描画スピードで、アプリケーションのイメージが変わってしまう。しかし、その描画したいタイミングは、アプリケーションごとに違うので、フレームワークでは、描画の大枠の機構だけ与えておいて、アプリケーションごとに描画のタイミングを与える方がよい。

#### (5) 機器制御部分

粒子線治療装置では、同種の機器であれば、2つ以上機器があつたとしても1クラス1インスタンスで管理で

メリット	<ul style="list-style-type: none"> <li>・デザインパターンは、画像処理に関する記述が多い。したがって、画像処理をするアプリケーションを作成する場合には、使用できるパターンが多いので、設計時の時間が短縮できる。</li> <li>・プロジェクトの開発者がデザインパターンを理解し始めると、共通の名称でクラスの構成の理解ができるようになり、議論がしやすくなる。</li> <li>・C++を用いたオブジェクト指向設計の一般的な再利用方法（抽象化）の技法が理解できる。</li> </ul>
デメリット	<ul style="list-style-type: none"> <li>・あらゆる設計個所にデザインパターンを当てはめようという傾向が出てくる。</li> <li>・デザインパターンは、総ページ400ページ以上もあるので、理解するのに時間が必要である。</li> <li>・複数人で開発する場合には、1人だけ理解しても、その人のみ理解できる設計になる可能性がある。</li> </ul>

表-3 デザインパターンのメリット・デメリット

きるだろうと考え、機器のクラスを抽出した。しかし、同一のクラスの内部で、2つの機器の制御を行うようにするために、構造が非常に複雑になった。このような結果にならないために、各機器の種類別に機器のクラスを対応付け、2カ所以上機器がある場合には、インスタンスをそれぞれの機器に対して生成し、その機器を管理するクラスを設ける必要がある。

#### (6) 一覧表示部分

一覧表示にもいろいろな形態のものがあるが、それにすべて対応しようとしたために、作りが複雑になった。できればフレームワークでは、単純な機能を提供し、アプリケーションごとにカスタマイズできるようにした方がよかった。

## 考察

以下では、今回粒子線治療装置のソフトウェア開発で、オブジェクト指向を適用した経験からオブジェクト指向開発についての全般的な観点で考察する。

### オブジェクト指向開発の最初の壁

オブジェクト指向開発での最初の壁は、一般的にいわれているとおり、オブジェクトの抽出である。これがうまくいかないと後の作業が全然進まなくなる。粒子線治療装置のソフトウェアでもオブジェクトの抽出ができるようになるまでに時間がかかった。当初、OMT法<sup>3)</sup>でいわれている要求仕様書の名詞からオブジェクトを抽出する方法をとったが、OMT法では、どのようにしてその名詞をオブジェクトと属性に分けるかを規定していないのでうまくオブジェクトの抽出ができなかった。そこで、粒子線治療装置のソフトウェアでは、コンピュータがない時代にこのようなシステムを構築したら、どのようにシステムを運営するかをシミュレーションすることによってオブジェクトを抽出した。

この方法によって、OMT法の抽出方法より格段にうまくオブジェクトを抽出できた。

### オブジェクト指向の拡張性、保守性

これは、分析・設計次第である。オブジェクト指向で分析・設計・実装をしたからといって、拡張性、保守性がよくなるとは限らない。特に、初めてオブジェクト指向を適用する場合には、それが顕著になると考える。確かに、オブジェクト指向には、拡張性、保守性を高める仕組みがあるが、それをすぐに使えるようにはなかなかならない。しかし、オブジェクト指向には、デザインパターン<sup>2)</sup>のように拡張性、保守性を可能にする設計方法が提唱されているので、これをうまく使うことによって、拡張性があり、保守性に優れたアプリケーションになる。実際、粒子線治療装置のソフトウェア開発でもデザインパターンを参考にして拡張性を持たせた設計個所が多くある。また、オブジェクト指向をスムーズに導入するには、オブジェクト指向経験者を、そのプロジェクトに参加させ、設計者全員レベルアップを図ることが必要である。

### デザインパターン

前節でも述べたように、粒子線のフレームワークでは、デザインパターンを利用した。そこから得た経験上、デザインパターンには、表-3のようなメリット・デメリットがある。

### 再利用性

粒子線治療装置ソフトウェアのうち、位置決め計算機の透視画像プロセスに焦点を当ててオブジェクト指向の画面とモデル部分を分離した設計方法の再利用性を考察してみる。位置決め計算機は、他の計算機に先駆けて、放射線医学総合研究所のHIMAC (Heavy Ion Medical Accelerator in Chibaの略) シミュレーション室に1998年11月に納め、その後、他の計算機とともにA, B, Cの3治療室に2000年4月に納めた。2000年4月にA, B, Cの3治療

