造船 CIMS のためのモデル情報の整合性維持に関する研究

正員 長 瀬 裕*

A Study on Consistency of Model Information for CIM for Shipbuilding

by Yutaka Nagase

Summary

CIM for shipbuilding, we call CIMS, has been studied by Ship & Ocean Foundation. This paper is one of the results of works for researching of CIMS.

I think product model is most important technology of CIMS. The works of creating information on shipbuilding must be done simultaneously. Therefore, it is important to communicate each other and to keep consistency of model information in a database.

The aim of this paper is to propose a mechanism based on version model for cooperative design. A prototype system of this mechanism is realized using Object-Oriented Database (OODB), and this mechanism is confirmed to be efficient.

1. 緒 言

(財)シップ・アンド・オーシャン財団では「造船 CIMS パイロットモデルの開発研究委員会」(委員長笹川陽平氏)においてパイロットモデルの作成を通して造船 CIMS の技術的課題を検証するための研究を平成元年度から 3 年間かけて行っている。本論はこの研究の一部として同時並行作業時のモデル情報の整合性の維持について検討した結果をまとめたものである。

造船においては設計に限らず工作情報の付加も含め、情報を生成する業務は多数の作業による同時並行作業となっている。造船 CIMS が実現してもそのような作業形態は変わらないであろうし、工期のより一層の短縮のために現在よりもさらに並行度が増すと考えられる。従来の造船用 CAD/CAM システムはこのような同時並行作業の支援を有効に行っているとは言えない。一方、造船 CIMS では扱う情報が従来の CAD/CAM システムと比べより複雑で膨大になるため、同時並行作業を行っている作業員間の情報伝達を助けたための協調活動支援機能が必須になると考え

られる。造船 CIMS では数十人の作業員が同時に並行して作業を行なう事になると考えられるので、本研究ではそのような作業形態を支援する方法として、データベースのバージョン管理を用いた作業員間の情報伝達のためのメカニズムを提案する。さらに同時に並行して作成された複数のバージョンをプロダクトモデルレベルの整合性を維持して統合するためのバージョンのマージ機構を考案し、オブジェクト指向データベース (OODB) を用いたプロトタイプングによりその有効性を確認した。

2. 造船の情報生成業務の現状

2.1 情報生成業務

一般に造船においては Fig. 1 に示すように設計管理から詳細設計までを「設計」と言い、工程設計などの生産準備から実際の製品を製作する生産工程までを「生産」と呼んでいる。しかし、「生産」と言う枠でくくられたそれらの業務の内、生産準備から作業指示までの業務は実際の製品を作成しているのではなく生産に必要な情報を生成している業務である。そこで本論では従来の組織の枠を離れ、設計管理から作業指示作成までの情報生成を主とする業務を「情報生成業務」と総称し、その業務内容を広い意味で「設計作業」と呼ぶことにする。さらに、それら情報生成作業に従事する作業員を広義の「設計者」と呼ぶことにする。

* (財)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委員会推進部専任チーム/三井造船(株)

Fig. 1 Classification on Information Creation

2.2 情報生成業務における情報伝達の現状

一般に CIM 先進産業と呼ばれている企業は、その製品が多品種少量生産と言われている分野の業種に属しており、生産に必要な情報を完全な形で整えてから実際の製品の生産を開始している。それに対して一品受注生産である造船では、各建造船毎に生産に必要な情報を短期間の内に揃える必要があるため、設計部門と生産部門が共同で、種々の業務毎にほぼ一斉に機能・工程・生産の各設計が行われ、生産に必要な情報ができた部分から直ちに生産が開始される。このため、船殻設計や舳装設計および生産管理などの各部門の多数の作業者が同時並行的に情報生成作業を進めることになる。

