

5.3 製品モデルの柔軟性と拡張性

製品モデルの概念に基づくシステムはプロトタイプング手法を用いて構築されるため、製品モデルを構成する情報や機能には繰り返し修正や変更が加えられる。また、設計から生産に至る全ての情報や機能を一度に構築することは困難であるため、設計・生産過程の別々の部分で異なるモデルを構築し、その後モデルを統合することも多い。更にGPMEプロジェクトで指摘されているように、全ての造船所に共通な情報や機能のみでなく、各造船所固有の情報や機能があり、このような情報や機能は各造船所において構築することが必要である[GPME96]。以上のような製品モデルの特徴を考慮すると、製品モデルが柔軟で拡張性に富む構造であることは非常に重要である。本節では、本研究の製品モデルの柔軟性・拡張性について検討する。

5.3.1 製品モデルの柔軟性と拡張性の確保のために

製品モデルの柔軟性と拡張性として以下の2項目を満足することが重要と考えられる。

(1) 情報や機能の柔軟性と拡張性

先に述べたように製品モデルを構成する情報や機能には繰り返し修正や変更が加えられるため、情報や機能を容易に追加し変更できることは非常に重要である。

ところで、これまでの造船における製品モデルに関する研究は、製品情報及び製品情報を生成・利用するための機能の明確化を中心に行われてきた。これまでの研究で提案された製品情報としては、部材（部品）、接合関係情報、中間製品、モールド、部屋などがある。これらの情報は現実世界の「物」あるいは「物の集合」を表現する情報であるため、本研究でいう実体層の情報である。また、これまでの研究で提案された機能としては、設計機能やモデル操作言語によって記述された機能などがある。これらの機能は「物」あるいは「物の集合」を生成するための機能であり、本研究でいう実体層の機能と考えられる。即ち、これまでの造船における製品モデルに関する研究は、実体層の情報や機能の明確化が中心とすることができる。実体層の情報や機能の明確化が研究の中心となっていた理由として以下の項目が考えられる。

- ・ 属性層の情報や機能は形状情報や物理的情報などであるため、造船固有のものではなく、一般的なものであること。従って、その検討も造船のみで行うべきでなく、他分野を含めて行うべきであること。
- ・ 属性層の情報や機能の中心となる形状に関する表現方法や処理方法については、これまでに多方面で検討がされ、その成果を利用できること。また同様に名前（文字列）なども新たな検討が特に必要でないこと。
- ・ 製品モデルが「設計者の認識に対応する様々な情報」を有するモデルであり、人間の認識がオブジェクト指向で指摘されているように物中心であること。

従って、実体層の情報や機能の柔軟性・拡張性については特に十分な配慮が必要である。

(2)情報や機能の集合物の柔軟性と拡張性

3.3節において述べたように、製品モデルの構築では設計から生産に至る全ての情報や機能を一度に構築することは困難であるため、設計・生産過程の別々の部分で異なるモデルを構築し、その後モデルを統合することも多い。このようにして開発されたシステムでは、統合の対象となる各部分は必ずしも1つの情報や機能のみによって表現されているのではなく、複数の情報や機能の集合物として表現されることもある。従って、製品モデルの柔軟性と拡張性について検討する時は、個々の情報や機能について検討するのみでは不十分であり、情報や機能の集合物の柔軟性と拡張性を考慮することが必要である。

5.3.2 情報や機能の柔軟性と拡張性

本研究における情報の定義方法は大きく分けて2種類に整理できる。1つは属性情報の定義である。属性情報は本研究における最小単位となっており、属性情報についてはこれまでと同様にその情報の表現に何が必要かを考慮し、それに適した構造で情報を定義する必要がある。もう1つは実体層の情報の定義である。実体層の情報は4.2節において述べたように、他の情報との関係付けによって定義される。言い換えれば、実体層の情報は参照関係の定義によって定義されるということができる。

次に本研究における機能の定義について考えてみる。機能の定義も情報の場合と同様に属性層の機能の定義と実体層の機能の定義とに大別できる。属性層の機能の定義は、属性情報の変換のために必要な情報処理手法について詳細に検討が必要である。これに対して実体層の機能の定義は、4.3節で述べたようにどのような機能を組み合わせて必要な機能を定義するか、言い換えればどのような制御関係を定義するかが重要となる。

以上より、情報や機能の柔軟性と拡張性については、関係付けによって情報や機能を定義することが製品モデルの柔軟性と拡張性にどのような影響を与えるかについて検討すればよいことになる。

製品モデルのように、情報と同時に情報間の関係が重要な場合は、定義する情報は関係する他の情報を参照する必要がある。このため、製品モデルを構成する各情報を関係をいわずに定義すると、情報の内部に他の情報との関係や関係の具体的な内容について記述する必要がある。従って、何らかの情報の追加や変更を行うと、情報の内部に他の情報との関係が記述されているため、その情報に関係する全ての情報が影響を受けてしまうといった問題が生じる。言い換えれば、情報がある一定の組み合わせを前提として定義され、そのような情報が組み合わせられて全体が構築される。このため、情報の追加や変更などの問題は、当初予期していた組み合わせ方を変更するということとなり、柔軟に対応することは困難である。以上は機能についても全く同様といえる。

一方、本研究では情報や機能は部品化されており、部品の変更は他の部分に影響を与えない。また、製品モデル全体がFig.5.3-1に示すように、情報や機能をノード、関係をリンクとするネットワーク構造によって構成される。このネットワークを構成する情報・機能・関係を4.6節で述べたように定義することによって、製品モデルを構成する情報や機能を自由に

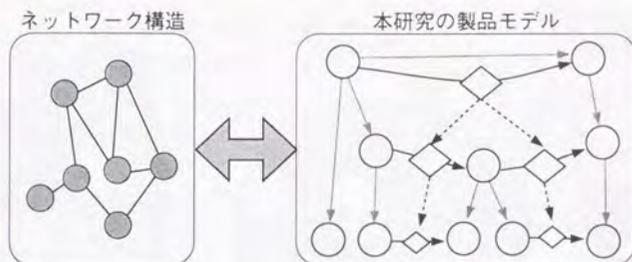


Fig.5-3-1 ネットワーク構造と本研究の製品モデル

組み合わせることが可能となり、組み合わせを変更しても他の部分に影響を与えない。従って、部品の変更も組み合わせの変更も自由に行えるため、情報や機能の追加や変更の問題に対して柔軟に対応することができ、情報や機能の柔軟性・拡張性が確保される。

5.3.3 情報や機能の集合物の柔軟性と拡張性

(1)モジュールの概念

前項までは部品化された情報及び機能を個別に定義することについて考えてきた。しかし造船の製品モデルのように大規模になると、情報や機能を個々にはではなく、それらの組み合わせとして取り扱いたい場合がある。例えば、部材を生成する機能や部材を分割する機能は、複数の機能が関係によって関係付けられた結合体として表現される。また、部材や中間製品も複数の情報が関係によって組み合わせられることによって表現される。これらを組み合わせ、新たな情報や機能を構築するためには、情報や機能の結合体を組み合わせることが要求される。このため本研究では情報や機能の結合体を縮約し、1つの纏まりとして取り扱うための概念としてモジュールを定義する(Fig.5-3-2)。そして、1つのモジュールを他のモジュールとを組み合わせることを考えてみる。

モジュールを組み合わせる際には、情報の生成に着目してモジュールを組み合わせる方法と、情報の利用に着目してモジュールを組み合わせる方法の2種類がある。

a)情報の生成に着目したモジュールの組み合わせ

情報の生成に着目してモジュール同士を組み合わせる際には、モジュール間の整合性を保持することが要求される。モジュールの整合性を保持するためには、モジュールを構成する個々の情報間の整合性を保持すること、及び、モジュール全体としての整合性を保持することの2点が要求される。そこで各々の整合性がどのように保持されるかについて検討する。

a1)情報間の整合性

本研究では情報の生成は変換関係を利用して行われる。従って情報間の整合性に関し

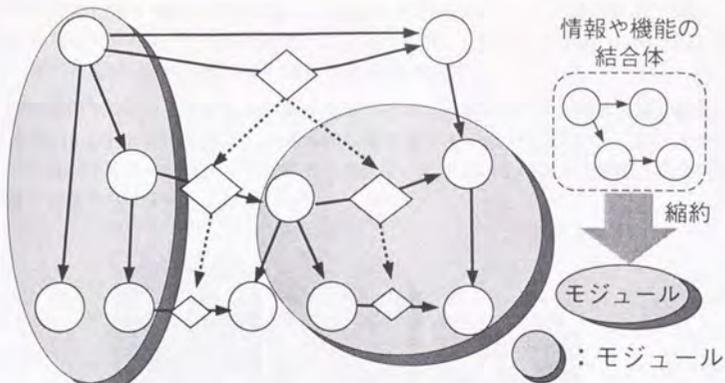


Fig.5-3-2 モジュールの概念

ては、変換関係によって保持される。このため、先ず変換関係を定義して情報間の整合性を保持する。

a2)モジュール全体としての整合性

本研究では、上位の層の機能と下位の層の機能とは制御関係によって関係付けられている。これによって、上位の層の情報と下位の層の情報との間の整合性が保持される。例えば、部材の分割機能が有する変換関係は、部材情報間の整合性を保持するのみであり、形状の分割機能が有する変換関係は形状情報間の整合性を保持するのみである。しかし、その2つの機能が制御関係によって関係付けられることによって、形状間の整合性を含めて、部材全体としての整合性が保持される。従って、個々の情報間の整合性の保持によって、モジュール全体としての整合性を保持することができる。

b)情報の利用に着目したモジュールの組み合わせ

本研究では情報の参照は、情報間の参照関係によって行われる。そして参照関係が存在すれば、関係の上流から下流へと複数の情報を辿り、必要とする情報が参照されることは先に述べた通りである。従って、複数のモジュールの情報の中から、参照が必要な情報の間に参照関係を定義すれば情報の利用が可能となる。

以上のように、情報や機能同士の結合が、全体としての結合として置き換えられる。従って、モジュールを組み合わせたい時には、モジュールを構成する個々の情報や機能に着目してそれらに関係付ければよい。言い換えれば本研究におけるモジュールの組み合わせは、モジュールを構成する要素の結合が縮約されることによって成り立っている。

(2)複数のモジュールの組み合わせ

3.3.5項で述べたように設計・生産過程の別々の部分で異なるモデルを構築し、その後モジュールを統合することも多い。設計・生産のある一部のモデルやモデルを生成するための機能

は、本研究では情報や機能が関係によって結合された結合体として表現される。即ち、設計・生産のある一部のモデルは、本研究では上記のモジュールとして定義される。従って、上記(1)で述べた方法に基づいて情報や機能を関係付けばよい。

例えば複数のモジュールを別々に定義するが、それらのモジュールの間で情報の参照が必要なときはFig.5-3-3(1)に示すように情報間に参照関係を定義すればよい。また上流のモジュールを基に下流のモジュールを生成するときは、Fig.5-3-3(2)のように機能と変換関係・制御関係を定義すればよい。

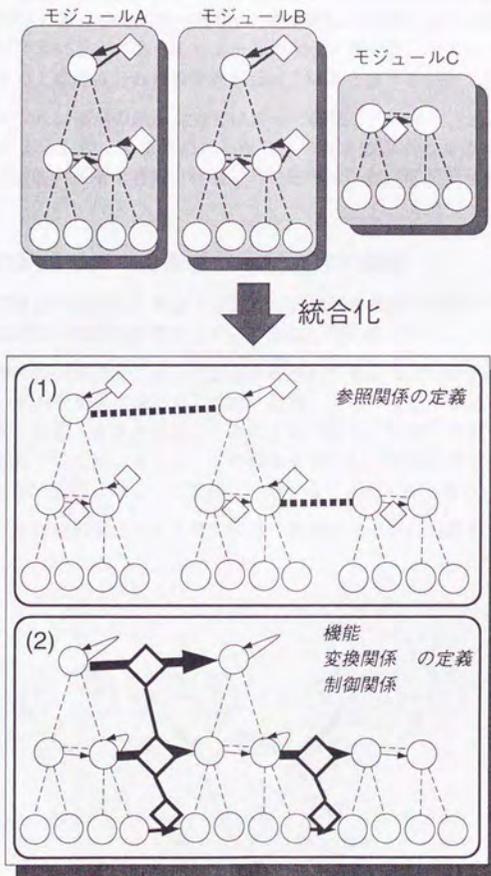


Fig.5-3-3 モデルの統合とモジュール

(3)要素に重複のあるモジュールの統合

上記(2)において述べたモジュールの統合は、モジュールを構成する要素に重複のないものを対象とした。しかし、実際にモジュールを統合する際にはモジュールの構成要素に重複があることもある。このため、モジュールの組み合わせでは構成要素の重複を考慮することが重要である。

Fig.5-3-4に示すような構造のモジュール1とモジュール2とを統合する場合には、情報Bと情報B'との重複、及び情報Dと情報D'との重複を考慮しなければならない。このような時は本研究では関係を付け替えることによって対応する。Fig.5-3-4の例では参照関係a・変換関係cを情報B'から情報Bに付け替え、参照関係b・変換関係dを情報D'から情報Dに付け替えている。言い換えれば要素に重複のあるモジュールの統合の際には、モジュールを構成する要素の対応を考え、その上で組み合わせの変更を行っていると言うことができる。

本研究の製品モデルは部品の組み合わせによって構築されており、更に部品の組み合わせの変更が柔軟に行える。従って重複のあるモジュールを組み合わせるとい問題についても、本研究の表現方法の長所を有効に利用して対応することが可能である。

5.3.4 本研究の製品モデルの柔軟性と拡張性の検証

製品モデルの柔軟性や拡張性を考慮する際には、情報や機能の柔軟性や拡張性と、情報や機能の集合物の柔軟性や拡張性の両方を考慮することが必要である。

本研究では情報や機能が関係によって組み合わせられて製品モデルが定義され、自由な組み合わせと組み合わせの変更を可能とする情報・機能・関係の構造を検討している。従って、情報や機能の追加・削除・変更を容易に行うことができる。更に、情報や機能の集合物の柔軟性・拡張性を確保するためにモジュールの概念を導入し、複数のモデルの統合などの際に必要で大規模な拡張の問題に対しても柔軟に対応することが可能である。

以上によって、本研究の製品モデルの柔軟性・拡張性は十分に確保されていると考えられる。

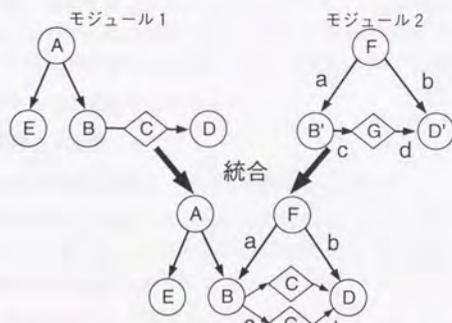


Fig.5-3-4 構成要素に重複のあるモジュールの組み合わせ

5.4 関係の利用

前節までは、本研究の製品モデルの表現方法によって、第3章で述べた製品モデルの表現方法に要求される項目がどのように満足されるかについて述べてきた。ところで、本研究の製品モデルの特徴の1つとして、情報や機能の間の「関係」が陽に記されていることが挙げられる。この「関係」を利用することによって、本研究の製品モデルの表現方法の有効性は更に増すものと考えられる。そこで本節では「関係」の利用方法について検討する。

5.4.1 関係同士の相性

本研究では参照関係・制御関係・変換関係（生成関係・抽出関係）を定義している。これらの関係は

- ・ 情報と機能に着目した整理
- ・ 製品モデルの内部と外部に着目した整理

を基に種類分けしたものである。前節までは上記の関係について個別に考えてきた。しかし関係は情報及び機能を接点として結び付くためお互いに関連があり、上記の2種類の整理に矛盾しないように定義されなければならない。従って何らかの関係を定義する際には、関係同士の関連を考慮することが重要である。本研究ではこのような関係同士の関連を関係同士の相性として捉える。本節では関係同士の相性について上記の2つの視点より検討する。

5.4.2 情報と機能に着目した関係同士の相性

情報の構造は参照関係によって表現され、機能の構造は制御関係によって表現される。これらが独立していれば、情報の構造や機能の構造は自由に定義することができ、参照関係や制御関係の相性について検討する必要はない。しかし情報と機能とは変換関係によって関係付けられる。このため、情報及び機能に矛盾が生じないように参照関係・制御関係・変換関係の相性について考慮することが必要である。

例えば、部屋を基に部材を生成することを考えてみる。Fig.5-4-1に示すように、部屋の情報は立体形状・名前との間に参照関係があり、部材の情報は面分・材質・名前との間に参照関係がある。部屋を基に部材を生成するためには、以下の関係の定義が必要である。

- a) 部屋から部材への変換関係を定義する
- b) 属性層の変換関係を定義する
- c) 実体層の機能と属性層の機能との間の制御関係を定義する

ここで、実際に部材の生成機能と制御関係を定義することが可能な属性層の機能について考えてみる。Fig.5-4-1の例において制御関係を定義することが可能なのは、(1)と(3)のみである。上位の層の機能と組み合わせが可能な下位の層の機能には、以下の2点が求められる。

- ・ 上位の層の変換関係の入力情報から下位の層の変換関係の入力情報を参照できること。

・下位の層の変換関係の出力情報が、上位の層の変換関係の出力情報の一部となっていること。

上記の2点を満足するためには、Fig.5-4-2に示すように変換関係と参照関係によって閉じた構造（ループ）となっている必要がある。変換関係と参照関係によって閉じた構造にならない機能を制御関係によって関係付けた場合には、関係の接点となる情報や機能に矛盾が生じ、情報を生成することが不可能となる。

以上のように情報の生成に着目すると、参照関係・制御関係・変換関係の構造を考慮して、複数の関係の接点となる情報や機能に矛盾が生じないように関係を定義する必要がある。言い換えれば、参照関係・変換関係との相性を考慮して制御関係を定義することが必要である。

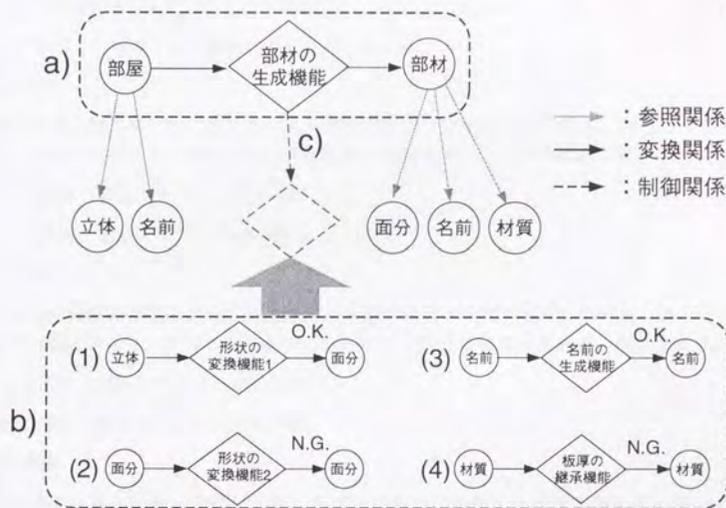


Fig.5-4-1 制御関係の定義と関係の相性

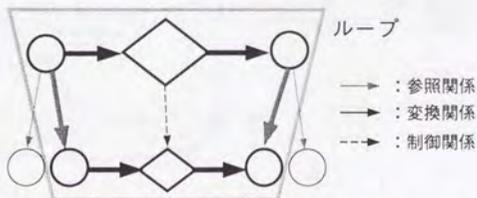


Fig.5-4-2 変換関係と参照関係によるループ

5.4.3 製品モデルの内外に着目した関係同士の相性

本研究では、情報を製品モデルの内部の情報と外部の情報とに分類し、それに伴い変換関係も製品モデルの内部の情報を生成する生成関係と製品モデルの外部の情報を生成する抽出関係とに分類している。この分類を基に関係同士の相性について考えると、ある関係によって製品モデルの内部とされた情報は、他の関係においても製品モデルの内部となっていなければならない。また製品モデルの外部となる情報についても同様である。従って関係を定義する時は、情報が製品モデルの内部か外部かについて矛盾が生じないように考慮することが必要である。

本研究において情報同士を直接的に関係付ける関係は参照関係のみである。また生成関係及び抽出関係も機能を中間に介して情報—機能—情報で1セットとなり、情報同士を間接的に関係付けていると考えられる。そこで参照関係・生成関係・抽出関係の入力情報と出力情報について製品モデルの内外区分を明確にし、その上で関係同士の相性について検討する。

(1)関係の入力情報・出力情報の明確化(Fig.5-4-3)

a)参照関係

参照関係は製品モデルに記述される情報を取り出すための関係である。従って、この関係に記述される情報は入力情報・出力情報ともに製品モデルの内部の情報と考えられる。

入力情報：製品モデルの内部の情報

出力情報：製品モデルの内部の情報

b)生成関係

生成関係は既に製品モデルに記述される情報を利用して製品モデルの内部の情報を生成するための関係である。従ってこの関係に記述される情報は製品モデルの内部の情報である。

入力情報：製品モデルの内部の情報

出力情報：製品モデルの内部の情報

c)抽出関係

抽出関係はある情報を変換して製品モデルの外部の情報を生成するための関係である。

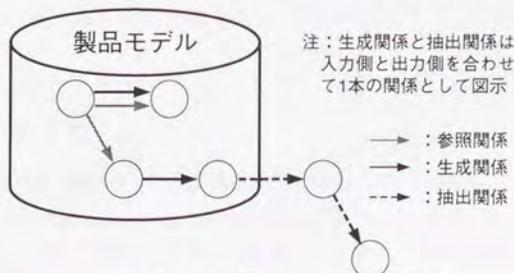


Fig.5-4-3 関係と製品モデルの内外

従って、この関係の出力情報は製品モデルの外部の情報と考えられる。抽出関係の例として面分の情報を基に面積を求める関係や、面積と厚さの情報を基に体積を求める関係などが挙げられる。ここで面分の情報を基に面積を求める関係の入力情報は面分であり、面分の情報は部材の形状などを表現する際に有効な情報である。従って、面分は製品モデルの内部の情報として捉えられる。また、面積と厚さの情報を基に体積を求める関係の入力情報は製品モデルの外部の情報の面積である。従って、抽出関係の入力情報は製品モデルの内部の情報・製品モデルの外部の情報のどちらでも取りえる。

入力情報：製品モデルの内部・外部の制限を受けない

出力情報：製品モデルの外部の情報

(2)関係同士の相性の検討

上記(1)において述べた入力情報・出力情報の内外区分を基に関係同士の相性について考えてみる。関係の整合性に問題が生じるのは、ある関係によって製品モデルの内部とされた情報が、他の関係によって外部とされる場合である。情報が製品モデルの内部とされるのは参照関係・生成関係の入出力情報であり、情報が製品モデルの外部とされるのは抽出関係の出力情報である。従って、抽出関係の出力情報に参照関係または生成関係がある場合に、情報の整合性に問題が生じることが理解できる。

5.4.4 関係同士の相性の利用

情報や機能の矛盾のない定義や製品モデルの内外区分は、製品モデルを定義する際に検討すべき重要な項目である。これまでは、情報や機能の間の関係が定義されていなかったため、一つ一つの情報や機能を定義する際に個々に検討する必要があった。しかし、本研究のように関係を明確に意識して関係を種類分けすることによって、情報や機能の矛盾のない定義や製品モデルの内外区分が、関係同士の相性として把握できることになる。従って関係同士の相性を利用することによって製品モデルの定義を支援することが可能である。具体的には、何らかの関係を定義する際に関係同士の相性をチェックする。そしてチェックの結果、問題が生じた場合には、どのような問題が生じたか、またどのような再構成が必要かを製品モデルの開発者に教えることが可能と考えられる(Fig.5-4-4)。

5.5 本章のまとめ

本章では、本研究の製品モデルの表現方法の有効性について検討した。

5.1節では、本研究の製品モデルの表現方法の特徴について整理した。これによって「製品モデルが情報と機能の構成として捉えられること」及び「情報や機能の部品化が実現されていること」を示した。

5.2節では、本研究の表現方法によって製品モデルの多面性が表現されるか否かを検討した。このために、情報や機能の詳細度に着目する視点、設計・生産過程（モデルの成長過程）に着目する視点、情報の生成・利用・管理に着目する視点など、様々な視点から本研究の表現方法を捉え直し、製品モデルの多面性を表現する際に関係の種類分けが有効であることを示した。

5.3節では、本研究の製品モデルの柔軟性と拡張性について検討した。本研究の製品モデルでは、関係を自由に定義・追加・変更・削除することが可能である。これによって情報や機能の追加・変更などの問題に対して柔軟に対応できることを示した。また、複数のモデルの統合などの問題に対処するために、モジュールの概念を導入した。

5.4節では、本研究の表現方法の特徴である関係の利用方法について検討した。本研究では関係を4種類に種類分けしているが、これらの関係の間には相性があり、この関係同士の相性を利用することによって、製品モデルの構築を支援できることを示した。

以上によって本研究の製品モデルの表現方法が有効であることが確認できた。

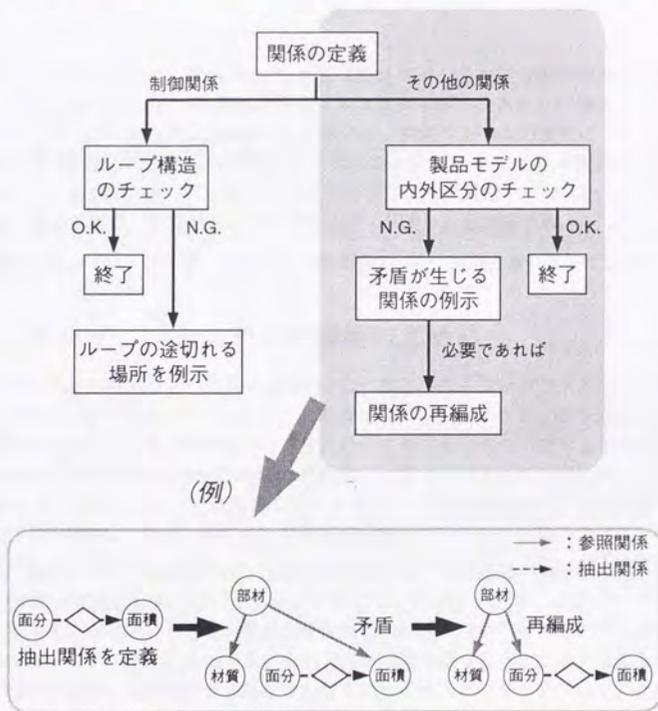


Fig.5-4-4 関係同士の相性の利用

第6章

プロトタイプ・システムの構築

前章までにおいて本研究の目的の1つである「製品モデルの表現方法の明確化」が整理された。そこで本章ではもう1つの目的である「プロトタイプ・システムの構築」について述べる。プロトタイプ・システムを構築するためには、先ずシステムの構成について検討し、その構成に基づいてプログラミングを行いシステム上に実装することが重要である。そこで、本章では、6.1節においてプロトタイプ・システムに何が求められるかについて整理し、6.2節において実際に計算機上にプロトタイプ・システムを実装する。

尚、構築されたプロトタイプ・システムの利用例については第7章において述べる。

6.1 プロトタイプ・システムの構築のために

これまでの造船の製品モデルに関する研究では、製品モデルはオブジェクト指向言語であるSmalltalkやC++などを用いてシステム上に実装されてきた。オブジェクト指向言語を用いると、情報はクラスとして、機能はクラスまたはメソッドとしてシステム上で表現される。そして実際の船を表現するための具体的な情報はインスタンスとして生成される。以上のように製品モデルをシステム上に実装すると、オブジェクト指向言語のクラス定義の特徴によって、情報や機能はプログラムによって直接的に結びつくことになる。

本研究の概念を用いて製品モデルを構築するためには、先ず情報や機能を定義し、次に定義された情報や機能を関係によって組み合わせることが必要である。上記したようにオブジェクト指向言語を用いると情報や機能が直接的に結び付くため、本研究の表現方法による製品モデルのシステム化のためにオブジェクト指向言語の環境を直接的に利用することはできない。情報や機能を関係によって組み合わせるためには、組み合わせのための仕組みと、組み合わせのための環境とが必要である。そこで本研究では、システムを開発する人々と組み合わせのための仕組みや環境を用意する人々とを明確に区別し、造船の製品モデルの関

Table 6-1-1 製品モデルの関係者の分類

設計者	構築された製品モデルを利用して船を設計する人
システム開発者	製品モデルを構築（情報や機能を定義）する人
開発支援者	情報・機能・関係の組み合わせのための仕組みや製品モデルの定義のための環境を提供する人

者を以下の3種類に分けて表現する(Table 6-1-1)。

(1)設計者：

構築された製品モデルを利用して、実際に船を設計する人。

(2)システム開発者：

製品モデルに必要な情報や機能を明確にし、情報や機能を組み合わせて製品モデルを構築する人。

(3)開発支援者：

システム開発者のために、情報や機能の組み合わせのための仕組みと環境を提供する人。

上記の分類によると、実際に情報や機能を組み合わせるのはシステム開発者である。しかし情報や機能を関係によって組み合わせるためには準備が必要であり、その準備を行うのが開発支援者である。本研究では著者は開発支援者の立場を取る。従って、本研究のプロトタイプ・システムは製品モデルを構築するためのシステムであり、開発支援者である著者は情報や機能の組み合わせのための仕組みと製品モデルを構築するための環境とをプロトタイプ・システムとして実装する。ここで、情報や機能の組み合わせのための仕組みについては4.6節で検討しているので、本節ではシステム開発者が製品モデルを構築するために必要な環境を中心に検討し、その検討に基づいてプロトタイプ・システムの構成を明確にする。

尚、システム開発者が使用する抽象的な情報と、設計者が使用する具体的なデータを有する情報とを明確に区別するため、本章以降ではシステム開発者が使用する抽象的な情報を「部材」「面分」などと記し、設計者が使用する具体的な情報を「部材A」「面分B」などと記す。また、単に「情報」と記した場合、システム開発者が使用する抽象的な情報を指す(Table 6-1-2)。これに伴い、システム開発者が抽象的な情報を作ることを「情報の定義」と呼び、設計者が具体的な情報を作ることを「情報の生成」と呼ぶ。開発支援者である著者はこれらの情報を直接的に扱うのではなく、これらの情報の組み合わせのための仕組みを検討することになる。

