5M-11

知識ベースとデータベースの相互変換法

山多　昭　中野　勝之　池田　幸雄
NTT電気通信研究所

1．はじめに

エキスパート・システムは，知識ベース構築支援ツールを使用して構築されるが，エキスパート・システムの作成に既存のデータベースの利用が必要であることがある。また，知識ベース構築支援ツールを用いてプロトタイプを作成し，その実現性を確認した後に，性能の要求からデータベース・システムとして実用化することが考えられる。即ち，既存の知識ベースとデータベースの資産を相互利用する方式を確立する必要がある。　本稿では，知識ベースから代表的知識表現であるフレーム表現，データベースから関係データベース及びCODASYL データベースを取り上げ，知識ベースとデータベースの相違を明確にする。次に，知識ベースとデータベース間で相互の形式に近似的に変換する方法を報告する。

2．知識ベースとデータベースの表現と相違点

2．1　フレーム表現

フレームは，ひとまとまりの知識であり，スロットとして情報が付加される。本稿では，知識ベース管理システム(KBMS)の知識表現を具体的なフレーム表現として取り上げる。[1] フレームには，型を表わすクラスとその実現値を表わすインスタンスがあり，フレームとスロットへアクセスする機能がある。スロットには，値を格納する変数，手続きを格納するメソッドがある。スロットはフレーム間の関係も表現できる。知識を構造化する関係として，IS-A（上位，下位の階層関係）PART-OF （全体と部分の関係）およびINSTANCE-OF （クラスとインスタンスの関係）がある。更に，フレーム間でスロット（属性）の継承ができる。

2．2　関係データベースの表現

関係は，定義域の直積集合の部分集合である。関係の各要素を組といい，関係の列を属性という。属性の集合と関係名をスキーマといい，データの型を表わす。組はデータの実現値を表わす。仮想の関係であるビューがある。データの操作はスキーマ，ビューを通じて組に対して行われる。また，データベースの一貫性を保つ制約条件（トリガ，ASSERT）がある。

2．3　CODASYL データベースの表現

レコードは，データの最小単位であるデータ項目の集合である。セットは，レコードとレコードの関連づけである。レコードとセットには，型であるスキーマと実現値であるオカレンスがある。部分データベース定義するサブスキーマがある。データの操作はスキーマ，サブスキーマを通じて実現値に対して行われる。制約条件として，レコードの挿入削除を規定するセットメンバシップ，セットへのアクセスを規定するセット選択基準や特定のデータ操作毎に呼び出されるCALL句がある。

2．4　相違点

フレーム表現は，関係データベースやCODASYL データベースより表現能力がある。表現の相違点を表１に示す。

表1　表現の相違点

項　目	フレーム表現	関係データベース	CODASYLデータベース
表現対象	データ，制約条件	データ，制約条件	データ，手続き
表現要素	関係，属性	レコード，セット，データ項目	フレーム，スロット
型と実現値の分離	有	有	フレーム内に混在
構造化する関係	無	セット	IS-A，PART-OF，INSTANCE-OF
属性の継承	無	無	有

3．変換法

3．1変換方針

データ，制約条件を変換の対象にする。変換元のデータの意味，制約条件，トリガは変換先の等価な表現に変換されるべきであるが，表現力が異なることにより等価な表現への変換は実現できない。また，変換先のアクセス機能により同等な情報が取り出せるように，データは変換される必要があるが，CODASYL データベースのセット選択基準のような表

A Transformation Method between Knowledgebase and Database
Akira YAMADA ,Katuyuki NAKANO ,Sajio IKEDA
NTT Electrical Communications Laboratories

現法固有のアクセス方法は，変換先で実現される必要はない。このため，変換元のデータ表現に基づいたデータの意味をデータの操作可能な変換先のデータ表現へ近似的に変換する。

３．２　変換規則

フレーム表現から関係データベースとCODASYL データベースへの変換では，フレーム，IS-A関係，属性の継承等をスキーマ，実現値，制約条件（継承される属性を更新するトリガまたはCALL句）へ変換する。メソッドは利用者が変換する。主な変換規則を表２，３に示す。また，フレーム表現から関係データベースの変換へ本規則を適用した例を図１に示す。

関係データベースからフレーム表現への変換では，スキーマ，実現値および制約条件が，それぞれクラスフレーム，インスタンスフレーム，一貫性を保つメソッドへ変換される。メソッドの起動は利用者が制御する。また，ビューの変換は，インスタンスを作成せずに実現すべきであるので行わず，利用者が必要に応じてメソッドを作成する。関係データベースからフレーム表現への主な変換規則を表４に示す。