情報生成作業を進めるに当たっては、それぞれの設計者毎に分担が決められ、各設計者はその範囲内についての情報を生成してゆく。各設計者の設計内容の多くは他の設計者の設計範囲とは独立しており、分担範囲内で独自の判断で意志決定を行い情報を生成してゆくことができる。しかし一般に、ある設計者の作業は他の上流設計の決定した情報がその前提条件となっていることが多い。さらに同位の並列作業者とのあいだでも、その境界部分においては互いに密接に関連し合っている上に同一の情報を共有している場合も多いので、連続性を保ち不整合が発生しないように調整した上で設計業務を進める必要がある。そのために設計者は他の設計者と密接に情報交換を行い、必要な情報を適宜入手しながら設計を進めてゆく。例えば他の設計者の生成した情報のうち、まだ未確定の情報は仮図という形で入手することになる。また、出図情報の整合性を確保するためには協議図によって他部署と協議するような仕組みが採用されている。

従来のほとんどの造船用 CAD/CAM システムはこのような造船の情報生成業務における情報伝達の仕組みを支援するようには作られていない。そのためコンピュータ上に表現された情報を一旦紙の上に図面と言う高度に記号化された形で出力し、その図面を介して情報を伝達しているのが普通である。

現在このような形で設計者間で伝達されている情報をそ

Table 1 Classification of Communication

情報の性質	伝達手段
未確定な情報	仮図
整合性を確保するための情報	協議図
確定した情報	出図図面

の性質によって分類すると Table. 1 に示すように「未確定な情報」、「整合性を確保するための情報」、「確定した情報」の 3 種類に分類することができる。また、それらの情報の具体的な伝達手段としてはそれぞれ「仮図」、「協議図」、「出図図面」といった名称で呼ばれている図面が用いられている。

3. 造船 CIMS と協調活動支援

3.1 造船 CIMS のシステム構成

造船 CIMS のシステム構成が如何にあるべきかは造船 CIMS 実現の時期、およびその時までのシステム関係の技術の進歩の度合いに大きく影響されるためまだ検討中である。しかしデータベースを中心として見た場合の概略のシステム構成は Fig. 2 に示すような形になると考えられる。

造船 CIMS パイロットモデルの開発研究においては、造船の全業務を統一的に表現するモデルとしてプロダクトモデルを位置づけている。プロダクトモデルは造船の対象製品である完成品としての船そのものを表すだけではなく、船を建造するために必要な情報も含んでいる。プロダクトモデルはそれらの情報を表現するために Entity-Relationship モデルとオブジェクト指向といった 2 つの理論に基づいて構築されている。Entity に相当する「物」の情報、および Relationship に相当する「物」と「物」の関係を表す情報の両方を共にオブジェクトの形で表現し、

Fig. 2 System Architecture of CIMS

Fig. 3 Concept of Product Model

それぞれのオブジェクトを「物オブジェクト」および「関係オブジェクト」と名付けている^{1)~2)}(Fig. 3 参照)。さらに、このようなプロダクトモデルという概念に従って整理された情報は論理的にはただ一つのデータベースに格納されることにより永続性を持つと共に、多数の設計者による共有が可能になる。

3.2 チェックイン・チェックアウトメカニズム

造船CIMSではその対象とするデータの量は現状のCAD/CAMなどと比べ膨大なものになると推定される。また、その情報の質も単なる形状情報や物情報だけと言うのではなく、プロダクトモデルとして種々の複雑な関係情報を含んでいるうえ、同時に数十人の設計者がアクセスすることが想定される。造船CIMSではそのような状況においても実用的なパフォーマンスを得ることが必要である。一方CAD/CAMやCIMなどのエンジニアリング分野でのデータベース操作は事務処理などとは異なり一つのトランザクションの実行時間が非常に長く、数時間から極端な場合は数週間におよぶ場合もある。この種のトランザクションは長時間トランザクションと呼ばれている。このような長時間トランザクションが行なわれるデータベースに対して並行処理制御を行なうには、チェックイン・チェックアウトメカニズムを用いたロック機構が有効と言われている。チェックイン・チェックアウトメカニズムは共用のパブリックデータベースから必要なデータをコピーして、個人用のパーソナルデータベースに借り出し（チェックアウト）、作業が終了した時点で実行結果をパブリックデータベースに併合すること（チェックイン）により実現される³⁾(Fig. 4 参照)

現在製品化されているデータベース管理システム(DBMS)の中でも、最近エンジニアリング分野において注目を集めているオブジェクト指向DBMS(OODBMS)には、このようなチェックイン・チェックアウトメカニズムをサポートしている製品もある。造船CIMSのような膨大

