

造船 CIMS パイロットモデル・プロダクト モデルシステムの開発

正員 田 畑 光 敏*

The Product Model System of CIM for Shipbuilding

by Mitsutoshi Tabata, *Member*

Summary

Ship & Ocean Foundation has developed the pilot model system of CIM (Computer Integrated Manufacturing System) for Shipbuilding by March, 1992. One of the most special feature of CIM for Shipbuilding is the product model which keeps the information about products (ships) independently of the application systems. The product model must satisfy "Integration", "Consistency" and "Universality".

"Integration" means that the product model must serve various application systems, not limited to a specific application system. For this purpose, the structure of information had better be similar to the structure of the product in real world.

"Consistency" means that the product model system can manage the data to keep consistency of the information. For this purpose, product model must have the information about relationship among the other information.

"Universality" means that the variety of products (kind of ships, pattern of construction, etc.) has little influence on the structure of the product model system or the application systems. For this purpose, the product model system must be constructed from functional units, and the communication protocol between the product model system and the application systems must be independent of the structure of the product model system.

By developing and using the pilot model of product model system (PPMS), it is clarified that the concepts of "Entity-Relationship Model", "Object-Oriented" and "Model Manipulation Language" have effect on the product model system to satisfy the previous items.

On the other hand, for developing the practical use system, the further study of some items, which includes the methodology for developing and compatibility between the hi-level functions and the performance, are needed.

1. 結 言

シップ・アンド・オーシャン財団では平成元年度より造船 CIMS (Computer Integrated Manufacturing System) パイロットモデルの研究開発を行ってきたが、平成 4 年 3 月に開発を完了した¹⁾。従来の CAD/CAM システムなどと

比べて造船 CIMS において最も特徴的なのは、製品に関する情報を各アプリケーションから独立して保持するプロダクトモデルの存在であるが、本研究開発において将来の実用プロダクトモデルシステムのキーとなる部分の検証を可能とするパイロットモデル・プロダクトモデルシステム (PPMS) を開発した。

本論文ではプロダクトモデルの概念、およびそれを実現するための基本的な考え方について述べた後に今回開発した PPMS について解説し、また開発プロセスについても述べる。最後に得られた知見及び今後の課題などについて述べる。

* (財)シップ・アンド・オーシャン財団造船業 CIM フレームモデルの開発研究委員会造船所チーム
／石川島播磨重工業(株)

原稿受理 平成 4 年 7 月 10 日

秋季講演会において講演 平成 4 年 11 月 9, 10 日

2. プロダクトモデルの概念

2.1 プロダクトモデルの定義

プロダクトモデルは今後の設計生産システムにおいて不可欠の概念として多くの研究がなされており、造船分野においても例外ではない²⁾。それに伴いプロダクトモデルの定義・性質についても多種の提案がなされているが、造船 CIMS パイロットモデルではプロダクトモデルを「特定の製品に付随する情報のうち、造船 CIMS の複数のアプリケーションシステムにわたって利用されるもの、あるいは残しておく必要があるものを 2.2 に述べるような要件を満たす形で格納したもの」と定義付けている。またプロダクトモデルシステムとは「プロダクトモデルをシステムとして実現し、さらにアプリケーションシステムから利用可能としたシステム」である。

2.2 プロダクトモデルの要件

プロダクトモデルの要件として特に以下の 3 つの特質が重要である。

2.2.1 統合化

統合化とはプロダクトモデルが複数のアプリケーションシステムから利用可能であることを意味する。そのためにはプロダクトモデルの情報構造は特定のアプリケーションシステムに対応して特化されたものではなく、全てのアプリケーションシステムから利用可能なものでなくてはならない。そのような情報構造を求めるのに最適なのは実世界の対象物（この場合は製品である船）をなるべくあるがままに情報構造に置き換えることである。これは各アプリケーションシステムの扱う情報は全て実世界の対象物のある一面であることから、当然のことと言える。

2.2.2 整合性

整合性とはプロダクトモデル内の情報に矛盾が無いことであり、これはプロダクトモデルの各部分の間の関係情報を持つことによって実現することができる。例えば隔壁にパイプ貫通孔が開いている場合その隔壁とパイプとを関係付けておくことにより、設計変更によってパイプの導設位置が変更されたにもかかわらず、孔の位置がそのままであるような矛盾の発生を防ぐことができる。

さらにプロダクトモデル内部の情報の冗長性を極力少なくすることにより、本質的に矛盾した情報を持たないようにすることも考えられる。例えばパイプ貫通孔の位置情報自体は持たずに必要な時点でパイプと隔壁の位置から交点を求めることとすれば、前述のような矛盾は発生しえない。

2.2.3 普遍性

プロダクトモデルは船種の違いのように一部分は共通だが他の部分は異なる対象を表現する必要がある。また、タンカーの主流がシングルハルからダブルハルに移ったように実世界の状況の変化に応じて表現すべき対象が変化する場合がある。それらに対して船種ごとに全く別々のシステ

ムを作成したり、対象の変化に応じて全く新しくシステムを作り直すのでは、そのコストについては勿論のこと、ある船の計画・生産過程の途中でシステムの切り替えが行われたため以前のデータが利用出来なくなったり、設計において他の船のデータを流用出来なくなるなどのデメリットがあると考えられる。そのように対象の違いに対してシステムが全く違ってしまうのを避けるためにプロダクトモデルでは対象の違いに対する普遍性が必要である。

