

CADにおけるリレーショナルモデルの一例

川井敏弘、宇野栄
(日本 IBM Scientific Center)

1 はじめに

設計作業は、設計者の頭に浮かんだイメージを、規格データ、設計資料、あるいは注文者による拘束条件などを考慮しながら具体化していく過程といえよう。設計対象物はラフなスケッチからしだいに精密な図面へと変わってゆく。この過程においてはたくさんの代替案が考案され、最善の案を選ぶためにいろいろな角度から工学的ならびに審美的な解析が行われる。このような設計業務にコンピューターを用い、設計者を援助するCADシステムとしてIDAS (Integrated Designer's Activity Support) [1] が日本 IBM Scientific Centerで開発された。IDASは汎用的に多くの設計分野で使用可能であり、対話的に設計を行うよう構成されている(付録参照)。コンピューターを用いて設計を行うとすると、当然のことながら設計対象物をコンピューターの内部にコンピューターの扱える形態で貯える必要がある。このデータはいわゆるデータベースの中に貯えられるべきものであるが、IDASでは設計者毎にデータベースから必要なデータを取り出して、都合の良い構造に再構成した作業領域(work area)をモデルと呼ぶ。本稿ではIDASのモデルのデータ構造について報告する。

モデルは次のような要求をみたす必要がある。

- 1) モデルは組織的かつ簡潔であること。
- 2) 設計者が思考してきた過程も記録されること。
- 3) 任意の設計時点にフィードバックでき、またそこから新しい設計案を生み出すことができること。
- 4) 設計者、プログラマーに対して、モデルは容易に理解でき、またモデルの加工が容易にかつ迅速にできること。
- 5) モデルは容易にデータベースと接続でき、他の設計者達に正確かつ迅速にその情報が伝達できること。

このモデルを表現するデータ構造として、リング構造を用いるのか、連想3つ組方式を用いるのか、リレーショナルモデルを用いるかは、上の要求にしたがって検討しなければならない[2, 3]。筆者はさらにリング構造、連想3つ組方式、階層構造、リレーショナルモデルの4つを、必要記憶容量、操作の容易さ、操作時間などの面から比較した結果、リレーションの基数(cardinality)がある程度の範囲内であれば、リレーショナルモデルが総合的に一番適当だと結論した。このためIDASではリレーショナルモデルを用いることにし、新しいリレーションのハンドラーを開発した。

2 モデルの表現

IDASが多くの設計分野に汎用的に使用できるためには、モデルは多くの設計対象物を汎用的に表現しなければならない。現実の設計対象物は次の3つのカテゴリに分けてとり扱われる[4]。

- 1) リンクモデル
- 2) 属性モデル
- 3) 代替案モデル

これらのモデルはそれぞれリレーションで表現される。

2.1 リンクモデル

リンクモデルとはエンティティ間の関係を表現するものである。エンティティ間の関係は種々あるが、現在のIDASでは、ネットワークモデルと平面形状モデルとを支援している。

A) ネットワークモデル

ネットワークモデルはライン(アーク)とポイント(ノード)間の関係を有向線点図として扱ったものである。もしアークの方向性が不要ならば、それは無視される。Fig. 1 にネットワークモデルの実例が示されている。このモデルは

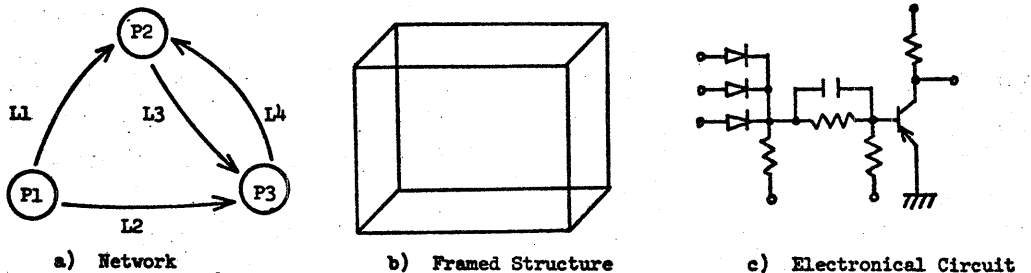


Fig. 1. Network Model

L-PP と名づけられた3次元のリレーションによって表現される。リレーションを構成する各定義域 (domain) は、ライン名、始点、終点である。Fig. 1-a に対応した L-PP リレーションが Table 1 に示されている。