Fig. 4 Check-in & Check-out Mechanism

データを扱うシステムはこの機能を有効に利用することにより、パフォーマンスの確保と並行処理制御を実現することが必要である。

3.3 協調活動支援

コンピュータを用いた人間のグループワーク支援に関する学問的な研究分野であるCSCW(Computer Supported Cooperative Work: 協調活動支援)に対する関心が近年急速に高まってきている。CSCWの具体的な応用分野は電子会議室から協同文書作成支援まで多岐にわたっており、多くの分野において研究が行なわれている^{4)~5)}。造船CIMSに関連の深い協調設計(Cooperative Design)の分野においては、Fig. 5に示すように共通の設計対象(Design Object)に対して多数のDesign Agentが協調して設計作業を行なう場合についての研究が進められている。協調設計の過程は「仕事の分解」と「分解された仕事の割付」から構成されており、「仕事の割付」をどのように行なうか、またそのために各Agent間のコミュニケーションを如何に行なうかが協調設計に関する主要研究テーマとなっている。

各Agent間のコミュニケーションはパラダイムとプロトコルに分解することができ、パラダイムとしては

- (1) 黒板モデル
- (2) メッセージパッシング

が提案されている。

一方、プロトコルの内容については現在までに

- (1) Contract Net Protocol

マネージャと呼ばれる特定のAgentが互いに干渉のないように作業を各Agentに割り振り、作業を割り振られたAgentは独立に作業を進めてゆく方法。

- (2) Functionally Accurate Cooperative Protocol

作業を割り振られたAgentは不足な情報があっても、とにかく自分だけで作業を全て処理し、マネージャはその結果をさらにその不足情報を持った他のAgentに処理させる方法。

- (3) Multi Agent Planning Protocol

Design Objectに対し共通の知識を持っており、お互いに他のAgentのことを思いやりながら、互いに干渉する作業は関係するAgentとコミュニケーションを取りつつ作業を進めてゆく方法。

の3つのアプローチが確立されつつあり、(1)、(2)、(3)

Fig. 5 Cooperative Design

の順にトップダウン的なアプローチになっている^{9)~11)}。

造船 CIMS においてもエキスパートシステムで全ての意志決定を行なうという事はあり得ず、重要な判断は必ず人が行なうことになると思われる。よって、このような協調設計に関する研究成果を造船 CIMS にそのまま適用することはできないが、各設計者間での情報伝達のための方法を確立するための指針になると考えられる。

3.4 造船 CIMS に求められる協調活動支援機能

現状の造船用 CAD/CAM システムには協調活動に対する支援機能はほとんど備えられていないが、造船 CIMS では次にあげるような項目に関してシステム側で支援を行なう必要があると考えられる。

(1) 同時複数案の検討

設計者が同時に複数の案を検討できるようにする。また、その検討内容を他の設計者が評価できるように共通のデータベースに格納する。

(2) プロダクトモデルのままでモデル情報の伝達

造船 CIMS では全てのモデル情報はプロダクトモデルの形に整理されているので、現状のように図面を介すのではなく、モデル情報をそのまま他の設計者に伝達する。

(3) データベースの安全性確保

全員が共通のデータベースにアクセスすることになるので、安全性確保のためにデータベースへのアクセス権限を個人単位で設定する。

(4) 設計者間でのモデル情報の不整合の解消

同時に並行して作業を行なっている複数の設計者間で発生するモデル情報の不整合を一定の規則に従って解消し、最終的にはただ一つの整合性のとれたモデルに統合する。

なお、上記(4)の整合性のうち、本研究で対象としているのはプロダクトモデルとしての整合性である。よって、製品の機能的な不整合は必要に応じてアプリケーション等で解消するものとし、本研究ではモデル情報の伝達に伴って発生する矛盾の解消のみを対象とする。

4. 造船 CIMS における協調活動支援

4.1 協調活動支援の実現方法

本研究ではパフォーマンス維持および同時実行制御のためのデータベースのチェックイン・チェックアウトメカニズムという考え方と情報のグルーピングとも言えるバージョンという考え方を結合し、同時並行作業時に前章で述べた協調活動支援機能を実現するための方法を考案した。その方法について以下に述べる。