Table 6-1-2 システム開発者と設計者の扱う情報の分類

	扱う情報	扱う情報の記述方法	情報を作ること
設計者	具体的な情報	部材A、面分B	情報の生成
システム開発者	抽象的な情報	部材、形状	情報の定義
開発支援者	情報の組み合わせのための仕組みを検討		

6.1.1 製品モデルの定義のために

本研究の概念を用いて製品モデルを構築するためには、システム開発者は、様々な情報や機能を定義し、定義された情報や機能を関係によって組み合わせることが必要である。このため、システム開発者を支援するためには、以下に示す3項目について検討が必要である。

- (1)情報の定義の支援
- (2)機能の定義の支援
- (3)関係の定義の支援

ところで上記3種類の要素を組み合わせる製品モデルを構築する過程において、モジュールの概念が利用されることも考慮されなければならない。そこで本節では、情報・機能・関係の定義の支援とモジュールの概念の利用のために何が必要かを個別に検討し、その結果を踏まえてプロトタイプ・システムの構成を検討する。

6.1.2 情報の定義の支援のために

本研究では情報は他の情報や機能と関係によって関係付けられる。そして属性情報についてはその内部に属性情報が表現する具体的な内容を記述することが必要である。従って、情報の定義は以下の2つの部分から構成されると考えられる。

- a)情報を定義し、他の情報や機能との関係を定義する部分
- b)情報の内部に記述する具体的な内容について検討する部分（属性情報のみ）

そこで、a)及びb)を定義する2つの環境を用意することによって情報の定義は可能になると考えられる。言い換えれば、情報の定義と関係付けを専門に行う環境と、情報の内部に記述する内容の定義を専門に行う環境とを別々に用意し、その2つの環境を用いて情報を定義する方法である(Fig.6-1-1)。

まず、a)について考えてみる。全ての情報は他の情報や機能と関係によって関係付けられる。これは、全ての情報に共通する型として捉えられる。従って、この型に基づいて情報を定義することが、システム開発者を支援するために有効と考えられる。

次にb)について考えてみる。属性情報の表現内容は個々の情報によって異なるため、b)については個別に表現内容についての検討が必要である。従って、b)を定義する環境としては、どのような内容でも表現することが可能な自由度の高い環境が望まれる。

以上より、情報を定義するためには、以下の環境が必要と考えられる。

- ・型に基づいて情報を定義する環境
- ・属性情報の表現内容を定義するための環境

6.1.3 機能の定義の支援のために

本研究における機能は、Fig.6-1-2に示すように他の情報や機能と関係によって関係付けられ、その内部に具体的な情報処理のための方法が記述される。そこで機能の定義も情報と同様に以下の2つの部分に分けて考える。

a)機能を定義し、他の情報や機能との関係を定義する部分

b)機能の内部に記述する具体的な情報処理について検討する部分

先ず、a)について考えてみる。全ての機能は他の情報や機能と関係によって関係付けられる。これは、全ての機能に共通する型として捉えられ、この型に基づいて機能を定義することにする。

また、b)の機能に記述すべき情報処理内容は面分の分割や名前の生成などの様々な情報処理を検討する必要がある。従って、b)を定義する環境としては、どのような情報処理でも表現することが可能な自由度の高い環境が望まれる。

以上より、機能を定義するためには、以下の環境が必要と考えられる。

- ・型に基づいて機能を定義する環境
- ・機能に記述する情報処理内容を定義するための環境

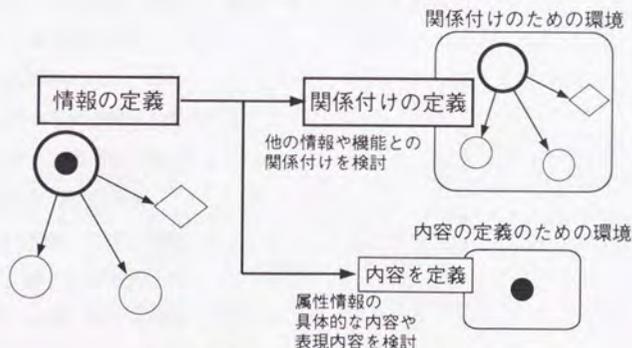


Fig.6-1-1 情報の定義の分類

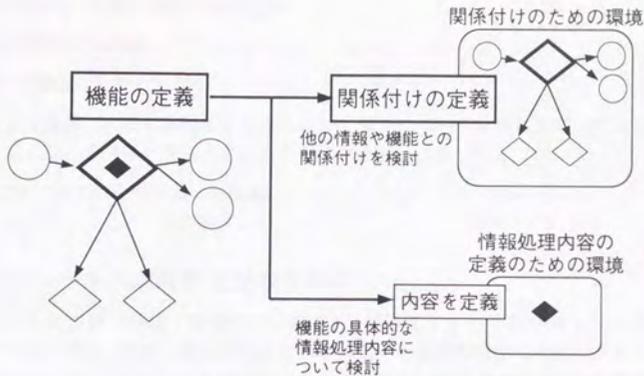


Fig.6-1-2 機能の定義の分類

6.1.4 関係の定義の支援のために

情報や機能を関係付ける関係を本研究では以下のように整理した。

(1)参照関係

入力：情報、出力：情報

入力側から出力側を参照できること

複数の関係を辿って情報を参照できること

(2)制御関係

入力：機能、出力：機能

入力側から出力側を参照し、制御することができること

(3-a)入力側の生成関係

入力：情報、出力：機能

入力側から出力側を制御できること

出力側から入力側を参照できること

(3-b)出力側の生成関係

入力：機能、出力：情報

(4-a)入力側の抽出関係

入力：情報、出力：機能

入力側から出力側を制御できること

出力側から入力側を参照できること

(4-b)出力側の抽出関係

入力：機能、出力：情報

以上は、各関係に共通する型として考えられ、関係についても型に基づいて定義することを考慮する。従って関係を定義するために以下の環境が必要である。

・型に基づいて関係を定義する環境

6.1.5 モジュールの定義の支援のために

モジュールは情報・機能・関係の結合体を1つの纏まりとして取り扱うための概念である。このため、情報・機能・関係が組み合わせられていく過程で自由に定義できることが重要と考えられる。また、モジュール同士の組み合わせは5.3.3項で述べたように情報や機能の組み合わせによって実現される。これらを考慮すると、モジュールの定義と組み合わせは、情報・機能・関係の定義や組み合わせと同じ環境で行えることが望ましいと考えられる。

6.1.6 製品モデルの定義の支援のために

本節で述べてきた製品モデルの定義のための環境をまとめると、以下の2種類に整理できる。

(1)組み合わせのための環境：

型に基づいて情報・機能・関係を定義し、それらを組み合わせで製品モデルを構築するための環境。この環境では、情報・機能・関係の定義とそれらの組み合わせが容易に行えることが要求される。モジュールも情報・機能・関係の纏まりとして定義されるため、モジュールの定義もこの環境において行うことが望ましい。

(2)情報や機能の内部に記述する内容の定義のための環境：

個々の属性情報の内部に記述する具体的な内容や、機能の内部に記述する具体的な情報処理内容を定義するための環境。様々な情報や情報処理の記述に耐えられる自由度の高い環境であることが要求される。

ここで、「(2)内容の定義のための環境」について検討してみる。属性情報の具体的な内容や機能の具体的な情報処理内容は、既存の船の製品モデルにも含まれており、特に不具合は指摘されていない。従って、既存の製品モデルの構築言語であるSmalltalkやC++などのオブジェクト指向言語の環境を利用できると考えられる。

そこで本研究において情報や機能の定義環境として特に検討が必要なのは「(1)組み合わせのための環境」であることが理解できる。

6.1.7 組み合わせのための環境について

本研究では、情報・機能・関係の定義と組み合わせを支援するために、組み合わせ環境を構築する。組み合わせ環境では、システム内に定義されている情報や機能はノードとして、関係はリンクとして表現され、全体がノードとリンクの結合体として定義される。このために情報や機能の容易な定義と組み合わせが可能である。具体的には、この環境は、情報、機能、関係、モジュールを参照する部分、型に基づいて情報・機能・関係を定義する部分、モジュールを登録する部分によって構成される。尚、表現の明確化のために、システム開発者の定義するノードやリンクを以降、メタノード、メタリンクと呼ぶ。

6.1.8 実際の船の表現のために

以上のようにして製品モデルが定義されると、設計者が実際の具体的なデータを有する船を設計することが必要である。このためには、システム開発者が定義したメタノード（情報・機能）やメタリンク（関係）を基に、実際の船を表現するための情報Aを生成することが必要である。

実際の船を表現する情報Aには以下の特徴があると考えられる。

- (1)システム開発者の定義する情報と、実際の船を表現する情報Aとの量的な比較を行ってみる。システム開発者が部材という情報を定義する場合、その数は1つである。しかし実際の船の表現のために必要な部材A、部材Bなどの数は数十万にも及ぶ。以上のように情報と情報Aとは1対多の関係にある。
- (2)本研究では情報を実体層・属性層の2つの層に分類した。この中で内部に情報の表現内容についての記述があるのは属性情報のみであり、実体層の情報には内部に情報の表現内容に関して記述する必要はない。このため、実際の船の具体的なデータを表現する際には、特に問題となるのは属性情報と考えられる。
- (3)システム開発者の定義する情報は、互いに関係によって関係付けられている。実際の船を表現するための情報Aも、この情報の構造に矛盾しないよう生成され管理されなければならない。

以上の特徴を考慮し、本研究では実際の船を表現する情報Aもノードとして生成し、それらをリンクによって関係付けて管理することにした。実際の船を表現するためのノードやリンクを「リアルノード」「リアルリンク」と呼ぶ。メタノード・メタリンクとリアルノード・リアルリンクとの関連をFig.6-1-3に示す。Fig.6-1-3に示すように、システム開発者の定義した情報の構造を有効に利用して実際の船を表現することが可能であり、リアルノードやリアルリンクの構造はメタノードやメタリンクの構造を基に幾つでも生成することができる。さらに実際の情報Aなどの生成の問題を属性層に限定することができ、上記(1)~(3)の特徴を十分に満足することが可能である。

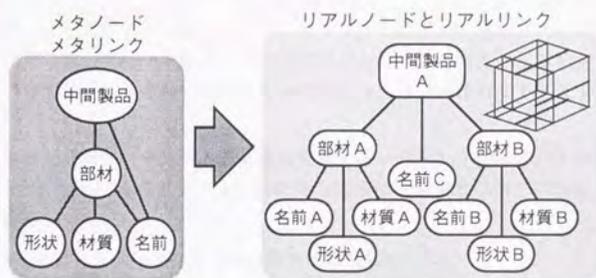


Fig.6-1-3 メタノードとリアルノード

6.1.9 システムの全体構成の明確化

これまでの整理を基にプロトタイプ・システムの全体構成について検討する。

先ず環境に着目すると、製品モデルを構築し、船を設計するためには以下の3つの環境が必要と考えられる。

- a) 実際の船を表現する情報を生成し管理する環境（設計者が利用）
- b) 情報・機能・関係を定義し組み合わせるための環境（システム開発者が利用）
（モジュールの登録もこの環境において行う）
- c) 属性情報の表現内容や機能の情報処理内容を定義する環境（システム開発者が利用）

これらの環境は以下のように考えることができる。

- a) 実際の船を表現する情報を生成し管理する環境は、メタノードやメタリンクの構造に合わせて定義するべきであり、システム開発者が定義すべきものと考えられる。
- b) システム開発者が利用する環境であるため、開発支援者が用意すべきである。
- c) 既存の言語の環境が利用できる。

従って開発支援者が用意すべき環境は、b)の環境である。

次に、各々の環境で設計者・システム開発者によって生成・定義されるのは以下のものである。

- a) 実際の船を表現する情報を生成し管理する環境
リアルノード、リアルリンク
- b) 情報・機能・関係を定義し組み合わせるための環境
メタノード、メタリンク、（モジュール）
- c) 属性情報の表現内容や機能の情報処理内容を定義する環境
属性情報の表現内容や機能の情報処理内容

上記の中で、メタノード・メタリンク、及びリアルノード・リアルリンクは型を基に定義・生成される。そして定義・生成されたノードやリンクは互いに組み合わせられなければならない。そこでこれらのノードやリンクの型に組み合わせのための仕組みを有させることによって、ノードやリンクを容易に作れるのみでなく自由な組み合わせを保証することができる。

以上を考慮すると設計者・システム開発者の利用するシステムを含めてシステムの全体像はFig.6-1-4のように整理することができる。開発支援者である著者は以下の項目を用意することが必要である。

- ・メタノード及びメタリンクの型（組み合わせを保証）
- ・リアルノード及びリアルリンクの型（組み合わせを保証）
- ・メタノード及びメタリンク（情報・機能・関係）を定義し組み合わせるための環境（モジュールの登録も可能）
- ・属性情報の表現方法や機能の情報処理方法を定義する環境（既存の言語を利用）

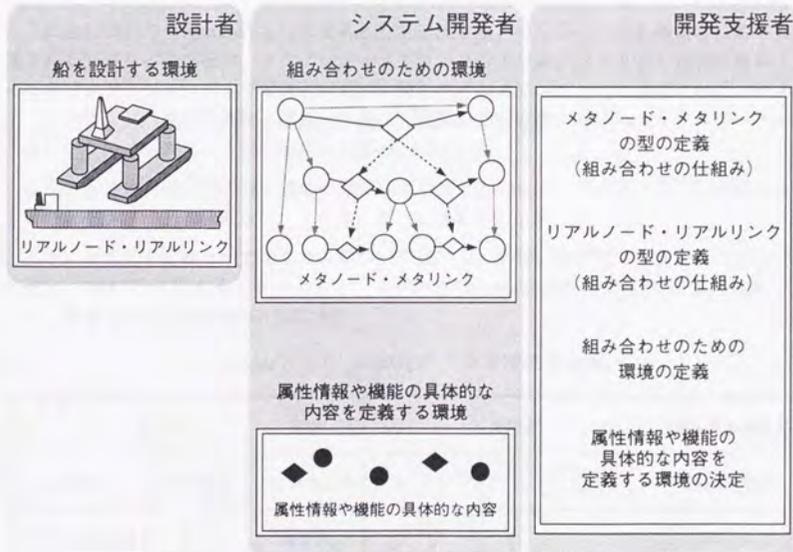


Fig.6-1-4 システムの全体構成の明確化

6.2 プロトタイプ・システムの実装

本節では前節の整理を基にプロトタイプ・システムを実装する。

本研究では、オブジェクト指向言語であるSmalltalk(Release4.1)の環境を利用して、その上に製品モデルの定義のための環境を構築した。

前節において述べたように、システム開発者が製品モデルを構築するためには以下の環境が必要であり、開発支援者はこれらの環境を構築することが必要である。

- a)メタノード及びメタリンク(情報・機能・関係)を定義し、それらを組み合わせるための環境(モジュールの登録も考慮)
- b)属性情報の表現方法や機能の情報処理方法を定義する環境

これらの環境の内、a)についてはSmalltalkを利用して、メタ・ブラウザというメタノード及びメタリンクを定義し、それらを組み合わせるための環境を構築した。また、b)についてはSmalltalkの環境を直接的に利用している。

また、本研究を用いて製品モデルを構築するためには以下の型を用意する必要がある。

- ・メタノードとメタリンク(情報・機能・関係)の型
- ・リアルノードとリアルリンクの型

これらについてもSmalltalk上にクラスとして定義する。そしてシステム開発者によって定義される部材などの情報は、メタノード・メタリンクのインスタンスとして定義されることになる(メタ・ブラウザ上で定義される)。そしてインスタンスとして定義されたメタノード・メタリンク(情報・機能・関係)を基に、実際の船を表現するためのリアルノードやリアルリンクのインスタンスが生成される(Table 6-2-1)。

また、属性情報の表現内容や機能の情報処理内容がSmalltalk上でクラスとして定義されるとメタ・ブラウザ上に登録され、メタノードに記述することが可能となる。

以下に本研究で定義した主なクラスについて述べる。また、本研究で定義したクラスの表記法は、OMT手法によるオブジェクト・モデル図[Rumbaugh92]を用いている。使用したハード環境はSun SPARCstation 20である。

Table 6-2-1 本研究のクラス定義の特徴

		通常	本研究		本研究の表記法
設計者	船の情報	インスタンス		リアルノード (インスタンス)	部材A、形状B
システム開発者	情報 (抽象的)	クラス	メタノード (インスタンス)		部材、形状
開発支援者	情報の型		メタノードの型 (クラス)	リアルノードの型 (クラス)	メタノードの型

6.2.1 メタノードとメタリンク

本研究では、情報・機能・関係の型を規定するクラスをSmalltalk上に定義した。この型によって情報や機能の組み合わせを保証している。メタノードやメタリンクには、次の種類がある(Table 6-2-2)。

- ・ 情報を表現するMeta_Information_Node
- ・ 属性情報を表現するMeta_Information_Node_Attribute
- ・ 機能を表現するMeta_Function_Node
- ・ 参照関係を表現するMeta_Link_Referential
- ・ 制御関係を表現するMeta_Link_Controll
- ・ 入力側の生成関係を表現するMeta_Link_Creation_Input
- ・ 出力側の生成関係を表現するMeta_Link_Creation_Output
- ・ 入力側の抽出関係を表現するMeta_Link_Extract_Input
- ・ 出力側の抽出関係を表現するMeta_Link_Extract_Output

以下各々のクラスについて述べる。また、これらのクラスのオブジェクト・モデル図をFig.6-2-1に示す。

(1)メタノード

a)Meta_Information_Node : Meta_Node-subclass

実体層の情報の型を規定するクラスである。4.6節で述べたように他の情報や機能との関係を記述するための器を有し、この器によって他の情報や機能との自由な組み合わせを保証している。このクラスを基に定義された部材情報を以下に示す。

Table 6-2-2 メタノードとメタリンクのクラス一覧

		クラス名		システム上での視覚化	
情報	実体層	Meta_Information_Node		ノード (球)	
	属性層	Meta_Information_Node_Attribute		ノード (球)	
機能		Meta_Function_Node		ノード (ドーナツ型)	
関係	参照関係		Meta_Link_Referential	リンク	
	制御関係		Meta_Link_Controll	リンク	
	変換関係	生成関係	入力	Meta_Link_Creation_Input	リンク
			出力	Meta_Link_Creation_Output	リンク
	抽出関係	抽出関係	入力	Meta_Link_Extract_Input	リンク
			出力	Meta_Link_Extract_Output	リンク

- 生成関係 (入力) : 立体の交面計算機能とのメタリンク
 (出力) : 面分分割機能、交線計算機能とのメタリンク
- 抽出関係 (入力) : 無し
 (出力) : 面積算出機能とのメタリンク

表現内容: Smalltalk上の面分を表現するクラスへのポインタ

c) *Meta_Function_Node* : *Meta_Node-subclass*

機能の型を規定するクラスである。機能も情報と同様に、他の情報や機能との関係を格納するための器を有する。また、機能の具体的な情報処理内容が記述される。

(例) 部材分割機能

- 変換関係 (入力) : 部材 (上記a)の例) とのメタリンク
 (出力) : 部材 (上記a)の例) とのメタリンク
- 制御関係 (入力) : ブロックの分割機能とのメタリンク
 (出力) : 面分分割機能とのメタリンク

情報処理: *Different_Level_Making*(6.2.2項a))

(2)メタリンク

全ての関係は方向性を有し、複数の情報や機能を関係付けるものである。これは関係に共通する型として捉えられる。そこで本研究では*Meta_Link*という抽象クラスを定義し、そのサブクラスとして以下の関係の型を定義している。また、変換関係 (生成関係と抽出関係の共通部分) の型を規定する*Meta_Link_Changing_Input*及び*Meta_Link_Changing_output*が*Meta_Link*のサブクラスとして定義されており、生成関係や抽出関係は更にそのサブクラスとして定義されている。

a) *Meta_Link_Referential* : *Meta_Link-subclass*

参照関係の型を規定するクラスである。参照関係には、関係付けの対象となる情報が記述される必要がある。この関係によって入力情報から出力情報を参照することが可能となる。例えば以下の例では、部材が面分を参照することが可能である。

(例) 部屋と面分との間の参照関係

- 入力: 部材 (実体層) を表現するメタノード(本項(1)a)の例)
 出力: 面分 (属性層) を表現するメタノード(本項(1)b)の例)

b) *Meta_Link_Controll* : *Meta_Link-subclass*

制御関係の型を規定するクラスである。制御関係には関係付けの対象の機能が記述される。この関係によって入力の機能は必要に応じて出力の機能を制御することが可能となる。

(例) 部材分割機能と面分割機能との間の制御関係

入力: 部材分割機能 (実体層) を表現するメタノード(本項(1)c)の例)

出力: 面分割機能 (属性層) を表現するメタノード

c) *Meta_Link_Creation_Input* : *Meta_Link_Changing_Input-subclass*

入力側の生成関係の型を規定するクラスである。生成関係の入力側には情報が記述され、出力側には機能が記述される。この関係によって情報は必要に応じて出力の機能を制御することが可能となり、機能は入力情報を参照することが可能となる。

(例) 部材と部材分割機能との間の入力側の生成関係

入力: 部材 (実体層) を表現するメタノード(本項(1)a)の例)

出力: 部材分割機能 (実体層) を表現するメタノード(本項(1)c)の例)

d) *Meta_Link_Creation_Output* : *Meta_Link_Changing_Output-subclass*

出力側の生成関係の型を規定するクラスである。

(例) 部材分割機能と部材との間の出力側の生成関係

入力: 部材分割機能 (実体層) を表現するメタノード(本項(1)c)の例)

出力: 部材 (実体層) を表現するメタノード(本項(1)a)の例)

e) *Meta_Link_Extract_Input* : *Meta_Link_Changing_Input-subclass*

入力側の抽出関係の型を規定するクラスである。基本的には入力側の生成関係と同じ振る舞いをする。但し、生成関係と区別するために別のクラスとして定義されている。

(例) 面分と面積算出機能との間の入力側の抽出関係

入力: 面分 (属性層) を表現するメタノード(本項(1)b)の例)

出力: 面積算出機能 (属性層) を表現するメタノード

f) *Meta_Link_Extract_Output* : *Meta_Link_Changing_Output-subclass*

出力側の抽出関係の型を規定するクラスである。

(例) 面積算出機能と面積との間の出力側の抽出関係

入力: 面積算出機能 (属性層) を表現するメタノード

出力: 面積を表現するメタノード

6.2.2 属性情報のクラスと機能のクラス

これらのクラスの例については本システムの利用例を示す第7章において述べることにする。本項ではこれらのクラスがSmalltalk上で定義された後の振る舞いについて述べる。

(1)属性情報の表現内容

本研究では、属性情報の表現内容は全てSmalltalk上にクラスとして定義する。定義されたクラスは、6.2.3項に示すメタ・ブラウザ上の属性情報ライブラリに登録され、Meta_Information_Node_Attributeに格納することが可能となる。

(2)機能の情報処理内容

機能の情報処理内容もSmalltalk上にクラスとして定義する。定義されたクラスは機能ライブラリに登録され(Fig.6-2-2)、メタリンクに記述することが可能となる。登録される機能の例として、面分と面分との交線を計算する機能(属性層)、実体層の機能、などが挙げられる。尚、機能は情報を表現するためのクラスではなく、情報処理を専門に行うクラスである。従って、機能が使用される時に、一時的に機能のクラスのインスタンスが生成される。

a)Different_Level_Making : Object-subclass

4.4.3項において述べた「実体層の機能」を表現するクラスである。下位の層の機能を使用した後に、上位の層の情報を生成し、上位の層の情報(リアルノード)と下位の層の情報(リアルノード)を矛盾なく参照関係(リアルリンク)によって関係付けることが可能である。

6.2.3 メタ・ブラウザ

組み合わせ環境としてメタ・ブラウザ(Meta_Browser : Browser-subclass)を定義した(Fig.6-2-3)。メタ・ブラウザは以下に示すリストを有し、定義された情報や機能などを詳細に調べることが可能である。

- ・定義されたメタノードとメタリンク
- ・定義された属性情報のクラス(属性情報ライブラリ)
- ・定義された機能のクラス(機能ライブラリ)
- ・登録されたモジュール

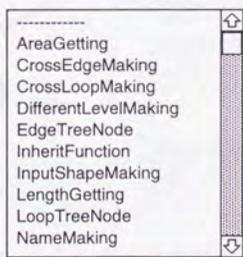


Fig.6-2-2 機能ライブラリ

メタ・ブラウザでは、6.2.1項に示したメタノードの型を基に、メタノードの定義や変更を行うこと、定義されたメタノードをメタリンクによって関係付けること、及び、モジュールの登録や組み合わせを行うことが可能である。

ところで、5.4節において関係同士の相性について述べた。この関係同士の相性を利用して様々なチェックを行うMeta_Relation_Checkというクラスを定義し、メタ・ブラウザからいつでも利用できるように構築した。

6.2.4 リアルノードとリアルリンク

実際の船を表現するためのリアルノードやリアルリンクの型としてReal_Node、Real_Node_Attribute、Real_Linkのクラスを定義した。これらのインスタンスはリアルノードやリアルリンクの型の構造と、メタノードやメタリンクとを基に生成される。このためリアルノードやリアルリンクの種類は、メタノードやメタリンクが記述されることによって表現される。例えば部材Aを表現するリアルノードのインスタンスには、部材を表現するメタノードのインスタンスが記述される。これによってリアルノードは自分が部材であることが分かる。また、属性層のReal_Node_AttributeはSmalltalk上の具体的なデータを有する属性情報のインスタンスへのポインタを有する。また、これらのクラスのオブジェクト・モデル図をFig.6-2-4に示す。

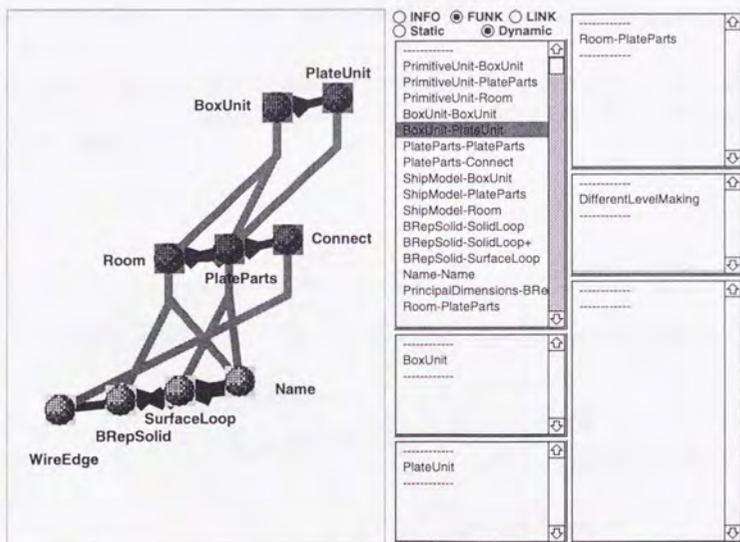


Fig.6-2-3 メタ・ブラウザ

6.2.5 リアルノードの生成と参照

本項では、本節でこれまでに述べてきた項目を用いて、実際の情報（リアルノード）の生成や参照がどのように行われるかについて述べる。

(1)情報の参照

ユニットを情報と参照関係によって表現した例をFig.6-2-5に示す。

- ・システム開発者はメタノードとメタリンクを用いてFig.6-2-5a)のようにシステムを構築する。
- ・実際のデータはリアルノードとリアルリンクによってFig.6-2-5b)のように表現される。
- ・Fig.6-2-5b)の各々のリアルノードとリアルリンクは、自分に対応するFig.6-2-5a)のメタノードとメタリンクを知っている。

情報の参照の流れを以下に示す(Fig.6-2-6)。

- 1)設計者によってユニットA（リアルノード）の部材（メタノード）を参照したいという要求が発せられる。
- 2)ユニットAはユニットに対して部材の参照の要求を発する。
- 3)ユニットは自分の有する参照関係（メタリンク）を選択し、各々の参照関係にユニットA（リアルノード）を始点として部材（メタノード）を参照する要求を発する。
- 4)各々の参照関係（メタリンク）は、自分の有するノードの中に部材があるか否かを確認する。
- 5a)部材がある場合は、ユニットAの有するリアルリンクの中の自分（ユニットと部材とのメタリンク）に対応するリアルリンクから部材A、部材Bを取り出し、取り出した部材A、部材Bを返す。

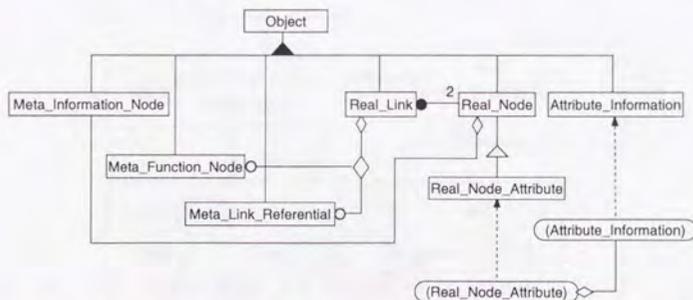


Fig.6-2-4 リアルノードとリアルリンクのオブジェクト・モデル図

5b)対象のノードがない場合（ユニットAの形状を取り出す時など）は、出力のリアルノード群（部材A、部材B）を取り出し、取り出したリアルノード群（部材A、部材B）に形状の参照要求を発生する。（あとは上記に示した流れが今度はユニットAではなく部材A、部材Bに対して行われ、形状A、形状Bが取り出される。）

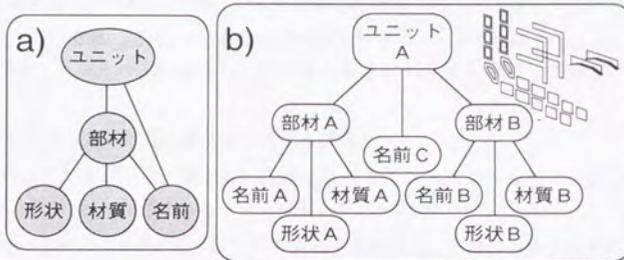


Fig.6-2-5 ユニットの表現例

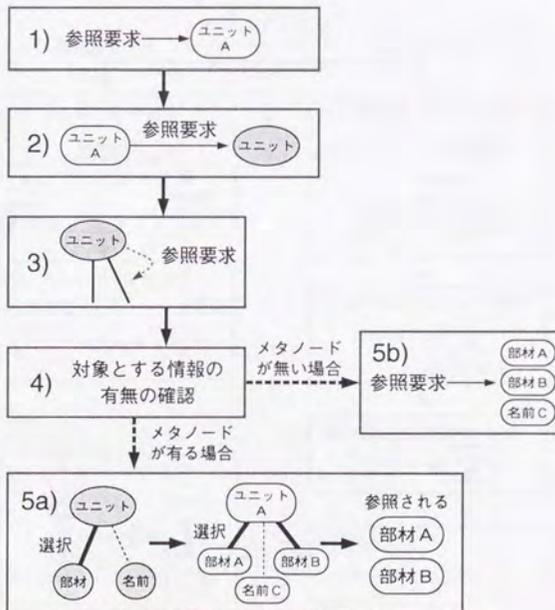


Fig.6-2-6 情報の参照の流れ

(2)情報の生成

Fig.6-2-5b)のような構造を実際に生成するためには、機能を利用してリアルノードやリアルリンクを生成することが必要である。リアルノードやリアルリンクの生成は以下のように行われる。(Fig.6-2-7)