普遍性にはプロダクトモデルシステム内部の普遍性と外部の普遍性がある。内部の普遍性というのは対象の違いによるプロダクトモデルシステム自体への影響が最小限であることで、そのためにはまずプロダクトモデルシステムが構造化され、独立した機能ユニットの組み合わせで構成されるのが望ましい。それによってシステムの変更が局部的にユニットを変更するだけで済む。

また異なる対象の内でも共通する部分については同じユニットで対応できるのが望ましい。例えば造船 CIMS パイロットモデルの対象としている船は全て基本的には鋼の板と骨とを組み合わせた船殻構造に艤装品が取り付けられているという点で共通である。従って船を板と骨と艤装品の組み合わせで表現すればそれに関するユニットは全く共通に使うことができる。そのように対象をなるべく共通部分が多くなるように表現して、それらを共通のユニットで扱うことにより普遍性を高めることができる。

一方プロダクトモデルシステムの外部の普遍性とはプロダクトモデルシステム内部の変化が外部のアプリケーションシステムに影響を及ぼさないことである。そのためにプロダクトモデルシステムとアプリケーションシステムとの間で通信するためのプロトコルはプロダクトモデルシステムの内部構造とは無関係に定められなければならない。

2.3 プロダクトモデルシステムの範囲

前記の定義のようにプロダクトモデルは特定の製品(船)に付随した情報を格納する。従って複数の製品に共通な情報(作業者や生産設備に関する情報、素材や艤装品のカタログ情報など)はプロダクトモデルには含まれない。しかし造船 CIMS の働きからいえばこれらの情報はプロダクトモデルの情報と密接な係わりがある。従ってプロダクトモデルシステムとしてはこれらの情報も含めて考慮せねばならない。

3. プロダクトモデルの実現

3.1 基本的な考え方

2.2 に表した要件を満たすプロダクトモデルを実現するために多くの考え方が利用されている。その内以下の 3 つが特に基本的である。

3.1.1 Entity-Relationship モデル³⁾

Chen により提案されたデータモデルで、対象を実体および関係をノードとするネットワーク構造で表す。関係情

報は実体に付属しているという考え方に対して関係も独立した情報単位として陽に扱う。対象物である製品(船)は複数の部分品から成り立っている、それら部分品を実体とすることにより、2.2.1で述べたように対象物となるべくあるがままに情報構造に置き換えることができる。実世界での事物という最も根本的なレベルをそのままプロダクトモデルに取り入れているため、わざわざ人為的に情報構造をアプリケーションごとに分ける必然性がなくなり、プロダクトモデルはおのずから統合化がなされることとなる。

しかし造船の作業で扱われるのは部分品に留まらない。ブロック、構造物、配管系統などの集合物や区画、孔、溶接継手などの仮想物、さらに生産作業などの出来事なども実体として扱う必要がある。ただしこれらは必ずしも実世界での事物そのものではなく、対象のある観点から観た場合に現われる概念物である場合が多い。例えばブロックなど集合物は特定の観点から観た場合に関連がある部分品を集めたものである。したがって必ずしも他の観点から也同样に関連が認められるとは限らない。言い換えるとそれらは対象の特定の一面を表すための特化された実体でありうる。

このようにプロダクトモデルでは対象物である製品(船)の部分品を統合化の基盤として、その上に特定の観点むけに特化された実体が存在する階層構造を成している(Fig. 1)。さらに階層構造は一通りではなく、各観点ごとに別なものがあろう。この多面性によりプロダクトモデルは様々な業務に対応することが可能となる。

さらにこれらの実体間にある関係を陽に扱うことにより2.2.2で述べたように整合性を保つことが可能となる。

3.1.2 オブジェクト指向

オブジェクトとはデータとそれを処理するためのプログラムとをユニットとしたもので、複数のオブジェクトが情報をやりとりすることにより情報処理が行なわれるという概念である。3.1.1で述べた実体と関係をそれぞれオブジェクトとすることにより Entity-Relationship モデルを利用したプロダクトモデルをコンピュータ上に実現することができる。

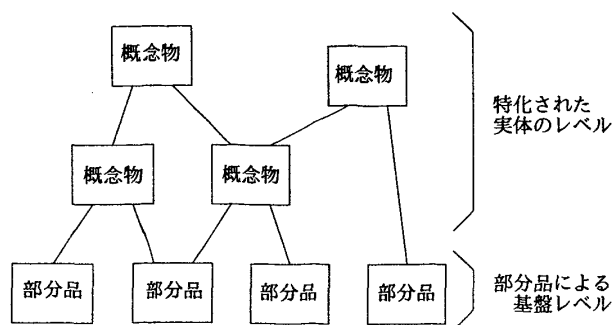


Fig.1 Hierarchy of Product Model

プロダクトモデルの実現にとってオブジェクト指向の利用には3つの主なメリットがある。第一はオブジェクトがデータだけではなくプログラムも持てることである。これによりオブジェクトすなわち実体と関係が能動的に動作できるようになり、例えばデータが実体/関係に設定されると関連する実体/関係に対してアクションを起こし、全体の整合性を保持したり、ある実体/関係に対してデータの参照が行なわれると関連する他の実体/関係までたどって参照するなどといった処理が容易に行なえる。

