6F−1

ソフトウェア自動合成シェルSOFTEX/S (1)
−設計思想とシステム構成−

山之内 徹＊　佐藤 明良＊　山田 純夫†　渡辺 正信＊　岡 浩†
NEC　＊C&Cシステム研究所　†C&C CASE技術本部

1 はじめに

ソフトウェアの自動合成には種々のアプローチがあるが、実用化されているものには、適用領域を狭く限定した専用の自動合成システムが多い[1][2]。本論文では、ソフトウェア自動合成を実用化し、これを通してソフトウェア開発を効率化するために、このような専用の自動合成システムを短期間に次々と構築可能とする、ソフトウェア自動合成シェルSOFTEX/Sを提案し、特にその設計思想、システム構成、これまでに得られた評価結果について述べる。

2 システムコンセプト

ソフトウェア自動合成シェルSOFTEX/Sの目的は、ソフトウェア自動合成技術を実用化し、現実のソフトウェア開発における生産性、品質を向上させることである。

ソフトウェア自動合成システムは、その適用対象の広さから、汎用のシステムと、特定分野を対象とした専用のシステムに分類することができる。ソフトウェア自動合成の目的は、ソフトウェアの品質、生産性、保守性を向上させることにあるため、適用対象が広く、かつ、適用時の生産性向上効果の大きいものが理想である。一方、専用の自動合成システムは、対象分野の計算モデルを仕様記述の枠組に組み込むことが可能なため、一般に、仕様記述のレベルを上げることが容易で、生産性向上効果が汎用なものよりも大きい。したがって、ソフトウェア自動合成システムを開発する立場からは、汎用か専用かという選択は、システムの適用範囲と適用した時の生産性向上効果のトレードオフの選択ととらえることができる。

企業における多くのソフトウェア技術者は、ある特定領域のソフトウェアを繰り返し開発することが多い。このような個々のソフトウェア技術者を利用者としてとらえた場合、汎用のシステムでも専用のシステムでも、これを利用するためには、自動合成への入力を記述するための仕様記述言語を1つ覚える必要がある。つまり、自動合成システムを導入する際の負担は同等で、利用した際の生産性向上効果が大きい分だけ、専用のシステムの魅力が大きい。

個々の利用者に大きなメリットを与えようとすると、専用のシステムを開発する必要があり、全体として大きな効果を上げようとすると、多くの領域で利用者を増やしたい。そこで、特定領域向けのソフトウェア自動合成システム(以下、専用ジェネレータと呼ぶ)を短期間に次々と構築することを可能とする環境として、ソフトウェア自動合成シェルSOFTEX/Sを提案する。SOFTEX/Sは、個々の専用ジェネレータに共通して必要となる機能を持つと同時に、共通でない部分の構築を支援する機能を持つため、共通部分の再利用と、非共通部分の効率的な構築で、短期間に専用ジェネレータを構築できる。

3 設計思想

実用的な専用ジェネレータを短期間に構築する環境として、以下の要件を満たす必要がある。

- 対象領域毎に専用の仕様記述言語を容易に実現できる。
- 構文的にも、意味的にも、仕様の内容と等価なプログラムを合成しやすく、また、そのことを確認しやすい。
- 出力プログラムの実行時性能を段階的に最適化しやすい。
- 入力仕様形式や出力プログラム構造の変更に対し、容易に専用ジェネレータを変更できる。
- 構築された専用ジェネレータは、高速に実行できる。

これらの要件を満足する合成手法として、仕様記述を変換規則で部分的かつ段階的に書換えていく、変換アプローチを採用した。特に、豊富な記述力よりも、仕様と等価なプログラムの合成を確認しやすいことを優先し、理論的な扱いが比較的容易な項書換えシステムを採用した。また、対象領域毎に仕様記述言語の構築を支援する枠組として、プログラミング言語C++を拡張し、C++にワイドスペクトラムな仕様記述言語を実現するための言語DSL/C++を定義した[3]。

4 システム構成

SOFTEX/Sは、個々の専用ジェネレータに共通する機能として、仕様記述言語と書換え規則を定義する言語DSL/C++、書換え規則コンパイラ、書換え実行時処理系を持ち、共通でない部分の構築を支援する機能として、個々の書換え規則の構文的等価性を検証する書換え規則検証システムと書換え規則全体制御の設計を支援する書換え規則解析システムを持つ。図1にこの構成図を示す。

DSL/C++は、プログラミング言語C++に以下の4つの機能を追加したものである。

1. 構文定義機能 (TERMDEF文)
2. 型定義機能 (SORTDEF文)
3. 書換え規則定義機能 (RELATION文)
4. 変数定義機能 (VARDEF文)

1と2の機能は、対象領域毎の入力仕様を指定する仕様記述言語のシンタクスを定義するためのもので、3と4の機能は、仕様記述言語の各構文毎に、CまたはC++プログラムの要素に段階的に書換えるための書換え規則を定義するためのものである。仕様記述言語のセマンティクスは、この書換え規則を通して定義される。

Software Synthesis Shell SOFTEX/S (1) –Design Philosophy and System Organization –
Toru YAMANOUCHI, Akiyoshi SATO, Sumio YAMADA, Masanobu WATANABE, Hiroshi OKA
NEC Corporation

