

グラフ型図面理解システム構築のためのドメイン分析・モデリング

伊藤 剛†‡ 橋本正明‡ 鵜林尚靖‡ 片峯恵一‡

類似システムを効率よく開発するための概念、指針としてドメイン分析・モデリング(DAM)がある。本稿では、DAM の基本的な指針として BCVS 分析法を提案し、グラフ型図面理解ドメインの分析・モデリングに適用することで評価する。今回の DAM の成果として、理解対象図面の知識記述の枠組みである FEOS(Framework for Extracting Object Structure)を構築した。FEOS を用いて 3 種類の図面に対し知識記述を行い、図面情報抽出実験を行うことで FEOS の有用性を評価した。

Domain Analysis and Modeling for Construction of Graph Type Drawing Interpretation System

Takeshi ITO † ‡, Masaaki HASHIMOTO ‡, Naoyasu UBAYASHI ‡ and Keiichi KATAMINE ‡

Domain Analysis and Modeling(DAM) is a concept to develop efficiently a similar system from the system family. We propose newly BCVS(Boundary, Commonality, Variation and Specificity) analysis method as a basic indicator of DAM and evaluate it by applying it for DAM of a graph type drawing interpretation domain. We constructed a knowledge description framework FEOS(Framework for Extracting Object Structure) and evaluated it by describing knowledge for extracting information data from three types of drawings.

1. はじめに

筆者らは、類似システムを効率よく開発するための概念・指針であるドメイン分析・モデリングに関する研究を行っている。本稿では、ドメイン分析・モデリングを略して DAM(Domain Analysis and Modeling)と呼ぶ。現時点においては、具体的な DAM 手法が確立されてないため、試行錯誤的に DAM を行うのが現状である。そこで筆者らは、過去の自身の DAM 研究やさまざまな DAM 関連論文を整理、調査し、DAM を行う際の基本的な指針として BCVS 分析法の提案、および、検討を行っている。

本稿では、BCVS 分析法の適用事例としてグラフ型図面理解ドメインを対象としている。以下、図面理解の必要性について述べる。建築業界では、柱、壁といった建築物の構成要素である部材をオブジェクトして取り扱い、それらを用いて建築物を設計できるオブジェクト指向 CAD の開発や、部材のデータ共有定義 IFC(Industry Foundation Classes)の規格化などで設計業務間の効率のよい情報交換を行おうと努力が続けられている。しかし、現状では、図面ファイルを介して情報交換することが一般的である。図面ファイルは基本的に図形データの集合にすぎないため、建築物の情報を得るためには、人間の目に頼らざるを得ない。したがって、建築物データの交換に多大なコストを要する。筆者らはこの問題に着目し、図面を単に電子化しただけの建築 CAD 図面データから、部材をオブジェクト、部材間の接続関係をリンクとするオブジェクト構造を抽出することにより、図面データをオブジェクト指向 CAD に取り込むことのできる支援環境の構築を行った[1]。本研究では、「図面理解」を「図面データから図面の表現する意味をオブジェクト構造として抽出すること」と定義する。図面理解を支援するシステムを開発することができれば、図面データを業務の主な情報交換の媒体としている業種においては、建築設計業務と同様に、その高度な活用が期待できる。

図面理解システムは推論処理を伴うため、一般のシステムより開発の難易度が高い。多様な図面を対象にした図面理解システムを構築するためには、理解したい図面に関する知識を入力する仕組みを考慮する必要があるため、さらに難易度が高くなる。また、たとえ構築できたとしても知識入力の方法が汎用的にならざるを得ないため、特定の図面を理解させるための知識記述作業自体も困難になってしまう。そこで、本研究ではシステム化の