またオブジェクト同士はお互いに相手の内部構造に影響されることなく情報のやりとりができる独立した機能ユニットであるため2.2.3で述べたようにプロダクトモデルシステム内部の普遍性を実現することができる。

第二はオブジェクト指向では差分プログラミングを可能としていることである。プログラミングにおいては一般に機能の一部が異なっているが他は同じであるというプログラムが複数必要となるが、オブジェクト指向によるプログラミングにおいても同様に一部が異なっているが他は同じであるというオブジェクトが複数必要となる。このような場合にオブジェクト指向ではオブジェクト間に親子関係をつけ、子オブジェクトに親オブジェクトの機能を受け継がせることができるため、親とは異なる機能のみを改めてプログラミングするだけでよい。このようなプログラミングの手法を差分プログラミングと呼ぶ。2.2.3で述べたようにプロダクトモデルシステムは共通の機能を共通のオブジェクト群に持たせることによりシステムのレベルでの普遍性を高めているが、差分プログラミングはプログラミングレベルでの普遍性を高めるものである。

第三はプロトタイピング手法¹⁰による開発の効率化である。プロトタイピング手法はシステム開発手法の一種でシステムの仕様を完全に定義してから開発を行う従来のウォーターフォールモデル型の手法に対して、仕様が固まらない段階でプロトタイプシステムを試作し、それを参考としながら仕様を明確化していく手法である。オブジェクト指向では機能をオブジェクト単位にユニット化することができ、また前記の差分プログラミングによりシステムの部分変更が行いやすく、プロトタイピング手法に適している。

3.1.3 モデル操作言語

2.2.1で述べた統合化のためにはプロダクトモデルに複数のアプリケーションシステムが同時にアクセス可能でなければならない。それを実現するためにプロダクトモデルシステムとアプリケーションシステムとはネットワーク上のサーバ・クライアントの関係とするのが適当と考えられる。それらの間で通信を行なうためのプロトコルが必要となるがそれをモデル操作言語と呼ぶこととした。

2.2.3で述べたように普遍性を保つためにはモデル操作言語はプロダクトモデルシステムの内部構造とは無関係でなければならない。またモデル操作言語はプロダクトモデ

ルを利用する全てのアプリケーションシステムから使われるため、アプリケーションシステムにとって使いやすいものでなければならない。従ってモデル操作言語は各アプリケーションシステムの業務に即したもので、言い換えると造船業者が日常業務で使っている言葉に即したものであるのが望ましい。例えば日常業務では同じ「……の重量を求める」と表現される処理であっても、弁の重量を求める際には弁の属性である重量データを直接参照するのに対して、パイプの重量を求める際にはまず単位長あたりの重量と長さを求めてからその積を求めるというようにプロダクトモデルシステム内部で行われる処理が異なる。このような場合でもそれぞれの処理に対するモデル操作言語は日常業務の言葉と同様に同じ表現であるのが望ましい。

今回開発したPPMSはオブジェクト指向のシステムであるので、モデル操作言語もオブジェクト指向形式とし、実体や関係に対して何らかのメッセージを送りその返事を受け取るという形にした。

3.2 三層スキーマ⁴⁾との比較

プロダクトモデルの実現のイメージを明確にするために、データベースの構造である三層スキーマに当てはめてみる。三層スキーマとはデータベースを内部構造である内部スキーマ、データベース化する対象をデータモデルに従って抽象化した概念スキーマ、そしてアプリケーションプログラムの観点から対象を観た外部スキーマの3スキーマの階層構造として捉えようとする考え方である。これにPPMSを当てはめてみると、大まかには各オブジェクトの内部のデータ形式やプログラム等が内部スキーマ、製品(船)の実体や関係に対応するオブジェクトが概念スキーマ、それぞれのアプリケーションシステムごとに利用するモデル操作言語の集合が外部スキーマに対応する(Fig. 2)。しかし概念スキーマとしたオブジェクトの内には

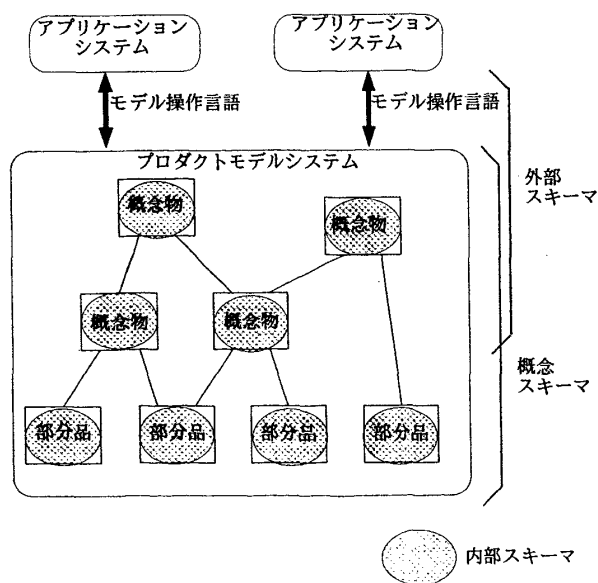


Fig. 2 Three Schema in PPMS

3.1.1で述べたように特定の観点向けに特化されたものもあり、それらはその観点を持つ限定されたアプリケーションシステムからしか利用されない。そのようなオブジェクトは概念スキーマと外部スキーマとの間に位置すると考えられるので、PPMSにおいては概念スキーマと外部スキーマとを明確に区別することができない。これはある造船業務には単一の観点のみが対応するのではなく、同時に複数の観点から対象を観ながら業務が行われる場合が多いためである。