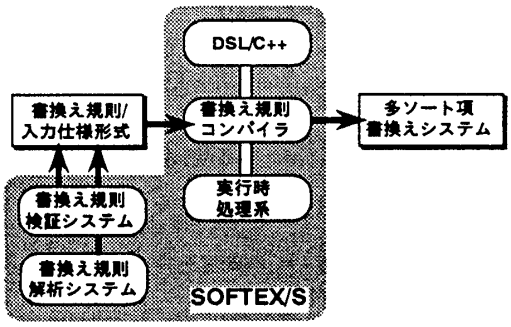

図 1: SOFTEX/S 構成図

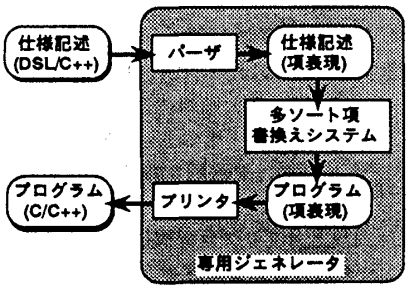

図 2: 専用ジェネレータ構成図

書換え規則コンパイラは、DSL/C++ で記述された書換え規則を入力し、実行時処理系をリンクして、C 言語で実現された多ソート項書換えシステムを出力するものである [5]。

書換え規則検証システムは個々の書換え規則が構文的等価性を満たすかどうかを検証するもので [6]、書換え規則解析システムは、書換え規則間の相互関連図を示し、全体制御の設計を支援するものである。

図 2 に実際のプログラム自動合成時の構成図を示す。プログラム自動合成時には、DSL/C++ で定義した仕様記述言語で書かれた入力仕様は、まず、パーザにより構文解析され、多ソート項で表現される。多ソート項表現された仕様が、書換え規則コンパイラによって生成された多ソート項書換えシステムに入力され、多ソート項表現された C または C++ のプログラムが出力される。仕様記述言語のパーザと C/C++ のプリンタは、DSL/C++ 言語の処理系として SOFTEX/S によって提供される。このプログラムがプリンタに入力され、C/C++ の構文を持ったプログラムとして出力される。パーザとプリンタが DSL/C++ で定義した仕様記述言語上の変換と、多ソート項集合上の書換えが同型であることを保証する [4]。

5 評価

5.1 自動合成シェルの評価

SOFTEX/S を利用して、専用プログラムジェネレータを構築する際の生産性向上効果を評価する目的で、表 1 に示す専用ジェネレータを構築した。SOFTEX/S を利用せずに、同機能の専用ジェネレータを構築するには、最低でも 2 ～ 3 人年の工数が必要と見積られ、SOFTEX/S の専用ジェネレータ構築における高い生産性が示されたと考える。

表 1 SOFTEX/S を利用した専用ジェネレータの開発

入力仕様	書換え規則数	開発工数
状態遷移表	178	1 人月
SDL 仕様	280	1 人月
交換機サービスシナリオ	387	3 人月
RPC 仕様 [7]	420	3 人月

5.2 専用ジェネレータの評価

上記専用プログラムジェネレータのうち、状態遷移表から C プログラムを合成するものについて、この領域の技術者による評価をおこなった。この評価では、同一の状態遷移表から、人手によるコーディングと専用ジェネレータによる自動合成を行ない、両者を比較した。まず、人手によるコーディングは、デバグを含めて 30 人日ほどの工数を要したが、自動合成では 1 人日で開発できることがわかった。1 人日の工数の中には、標準的でないコーディングを出力するために必要であった、一部書換え規則のカスタマイズ工数を含む。このことは、専用ジェネレータのターゲットソフトウェアの生産性向上効果が高いこと、出力プログラム構造の変更に対する書換え規則の変更が容易なことを示している。次に両者によって得られたプログラムの品質を評価した。この結果、両者の実行速度、メモリ使用量の両面で、ほぼ同等であり、実用上問題が無いことがわかった。

6 おわりに

本論文では、限定された領域に専用のプログラムジェネレータを短期間に次々と構築可能とする、ソフトウェア自動合成シェル SOFTEX/S を提案し、専用ジェネレータ開発事例を通して、その実用可能性を示した。今後の課題を以下にあげる。

1. さらに多くの専用ジェネレータを構築し、本手法の有効範囲を明確にする。
2. 専用ジェネレータ構築のコストを含めたトータルなソフトウェア開発効率化効果を評価する。
3. SOFTEX/S を用いて専用ジェネレータを構築する際の方法論をまとめる。

参考文献

[1] Kant, E. : Knowledge-Based Support for Scientific Programming, *7th KBSE Conf.*, 1992.

[2] Graves, H., Louie, J. and Mullen, T., : A Code Synthesis Experiment, *7th KBSE Conf.*, 1992.

[3] Yamanouchi, T., Sato, A., Tomobe, M., Takeuchi, H., Takamura, J. and Watanabe, M.: Software Synthesis Shell SOFTEX/S, *7th KBSE Conf.*, 1992.

[4] 佐藤, 山之内, 渡辺、ソフトウェア自動合成シェル SOFTEX/S(2) - 文脈自由言語と多ソート項集合の同型写像 -、第 46 回情報処理学会全国大会。

[5] 友部, 山之内, 渡辺、ソフトウェア自動合成シェル SOFTEX/S(3) - 項書換え規則の直接実行方式とその高速化 -、第 46 回情報処理学会全国大会。

[6] 三木, 佐藤, 山之内, 渡辺、ソフトウェア自動合成シェル SOFTEX/S(4) - 項書換え規則の互換性検証システム -、第 46 回情報処理学会全国大会。

[7] 三木, 桐葉, 山之内、拡張容易性を有する分散管理アプリケーションジェネレータ、1993 年電子情報通信学会春季大会。