† 現在、情報通信系メーカーに勤務

‡ 九州工業大学

Kyushu Institute of Technology

対象を図面要素が網の目状の構成されているグラフ型図面の理解に特化することで、それらの問題の解決をはかった。

現在、グラフ型図面理解ドメインについて、新たに重み付き有向グラフ図からのグラフ情報の抽出、UMLのクラス図からのモデル情報の抽出を想定し、前述の研究成果を含めて、分析・モデリングを行っており、本稿では現段階で得られているドメインモデルについて報告する[2]。2章でドメイン分析・モデリングの概要とBCVS分析法について述べる。3章でグラフ型図面データがアプリケーションに入力され、活用されるまでの環境を説明する。4章でグラフ型図面理解ドメインの分析・モデリングを行い、その成果に基づき構築したドメインモデルであるオブジェクト構造抽出のための知識記述フレームワーク FEOS(Framework for Extracting Object Structure)について述べる。5章で図面から抽出したオブジェクト構造を適用したいアプリケーションのフォーマットに変換するデータモデル変換の方法について述べる。6章で、研究成果の考察を行う。

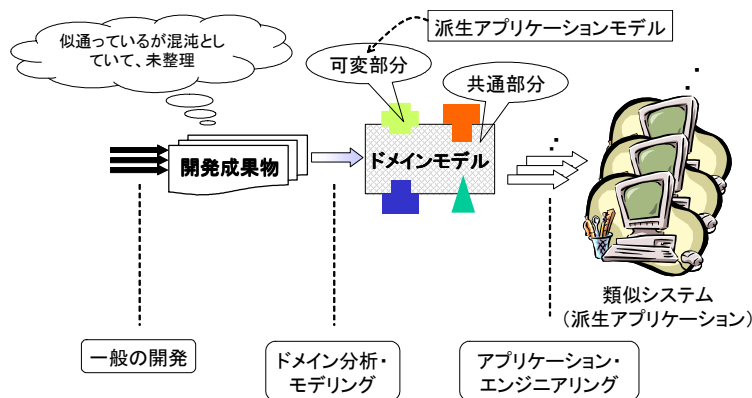


図 1 ドメイン指向システム開発

2. ドメイン分析・モデリング

2.1 概要

類似システム開発の体系的な再利用を推進する概念、指針がドメイン分析・モデリング[3]である。筆者らは、ドメイン分析・モデリングを「同じ問題領域（ドメイン）に属する類似システムを容易に開発できるようにするために、類似システム群の仕様書、ソースコード、あるいはシステムそのものに対して、共通な性質と可変な性質が何であるかをさらに分析し、共通な性質を内包するように抽象化し、それをドメインに固有な枠組みとして表現するプロセスである。」と定義する。そして、その枠組みのことをドメインモデルと定義する。

図 1 にドメインモデルを用いた開発の流れを示す。開発成果物は、システム開発の過程で得られる仕様書群やソースコード群などである。それらについてドメイン分析・モデリングを行い、ドメインモデルを得る。ドメインモデルにおける可変部分を、今回の開発要件に沿ってカスタマイズし、類似システムを構築する。本稿では、このドメインモデルより派生した類似システムを構築するプロセスをアプリケーション・エンジニアリングと呼び、このとき作成するモデルを派生アプリケーションモデルと呼ぶ。

2.2 BCVS 分析法

ドメイン分析の方法として、SCV(Scope, Commonality, Variation)分析法[4]などがあるが、前項で述べたドメインモデルの固有性やドメインの境界などに着目した分析について明示的に説明している文献は見あたらない。そこで筆者らは、DAMの基本的な指針として、BCVS(Boundary, Commonality, Variation, Specificity)分析法を提案し、その洗練をはかっている。以下、B,C,V,Sのそれぞれの分析の観点を説明する。

● 境界 (Boundary)分析

境界分析では、ドメインの境界や範囲とドメインモデルの性能の限界について分析を行う。この作業を怠るとドメイン分析プロセスが収束しなかったり、有用なドメインモデルを構築できないだけでなく、そのドメインからずれたシステムをそのドメインモデルを利用して開発しようとして、開発が頓挫してしまうことにもなりかねない。また、境界分析ではドメインの中にサブドメインが存在する場合もあるので、それにも留意する。

- 共通性(Commonality)分析

共通性分析では、ドメインの過去の開発成果物における共通の構造や振る舞いを分析する。この作業では、共通部分の構造が振る舞いが派生アプリケーションモデルの作成者からなるべく隠蔽されるように注意する必要がある。

