

携帯端末向けファイルアクセスソフト自動合成環境 (3)

3S-9

ドキュメント生成プログラムの自動合成

工藤 智広 佐藤 明良 山之内 徹

NEC C&C 研究所

1 はじめに

本稿では、携帯端末向けファイルアクセスソフト自動合成環境 POT-DB [1] [2] に関して、POT-DB を構築する際のドキュメント生成プログラム (ドキュメンタ) の自動合成について述べる。ドキュメンタでは、ドキュメントを既存の形式に合わせることや見やすさなどの観点を満たす必要があることなどのため、カスタマイズ要求が多く発生する。したがって、開発を効率よく行えることが必要である。

2 本自動合成方式の概要

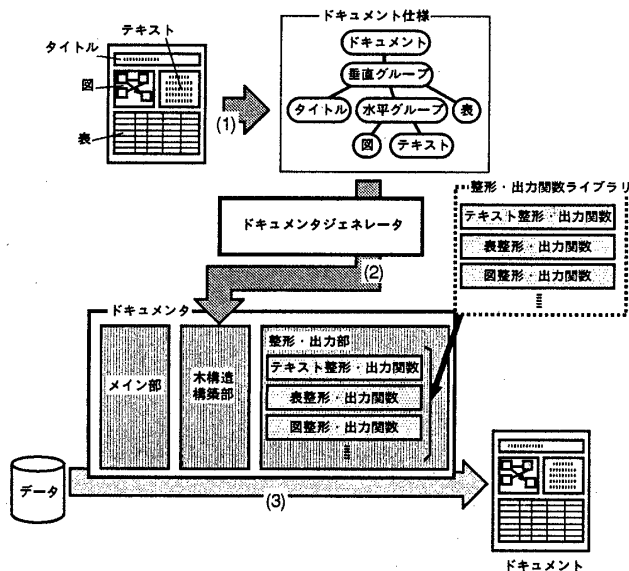


図 1: ドキュメンタ自動合成の概要

本自動合成方式 (図 1) では最初に対象とするドキュメント形式 (フォーマット) をドキュメント仕様として記述する (図 1 の (1))。ここで、ドキュメント仕様は各ドキュメント構成要素を葉ノードとし、これらの要素群を配置方法にしたがってグループ化して組み上げた木構造である。

Software Synthesis Environment for Portable Terminal File Access Programs (3) - Synthesis of Document Generating Programs

Tomohiro Kudo, Akiyoshi Sato and Toru Yamanouchi
NEC Corporation.

続いて、このドキュメント仕様を入力としてドキュメンタジェネレータを実行し、ドキュメンタを出力として得る (図 1 の (2))。得られたドキュメンタは指定したドキュメント形式専用のものであり、表示すべきデータを入力として与えるとドキュメントを出力 (印刷) する (図 1 の (3))。

本方式では、ドキュメンタを以下のように特性によってモジュール分割し、自動合成と部品化再利用を適用することで、新規ドキュメンタ作成時に必要なコード記述量の削減を図った。

- (1)ドキュメント形式によらず共通な部分。全てのドキュメンタでそのまま利用する。
- (2)ドキュメント形式ごとに常に変化する部分。自動合成ツール (ドキュメンタジェネレータ) でドキュメント形式の定義から自動合成する。
- (3)図や表などドキュメント構成要素に依存する部分。ドキュメント構成要素に対応する部品を用意しこれらを組合せることでこの部分を構成する。

なお、ドキュメンタジェネレータの開発にはソフト自動合成シェル SOFTEXSHELL[3] を利用した。ドキュメンタジェネレータは現状では Windows NT 上で動作し、C++ 言語で記述したドキュメンタを自動合成する。

3 ドキュメンタの構成

ドキュメンタは次の各部によって構成されている。

メイン部: ドキュメンタのメイン処理を行う部分。まず木構造構築部を呼び出してドキュメントの木構造を構築した後、この木構造をパラメータとして整形・出力部を呼び出してドキュメントの整形・出力処理を行う。ここはドキュメント形式によらず、すべてのドキュメンタに含まれる (2 の (1) に対応)。

木構造構築部: ドキュメント仕様で指定した木構造をメモリ上に構築する部分。ドキュメント形式ごとに常に変化する (2 の (2) に対応)。ドキュメンタジェネレータはこの部分を自動合成する。

整形・出力部: 木構造構築部できあがった木構造を参照してドキュメントの整形処理 (線や文字等描画要素の位置決定) および出力処理を行う。整形

出力部はドキュメント構成要素に対応した部品(整形・出力関数)によって構成する(2の(3)に対応)。部品としてはテキストや表などの汎用的なものをライブラリとして用意している。