- 1)設計者によって部材A（リアルノード）と1つの機能（メタノード）が選択される。
- 2)機能は、入力情報（部材A）を基にして出力のリアルノードを生成する。尚、機能による出力のリアルノードの生成を、属性層の場合と実体層の場合とに分けて以下に示す。
 - a)属性層の場合（形状を基に形状を生成する機能）
 - a1)入力のリアルノード（形状A）に記述されるインスタンスを基に、出力のリアルノード（形状B）に記述されるインスタンスを生成する。
 - a2)出力のメタノードに対応するリアルノードを生成し、そのリアルノードに上記a1)で生成されたインスタンスを記述する。

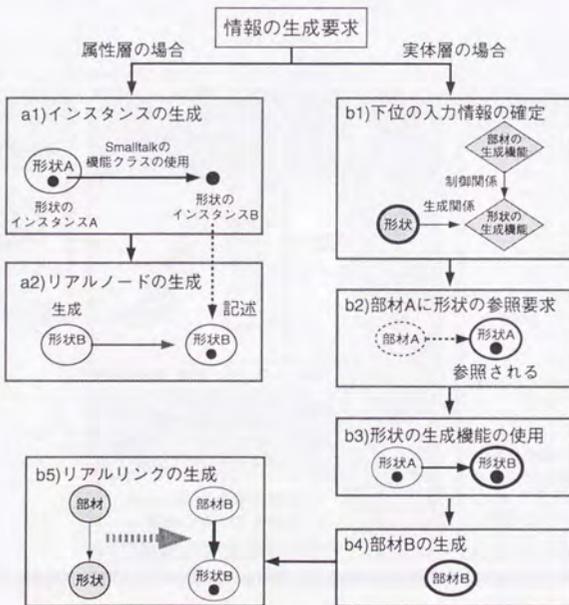


Fig.6-2-7 情報の生成の流れ

b)実体層の場合（部材を基に部材を生成する機能）

b1)自分の有する制御関係と、下位の機能の生成関係を用いることによって下位の機能の入力情報の種類（メタノード、例えば形状）が分かる。

b2)自分の入力情報である部材A（リアルノード）の有する参照関係を用いて、形状A（リアルノード）を取り出す。

b3)下位の層の機能を用いて形状B（リアルノード）が生成される。

b4)自分の出力のリアルノード（部材B）を生成する。

b5)出力のメタノード（部材）の有する参照関係（メタリンク）を基に、部材Bと形状Bとの間に参照関係（リアルリンク）を生成する。

6.2.6 プロトタイプ・システムの概要

本研究で実装したプロトタイプ・システムの全体像をFig.6-2-8に示す。Fig.6-2-8に示すように、本研究のプロトタイプ・システムでは、Smalltalk上で部品が定義され、部品の組み合わせはメタ・ブラウザ上で行われる。またメタ・ブラウザ上では、情報・機能・関係はノードやリンクとして表現され、視覚的にそれらの構成を把握することが可能である。

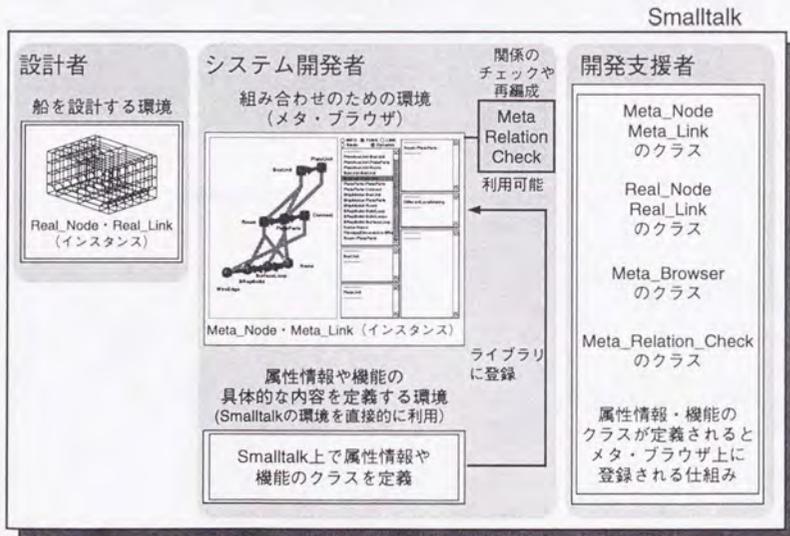


Fig.6-2-8 プロトタイプ・システムの全体像

ところで、5.4節で述べた関係同士の相性をチェックするMeta_Relation_Checkを、メタ・ブラウザからいつでも利用できる。このため、システム開発者は、メタ・ブラウザ上で視覚的に情報や機能を定義し組み合わせながら、それらの組み合わせの良否をチェックすることも可能である。

6.3 本章のまとめ

本章では、本研究の目的の2つめである「プロトタイプ・システムの構築」を行った。

6.1節では、本研究の表現方法に基づいて製品モデルを構築するために何が必要かを整理した。その結果、オブジェクト指向言語などを本研究の製品モデルの開発言語として直接的に用いることができず、製品モデルを定義するための環境と情報や機能を関係によって組み合わせるための仕組みから考える必要があることが分かった。そこで、本研究のプロトタイプ・システムを製品モデルを構築するためのシステムと位置づけ、プロトタイプ・システムの構成について検討を加えた。その結果、以下のシステム化が必要であると考えた。

- a)情報・機能・関係の型を定義し、組み合わせの仕組みを提供すること
- b)情報・機能・関係を定義し組み合わせるための環境を提供すること

6.2節では、6.1節の検討を基にSmalltalkを利用してプロトタイプ・システムを構築した。そして、プロトタイプ・システムを構築する際に定義されたクラスについて詳述した。

第7章

本システムを利用した 製品モデルの構築

本章では、本研究で構築されたシステムを用いると製品モデルがどのように構築され、どのように表現されるかについて述べる。7.1節では本システムを利用する場合の、製品モデルの構築の流れについて述べる。7.2節~7.4節では本システムを用いて実際に製品モデルを表現し、本研究の表現方法の有効性を実践の面から確認する。言い換えれば、本章ではシステム開発者の立場からプロトタイプ・システムの利用の面について述べる。

7.1 本システムを利用した製品モデルの構築の流れ

本節では、本システムを利用すると製品モデルがどのように構築されていくかについて述べる。尚、本システムを用いてシステムを構築する場合、属性層の情報や機能はSmalltalk上でクラスを定義した後に、メタ・ブラウザ上でメタノードを定義することが必要になる。Smalltalk上のクラスとメタ・ブラウザ上のメタノードを明確に区別するために、属性情報や機能の具体的な内容を表現するSmalltalk上のクラスを「属性情報のクラス」「機能のクラス」と呼び、メタ・ブラウザ上のメタノードを「属性情報のノード」「機能のノード」と呼ぶ。

7.1.1 システム開発者による情報の定義

情報は以下の流れで定義される。

(1)情報の分類

システム開発者は、定義する情報が実体層であるか属性層であるかを明確にする。

(2)情報の定義

a)情報が属性層の場合

定義しようとする情報を表現する属性情報のノードが既に定義されているか否かをメタ・ブラウザ上で調べる。

情報が定義されていない時は、Smalltalk上で属性情報のクラスを新たに定義する。これによって定義されたクラスが、メタ・ブラウザ上の属性情報ライブラリに登録される。そして、メタ・ブラウザ上で新たに属性情報のノードを定義し、属性情報のノードに属性情報のクラスを格納する。

b)情報が実体層の場合

定義する実体層の情報に必要な下位の層の情報を明確にする。メタ・ブラウザ上に必要な下位の層の情報のノードが全て定義されている時は、それらを組み合わせて実体層の情報のノードを定義できる。また、メタ・ブラウザ上に必要な下位の層の情報のノードが存在しない場合は、先ず下位の層の情報のノードを定義し、その次に実体層の情報のノードを定義する。

以上の流れを中間製品を例として述べる(Fig.7-1-1)。

- 1)中間製品の属する層について考える。中間製品は部材の集合であるので、実体層の情報である。
- 2)中間製品を表現するために必要な下位の層の情報について検討する。そして、部材と名前とが必要であることが分かる。
- 3)メタ・ブラウザ上で既に定義されているノードを調べる。その結果、部材を表現するノードが既に定義されていることが分かる。
- 4)名前のノードを先ず定義する。名前は属性層の情報であるため、Smalltalk上で名前を表現するクラス[Name]を定義する。その結果、メタ・ブラウザ上の属性情報のライブラリにクラス[Name]が登録される。そしてメタ・ブラウザ上で属性情報のノードを定義し、定義されたノードにクラス[Name]を記述する。
- 5)最後にメタ・ブラウザ上で中間製品のノードを定義し、部材ノード・名前ノードとの間に参照関係を定義する。

7.1.2 システム開発者による機能の定義

システム開発者による機能の定義について以下に示す。

(1)入力情報及び出力情報の明確化

機能の入力情報及び出力情報を明確にする。また、定義する機能が製品モデルの内部の情報生成するための機能か、あるいは製品モデルの情報を利用して製品モデルの外部の情報獲得するための機能かを考える。

(2)層に応じた機能の分類

定義する機能が実体層であるか属性層であるか、その属する層を明確にする。

(3)機能の定義

a)属性層の場合

定義する機能と同一の機能のノードが既に定義されているか否かをメタ・ブラウザ上で調べる。

定義されている時は、その機能のノードをそのまま利用すればよい。

定義されていない時は、機能のノードに記述する機能のクラスを新たにSmalltalk上で定義する。これによって機能のクラスが、メタ・ブラウザ上の機能ライブラリに登録される。次に、メタ・ブラウザ上で機能のノードを定義し、定義されたノードに機能のクラスを格納する。そして入力及び出力となる情報のノードを指定して機能のノードとの間に関係を定義する。この際に製品モデルの内部の情報を生成するための機能の場合は生成関係を定義し、製品モデルの情報を利用して製品モデルの外部の情報を獲得するための機能の場合は抽出関係を定義する。

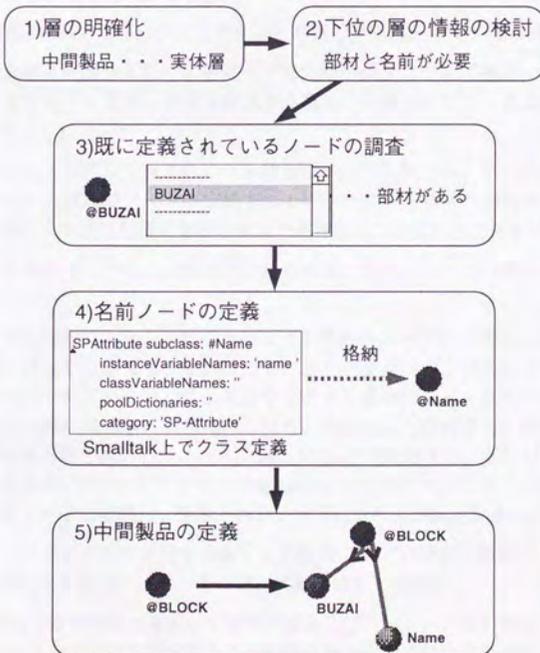


Fig.7-1-1 情報の定義の流れ

b) 実体層の場合

実体層の機能は以下の流れで定義される。

- ・ 定義する機能に必要な、下位の層の機能の把握
 - ・ メタ・ブラウザ上で機能のノードを定義
 - ・ メタ・ブラウザ上で実体層の機能のクラス (6.2.2項(2)a) を格納
 - ・ メタ・ブラウザ上で入力及び出力の情報のノードとの間に関係 (生成関係または抽出関係) を定義
 - ・ メタ・ブラウザ上で下位の層の機能のノードとの間に制御関係を生成
- (必要な下位の層の機能のノードが定義されていない時は、a)b)で述べた順序で下位の層の機能のノードを新規に定義する。)

以上の流れを部材の分割機能を例として述べる(Fig.7-1-2)。

- 1) 部材の分割機能の入力情報と出力情報を明確にする。この例の場合は入力情報、出力情報とも部材である。また、部材は製品モデルの情報であるため、製品モデルの内部の情報を生成するための機能と考えられる。
- 2) 部材の分割機能の属する層について考える。部材の分割機能は実体層の機能である。
- 3) 部材の分割機能を表現するために必要な下位の層の機能について検討する。そして例えば、面分を分割する機能、材質を継承する機能、名前を生成する機能が必要であることが分かる。
- 4) メタ・ブラウザ上で既に定義されている機能のノードを調べる。その結果、部材の分割機能を表現する機能のノードが定義されていないため、部材の分割機能のノードを定義し、機能ライブラリに既に登録されている実体層の機能のクラスを格納する。
- 5) 部材の分割機能のノードの入力側と出力側の両方に部材ノードとの間の生成関係を定義する。
- 6) 面分を分割する機能のノード、材質を継承する機能のノード、名前を生成する機能のノードが既に存在するかを調べる。この時、メタ・ブラウザ上に名前を生成する機能のノードが定義されていないため、名前を生成する機能のノードを先ず定義する。名前を生成する機能は属性層の機能であるため、Smalltalk上で名前を生成する機能の具体的な情報処理内容を検討し、クラス(Name_Making)を定義する。その結果メタ・ブラウザ上の機能のライブラリにクラス(Name_Making)が登録される。そしてメタ・ブラウザ上で機能のノードを定義し、定義されたノードにクラス(Name_Making)を記述する。
- 7) 最後にメタ・ブラウザ上で面分を分割する機能のノード、材質を継承する機能のノード、名前を生成する機能のノードとの間に制御関係を定義する。

ところで、本研究では層の概念を用いて情報が部品化され、これに伴い機能も部品化される。このため、ユニットなどを生成するための複雑な機能が、形状の処理機能、名前を生成する機能などの比較的単純な機能の組み合わせによって定義できる。更に機能の部品化が層

の概念を用いて行われるため、開発者による部品化方法の相違が生じず、あるシステム開発者によって部品化された機能を、他のシステム開発者が再利用することも可能となる。

7.1.3 システム開発者による関係の定義

本システムでは情報や機能の定義の際に、参照関係、生成関係、抽出関係、制御関係の4種類の関係(リンク)を必要に応じてノード間に定義することが要求される。これらの関係は、メタ・ブラウザ上で関係付けが必要な情報や機能のノードを選択し、関係の種類を指定することによって定義される。ここで、4種類の関係の関係付けの対象は以下のように整理できる。

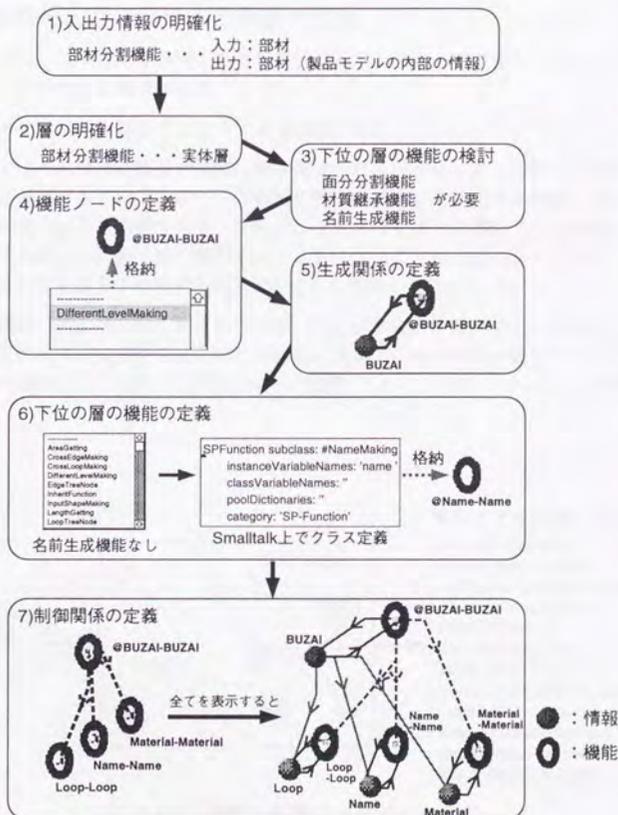


Fig.7-1-2 機能の定義の流れ

- ・参照関係 入力：情報、出力：情報
- ・制御関係 入力：機能、出力：機能
- ・生成関係 (入力側) 入力：情報、出力：機能
(出力側) 入力：機能、出力：情報
- ・抽出関係 (入力側) 入力：情報、出力：機能
(出力側) 入力：機能、出力：情報

この関係付けの対象に着目すると、参照関係と制御関係は特定できる。また、生成関係と抽出関係についても入力側か、あるいは出力側かについては特定できる。従って実際に関係の種類を指定する必要があるのは、生成関係と抽出関係との種類分けのみである。

7.1.4 関係を利用したシステム開発者の支援

本システムでは、関係及び関係同士の相性を利用してシステム開発者を支援することが考慮されている。具体的な支援内容を以下に示す。

(1)ある情報から生成・獲得することのできる情報の検索

製品モデルのように情報や機能が複雑に関係付けられる場合は、ある情報から直接的に取り出せる情報は分かりやすいが、幾つもの関係を経て間接的に獲得できる情報は分かりにくい。関係が無い場合は更に困難であり、プログラムを1つ1つ調べて獲得できる情報を確認することが必要である。本研究では、関係を辿ることによって、ある情報から直接的・間接的に獲得することが可能な全ての情報を例示することを考慮した(Fig.7-1-3)。

更に、ある情報(例えば部材)から他の情報(例えば重量)が獲得できるか否かについても関係を辿ることによって分かる。このため、ある情報から他の情報が獲得できるか否か、また獲得できる場合はどのような経路を辿って獲得できるかを例示することも考慮した(Fig.7-1-4)。

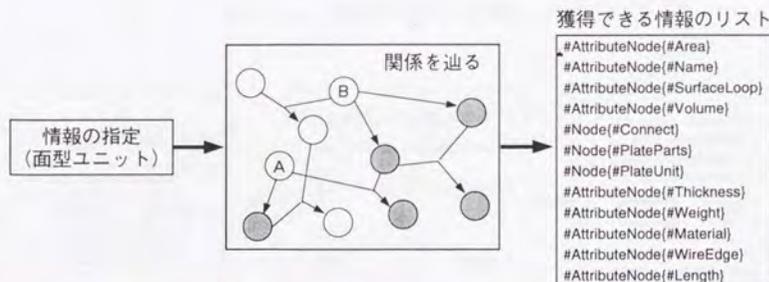


Fig.7-1-3 関係を利用した情報の検索(1)

(2)機能の組み合わせの支援

AからBを求める機能、AからCを求める機能、BからDを求める機能、CとDからEを求める機能があったとする。この4つの機能を組み合わせればAからEを求める機能を構築できる。

このような組み合わせの際には、機能の詳細な情報処理よりも、どの入力からどの出力が求められるかという関係が重要である。機能の有する変換関係を利用することによって、組み合わせが可能か否か、また組み合わせられる時は入力の情報と出力の情報が何か、どのような流れで組み合わせられるかが分かる(Fig.7-1-5)。

(3)制御関係の定義の支援

5.4.2項で述べたように制御関係と参照関係・変換関係との間には相性があり、制御関係を生成するときは関係する参照関係と変換関係が閉じた構造(ループ)となっていないなければならない。この相性を利用して制御関係の定義の際にチェックを行い、不適切な制御関係が定義されないように考慮されている(Fig.7-1-6)。

(4)関係のチェックと再編成

5.4.3項で述べたように、製品モデルの内外に着目すると抽出関係と他の関係との間に相性

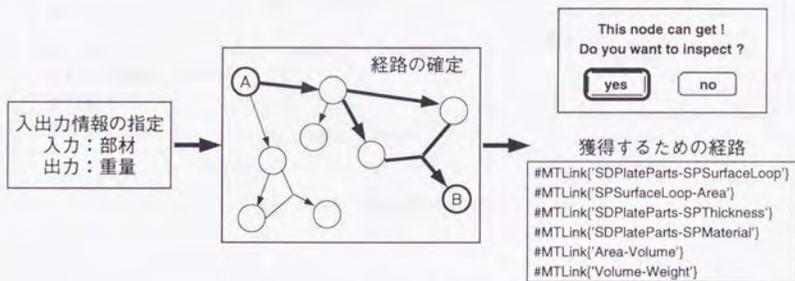


Fig.7-1-4 関係を利用した情報の検索(2)

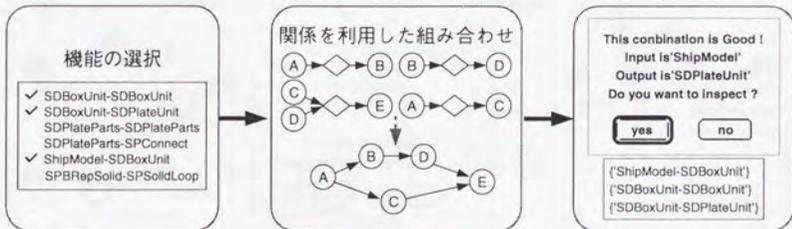


Fig.7-1-5 機能の組み合わせの支援

が存在する。このため、抽出関係が定義される時には、出力側の情報のノードの有する関係をチェックして、不適切な関係をシステム開発者に知らせる。参照関係についてはシステム開発者が望めば、システムが再編成を行う(Fig.7-1-7)。また参照関係や生成関係が定義される時も以上のようなチェックを行い、関係が適切か否かをシステム開発者に知らせることが考慮されている。

7.1.5 情報や機能の追加や変更

本システムでは、情報や機能の追加・変更は以下のように行われる。

(1)情報や機能の追加

一旦システムが構築された後に、情報や機能を追加する時は、7.1.1項、7.1.2項で述べたように情報や機能を新たに定義すればよい。本研究では、情報や機能が部品化されており情報や機能の独立性が確保されているので、既存のシステムに影響を与えずに情報や機能の追加が可能である。

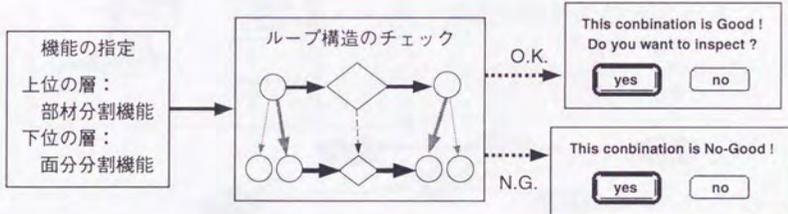


Fig.7-1-6 制御関係の定義の支援

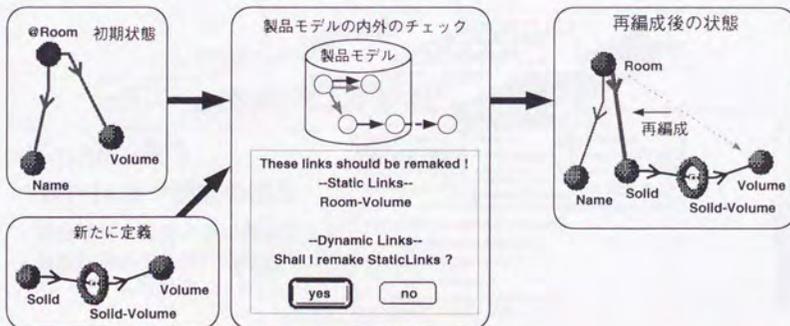


Fig.7-1-7 関係の再編成

(2) 組み合わせに影響を与えない変更

例えば、形状の定義方法や情報処理方法を変更する場合はこれに当たる。このような時は、Smalltalk上で新たにその情報や機能のクラスを定義する。そして、対応するノードに新たに定義したクラスを記述すればよい。このような変更はシステムを構成する部品の変更と考えられる。

(3) 組み合わせに影響を与える変更

現状の情報や機能の関係付けに問題が生じ、関係を変更する場合はこれに当たる。例えば、部屋の情報を基に部材情報を生成しているが、部材情報を部屋ではなく他の情報から生成する場合などである。このような時は、問題となる関係を削除し、新たに正しい関係を定義すればよい。このような変更は組み合わせの変更と考えられる。

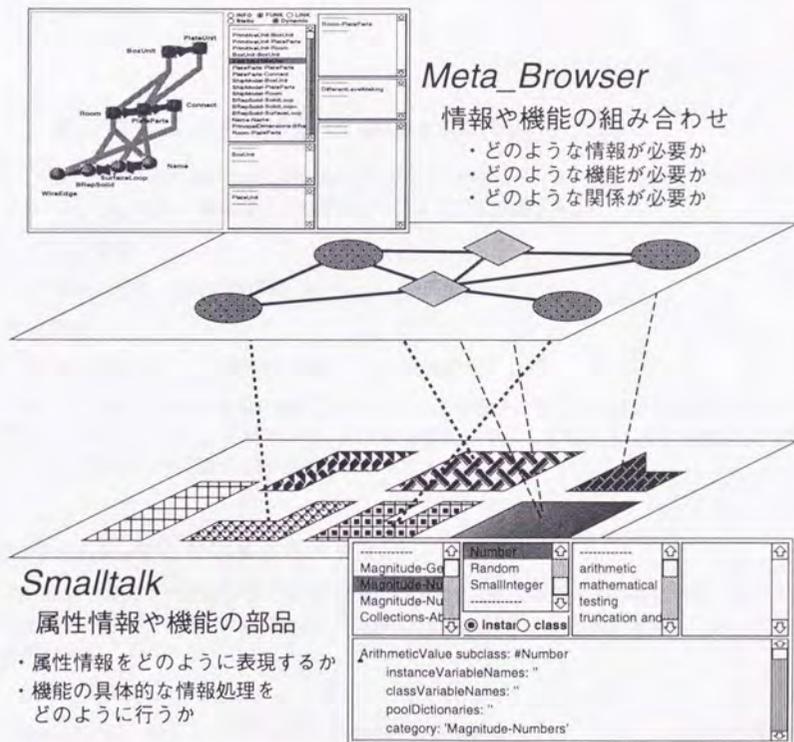


Fig.7-1-8 メタ・ブラウザとSmalltalk

ところで本研究の概念を用いると、製品モデルの概念に基づいたシステムの構築は、大別して2つの部分から構成される(Fig.7-1-8)。

1つはメタ・ブラウザを用いたノードとリンクの定義である。メタ・ブラウザを使用することによって「何の情報が必要か」「どの情報からどの情報を生成するか」「どのような機能が必要か」などの検討が支援される。即ち、メタ・ブラウザでは、製品モデルに必要な様々な情報や機能の組み合わせが検討される。

もう1つはSmalltalk上での属性情報や機能のクラスの定義である。属性情報や機能のクラスを定義するためには、例えば「形状の表現をどのように行うか」「形状の変換をどのように行うか」などが検討される。即ち、具体的な情報の表現内容や情報処理内容など、製品モデルを構成する部品の内容について詳細な検討が行われる。

このため、組み合わせの変更は部品に影響を与えず、部品の変更は組み合わせに影響を与えない。従って、製品モデルに対する情報や機能の追加や変更の問題に対して柔軟に対応することが可能となっている。

7.2 本システムの利用例(1) - SODASの製品モデル -

2.4節において述べたSODASの製品モデルのモデラ部を本システムを用いて表現する。具体的には、以下に示す製品情報と機能を本システム上に実装する。

- ・製品情報
 - 部材、部屋、接合関係情報、ユニット、中間製品（組立モジュール）
- ・機能
 - 空間設計機能、内部構造設計機能、カット機能

尚、7.2.1項に示す属性情報や機能のクラスがライブラリに登録されている状態で、メタ・ブラウザを用いてメタノードやメタリンクを定義し、それらを組み合わせてSODASの製品モデルを構築するのに要する時間は2時間程度である。

7.2.1 Smalltalk上に定義したクラス

SODASの製品モデルを実装する際にSmalltalk上に定義したクラスを以下に示す。本システムを用いると情報や機能が汎化され、SODASと比較するとクラス数が減少していることが理解できる。

(1)属性情報のクラス

SODASには、情報を表現するために19個のクラスが存在する。しかし、本研究で構築されたシステムでは、形状（立体、面分、線分）、材質、名前など、属性情報を表現する計8個のクラスのみである。これらのクラスを利用してメタ・ブラウザ上でメタノードやメタリ

リンクを定義し、それらを組み合わせることによって、既存のシステムと同様の情報を表現することが可能である。属性情報ライブラリに登録されている属性情報のクラスをTable 7-2-1に示す。尚、Table 7-2-1の形状に関するクラスは平面、円筒面を対象としている。

(2)機能のクラス

SODASには、空間設計機能、内部構造設計機能、カット機能という機能があり、更にこれらの機能を構成する細かい機能が多々ある。本研究ではこれらの機能を層の概念を用いて分類・整理し以下の機能のクラスを定義している。尚、機能ライブラリに登録されている機能のクラスの一覧をTable 7-2-2に示す。

a)実体層の機能のクラス

既存のSODASでは、実体層の機能は生成する情報毎に個別に定義されている（20個程度のクラスに分散）。本研究では、この機能も汎化され、6.2.2項に示した1個のクラスのみを定義した。

b)属性層の機能のクラス

本研究では交線計算機能、立体分割機能、名前の生成機能、材質の継承機能など計17個のクラスを定義した。これらの機能は既存のSODASに定義されている機能を整理して利用した。

7.2.2 製品情報と機能の表現

(1)製品情報の表現

メタ・ブラウザ上に表現された製品情報をFig.7-2-1に示す。Fig.7-2-1は情報のノードと参照関係によって表現されており、製品情報の構成が一目で分かるように表現されている。