4. PPMSの構成

PPMSの機能構成をFig. 3に示す。構成要素は以下の通りである。なおPPMSは通信機能と幾何機能の一部がC言語で作成され、他は全てSmalltalk V 2.5で作成されている。

(1) 通信機能

PPMSがアプリケーションシステムとの通信を行なう機能。不特定受信(自分宛の通信は発信元に関わらず受信する)/特定発信(送り先を指定して発信する)形式で、送り先の指定は各システムに付けられている通信用プロセスIDによって行う。また通信内容には自動的に発信元の通信用プロセスIDが付加されるので、それによって受信側が発信元を知ることができる。

(2) モデル操作言語解釈機能

アプリケーションシステムから受信したモデル操作言語(文字列)の構文解析をして、現われたオブジェクト名やデータ表現に対応するオブジェクトやデータに変換してそれを実行する。さらに実行結果のオブジェクトやデータをオブジェクト名やデータ表現の文字列に変換してアプリケーションシステムへ送り返す。

(3) 名前管理機能

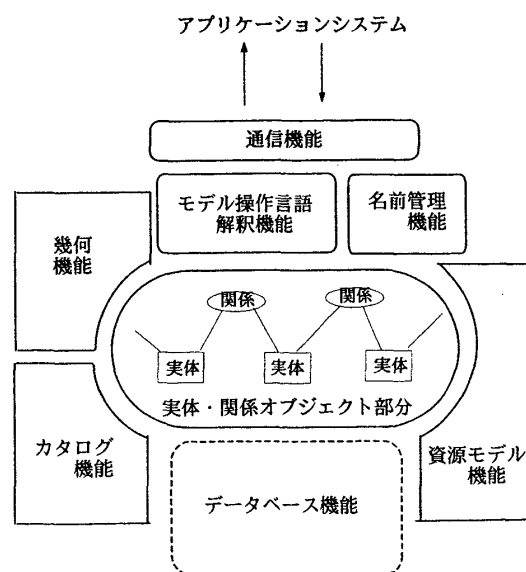


Fig. 3 Architecture of PPMS

オブジェクト名の管理を行なう機能。これについては5.3で詳細を述べる。

(4) 実体・関係オブジェクト部分

実体・関係オブジェクトの集合。先に述べたように統合化されており実質的な区分けはないが、アプリケーションシステムとの対応から名目的に、船殻、配管、工作の3ジャンルに分けている。それらについては既報⁵⁾⁶⁾⁷⁾に概念が述べられているので本論文では触れない。

(5) 資源モデル機能

資源モデルは工場において工作作業に係わる作業者および設備をモデル化したもので、前述のように厳密にはプロダクトモデルには含まれないが、PPMSには含まれなければならない。資源モデルについても既報⁹⁾に概念が述べられているので本論文では触れない。

(6) カタログ機能

カタログ機能は艤装品、船殻材料、標準的な船殻構造など、プロダクトモデル内に多数現われる情報を一箇所にまとめて管理するものである。例えば同じ型の艤装機器や標準的な工作法などの詳細な情報(要目、形状等)はカタログにのみ持ち、それぞれの機器はそれを参照する。本システムでは以下の情報のカタログを持つ。

配管関連：パイプ、フランジ、配管機器

船殻関連：船殻材質、型鋼、型鋼の端部形状、板鋼の端部開先形状、軽目孔・スカラップ形状、トリッピングブラケット

(7) 幾何機能

プロダクトモデル内の全ての幾何的な情報を表現し幾何計算を行なう機能である。これについては5.4で詳細を述べる。

(8) データベース機能

プロダクトモデルは複数の作業者が並行して使用するのが一般的と予想される。また計画作業の進展に伴って時間的にバージョンが更替されていくだけではなく、計画途中では同時に複数の案が異なるバージョンとして存在することがありうる。これらに対処するのがデータベース機能であり、プロダクトモデルシステムには不可欠な機能であるが、今回の開発においては検討の都合上データベース機能はプロダクトモデルと別個に検討しており⁹⁾、PPMSには組み入れていない。

5. PPMSの要素技術

本システムに組み入れられた要素技術について述べる。

5.1 番船管理

PPMS内には複数の船のプロダクトモデルが格納されているが、その中の一船分のオブジェクト群を管理するための番船管理用のオブジェクトが必要である。このオブジェクトは設計作業の最初に作成され、5.3で述べる船ごとの名前対応表を管理し、また船名や主要目など船全体に関

する情報を保持する。

原則としてアプリケーションシステムは一時に一船のプロダクトモデルにしかアクセスせず、PPMSの使用開始時にどの船をアクセスするかを指定する。ただしパイロットモデルの試用においては常に一船のプロダクトモデルしかアクセスしないため、現状のPPMSでは指定なしで特定の船にアクセスするようになっている。