Table 1. The L-PP Relation

Line Name	Start Point	End Point
L1	P1	P2
L2	P1	P3
L3	P2	P3
L4	P3	P2

B) 平面形状モデル

平面形状モデルは、エリアとそれを囲むライン群間の関係を扱ったものである。Fig. 2 に平面形状モデルの実例が示されている。ライン群はネットワークモデルと同

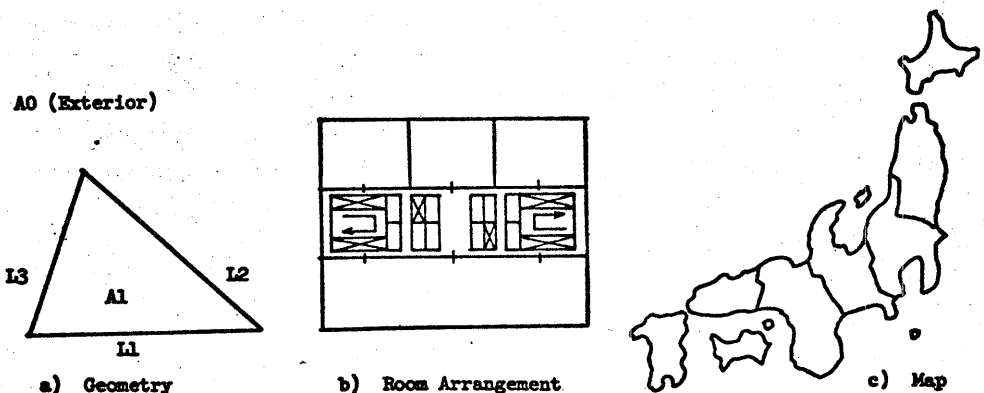


Fig. 2. Geometric Model

じ方法でポイント群と結合している。平面形状モデルはこの L-PP リレーションと A-LL と名づけられた3次元のリレーションとによって表現される。A-LL リレーションの各定義域は、エリア名、基本ライン、隣接ラインである。エリアは

3本あるいはそれ以上のラインから構成されるので、エリアに関する情報を十分に表現するためには、一個のタプル (tuple) では不可能である。基本となるラインとその隣接ライン (ラインによって囲まれたエリアが内部であるならば反時計方向、外部エリアであるならば時計方向) を対にして、周囲を囲っているラインの本数に等しいタプルで表現する。このようにした理由は周囲を囲っているラインの数が不定だからである。隣接ラインは一見不要のように思われるが、エリアに関する全貌を迅速に把握するためには有効である。Table 2は Fig. 2-aを表現した A-LLリレーションである。

Table 2. The A-LL Relation

Area Name	Key Line	Neighboring Line
A0	L1	L3
A0	L3	L2
A0	L2	L1
A1	L1	L2
A1	L2	L3
A1	L3	L1

2.2 属性モデル

エンティティの属性値は属性モデルに蓄積される。リンクモデルにおいては、エンティティは白紙のフレームワークとしてみなされる。属性モデルがエンティティに対して白色から有色に色づけする役割をはたす。属性値のデータは整数、実数、文字である。属性モデルは数個のリレーションによって構成されている。ネットワークモデルではラインとポイントのリレーションが、平面形状モデルではエリア、ライン、ポイントのリレーションがある。これらのリレーションの定義域はまったくアプリケーションに依存するものであり、次元数 (degree) はアプリケーションの要求に従って決定される。たゞし平面形状モデルのポイントリレーションはXとYの座標値をもたねばならない。Table 3は一例としてラインリレーションを示している。