Table 7-2-1 属性情報ライブラリのクラス一覧

クラス名	内容
Wire_Edge	線分を表すためのクラス
Surface_Loop	面分を表すためのクラス
Solid_Loop	面分（法線方向あり）を表すクラス
BRep_Solid	立体を表すためのクラス
Material	材質を表すためのクラス
Name	名前を表すためのクラス
Thickness	板厚を表すためのクラス
Princial_Dimensions	主要目を表すためのクラス
Number (Smalltalkに既成)	量を表すためのクラス (重量、面積、長さ等を使用)

Table 7-2-2 機能ライブラリのクラス一覧

クラス名	内容
Edge_Tree_Node	線分を分割するためのクラス
Loop_Tree_Node	面分を分割するためのクラス
Solid_Tree_Node	立体を分割するためのクラス
Cross_Edge_Making	交線を計算するためのクラス
Cross_Loop_Making	交面を計算するためのクラス
Surface-Solid_Changing	面分の集合を基に立体を獲得するクラス
Solid-Surface_Changing	立体を構成する面分を獲得するクラス
Wire-Surface_Changing	線分の集合を基に面分を獲得するクラス
Input_Shape_Making	設計者の入力した 線分や面分を生成するためのクラス
Pillar_Shape_Making	軸と断面を基に立体を獲得するクラス
Name_Making	名前を生成するためのクラス
Inherit_Function	材質・板厚などの情報を継承するためのクラス
Length_Getting	線分の長さを獲得するためのクラス
Area_Getting	面分の面積を獲得するためのクラス
Volume_Getting1	立体の体積を獲得するためのクラス
Volume_Getting2	面積と高さから体積を獲得するためのクラス
Weight_Getting	体積と比重(材質)から重量を求めめるクラス
Different_Level_Making	実体層の機能のクラス

(2)機能の表現

メタ・ブラウザ上に表現された設計機能をFig.7-2-2に示す。Fig.7-2-2のそれぞれの設計機能は、機能のノードと制御関係によって表現されている。空間設計機能、内部構造設計機能、カット機能は各々異なる設計機能である。しかしFig.7-2-2に示すように部分的には共通部分が多く、定義されたメタノードやメタリンクが有効に再利用されていることが理解できる。

(3)製品情報と機能との関係

空間設計機能、内部構造設計機能、カット機能を生成関係に着目して表現した例をFig.7-2-3に示す。更に参照関係や制御関係を含めて表現するとFig.7-2-4のようになる(実体層に限定)。Fig.7-2-4より分かるように全ての関係を一度に把握することは困難である。現状ではこれらの関係の全てを暗に考えながら製品モデルに関する検討が行われている。本研究では関係を陽に意識し、かつ関係の種類分けを行ったことによって、Fig.7-2-1 ~ Fig.7-2-3に示したように簡潔にこれらの関係を把握することができる。

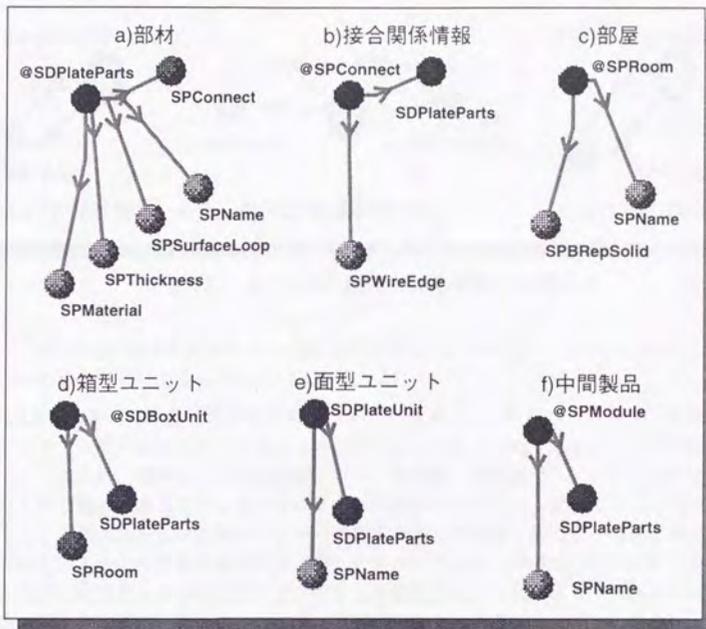


Fig.7-2-1 SODASの製品情報の表現例

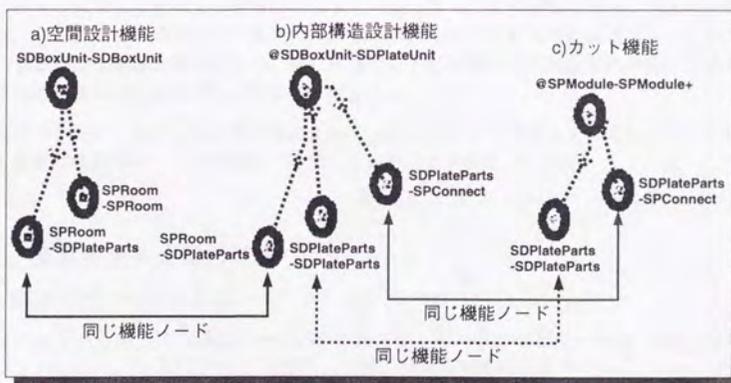


Fig.7-2-2 SODASの機能の表現例

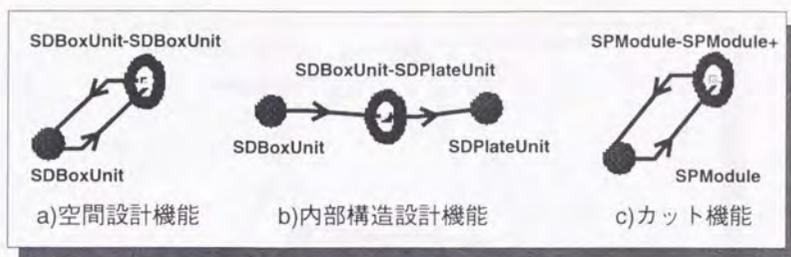


Fig.7-2-3 SODASの製品情報と機能との関係(1)

ここでSODASにおける表現方法と本研究の表現方法との相違を、箱型ユニット、面型ユニット、部材を例として述べる(Fig.7-2-5)。

SODASでは各々の製品情報は情報（データ）と機能（アルゴリズム）とが一体化されたオブジェクトとして定義されており、各々のオブジェクトは情報を中心として関係付けられている。このため、箱型ユニットと面型ユニットとの間、及び面型ユニットと部材オブジェクトとの間に関係があることは分かるが、各々の関係がどのような関係であるかは分からない。また、面型ユニットは箱型ユニットの情報を利用して生成されるが、情報と機能とが一体化されている。このため情報の生成の際にどのオブジェクトの機能が用いられるかは明確に分からず、オブジェクトの内部のプログラムを実際に見ることによって、初めて使用される機能が分かる。また、何らかのオブジェクトに記述される機能を変更する場合には、他のオブジェクトの有する機能と矛盾を生じないように変更しなければならない。

本研究の表現方法を用いると、箱型ユニットと面型ユニットは生成関係によって関係付けられ、面型ユニットと部材とは参照関係によって関係付けられる。これによって様々な情報がどのような関係にあるかを明確に認識することができる。また情報と機能とが別々に部品化され、関係によって関係付けられているため、Fig.7-2-5に示すように面型ユニットの生成に関与する機能を明確に認識できる。更に情報や機能が部品化され部品の独立性が確保されているため、各々の変更を柔軟に行うことができる。

以上のように、本研究の表現方法によって、製品モデルの構成を把握することが容易になり、変更の問題に対しても柔軟性が増加していることが理解できる。

7.2.3 製品モデル全体の表現

(1) 製品モデルの全貌の表現

本システムを用いて表現したSODASの製品モデルのモデラ部分の全貌をFig.7-2-6及びFig.7-2-7に示す。Fig.7-2-6は全ての情報、機能、関係を表示したものである。また、Fig.7-2-7では制御関係が省略され、機能と機能の入出力の生成関係とが1つのリンクによって表現されている。

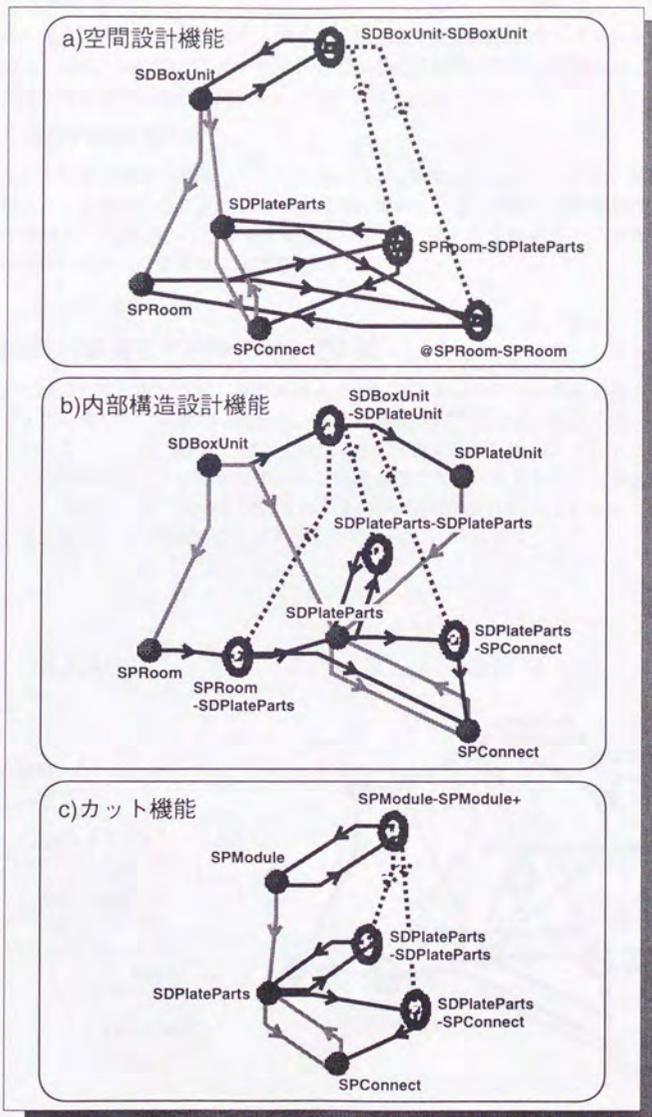


Fig.7-2-4 SODASの製品情報と機能との関係(2)

(2)モデルの成長の表現

Fig.7-2-6の実体層の上位の層の部分（情報、機能、及び生成関係に着目する）を示したものをFig.7-2-8に示す。Fig.7-2-8によって製品モデルの成長過程が明確に表現されており、各機能がモデルの成長のどの部分を担っているかを明確に理解できる。

(3)各々の製品情報の位置づけ

Fig.7-2-7の実体層の部分表現したものをFig.7-2-9に示す。ユニット、部屋、壁は上流の部分で使用され、中間製品は下流の部分で使用されている。また部材と接合関係情報は製品モデル全体を通じて使用されていることが分かる。以上のように製品モデル全体の中での各々の製品情報の位置づけが明確に理解できる。

7.2.4 SODASの製品モデルの表現のまとめ

7.2.2項、7.2.3項に示したように、本システムを用いると製品モデルを様々な視点から捉えることができる。また、7.2.1項7.2.2項から、情報や機能の部品化によって汎用性・再利用性が向上していることが理解できる。従ってSODASの製品モデルを本システム上に表現することによって本研究のシステムの有効性がある程度確認できたと考えられる。最後にこのように構築された製品モデルを用いて設計された船の例をFig.7-2-10に、またブロック分割を行い重量や溶接線長などの管理物量を取り出した例をFig.7-2-11に示す。

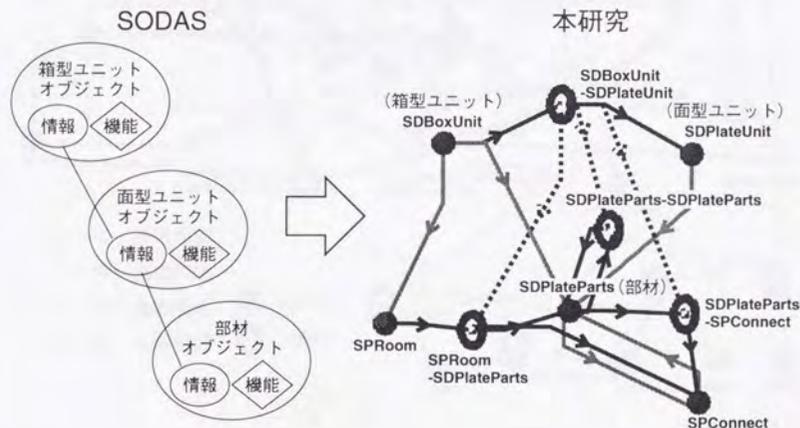


Fig.7-2-5 SODASの表現方法と本研究の表現方法との相違

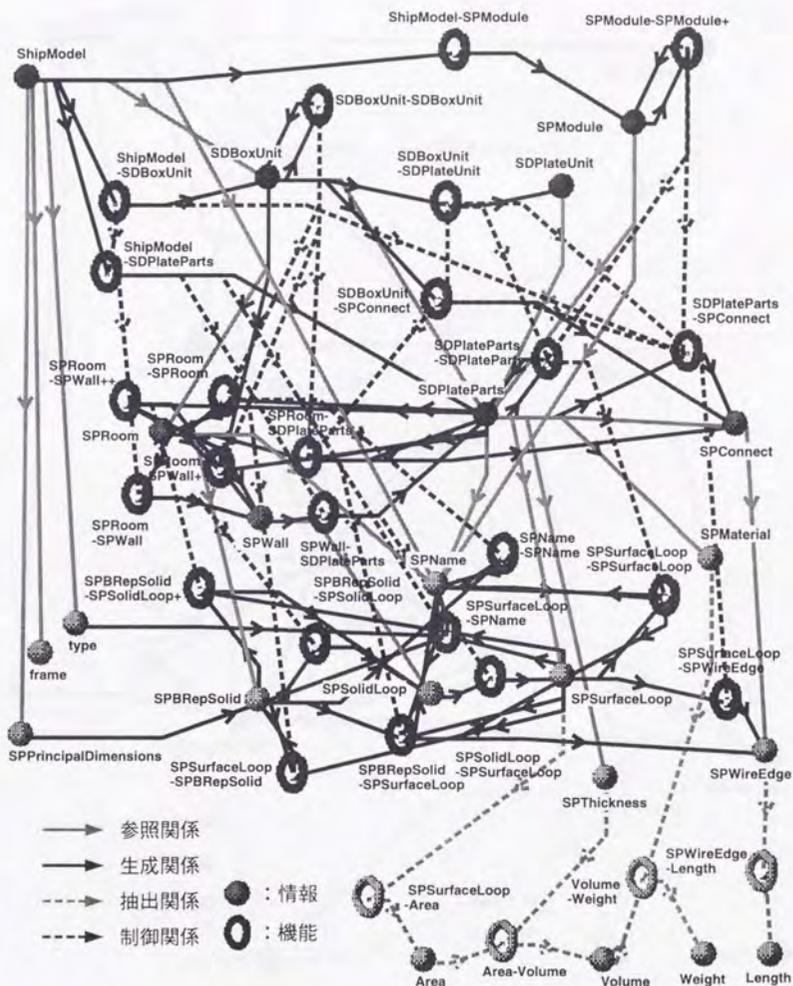


Fig.7-2-6 SODASの製品モデルの表現例(1)

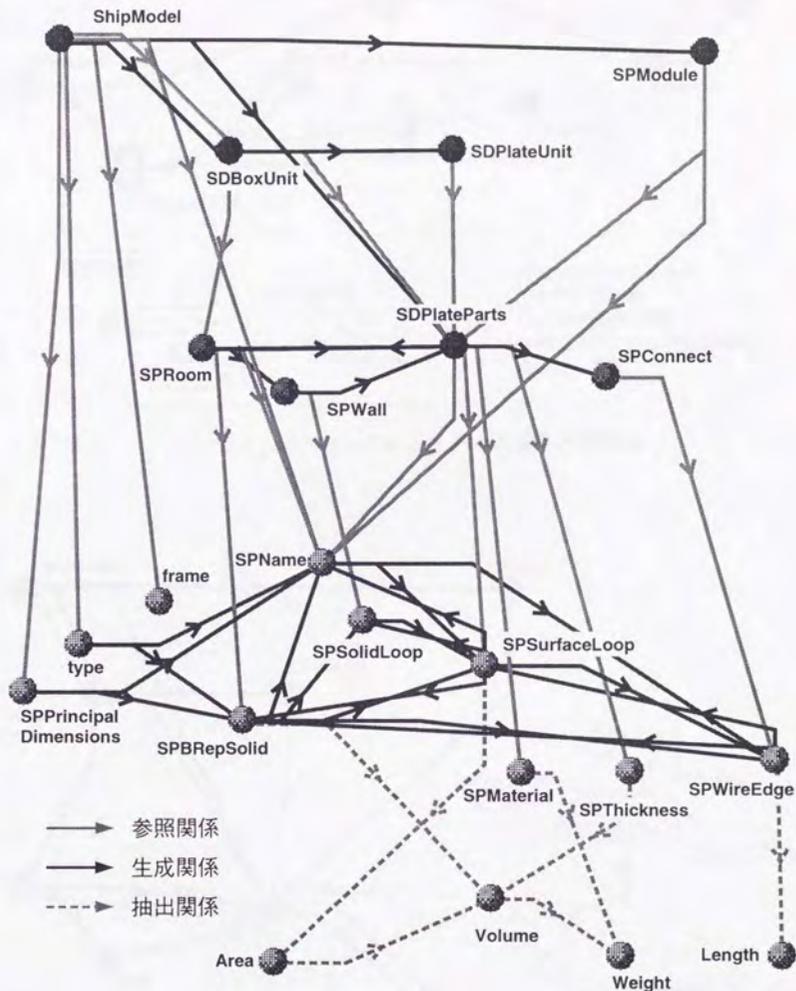


Fig.7-2-7 SODASの製品モデルの表現例(2)

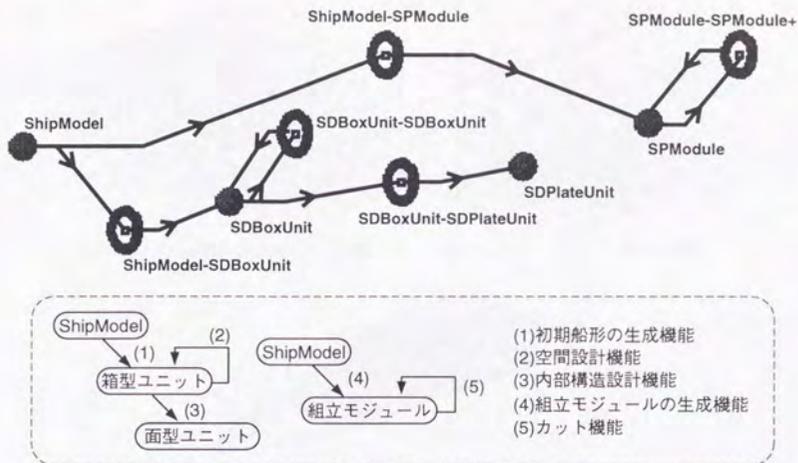


Fig.7-2-8 SODASにおけるモデルの成長の表現例(1)

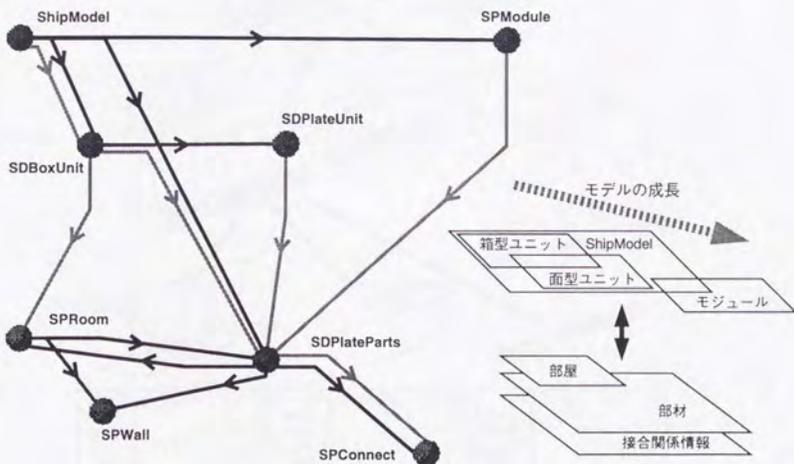


Fig.7-2-9 SODASにおけるモデルの成長の表現例(2)

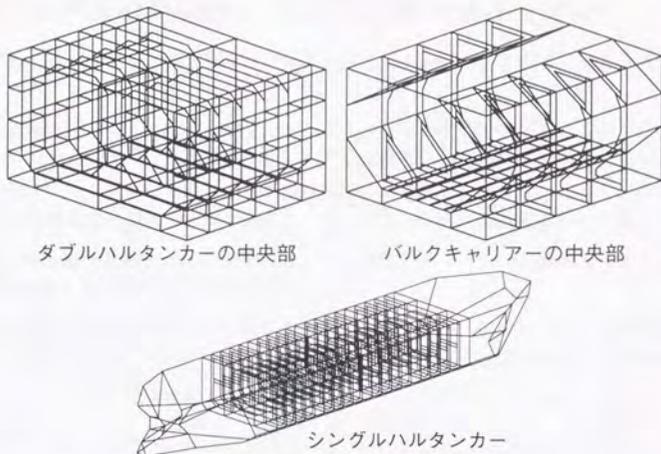


Fig.7-2-10 船の設計例

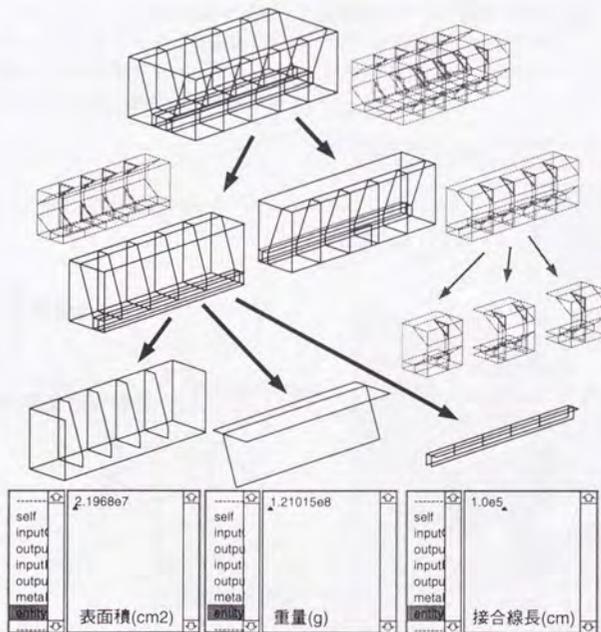


Fig.7-2-11 船のブロック分割例

7.3 本システムの利用例(2) - 海洋構造物の製品モデル -

本節では、海洋構造物の製品モデル(Fig.7-3-1) (Appendix-Bにて詳述) を本システム上に表現することを試みる。

海洋構造物の製品モデルは船の製品モデルに上流設計の部分を追加したものとして捉えられる。また、7.2節で行ったSODASの製品モデルの表現では、製品モデルを構築する際に重要な複数のモデルの統合の問題について検証することができなかった。そこで、以下の流れによって海洋構造物の製品モデルを構築し、特にモデルの統合の問題について検証する。

- (1)海洋構造物の製品モデルの上流部分、即ち機能要素モデル、主要構造モデル、線構造設計機能
- (2)海洋構造物の製品モデルの下流部分である部屋モデル、部材モデル、空間設計機能、内部構造設計機能については、7.2節のSODASのノードやリンクをそのまま利用する。
- (3)上記(1)と(2)とを統合する。

7.3.1 上流部分の構築

本システムを用いて機能要素モデル、主要構造モデル、線構造設計機能を表現した例をFig.7-3-2に示す。また、海洋構造物の製品モデルの上流部分全体を表現した例をFig.7-3-3に示す。以上の実装の際にSmalltalk上に定義したクラスは、1本の線の情報を基にWFモデルの情報を生成する機能(属性層)のみである。

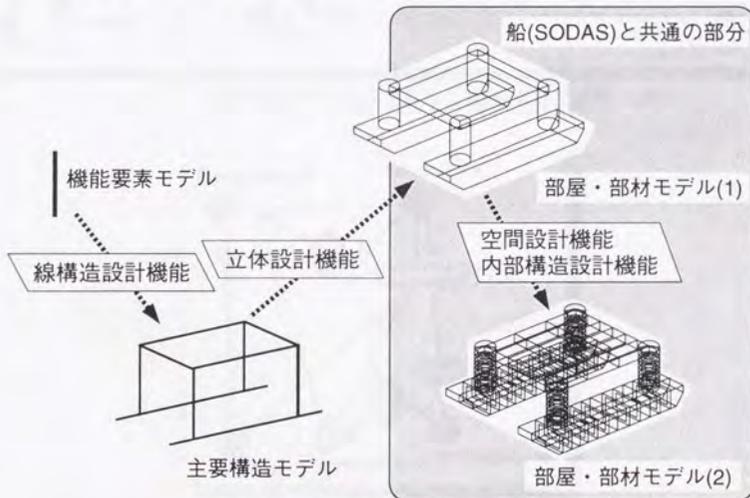


Fig.7-3-1 海洋構造物の製品モデルの概要

7.3.2 上流のモデルと下流のモデルとの統合

海洋構造物の製品モデルの下流部分はSODASのノードやリンクがそのまま利用できる。そこで、7.3.1項のノードやリンクとSODASのノードやリンクとを統合化する。

5.3.3項において述べたように、本研究ではノードやリンクの集合を1つの纏まりとして捉える概念としてモジュールの概念を導入している。そこで海洋構造物の製品モデルの上流部分とSODASの製品モデルとを各々モジュールとして登録し、そして登録された2つのモジュールを統合する。複数のモジュールを統合する際には以下の項目が求められる (Fig.7-3-4)。

- (1)モジュールの構成要素の重複を確認し、関係の付け替えを行うこと
- (2)モジュール間の情報の参照や変換に対応すること

以下に上記(1)及び(2)について述べる。

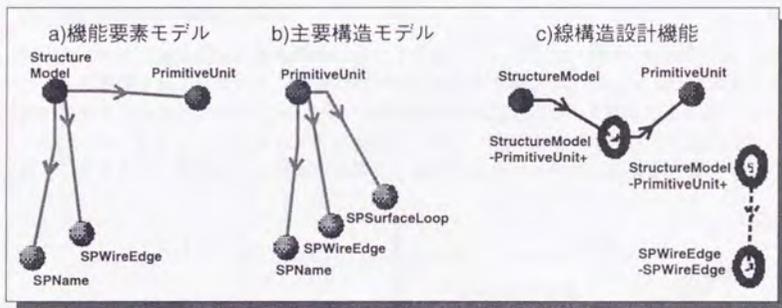


Fig.7-3-2 海洋構造物の製品モデルの製品情報と機能の表現

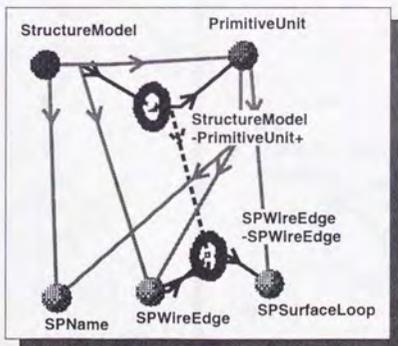


Fig.7-3-3 海洋構造物の製品モデルの上流部分の表現

(1)構成要素の重複の確認

海洋構造物の製品モデルの上流部分の構成要素とSODASの製品モデルの構成要素とを比較すると、名前のノード、線分のノード、面分のノードが重複していることが分かる。このためFig.7-3-4(1)に示すように重複しているノードの有するリンクを付け替えなくてはならない。この統合においては、SODASを主、海洋構造物を従と考え、海洋構造物の名前のノードなどが有するリンクをSODASの名前のノードなどへ付け替えている。以上の部分はシステム開発者がSODASを主、海洋構造物を従と指定すればシステムによって自動的に行われる。

(2)情報の参照や変換への対応

情報の参照や変換を考慮すると、この統合においては以下の2項目が求められる。

- 1)主要構造モデルから部屋や部材の情報を参照できること
 - 2)主要構造モデルの情報を利用して部屋や部材の情報を生成できること
- (立体設計機能を定義すること)

本研究においては上記1)は参照関係によって実現され、上記2)は機能、制御関係、生成関係によって実現される。従って、複数のモデルを統合する際には、Fig.7-3-4(2)に示すように関係や機能を定義すれば良い。そこで、海洋構造物の上流部分と下流部分との統合の際には、Fig.7-3-5に示すノードとリンクを定義した。また、この統合の際に新たにSmalltalk上に定義したクラスは、線分あるいは線分の集合を基に立体形状を獲得する機能（属性層）のみ

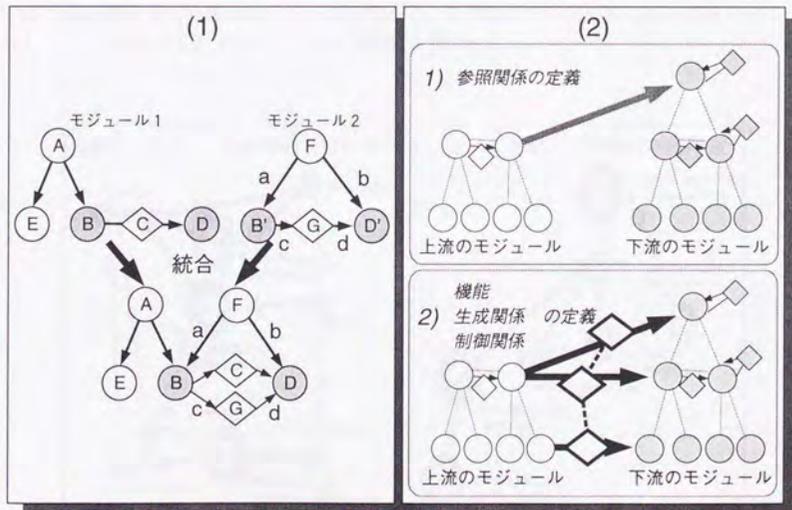


Fig.7-3-4 上流のモデルと下流のモデルとの統合

である。以上のように本システムを用いると、複数のモデルの統合の問題に対しても柔軟に対応することが可能である。

7.3.3 海洋構造物の製品モデルの全体の表現

7.3.1項及び7.3.2項のようにして構築された海洋構造物の製品モデルのモデラの全貌をFig.7-3-6に示す。Fig.7-3-6はFig.7-2-7と同様に制御関係を省略し、機能と機能の入出力の生成関係とを1つのリンクによって表現している。また、海洋構造物で新たに定義されたノードやリンクは強調されている。Fig.7-3-6より分かるように船で定義したノードやリンクが有効に利用されている。また、SODASのノードやリンクと、海洋構造物の製品モデルに必要な機能のクラスが定義されている状態で、本システムを用いて海洋構造物の製品モデルを構築するのに要する時間は1時間弱である。