5.2 モデル操作言語の文法

5.2.1 形式

3.1.3で述べたようにモデル操作言語はオブジェクト指向の形式をしている。具体的にはPPMS内の既存のオブジェクトにメッセージを送る形式をしている。Fig.4にモデル操作言語の例を示す。一単位のモデル操作言語は文と呼ばれ、下の形式をしている。

[レシーバ セレクタ パラメータ]

レシーバはメッセージを受け取るオブジェクトの名前であり、セレクタとパラメータはメッセージを構成する。セレクタはメッセージの種類を表し、レシーバが示すオブジェクトが持っている特定のプログラムと対応している。パラメータはそのプログラムに渡されるデータの配列である。

一つの文がPPMSに送られるとその処理結果として一つの返事がPPMSから返される。これは返値と呼ばれ、下の形式をしている。

[データ]

さらに複数の文を()で囲み一時に送ることもできる。その場合は送った文と同数の返値が()で囲まれて返される。

5.2.2 データ

モデル操作言語で使用されるデータは、シンボル、文字

```
[HMOLD NEW: SURFACE: (#DECK [GPLANE Z: (18600)])]

[DECK FREE-EDGE: ((GBORDER-LINE-S-ON-MOLD LOOP-ON: WITH:
  (DECK ([GPLANE Y: (22000)] [GPLANE X: (0)]
    [GPLANE Y: (-22000)] [GPLANE X: (250000)] ))))]

[HMOLD NEW: SURFACE-PLANE: TRIMMED-BY-MOLDS: (
  #TBHD-AFT [GPLANE X: (40000)]
  ((PS-SHELL [GPOINT XYZ: ((40000 0 1000))])
  (DECK [GVECTOR XYZ: ((0 0 -1))])))]

[HMOLD NEW: SURFACE-PLANE: TRIMMED-BY-MOLDS: (
  #INNER-BOTTOM [GPLANE Z: (3000)]
  ((PS-SHELL [GPOINT XYZ: ((40000 0 3000))])
  (TBHD-AFT [GVECTOR XYZ: ((1 0 0))])
  (TBHD-FORE [GVECTOR XYZ: ((-1 0 0))])))]

[INNER-BOTTOM SET-BAR-MOLD: AT: SIDE: (
  #LONG-IB1 [GPLANE Y: (11000)] #LOWER)]

[INNER-BOTTOM SET-BAR-MOLD: AT: SIDE: (
  #LONG-IB2 [GPLANE Y: (12000)] #LOWER)]

[INNER-BOTTOM FITTING-BAR-MOLDS]
```

Fig. 4 Model Manipulation Language

列, 数値などであり, さらにオブジェクト名もデータとして扱える。また複数のデータを () で囲んだ配列もデータとなる。配列はネスティング, 即ち配列が他の配列の要素になることを許す。数値としては整数, 倍精度実数, 分数が扱える。またシンボル, 文字列には英数字の他に日本語が使用できる。

5.2.3 文のネスティング

文の中のオブジェクト名あるいはデータがあるべきところに他の文をおくことが出来る。例えばレシーバのところに戻値としてオブジェクト名を返すような文があればその文が先に実行され返値がレシーバとなる。この機能により複雑な操作を少数のモデル操作言語で行える。

5.3 名前管理

5.3.1 名前管理の必要性

モデル操作言語においてオブジェクトは名前で特定されるが, 名前は単に PPMS がオブジェクトを特定するだけのものではない。即ち, 実際の造船業務においては部品等に系統的な名前を付けその属性を表す文字などを付加しており, 作業者がその名前をみれば物の属性がある程度わかるようになっている。造船 CIMS においても人間系が不可欠のことから名前がこのような機能を持つのが望ましい。しかしながら作業者が日常使う名前はコンピュータシステムが使う名前とは異なり名前と対象物との間に一対一の対応関係があるとは限らない。たとえば異なる造船ステージで同じものに異なる名前を付ける場合があるし, あるいは特定の船のメインエンジンを船名を省略して単純に「メインエンジン」とだけ呼ぶ場合もある。これに対応するために PPMS では受け取ったモデル操作言語に含まれるオブジェクト名を発信元のアプリケーションシステムの種類に従って適当なオブジェクトに変換し, 返値を返す場合にも適当なオブジェクト名を選択できるような名前管理機能が必要である。

5.3.2 アプリケーション用オブジェクト名

5.3.1 で述べた機能は PPMS 内で船ごとかつアプリケーションごとに名前-オブジェクトの対応表を持つことによって実現される。5.1 で述べたように各アプリケーションシステムは船を指定して PPMS にアクセスし, また 4(1) で述べたように通信から発信元アプリケーションシステムを知ることができるため, PPMS からはどの対応表を使用して名前とオブジェクトとの変換を行えばよいかわかる。

5.3.3 共通オブジェクト名

オブジェクト名の中には全てのアプリケーションから使われるものもある。そのために各船ごとに共通名とオブジェクトとの対応表を持つ。あるオブジェクトに付けられる共通名はユニークである。

5.3.4 船以外オブジェクト名

資源モデルやカタログ機能に含まれるオブジェクトは特

定の船には所属せず, 従ってアプリケーション用オブジェクト名や共通オブジェクト名のように船ごとの対応表では処理できないので, 船とは独立の対応表を持つ。この名前は PPMS 内でユニークである。