Table 3. The Line Relation

Line Name	Color	Width	Cost
L1	RED	0.3	70
L2	BLUE	0.3	120
L3	GREEN	0.5	110

2.3 代替案モデル

代替案モデルは、いろいろ考察してきた代替案の関係を表現したものである。代替案の構造はツリー構造であり、一般に上層の案は設計過程の上流において作られた単純な案であり、下層の案は設計過程の下流において作られたものである。一つ上層の案から下層の案に加えた手順をアクションレコードとして記録する。この代替案の構造を表現したのが、代替案モデルである。このモデルは代替案リレーションに蓄積される。このリレーションの次元もアプリケーションに依存している。なぜならアプリケーションの必要に応じて、任意個のリンクモデルおよび属性モデルを同時に取扱うことができるからである。定義域として必要なものは次のものである。

1) 代替案の構造を管理するための2個の定義域

代替案名とその親の案名

2) 親の案から現在の案を作成してきた一連のアクションを記録したアクションレコードを蓄積する1個の定義域

3) 1個のネットワークモデルにつき、リレーションの識別名を蓄える3個の定義域

L-PP、ライン、ポイントのリレーションの識別名

4) 1個の平面形状モデルにつき、リレーションの識別名を蓄える5個の定義域

A-LL、L-PP、エリア、ライン、ポイントのリレーションの識別名
5) 代替案の属性を表現するための定義域

数はアプリケーションに依存している。たとえば代替のコスト等
代替案リレーションは第一次正規形をしていない。しかし別に不都合を生じないし、この場合は便利である。代替案リレーションは一種の制御テーブルであり、参照される従属したリレーションを指定するために用いられる。ある一つの案に付随するリレーションだけが一度に二次記憶から引きだされ、その内容は他の案とは独立である。

以上、リンクモデル、属性モデル、代替案モデルについて述べた。3次元物体のリンクモデルについては述べていないが、ネットワークモデルを用いれば例えばFig. 1-1bのように表現することが可能のはずである。また多くの場合、2次元の表現で十分である。

3 モデルの操作

モデルが設計者の操作によってどのように構成されたり、変更されるかを議論することは、モデルの表現の問題と同様重要である。リンクモデルは完全にIDASシステムによって処理される。代替案モデルも又、代替案の属性データをのぞいて処理される。属性モデルについては読み書きのインターフェースを提供している。本章では、代表的な例として、平面形状モデルにおいてラインが追加された時のモデルの処理について述べる。

3.1 モデルの変更のリクエスト

設計作業は、映像表示装置を用いて対話的に行われる。画面上にエンティティが絵として表示され、そして各々のエンティティはライトペンで直接識別される。Fig. 3に示れた絵が画面に表示されているとしよう。一本のラインをA1のエリアを横切って加えようとするすると、一連の操作は次のようになる。

ステップ1: ラインを定義するプログラムド・ファンクション・キーを押す。

ステップ2: 修正しようとするエリアA1をピックアップする。

ステップ3: 始点 (Fig. 5においてP5) をピックアップする。

ステップ4: 終点 (Fig. 5においてP6) をピックアップする。

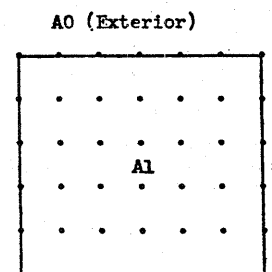


Fig. 3.
Image on the Screen

一方、IDASシステムの動きは次のとおりである[5]。

ステップ1: ラインを定義するアクション・ハンドラーを呼びだし、コントロールをこのアクション・ハンドラーにわたす。

ステップ2: 修正されるエリアA1を知る。

ステップ3: P5がL2上にあることを見つける。

ステップ4: P6がL4上にあることを見つける。

アクション・ハンドラーはアクションレコードを作る。アクションレコードは、①モデルを操作する入力情報、②設計過程の履歴情報、を表現するものである。アクションレコードの長さは、なされたオペレーションに従って不定である。

