

3J-6

ソフトウェア設計・製作支援システム CADRIS (3)
 - 仕様蓄積方式 -

須田 正志^{*} 川又 康宏^{*} 柳 一美^{**} 高橋 勇喜^{***}
 茨城日立情報サービス(株)^{*} (株)日立製作所 システム開発研究所^{**} (株)日立製作所 大みか工場^{***}

1. はじめに

日立製作所大みか工場では、計算制御のソフトウェア開発を対象にソフトウェア設計・製作支援システム CADRIS (Computer Aided Design and Reuse Environment with Intelligent Support system)を開発し、社内での適用を進めており、ソフトウェアの生産性、信頼性の向上に大きな成果をあげている。

ソフトウェアの開発環境では、開発プロセスで生成される成果物を適切に管理することが要求されている。

CADRISでは成果物の設計仕様、ソースコード等の全ての情報を実体・関連モデルに基づきモデル化し、このモデルに従って全ての支援機能を有機的に結合している。

本稿では、設計工程、プログラミング工程を主対象とした仕様データベース(SEDB: Software Engineering DataBase)の特徴とインプリメントについて述べる。

2. CADRISのデータベースの特徴

ソフトウェア仕様データベースとして、商用の高機能データベースを用いた場合、

- (1) ソフトウェアの仕様を表現するために必要な記述言語やプログラミング言語に従った複雑な構造をもつデータを扱うことが難しい。
- (2) データベースマネージャ等の常駐タスクの存在による負荷が高く、多くのツールが同時に実行される開発環境には向かない。
- (3) データベースを利用する開発者は、多くの場合一人ではない。ツールが多数の開発者によって複数起動された時、使用に耐えられるだけの処理速度を得ることは、難しい。

等の問題が挙げられる。
 CADRISのデータベースの開発では、

- (1) ソフトウェアの仕様の表現に適していること。

- (2) データベースを使用するCADRISのツールのアクセスを想定したデータ管理により、高速でコンパクトなデータベースを実現すること。

を目標とした。

2. 1 実体・関連モデルの採用

周知の通り、設計工程で扱われる仕様情報は、ソースコードを含め数十種類に及ぶ。しかも、これらの情報は複雑な構造を持っており、情報間には、入り組んだ依存関係が存在する。更に、これらの仕様情報は、常に矛盾のない状態で保持する必要がある。そこでCADRISでは、実体・関連モデルを採用し、図1のように、ソフトウェアの構造を定義することとした。

2. 2 データ管理の特徴

CADRISのデータベースは、実体と関連によって構成されている。実体は、データベース内でユニークな存在であり、仕様情報を一意に示している。また、関連を辿ることにより、変更のあった実体から関連するすべての実体を導き出すことができる。

(1) 実体

実体は、その存在を示すためのID情報(Identification number)と実体を表現するための属性からなる。

ID情報はデータベース内で一意性を保証している。属性は、整数型、名前型、文字列型、テキスト型、ファイル型のレコードからなり、仕様情報を保

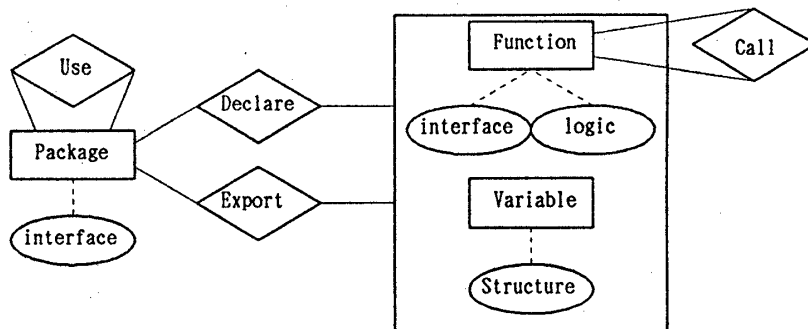


図1 ソフトウェア構成定義例

"Design Support Environment for Realtime Software System CADRIS, A Method of Stored Software Specification"
 Masashi Suda^{*} Yasuhiro Kawamata^{*} Kazumi Yanagi^{**} Yuki Takahashi^{***}
 Ibaraki Hitachi Information Service Co.,Ltd^{*} Systems Development Laboratory, Hitachi, Ltd.^{**} Omika Works, Hitachi, Ltd.^{***}