実体層の中の上位の層の部分に着目すると、Fig.7-3-7に示すようにモデルの成長過程が明確に表現されている。以上のように本研究の表現方法は、船の製品モデルのみでなく、海洋構造物の製品モデルにおいても有効である。また、構築された製品モデルを用いた海洋構造物の設計例をFig.7-3-8に示す。

ところで本研究の製品モデルは、情報や機能を頂点、関係を枝とする有向の連結グラフによって表現される。このためグラフ理論の概念を利用して製品モデルの大きさを把握することが可能である[小野寺68]。例えば情報と参照関係によるグラフを考えると先のSODASの半径は9、海洋構造物の半径は11ということになる。これによって海洋構造物の製品モデルが船より大きな製品モデルであることが定量的に表現される。

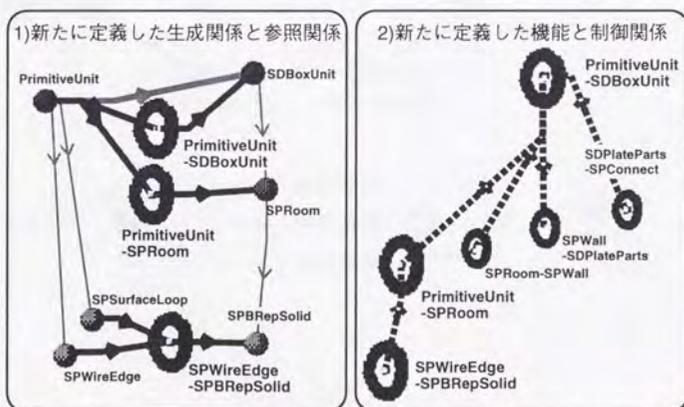


Fig.7-3-5 モデルの統合のために定義したノードとリンク

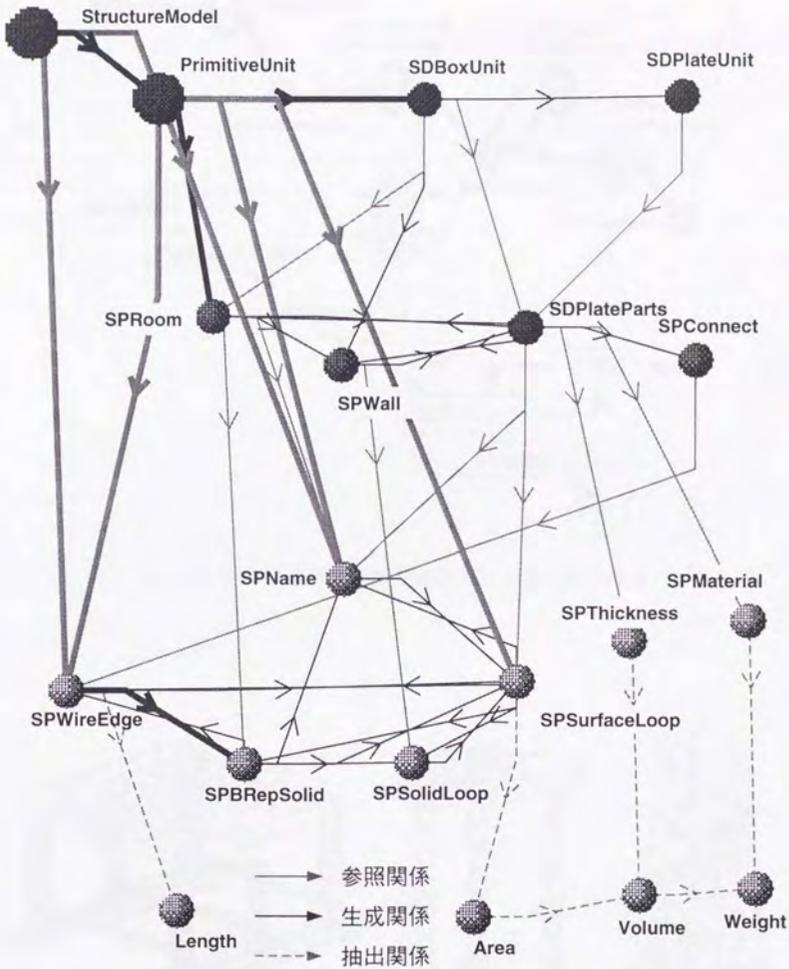


Fig.7-3-6 海洋構造物の製品モデルの表現例

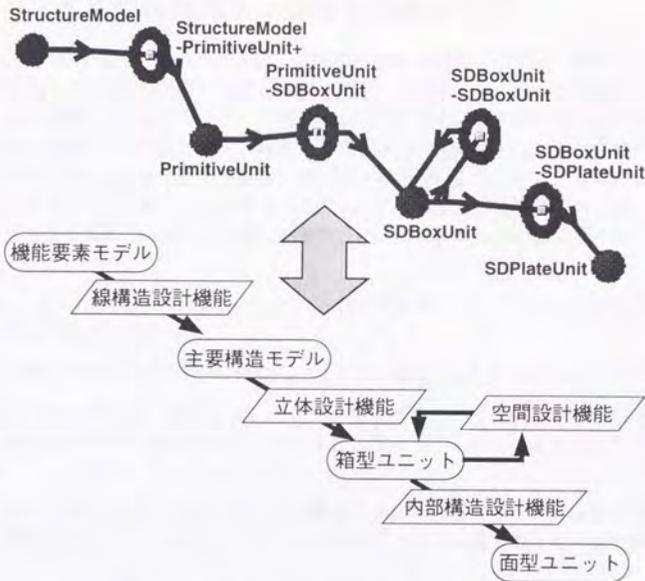


Fig.7-3-7 海洋構造物の製品モデルの成長の表現

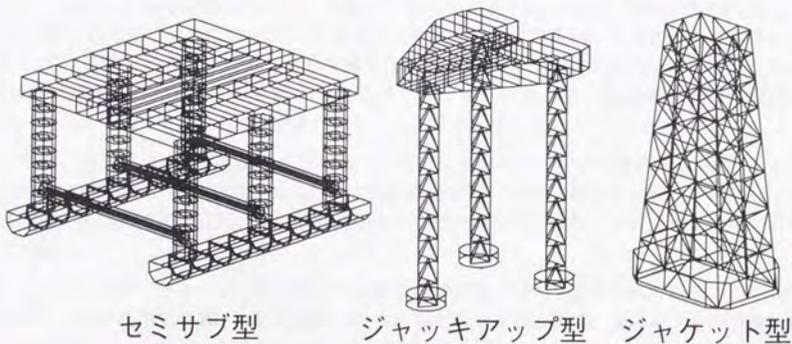


Fig.7-3-8 海洋構造物の設計例

7.4 本システムの利用例(3) - 長瀬らの製品モデル -

7.2節及び7.3節で扱った製品モデルは、著者の所属する研究室で研究・開発されたものである。本研究はこれらの製品モデルの検討に立脚しているため、これらの製品モデルを表現できることは当然といえる。しかし、本研究の製品モデルの表現方法の良否を検証するためには、著者の所属する研究室で研究・開発された製品モデルのみでなく、他の機関で開発された製品モデルも表現できることを確認しなければならない。そこで、2.3.2項において述べた長瀬らが提案している製品モデルを本システム上に実装することを試みた。尚、実装に当たって以下の点を考慮し、実際に長瀬らが提案している製品モデルの簡易版を本システム上に実装する。

- ・製品モデルの構成としてはFig.7-4-1に示す図を基本と考え、これらの情報が正確に管理されることを考慮する。
- ・情報の操作を行うモデル操作言語(Fig.7-4-2)は、機能として本システム上に実装する。
- ・形状の具体的な表現方法などは、本研究で特に対象としてはいない。このため、長瀬らの製品モデルでは曲面や曲線を扱えるが、本システムでは平面と直線のみしか扱わない。
- ・長瀬らの製品モデルには製品モデルの情報を生成・利用するための有効な外部システムが多々存在する。これらのシステムは製品モデルの外部と考えて実装の対象としない。

以上の考えに基づいて実際に長瀬らの製品モデルのモデルを本システム上に表現した例をFig.7-4-3及びFig.7-4-4に示す。Fig.7-4-3は実体層の全てのノードとリンクを示しており、Fig.7-4-4が明確に表現されていることが理解できる。また、Fig.7-4-4は、機能と機能の入出力の生成関係が1つのリンクによって表現されており、新たに定義したノードやリンクは強調されている。

ところで、SODASの製品モデルと長瀬らの製品モデルをFig.7-4-5に同時に載せる。Fig.7-4-5より分かるように、SODASでは生成関係や機能を表現する黒いリンクが中心であるのに対して、長瀬らの製品モデルでは参照関係を表現する灰色のリンクが中心である。これはSODASが情報の生成に主眼を置いた製品モデルであり、長瀬らの製品モデルが情報の管理・利用に主眼を置いた製品モデルであることが示されている。

また、本システム上に実装された長瀬らの製品モデルを用いた設計例をFig.7-4-6に示す。SODASの製品モデルでは初期船形を基に設計機能を用いて設計が行われる。これに対して長瀬らの製品モデルではFig.7-4-6(1)に示すような部分構造が直接的に設計されることが理解できる。

以上のように、本システムは、著者の所属する研究室で研究・開発された製品モデルのみでなく、他の機関で提案された製品モデルも表現可能である。更に、製品モデルの特徴についても本システムによって表現されることが示された。

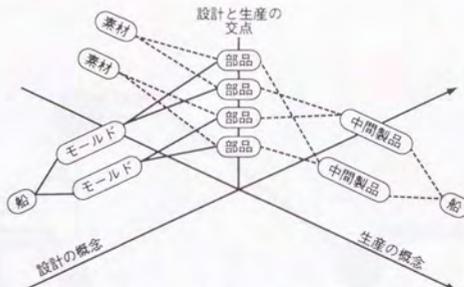


Fig.7-4-1 長瀬らの製品モデルの基本構成 [長瀬92]

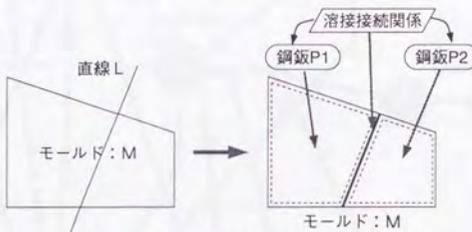


Fig.7-4-2 モデル操作言語 [長瀬92]

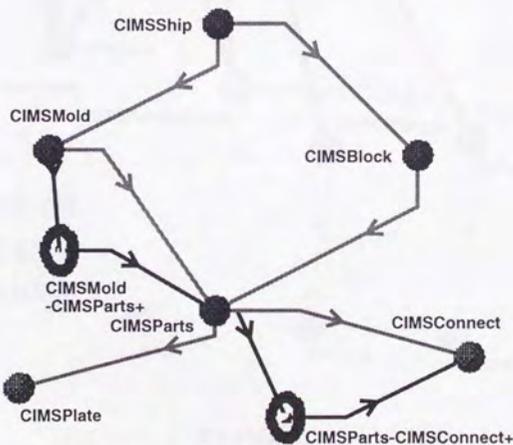


Fig.7-4-3 長瀬らの製品モデルの表現例(1)

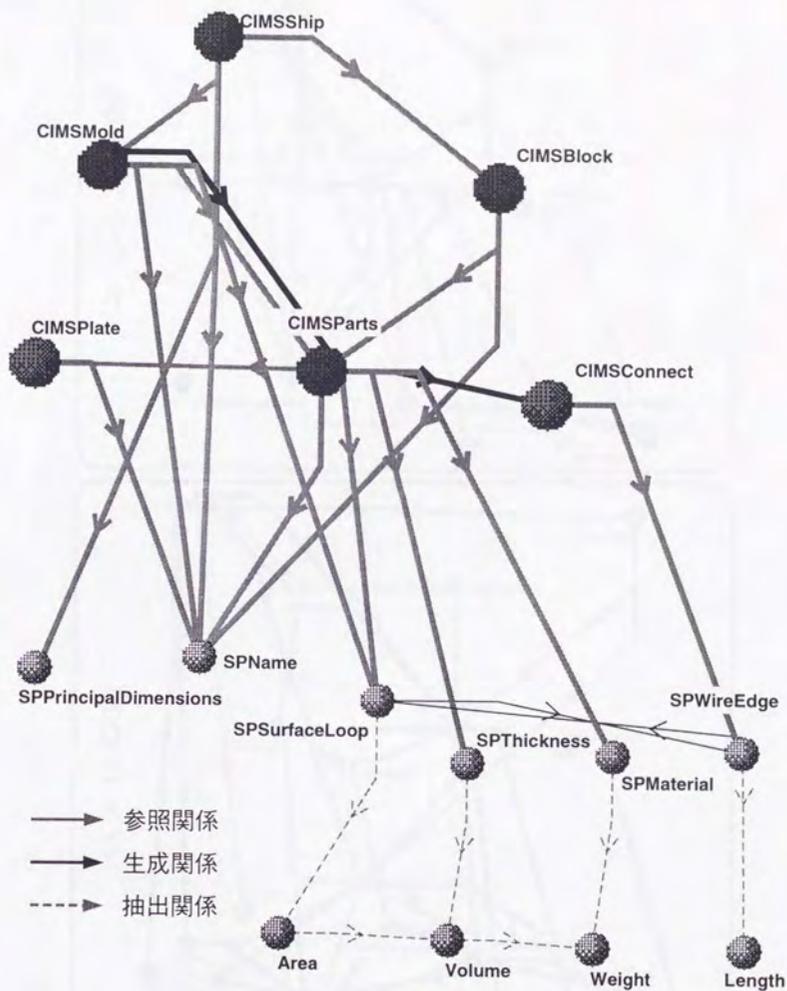


Fig.7-4-4 長瀬らの製品モデルの表現例(2)

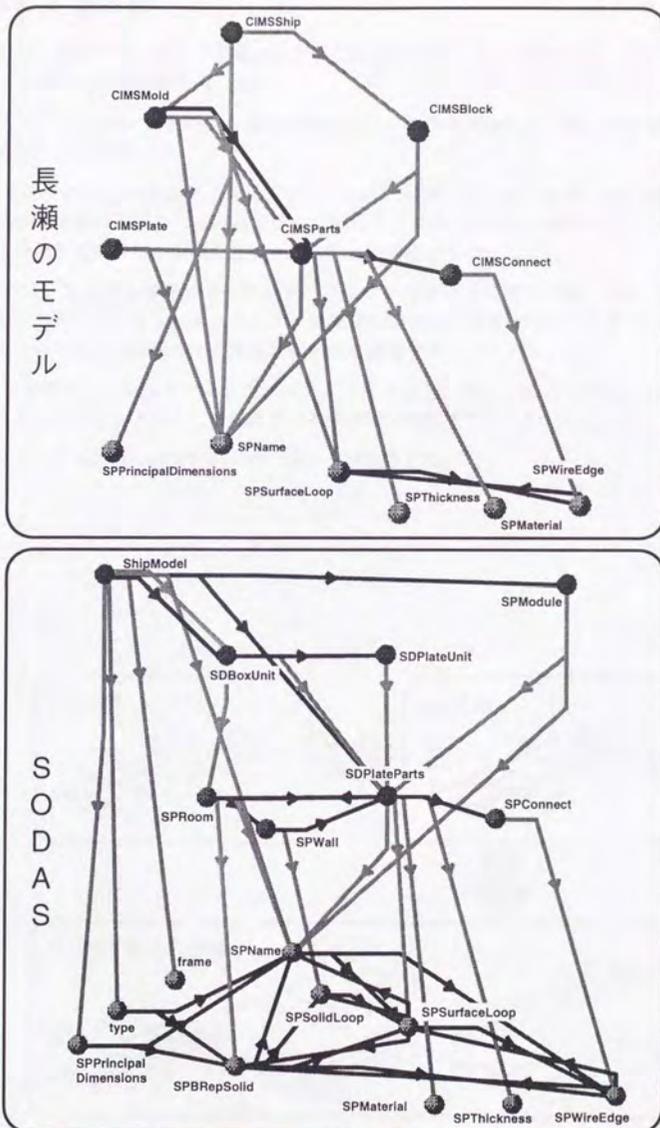


Fig.7-4-5 長瀬らの製品モデルとSODASの製品モデルとの比較

7.5 本章のまとめ

本章では、本研究のシステムを用いると製品モデルがどのように構築され、どのように表現されるかについて実例を含めて述べた。

7.1節では、本システムを用いた場合の製品モデルの構築の流れと、情報や機能の追加・変更の方法について述べた。

7.2節では、SODASの製品モデルを本システム上に実装した。その結果、製品情報・機能などの構造が明確に表現されるのみでなく、製品モデルの成長過程や製品モデルの中での製品情報や機能の位置づけが明確に表現されることが確認できた。

7.3節では、海洋構造物の製品モデルを特にモデルの統合の問題を考慮して本システム上に実装した。そして、本システムがモデルの統合の問題にも柔軟に対応できること、船の製品モデルのみでなく海洋構造物の製品モデルにも適用できることを示した。

7.4節では長瀬らの製品モデルを本システム上に実装し、幅広い製品モデルを本システムによって表現できること、そして製品モデルの特徴を表現できることを示した。

以上によって本研究の有効性が実践の面から検証された。

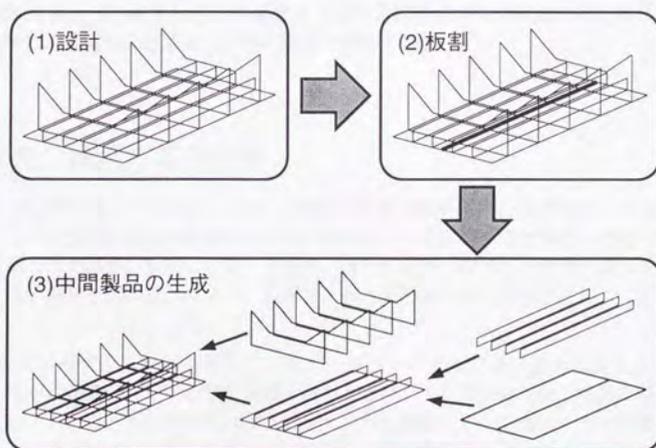


Fig.7-4-6 本システム上の長瀬らの製品モデルを用いた設計例

第8章

造船CIMと本研究

本研究では製品モデルの表現方法について検討した。製品モデルは造船CIMの中核となる重要な部分であるが、造船CIMの全てではない。造船CIMを構築するためには、製品モデルとCAE(Computer Aided Engineering)などのApplication Systemとの統合化や、製品モデルと生産環境モデルとの関連なども含め、更に幅広い検討が必要である。

そこで本章では、先ず8.1節において本研究の特徴を再整理する。その上で、製品モデルとCAEとの統合化や、生産環境モデルを含めた検討の際に本研究がどのように利用できるかについて検討する。これによって、本研究の今後の発展性について検討すると共に、造船CIM全体に対して本研究を適用するための課題を明確にする。

8.1 本研究の表現方法の特徴

本研究では、製品モデルを情報、機能、4種類の関係（参照関係、制御関係、生成関係、抽出関係）という構成要素の組み合わせとして表現している。情報の生成は、機能・制御関係・生成関係（抽出関係）を中心として行われ、情報の管理（参照）は情報・参照関係を用いて行われる。製品モデルをこのように表現することが必要であった理由をここで考えてみる。

5.2節において述べたように製品モデルをデータベースとして単に静的に捉える場合には、製品モデルに記述されている情報を取り出すことができればよい。即ち情報と参照関係のみあればいい。このような時はFig.8-1-1に示すような構造によって製品モデルの構造を表現することができ、C言語の構造体でも、オブジェクト指向言語のオブジェクトによっても十分に表現することが可能である。

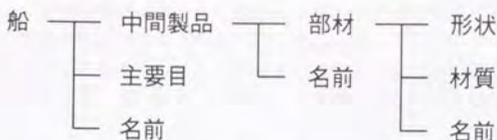


Fig.8-1-1 情報の静的な表現

しかし、製品モデルを動的な視点と静的な視点の両方から、情報の生成・利用・管理を含めて捉える場合、情報と同時に機能に着目しなければならない。その結果、様々な関係が複雑に交錯し、情報間の関係のみでは製品モデルを表現しきれなくなる。このために機能を含めた表現が必要となり、本研究のような表現方法が要求される。

従って、本研究の表現方法は、情報の生成・利用・管理を含めて捉えなければならない問題に対して有効であり、情報・機能・関係の複雑な組み合わせを整理して表現できるという特徴を有する。

8.2 CAEの表現と本研究

8.2.1 造船CIMとCAE

造船業の技術競争力強化のために、これまでの船級協会規則に基づいた設計手法(Design by Rule)から脱却し、計算機による直接計算に基づいた設計手法(Design by Analysis)へ移行することが求められている[大坪87]。Design by Analysisを実現するためには、設計—荷重解析—応力解析—構造強度算定—信頼性評価—総合評価などから構成される設計サイクルを計算機によって強力に支援することが要求される。このために日本造船研究協会において、新船体構造設計法が調査研究されている[藤野92]。

ところで、上記のような次世代の設計支援システムを構築するためには、製品を定義するシステムとCAEとの統合が重要である[相澤88]。

Fig.8-2-1に示すようにシミュレーションが設計の初期で行われれば行われる程その効果は大きいため[Buckley92]、CAEと統合化する製品定義システムには上流の設計の情報が記述されていることが望ましい。また製品モデルには、設計の最終結果のみが記述されるのではなく、設計の上流で生成される情報も記述される。このため、設計の上流で生成される情報を利用して、設計の早期に評価検討などのシミュレーションを行うことが期待できる。

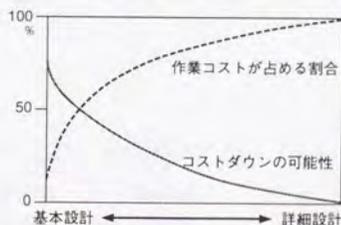


Fig.8-2-1 コストダウンの可能性

更に、製品モデルには形状情報のみでなく製品の物理的な情報なども記述され、このような情報はCAEの入力情報の生成の際に有効に利用できる。従って製品モデルはCAEと統合化する製品定義システムとして適切であり、この統合化によって21世紀の設計手法として着目されているシミュレーション・ベースド・デザインを実現するための基盤が確立されると考えられる[綾96]。

以上のように、造船CIMにおいて設計を強力に支援するための手法として、造船CIMの中核となる製品モデルと、船体運動解析システムやFEMなどのCAEとの統合は非常に重要と考えられる。

8.2.2 CAEを含めた表現のために

(1) CAEの表現

CAEは入力情報を基に出力情報（解析結果）を生成するものであるため、本研究の種類分けでは機能として考えられる。そこで本研究の表現方法を用いてCAE、入力情報、出力情報を表現するとFig.8-2-2のように表現される。図中のリンクは情報と機能との関係であるので変換関係として捉えられる。

(2) 製品モデルとCAEとの統合

製品モデルとCAEとを統合するためには、製品モデルの情報を有効に利用してCAEの入力情報を生成することが重要と考えられる。製品モデルの有する情報をCAEの入力情報として直接的に利用できる時は、Fig.8-2-3(1)に示すように入力情報と製品モデルの内部の情報との



Fig.8-2-2 CAEと入出力情報の表現

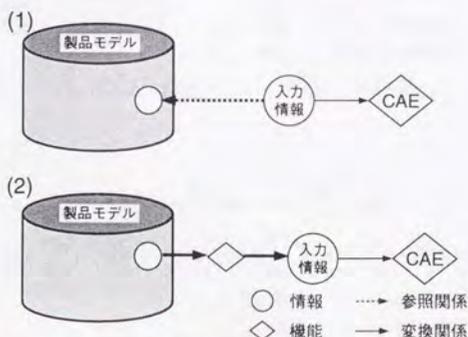


Fig.8-2-3 製品モデルとCAEとの統合化(1)

間に参照関係を定義すればよい。

また、FEMの解析対象が有限要素の集合として表現されるように、様々なCAEの解析対象は製品モデル内部の情報とは異なったモデルによって表現される場合が多い。このような時は、製品モデルの情報を変換し、CAEの解析対象を生成する（製品モデル内の情報を機能によって変換し、CAEの入力情報を生成する）ことが必要になる。このような入力情報の生成を、本研究の方法によって表現するとFig.8-2-3(2)のように表現される。

以上のようにして、製品モデルとCAEとの統合を複数回行うとFig.8-2-4のように表現される。

(3)CAE同士の関連

船体運動解析の出力情報として得られた船体各部にかかる荷重の情報は、構造解析を行う際に有効に利用できる。このようにCAEの間には関連があり、あるCAEの出力情報が他のCAEの入力情報として有効に利用できることは少なくない。あるCAEの出力情報を他のCAEの入力情報として直接的に利用できる時は、本研究の方法を用いると、Fig.8-2-5(1)に示すように参照関係を定義することによって情報の利用が可能となる。また、情報の加工が必要な時は、機能を介してCAE1の出力情報とCAE2の入力情報とを関係付ければ良い(Fig.8-2-5(2))。

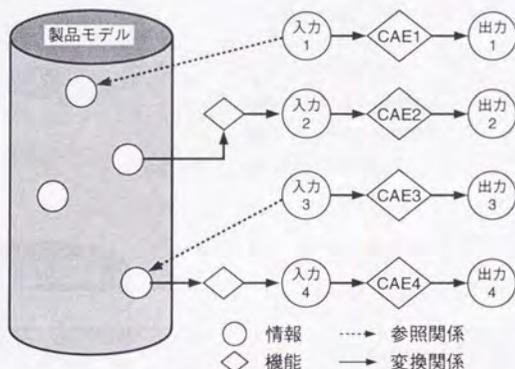


Fig.8-2-4 製品モデルとCAEとの統合化(2)

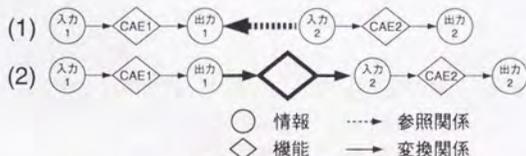


Fig.8-2-5 CAE同士の関連(1)

このようにして、Fig.8-2-4のCAE同士の関連を考慮すると、例えばFig.8-2-6のように表現される。このように表現されると、様々なCAEの間の関連が一目で分かることが理解できる。

(4)製品モデルの成長とCAE

本研究の製品モデルは情報と機能の構成として表現され、変換関係（生成関係）に着目することによって製品モデルの成長が表現される。そこでFig.8-2-6を製品モデル内部の変換関係（生成関係）を含めて表現するとFig.8-2-7のように表現される。このように表現される

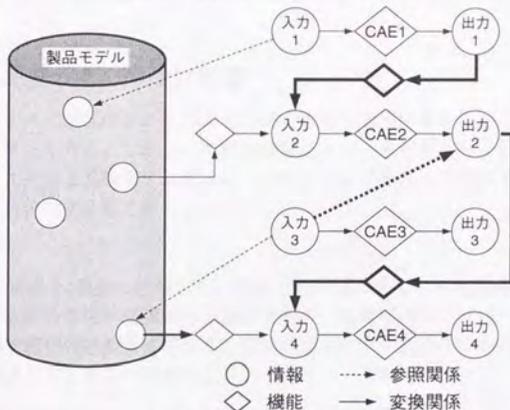


Fig.8-2-6 CAE同士の関連(2)

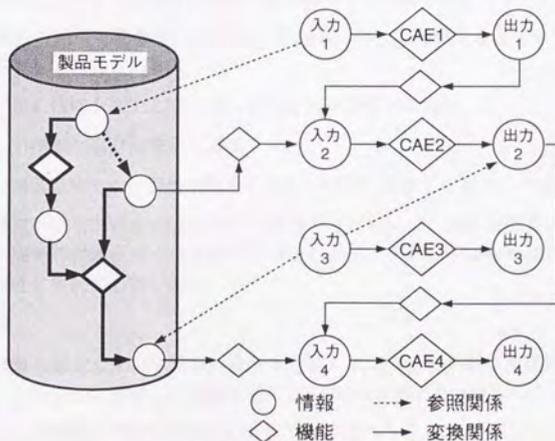


Fig.8-2-7 製品モデルの成長とCAE

と、モデルの成長とCAEとの関連についても把握することができ、本研究の表現方法が有効に利用できることが期待される。

以上に示したように、本研究の表現方法は製品モデルとCAEとの統合の際にも有効に利用できると考えられる。これは、製品モデルとCAEとを統合するためには、情報の生成・利用を含めて情報・機能・関係の複雑な組み合わせを検討する必要がある、このような特徴を有する問題に対して本研究の表現方法が有効なためと考えられる。

8.2.3 CAEの表現のための検討課題

前項に示したように製品モデルに加えてCAEを含めて表現する際にも、本研究の手法が基本的には有効と考えられる。しかし、本研究は製品モデルを対象として行っているため、本研究の表現方法をそのまま適用する訳にはいかない。CAEを含めて表現するためには、以下の項目について再検討が必要である。

(1)関係の種類

本研究では、関係を4種類に分類した。これらの関係の中で、参照関係は製品モデルの内部の情報同士を参照の目的で関係付ける関係であり、変換関係は「情報-機能-情報」を情報の変換の目的で関係付ける関係である。変換関係の中で、製品モデルの外部の情報を有するものは抽出関係として考えられている。

ところで、製品モデルとCAEとの統合化のためには、前項に示したように製品モデルの内部の情報とCAEの入力情報(製品モデルの外部の情報)とを参照関係によって関係付けることが要求される。このような問題は現状の種類分けでは考慮されていない。

