高信頼性ソフトウェアをめざして

研究ノート

谷口健一*，東野輝夫**

Toward High Reliable Software

Key Words : high reliable software, software engineering, formal description technique

1．はじめに

仕様を満たしているという意味で正しいことが保証されたソフトウェアを作成することは，ソフトウェア科学分野における重要な研究テーマの一つである．高信頼性ソフトウェアの開発手法としては，(1) 形式記述技法を用いて仕様やプログラムをモデル化し，数学的な論法を用いて，開発したプログラムの正しさを証明する方法や，(2) 抽象的な仕様と詳細化のための条件から下位レベルの仕様あるいはプログラムを機械的に合成する方法，(3) 一定の条件を満たすプログラムに対して，組織的なテストやデバッグ手法を用いてプログラムを検査する方法，などが研究されている．ここでは，高信頼性ソフトウェアの設計開発について，我々の行っている研究を紹介する．

2．高信頼性ソフトウェアの設計開発法

正しさの保証されたプログラムを作るためには，もととなる仕様を形式的に記述し，それらを段階的に詳細化してプログラムを得る，そして，詳細化の正しさを数学的に証明するという手法が考えられる．そのような手法の一つとして代表的手法がある．代数的手法では，仕様やプログラムを等式（公理と呼ぶ）の集合で表現する．各公理は，入力と出力などの間で成り立つべき性質を表したり，計算したい式やその途中結果を表す関数の値を定義する．

一般に上位レベルの仕様では，導入した関数の間で成り立つべき性質のみを記述し，詳細化の各段階でそれらの値をどのように計算していくかを順次指定していく．この過程で，上位レベルの各公理の性質が，下位レベルの仕様上でも成り立つことを数学的に証明することにより，詳細化の正しさを保証する．一般には検証者が証明の筋道を考案する必要があるが，証明の各過程は，公理による式変形や論理式の判定，場合分けなど多くの部分で機械化や計算機による支援が可能である．このため，有効な証明支援環境を構築することが，正しいことが保証されたソフトウェアを作成していく上で重要である．また，コンパイラでプログラムを解析し，効率のよい目的コードを自動生成することが望ましい．

我々の研究グループでは，代数的手法を用い

* Kenichi TANIGUCHI
1970年大阪大学大学院基礎工学研究科博士課程修了
現在，大阪大学大学院基礎工学研究科情報数理系専攻，教授，工学博士，形式記述技法
TEL 06−850−6605
FAX 06−850−6609
E-Mail taniguchi@ics.es.osaka-u.ac.jp

**Teruo HIGASHINO
1984年大阪大学大学院基礎工学研究科博士後期課程修了
現在，大阪大学大学院基礎工学研究科情報数理系専攻，助教授，工学博士，分散協調システム
TEL 06−850−6607
FAX 06−850−6609
E-Mail higashino@ics.es.osaka-u.ac.jp

て正しさの保証されたソフトウェアを開発する際の設計法及び証明の計算機支援の方法，コンパイラの作成法などの検討を行い，支援系やコンパイラの実現とそれらの評価を行っている．本研究グループではASLと呼ばれる代数的言語を開発し，その形式的な意味定義やASL上でのソフトウェアの仕様記述法，プログラムへの詳細化法や詳細化の正しさの証明法，効率的な目的プログラムの生成法などを検討し[1]．証明支援系やコンパイラなどを含むASLプログラム開発システム（ASLシステム（図1参照））を試作し，一般に公開している．また，抽象レベルの仕様からプログラムへの詳細化とその正しさの証明を行うための実用的な例題として，スクリーンエディタや在庫管理問題などを取り上げ，実際に詳細化と正しさの証明[2]，開発したコンパイラを用いた目的コード（Cプログラム）への変換などを行っている．目的コードには既存のCプログラムとのリンク機能があり，C言語用ライブラリ群などとのリンクが容易に行える．これらの実験例や以下で述べるCPUなどのハードウェアの設計/開発などでは，実用的な時間で詳細化や証明が行えることが確かめられている．また，得られた目的コード（Cプログラム）は，同じアルゴリズムのプログラムを人手で工夫しながら直接Cで書き下した場合と比べて，遜色のない程度のスピードで実行されることも確かめられている．

