TR-0785

並列プログラムの知的プログラミング環境　MENDELS ZONE

本位田　真一、大須賀　昭彦
内平　直志（東芝）

July, 1992

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

並列プログラムの知的プログラミング環境MENDELS ZONE

本位田真一、大須賀昭彦、内平直志

1　まえがき

近年のハードウェアの進歩に伴い、分散処理システム、並列システムの形態によるアプリケーションが今後増加していくことが予想される。当然のことながら、アプリケーションにおける、高信頼性、高生産性も要求される。しかしながら、並列プログラムの開発には、その正しさの検証やプログラムのデバッグに多大な労力を要する。そのため、高信頼性の保証されたプログラムを作成するための手段の確立が特に必要となる。MENDELS ZONEは、この手段の確立を目指した計算機支援環境である(1)(6)(7)。

2．MENDELS ZONEのねらい

MENDELS ZONEは、作成したプログラムの高信頼性を保証するために、形式的手法を採用している。形式的手法とは数学的枠組みに基づいた手法であり、従来より、信頼性を保証するための有効な手法として知られている。しかしながら、形式的手法を実用に供するためには、次の問題を解決しなければならない。

(1)単一の形式的手法によってシステムの全ての性質を記述することは極めて困難である。

(2)形式的手法においては、プログラムの生成や検証をするためには、多くの時間を要する。

(3)大規模なシステムの記述には不向きである。

(4)仕様記述量が多くなると、理解性が低下する。

(5)仕様の詳細化プロセスが確立していないため、仕様記述には経験を要する。

MENDELS ZONEでは、これらの課題に対して次の様に対処している(2)。

(1)に対して

一般に並列プログラムは同期に関する部分と機能に関する部分から構成される。MENDELS ZONEは同期に関する部分には時相論理、機能に関する部分には等式論理を用いている。この理由は記述する特性に即した論理を採用したためである。

(2)に対して

時相論理、等式論理の検証系である定理証明系の実行には多大な処理時間を要する。そこで定理証明系を並列実行することにより応答時間の短縮を図っている。これは、MENDELS ZONE自身が並列マシン上に実装されていることによる効果である。

(3)に対して

大規模システムを構築するためには部品化再利用技術が必要不可欠である。そこでMENDELS ZONEでも、部品化再利用を採用しており、機能に関する部分を部品

として扱い、同期に関する部分を部品間のメッセージの順序関係として扱っている。すなわち、等式論理の記述から部品を生成し、時相論理から部品間のメッセージの順序関係を生成することになる。

(4)に対して

形式的仕様によって記述された仕様は一般に理解性に乏しいものである。MENDELS ZONEでは、理解性を高めるためおよび記述量を減らすために、種々の図形仕様を併用している。すなわち、等式論理での記述内容を自動的にデータフローダイアグラムで表現することによって、ユーザによる視認性を高めることを可能としている。また、時相論理による記述量を大幅に減らすために、対象システムの構造的な制約を与えるためにペトリネットを併用している。

(5)に対して

形式的仕様で仕様を記述する場合の大きな問題点として、上位レベルの抽象的な仕様からどのような手順で下位レベルの具体的な仕様に展開するかというプロセスが確立していないことがあげられる。MENDELS ZONEは、等式論理による仕様記述において、等式論理とデータフローダイアグラムを組み合わせることにより詳細な仕様展開プロセスを規定した。

以上、述べたようにMENDELS ZONEは、形式的手法を実用に供するための種々の方策を並列プログラムを対象として提案している。MENDELS ZONEの対象言語はMENDELである。MENDELはペトリネットをベースの並列言語であり、並列論理型言語KL1のユーザ言語の一つである。MENDELで記述されたプログラムはKL1に変換される。MENDELS ZONEはMENDELで記述された部品を生成するサブシステムと部品の中身も踏まえた部品間の同期部を生成するサブシステムの2つから構成される。次節以降において、各々について述べる。さらに、MENDELS ZONEのソフトウェア開発環境としての有効性を評価するために、あるアプリケーションを構築したので、その結果についても述べる。

3．部品生成部(Metis)

3.1 MENDEL部品の検証・生成

部品生成部では，等式論理で記述された部品仕様の検証や，仕様からMENDEL部品への変換を行う．検証・変換には，サブシステムMetis(4)を用いる．以下で，MetisによるMENDEL部品の生成過程を説明する．

(1) データフローダイアグラムで記述された部品仕様が，仕様展開部によって，等式論理で記述された仕様へと変換されMetisへ与えられる．

(2) Metisに与えられた部品仕様は，並列完備化によってTRSへ変換される．TRSとは仕様と等価な内部表現で，この変換の際に，仕様記述の曖昧性が取り除かれる．

(3) TRSがインタプリタによって実行される．この結果，部品仕様の正しさが部分的に(つまり，実行したデータについて)確認される．

(4) TRSが並列検証によって検証される．この結果，部品仕様の正しさが全体的に(つまり，検証項目を満たす任意のデータについて)確認される．

(5) 実行・検証によって仕様の誤りが発見された場合，等式論理またはデータフローダイアグラムまで戻って仕様を修正し，再び完備化～検証を行う．この繰返しによって，仕様の正しさが保証される．
(6) 仕様の正しさが確認されたら，TRSをそれと等価なMENDEL部品へ自動変換する．これによってMENDEL部品の正しさも保証される．