持っている。

名前型、文字列型、テキスト型はともに文字列であるが、名前型は、実体の名前として使用され、局所的な一意性を保証する。テキスト型は、可変長のデータ型である。

ファイル型は、オブジェクトコードやロジック情報を記憶するために準備している。

(2) 関連

関連は、実体と実体間の関連を示す情報であり、二つの実体のID情報によって表現される。

関連には、単純に二つの実体間の関連を示すものと存在依存の関連を示すものがある。

2. 3 再利用・流用

計算機制御のソフトウェア開発では、分野ごとの技術の蓄積を図らなくては、早期開発に対応できない。再利用・流用率60～90%とも言われる開発・保守作業を円滑に行うための技術が重要である。CADRISのデータベースでは、ソースコードを含めた仕様レベルの再利用・流用をサポートしている。

3. インプリメント

3. 1 ファイル構成

CADRISのデータベースは、実体に対する実体ファイル、関連情報を格納した関連ファイル、名前型データを格納した名称管理ファイルおよび存在依存関連をしめすディレクトリによって構成される。(図2)

3. 2 存在依存関連の実現

存在依存関連は、二つの使用目的を想定し、異なった二つの方法によって実現している。一つは仕様間の包含関係であり、もう一つはファンクション仕様におけるパラメータのような仕様内の包含関係である。

仕様間の包含関係はディレクトリ構成によって実現し、仕様間の包含関係は実体ファイル内のデータとして管理することにより実現している。各々、その存在を他に依存している実体は、上位の実体(ディレクトリまたはファイル)を削除されることにより自動的に消滅する。

3. 3 検索

CADRISでは、実体と関連に対して集合検索を用意している。

CADRISのデータベース構造は、ツールのアクセスを想定したものであり、データベースそのものが、検索の条件に基づいた構造になっている。従って、頻繁に発生する検索に対しては、検索方法に特別な技法を用いることなく高速な検索を実現している。例えば、パッケージが包含するファンクションを取り出すと言う関連の集合検索を行う場合、パッケージとファンク

ションの間の包含関係は存在依存関連であるため、目的とするファンクションは、パッケージ名に対応するディレクトリの下を検索するだけでよい。更に、ファンクションの名前は、名称管理ファイルによって一括管理されており、目的とするファンクション名はファンクションの名称管理ファイルから取り出すだけでよい。

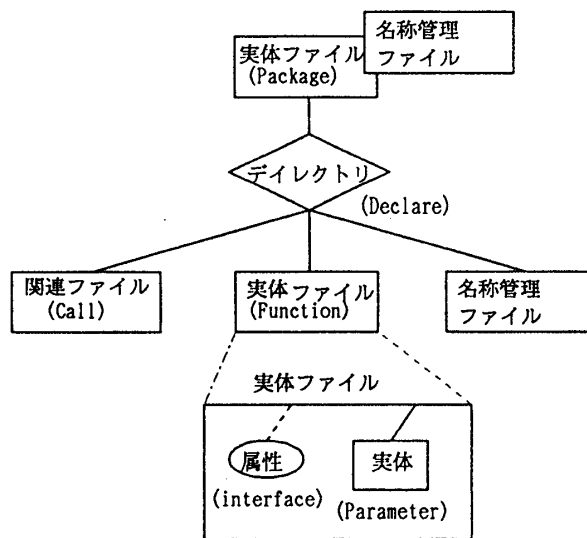


図2 ファイル構成

3. 4 排他制御

CADRISのデータベースでは、ファンクション等の仕様を一つの編集単位とし、一つの仕様は一つのファイルに割り付けられている。従って、排他制御も極めて単純であり、対象となる仕様(実体)ファイル全体をロックすることにより実現している。

4. おわりに

今回の仕様データベースの開発では、多人数による操作時の応答性及び、大量情報の中から目的とする情報取り出しの容易性を十分に確保することができた。

また、最近では分散開発環境・テスト環境との統合化支援に対する要求が高まっており、今後は、CADRISの仕様データベースもこれらの要求を満たしたものに発展させていく予定である。

参考文献

- (1) 大脇、他：ソフトウェア設計・製作支援システム CADRIS - 開発思想 -, 第44回情報処理学会全国大会 論文集 3J-4(1992年3月)
- (2) 高橋、他：ソフトウェア設計・製作支援システム CADRIS - プロセス支援方式 -, 第44回情報処理学会全国大会 論文集 3J-5(1992年3月)