また、本研究の関係の種類分けをそのままCAEに対して適用すると、以下の2つの関係は共に抽出関係として定義される。

- ・製品モデルの情報を加工してCAEの入力情報を獲得する関係
- ・CAEの入力情報と出力情報との関係

しかしこの両者は明らかに目的が異なるため、明確に区別して捉えるべきと考えられる。

従って本研究の表現方法をCAEに対して適用するためには、関係の種類別の再検討が必要である。また、関係の種類分けの再検討のみで済むのか、それとも情報や機能の種類分けも行うべきかも同時に検討が必要である。

(2)層の概念

本研究では層の概念を用いて情報や機能を分類しており、最小単位を属性層の情報や機能としている。このように分類した理由は主に以下の2つの項目を考慮したためである。

- ・形状などの属性層の情報や機能については造船のみならず多方面で研究されているため、属性層を最小単位にすると、これらの研究の成果を有効に利用できる。

- ・製品モデルは実体層の情報を中心となっているため、情報や機能の分類は実体層を中心に行うべきである。

従って、CAEに対して層の概念を用いるためには、対象とする情報や機能が上記の項目を満足していることを確認し、満足していない時は別の考え方を導入することが必要である。言い換えれば、CAEを構成する部品をどのような単位で構築すべきかについて検討が必要である。

8.3 生産環境モデルの表現と本研究

8.3.1 造船CIMと生産環境モデル

生産活動は、生産環境に素材や中間製品が流されることによって、生産環境の有する生産方法や生産技術を用いて素材や中間製品の形態が変換され、設計活動で設計された製品が製造される過程である[岩田他82]。現実の生産はその造船所独自の生産環境を用いて行われるため、生産環境の能力は生産活動に対して大きな影響を与える。従って、生産計画は、製品に関する情報のみでなく生産環境を考慮して立案される必要がある。

造船CIMにおいて生産や生産計画を支援するシステムを構築することは非常に重要な課題である。このようなシステムを構築するためには、製品に関する情報を統合的に管理する製品モデルのみでなく、生産環境を表現する生産環境モデルが必要である。従って造船CIMを構築するためには、製品モデルの表現に関する検討のみでなく、生産環境モデルの表現に関して検討し、更に製品モデルと生産環境モデルの関連について検討することも重要である。

8.3.2 生産環境を含めた表現のために

(1)生産環境モデルの表現

2.4.4項において述べた生産環境モデルを本研究の手法によって表現することを試みる。

生産環境モデルとして「工場モデル」「工程資源モデル」「作業資源モデル」が挙げられる。工場モデルは工程資源モデルを管理する。工程資源モデルは作業資源モデルを管理し、自身の能力と名前を有する。また、作業資源モデルは、自身の能力と名前を有する。ここで能力と名前は属性情報と捉えられるので、これらの生産環境モデルを本研究の手法によって表現するとFig.8-3-1のように表現される。

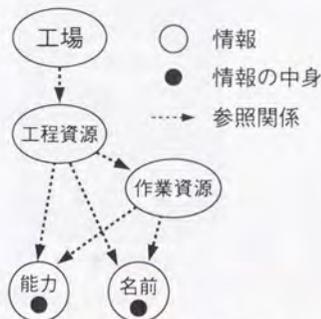


Fig.8-3-1 生産環境の表現

(2)作業（製品モデルと生産環境モデルとの関連）の表現

2.4.4項で述べたように、製品モデルと生産環境モデルとの関連として「作業」が定義される。作業は、自分の行われる時刻の情報、自分の作業対象である製品モデルの情報、自分の作業環境である生産環境の情報とを有する。作業には工程作業と実作業があり、工程作業は中間製品と工程資源を有し、実作業は部材あるいは接合関係情報と作業資源の情報とを有する。また、工程作業は実作業の情報を管理する。そこで本研究の手法によって作業を表現するとFig.8-3-2のようになる。また、生産環境モデルや製品モデルを含めて作業を表現するとFig.8-3-3のようになる。

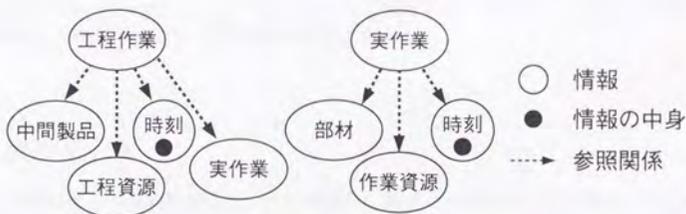
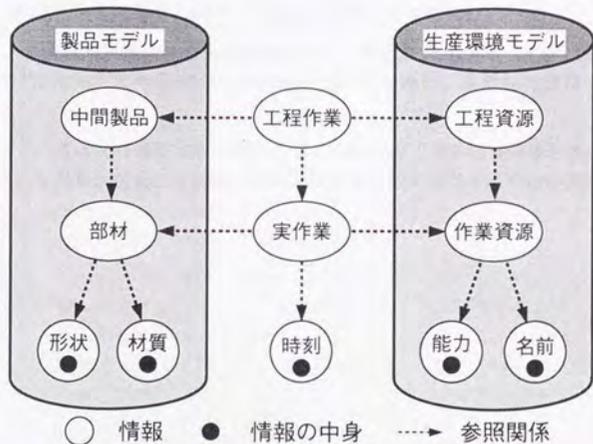


Fig.8-3-2 作業の表現(1)



注：参照関係は一部省略

Fig.8-3-3 作業の表現(2)

8.3.3 生産環境を含めた表現のための検討課題

現状では、生産環境モデルや作業モデルは、製品モデルと比較して不明確な部分が多く、特に機能は未整理であると考えられる。また、本研究は情報や機能が明確化されたなら、その情報や機能を如何に表現すべきかを検討しており、情報や機能の明確化を自動的に行うことを目的としてはいない。従って本研究の手法によって生産環境を表現するためには、先ず生産活動を支援するために必要な情報や機能を明確にすることが必要である。

但し、情報や機能を明確にしていくプロトタイピングの過程に本システムを用いることによって、情報や機能の間の関係を明確に意識することができる。これによって情報や機能の明確化を間接的に支援することが期待される。但し、本システムが情報や機能の明確化にどのような効果があるかについては、本システムを実際にプロトタイピングに使用することによって検討すべき項目であり、今後の検討課題と考える。

8.4 本章のまとめ

本章では本研究の今後の発展性について検討し、製品モデルのみでなく造船CIM全体に本研究の表現方法を適用することが可能か否かを述べた。

8.1節では本研究の表現方法の特徴を再整理し、本研究の表現方法が情報の生成・利用・管理が複雑に絡み合う問題に対して有効であることを示した。

8.2節では、本研究の表現方法をCAEに対して適用することを検討した。そしてCAEに対して本研究の表現方法を適用する時の、有効性と課題とを明確にした。

8.3節では、本研究の表現方法を生産環境モデルに対して適用することを検討した。そして生産環境モデルに対して本研究の表現方法を適用する時の、有効性と課題とを明確にした。

以上によって、本研究の表現方法は製品モデルのみでなく造船CIM全体を表現する時にも有効であることを示し、造船CIM全体に本研究の表現方法を適用するための課題を明確にした。

第9章

結言

9.1 本研究の結論

本研究では造船界が製造対象とする船や海洋構造物などの板骨構造物を対象として、製品モデルの表現方法について検討した。本節では本研究で得られた知見を整理し、本研究の結論を総括する。

9.1.1 本研究の位置づけ

造船界におけるこれまでの製品モデルに関する研究は、製品モデルを構成する情報や機能の明確化を中心に行われてきた。しかし統合化された設計・生産システムを構築するためには、要素の明確化のみでなくシステム全体の構成についての検討が重要である。従って製品モデルを明確にするためには、製品モデルを構成する情報や機能などの個々の要素の明確化のみでなく、製品モデル全体の構成についての検討が必要である。

このため、本研究では製品モデルの構成を含めて製品モデル全体としての表現方法について検討している。言い換えれば、これまでの研究は造船という現実世界をモデル化し、更にモデルを基にシステムを構築してきたと考えられる。これに対して本研究はモデルあるいはシステムとしての特徴を踏まえて製品モデルを捉え直し、製品モデルの在り方を検討した研究と考えられる(Fig.9-1-1)。

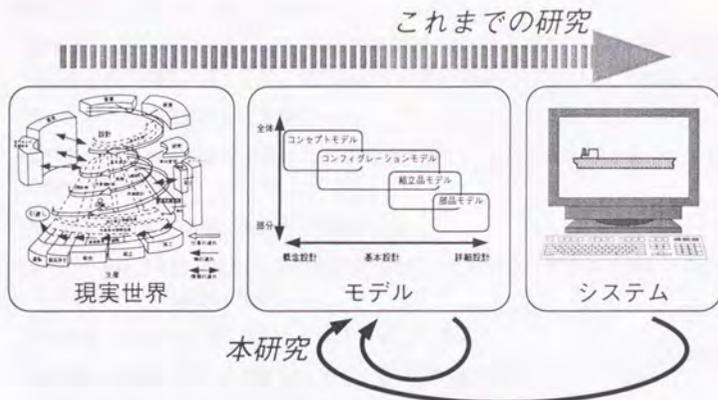


Fig.9-1-1 本研究の位置づけ

9.1.2 製品モデルの表現方法の検討

本研究では、先ず以下の3項目について検討し、製品モデルの表現方法を理論面から明確にした。

(1) 製品モデルの表現方法に要求される項目の整理

製品モデルの表現方法について検討するためには、製品モデルを様々な視点から捉え、製品モデルの特徴を考慮することが重要と考えた。このため、設計・生産活動の支援の視点、情報処理の視点などから製品モデルの特徴を検討した。そして検討した製品モデルの特徴を基に、製品モデルの表現方法に要求される項目を以下の4つに整理した。

- a) 製品モデルを情報や機能の構成として表現できること
- b) 製品モデルの多面性を表現できること
- c) 情報や機能が部品化されていること
- d) 製品モデルの柔軟性・拡張性が確保されていること

(2) 製品モデルの表現方法の検討

製品モデルを構成する情報や機能の間には「関係」が存在し、「関係」は情報や機能の結合において重要な意味合いを持つ。そこで本研究では製品モデルを構成する情報や機能を部品化し、部品化された情報や機能を利用して組み合わせることによって、製品モデルを表現することが有効と考えた。このために製品モデルを構成する様々な情報や機能を関係を考慮して部品化した。そして部品化された情報や機能をその特徴に応じて分類し、情報や

機能を分類するための概念として層の概念を導入した。その結果、製品モデルを構成する情報や機能は以下に示す2つの層に分類される。

- ・実体層：現実の世界における物や物の集合物に対応する情報、及びそれらの情報を生成するための機能が属する層。他の情報や機能との関係付けによって定義される。

(例) 部材、中間製品、カット機能etc.

- ・属性層：実体の一側面を表現するための属性情報、及びそれらの情報を生成するための機能が属する層。

(例) 形状、材質、名前、面分の分割機能etc.

更に、上記の部品を組み合わせる役割を担う関係をその特徴に応じて分類し、製品モデルにおける関係を以下の4種類に整理した。

- ・参照関係：情報の参照を目的とする情報同士の関係
- ・制御関係：情報の変換の実現を目的とする機能同士の関係
- ・生成関係：製品モデルの情報の生成を目的とする情報-機能-情報の関係
- ・抽出関係：製品モデルの情報の利用を目的とする情報-機能-情報の関係

そして以上の整理を基に製品モデルを表現すると、情報や機能が関係によって結合された結合体として製品モデル全体が表現されることを示した。

(3)表現方法の検証

上記(2)の表現方法を用いた製品モデルの特徴は以下のように整理することができる。

- a)情報と機能を同格と見做し、各々を明確に認識することができる。
- b)情報及び機能が分解され部品化されている。
- c)部品化された情報及び機能を関係によって組み合わせ、情報や機能の結合体として製品モデルが表現されている。
- d)関係をその各々が果たすべき役割を考慮し、参照関係・制御関係・生成関係・抽出関係の4種類の関係に整理している。

この特徴によって、本項(1)の要求項目の内、「a)製品モデルを情報や機能の構成として表現できること」、「c)情報や機能が部品化されていること」は満たされる。また4種類の関係を使い分けることによって「b)製品モデルの多面性を表現できること」を示した。更に、情報・機能・関係の組み合わせを自由に行える構造について検討し「d)製品モデルの柔軟性・拡張性が確保されていること」を示した。

以上によって本研究の方法が、製品モデルの表現方法に要求される項目を満足することを示した。更に本研究の表現方法の特徴である「関係」には、関係同士の相性があることを示し、その相性を利用することによってシステム開発者を支援できることを示した。

9.1.3 プロトタイプ・システムの構築

(1) プロトタイプ・システムの構築

本研究の表現方法に基づいた製品モデルをシステムとして構築するために何が必要かを整理した。その結果、オブジェクト指向言語などを本研究の製品モデルの開発言語として直接的に用いることができず、製品モデルを定義するための環境と情報や機能を関係によって組み合わせるための仕組みから考える必要があることが分かった。そこで、本研究のプロトタイプ・システムを製品モデルを構築するためのシステムと位置づけ、プロトタイプ・システムの構成について検討を加えた。その結果、以下のシステム化が必要であることを示し、Smalltalkを利用してプロトタイプ・システムを構築した。

- a) 情報・機能・関係の型を定義し、組み合わせの仕組みを提供すること
- b) 情報・機能・関係を定義し、組み合わせるための環境を構築すること

(2) プロトタイプ・システムの利用

SODASの製品モデル、海洋構造物の製品モデル、長瀬らの提案する製品モデルを上記(1)で構築したプロトタイプ・システム上に実装した。SODASの実装によって、柔軟性・拡張性が高く、様々な側面からその構成を捉えることが可能な製品モデルが定義された。海洋構造物の製品モデルの実装によって、複数のモデルの統合の問題に有効であること、情報や機能を有効に再利用できることが確認できた。また長瀬らの提案する製品モデルの実装によって、著者の所属する研究室の製品モデル以外も本研究の手法によって表現できること、製品モデルの特徴の相違を表現できることが確認できた。

以上によって本研究の有効性を実践の面から確認することができた。

9.1.4 既存システムと本システムとの相違

本項では、情報、機能、関係という本研究のキーワードを用いて、既存システム(CIMS、SODAS)と本研究のシステムとの相違について述べる。

(1) 情報について

既存システムはオブジェクト指向言語であるC++やSmalltalkを用いて構築されている（オブジェクト指向についてはAppendix-Aにて詳述）。このため、部材・中間製品などの製品情報は、部材クラスあるいは中間製品クラスとしてシステム上に実装されている。そして部材や中間製品などを表現するために必要な形状や名前などの情報は、部材オブジェクトあるいは中間製品オブジェクトのインスタンス変数として管理される。

このように製品情報が定義されると、製品情報の構造を変更するためには、

- a) インスタンス変数を変更する
- b) 変更したインスタンス変数に関連するメソッドを変更する

という2つの処理をオブジェクト指向言語の環境で行う必要がある。

本システムには、部材や中間製品などのクラスは存在しない。これまでのクラスに対応するものは、メタノードのインスタンスとして定義される。また、これまでにはインスタンス変数として表現されていた情報間の'has-a'の関係はリンク（参照関係）によって表現される。従って、リンクの定義がインスタンス変数の定義に対応し、また情報構造の変更はリンクの再定義（削除や追加）という1つの処理によって行われる。

(2)機能について

既存システムのように部材の製品情報を部材クラスとして定義すると、部材クラスの有すべきメソッドは以下の3種類に整理できる。

- a)データを格納するためのメソッド
- b)データを取り出すためのメソッド
- c)データを加工（変換）するためのメソッド

上記3種類のメソッドは全て情報処理機能の一種と考えられるが、製品モデルの視点からは情報の生成に関与する情報処理機能と、情報の管理・利用のための情報処理機能とを明確に区別することが重要である。そこで、本研究では「c)データを加工（変換）する情報処理機能」を単に「機能」と呼んでいる。言い換えれば、本研究における「機能」は既存システムにおけるメソッドの一部に対応している。

ところで関係を情報と同様に重要視し、情報と関係を別々のオブジェクトとして定義することを考えると、情報のオブジェクトや機能のオブジェクトが有すべきメソッドについて検討しなければならない。言い換えれば、上記a)~c)のメソッドを情報と一体化すべきか、それとも関係と一体化すべきかについて検討が必要である。「a)データの格納」は情報と深く係わりがあり、情報と一体化すべきメソッドと考えられる。一方「b)データの取り出し」は複数の情報の間の情報構造を利用して行われることが多く、「c)データの加工（変換）」も入力情報や出力情報が関係する。従って上記b)及びc)については単一の情報のみでなく複数の情報と係わりがあるため、情報ではなく関係と一体化すべきメソッドであると考えられる。このため本研究では、上記b)及びc)のメソッドは関係と一体化している。

また、オブジェクト指向における最近の動向として、複雑な情報を生成する情報処理機能は、メソッドとして定義するよりも別のクラスとして定義する方がシステムの柔軟性や再利用性が向上することが指摘されている[Gamma95]。このため本研究においても情報を生成する「機能」はクラスとして定義されている。

以上のように、既存システムでは上記a)~c)は全て情報と一体化されていたが、本システムでは上記b)及びc)は関係と一体化されている。更に上記c)についてはメソッドではなくクラスとして定義されている。これが既存システムと本システムとの機能に関連する相違点である。

(3)関係について

製品モデルのように様々な情報が複雑に絡み合う場合には、情報と同時に関係が重要とされている。従って製品モデルには情報や機能と同時に関係を記述することが重要であり、CIMSでは情報間の関係を有効に利用して、システムの柔軟性を確保することに成功している。

本研究では関係を参照関係、変換関係（生成関係・抽出関係）、制御関係の3種類に分類しているが、これらの関係の内、参照関係についてはCIMSで既に指摘されている。従って本研究で新たに定義された関係は変換関係と制御関係である。

以上に述べたように、これまでのシステムと本システムとの相違は、本システムでは情報の生成や利用のための情報処理機能が関係によって管理されている点にある。情報の生成や利用のための情報処理機能が関係によって管理されているために、これまでのように部材などの情報をクラスとして定義する必要はなく、関係によって他の情報や機能と組み合わせることによって情報の定義が可能となっている。言い換えれば、これまでのシステムと本システムとの最大の相違は、情報の生成や利用が関係を中心として行われる点にあると言える。

9.2 今後の展望

本研究の表現方法は製品モデルのみでなく、情報の生成・利用・管理が複雑に絡み合う問題に対して有効と考えられる。また本研究の手法によって表現された製品モデルは、これまでの製品モデルにない特徴を有する。従って、本研究を進展させることによって現時点では未解決の問題を解決することが期待される。そこで最後に本研究の今後の展望を述べる。

(1)造船CIMへの展開

第8章において述べたように、造船CIMの構築のためには製品モデルのみでなく、CAEや生産環境モデルを含めて検討することが重要である。CAEや生産環境モデルも情報の生成・利用・管理が複雑に絡み合う問題であるため、CAEや生産環境モデルの表現方法として本研究の手法は基本的に有効と考えられる。また、CAEや生産環境モデルへと進展させることによって大規模なモデルあるいはシステムの表現方法について、より一般化された知見を得ることが期待される。従って、本研究を造船CIM全体へと展開することは非常に重要と考えられる。

(2)他の製品モデルへの展開

本研究を用いて製品モデルを構築するためには情報・機能・関係を明確に認識し、それらの構成として製品モデルを表現することが必要である。このため、本研究のシステムを用いて製品モデルを一から構築する時は、現実世界の様々な実体や事象を、情報・機能・関係に

よって整理して捉えることが要求される。従って、本研究のシステムをCASEツールとして用いると、情報や機能の明確化にも有効と考えられる。

本研究の実例として用いた船や海洋構造物の製品モデルでは、製品モデルを構成する情報や機能は既に明確になっており、上記の項目について検討することができなかった。そこで例えば橋梁などの他の製品モデルの構築に本研究のシステムを利用することによって、情報や機能の明確化への影響を検討することが可能と考えられる。

(3)システムの高効率化

本研究の手法によって製品モデルを構築すると、形状表現や数値計算などの要素技術部分は部品化される。また、このような部品の変更はシステム全体としての組み合わせに影響を与えないという特徴を有する。

ところで、形状表現や数値計算などの要素技術には、それぞれに適した言語がある。また本研究の上記の特徴を考慮すると、様々な要素技術部分をそれぞれに適した言語で構築しても、その統合は容易と考えられる。従って、製品モデルを構成する各部品を各々に適した言語を用いて構築することによって各部品の処理速度や信頼性は向上し、システムの高効率化が期待される。

(4)システム開発者の支援のために

本研究で構築されたシステムでは、製品モデルに必要な全ての属性情報や機能がライブラリに登録されている訳ではない。このため現状のシステムでは足りない部品があった時は、システム開発者がSmalltalk上で部品を定義することが必要である。しかしノードとリンクを組み合わせることのみによってシステムを構築することができれば、システム開発者を更に強力に支援することができる。

以上のような製品モデルの構築を可能とするためには、以下の何れかが要求される。

- a)システム開発者が必要とする全ての部品を、開発支援者が予め用意しておく。
- b)システム開発者が必要とする部品が明確になった時に開発支援者が部品を作成し、その部品をシステム開発者に提供する。

現実問題として上記a)は困難であるため、上記b)のような製品モデルの開発体制が望まれると考えられる。上記b)のような開発体制を確立するためには、製品モデルの開発における開発支援者の役割、システム開発者の役割などについて検討を加え、各々が製品モデルの開発のどの部分を行うべきかを明確にすることが重要と考えられる。

参考文献

A:

- [相澤 88] 相澤・前川編著：CAE（新製品開発・設計支援コンピュータ・ツール）、共立出版、1988
- [青山 95] 青山和浩：造船における設計・生産情報の獲得支援のためのモデリングに関する研究、東京大学博士学位論文、1995
- [青山他 95] 青山和浩・野本敏治・澤田和弥・渡辺祐介：造船設計における統合CAEシステムの構築－造船CIMの為の設計生産情報獲得支援システムの構築（その9）一、日本造船学会論文集 第178号、1995
- [雨宮 91] 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号および第170号、1990および1991
- [綾 96] 綾日天彦：インターネット時代の協創設計、情報フロンティアシリーズ12、共立出版、1996
- [荒井 91] 荒井良尚：経営工学、日本機械学会誌、Vol.94、No.873、1991

B:

- [Buckley 92] M.J.Buckley：Conceptual Design Tools－A Key Concept of Future CIM－、第2回国際CIMシンポジウム論文集、国際CIMシンポジウム運営委員会、1992

C:

- [CIMS 89] 造船CIMSに関する調査研究報告書、日本造船研究協会、1989
- [CIMS 91] 平成2年度 造船CIMSパイロットモデルの開発研究報告書、財団法人シップ・アンド・オーシャン財団、1991
- [Chen 76] P.Chen：The Entity-Relationship Model Toward a Unified View of Data、ACM Transactions on DataBase Systems Vol.1 No.1、1976
-

D:

- [土井 91] 土井憲治：造船CIMSのための倣い設計エキスパートシステムの研究、日本造船学会論文集 第170号、1991

F:

- [福田 93] 福田収一：計算力学とCAEシリーズ、コンカレントエンジニアリング、倍風館、1993
- [藤野 92] 藤野正隆：運動性能研究と船体構造設計、「第2回新しい造船学シンポジウム」テキスト、日本学術会議、1992
- [船岡他 92] 船岡宏一朗・覚野博幸：造船CIMSとは、日本造船学会誌 第759号、1992
- [古川 88] 古川勇二：パラダイム変化とこれからの生産システム、日本機械学会誌 Vol.91 NO.836、1988

G:

- [GPME 96] 平成7年度 組立産業汎用プロダクトモデルの開発研究報告書、財団法人シップ・アンド・オーシャン財団、1996
- [Gamma 95] E.Gamma・R.Helm・R.Johnson・J.Vlissides著、本位田真一・吉田和樹訳：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

H:

- [服部他 84] 服部幸英・笠原協之：造船におけるシステム化のあゆみ、造船におけるシステム技術 第1章、日本造船学会 システム技術委員会第I部会編、1984
- [人見(1) 90] 人見勝人：生産システム論、同文館、1990
- [人見(2) 90] 人見勝人：生産システム工学、共立出版、1990

I:

- [石黒他 95] 石黒憲彦・奥田耕士：CALS—米国情報ネットワークの脅威—、日刊工業新聞社、1995

- [伊藤 91] 伊藤公俊：メンタルモデルとインテリジェントCAD、インテリジェントCAD（下）、朝倉書店、1991
- [伊藤 92] 伊藤健：造船CIMSプロジェクト、日本造船学会誌 第759号、1992
- [岩城 76] 岩城正夫：原始技術史入門、新生出版、1976
- [岩田他 82] 岩田一明・沖野教郎・佐田登志夫・橋本文雄・室津義定・大場文憲：生産システム学、コロナ社、1982

J:

- [Jacobson 95] I. Jacobson他著、西岡利博他訳：オブジェクト指向ソフトウェア工学 OOSE -use-caseによるアプローチ、トッパン、1995

K:

- [河村 95] 河村一樹：ソフトウェア工学、ソフトバンク、1995
- [木村 86] 木村文彦：CAD/CAM統合化のためのプロダクトモデリングと知識処理、日本機械学会誌 Vol.89 No.815、1986
- [木村 92] 木村文彦：CIMにおけるCAD/CAMの新展開、第2回国際CIMシンポジウム論文集、国際CIMシンポジウム運営委員会、1992
- [小山 89] 小山健夫：総論（造船CIMにむけて）、造船におけるシステム技術、システム技術委員会シンポジウムテキスト（日本造船学会編）、1989

M:

- [南崎 96] 南崎邦夫：船舶建造システムの歩み、成山堂書店、1996
- [Minsky 90] M.Minsky著、安西祐一郎訳：心の社会、産業図書、1990

N:

- [中井 90] 中井裕司：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、pp.623-628、1990
- [長瀬 92] 長瀬裕：造船CIMSの概要、日本造船学会誌 第759号、1992
- [野本他 94] 野本敏治・武市祥司・青山和浩：製品定義システムにおける建造ブロック搭載シミュレーションシステムの開発、日本造船学会論文集第176号、1994
-

O:

- [大須賀 85] 大須賀節雄：次世代CAD/CAMのための知識処理の応用、マグロウヒルブック、1985
- [大高 95] 大高哲彦：STEP-製品のライフサイクル支援のためのプロダクトモデル、「CALC-その目的と基盤技術」講習会テキスト、日本機学会、1995
- [大坪 87] 大坪英臣：船体構造設計におけるDesign by Analysisのための技術開発、第13回日本造船学会夏期講座「新しい造船学」、日本造船学会、1987
- [沖野 93] 沖野教郎：生物型生産システム、浅倉書店、1993
- [小野寺 68] 小野寺力男：グラフ理論の基礎、森北出版、1968

R:

- [Rumbaugh 92] J.Rumbaugh他著、羽生田栄一監訳：オブジェクト指向方法論OMT、トッパン、1992

S:

- [SR207] 第207研究部会 総合報告書：船殻構造の強度評価と管理目標の定量化の調査研究、日本造船研究協会、1993
- [坂本 91] 坂本千秋：CIMにおける開発と生産の統合化、コンピュータロール36、コロナ社、1991
- [樫木/河村 81] 樫木義一・河村和彦：参加型システムズ・アプローチ手法と応用一、日刊工業新聞社、1981
- [鈴木 91] 鈴木宏正：プロダクトモデリングとインテリジェントCAD、インテリジェントCAD（下）、朝倉書店、1991
- [砂川 83] 砂川祐一：造船におけるNC、造船におけるシステム技術、日本造船学会 システム技術委員会第I部会編、日本造船学会、1983

T:

- [武市 96] 武市祥司：統合化生産システムにおける溶接構造物の精度管理に関する基礎的研究、東京大学博士学位論文、1996
- [田渕他 95] 田渕寛・Y.Jin：GPMEプロジェクトについて、日本造船学会誌 第759号、1995

- [寺野 93] 寺野隆雄：知識システム開発方法論、浅倉書店、1993

V:

- [Vietch 96] J.Vietch：Dynamic Objects (Beyond Component Software)、Object World Expo / Tokyo 1996 Conference Notes、IDG World Expo Japan & Object Management Group、1996

W:

- [和田(1) 90] 和田龍児 監修：CIM/MAP戦略絵とき読本、オーム社、1990
[和田(2) 90] 和田龍児 監修：CIM/MAP実戦絵とき読本、オーム社、1990
[渡辺 77] 渡辺茂・須賀雅夫：システム工学とは何か、NHKブックス、1977

Y:

- [山口 88] 山口富士夫：コンピュータディスプレイによる形状処理工学 [I][II][III]、日刊工業新聞社、1988
[吉川/富山 89] 吉川弘之・富山哲男：インテリジェントCAD(上)ー理念とパラダイム一、朝倉書店、1989
[米澤/柴山 92] 米澤明憲・柴山悦哉：岩波ソフトウェア科学17 モデルと表現、岩波書店、1992

Z:

- [造船学会 88] 日本造船学会：新しい造船学、第13回夏期講座、1988
-

本論文に関連した発表論文

本論文に関連した発表論文を以下に示す。

1. 野本敏治・青山和浩と共著、関係情報の記述による船体構造の設計法に関する基礎的研究、日本造船学会論文集 第177号、1995.8
 2. 野本敏治・青山和浩・酒巻宏典と共著、海洋構造物を対象とした設計支援システムに関する基礎的研究、日本造船学会論文集 第178号、1995.12
 3. 野本敏治・青山和浩と共著、海洋構造物を対象とした設計支援システムに関する基礎的研究（その2）－海洋構造物の構造設計支援システムの構築－、日本造船学会論文集 第180号、1996.12
 4. 野本敏治・青山和浩と共著、造船の設計支援システムの部品化に関する研究、日本造船学会論文集 第180号、1996.12
-

謝 辞

本研究を行うにあたり、ご支援、ご指導を頂いた全ての関係者の皆様に感謝いたします。

指導教官である野本敏治教授には、時に厳しく、時に温かく、本研究に関するご指導・ご助言を頂きました。野本先生のご指導がなければ本論文は完成しなかったことと思います。更に研究内容に関するご指導のみでなく、卒論の時には研究の楽しさを、大学院では研究者としての心構えや研究の進め方など、本当に幅広く温かいご指導をいただきました。野本先生のご指導は今後研究を行う上で大きな財産となりました。ここに改めて厚く御礼申し上げます。