4.1.1 バージョンの導入

造船の設計業務は基本的には管理者によって作業が割り振られ、各設計者が独立に作業を進める形なので、前出の Contract Net Protocol による協調活動支援が必要であ

る。そこで、本研究では作業を割り振られた各設計者が他の設計者の影響を全く受けずに独立して作業を進められる環境の提供方法を提案する。独立した作業環境を確保するために本論ではデータベースにバージョンの概念を導入する。バージョン管理によって、一つのバージョン内ではプロダクトモデルの概念により整合性が維持される。一方、バージョン間では互いに矛盾を許すことにより、各設計者は独立に作業を進めることができ、また複数のバージョンを同時に並行して作成することにより同時複数案の検討が可能になる。各設計者はパブリックデータベースからバージョン単位でモデル情報をチェックアウトし、パーソナルデータベースに自分だけの作業環境を作成する。そこで作業を行なった後チェックインしてパブリックデータベースにその結果を反映させる。これにより、バージョン単位でのモデル情報の独立および並行処理制御が実現される。さらに大量データによる作業中の処理速度の低下を防ぐことができる。

4.1.2 設計者間の情報伝達

前述のごとく造船の設計業務は基本的には Contract Net Protocol 的に進められると言っても、実際の情報生成に当たっては設計者やエキスパートシステムの知識として他の設計者のことをある程度考慮に入れた設計を行なうことになる。また、互いに設計情報を提供し合うことにより、他設計者の設計対象範囲と密接な関連を持った部分の設計を進めるのが合理的である。このような作業の進め方は Multi Agent Planning Protocol に基づいた協調活動と言える。Multi Agent Planning Protocol による協調活動支援において、伝達情報の内容を判断するのが人である場合には電子メールでこの機能を果たすることができるが、さらにモデル情報自体をプロダクトモデルの形のままで伝達するために下記のような方法を提案する。

モデル情報伝達のためにバージョンに三種類の区分を設け、それぞれ下記に示すように権限を有する者のみが参照・更新をできるようにする。

- ・作業バージョン：個人的な作業用バージョンであり、バージョン所有者のみが参照および更新できる。
- ・検討バージョン：各作業者間の情報伝達用のバージョンであり、バージョン所有者以外でも内容を参照することができる。ただしバージョンの更新に関しては情報の追加のみができ、すでに登録されている内容を変更・削除することはできない。
- ・公認バージョン：1 隻につき 1 バージョンのみ存在するオフィシャルな情報を保持するバージョンであり、登録権限所有者のみが登録することができる。(従来の出図のイメージ)

公認バージョンが2.2節の「確定した情報」、検討バージョンが「未確定な情報」および「整合性を確保するための情報」を格納するバージョンとなる。検討バージョンを利用することにより、プロダクトモデルの形のままで仮図に相当するようなモデル情報の伝達を行なうことができる。また出図前に検討バージョンにモデル情報の伝達を登録し、同時に電子メールで協議を他の設計者に依頼することにより出図情報の整合性を事前に確保することができる。このように造船CIMSにおけるモデル情報の伝達は黒板モデルのパラダイムに基づいた方法が有効と考えられる。

4.1.3 モデル情報の統合

前述のごとく公認バージョンは1隻につきただ一つのみ存在するので、同時並行的に情報を生成してきた各設計者は自分の作業バージョンの情報を公認バージョンに統合する必要がある。その際、本来一つである実体が公認バージョンと作業バージョンの両方で定義されている場合、単純に作業バージョンの情報を公認バージョンに付け加えただけでは二重にオブジェクトが定義されてしまう。そこで本論ではオブジェクトの意味的な共通性を表す「共通性 ID」という識別子を考案し、さらにバージョンおよびオブジェクトに所有者と言う属性を持たせた。これらを利用することにより、上記のような矛盾を起こさずにバージョンをマージすることができる。このバージョンのマージ機構については4.2.4項にて説明するが、このメカニズムにより複数のバージョンを統合することができるようになり、干渉チェックなどを含めたより高度な意味的な整合性を維持するための基盤が整備されたと言える。