3．ハードウェアの高位レベル設計法

上で述べた高信頼性ソフトウェアの設計法はハードウェアの高位レベル設計（回路の機能を記述した要求仕様レベルから論理設計レベルに至るまでの部分）にも応用可能である．下流工程での誤り修正には多大なコストがかかるため，高位レベル設計の段階で回路設計の正しさを保

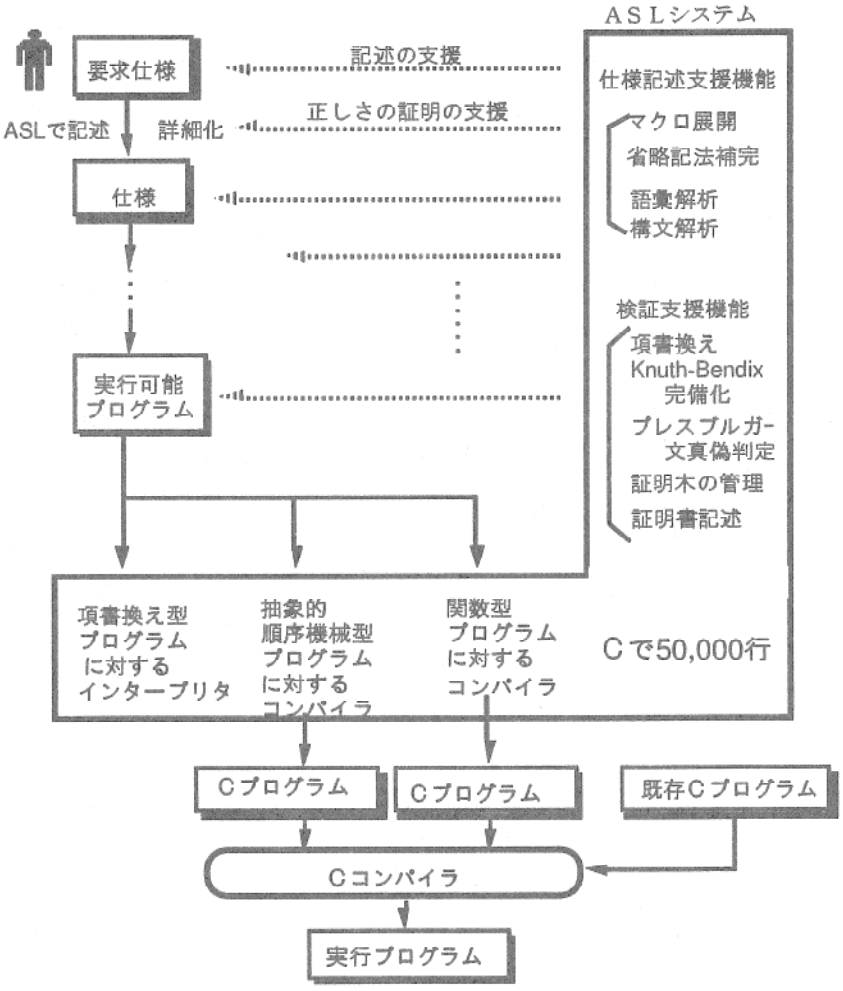

図1　ASLシステムを用いたプログラム開発

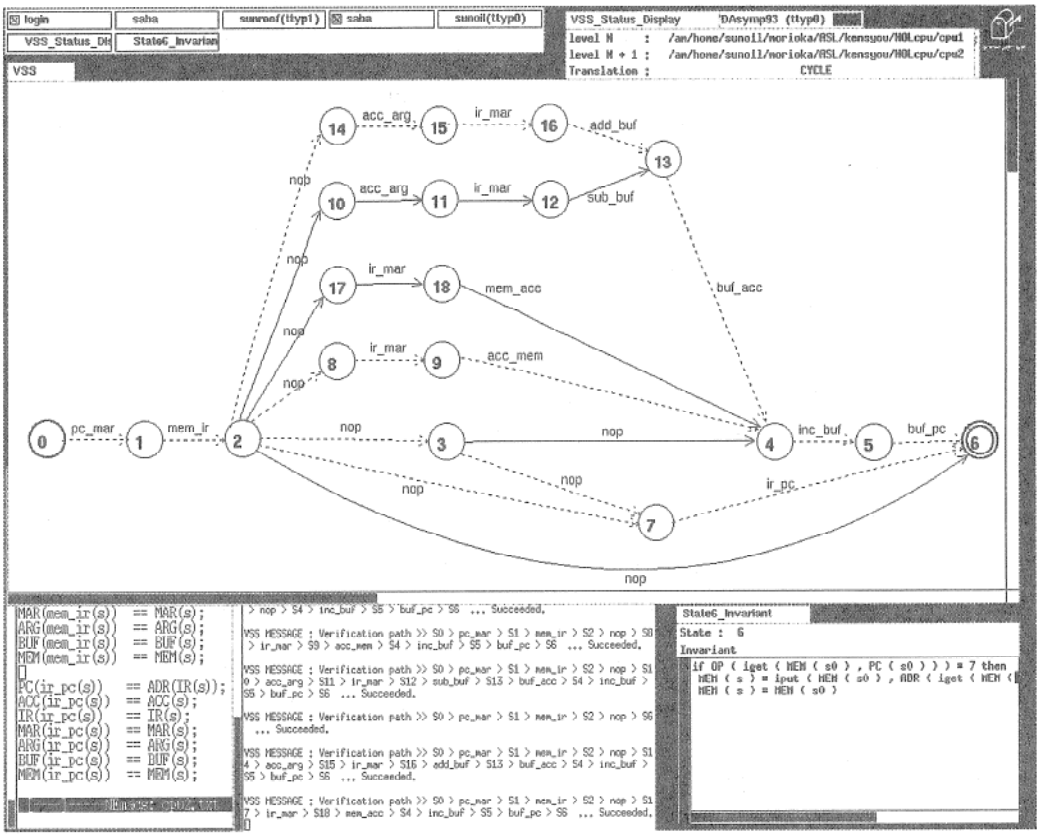

図2　回路設計検証支援システムの使用中画面

証できることが望ましい．我々の研究グループでは代数的手法を用いた回路設計や検証の方法，支援システムなどについて研究している．

設計は段階的に行う．例えばCPUでは，要求仕様には，各命令ごとその実行によって各レジスタやメモリがどのように変化すべきかを等式（公理）で記述する．下位レベルでは，より具体的な処理内容を持つ遷移を導入し，もとの記述の各遷移をどのような遷移系列で実現すべきかを指定することによって具体化する．この手順を繰返し，アーキテクチャ，部品の接続関係，各部品の制御論理などを決めることにより，最終的に論理設計レベルの回路記述を得る[3]．