5.3.5 他アプリケーション用オブジェクト名サーチ機能

アプリケーション間で情報の受渡しが行われる時に, 一時的に他のアプリケーション用オブジェクト名からオブジェクトを参照したい場合がありうる。そのためにアプリケーションが PPMS に対して指示すれば他のアプリケーション用の対応表も利用できる機能を用意した。他のアプリケーション用の対応表の利用開始や終了の指定はモデル操作言語によって行われる。

5.4 幾何オブジェクト

5.4.1 幾何オブジェクト

幾何形状情報が主体であった従来の CAD とは異なりプロダクトモデルにおいては幾何形状情報は実体や関係に付属している属性の一部である。PPMS では幾何形状を表現する幾何オブジェクトを実体オブジェクトや関係オブジェクトに付属させることによりこれを表現している。幾何オブジェクトとしては, ベクトル, 点, 直線 (有限&無限), 円弧, 自由曲線 (有限), 線列 (直線, 円弧, 自由曲線等をつなげたもの), 平面 (無限&有限), スウィープ面 (有限), 直方形, 円筒形がある。幾何オブジェクト同士の間にも関係がありうる。例えば無限直線は点とベクトルから定義されるので, 無限直線のオブジェクトは点のオブジェクトとベクトルのオブジェクトを含む。

各オブジェクトは幾何計算プログラムを持つ。たとえば直線オブジェクトは他の線・面との交点計算, 線長計算, 他の直線と垂直かあるいは平行かのチェック等の機能を持つ。

5.4.2 expression による定義

設計作業においては点が2線の交点として定義されたり直線が曲線の接線として定義されるなど, 幾何情報が他の幾何情報を参照して決められる場合が多い。さらにその参照関係がその後に影響し, 設計変更によって参照されている幾何情報が変化した時に参照している幾何情報も追従して変化すべき場合も多い。このような機能を実現するために幾何オブジェクトに他の幾何オブジェクトを参照した定義 (これを expression と呼ぶ) から幾何計算して自分自身についての幾何情報を算出できる機能を持たせた。さらにその定義自体をも保持しておいて, 参照先のオブジェクトが変化した場合に再計算を行なうことによって変化に追従できるようにした。また定義については他の定義と組み合わせで初めて幾何情報を算出できるような部分的な定義も可能とした。例えばある点 P1 が直線 L1 と L2 上に位置しているという条件がある場合にはそれぞれ「P1 は L1 上にある」「P1 は L2 上にある」との意味の expression を個別にモデル操作言語として PPMS に送っておけば必要

な時点でL1とL2との交点計算が行われP1の座標値を求める。計算後はアプリケーションシステム側からの指定によって、1) その座標軸を保持してL1もしくはL2が変化しても追従しない。2) 必要な場合は再計算する。の2モードが選択可能である。

5.5 コメントオブジェクト

実際の作業でのドキュメントにはそれを解説する何らかのコメントが付随することが普通である。これに対応してPPMSではコメントオブジェクトという機能を導入した。このオブジェクトは単にコメントを文章として持つだけでなく、他のオブジェクトに対する参照関係を持つことができる (Fig. 5)。例えば「隔壁BにパイプPを通すためにクリアランス 10 mm の孔を開ける必要がある」というコメントの隔壁BやパイプPは単なる文字列ではなく、他のオブジェクトに対する参照関係であるので、このコメントに関連する隔壁Bの形状やパイプPの外径及び位置の情報を自動的に入手するなどのハイパーテキストとしての処理が可能である。またオブジェクトにアプリケーションシステムによって異なる名前が付けられていても、5.3で述べた名前管理機能によってそれぞれのアプリケーション向けの名前でコメントの読み書きが可能である。さらにコメントオブジェクトには宛先ユーザを指定できるので、作業者同士がプロダクトモデルについて検討・連絡する際の通信に利用することができる。

5.6 実行の効率化のための工夫

PPMSにおいては存在するオブジェクトの数が膨大であり、さらに一つのモデル操作言語に対する処理でも多数のオブジェクトの間で情報のやり取りが必要となる場合が多い。そのため必要な記憶容量や処理時間も大きくなる。これらをできるだけ少なくするために幾つかの工夫を行っている。

5.6.1 簡略化オブジェクト

PPMSでは原則的には実在物は全て単独のオブジェクトとしているが、単独で管理する必要がなく、数の多いものについては例外としている。例えばボルト・ナットなどの情報は接続関係オブジェクトの属性情報とみなしている。また、孔につくカラープレートは孔の付属物として他

種の板のオブジェクトにくらべて持つ情報を簡略化している。カラープレートと他の部材との間の溶接接続関係オブジェクトについても同様に簡略化している。

5.6.2 データ保持

3.1.2で述べたようにPPMSではあるオブジェクトに対してデータの参照が行われた場合に関連する他のオブジェクトまでたどってデータを参照するという処理が多く行われている。例えばある構造ブロックの重量を求める場合には含まれている全ての部材のオブジェクトを参照してそれらの重量を求めた後にそれらの和を求める。原則的にはこれらの処理はデータ（この場合はブロック重量）が必要となる度に毎回行われるが、板材の周囲形状や溶接接続関係の溶接線形状等、特に利用頻度が多く求めるのに処理時間がかかるデータに関しては一度処理した結果を保持しておき、再処理を省くことができるようにしている。しかしこれによってプロダクトモデルの整合性の維持が制限される場合がある。例えば一度板材の周囲形状を求めた後で、その板材をトリミングしている部材の位置が変化した場合にそれが板材の周囲形状に反映されないなどの場合がある。