青山和浩講師には、研究の具体的な方向性や方針に関する数多くのご指導をいただきました。元々は計算機を余り知らなかった私がこのような論文を執筆することができたのも、青山先生のご指導あつてのことと思います。また研究に関するご指導のみでなく、公私にわたる様々な悩み・疑問に対して、兄のように親身なアドバイスをいただきました。心より感謝いたします。

東京大学大学院工学系研究科船舶海洋工学専攻の町田進教授、吉田宏一郎教授、小山健夫教授ならびに大坪英臣教授の諸先生方には、本論文の審査を通じて貴重なご指導、ご指摘を頂戴しました。大変お忙しい中、貴重なお時間を頂戴して審査していただきましたことに、ここに改めて感謝の意を表します。

船舶海洋工学科助手の武市祥司氏とは公私にわたって共に多くの時間を過ごし、研究に関する多くの議論を行いました。更に論文の執筆に至っては激励の言葉を何度も頂きました。ここに御礼申し上げます。

澤田和弥氏をはじめとする生産システム工学研究室の先輩・後輩の皆様には、研究室において多くの議論を交わすとともに、様々な面においてお世話になりました。研究室での議論を通じて本研究は少しずつ明確になっていきました。また論文の作成に当たっては数多くのご協力を頂きました。感謝の意を表するとともに、すばらしい論文を完成させることをお祈りいたします。

秘書の恩田実代子さん、元秘書の鈴木聡子さんには特に事務全般にわたってお世話になりました。ここに深謝いたします。

多くの造船所の方々から、本研究に関して様々な視点からご支援・ご指摘を賜りました。本研究が少しでも皆様のお役に立てば幸いです。

また、本年度5月に急逝された野本教授令夫人文恵様には、温かいお言葉を何度もいた

きました。ここに御礼申し上げるとともにご冥福をお祈りいたします。

最後に、トヨタ自動車を退社し大学に戻り勉強をしたいという私の我儘を許し、温かく見守ってくれた両親にも感謝します。

濱田邦裕

Appendix-A

オブジェクト指向

本研究では、計算機内にシステムを実装する際に、オブジェクト指向言語であるSmalltalkを利用した。そこで、オブジェクト指向について述べる。

A.1 オブジェクト指向の基礎概念

本節では、オブジェクト指向の基礎概念について述べる[1]-[5]。

A.1.1 オブジェクト指向

オブジェクト指向の「オブジェクト(Object)」は、「1.もの、物体 2.目的、対象」と訳すことができる。「オブジェクト」を「もの、実体」として理解した場合、「実体」には「1.事物の本体、内容 2.哲学的に、不変の本質的存在」という意味がある。つまり、「オブジェクト指向」の中心となる「オブジェクト」は現実世界に存在する「もの」と密接な関係を持ち、「もの」の内容、本質的存在を表現するものであると考えることができる。

従って、現実の世界が多くの「もの」で成立ち、これらの「もの」の関係によって日々の営みが行われるように、「オブジェクト指向」の世界も様々な「オブジェクト」によって構成され、それらが互いに機能する世界であると考えることができる。

A.1.2 オブジェクト指向によるプログラミング

「もの」を中心に考える概念である「オブジェクト指向」を、プログラミングによってシステムの概念として利用するということは、「オブジェクト指向」の世界を計算機の中に表現することである。それは同時に「オブジェクト指向」の中心である「オブジェクト(もの)」を計算機内部に取り込む必要があるということである。

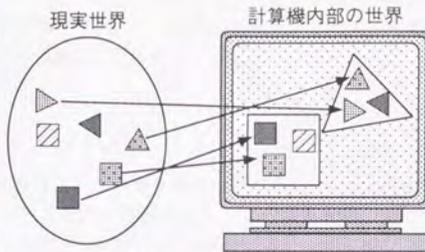


Fig.A-1-1 オブジェクト指向によるプログラミング

当然のことながら、現実社会の形あるものをそのまま計算機内部に取り込む事は困難である。従って、何等かの別の表現方法（計算機内で表現できるもの）を用いて取り込むことが必要となり、モデル化が行われることになる。つまり、このモデル化によって、現実社会に存在する「もの」を、計算機内部に「オブジェクト」として表現することが可能となるため、モデル化による表現方法が「オブジェクト指向プログラミング」の骨格となる。つまり、モデル化された「もの」が実世界の「もの」と対応することによって、「もの」によって成り立つ現実社会を計算機内部に取り込むことが可能となる（Fig.A-1-1）。

A.1.3 オブジェクト

手続き型のプログラミング言語では、情報とそれに対する処理機能とは切り放して記述するのが通常となっている。しかし、情報が変化すれば、それを扱う処理機能は変化すべきであり、情報の定義と処理機能の定義とは密接に絡み合い不可分なものであると考えられる。そこで「オブジェクト指向」では現実の「もの」が属性と機能を一体として保持しているように、「オブジェクト」に対して「内部データ（情報）」と「アルゴリズム（処理機能）」を一体化(Package)して定義している（Fig.A-1-2）。

例えば「W ton のタンカーA」を「オブジェクト」として定義することを考えてみる。「タンカー」を「オブジェクト」として定義するためには、先ず「タンカー」の表現方法を考えなければならない（モデル化）。タンカーを分析すると、大きさ、形状、積載能力等のどちらかという「もの」の属性に深く依存する情報部分と、海上を航行する、油を積むといった「もの」の機能に関する操作部分から成り立っている。このため

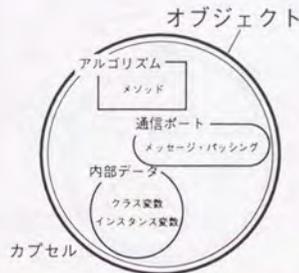


Fig.A-1-2 オブジェクト

「タンカーA」をオブジェクトとして定義すると、オブジェクトに対して、大きさ、形状、積載能力等の属性値（内部データ）と、海の上を航行するといった機能を満たすための操作（アルゴリズム）が記述されることになる。

A.1.4 メッセージとメソッド

現実の世界では船を操縦したり、油を積んだりするためには人間が何等かの操作を船に対して行う。「オブジェクト指向」の世界でも、この現実世界と同じように、「オブジェクト」として定義されたタンカーに対して「操縦する」「油を積む」という操作が加えられる。この操作が「メッセージ」と呼ばれるものであり、「オブジェクト」に操作を加える唯一の方法として準備されている。具体的に「前進しろ」というメッセージをタンカーに送ると、そのメッセージを受けたタンカーは自分が持つ情報（速力）と処理機能（速力いくつで航行する）とによって前進するという動作を実行する。一般的にオブジェクトは、他の任意のオブジェクト（自分自身も含めて）にメッセージを送信（メッセージ・パッシング）することが可能であり、メッセージを受け取ったオブジェクトは、メッセージに対応する処理を開始する。メッセージは、他のプログラミング言語の手続き呼び出しや関数呼び出しに相当する。また、呼び出されるものである「オブジェクトの動作を記述したもの」を特に「メソッド」と呼ぶ。メソッドはメッセージ・パッシングによって起動するアルゴリズムに相当するもので、「オブジェクト」に記述された「処理機能」のことである(Fig.A-1-3)。

また、オブジェクトを操作するのはメッセージのみとすることを実現している。従って、オブジェクトに定義されている操作を通してのみオブジェクトは操作されると約束を決めることによって、オブジェクトの外的な挙動は完全に特徴づけられる（データ抽象）。また、オブジェクトが持つ情報も、オブジェクトに対するメッセージ（操作）のみによってその入力、書き換えが行われる（情報隠蔽）。

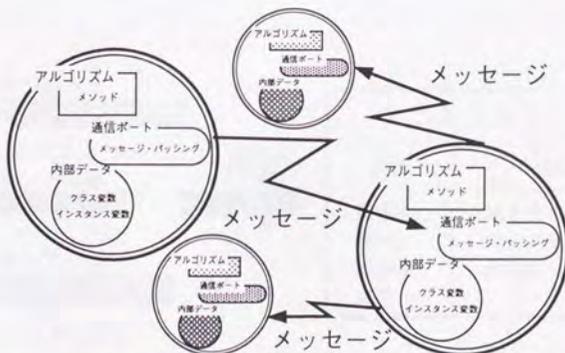


Fig.A-1-3 オブジェクト間のメッセージ通信

A.1.5 クラスとインスタンス

現実の世界では、タンカーと呼ばれるものは数多く存在する。逆の見方をするとこの船はタンカーに属すると考えることができる。この様に、あるものを分類するという事は、分類する対象となるものの機能、属性を基にして一般化しているのであり、現実世界の「もの」の表現方法としては重要なことである。前述した「タンカーA」の様に、「タンカーB」「タンカーC」を各々オブジェクトとして定義することは可能だが、オブジェクトが持つ機能、属性を基に分類すると、これらのタンカーは「タンカー」という項目で分類整理することができる（一般化）。この分類の項目となる「タンカー」を「オブジェクト指向」ではクラスと呼び、クラスは同一の情報構造を持つオブジェクトを生成する際の「ひな型」となる。つまり、「タンカーA」も「タンカーB」も「タンカー」というクラスから生成される「オブジェクト（インスタンス）」であり、オブジェクトが有する属性値（大きさ、形状等）の具体的な値が異なるものとして生成される(Fig.A-1-4)。

実際のプログラミングでは、例えばタンカーを一般的に表現するためには、属性値を納める変数として何が必要か、操作に対応する処理機能（メッセージに対応するメソッド）はどのようなものが必要か等の記述が、クラス「タンカー」に対して行われる。その後、このように定義されたクラスを基に、「タンカーA」等の現実世界の「もの」に対応し、実際的な属性値（データ）を有する「インスタンス」が生成されることになる。

A.1.6 インヘリタンス（継承）

オブジェクト（クラスオブジェクト）は階層構造を構成することが可能であり、上位クラス（スーパークラス）から下位クラス（サブクラス）にデータ構造（変数）やメソッドなどの性質（機能）を継承（インヘリタンス）することが可能である。これによって共通点のあるクラスをその共通点において抽象化した親クラス（スーパークラス）を定義することがで

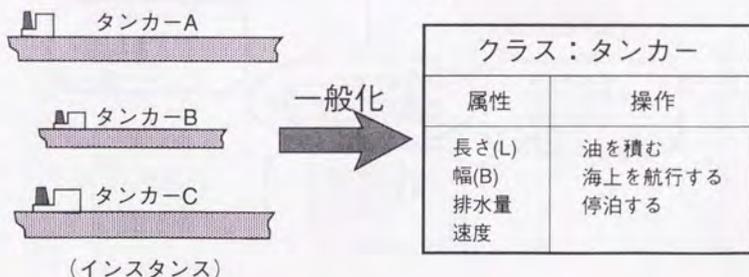


Fig.A-1-4 クラスとインスタンス

き、データの構造化に役立っている（抽象化）。例えば、「タンカー」と「バルクキャリア」というクラスにおいては、積荷の種類、船体構造は異なるが、船としての機能は同じである。従って「船」というスーパークラスに、「船」に共通の属性（例えば鋼構造）と機能（例えば、海上（水面上）を航行する）とを定義し、「タンカー」「バルクキャリア」をそのサブクラスとして定義する(Fig.A-1-5)。

以上のようにスーパークラス、サブクラスを定義すると、「タンカー」「バルクキャリア」はスーパークラスである「船」から船としての属性と機能とを継承することが可能となる。また、「タンカー」「バルクキャリア」が有する特有の性質（例えば「タンカー」は油を積むとか「バルクキャリア」は二重底構造を持つバラ積み専用船である）をそれぞれのクラスで定義する。そうすれば「タンカー」は鋼できており、海上を航行し、油を積むというクラス「タンカー」を定義することができる。

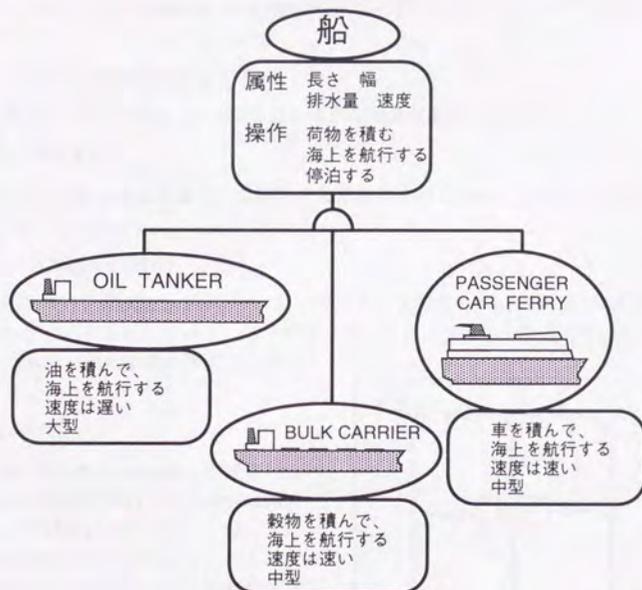


Fig.A-1-5 インヘリタンス

A.2 オブジェクト指向方法論

オブジェクト指向開発については、以下に示すような非常に多くの効用が宣伝されており、時には非現実的な期待さえかけられている[6]。

- ・ 開発速度の向上
- ・ 以前の成果の再利用
- ・ モジュール化アーキテクチャ
- ・ 問題領域への適合

しかし、オブジェクト指向に基づいてソフトウェアを設計することは簡単なことではなく、オブジェクト指向に基づいて再利用可能な良構造のソフトウェアを設計することはさらに難しい[7]。このため、これまでに様々なオブジェクト指向方法論が提案されてきた。

オブジェクト指向方法論は、オブジェクトの表記法を中心とした第一世代を経て、現在は第二世代の方法論が用いられている[8]。第二世代の方法論に共通した特徴として

- ・ 分析(OOA: Object Oriented Analysis)、設計(OOD: Object Oriented Design)、実装(OOP: Object Oriented Programming)の3段階に分けてオブジェクト指向開発を捉えている。
- ・ オブジェクトの表記法を有する。
- ・ ラウンドトリップ(Fig.A-2-1)によるシステム開発が重視されている。

などが挙げられる。

第二世代の代表的な方法論として以下の方法論が挙げられる。これらの方法論の特徴をTable A-2-1に示す[9]。

(1)Shlaer & Mellor手法[10]

本手法では、現実世界をオブジェクトモデル、状態モデル、プロセスモデルの3つのモデルを中心に分析している。この手法は第二世代方法論の中で最も早く提案された手法であり、他の手法の原形となっている。

(2)OMT手法[11]

OMT手法はRumbaughらが開発した手法である。OMT手法の特徴は図表などが詳細に決定され、十分な記述能力を有する点にある。このため、本研究でもオブジェクトの表記法はOMT手法の表記法を利用している。但し、設計フェーズに関しては他の手法より弱い。

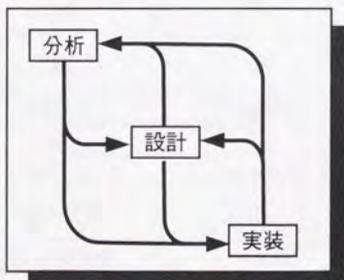


Fig.A-2-1 ラウンドトリップ

(3)Coad & Yourdon手法[12][13]

この手法の特徴は、オブジェクトの確定、構造の定義、属性の定義、メソッドの定義をボトムアップによって同時平行に行うべきとしている点、言い換えればラウンドトリップによる開発を非常に重視している点である。但し、動的な構造に関しては他の手法より弱い。

また、他の手法としては、特にシステムの利用の方法(use-case)に着目したOOSE手法[14]などが挙げられる。

以上に示したように、オブジェクト指向方法論には万能の方法論などは存在せず、様々な手法があり、それらは互いに長所・短所がある。従って、システムの開発者が開発するシステムの特徴に応じて、様々な方法論を使い分けていくことが必要である。

Table A-2-1 オブジェクト指向方法論の比較 [9]

手法	Shlaer & Mellor 手法	OMT手法	Coad & Yourdon 手法
開発者/ 会社	Sally Shlaer/ 米Project Technology社	J.Rumbaughなど/ 米General Electric社 研究開発センター	Peter Coad/ 米Object International社
概要	意味モデルをベースにしている	構造化分析からデータフローと状態遷移図を受け継ぎ、オブジェクトモデリングはE-R図などの影響	E-R図やShlaer & Mellorの手法とOOPおよび知識ベース・システム概念を合わせて発展させたもの
記述力	高	高	低
分析・設計 プロセス	自然	自然	—
OODへの 連結度	低	高	高
発表年	1988年	1991年	1990年
適用実績	豊富(1979~)	少	少
支援ツール	Teamwork	OMTool	OOATool
ノウハウの 公開度	低	極めて高	高
対象規模	小~大規模	小~中規模	小規模

A.3 オブジェクト指向と本研究

前節までに述べてきたように、オブジェクト指向は万能な方法ではなく、システム開発を行う際には様々な配慮が必要である。また、青木が指摘しているように、オブジェクト指向の構造中心の考え方と、手続き型言語の機能中心の考え方には、互いに長所・短所がある[15]。従って、盲目的にオブジェクト指向を用いるのではなく、目的と対象に応じてオブジェクト指向的な考え方と他の考え方をミックスすることが重要である。

ところで本文中で述べたように、製品モデルのような製品を表現するための多種多様な情報を統合的に取り扱う環境では、参照関係や生成関係などの関係をシステム上に残すことは非常に重要と考えられる。しかしオブジェクト指向言語を用いた場合、クラスAとクラスBとの関係としてシステム上に残せるのは「汎化-特化の関係」のみとなる。関係を表現するクラスを定義することは可能であるが（接合関係などはこれに当たる）、その場合も実際に関係によって結び付けられるのはインスタンス同士となる。従って、オブジェクト指向言語を用いて部材や形状をクラスとして定義すると、クラス間の関係をシステム上に残すことはできなくなる。

また、ある情報を他の関係する情報を考慮して矛盾なく生成するためには、情報間の関係のみでなく情報を生成するための情報処理機能が重要である。例えば、部材情報を生成するためには、部材の形状を矛盾なく生成することが必要であり、部材を生成するための情報処理機能と部材の形状を生成するための情報処理機能とを同時に考える必要がある。従って、情報の生成に着目すると、関係と機能とを同時に検討することが重要である。オブジェクト指向のような構造主義的な言語では当然のことながら、関係と機能とを同時に検討することは困難である。

このような製品モデルの特徴を考慮すると、システム開発の際にオブジェクト指向言語を用いることは必ずしも得策ではなく、製品モデルの特徴を踏まえた開発環境が望ましい。

本研究では、オブジェクト指向言語であるSmalltalkを用いてシステムを構築している。しかし詳細には、SmalltalkがC言語を用いて構築されているのと同様に、Smalltalk言語を使用して、情報や機能の間の関係や、関係と機能とを同時に考慮することのできる環境を構築し、この環境の上に製品モデルを構築している。言い換えれば、製品モデルの構築用の言語を定義したと言うこともできる。

以上のように、本研究とオブジェクト指向言語との最大の相違は、クラス間の様々な関係を記述できる点にある。この関係が存在することによって、システムの柔軟性・拡張性が増加し、システムの構成そのものも把握しやすくなったと考える。

参考文献

- [1] 竹内郁雄・梅村恭司：Smalltalk-80入門、サイエンス社、1986
 - [2] 上谷見弘：統合化プログラミング環境 Smalltalk-80とInterlisp-D、丸善、1987
 - [3] A.Goldberg・D.Robson：SMALLTALK-80 一対話型プログラミング環境一、オーム社、1986
 - [4] A.Goldberg・D.Robson：SMALLTALK-80 一言語詳解一、オーム社、1987
 - [5] B.Meyer：オブジェクト指向入門、アスキー出版、1990
 - [6] B.F.Webster著、細井拓史訳：オブジェクト指向開発の落とし穴、Prentice Hall、1995
 - [7] E.Gamma・R.Helm・R.Johnson・J.Vlissides著、本位田真一・吉田和樹訳：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
 - [8] 羽生田栄一：統一方法論を使ったオブジェクト指向分析・設計手法、オブジェクト指向分析・設計手法セミナー資料、1995
 - [9] 本位田真一・山城明宏：オブジェクト指向システム開発、日経BP出版センター、1993
 - [10] S.Shafer・S.J.Mellor著、本位田真一・山口亨訳：オブジェクト指向システム分析、啓学出版、1990
 - [11] J.Rumbaugh他著、羽生田栄一監訳：オブジェクト指向方法論OMT、トッパン、1992
 - [12] P.Coad・E.Yourdon：Object-Oriented Analysis、Prentice Hall、1991
 - [13] P.Coad・E.Yourdon：Object-Oriented Design、Prentice Hall、1991
 - [14] I.Jacobson他著、西岡他訳：オブジェクト指向ソフトウェア工学OOSE-use-caseによるアプローチ、トッパン、1995
 - [15] 青木淳：オブジェクト指向システム分析設計入門、SRCハンドブック、1993
-

Appendix-B

海洋構造物の製品モデル

7.3節において、本研究のプロトタイプ・システムを用いて表現した製品モデルの例として海洋構造物の製品モデルを紹介した。海洋構造物の製品モデルはFig.B-1-1に示すような多種多様な海洋構造物の全てを対象とする一般化された製品モデルであり、SODASで定義されている船の製品モデルをベースとして構築されている。

Appendix-Bでは海洋構造物の製品モデルについて述べる。

B.1 海洋構造物の製品モデルの構築のために

製品モデルを構築するためには、本研究で検討した表現方法に加えて、設計・生産活動の支援の視点、即ち

- ・上流から下流までの全ての設計・生産過程における様々な製品の認識に対応して製品を表現できること

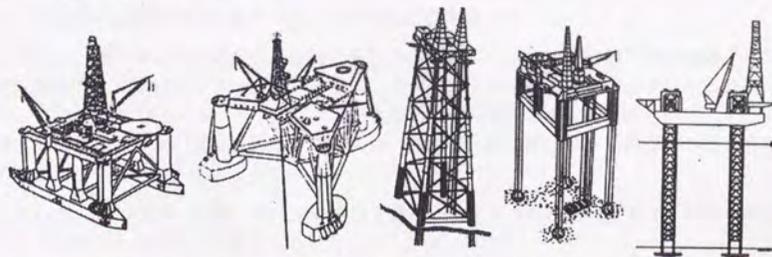


Fig.B-1-1 様々な海洋構造物

・上流の設計において生成された製品モデルの情報を有効に利用して、下流の設計における製品情報の生成を支援すること

が重要である。そこで本節では、海洋構造物の様々な捉え方について検討し、その後に製品情報の生成を支援する設計機能について検討する。

B.1.1 海洋構造物の様々な捉え方

(1) 要求される機能からの捉え方

海洋構造物は、「海洋上に十分な広さを持った活動の場を提供する」ことを目的に設計される。従って、海洋構造物には「十分な広さを持った活動の場」、「活動の場を海面上に保持するための支持」、および「活動の場や支持の荷重に対する反力を生む部分」が要求される。ここで、「安全で十分な広さを持った活動の場」を上部要素、「活動の場を海面上に保持するための支持」を支持要素、「活動の場や支持の荷重に対する反力を生む部分」を下部要素と呼ぶことにし、これらの要素を総称して機能要素と呼ぶことにする。

各々の機能要素は、最終的には異なった構造を有する。例えば、浮遊式の海洋構造物の下部要素は複数のローハルで構成され、接地式の海洋構造物の下部要素はケーソン部分等になる。しかし、これら異なる構造が満足すべき機能について考えると、下部要素という同じ機能要素によって認識することが可能である。

以上のように、海洋構造物の満足すべき機能について考えると、海洋構造物一般を表現できる機能要素を認識することができる。このような3つの機能要素による海洋構造物の捉え方は、詳細な情報が生成される以前の概念設計における海洋構造物の捉え方と考えることができる。

(2) 全体的な構造を検討する際の捉え方

上記(1)において述べた機能を満足するために、海洋構造物全体の構造や位置関係を決定する際の認識について考えてみる。このような場合、具体的な部材などといった詳細な物の捉え方ではなく、海洋構造物を構成する主要な構造に視点を絞り、ローハル、コラム、ブレースなどの主要な構造の集まりとして海洋構造物を捉える。

例えば、海洋構造物の構造を総合的に検討する際には、全体を立体的な骨組構造として強度計算を行うために、ローハル、コラム、ブレースなどの構造を直線のフレームと捉える[1]。ここで、ローハル、コラム、ブレースなどを海洋構造物の主要構造と呼ぶことにすれば、海洋構造物全体の構造を検討する際には、設計者は海洋構造物を主要構造の集まりとして認識していると考えられることができる。

従って、(1)で述べた機能を受け持つ主要な構造が決定される初期設計では、ローハル、コラムなどの主要構造が認識されると考えることができる。

(3) 詳細な検討を行う際の捉え方

上記(2)では、海洋構造物を主要構造の集まりとして捉えたが、他の捉え方が必要な場合もある。以下、詳細な検討を行う際の海洋構造物の捉え方について述べていく。

a) 空間の集合としての捉え方

浮体構造では部材などに囲まれた閉空間の情報が重要である。浮力や安定性に関する検討を行う場合には容積の情報が重要であり、海洋構造物を複数の空間の集まりとして捉えることが必要になる。

海洋構造物の有する様々な空間は、主要構造の内部に存在する。従って、海洋構造物の安定性などについて検討する場合には、各主要構造の内部の板で囲まれた閉空間に着目し、様々な空間を取り出すことが要求される。

b) 部材の集合としての捉え方

海洋構造物は複数の鋼板が加工されて、箱または円筒状に組み立てられることによって構成される物が多い。このため、詳細な構造強度などを検討する詳細設計、生産を考慮する生産設計においては、海洋構造物を部材の集まりとして捉える。

海洋構造物が有する部材には、主要構造を構成する部材と、複数の主要構造の間の格点部分を構成する部材とがある。従って、詳細設計や生産設計では、主要構造や主要構造の間の格点部分を詳細に捉え、部材の集まりとして海洋構造物を捉えることが要求される。

(4) 海洋構造物の捉え方の整理

以上のように、上流から下流へと設計が進行するにしたがって、異なる4つの視点によって海洋構造物は捉えられる(Fig.B-1-2)。

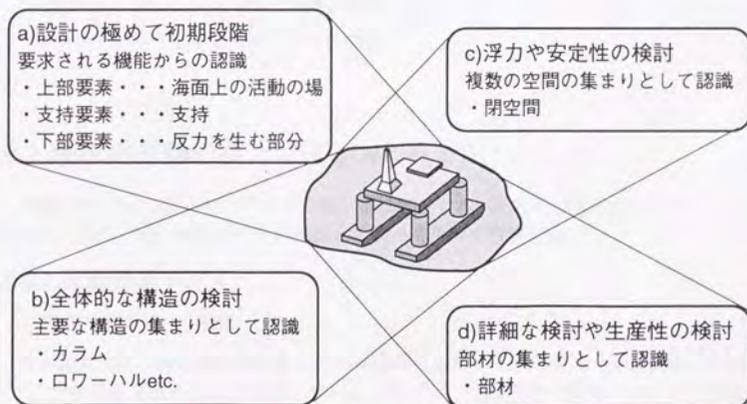


Fig.B-1-2 海洋構造物の様々な捉え方

- a)機能に着目して上部要素・支持要素・下部要素などの機能要素で海洋構造物を捉える視点
- b)ローハル、カラムなどの主要構造の集まりとして海洋構造物を捉える視点
- c)安定性などを検討するために空間の集まりとして海洋構造物を捉える視点
- d)詳細な検討や生産を考慮するために部材の集まりとして海洋構造物を捉える視点

従って、海洋構造物の製品モデルを構築するためには、上記4つの認識に対応するモデルを定義することが重要と考えることができる。

B.1.2 海洋構造物と設計機能

前項では、海洋構造物の様々な捉え方について検討し、海洋構造物を捉える4つの視点を明確にした。しかし、これらの認識に対応するモデルを構築しただけでは、個々のモデルが独立して定義されるのみである。従って、個々のモデルの有する情報は、そのモデルの中でしか利用することができず、製品モデルとしては不十分である。

上流の設計で決定される情報と下流の設計で決定される情報とは密接に関連しており、新たな情報が生成される際に既に定義されている情報は有効に利用できる。本研究では、このような情報の生成を設計機能によって実現することを考えた。

海洋構造物の製品モデルには、概念設計の3つの機能要素、初期設計で重要な主要構造、空間、部材などの情報が含まれる。従って、以下に示す設計機能が必要と考えられる。

- (1)3つの機能要素の情報の生成を支援する設計機能
- (2)様々な主要構造の情報の生成を支援する設計機能
- (3)空間の情報の生成を支援する設計機能
- (4)部材の情報の生成を支援する設計機能

B.2 海洋構造物の様々なモデル

本節では、B.1.1項において述べた海洋構造物の4つの捉え方に対応するモデルについて検討する。更に、船の製品モデルの情報との関係も併せて考察する。

B.2.1 機能要素モデル

(1)機能要素モデルの定義

概念設計では、海洋構造物に要求される機能に着目して上部要素・支持要素・下部要素という3つの機能要素で海洋構造物を捉える。従って、概念設計の段階では、海洋構造物を構成する個々の要素の存在と、各要素の接合に関する情報が獲得されることになる。

本研究では、これらの各要素を表現するモデルとして機能要素モデルを定義する。

機能要素モデルには、どのような機能を満足すべきかという情報と、自分と接合関係にある他の機能要素モデルの情報が記述される。また、海洋構造物のモデルは常に3つの機能要素モデルを有する訳ではなく、2つ又は1つの機能要素モデルによって表現される場合もある。

機能要素モデル：海洋構造物を機能面から捉えた上部要素・支持要素・下部要素という3つの機能要素を表現するモデル

(2)機能要素モデルの形状表現

接地式の海洋構造物では、海洋構造物を設計することが決定された瞬間に、海洋構造物の高さ（上部要素と下部要素の間の距離）が必然的に決定される。また、浮遊式の海洋構造物では、支持要素を構成する主要構造（カラムなど）は、同じ高さである場合が多い。

以上を考慮して、本研究では、概念設計における海洋構造物の形状を1本の線分として表現することにした。即ち、上部要素、下部要素に自身の位置を表現する「点」を記述し、更に、それを結ぶ支持要素には「線」の情報を記述することによって、海洋構造物の形状を1本の線分として表現している。