設計の各段階ごと，その具体化の正しさを代数的手法により証明する．ハードウェアの設計では，ソフトウェアの場合よりも実現すべき機能が比較的単純で，実現方式もある程度固定されているので，証明作業をより自動化できる可能性がある．我々は，ハードウェア向けの半自動証明手順を検討し，証明支援システムを作成している（図2参照）．また，回路性能の向上のための状態図の等価変換支援系を作成している．得られた論理設計レベルの回路記述からNTTが開発した回路自動合成システムPARTHENONを用いてネットリストを自動合成できる．これまで，TPCD'94（回路の証明系に関する国際会議）のベンチマーク用CPUやKUE-CHIP2などのCPU，ソーティング回路，基本的なパイプライン方式CPUなどについて，数分から数時間程度という実用的な時間で，自動で証明を行えることが確かめられている[4]．

4．協調分散システムの設計開発法

一般に分散システムを設計する場合，各ノード（計算機）間でどのようなメッセージ交換をする必要があるかを設計者自身が記述するのは非常に繁雑であり，間違いも起こりやすい．分散システムの設計法として，設計者は分散システム全体の処理内容やそれらの実行順序のみを

要求仕様（サービス仕様）として記述し，その仕様から，各ノードがどのようなタイミングでどのノードとどのようなメッセージを交換しながら自ノードの処理を実行すべきかを記述した動作仕様の組（プロトコル仕様）が自動生成できることが望ましい．

