

データ駆動パラダイムによる図的仕様記述体系：AESOP

4M-8

岩田 誠 寺田 浩詔
(大阪大学 工学部 情報システム工学科)

1 はじめに

本研究は、自然な要求仕様記述法として、多様な半形式的図的記述を採用し、以後データ駆動パラダイムに基づいて、これらの記述を実行可能プログラム生成の水準まで一貫的に保存することによって、要求定義水準でのソフトウェアの継承・保守を可能とする体系の確立を目指している。

本稿では、この一つの試みである図的ソフトウェア仕様記述体系 AESOP (Advanced Environment for Software Production) の構想ならびに、開発中のプロトタイプシステムの構成手法を述べる。

2 データ駆動パラダイムの特質

ソフトウェア仕様化に際しては、将来的な要求変更にも柔軟に適應できるよう、システム全般に渡る大局的観点から決定された機能分担やそのインタフェースを定義した要求仕様記述を源として、これから継承されたソフトウェア仕様記述に、さらにアルゴリズムやデータ構造を階層的かつ段階的に付加して、実行可能なプログラムの水準まで一貫して次第に洗練化できる環境が準備されることが重要である。当然、これらの仕様記述に、そのシステムの実現法が当初から全面的に反映することは決して望ましいことではない。

本研究はこのような環境の一実現法として、システムの発注者側にも容易に理解可能な図的表現を許す、ソフトウェア仕様記述手段をまず導入し、更にこの記述から、安全な高度並列処理プログラムを機械的に生成する一貫した体系を提供しようとするものである。

本仕様記述体系の基礎に採用しているデータ駆動原理は、並列処理方式の基本原則として有望なだけでなく、システムのモデル化手法としても、自然な処理パラダイムを提供する。例えば、これまでに、要求仕様定義水準では、データフロー的なパラダイムを用いて仕様記述する多くの試みがなされている [1]。また、プログラム水準では、データ依存関係による入出力の明示的表現能力を活用して、徹底した部品化手法によるソフトウェア生産・保守が試みられている [2]。さらに、実行機械の水準でも、トークンの追い越しをも許

す動的データ駆動処理方式の処理装置の試作が、筆者らを含めて各所で進められている [3]。

このような環境を統合的に活用するために、仕様記述体系にデータ駆動パラダイムを一貫して適用すれば、要求定義水準の図的仕様記述がほぼ同型のまま実行可能プログラムに変換されるため、ソフトウェア仕様記述水準でデータ駆動型プログラムの持つ検証性・部品化能力を利用でき、かつ、プロトタイプをデータ駆動型プロセッサシステム上で高度並列に実行できる体系の実現が可能である。

3 AESOP のソフトウェア仕様記述体系

本仕様記述処理体系では、図1の概要図に示すように、要求定義水準の記述に用いられた半形式的な図的記述の情報をソフトウェア仕様水準へ効果的に継承するために、複数の半形式的図的表現によって多面的かつ階層的な要求仕様記述を許している。プログラム仕様記述の水準では、これらの記述のうちソフトウェア実現される部分の記述をそのまま継承し、これに実現手法上必要となる、データ構造とその処理アルゴリズム

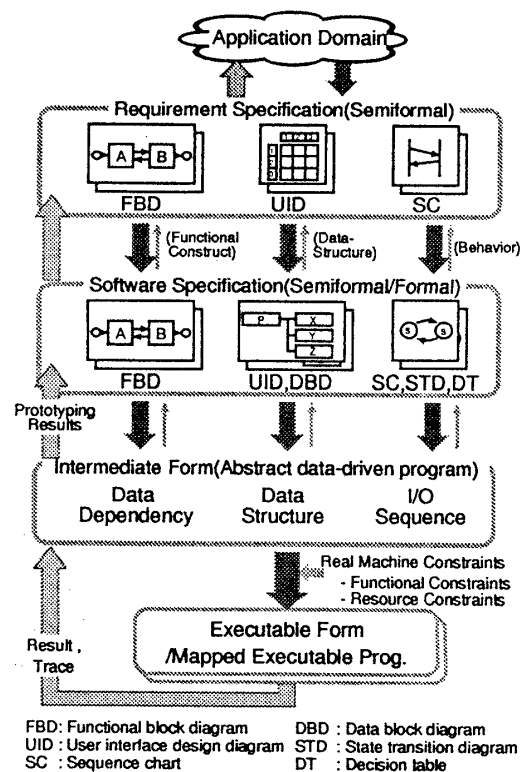


図1: AESOP のソフトウェア仕様記述処理体系

ムを加えて、漸次階層的に詳細化を繰り返して、抽象的なデータ駆動型プロセッサ上で実行可能なプログラムを対話的に生成する。この中間的なデータ駆動型プログラム ADP (Abstract Data-Driven Program) の水準は、要求仕様から継承したソフトウェア仕様に含まれるアルゴリズムとデータ構造を表現し、この水準で論理的検証や部品の蓄積を行なう機能を実現している。さらに、本体系では、データ駆動型プロセッサシステム Qv-x[3] を実行機械に想定しているが、それらの機能的制約 (Functional Constraints) や資源制約 (Resource Constraints) を考慮した実行可能プログラムを自動生成する方式をとっている。この Qv-x 上でのプロトタイプ実行の状況や結果は、その逆順に変換され、図的仕様記述上に多面的に視覚化される。

