

7D-6

ECU 組込みソフトウェア開発のための  
技術文書と設計ツールとの統合

荒木 円博, 手嶋 茂晴  
株式会社 豊田中央研究所

E-mail: foison@mes.tytlabs.co.jp, teshima@mes.tytlabs.co.jp

1 はじめに

我々は、ECU(Electronic Control Unit) 組込みソフトウェア(車載電子制御システム用のソフトウェア)の開発を支援するため、設計・開発用のツールと技術文書の統合化について研究している。ECU組込みソフトウェア開発では、プログラムサイズや応答時間の制約が厳しいため、仕様が車種ごとに特化して具体化される。そのため、要求仕様レベルで同じ機能のソフトウェアの派生版が多数作られる。このような派生の関係も考慮して、ソフトウェアを構成する多数の技術文書を扱うためには、ファイルシステムの提供する階層構造では不十分である。

そうした問題を解決する既存の開発環境として、PCTE<sup>[1][2]</sup>などがある。これらは、文書間のさまざまな関係に従って文書間を巡航する機能と、文書の作成や変更に対して文書間の関係の一貫性を維持する機能を提供している。しかし ECU 組込みソフトウェア開発では、ファイルシステム上の文書を直接扱うツール(e.g. 制御系シミュレータ)を使うため、一貫性の維持の点で問題がある。

そこで、我々は、これらの問題を解決するための文書管理モデル *Liaison-model* を考案し、それに基づく開発環境のプロトタイプを開発した。

2 文書管理モデル *Liaison-model*

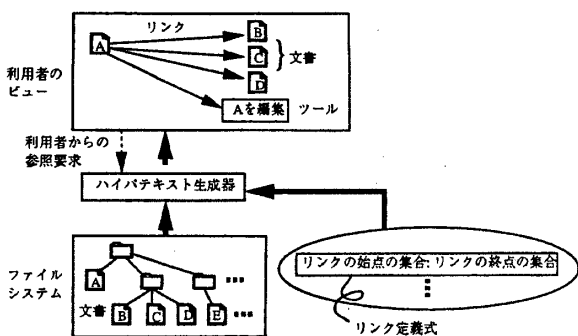


図 1: 文書管理モデル *Liaison-model*

Integration of Documents and Tools for ECU Embedded Software Development  
Mitsuhiro Araki, Shigeharu Teshima  
Toyota Central R&D Labs., Inc.

我々が提案する文書管理モデル *Liaison-model*(図 1 参照) は、文書そのものを管理するのではなく、文書間の関係において一貫性を保ったビューを管理するモデルである。一貫性の維持は、文書の参照要求の都度、後述のリンク定義式に従って、参照要求された文書とその文書に関係のある文書やツールとの間にリンクを付与したハイパテキストを生成することによって行う。参照要求時にハイパテキストを生成するので、ファイルシステム上の最新の状態を反映したビューができる。

利用者は、このハイパテキストによるビューを利用して、直観的に、さまざまな関係に従って文書間を巡航したり、ツールを起動することができる。

リンク定義式は、始点の集合と終点の集合の対で、複数存在する。始点の集合は、文書の集合か、文書とそれに含まれる字句の対の集合で、パス名や内容の正規表現、その and, or による組合せによって内包的に表す(e.g. 「ABSダイレクタリ以下にあり、かつ拡張子が .c であるファイル」)。終点の集合も同様であるが、定義の中で始点の集合を表す正規表現にマッチした文字列が使える(e.g. 「始点のファイル名で、かつ拡張子が .text であるファイル」)。これらのリンク定義式のうち、参照要求された文書が、始点の集合に該当するものを使って、始点の集合と終点の集合との間でリンクづける。

このリンク定義式によって、ECU 組込みソフトウェア開発に必要な以下のリンク(図 2,3 参照)が表現できることを示す。

- (1) 具体化のリンク(e.g. 仕様書→ソースコード)
- (2) 具体化の根拠(e.g. 仕様に関する車種ごとの特殊事情)へのリンク
- (3) 抽象化のリンク((1)の逆向きのリンク)
- (4) 文書中のファイル名からのリンク
- (5) 索引の文字列から、本文へのリンク
- (6) 改訂履歴へ/からのリンク
- (7) 文書の一覧から、一覧中の各文書へのリンク
- (8) ある文書から、その文書を含む一覧へのリンク
- (9) 文書操作ツールへのリンク(e.g. エディタへ)