我々の研究グループでは，拡張有限状態機械，レジスタ付きペトリネット，並行分散システムの形式仕様記述言語LOTOSなどで書かれた分散システム全体のサービス仕様から，サービス仕様通りに動作するプロトコル仕様を自動生成するためのアルゴリズムを考案し，その処理系の開発を行っている．また，リアルタイムシステムなどでは，各処理の実行開始時刻間の関係などに関する「時間制約」が付加されている場合が多い．このような時間制約付きの要求仕様を並行プログラムや回路として実現しようとする場合，実際のハードウェアや通信回線の性能などの制約からもとの要求仕様に書かれた時間制約を変更する必要が生じる場合がある．しかし，時間制約を変更すると，もとの要求仕様では実行可能であった処理系列が実行不可能になったり，逆にもとの要求仕様では許されない処理系列が実行可能になるなど，変更の前後の仕様の等価性（双模倣等価性）が保存されない恐れがある．本研究グループでは，LOTOSに時間制約を記述する機能を付加した新しい言語LOTOS/Tを考案し，その言語で書かれた二つの仕様の等価性を機械的に効率よく判定するアルゴリズムの考案などを行っている．さらに，時間制約付きのサービス仕様からプロトコル仕様を自動生成するためのアルゴリズムの考案なども行っている[5]．

このような並行システムのプログラム（並行プロセス）には，プロセス間の同期（マルチランデブー），選択，割込，並列実行が多数含まれる場合が多い．従って，多数の並行動作の実行管理や階層的な同期，選択，割込処理の実現などが高速に動作するプログラム群を生成するためのポイントとなる．我々は，プロセス間通信を高速に行えるようにするため，BSD UNIX上で動作する移植性のよいマルチスレッド機構（一つのユーザプロセス内に複数の並列処理単位を生成し高速に実行する仕組み）を開発し，PTL（Portable Thread Library）として一般に公開している．また，このマルチスレッド機構を用いて実行効率の高い目的プログラムを生成するLOTOSコンパイラなどを作成している[6]．異なる計算機間でのプロセスの同期の実装法などについても研究を行っている．

5．ま　と　め

高信頼性ソフトウェアの設計開発は，オンラインシステムや電話の交換システムなどの社会システム，飛行機や人工衛星などのソフトウェア，プラントや発電所などの制御システム用ソフトウェアなど，バグの発生が社会的に大きな影響を及ぼす分野では特に重要である．既存の大規模プログラムが正しいかどうかを検証することは現実には不可能である．しかし，信頼性の高いソフトウェアを組織的に構築していく手法については近年かなり活発に研究されており，実際の実用システムの作成に有効な技術もいくつか得られている．バグのない高信頼性ソフトウェアの設計開発という最終目標に向けて，今後もさらに研究を進めていくつもりである（筆者らの研究室の研究内容については，WWWのページ（http://sunfish.ics.es.osaka-u.ac.jp/）にても情報提供致しております）．

参　考　文　献

1）東野，関，谷口：代数的仕様から関数型プログラムの導出とその実行，情報処理，29-8，881-896（1986）．
2）森岡，岡野，東野，谷口：関数データベースを用いた在庫管理プログラムの記述とその詳細化の正しさの証明，情報処理学会論文誌，36-5，1091-1103（1995）．
3）谷口，北道：代数的手法による仕様記述と設計及び検証，情報処理，35-8，742-750（1994）．
4）Kitamichi，Morioka，Higashino and Taniguchi：Automatic Correctness Proof of Implementation of Synchronous Sequential Circuits Using Algebraic Approach，2nd Int．Conf．on

Theorem Provers in Circuit Design, LNCS-901, Springer Verlag, 165-184 (1994).

5) 中田，東野，谷口：時間制約の記述されたLOTOS仕様からのプロトコル合成，情報処理学会論文誌，37-5，672-686(1996).

6) Yasumoto, Higashino, Abe, Matsuura and Taniguchi : A LOTOS Compiler Generating Multi-threaded Object Codes, 8th IFIP Int. Conf. on Formal Description Techniques, 271-286, Chapman & Hall (1995).