4 AESOP 体系の一実現法

本体系の実現にはまず、図的ソフトウェア仕様記述の表現形式の選択、および、これらの抽象データ駆動型プログラムへの変換手法が重要な検討課題となる。

ソフトウェア仕様記述の図的表現形式に関しては、現在のところ、図 1 上部に示すように、(1) 機能ブロック図 FBD のように、システムの当面の機能とその継承・展開を想定した図的表現、(2) このシステムに対する外部データ構造への要求を示す、ユーザ・インタフェースの図的表現 UID、(3) システムに対する入・出力相互間の時間的関係を示す、シーケンス図 SC が、宣言的なシステム要求定義の水準の表現としても有望であるとの見通しを得ている。また、これらの表現形式を用いて定義されたソフトウェア仕様記述をプログラム仕様記述の水準まで洗練化するには、より形式的なデータブロック図 DBD や状態遷移図 STD などの図的表現の導入が効果的であるとの結論を得ている [5]。

一方、これらの図的仕様記述の変換手法に関しては、純粋なデータ駆動原理では扱えない履歴依存性のある処理を規定できる、拡張データ駆動型処理モデルを導入して、このモデルに基づく抽象データ駆動型プログラムを生成する手法の検討が必要となる。

現在のところ、アルゴリズムに関する履歴 (状態) とデータ構造に関する履歴 (ファイル) に依存する処理をそれぞれ規定できるよう拡張した、抽象データ駆動型処理モデルを提案している。また、この拡張した処理構造部分をブラックボックス化して、Well-behaved 性を保存する形態で抽象データ駆動プログラムの生成・プロトタイプ実行を行なう手法が、変更・保守に対するロバスト性のあるソフトウェア仕様記述環境を提供する上で重要であるとの見通しを得ている [7]。

この拡張した抽象データ駆動型プログラムの構成要素を図的仕様記述から生成する手法に関しては、仕様記述の記述能力や抽象データ駆動型処理モデルの将来的な拡張性を残すために、プリミティブな図的記述要素に対する生成規則を定め、これらの規則により生成されたプログラム構成要素を、その時点で既に生成されているプログラム情報に統合する手法を採用した [4]。この生成手法では、個々のプログラム要素への一対一変換により、抽象データ駆動型プログラム構造が直接的に生成されるために、いわゆる、加法的変換が可能であり、また、各生成規則の逆変換規則を定義しさえすれば、別項に述べる多面的仕様記述間の相互変換が容易に実現できる見通しが得られている [6]。

5 おわりに

本報告では、図的ソフトウェア仕様記述体系 AESOP の構想と開発中のプロトタイプシステムの構成手法を述べ、その設計思想を明らかにした。

本手法は、ソフトウェアに限らず、ハードウェア機能の仕様記述環境にも原理的に適用可能であるため、ソフトウェア・ハードウェアを含めた一貫したシステム仕様記述環境の一つの基本的な枠組を提供する。

AESOP の検討の詳細は別稿 [5]~[7] に述べるが、今後は、さらに高い水準の仕様記述を可能にするため、各応用分野に応じた部品の蓄積ならびにその再利用の効果的な支援環境を構築し、実用的な応用分野への適用とその評価を行なう必要がある。

謝辞 御指導・御支援頂いた AESOP 研究の関係各位ならびに寺田研究室の各位に深く感謝する。なお、本研究の一部は、文部省科研費 (一般 B-2 05452363, 試験 B-1 06555110) の援助を受けている。

参考文献

- [1] D.T.Ross: *Structured Analysis(SA): A Language for Communicating Ideas*, IEEE Trans. on Softw. Eng., Vol. SE-3, No.1, pp.16-34 (1977).
- [2] H.Shirasu: *Innovative Approach to Switching Software Design Using Dataflow Concept*, ISS, S-B 4.3 (1987).
- [3] H.Terada, M.Iwata et al.: *Superpipelined Dynamic Data-Driven VLSI Processors*, Proc. ACM ISCA '92 Workshop on Data-Flow Computing, (to be published).
- [4] 岩田, 寺田: 図的仕様記述からのデータ駆動プログラムの生成手法, 情報処理学会計算機アーキテクチャ研究会, 94-ARC-107-8, pp.57-64 (1994).
- [5] 笠原 他: AESOP の多面的な図的仕様記述手法, 第 49 回情報処理学会全国大会, 4M-9 (1994).
- [6] 唐沢 他: AESOP における仕様記述相互間の変換手法, 第 49 回情報処理学会全国大会, 4M-10 (1994).
- [7] 種田 他: AESOP におけるプロトタイプ手法, 第 49 回情報処理学会全国大会, 4M-11 (1994).