- 可変性(Variation)分析

可変性分析では、ドメインの過去の開発成果物や、可変な構造や振る舞いを分析する。また、将来の類似システムの開発で想定される可変性も考慮する。

- 固有性(Specificity)分析

ドメインの共通性や可変性を十分に分析し、少ない工程で派生アプリケーションモデルの作成ができたとしても、直感的、あるいは、本質的に行えないならば、そのモデルの他人による確認や変更が困難になる。固有性分析では、そのような状況にならないように、ドメインモデルの固有性や特有性について分析する。ドメインモデルとしてオブジェクト指向型のドメイン特化言語を構築する場合の例で具体例を挙げると、この作業はドメイン固有の用語を予約語にしたり、クラス間の関連の種類や多重度を限定することを意味する。

ドメインの境界は、共通性や可変性も検討しなければ限定できない。共通性と可変性は密な関係にあるため、お互いを意識しなければならない。固有性は、共通性や可変性を検討した結果、生じる場合もある。したがって、BCVS 分析に手順はなく、それぞれの分析を行きつ戻り、そして、B,C,V,S のバランスを考慮しながら行う。

3. グラフ型図面データ活用支援環境

建築 CAD 図面、クラス図を例に、図面データが有効活用されるまでの流れを図 2 に示す。適用先のアプリケーションには、それぞれ入出力フォーマットがあり、その形式もまちまちであるため、図面データ活用のためには次のような 2 段階のフェーズが必要である。

オブジェクト構造抽出フェーズ：ベクトル型図面データから、汎用的なオブジェクト構造を抽出し、これを中間フォーマットとして出力する。この抽出過程がグラフ型図面理解ドメインである。

データモデル変換フェーズ：その中間フォーマットを適用先のアプリケーションの入出力フォーマットへ変換する。

この中間フォーマットのことを本稿では FEOS フォーマット、そのデータファイルを FEOS オブジェクトファイルと呼ぶ。FEOS については後述する。

この FEOS フォーマットは、建築設計の場合は IFC、システム開発の場合は XMI(XML Metadata Interchange) に変換され、それぞれ、オブジェクト指向 CAD,UML CASE ツールに取り込まれ、活用される。

利用者は、この支援環境を利用することで、手作業で、図面から必要な情報を得て、別のアプリケーションに入力するという煩雑な作業から解放される。

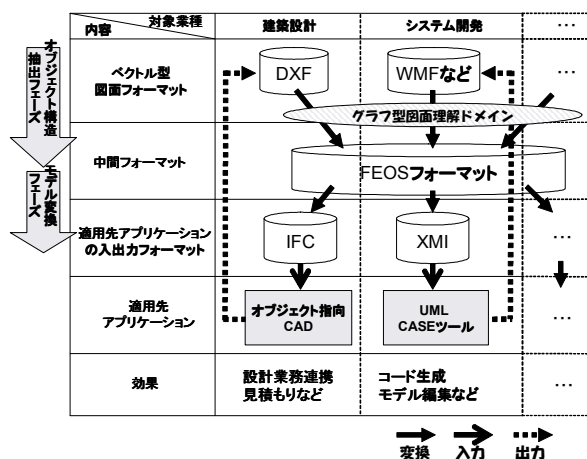


図 2 グラフ型図面データ活用支援環境

4. グラフ型図面理解ドメインの分析・モデリング

図面からオブジェクト構造を抽出するためには、対象図面の製図規則とその図面で表現される世界、つまり、建築図面の場合は部材の大きさの制約や部材同士の接続関係といった建築物モデル、UML のクラス図の場合は UML メタモデルが必要になる。したがって、それらに関する知識を記述するための枠組みをドメインモデルとして獲得することができれば、さまざまな業種の図面に楽に対応できる。しかし、多種の図面を対象にして枠組みを構築することは困難であり、また、たとえ構築できたとしても、知識記述作業が煩雑になったり、直感的でなくなる恐れがあるため、グラフ型図面理解にドメインを限定して分析・モデリングを行った。以下、BCVS 分析法に基づく DAM を説明する。