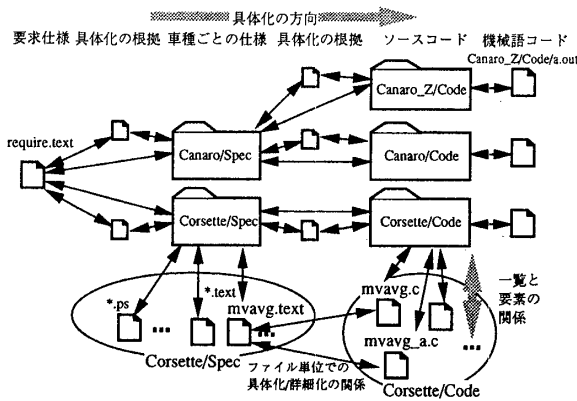


図 2: 文書単位でのリンクの例

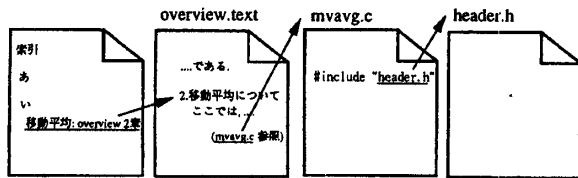


図 3: 字句単位でのリンクの例

(1),(2),(3) は、パス名の命名規則を、例えば

- 具体化する前の名前と同じものにする。
- 派生版は `..派生名` を追加した名前とする。

と定めておけば表現できる。例えば図2の車種ごとの仕様書からソースコードへのリンクは、  
 始点の集合: `(*)/Spec/(*) .text`  
 終点の集合: `$1(*_*)/Code/$2(*_*) .c`  
 と定義できる (\*: 任意文字列, (): グループ化, x|y: xかy, \$n: n番目のグループにマッチした文字列)。

(4) は、例えば図3の `#include` からの場合、  
 始点の集合 (文書): `*(.c|h)`  
 始点の集合 (字句): `#include "(*)"`  
 終点の集合 (文書): `$2`  
 と定義できる。(5) も同様である。

さらに、コマンド実行結果も「文書」とみなすことにより(6),(7),(8),(9)も表現できる。例えば(6)は版管理コマンドの出力を文書とみなせばよい。ツールの起動だけ行う(9)の場合は、終点の字句を空文字列とすることによって表す。

### 3 プロトタイプの実現方法

クライアント-サーバのシステム構成とし(図4参照)、クライアントとの通信にWWWを利用した。ファイルシステムにはNFSなどの分散ファイル

システムを利用し、ハイパテキスト生成器はPerl言語で実装した。なお、リンク定義式は、生成器の中に手続き的にハードコーディングした。

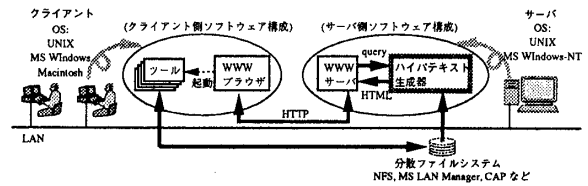


図 4: プロトタイプのシステム構成

### 4 適用例

ABS(Anti-lock Brake System)用ソフトウェアを対象に下記の機能を実現し、実際の開発チームを対象に評価を行った。

- 仕様書と、それを具体化したソースコードとの間のリンクづけ
- ディレクトリー一覧へ/からのリンクづけ
- 全文検索結果からのリンクづけ
- テキストエディタの起動
- 版管理ツール RCS との連携

その結果、実用に耐える応答時間でリンクづけでき、文書参照の効率が改善されたことを確認した。

ただし、ABS用ソフトウェアの特徴である大域変数を介したデータフローに基づくリンクについては、参照中のモジュールと変数名から、データフローの上流や下流のモジュールを得るコマンドが未実装なので、対応できていない。

### 5 まとめ

ECU組込みソフトウェア開発を支援するため、文書参照時にその文書に関係のある文書やツールへのリンクづけを行う文書管理モデル *Liaison-model* を考案した。また、*Liaison-model* に基づく開発環境について、プロトタイプによって有用性を確認した。

今後は、データフローに基づくリンクの実現など、実用化を目指したシステム開発を行う予定である。

### 参考文献

- [1] 鱒坂恒夫, 沢田篤史, 満田成紀: Emerald PCTE, コンピュータソフトウェア, Vol.10, No.2, pp.65-77 (1993).
- [2] F.Long, E.Morris: An Overview of PCTE: A Basis for a Portable Common Tool Environment, CMU/SEI-93-TR-1 (1993).