以上より、海洋構造物を機能要素モデルの集まりとして認識することによって、以下の情報が獲得される(Fig.B-2-1)。

- 1)構造物全体を構成する機能要素モデルの情報
- 2)機能要素モデル間の接合関係
- 3)構造物全体の形状を表現する1本の線分の情報

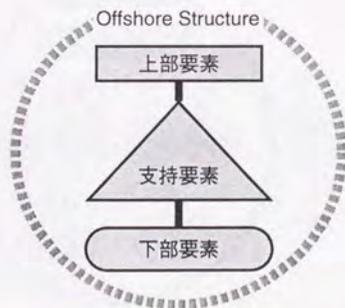


Fig.B-2-1 機能要素モデル

B.2.2 主要構造モデル

(1) 主要構造モデルの定義

全体的な構造を検討する初期設計では、海洋構造物を大きな視点で捉え、ローワーハル、コラムなどの主要構造が認識される。本研究では、これらの情報を表現するために主要構造モデルを定義する。

主要構造モデルは、どの主要構造であるかという情報と、他の主要構造との接合関係の情報とを有する。また、主要構造モデルが機能要素モデルに記述されることによって、各々の主要構造がどの機能要素に属するかという情報が表現される。

主要構造モデル：海洋構造物の全体的な構造を検討する際の個々の主要構造を表現するモデル

(2) 主要構造モデルの形状表現

主要構造モデルを用いた海洋構造物の形状表現について考えてみる。B.1.1項で述べたように、この時点では海洋構造物の形状を部材や空間の集まりとして詳細に捉える必要はなく、個々の主要構造を線分（又は線分の集まり）として表現する3次元的なワイヤー・フレーム・モデル（以下、WFモデルと略す）によって十分に表現可能と考えることができる。

そこで本研究では、各主要構造モデルの形状を1本の線分あるいは線分の集まりによって表現することにした。また、主要構造の間の格点部分は、線と線との交点として表現されている。

以上より、海洋構造物を主要構造モデルの集まりとして認識することによって、以下の情報が獲得される (Fig.B-2-2)。

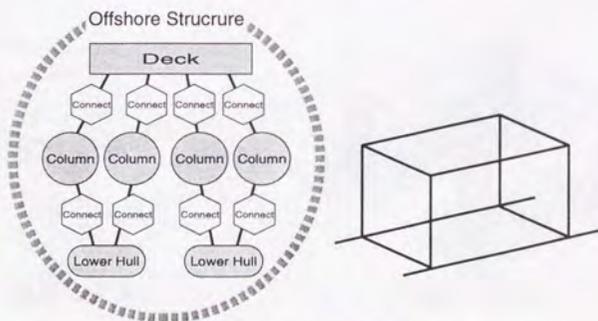


Fig.B-2-2 主要構造モデル

- 1)全体がどのような主要構造から構成されているかという情報
- 2)主要構造同士の接合関係
- 3)各主要構造の形状（線分）に関する情報

B.2.3 部屋モデルと部材モデル

海洋構造物の詳細な検討を行う際には、各々の主要構造を構成する空間や部材の情報が重要である。また、部材は他の部材と接合されているので、部材間の接合関係を表現する情報も獲得する必要がある。

これらの情報は、既に船の製品モデルで構築されている部屋モデル・部材モデル・接合関係情報のモデルと同一の物と考えることができる(Fig.B-2-3)。

B.2.4 船の製品モデルとの関係

以上より、海洋構造物の設計においては、上流から下流へと設計が進行するにつれて、設計対象が、機能要素モデル、主要構造モデル、部屋モデル、部材モデルへと、徐々に詳細度が増すものと考えられることができる。このようなモデルの定義によって、1本の線分が、海洋構造物全体・主要構造・空間や部材の輪郭の一部などの様々な形で認識されることになる。

また、機能要素モデルと主要構造モデルは、船の製品モデルには存在しないモデルである。このことは、船の設計と海洋構造物の設計とを比較することによって以下のように考えることができる。

海洋構造物の設計では、設計の極めて初期において海洋構造物に要求される機能が分解され、分解された機能に対応して実体である主要構造が設計される。これに対して船の場合、船に求められる「荷物を積んで安全に海上を航行する」という機能が1つの船体によって実現される。このため、船の設計では、設計の初期段階から船倉や部材などが決定される。し

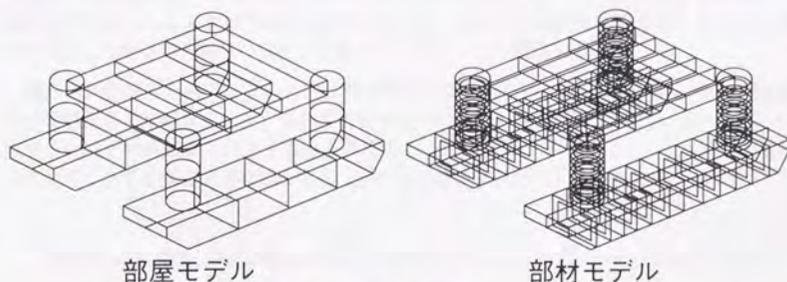


Fig.B-2-3 部屋モデルと部材モデル

かし海洋構造物の設計の場合には、海洋構造物に求められる機能を分解する概念設計を明確に認識することが重要になる。また、船の設計でも、双胴船などのように、設計の初期において機能を分解する必要がある場合には、海洋構造物と同様の考え方をすることができる。

従って、海洋構造物の製品モデルは、機能要素モデルと主要構造モデルという上流の設計部分を船の製品モデルに追加したものと捉えられる。

B.3 海洋構造物の製品モデルに必要な設計機能

B.3.1 機能要素モデルの生成の支援

機能要素モデルの生成は海洋構造物の最初の設計過程であり、これより上流のモデルは存在しない。従って、設計者の入力した情報を基に、1本の線分としての形状情報を獲得することが必要である。本研究では、各々の機能要素モデルと、下部要素から上部要素までの高さを設計者が入力する。そして、上部要素や下部要素を点、支持要素を線と捉え、機能要素モデルと高さの情報を基に1本の線分としての形状情報が獲得される。

B.3.2 主要構造モデルの生成の支援

機能要素モデルで獲得された海洋構造物の形状を表現する1本の線分の情報は、個々の主要構造モデルを線分として表現し、海洋構造物全体を線分（主要構造）の集まりとして表現する際に、有効に利用されると考えられる。

そこで本研究では、海洋構造物全体を表現する1本の線分に断面を入力することによって、3次元的なWFモデルを生成することを考えた。この機能を、線構造設計機能として定義する。

線構造設計機能は、1本の線分の任意の位置に複数の断面が入力された場合、WFモデルが生成される機能である(Fig.B-3-1)。WFモデルを構成する線分は、入力された断面の最寄りの頂点の間を結ぶことによって生成される。また、このような単純な処理で対応できない場合のために、生成される線分の頂点を任意に入力することも考慮している。

線構造設計機能を利用することで、機能要素モデルで獲得される1本の線分から、構造物を主要構造モデルの集まりとして表現するWFモデルが獲得される。この時、WFモデルを構成する1本1本の線は個々の主要構造を表現することになる。また獲得されたWFモデルの交点の情報から主要構造同士の接合関係（即ち格点部分の認識）も自動的に生成される。

線構造設計機能：1本の線分と断面の情報を基に、3次元的なWFモデルを生成する機能

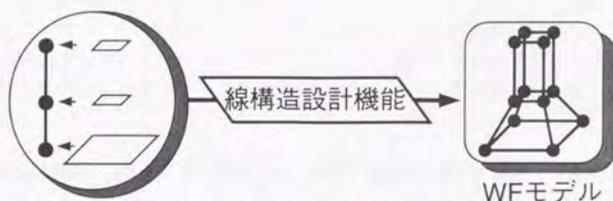


Fig.B-3-1 線構造設計機能

B.3.3 部屋モデルや部材モデルの生成の支援

部屋モデルの生成過程は、以下の2種類に整理することが可能である。

- (a1)主要構造を表現する1本の線分の情報を基に、主要構造の立体形状を表現する部屋が設計される過程
- (a2)主要構造の立体形状を表現する部屋の情報を基に、その内部に更に詳細な部屋が設計される過程

同様に、部材モデルの生成過程も、以下の4種類に整理することが可能である。

- (b1)主要構造を表現する線分の情報を基に、主要構造の外郭を構成する部材が設計される過程
- (b2)主要構造の内部に詳細な部屋を定義するための仕切り板が設計される過程
- (b3)主要構造を構成する部屋や部材の情報を利用して内構部材が設計される過程
- (b4)主要構造の間の格点部分の部材が設計される過程

ここで、(a2)および(b2)は、船の製品モデルにおいて既に構築されている空間設計機能が利用可能であり、空間設計機能を利用することによって、部屋・部材・接合関係情報が獲得される。また、(b3)についても、内部構造設計機能が船の製品モデルにおいて既に提案されている。内部構造設計機能を用いれば、主要構造や部屋の立体形状を利用して部材情報および接合関係情報が獲得される。

従って、新たに定義することが必要なのは、(a1)、(b1)、および(b4)の場合において製品情報の生成を支援する設計機能と考えられる。以下、各々について検討する。

(1)主要構造の情報を利用した部屋や部材の生成

ここでは、主要構造の有する線分の情報を利用して、主要構造の外郭を構成する個々の部材の面分や、板に囲まれた内部の空間を表現する部屋の情報が獲得される。本研究では、まず、前述の線構造設計機能を主要構造に適用して主要構造の立体的な形状を表現するWFモデルを生成する。次に、そのWFモデルを基にサーフェス・モデルやソリッド・モデルの情報を生成し、部屋や部材の形状情報を獲得する。

まずWFモデルが生成されることによって、面分や立体の獲得に必要な稜線は全て定義される。そして、WFモデルを定義する際の断面の情報と複数の断面の間の頂点の関係を利用することによって、サーフェス・モデルやソリッド・モデルが生成され、部屋や部材の形状情報が獲得される。本研究では、この機能を立体設計機能と呼ぶ(Fig.B-3-2)。

立体設計機能：WFモデルの情報に基づき部屋や部材の情報を獲得する機能

(2)格点部分の部材の生成

格点部分は海洋構造物の構造的な特徴で、その設計は極めて重要である。格点部分はコラムやブレースなどの主要構造の接合部に対応するため、格点部分の部材の設計は主要構造モデルの情報を利用して部材を定義する設計と捉えられる。本研究では主要構造モデルの情報を利用して格点部の部材の設計を支援する機能として格点部材設計機能を定義している。

格点部材設計機能：主要構造モデルの情報を利用して、格点部の部材の設計を支援する機能

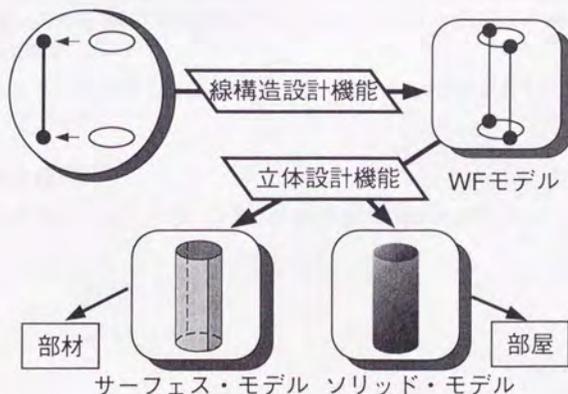


Fig.B-3-2 立体設計機能

B.4 システムによる設計例

B.4.1 設計の流れ

本システムを用いた設計の流れをセミサブ型（2ロワーハル）を例として以下に示す。

(1)機能要素モデルの生成

個々の機能要素モデルを表現するノードと、海洋構造物の高さが入力されることによって、機能要素モデルが生成される。海洋構造物全体の形状は1本の線分として表現されている(Fig.B-4-1(1))。

(2)主要構造モデルの生成

上部要素・下部要素を表現する「点」に対して、面分の情報が入力され、線構造設計機能を用いて海洋構造物の全体形状を表現するWFモデルが生成される。この時点では、カラムなどの個々の主要構造は線分として表現されている(Fig.B-4-1(2))。

(3)主要構造の立体形状の生成

個々の主要構造モデルを表現する線分に対して断面形状が入力され、部屋や部材が獲得される(Fig.B-4-1(3))。

(4)主要構造の内部の設計

空間設計機能や内部構造設計機能が用いられることによって、主要構造の内部の部屋、部材、接合関係情報などが設計される(Fig.B-4-1(4))。

(5)格点部分の設計

主要構造の間の接合部に優劣の情報が入力されることによって、主要構造の間の関係が決定される。更に内部構造設計機能が用いられることによって、格点部分の補強部材が設計される(Fig.B-4-1(5))。

このようにして、最終的に(Fig.B-4-1(6))に示すような構造が設計される。

B.4.2 様々な設計例

本システムを用いて設計した様々な海洋構造物の例をFig.B-4-2に示す。

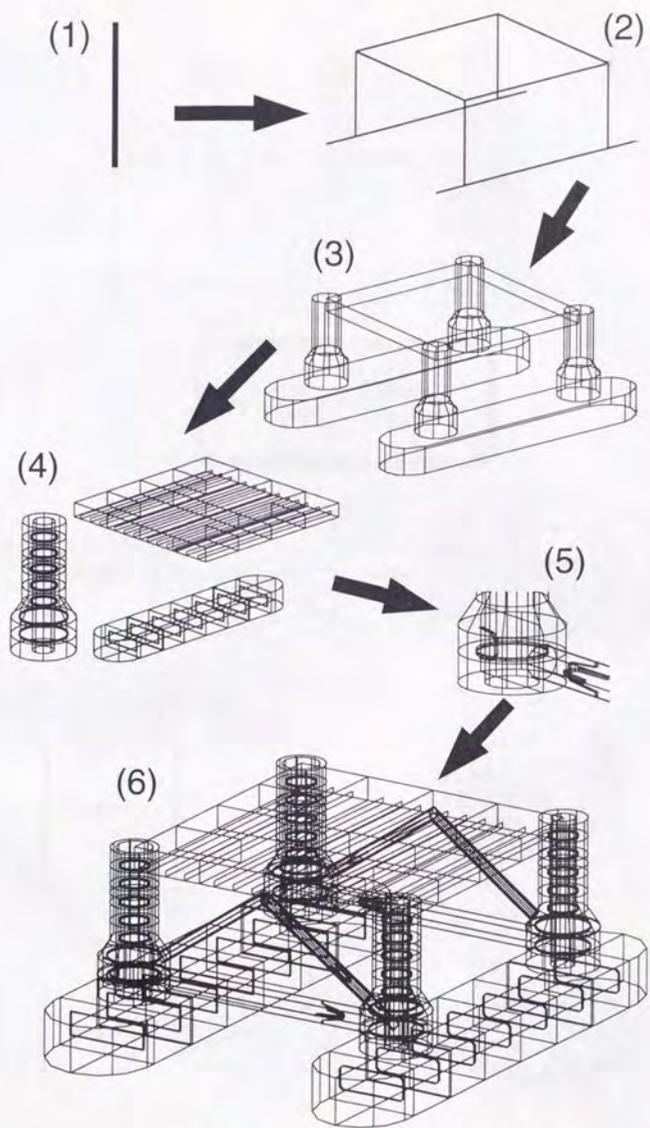


Fig.B-4-1 海洋構造物の設計の流れ

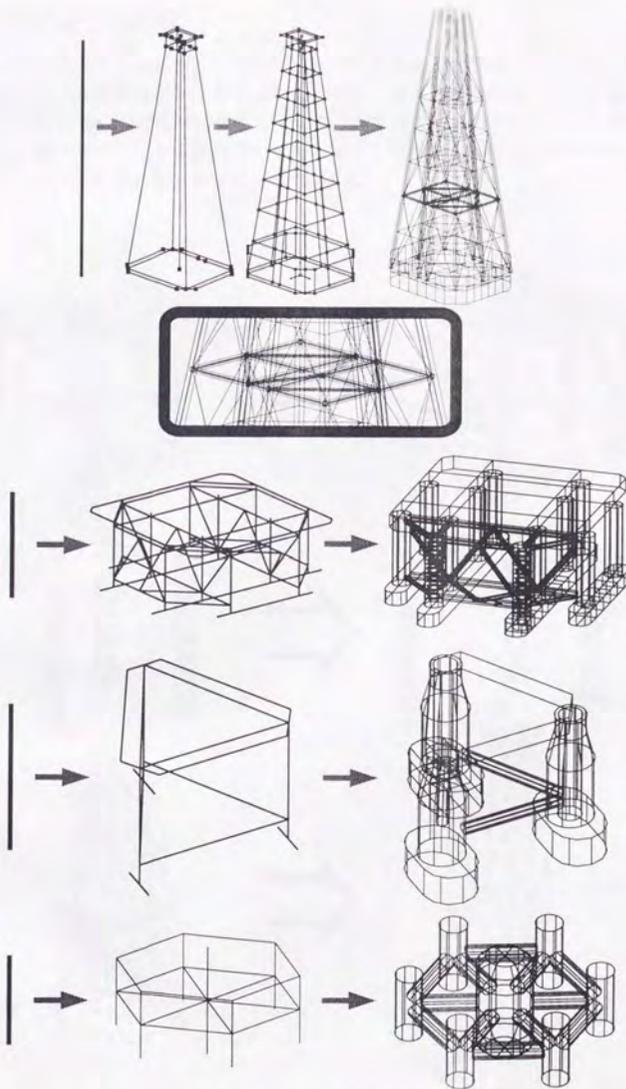


Fig.B-4-2 海洋構造物の設計例

B.5 製品モデルの情報の利用

B.5.1 設計変更への対応

海洋構造物の製品モデルは設計変更の問題に対応することが考慮されている。海洋構造物の様々なモデルは階層構造として記述される。また、様々なモデルを生成する際に設計者によって入力される寸法[2]が設計情報として管理されている。このため何らかの変更が要求された際には、記述されている設計情報を利用して、上流のモデルから下流のモデルへと変更を伝えることによって設計変更が行われる(Fig.B-5-1)。

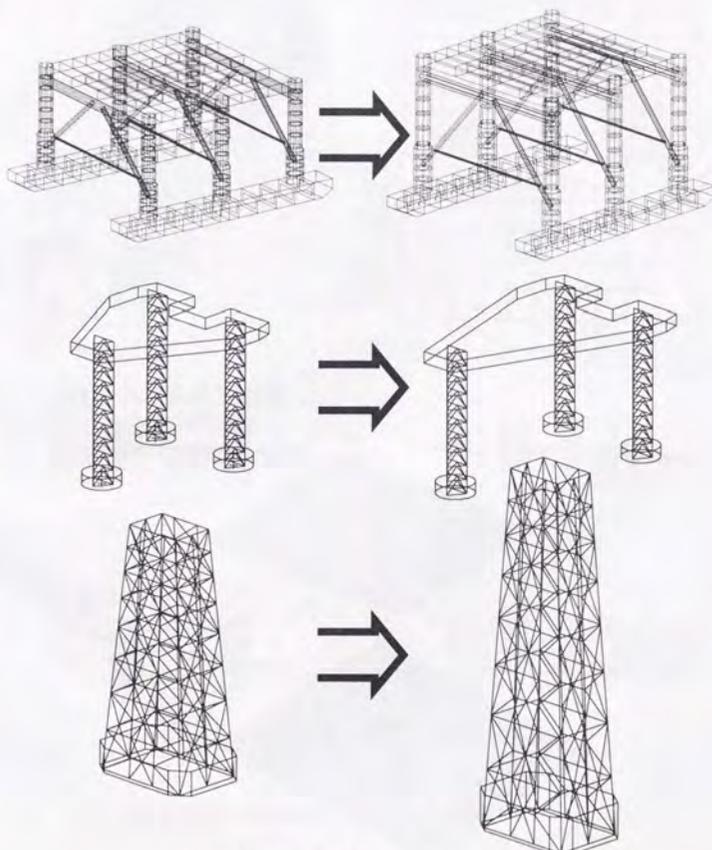


Fig.B-5-1 海洋構造物の設計変更例

B.5.2 安定性に関する情報の獲得

海洋構造物の製品モデルには、製品情報として部屋・部材などが記述される。これらの製品情報を利用すれば、海洋構造物の安定性に関する情報の獲得が期待できる。

そこで静水中での載貨時の海洋構造物の姿勢・安定性に関する情報を算出することを試みた。その結果をFig.B-5-2に示す。この安定性に関する情報を算出する機能は、船の製品モデルであるSODASにおいて既に構築されている機能が共通して利用されている[3]。

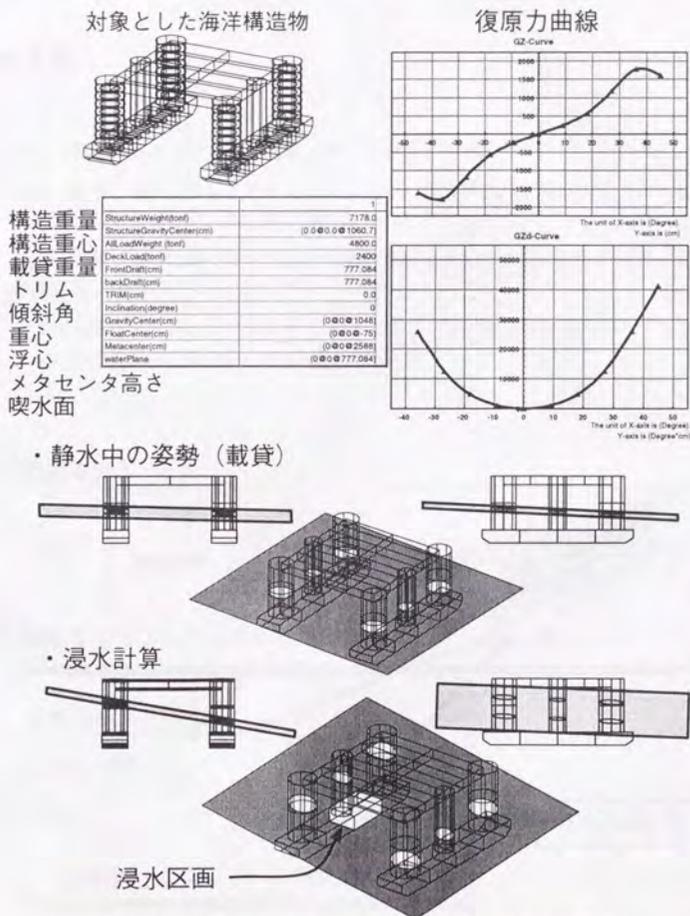


Fig.B-5-2 海洋構造物の安定性に関する情報の獲得

B.5.3 FEMとの統合

海洋構造物の構造解析は、骨組に近似したモデルと、板骨構造物としてのモデルとの異なる2つの解析モデルを用いて行われる。これらのモデルは、海洋構造物の製品モデルの部材情報、接合関係情報、主要構造の情報を利用することによって生成可能と考えられる。そこで海洋構造物の製品モデルはFEMとの統合が実現されている(Fig.B-5-3)。実際に構造解析を行った結果をFig.B-5-4に示す。尚、メッシュの生成は大坪らの研究を参考にしている[4]。

参考文献

- [1] 元網・熊倉：海洋工学の基礎知識、成山堂書店、1990
- [2] 鈴木・木村・佐田：プロダクトモデルに基づく幾何学的拘束関係の記述と寸法処理への応用、精密工学会誌、第52巻、第6号、1986
- [3] 澤田・伊藤：造船CIMモデルを用いた基本設計支援システムに関する研究、東京大学卒業論文、1993
- [4] 大坪・久保田・川村：オブジェクト指向に基づく船体構造解析用FEMモデリングシステム、日本造船学会論文集 第170号、1991

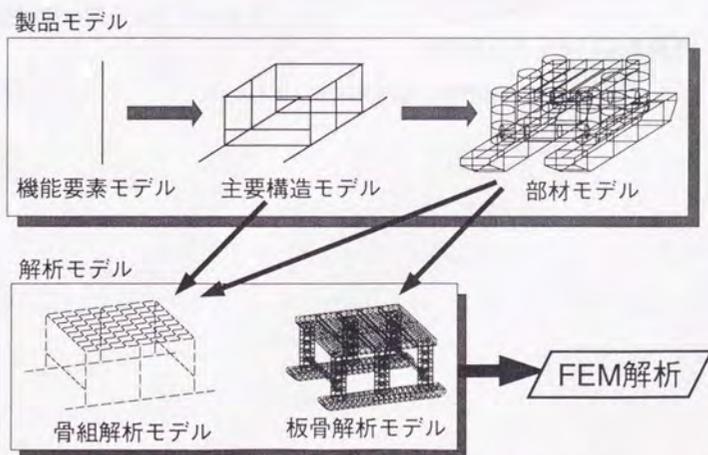


Fig.B-5-3 海洋構造物の製品モデルとFEMとの統合

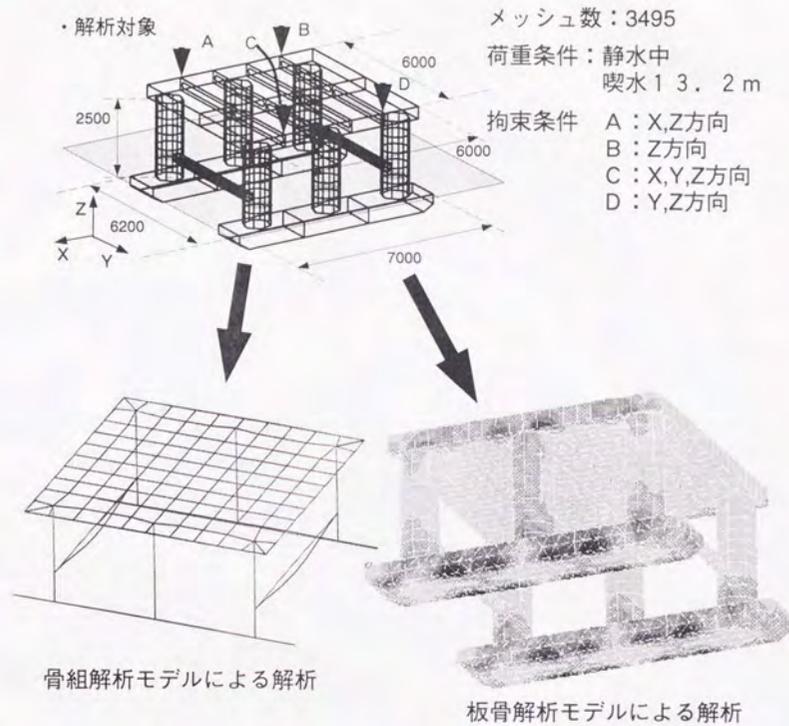
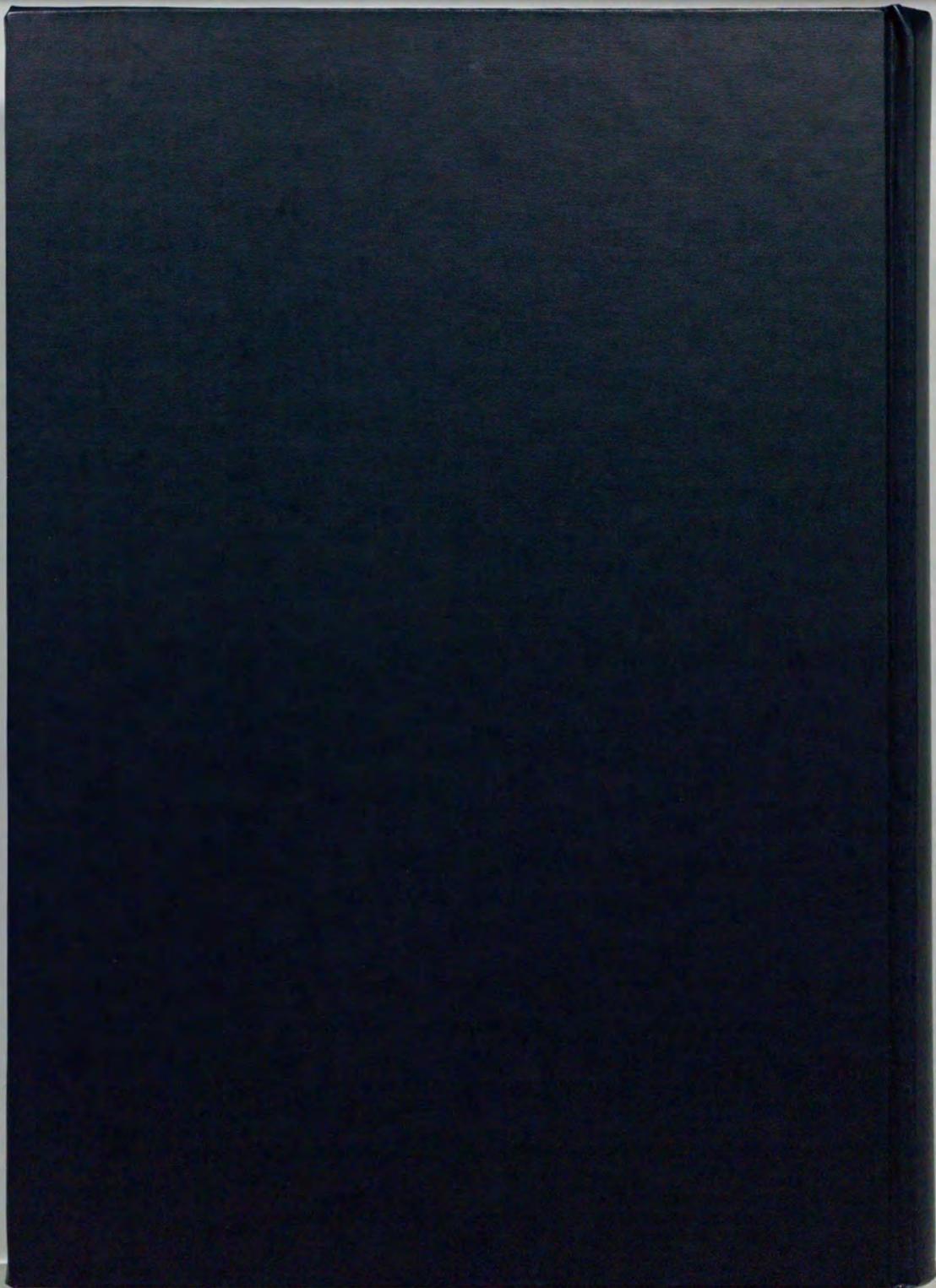


Fig.B-5-4 海洋構造物の構造解析例



inches
cm 1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

Kodak Color Control Patches

© Kodak, 2007 TM Kodak



Kodak Gray Scale



© Kodak, 2007 TM Kodak

A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19