この例では7ワードの長さである。

<Define Line><A1><L2><L4><2><P5><P6>

3.2 モデルの変更

Table 4 は Fig. 4 に対応したリレーションを示している。これらのリレーションは上記のアクションレコードに従って変更される。その手順は次のとおりである。

ステップ1 : P5とL5 がポイント、ラインのリレーションにそれぞれ追加される。A-LL リレーションにおいて (A1, L2, L3) が (A1, L2, L5) と (A1, L5, L3) に分割される。また (A0, L3, L2) も (A0, L3, L5) と (A0, L5, L2) とに分割される。L-PP リレーションでは、(L2, P2, P3) が (L2, P2, P5) に変換し、(L5, P5, P3) が新しく挿入される。

ステップ2 : L4 が P6 によってステップ1と同じ手順で L4 と L6 に分割される。

ステップ3 : L7 がラインリレーションに加えられ、L-PP リレーションに (L7, P5, P6) が挿入される。A-LL リレーションの (A1, L2, L5) が (A1, L2, L7) に変更され、(A1, L7, L6) が新しく挿入される。ここで "L5-...-L4-L7" で1つの新しいエリアが生成されることが判明したため、エリアリレーションに A2 が加えられる。次に (A1, L4, L6) が (A2, L4, L7) に変更され、(A2, L7, L5) が挿入される。あと順番に (A1, L5, L3) が (A2, L5, L3) に、(A1, L3, L4) が (A2, L3, L4) と変更される。

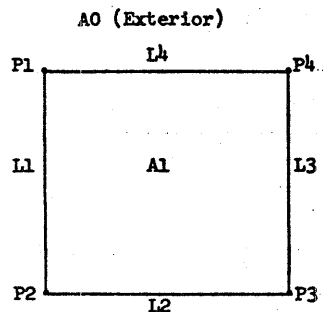


Fig. 4. Entities before Addition of a Line

Table 4. Relations before Addition of a Line

a) The A-LL Relation

Area Name	Key Line	Neighboring Line
A0	L1	L4
A0	L4	L3
A0	L3	L2
A0	L2	L1
A1	L1	L2
A1	L2	L3
A1	L3	L4
A1	L4	L1

b) The L-PP Relation

Line Name	Start Point	End Point
L1	P1	P2
L2	P2	P3
L3	P3	P4
L4	P4	P1

c) The area Relation

Area Name	
A0	
A1	

d) The Line Relation

Line Name	
L1	
L2	
L3	
L4	

e) The Point Relation

Point Name	
P1	
P2	
P3	
P4	

このようにラインの追加処理は複雑なプロセスを経て終了する。結果としての Fig. 5 に対応したリレーションが Table 5 に示されている。

Table 5. Relations after Addition of a Line

a) The A-LL Relation

Area Name	Key Line	Neighboring Line
A0	L1	L6
A0	L4	L3
A0	L3	L5
A0	L2	L1
A1	L1	L2
A1	L2	L7
A2	L3	L4
A2	L4	L7
A0	L5	L3
A0	L5	L2
A1	L6	L1
A0	L6	L4
A1	L7	L6
A2	L7	L5

b) The L-PP Relation

Line Name	Start Point	End Point
L1	P1	P2
L2	P2	P5
L3	P3	P4
L4	P4	P6
L5	P5	P3
L6	P6	P1
L7	P5	P6