4.2 協調活動支援実現のためのバージョン管理

4.2.1 造船CIMSにおけるバージョンの定義

本論では作業単位毎にグルーピングされたオブジェクトの集合をバージョンと名付ける。一般にバージョンと言うと、Fig. 6に示すように一つの対象に対して時間の経過と共に直列的に作成された複数の独立した情報のそれぞれを意味することが多い。しかし造船CIMSにおいては同時並行作業により始めから複数の並列的なバージョンが作成され、さらにFig. 7に示すようにそれらのバージョンの統合・分離を繰り返しながら設計作業が進められてゆく。そのようにして最終的にはただ一つの確定情報を持ったバージョンに統合されてゆく。

物オブジェクト・関係オブジェクト共に全てのオブジェクトは必ずいずれかのバージョンに所属する。設計者がバージョン単位で設計作業を進め情報を生成してゆくために、バージョン単位でのモデル情報のデータベースからの取り出しおよび格納ができる。

なお、バージョンのコピーを行なっても、そのオブジェクトは両方のバージョンに属す事になっただけで実体はただ一つしか存在せず、どちらかのバージョンのオブジェクトが変更された時に初めてオブジェクトのコピー（実体）

Fig. 6 Simple Version Transition

Fig. 7 Version Transition of CIMS

Fig. 8 Concept of CIMS Version Model

が作成される。

以後、本論では既存のバージョンに属する全オブジェクトをそれとは別の新たなバージョンに複製することをバージョンの「コピー」、2つの異なるバージョンの情報を結合することをバージョンの「マージ」と言うことにする。

4.2.2 オブジェクトの識別子

バージョンのマージを実現するためには、バージョンは異なるが本来一つの物を表すオブジェクトを認識する必要がある。そこで、データベース内の全てのオブジェクトに次の2種類の識別子を持たせることにする。

- ・オブジェクトID：そのオブジェクトそのものを指すIDで、データベース内でユニーク。OODBの場合はOODBMS自体が備えている機能により自動的に設定される。
- ・共通性ID：複数のバージョン間で共通のオブジェクトであることを示すためのIDであり、一つの共通性IDは現実世界の一つの実体に対応する。

一つのバージョンに属するオブジェクトは全て異なる共通性IDを持つことになる。バージョンのマージやコピーによって別バージョンからオブジェクトをコピーした場合、そのオブジェクトの共通性IDは元のオブジェクトと等しくなる。以後、共通性IDの等しいオブジェクトを互いに「対応するオブジェクト」と呼ぶことにする (Fig. 9参照)。

Fig. 9 Identification and Owner of Objects

なお一つの实体を複数の部品に分割した場合、他のバージョンとの関係ではそれらの部品は一つの共通性IDで識別される必要がある。そこで部品分割が行なわれた場合は共通性IDに子番号を付け、それらの部品間の識別を行なう。一方、他のバージョンとの間では親番号のみで識別を行なうことにより、分割前の部品と分割後の部品を対応させる。

4.2.3 バージョンおよびオブジェクトの所有者

データベース内の各バージョン情報およびオブジェクトにはそれぞれユニークに所有者が存在する。基本的にはバージョンまたはオブジェクトを作成した人がその所有者になるが、バージョンのマージやコピーによって他のバージョンからオブジェクトをコピーした場合は元のバージョンでのオブジェクトの所有者と同一になる (Fig. 9 参照)。

4.2.4 バージョンのマージ機構

以下のような規則に従って行なうバージョンのマージ機構を考案した。

- (1) マージするバージョンにはマージする側とされる側の区別がある。
- (2) マージする側のバージョンに属するオブジェクトのうちマージの対象となるのは、オブジェクトの所有者がバージョンの所有者と等しいオブジェクトのみとする。
- (3) マージする側のバージョンのマージ対象オブジェクトと対応するオブジェクトがマージされる側のバージョンに存在する場合は、マージする側のオブジェクトが優先される。
- (4) 前記(3)のルールによってマージされる側のオブジェクトがマージする側のオブジェクトによって置き換えられた場合、その元のオブジェクトを参照していたオブジェクトは参照先を置き換えられた新しいオブジェクトに変更する。
- (5) マージ対象オブジェクトがオブジェクトIDによって他のオブジェクトを参照している場合、その参照先のオブジェクトはそれ自身がマージ対象オブジェクトでなければコピーされない。よってマージされる側のバージョンに対応するオブジェクトが存在しない場合、参照先は空値となる。逆に