5.7 エラー処理

PPMSはモデル操作言語に文法的な誤り及び未定義のオブジェクト名が存在した場合には返値としてエラー返値を返す。エラー返値にはエラーの内容を示すエラーメッセージが含まれている。

6. 開発プロセスの実際

今回のPPMSの主要部分の開発は以下の段階を追って行われた。

- (1) オブジェクト及び情報構造の決定
- (2) オブジェクトが持つ属性情報の決定
- (3) モデル操作言語の決定
- (4) 作成

ただしこれらは途中で検討結果を前段階にフィードバックしながら行われ、作成の途中で情報構造が変更される場合があった。上のうち(1)、(2)、(3)は設計段階であり、(4)はプログラミング段階であるが、それぞれについてより詳細に述べる。

6.1 設計段階

対象をどのような実体と関係で表現し、それらがどのような階層構造をなし、どのような属性情報を持ち、どのようなモデル操作言語の処理を行うかを設計する段階であり、実際にはこの作業の中には対象をオブジェクトの枠組みから分析する作業も含まれているが、今回は分析と設計とは明確には区別されことなく行われた。

作業は各アプリケーションシステム担当者がアプリケーションシステムがPPMSを利用する立場からの提案を行い、PPMS担当者がそれらを統合化、整合性、普遍性とい

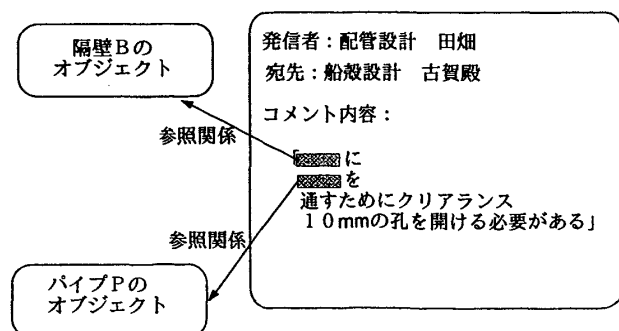


Fig. 5 Comment Object

う要件を満たすように修正したものを再提案する、といった形で共同作業により行われた。システムの設計のための技法については、実体間の関係をオブジェクトとすべきかどうかの判断基準などの明文化されたものも若干は存在したが、大部分は PPMS 担当者の感覚的な判断による場合が多かった。そのため各アプリケーションシステムの担当分野に対する PPMS 担当者の理解度などにより、普遍性のレベルやモデル操作言語の抽象化のレベルなどにバラツキが存在する結果となった。

6.2 プログラミング段階

プログラミング段階は単に設計段階の結果をプログラミングするのではなく、プログラミングを行いながら作成された部分の試用及び設計の再検討を行い、必要に応じて設計変更を行うプロトタイピング手法に基づいて作業を行った。部分的な設計変更に伴うシステムの変更は、モデル操作言語の変更のようにプログラムのみの変更ですむ場合は、オブジェクト指向言語のユニット化、差分プログラミングおよび Smalltalk 自体の開発環境などによって比較的効率的に行うことができた。しかしながらプログラムの変更だけではなく、オブジェクトが成す情報構造やオブジェクトの持つ属性にまでの変更が及ぶ場合には多くの作業量が必要となった。

7. 知見および今後の課題

7.1 プロダクトモデルの要件と基本的な考え方に関して

7.1.1 統合化

PPMS とアプリケーションシステムとを切り離しネットワーク上でサーバーとクライアントとしてモデル操作言語のみを利用して通信させることにより、複数のアプリケーションシステムが同時に並行して PPMS を利用できることを確認した。さらに Entity-Relationship モデルの考え方に基づき部分品を実体としてモデル化し、それを基盤とすることによって複数のアプリケーションシステムが共通して利用することが可能なプロダクトモデルを構築できることを確認した。今後の課題としては今回は PPMS の範囲外としたバージョン管理や並行作業制御などのデータベース機能の取り込みがある。

7.1.2 整合性

Entity-Relationship モデルにより関係を表現し、情報の冗長性を減らすことにより矛盾が発生しにくいプロダクトモデルを構築できることを確認した。さらにオブジェクトの持つプログラムによって能動的に整合性を保持する機能を実現できることを確認した。しかしながら今回 PPMS で構築したプロダクトモデルの持つ情報は実用プロダクトモデルに比較して極く一部であり、より多くの情報の間の整合性を確実かつ効率的に保持できるかどうかは今後の課題である。

7.1.3 普遍性

オブジェクト指向による PPMS のユニット化と差分プログラミングによって、対象の違いに伴うシステムの変更が比較的少なく済み、プロダクトモデルシステム内部の普遍性が高まることを定性的に確認した。さらにプロダクトモデルシステムの内部構造の変化がアプリケーションシステムに影響しないようなモデル操作言語を作成し、プロダクトモデルシステム外部の普遍性を高めることができた。今後の課題としてはどのようにシステムをユニット化すれば最も普遍性が高まるかを検討する必要がある。