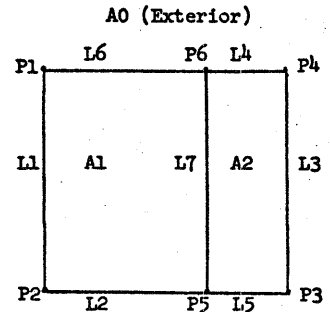


Fig. 5. Entities after Addition of a Line

c) The Area Relation

Area Name	
A0	
A1	
A2	

d) The Line Relation

Line Name	
L1	
L2	
L3	
L4	
L5	
L6	
L7	

e) The Point Relation

Point Name	
P1	
P2	
P3	
P4	
P5	
P6	

4 BARTH について

以上、モデルの表現ならびにその操作について述べた。本章では我々の開発したリレーションハンドラー BARTH (Basic Relation Table Handler) [6] を紹介する。BARTH は GAMMA-O [7] レベルの基本ルーチンであり、対話的に使用されることを考慮して時間効率に力点を置いている。

4.1 BARTH の機能

A) リレーション単位

- 1) リレーションの定義
- 2) リレーションの削除
- 3) リレーションの活性化
- 4) リレーションの非活性化
- 5) リレーションのコピー

リレーションが非活性化されると、そのリレーションは主記憶装置から追いだされる。この時、そのリレーションはそれまでの変更部分を考慮して二次記憶にしまうのか、または無視してしまうのか任意である。これはリレーションを試行錯誤の過程で仮に変更したり、または途中でそれまでの過程を放棄したりする時に便利である。またリレーションのコピーも、わずかに異なる代替案を作成する場合などで有効である。

B) タプル単位

- 1) タプルの定義
- 2) タプルの削除
- 3) タプルの変更

4) タップルの識別名を与えることによるタップルの読み込み

5) 一つの定義域に条件を与えることによるタップルの読み込み

データのタイプとしては、整数と実数ならびに4バイトの文字である。検索条件は、 $>$, $\geq$, $=$, $\leq$, $<$, キ, 無条件の7つである。

4.2 BARTHの内部構造

リレーションはRID (Relation Identifier) で、タップルはTID (Tuple Identifier) で識別される。そしてRIDやTIDはリレーションやタップルが作成された時にBARTHによって割り当てられる。TIDは1つのリレーションの中でのタップルの順番を示している。このためTIDを用いることにより、直接リレーションの中のタップルを検索することができる。種々のリレーションを制御するため、マスターリレーション(MR)を導入して、MRの1つのタップルで1つのリレーションの情報を保持している。MRのTIDはそれが制御するリレーションのRIDを示している。

リレーションを蓄積する場所はページングしている。ページの容量をこえて新しいタップルが定義されると、新しいページがそのリレーションに割り当てられる。ページの中ではタップルはTIDの順番に従って蓄えられる。この手法により、XRM[8]などと異なり、ハッシングすることなく、迅速に必要なタップルを検索することができる。ただ、1つのページは1つのリレーションしか含むことができない。このためページの大きさはアプリケーションに応じて定めることができるようになっている。またすべての活性化したリレーションは主記憶装置へ存在する。これはリレーションが非常に大きかったり、活性化されたリレーションの数が多いと問題となる。しかしCADにおける実際のリレーションはそんなに大きくないし、一時期に必要な活性化されたリレーションの数もそう多くない。

映像表示装置を使用するため、エンティティを指示するのにライトペンを用いて直接行うことができる。このことによりIDASではエンティティの名前はTIDで行うことができる。リンクモデルでエンティティ名が現われたが、これらはすべてTIDである。対応するエンティティはこのTIDを用いて関係モデルから直接検索することができる。このようにBARTHは、特に対話的に用いられる時有効である。

5 おわりに

以上述べたIDASのモデルならびにBARTHを検証するために、建築計画に応用したプロトタイプ[9]を作成した。このアプリケーションでは、ネットワークモデルを構造モデルに使用した。ポイント、ラインにはそれぞれ柱と梁が対応する。一方、平面形状モデルを配置モデルに使用した。エリアは部屋に、ラインは壁に対応する。Fig. 6はオフィスの平面計画の一例である。Fig. 6に対応する代表的リレーションの次元と基数とがTable 6に示してある。これらリレーションのデータ量は、

