

鉄鋼プロセス制御システム開発一貫支援システム

—設計データベース—

1 Q-5

北山貴広 川辺誠司 竹中一起  
住友金属工業(株)

1. 緒言

鉄鋼プロセス制御システム開発一貫支援システム<sup>①</sup>では、約50種類のドキュメントエディタを提供しているが、これらのエディタを介して入力された設計情報は、設計データベースで一元管理され、細粒度のデータ統合を実現している。既存ツールの切り貼り機能では、オリジナルの部分を変更しても貼り先は元のままであるが、本支援システムでは一箇所修正すれば、同一のデータを表示しているところはすべて自動的に変更される。

本稿では、設計データベースの構造、データ統合のメカニズム、設計検証機能について述べる。

2. RDBによる設計情報の一元管理

設計情報のような複雑な構造を持つ情報をデータベース化する場合、複雑な構造をそのまま格納できる点でオブジェクト指向データベースが有利であるが、今回は確立した技術を採用したいとの

意向から、リレーショナルデータベース(RDB)を採用し、個々の設計工程毎に、オブジェクト間の関連や構造を以下のようにモデル化した。

- (1) Node Entity表: ノードに関する親子の構造情報と、名称や概要説明等の属性情報を保持する。
- (2) Relation表: ノード間の接続情報とアーク名とのリンク情報を保持する。
- (3) Arc Entity表: アークに関する親子の構造情報と、名称や概要説明等の属性情報を保持する。

個々のオブジェクトを詳細化したメンバについても同様にモデル化した(図1)。

3. エディタとのインターフェイス

設計データベースは物理的には、前述のように各工程、各レベル毎に複数の表で実現するが、エディタからは論理的な意味を持った形で取り扱いたい。そこでこの間の意味上のギャップを埋めるインターフェイスを開発し、エディタからはER

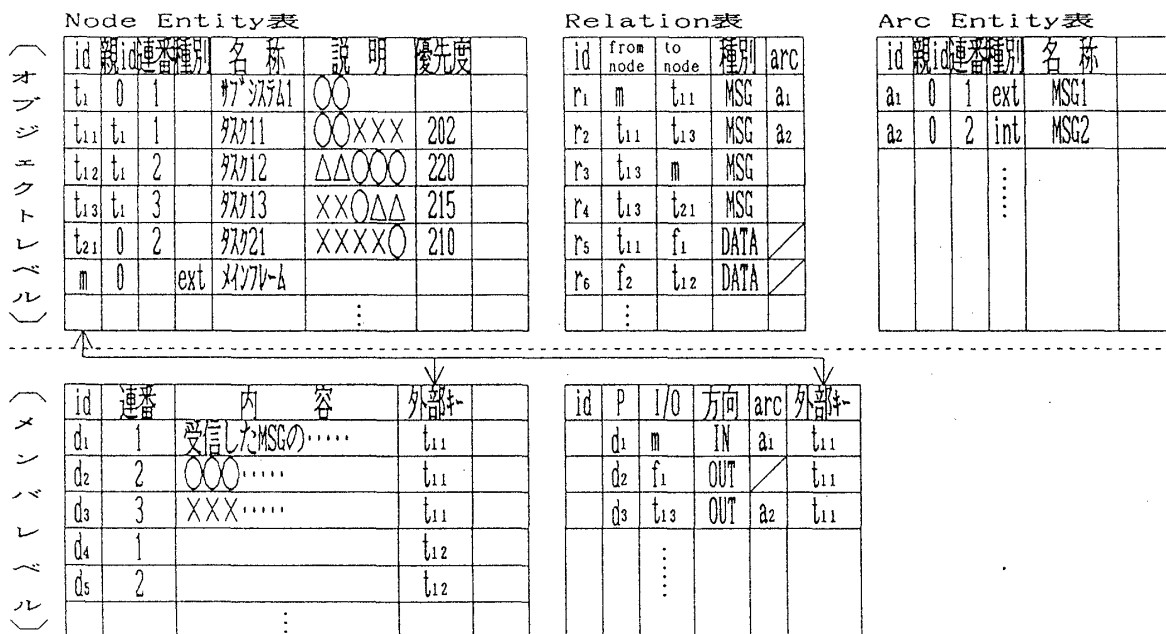


図1 RDB上に構築した設計データベースの構造

モデル型でアクセス可能とした。

またエディタと設計データベースとの間の対応をこの層で切り替えることにより、同種のエディタは一つの実行モジュールで複数のドキュメントエディタとして機能するようにしている。