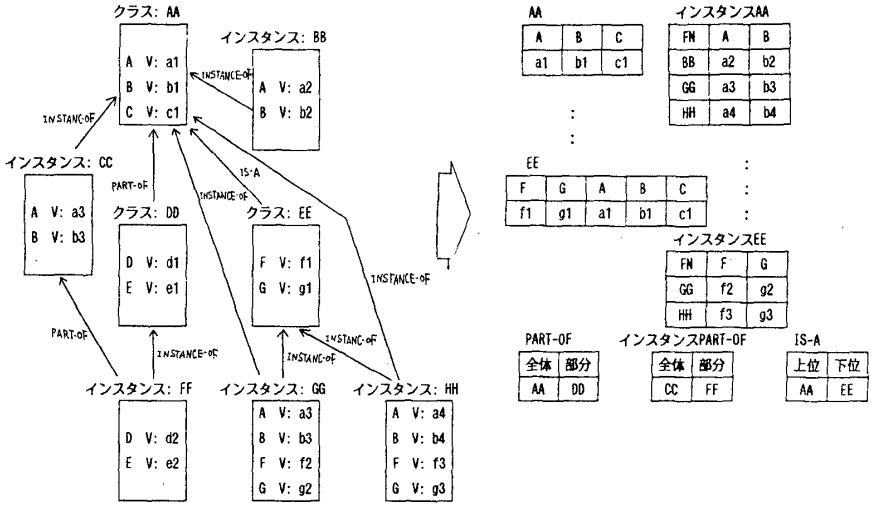
CODASYL データベースからフレーム表現への変換では，セットは，PART-OF 関係へ変換され，セットメンバシップは，部分クラスのインスタンスの存在を制限するメソッドへ変換される。

表２　フレーム表現から関係データベースへの主な変換規則

フレーム表現の要素	規　則
フレーム	・クラス毎に変数を属性，フレーム名を関係名としたスキーマへ ・クラス単位にインスタンスの変数とフレーム名を属性，インスタンスで修飾したクラス名を関係名としたスキーマへ ・変数の値とインスタンス名を組へ
ＩＳ－Ａ関係	・上位と下位のクラス名を属性，ＩＳ－Ａを関係名としたスキーマへ ・下位クラスのインスタンスに対応するスキーマと組から上位のクラス名のインスタンスからに対応するスキーマと組の削除
ＰＡＲＴ－ＯＦ関係	・クラスとインスタンスに全体と部分のフレーム名を属性，ＰＡＲＴ－ＯＦを関係名としたスキーマへ ・フレーム名を組へ
属性（変数）の継承	・継承する変数と値を対応するスキーマと組へ追加 ・継承値の更新を制約条件へ
メソッド	無

表３　フレーム表現からＣＯＤＡＳＹＬデータベースへの主な変換規則

項　目	規　則
フレーム	・クラス毎に変数をデータ項目，フレーム名をレコード名へ ・クラス単位にインスタンスの変数とインスタンス名をデータ項目，インスタンスで修飾したクラス名をレコード名へ ・クラスとインスタンスに対応するレコード間へセットを追加 ・変数の値やフレーム名はオカレンスへ
ＩＳ－Ａ関係	・上位と下位のクラスに対応するレコード間へセットを追加 ・上位クラスと下位クラスのインスタンスに対応するレコード間へセットを追加
ＰＡＲＴ－ＯＦ関係	・全体と部分のフレームに対応するレコード間へセットを追加
属性（変数）の継承	・継承する変数をデータ項目としてレコードに追加 ・継承する変数の値をデータ項目の値として追加 ・継承値の更新をＣＡＬＬ句へ
メソッド	無



(注１) XX V: YYのとき　XX: 変数，YY: 値　　(注２) FN：フレーム名

(1)　フレーム表現　　(2)　関係データベース

図１　フレーム表現から関係データベースへの変換例

表４　関係データベースからフレーム表現への主な変換規則

項　目	規　則
スキーマ	・属性を変数，関係名をクラス名としたクラスへ
組	・組をインスタンスの変数の値としたインスタンスへ
制約条件	・制約条件を一貫性を保つメソードへ
ビュー	・利用者がビューを実現するメソッドを作成

４．おわりに

フレーム表現と関係データベースやCODASYL データベース間でデータを主体として相互の形式に近似的に変換する方法を提案した。本方法には，利用者がメソッドの起動制御を意識しなければならないという問題がある。今後手続きの変換方法を検討するとともに，上記の問題の対策を検討する予定である。

参考文献　[1] 服部他，“知識ベース管理システム(KBMS)”情処知識工学と人工知能研究会資料 41-6(1985)