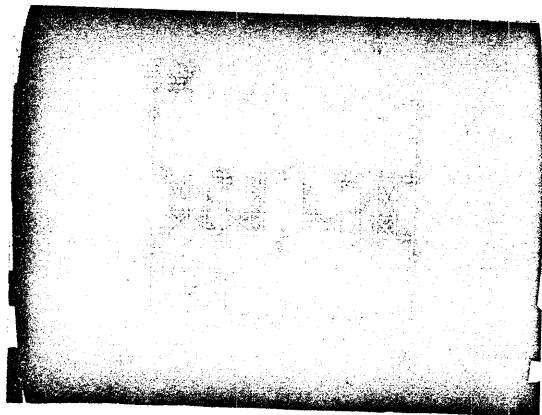


Fig. 6. Room Arrangement in the IDAS system

1 K バイト ページを用いた場合、約 20 ページ——すなわち 20 K バイトである。Table 6 に示していないリレーションが Fig. 6 には存在するが、データ量のオーダーから考えると、全部のリレーションは主記憶装置に十分はいりうる。他の案に付随するリレーションは、主記憶装置ではなく二次記憶装置に蓄えられている。

リレーショナルモデルはプログラマーや設計者にとって非常にわかりやすい形式をしている。設計物の中味を調べるためには、ポインターを追うなどという必要はなく、テーブルを調べさえすればよい。この点は重要である。特にモデル操作のサブルーチンの開発の際、虫取りをする作業でリレーショナルモデルの有効性を強く認識した。

以上 IDAS におけるモデルについて述べてきたが、計画、基本設計段階で決定されたデータは設計の下流段階、生産段階へ伝達される必要があり、そのためには計画、基本設計段階のモデルは、単一なデータベースに統合化される必要がある。IDAS ではこの統合化されたデータベースをプロジェクトデータベース [10] と呼んでいるが、プロジェクトデータベースは多人数の設計者（建築設計でいえば、意匠設計者、構造設計者、設備設計者等）が、各自別々のモデルで設計を進めた場合も、情報伝達の接点となる統一化されたデータベースである。各設計者は、自分の設計の開始時にプロジェクトデータベースから自分のモデルに必要なデータを引きだす（Data Base Extraction という）、設計の終了時にプロジェクトデータベースへ変更された部分をもどす（Data Base Integration という）。この統一化されたデータベースならびに Data Base Extraction/Integration の問題は次の研究課題と考えている。

Table 6. The Size of Typical Relations

	Relation Name	Degree	Cardinality
Geometric Model	A-LL	3	160
	L-PP	3	80
	Area (Room)	13	28
	Line (Wall)	16	80
	Point	4	54
Network Model	L-PP	3	52
	Line (Girder)	6	52
	Point (Column)	9	32

参考文献

- [1] 松家、川井、宇野、「設計活動援助統合化システム (IDAS) の概念」、IBM Scientific Center, N: GE18-1813, 1975
- [2] 古川、「コンピュータ・グラフィックスにおけるデータ構造の問題」、情報処理、Vol 11, No. 9, P. 523-532, 1970
- [3] Williams, R., A Survey of Data Structures for Computer Graphics Systems Computing Surveys, Vol 3, No. 1, P. 1-21, 1971
- [4] 川井、「設計活動援助統合化システム (IDAS) におけるモデル管理」、IBM Scientific Center, N: GE18-1817, 1975
- [5] 宇野、「設計活動援助統合化システム (IDAS) における対話管理」、IBM Scientific Center, N: GE18-1820, 1975
- [6] Uno, S., Basic Relational Table Handler, IBM Scientific Center, GE18-1816, 1975
- [7] Björner, D., Codd, E.F., Deckert, K.L., Teague, I.L., The GAMMA-D n-ary relational data base interface specifications of objects and operations