マージ前

マージ後

Fig. 10 Example of Version Merge

参照先のオブジェクトと対応するオブジェクトがマージされる側のバージョンに存在する場合は、参照先のオブジェクトIDをその対応するオブジェクトに変更する。

今、二つのバージョンが存在するとし、それぞれのバージョン名を“Version-A1”および“Version-B1”とし、バージョンの所有者はそれぞれ“A”氏および“B”氏とする。前記のような規則に従って“Version-A1”を“Version-B1”にマージした場合の、それぞれのバージョンに属するオブジェクトの様子を Fig. 10 に示す。

5. プロトタイピングによるマージ機構の有効性の確認

5.1 プロトタイプシステムの概要

前記の規則に従ってバージョンのマージを行うプロトタイプシステムをオブジェクト指向データベース (OODB) を用いて試作した。バージョン管理のためのメカニズムはオブジェクトのメソッドとして全て OODB 内に格納されている。ホスト言語としてはC言語を用いており、ユーザが対話式に画面からマウス・クリックによってメニューを選択してゆくことにより、バージョン操作が行なえるようになっている。ユーザがマウス・クリックにより指示した操作に対応したメッセージがCプログラムからデータベースに送られ、データベース内でメソッドによってバージョン操作が行われる。また、データベース内の幾何情報を三次元の鳥瞰図の形で表示することにより結果の確認が視覚によって容易にできるようになっている。さらに表示図形の部品をマウス・クリックすると、その部品と関係情報

によって結ばれている部品をデータベースから検索し色を変えて表示する機能も持たせた。この機能により、マージ後も部品間の関係情報の整合性が維持されていることを確認できる。

5.2 プロトタイプシステムによる検証

船殻設計と配管設計が並行して作業を行っているとの状況設定で、両者が情報の伝達を行ないながら作業を進め公認バージョンに登録して出図するまでの経過について、試作システムを用いてシミュレーションした経過を示す。なお、本プロトタイピングでのオブジェクトのクラス構成を Fig. 11, 作業の流れ及びバージョンの変遷を Fig. 12 に、また各バージョンのモデルの形状を Fig. 13 に示す。

- (1) 公認バージョン“Ver. O1”に外板とトランスのモールド面のみが登録されている状況で、船殻設計が“Ver. H1”という自分用の作業バージョンで船殻構造の設計作業を進めている。
- (2) 船殻設計は配管設計の要請により“Ver. H1”のモデル情報を仮図として“Ver. C1”という名称で検討バージョンに登録する。
- (3) 配管設計は船殻の情報を下絵情報として利用するために“Ver. C1”を自分用の作業バージョン“Ver.

P1”にコピーする。

- (4) 配管設計はパイプおよびパイプを貫通させるためのトランスの開口、さらにロンジから取ったパイプサポートの情報を“Ver. P1”に付加し、それを“Ver. P2”として登録する。ここで、パイプとトランスの間には孔明け関係が結ばれている。
- (5) 一方、船殻設計では設計作業の進行に伴いトランスの深さとロンジのサイズを変更し、新しい情報を“Ver. H2”として登録した。この時点で、配管設計が設計の前提条件とした船殻構造が最新の船殻構造と異なってしまったため、不整合が生じてしまっている。
- (6) 船殻設計は出図前の協議のために“Ver. H2”の情報を検討バージョン“Ver. C2”として登録するとともに、電子メールで他の設計者に協議を依頼する。
- (7) 船殻構造に変更があったとの連絡を受けた配管設計は“Ver. C2”を“Ver. P2”にマージして整合性が確保されている事を確認する。それから船殻設計に対して協議 OK の電子メールを送ると共に、マージ後のモデル情報を“Ver. P3”として登録する。これにより下絵とした船殻情報のうちトランスとロンジ