4.1 グラフ型図面理解ドメインのBCVS分析

4.1.1 知識記述の枠組み構築のための共通性分析

(1) 知識記述の枠組み自身

文献[1]の研究で、我々は、建築 CAD 図面から部材オブジェクト構造を抽出するための知識記述の枠組みを、以下のように整理した。これらの枠組みに従って、抽出知識をオブジェクトモデルと属性の制約で非手続き的に記述する。

- 部材知識
個々の部材の大きさに関する知識。部材の属性の制約で表される。
- 部材間接続知識
部材同士の接続に伴う属性間の制約を表す知識。
- 描画図形知識
個々の部材が CAD 図面上でどのように描かれているかを表す知識。

クラス図からオブジェクト構造を抽出することを考えた場合、クラスや継承の概念そのものは部材知識で記述でき、クラスと継承の関係やクラスと属性の関係などが部材間接続知識で記述できる。継承が三角形と線分で図面上に表現されることは描画図形知識で記述できる。したがって、上記の枠組みの基本的な構造は他業種の図面においても通用することが推測される。これらの枠組みをそれぞれ、エンティティ知識、関連知識、ビュー知識と命名しなおした。

(2) 推論手順

4.1.1(1)の枠組みを用いて記述された知識からオブジェクト構造を抽出するための推論手順を考案した。図3は、記述知識の建築物オブジェクトモデルと部材オブジェクト構造の抽出手順を示している。このように、推論手順は、それらの知識を参照しながら伝播的に隣接オブジェクトを抽出していくという極めて単純なものである。配管図、回路図、クラス図などを見ると、図面要素が網の目のように連結され、各図面要素は、近傍の図面要素が確定すればそれ自身も確定されることが容易に確認できるため、この推論手順はさまざまな図面に適用可能である。

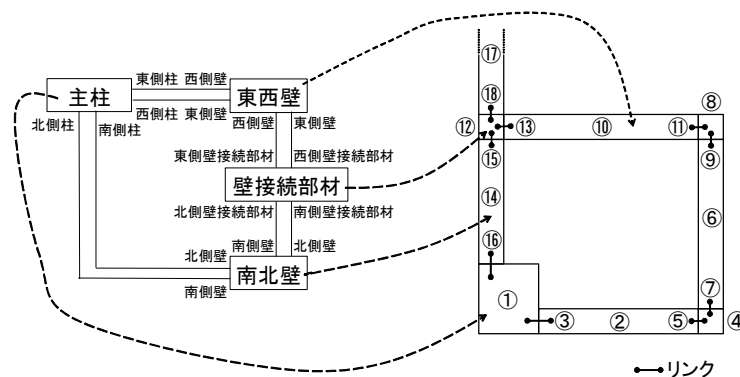


図3 推論手順

(3) 図形ライブラリ

ベクトル型図面データを扱うため、線分、円、文字（列）などのプリミティブな図形についてクラスライブラリを用意する必要がある。

4.1.2 知識記述の枠組み構築のための可変性分析

(1) 抽出知識オブジェクトモデルごとに入れ替わる場合

図面が作成される業種ごとに知識記述の枠組みに沿って抽出知識オブジェクトモデルをゼロから記述する。

(2) 描かれ方のみ異なる場合

同業種の図面で構成要素が同じでも描かれ方のみ異なる場合がある。例えば、建築図面におけるドアについて、

ドアそのものの属性である幅や長さなどは同じでも、設計事務所などにより描かれ方が異なる。このような場合、ビュー知識のみ入れ替わる。クラス図の場合でも UML 記法と OMT 記法で描かれ方のみが異なる場合があり、同様のことが言える。

4.1.3 知識記述の枠組み構築のための固有性分析

(1) 宣言的な知識記述

知識記述者の立場になると、図面の構成要素の構造や要素間の関係といった静的な性質を意識することは、比較的容易である。しかし、抽出の手順を意識することは、直感的ではない。また、図 3 で示した推論処理は、①の柱からでなく、例えば、②の壁から始めても抽出可能である。したがって、プロダクションシステムのようなルールベースで知識を記述することを仮定すると、同種図面からオブジェクト構造を抽出するための知識記述に多様な解が生じるため、記述された知識の他人による確認や変更が困難になる。以上を踏まえると図面理解のための知識記述は宣言的であることがドメインモデルの固有性を満たすと言える。