IBM Research, RJ1200, 1973

- [8] Lorie, R. A., XRM—An Extended (N-ary) Relational Memory, IBM Scientific Center, G320-2096, 1974
- [9] 松家、川井、宇野、「設計活動援助統合化システム—建築計画用—(IDAS-A) の概念」、IBM Scientific Center, GE18-1818, 1975
- [10] 樺、「CHEIS—Chiyoda Engineering Information System」 IBM 第2回 CAD シンポジウム 報告書、P.122-148, 1973

付録

IDASは主に4つの構成要素、アクション・ハンドラー、モデル・ハンドラー、プロジェクト・ハンドラー、ディスプレイ・ハンドラーよりなる。アクション・ハンドラーは対話管理とアクションレコードの作成を行う。モデル・ハンドラーはアクションレコードに従ってモデル操作を行う。プロジェクト・ハンドラーは、点とか線というようなイメージの情報をもつプロジェクトレコードをモデルの情報より作成する。ディスプレイ・ハンドラーは映像表示装置へ絵を表示する。

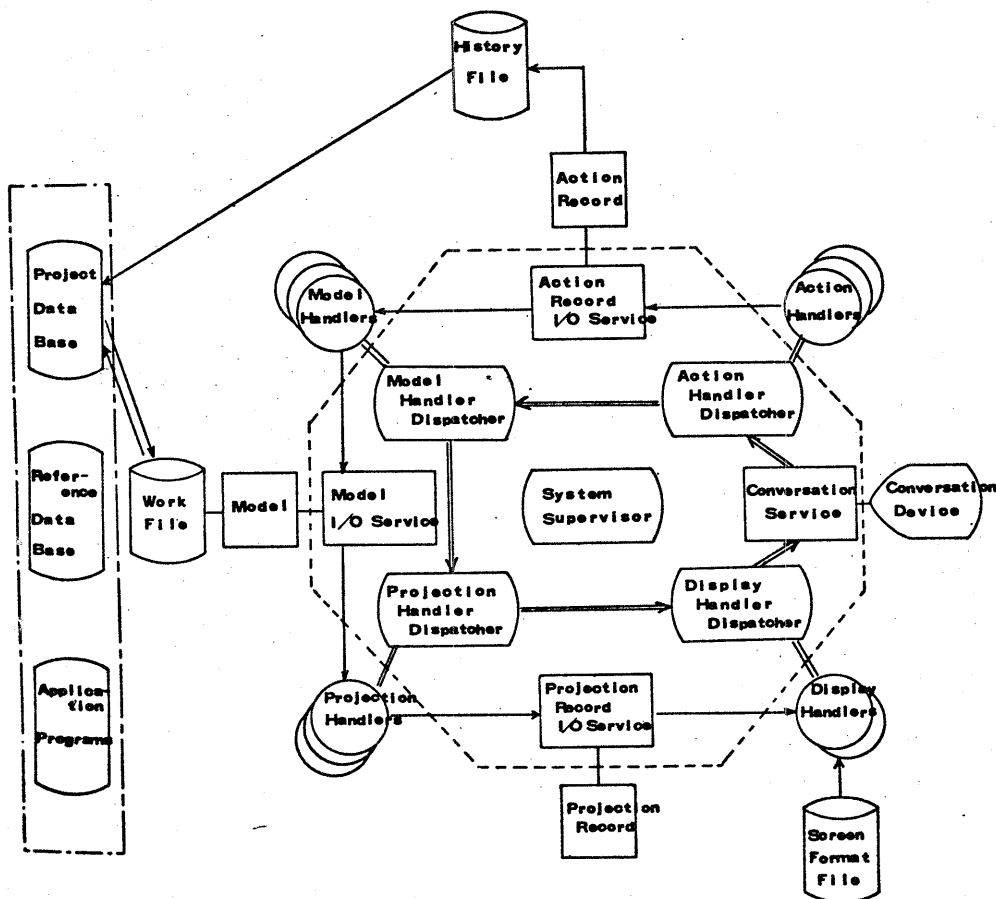


Fig. 7. Structure of the IDAS System