#### 4. モデル・ビュー間の整合性の維持

ドキュメントエディタによる設計結果のうち、本質的なモデル情報は設計データベースに格納されるが、ドキュメント上の位置や大きさ等のビュー情報はファイル内に格納されている。ここではモデル情報とビュー情報間の整合性維持メカニズムについて述べる。

##### (1) 起動時

前回の編集セッション後、別のドキュメントエディタにより、設計データベースの内容が変更されている可能性がある。そこで設計ドキュメントがロードされると、まずビュー上に存在する全てのオブジェクトについて消去されていないかと内容に変更がないかを調べ、その結果をビューに反映する。次に設計データベース中に新たに生成されたオブジェクトを求め、存在する場合はそれを適当な位置に記述する。新規のオブジェクトは、編集セッションの中で見やすい位置に配置すれば、そのビュー情報は以後も保存される。

##### (2) 編集時・他エディタの編集終了時

設計ドキュメントの編集は、ある工程のあるレベルの設計データベースを対象に行われるが、エディタによっては上流工程のデータベースを参照している場合がある。他のエディタで上流の設計データベースの内容が変更された場合、変更されたオブジェクトのIDリストが配信されてくるので、そのオブジェクトがエディタに含まれているか否かを調べ、含まれている場合は最新の設計データベースの内容を読みだし、ビュー上にも反映する。

#### 5. 検証機能

設計データベースの内容を基に以下のような検証を行い、不足点や矛盾点を設計者に指摘する。

##### (1) メンバレベルの未定義

機能に対しては機能仕様書（I P O）、ファイルに対してはファイル構造図のように、オブジェクトに対する詳細が定義されているかを、設計データベースのメンバレベルを検索して調べる。

##### (2) 外部イベントの網羅性

上流で定義された通信メッセージやプロセスイベント等の外部入出力が、下流の対応するドキュメント上に最低1回は出現しているかを調べる。

##### (3) 接続情報の整合性

図1に示すように、各工程の設計データベースには、オブジェクトレベルとメンバレベルにそれぞれRelation表があり、接続関係の情報を保持している。ここではこれら2つのレベルの接続関係に過不足がないかを調べる。具体的には、オブジェクトレベルに存在する接続関係がメンバレベルに存在しない場合は、「機能関係図で機能Aから出力しているYYが、機能仕様書で定義されていません」、逆の場合は「機能仕様書Aで定義された出力のZZが、機能関係図に反映されていません」等の指摘を行う。

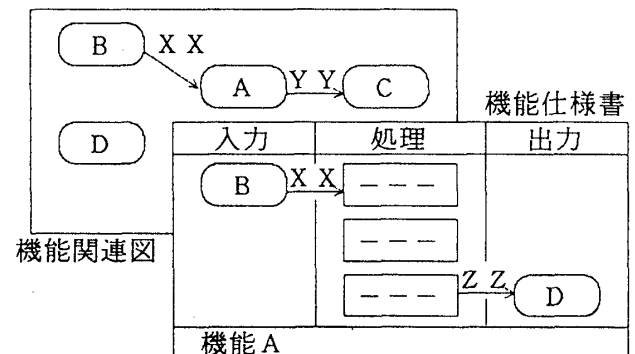


図2 接続情報の検証

#### 6. 結言

ソフトウェアの設計情報をRDB上でどのように表現するかについて述べた。設計情報をデータベースで一元管理することにより、あるドキュメント上で設計・変更された情報は、別のドキュメントにも自動的に反映できるようになり、細粒度のデータ統合を実現できた。

#### 【参考文献】

- [1] 竹中他: 「鉄鋼製制御系開発一貫支援システムの概要」  
情報処理学会第47回全国大会予稿集 IQ-3(1998).