Fig. 11 Classes of Prototype System

Fig. 12 Example of Version Transition

Fig. 13 Geometric Shape of Each Version

に関する情報が最新のものと置き換えられる。また、パイプとトランスの間に結ばれている孔明け関係も新しいトランスとの間に結び直される。よって Fig. 14 に示すように、パイプをマウスでクリックするとマージ後もマージ前と同様にトランスのウェブプレートとパイプサポートの色が変わる。このことから“Ver. P2”においてパイプとウェブプレートとの間に付けられた孔明け関係がマージ後もオブジェクト自体は異なるが、対応するウェブプレートのオブジェクトに受け継がれていることがわかる。

- (8) 一方、船殻設計では協議 OK の電子メールを受け取り、“Ver. H2”を公認バージョンに登録する。公認バージョンには元々モールド面のみが登録されて

いたので、モールド面に船殻部材の情報が付加される形で登録され、“Ver. O2”となる。

- (9) 配管設計も同様に“Ver. P3”を公認バージョンに登録する。公認バージョンへの登録は既存の公認バージョンへのマージになるので、“Ver. P3”の情報のうちパイプとパイプサポートおよびそれに関連した関係情報のみが公認バージョンに取り込まれ、船殻情報などは二重定義されることはない。登録後の公認バージョンは“Ver. O3”となり Fig. 13 の最下端の図のようになる。

6. 結 言

本研究により以下の知見が得られた。

Fig. 14 Relations between Objects

- (1) チェックインチェックアウトメカニズムによる並行処理およびパフォーマンス向上と、オブジェクトの集合としてのバージョンの概念を統合したバージョン管理機構の形態が確立できた。
- (2) バージョン管理機構によって、協調設計を行なっている設計者間においてプロダクトモデルの形のままでの情報伝達の仕組みが確立できた。
- (3) バージョンのマージ機構を利用することにより、各設計者がバラバラに作業を進めても、そのバラバラな情報を順にマージしてゆくことによって最終的にプロダクトモデルレベルでは整合性の取れたただ一つの公認バージョンに統合してゆくためのメカニズムが確立できた。
- (4) オブジェクト指向データベースは情報をオブジェクト単位で扱うので、モデル情報の整合性の維持に関してはリレーショナルデータベースよりもプログラミングおよびメンテナンスが簡単になることが確認できた。

本研究でのプロトタイピングによる検証はごく簡単で小さなモデルを対象に行なっているので、より実船に近い複雑で巨大なモデルでも有効かどうか、および実用的なパフォーマンスが得られるかどうかについてさらに検討を続ける必要がある。さらに、本論においてはバージョンの

マージはバージョン単位でしかできないが、実際の運用を考えると任意の単位でオブジェクトを指定して他のバージョンにマージできるようにすることが必要と思われる。そのためのオブジェクトの指定の方法、および互いに参照し合っているオブジェクト集合の一部のみを取り出し、再び矛盾なく格納するためのメカニズムについても検討を行う必要がある。また、本研究では対象から外したモデルの意味的な矛盾まで含めた整合性維持の実現のためには、協調設計などに対する今後の学問的な研究成果を取り込んで行くことが重要と思われる。

謝 辞

本研究は、モーターボート競走公益資金による財団法人日本船舶振興会（会長笹川良一氏）からの補助金を受けており、関係各位に深謝します。また、ご指導いただいた東京大学小山健夫教授、ならびに造船 CIMS 開発研究委員会の皆様に感謝の意を表します。

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
- 2) 中井裕司他：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11月, pp.623~628。
- 3) 西尾章治郎: オブジェクト指向データベースシステムにおける並行処理制御, 情報処理 vol. 32, No. 5, 1991年5月, pp.540~549。
  - 4) 小山健夫: The Role of Computer Integrated Manufacturing for The Future Ship Building, IMSDC'91, 1991年5月, pp.55~60。
  - 5) (財)日本情報処理開発協会: 知的ハイパーテキストに関する調査研究 (編集版), 1991年5月。
  - 6) Raymond E. Levitt et al: Using Blackboard and CDPS Techniques to Support Concurrent, Multidisciplinary Design, 1991。
  - 7) Reid G. Smith: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver。
  - 8) Yan Jin and Takeo Koyama: Multiagent Planning through Expectation-based Negotiation, 10th AAAI International Workshop on Distributed Artificial Intelligence。
-