(2) エンティティ知識とビュー知識の区別

建築図面の壁の例で説明すると、壁の厚さの制約と壁が 2 本の平行な線分で描かれるという線分同士の配置の制約は、本質的に異なる。したがって、抽出対象そのものの知識であるエンティティ知識と描かれ方に関する知識であるビュー知識を分離することは、ドメインモデルの固有性を満たすと言える。

4.1.4 知識記述の枠組み構築のための境界分析

4.1.1(2)で説明した推論手順が有効な範囲は、図面要素がグラフのように構成され、かつ、図面上の広範囲を見なければ判断できないオブジェクトが存在しない図面の理解に限定される。したがって、その範囲がドメインの境界である。

建築図面においては、図面要素の構成に変化がなく、ドアや窓のように要素の描かれ方がよく変化するため、建築図面理解をサブドメインとし、描かれ方のみを直感的に変更できるような知識記述の枠組みも構築しておくことが望ましい。

4.2 知識記述フレームワーク FEOS

4.2.1 フレームワークの構成

前節の分析結果に基づき、オブジェクト構造抽出のための Prolog ベース知識記述フレームワーク FEOS(Framework for Extracting Object Structure)を構築した。その構造を図 4 にクラス図で示す。UML のメタモデルと比較するとエンティティ知識、ビュー知識はクラス、関連知識は関連クラスに相当する。図面データにおける線分や文字オブジェクトのクラス定義はビュー知識のインスタンスで対応させ、定義済みクラスの形で利用することにした。

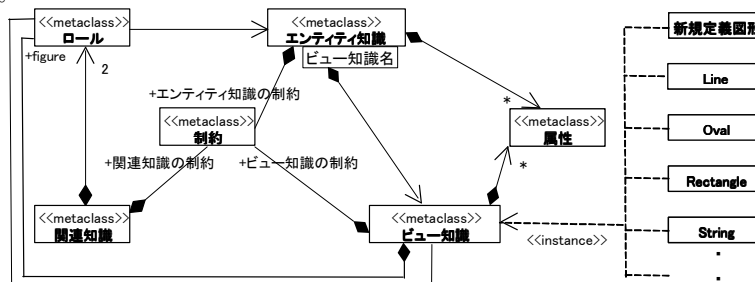


図 4 FEOS メタモデル

4.2.2 フレームワークの仕様と記述例

建築図面、クラス図からオブジェクト構造を抽出するための知識記述例を通してフレームワークの仕様を説明する。クラス図と重み付き有向グラフ図に関しては図 5 にそれらの FEOS モデルを示す。(a)については、紙面の都合上、関連知識を記載していない。なお、推論手順の説明は論文[2]に譲る。

(1) エンティティ知識

エンティティ知識では、属性の定義、属性の取りうる値の範囲やビュー知識の属性との関係を記述する。記述された知識は、抽出されるオブジェクトのクラス定義にあたる。属性の型については、数値型として「num」、文字列として「string」を指定する。ビュー知識の属性値を参照するには、以下の述語を用いる。

- attribute_value_of_view(<属性名>,<変数>).

図 6(a)に建築図面における東西壁のエンティティ知識記述例を示す。この記述における制約について説明すると、例えば、壁の厚さが 200mm~600mm の値を取りうることを示している。

(2) 関連知識

関連知識では、エンティティ知識間の関連に伴う属性間の制約をエンティティ知識名、ロール名を用いることにより記述する。記述された知識は、オブジェクト構造におけるリンクのクラス定義にあたる。関連の多重度については、1 と多についてそれぞれ「1」、「n」を指定する。ロール先のエンティティ知識の属性値を参照するには、以下の述語を用いる。

- attribute_value_of_role(<ロール名>,<属性名>,<変数>).

また、クラス図のような論理型の図面は、エンティティ知識が幾何的な属性を持たない場合がある。そのような場合、エンティティ知識のビュー知識の属性値を参照する以下の述語を用いる。

- attribute_value_of_role_view(<ロール名>,<属性名>,<変数>).