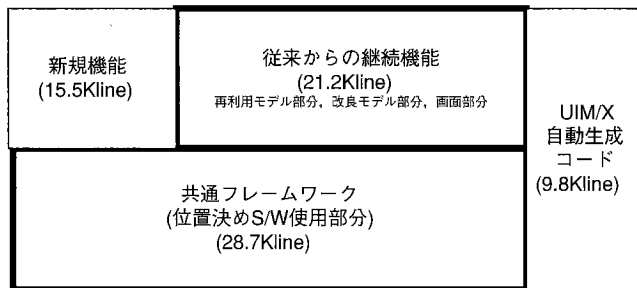


図-7 位置決め計算機 透視画像プロセスソフトウェア構成

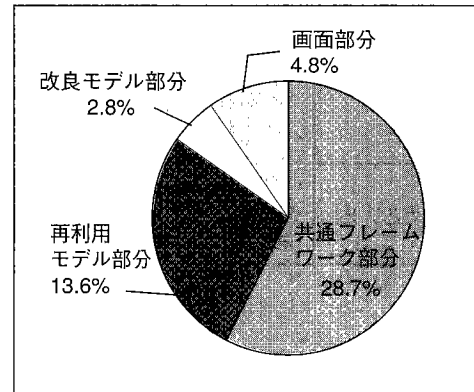


図-8 シミュレーション室と同等機能実現箇所

$$\text{再利用率} = \frac{\text{共通フレームワーク部分} + \text{再利用モデル部分} + \text{改良モデル部分} + \text{画面部分}}{\text{共通フレームワーク部分} + \text{再利用モデル部分}} = \frac{49.9\text{Kライン}}{42.3\text{Kライン}}$$

注) ライン数は、コメントを除く実ライン数である。

室に適用するにあたり、シミュレーション室に納めた位置決め計算機の透視画像プロセスに対して、運用面からの改良・改善要望があった。その大部分は、画面構成であったので、2000年度に3治療室に納品するにあたっては、一から画面構成を設計し直した。再設計した位置決め計算機の透視画像プロセスのソフトウェアの構成は、図-7のようになっており、そのうち、シミュレーション室とほぼ同等の機能を実現する個所が、図-7の黒枠の部分(UIM/Xの自動生成コードは、入れない)であり、その部分のソースコードの詳細は、図-8(数字は、Kライン数)である。図-8の画面部分と改良モデル部分が新たに作り直した個所であり、共通フレームワーク部分と再利用モデル部分は、若干の手直しはあったがほぼそのままソースコードを再利用した。したがって、共通フレームワークを含めたソースコードの再利用率は、84.8%(上記計算式使用)になった。

また、新規機能についても、共通フレームワークの上に構築しているので、共通フレームワークの再利用性も高まっている。この結果より、画面と内部モデルを分離するオブジェクト指向の設計方法は、再利用率が高いし、共通フレームワークの再利用性も高いものといえる。

## 今後の課題

今回、オブジェクト指向で共通的なフレームワークを構築したが、各アプリケーションごとに、必要なフレームワークが違うので、アプリケーションとフレームワークのバージョンの関係が多対多になり、管理が複雑化している。したがって、拡張性と保守性とバージョン管理の絡みを、そのようなフレームワークとアプリケーションの多対多の関係がある中で、どのように簡素にできるかが今後の課題である。

**謝辞** 本稿を執筆するにあたり、開発の機会を与えていただき、ご指導を賜ったHIMAGの諸先生方、ならびに三菱電機(株)通信機製作所の関係各位に深く感謝いたします。また、粒子線治療装置の開発に携わった弊社関係各位に感謝いたします。

### 参考文献

- 1) Booch, G.: The Unified Modeling Language User Guide. [オージス総研オブジェクト技術ソリューション事業部訳: UMLユーザーガイド.]
- 2) Gamma, E., Helm, R., Johnson, R. and Vlissides, J.: Design Patterns. [本位田真一, 吉田和樹監訳: オブジェクト指向における再利用のためのデザインパターン, ソフトバンク(1995).]
- 3) Rumbaugh, J.: Object-Oriented Modeling and Design. [羽生田栄一監訳: オブジェクト指向方法論 OMT (モデル化と設計).]

(平成12年9月7日受付)