なお、整形・出力関数では、ノードが木構造のどこにあっても部品として利用可能にするため、レイアウト空間上の相対的な座標系を利用して整形・出力処理を行っている。

4 POT-DB 開発への適用

4.1 適用対象

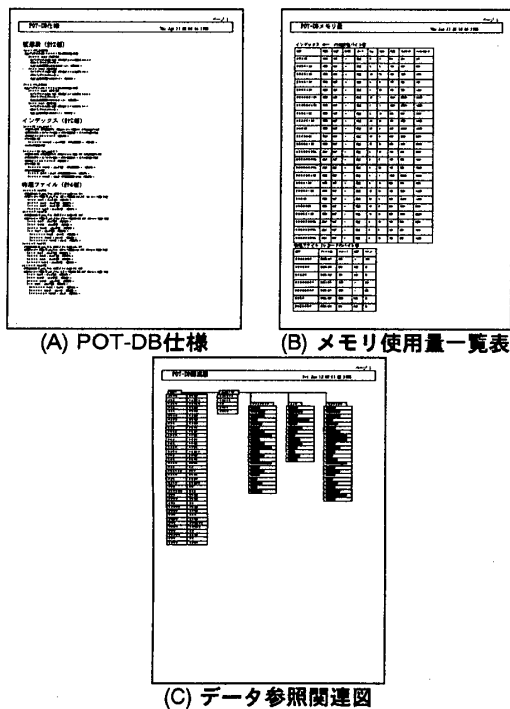


図 2: POT-DB ドキュメントの例

POT-DB では図 2 の 3 種類のドキュメントに対するドキュメンタを開発した。これはそれぞれ、POT-DB 仕様 (A)、インデックスメモリ使用量一覧表 (B)、データ参照関連図 (C) を表したものである。

4.2 評価

図 3 の表に、ドキュメンタジェネレータを用いた開発結果 (A) と、同じアーキテクチャですべてハンドコーディングしたと仮定した場合 (B) とのコード記述量比較を示す。(A) と (B) の両者を比較すると、合計値で A が B の約 1/8 となっており、ドキュメンタジェネレータによってコード量の削減が実現されていることがわかる。

このうち、ドキュメンタジェネレータが行う自動合成では 329 line (ドキュメント仕様分) → 268 line (木

モジュール名	ライン数	A	B	記述言語
メイン部	4545	—	○	C++言語
ドキュメント仕様 (3種類合計)	268	○	—	SOFTEXSHELL(項形式)
木構造構築部 (3種類合計)	329	—	○	C++言語
既存整形・出力関数 (部品数11)	974	—	○	C++言語
新規整形・出力関数 (部品数1)	506	○	○	C++言語
合計		774	6354	

図 3: 記述ライン数

構造構築部分) と 20% 減の削減に留まっている。これは、POT-DB におけるドキュメント形式があまり複雑でなく木構造のサイズがそれほど大きくなかったことによる

一方の部品化・再利用では、常に共通であるメイン部を入れれば 6025 line (=4545+974+506) → 506 line と 1/12 の削減、メイン部を除けば、1480 line (=974+506) → 506 line と 1/3 の削減となった。

その他、今回の適用により次のことを確認した。

- 新規ドキュメンタ開発の際、既存部品で済む場合は、プログラミング言語やライブラリ、出力方式といった複数の知識を知らなくても、ドキュメント仕様の記述方法だけを知るだけでドキュメンタの開発が可能であった。
- ドキュメンタをカスタマイズする場合、プログラムを直接変更するのではなくドキュメント仕様レベルで変更を行うため、変更箇所を発見しやすかった。

5 おわりに

本稿では、ドキュメンタジェネレータを利用した自動合成と整形・出力関数の部品化・再利用とを行うドキュメンタ自動合成方式について、POT-DB のドキュメンタ開発に適用した事例について述べ、その結果、開発の効率化が図れたことを示した。

今後は、本方式の適用範囲や適用の際の問題点を明確化して本方式をより洗練していくとともに、適用事例を増やすことで自動合成、再利用による効果をさらに本格的に実証していく予定である。

参考文献

- [1]三木、佐藤、山之内：携帯端末向けファイルアクセスソフト自動合成環境(1) - プロトタイピングを可能にする統合開発環境、情処第 52 回全大,3S-7,1996
- [2]友部、三木、佐藤、山之内：携帯端末向けファイルアクセスソフト自動合成環境(2) - 自動合成シェルを用いた統合開発環境の構築、情処第 52 回全大,3S-9,1996
- [3]Yamanouchi, Sato, Tomobe, Takeuchi, Takamura and Watanabe: Software Synthesis Shell SOFTEX/S,7th KBSE Conf., 1992.