図 6(b)にクラス図における親クラスと継承の関連知識を示す。制約では、継承を表現する矢印の先端がクラスを表現する箱の淵に近いことを記述してある。

(3) ビュー知識

ビュー知識では、エンティティが図面上で線分や文字列によってどのように描かれているかを記述する。属性や制約に関する内容は基本的にエンティティ知識と同様である。「element_figures」構文で、自分自身を構成する図形を「figure」として挙げる。つまり、ビュー知識自身と「figure」で指定される図形は集約関係にある。<変数>の名前は、部分クラスのロール名の役割を果たす。ここでの図形は、図 4 に示すようなすでに定義されているプリミティブ型の図形でも記述者によって作成したビュー知識でも構わない。図 6(c)にクラス図におけるクラスのビュー知識の記述例を示す。

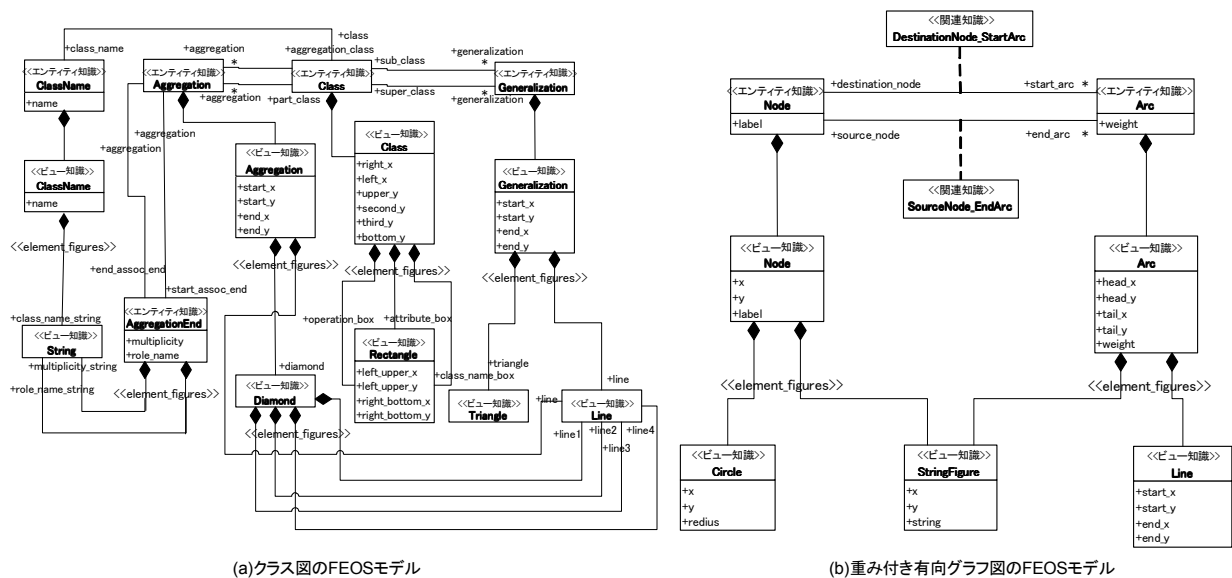


図 5 FEOS モデルの例

```

entity_knowledge(ew_wall,    ...<エンティティ知識名>
attributes([
  attr(num, east_x, EastX),  ...<型>, <属性名>, <変数>
  attr(num, west_x, WestX),
  attr(num, north_y, NorthY),
  attr(num, south_y, SouthY)
]),
constraints(                ...制約
(
  WestX < EastX,
  WidthY is NorthY - SouthY,
  200 =< WidthY,
  WidthY =< 600,

  attribute_value_of_view(east_x, ViewEastX),
  attribute_value_of_view(west_x, ViewWestX),
  attribute_value_of_view(north_y, ViewNorthY),
  attribute_value_of_view(south_y, ViewSouthY),

  EastX = ViewEastX,
  WestX = ViewWestX,
  NorthY = ViewNorthY,
  SouthY = ViewSouthY
)
)
).

(a)東西壁のエンティティ知識記述例

relationship_knowledge(
class, super_class, 1,    ...<エンティティ知識名>, <ロール名>, <多重度>
generalization, generalization, n,
constraints(
(
  attribute_value_of_role_view(super_class, right_x, SuperClassRightX),
  attribute_value_of_role_view(super_class, left_x, SuperClassLeftX),
  attribute_value_of_role_view(super_class, upper_y, SuperClassUpperY),
  attribute_value_of_role_view(super_class, bottom_y, SuperClassBottomY),

  attribute_value_of_role_view(generalization, end_x, GeneralizationEndX),
  attribute_value_of_role_view(generalization, end_y, GeneralizationEndY)

  (near(SuperClassBottomY, BeginGenerationBeginY),
  SuperClassLeftX =< BeginGenerationBeginX),
  BeginGenerationBeginX =< SuperClassRightX);
  (
  ...
);
...
)
)
).

(b)クラスと継承間の関係知識記述例

view_knowledge(class,    ...<ビュー知識名>,
attributes([
  attr(num, right_x, RightX),
  attr(num, left_x, LeftX),
  attr(num, upper_y, UpperY),
  attr(num, second_y, SecondY),
  attr(num, third_y, ThirdY),