また 3.1.3 で述べたようにモデル操作言語はアプリケーションから使いやすい必要があるが、それを追及しすぎ、アプリケーションシステム向けに特化しすぎると、逆にプロダクトモデルシステム内部の普遍性が低下することが予想される。その折り合いをどこでつけるのが造船 CIMS にとって有利であるかも今後の課題である。

7.2 要素技術に関して

要素技術に関する主な知見及び今後の課題を上げる。

7.2.1 名前管理・番船管理

これらの機能は作成されたが一船しか対象とせず、オブジェクトの数も名前の種類も比較的少ない今回の試用においてはその有効性を確認することができなかった。より実用レベルでの試用による確認が今後の課題である。また今回の名前管理機能は同時に複数の番船や工場にアクセスするアプリケーションシステムに対応していないのでこれについても今後の課題となる。

7.2.2 幾何オブジェクト

幾何オブジェクトにより幾何情報を実体及び関係の属性情報として表現でき、さらにそれらに幾何処理機能を持たせることができることを確認した。しかしながら expression による定義機能については作成はしたが、業務への効果的な利用方法が未検討であり、その有効性の確認はなされていない。

7.2.3 実行の効率化のための工夫

情報の冗長性を減らしたり、オブジェクトが能動的に動作するなどの整合性を保持するための機能や expression 定義などの高度な機能はその一方で処理に時間がかかることが判明し、実行の効率化のための工夫が不可欠であることが確認された。今回行なった実行の効率化のための工夫は処理の効率化に役立ったものの、その代償として高度機能を一部制限する結果となっている。高度機能を制限することなく効率化するための方法や、高度機能をどこまで制限するのならプロダクトモデル全体の有用性を損ねずに効率化が可能となるかの検討が今後の課題である。

7.2.4 エラー処理

今回開発したエラー処理はごく表面的な文法エラー等のみを対象としており、それ以外のエラーに対しては PPMS 自体が中断してしまうため、アプリケーションシステム側

はエラー発生の実事さえ知ることができない。実用のプロダクトモデルシステムでは、エラー発生の実事およびその種類をアプリケーションシステム側に知らせるだけでなく、アプリケーションシステムが再実行できるようにプロダクトモデルの状態を保持できる必要がある。

7.3 開発プロセスに関して

今回の開発は明確な設計技法が殆ど無い状態で行なわれたため6.1で述べたように設計段階における不具合が生じた。6.2で述べたように設計段階にまで遡っての変更はプログラム段階に大きな影響を与えることが今回の開発の経験からも明らかとなり、プロダクトモデルの分析、設計のための設計技法の必要性が確認された。著者は今回の開発経験を基にして造船CIMSプロダクトモデル開発のためのオブジェクト指向分析技法を提案したが⁹⁾、それもまだ十分といえるものではない。今後さらに確立していくことが必要であるが、そのためには不十分な技法であっても取敢えず利用してみることで知見を深めることが必要と考えられる。

8. 謝 辞

本研究は、モータボート競走公益資金による財団法人日本船舶振興会の補助事業として実施された、シップ・アンド・オーシャン財団「造船CIMSパイロットモデルの開発研究」プロジェクトの一環としてなされました。同プロジェクト及びその後継プロジェクトである「造船業CIMフレームモデルの開発」プロジェクトの関係者各位のご助力に謝意を表します。また多くのご指導・ご助言を下さいました東京大学工学部の小山健夫教授に謝意を表します。

参 考 文 献

- 1) (財)シップ・アンド・オーシャン財団：平成3年度造船CIMSパイロットモデルの開発研究報告書，1992年3月
- 2) 野本敏治，片山真一，青山和浩，杉本晃：製品モデルを用いた工作情報の獲得手法の検討，日本造船学会論文集第168号（1990年12月），p 615-622
- 3) CHEN, P. P.: The Entity-Relationship Model-Toward a Unified View of Data, ACM Transactions on Database Systems Vol. 1 No. 1 (Mar. 1976), p 9-36
- 4) 長尾他：情報の構造とデータベース，岩波書店(1983年)
- 5) 田畑光敏，古賀輝一：造船CIMSプロダクトモデルのためのオブジェクト指向分析技法の提案，日本造船学会論文集第170号（1991年12月），p 815-825
- 6) 中井裕司，田畑光敏：オブジェクト指向に基づく配管モデル表現に関する研究，日本造船学会論文集第168号（1990年12月），p 623-628
- 7) 雨宮俊幸：造船CIMSのための工程設計システムの構築，日本造船学会論文集第168号（1990年12月），p 629-638
- 8) 峯村隆久，雨宮俊幸，堀内一敏：造船CIMSのための工程管理システムの構築，日本造船学会論文集第170号（1991年12月），p 827-841
- 9) 長瀬裕：造船CIMSのためのモデル情報の整合性維持に関する研究，日本造船学会論文集第170号（1991年12月），p 857-866
- 10) (財)シップ・アンド・オーシャン財団：平成2年度造船CIMSパイロットモデルの開発研究報告書，1991年3月