  attr(num, bottom_y, BottomY)
]),
element_figures(
(
  figure(rectangle, ClassNameBox), ...<ビュー知識名>, <変数(=ビュー知識のロール名)>
  figure(rectangle, AttributeBox),
  figure(rectangle, OperationBox),
)
),
constraints(
(
  attribute_value_of_view(ClassNameBox, left_upper_y, ClassNameBoxUpperY),
  attribute_value_of_view(ClassNameBox, right_bottom_y, ClassNameBoxBottomY),
  attribute_value_of_view(ClassNameBox, right_bottom_x, ClassNameBoxRightX),

  attribute_value_of_view(AttributeBox, left_upper_y, AttributesBoxUpperY),
  attribute_value_of_view(AttributeBox, right_bottom_y, AttributesBoxBottomY),
  attribute_value_of_view(AttributeBox, right_bottom_x, AttributesBoxRightX),

  attribute_value_of_view(OperationBox, left_upper_y, OperationsBoxUpperY),
  attribute_value_of_view(OperationBox, right_bottom_y, OperationsBoxBottomY),
  attribute_value_of_view(OperationBox, right_bottom_x, OperationsBoxRightX),
  attribute_value_of_view(OperationBox, left_upper_x, OperationsBoxLeftX),

  ClassNameBoxBottomY = AttributesBoxUpperY,
  AttributesBoxBottomY = OperationsBoxUpperY,

  ClassNameBoxRightX = AttributesBoxRightX,
  AttributesBoxRightX = OperationsBoxRightX,

  UpperY = ClassNameBoxUpperY,
  SecondY = AttributesBoxUpperY,
  ThirdY = OperationsBoxUpperY,
  BottomY = OperationsBoxBottomY,
  RightX = ClassNameBoxRightX,
  LeftX = OperationsBoxLeftX
)
)
).

(c)クラスのビュー知識記述例

```

図 6 知識記述例

5. データモデル変換

FEOS モデルにおけるエンティティ知識と関連知識は FEOS オブジェクトのスキーマにあたり、ER モデルと同等であるため、SQL のスキーマに変換可能である。SQL に変換できれば表の結合が可能であるため、オブジェクトの検索のための制御文を記述する必要がなくなり、データ変換プログラムの作成が比較的容易になる。そこで FEOS モデルのエンティティ知識と関連知識を SQL の CREATE TABLE 文に、FEOS オブジェクトを INSERT 文に変換するデータモデル変換プログラム FEOS2SQL を作成した。

6. 抽出実験と適用実験

6.1 クラス図

クラス 3 つ、継承関係 1 つ、集約関係 1 つを含む簡単なクラス図を UNIX 系 OS でよく使われる描画ツール Tgif[5]で描き、その保存ファイル(.obj ファイル)からモデル情報を抽出する実験を行った。その FEOS モデルを図 5(a)に示す。抽出の結果、図 6 の FEOS オブジェクトファイルを得た。

次に、FEOS フォーマットを XMI に変換するプログラムを前章で説明した FEOS2SQL, Java を用いて作成し、モデル変換実験を行った。変換された XMI ファイルを UML CASE ツール IIOSS[6]にとり込むことで実証実験を行った。図 7 その表示結果を示す。

6.2 重み付きグラフ図

Tgif で描かれたノードが7つで構成される重み付きグラフ図からグラフ情報を抽出する実験を行った。その FEOS モデルを図 5(b)に示す。抽出した FEOS オブジェクトをネットワークにおける最大流と最小カットを求めることができるツール[7]に取り込み、実証実験を行った。図 8 にその表示結果を示す。

```
%オブジェクト
object(65, class, []).
object(66, class, []).
object(67, generalization, []).
object(68, class_name, [attr(name, 'Figure')]).
object(69, class_name, [attr(name, 'Line')]).
.....

%リンク
object_link(65, class, super_class, 67, generalization, generalization).
object_link(67, generalization, generalization, 66, class, sub_class).
object_link(68, class_name, class_name, 65, class, class).
object_link(69, class_name, class_name, 69, class, class).
.....
```

図 6 FEOS オブジェクト

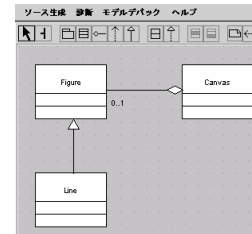


図 7 XMI 取り込み結果

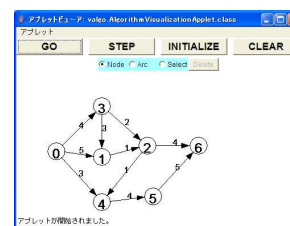


図 8 最大流、最小カット計算ツールへの取り込み結果

7. 考察

7.1 FEOS の評価

知識記述者は、図面の意味を表現するオブジェクト構造を要求しているわけだが、そのモデルをエンティティ知識と関連知識で素直に記述でき、それらは各エンティティがどのように描かれているかを表すビュー知識と明確に分離される。そしてそれぞれの属性の制約を付随させる形で、宣言的に、かつ、抽出手順を一切意識することなく記述できる。したがって、非常に直感的に知識記述でき、可読性も高く、記述量も少ない。また、FEOS メタモデルは UML メタモデルをベースに構築したため、図面の表すモデルをオブジェクト指向分析のような感覚で記述できる。以上を踏まえれば、記述方法を少ない教育期間で習得できることが容易に想像できる。

7.2 BCVS 分析法の評価

DAM は、それぞれのドメインに固有なドメインモデルを得る作業であるため、そのプロセスに汎用的な方法論を与えることは、根本的に難しい。しかし、指針があるかないかの違いは大きい。本稿で提案した BCVS 分析法は、前節で述べたような有用なドメインモデルを得ることができたため、指針として大いに有用であった。

8. 今後の課題

FEOS に関しては、さまざまな業種の図面に対し、知識記述、オブジェクト構造抽出実験を行い、さらなる洗練を図る。BCVS 分析法に関しては、過去の DAM 研究の文献をさらに調査し、それぞれのドメインの共通性、可変性、ドメインモデルの固有性、および、ドメインの境界について、分析することで洗練を図る。

参考文献

- [1]伊藤剛, 廣田豊彦, 橋本正明, “建築 CAD 図面データからのプロダクトモデル生成とその活用”, 電気学会論文誌 D(産業応用部門誌), vol.122-D, no.8, pp.862-870, Aug. 2002.
- [2]伊藤剛, 廣田豊彦, 橋本正明, “図面理解ドメインの分析・モデリング”, 信学技報.KBSE, vol.103, no.217, pp.23-28, July.2003.
- [3]廣田豊彦, 伊藤潔, 熊谷敏, 吉田裕之, “特集 ドメイン分析とドメインモデリング”, 情報処理, vol.40, no.12, pp1171-1191, Dec.1999.
- [4]James O. Coplien(著), マルチパラダイムデザイン, ピアソンエデュケーション, Dec.2001.
- [5]<http://bourbon.usc.edu:8001/tgif/index.html>
- [6]鈴木重徳, 佐野元之, 倉骨彰, 垣花一成, IIOSS UML に基づく設計/開発環境のすべて, アスキー, July 2001.
- [7]品野 勇治, Java 言語によるアルゴリズムの視覚化, 技術評論社, Java Press vol.12,13,14.