3.2 Metisの機能概要

MENDEL部品の検証・生成を支援するMetisの機能について簡単に述べる．部品仕様の完備化には，並列実装した無向完備化手続き(4)を用いる．ここで無向完備化手続きとは、従来の完備化が失敗に終わっていた場合を回避するように手続きを拡張したものである．

仕様の検証には定理の自動証明手続きを用いる．仕様の実行では与えられたデータについてのみ結果が確認されるのに対し，定理証明では検証項目を満たす任意のデータについての正しさが数学的に保証される．特に仕様やプログラムの検証においては，帰納的定理の証明が有効であることが報告されている．そこで，Metisでは，一般的な定理証明(3,5)と帰納的定理証明の2種類の手続きを用意し，検証項目によって適用する手続きを使い分けている．

検証によって仕様の正しさが保証されても，それが部品(プログラム)の正しさに反映されなければ有効な支援とはならない．Metisでは，仕様を等価なMENDEL部品へ自動変換する．この方法では，仕様の正しさと同じ意味での部品の正しさが保証され，また部品の並列実行においても計算の停止性が保証される．

3.3 検証例

図1にエレベータのボタン操作に関する仕様の一部を与える．この仕様が満たすべき最も重要な性質として，「エレベータはボタンを押した階に必ず停止する」ことが挙げられる．この性質は，以下の式で表現される．

動作指示参照(F,ボタン押下げ(F,S))=停止

この式は「任意の状態Sにおいて，F階の停止ボタンが押されたら，F階への動作指示は停止になっている」と読むことができる．これに対して帰納的定理証明を行うと，部品仕様がこの性質を満足することが確認される．

```
object  エレベータ内ボタン操作  has
  ボタン押し下げ(F,init)=add(F,init);
  ボタン押し下げ(F,add(Fc,S))
  = if eq(F,Fc) then add(Fc,S)
                else add(Fc,ボタン押し下げ(F,S));
  動作指示参照(F,init)=通過;
  動作指示参照(F,add(Fc,S))
  = if eq(F,Fc) then  停止  else  動作指示参照(F,S);
end.
```

図1 エレベータ部品の仕様(一部)

4．同期部生成部

ここでは，部品ライブラリに蓄積されたMENDEL部品を検索，結合し，目的のMENDELプログラムを生成する部品再利用方式を示す．ここで，単純に部品を結合しただけのMENDELプログラム（本体部と呼ぶ）は，機能面の仕様は満たしているが，タイミング面に関しては何も考慮されていない．たとえば，デッドロックを起こす危険性がある．そこで，タイミング面の仕様（たとえば，デッドロックフリー）を時相論理で記述し，その仕様を満たすように部品間の同期を制御する部分（同期部と呼ぶ）を定理証明手法を用いて自動生成する．各部品の正当性は既に保証されているので，本体部と同期部を結合してできたMENDELプログラムは，機能面とタイミング面の両方の仕様を満たすことが保証される．生成されたMENDELプログラムは，最終的にKL1コードにコンパイルされ並列マシン上で実行できる．

4.1 本体部生成(6)

MENDEL部品はペトリネットで表現でき，部品結合はペトリネットエディタを用いて実現される．MENDELS ZONEでは，この部品検索，結合フェイズ（本体部生成フェイズ）における支援機能として部品管理ブラウザと部品結合支援サブシステムGARNETを提供している：

(a) 部品管理ブラウザ

各部品の外部インターフェイスの管理および検索，複数の視点からの表示機能を持つ．

(b) GARNET

各部品の外部インターフェイスに持たせたデータ仕様（意味ネットで表現されている）から，インターフェイスの構文的および意味的整合性を考慮してお互いに結合可能な部品候補を選び出す．

4.2 同期部生成(7)

本体部生成フェイズでは，主に部品間のデータの整合性に注目して部品結合を行った．タイミングに関しては何も制約がないので，このままではデッドロック等のタイミング面での不具合を生じる可能性がある．そこで，タイミング面の仕様を時相論理で記述し同期部を生成する．

(a) 時相論理とタブロー法

時相論理は古典論理に時間を扱う時相オペレータを追加した論理である．同期部生成では，時相論理の中でも線形時間時相命題論理（LPTL;Linear time Propositional Temporal Logic）を採用した．LPTLでは以下の仕様記述が可能である：

(1)デッドロックの有無（例：動作ａは無限回生起する）

(2)動作の順序関係（例：動作ａの後には動作ｂが生起する）

(3)動作の禁止（例：動作ａが生起したらそれ以降動作ｂは生起しない）

時相論理で記述された仕様を満足する並行プログラムの動作系列を求める方法

に，時相論理の定理証明手続きであるタブロー法がある．

(b) 同期部生成手法

本体部の構造はペトリネットとして抽出できる．同期部生成は，本体部から抽出されたペトリネットがLPTL式で記述された仕様を満たすように，ペトリネットのトランジションの発火系列を求めることとして定式化できる．与えられたペトリネットと時相論理から両者を満たす発火系列を求めるアルゴリズムが存在することが示されている(8)．ただし，MENDELS ZONEの実装では，有界なペトリネットに限定してアルゴリズムを高速化している（有界でない場合は有界なペトリネットに近似する）．このアルゴリズムはタブロー法の拡張になっている．このとき，ペトリネットと時相論理の両方を満たす全ての発火系列の集合を状態遷移グラフで表したものが生成された同期部である．

(c) タブロー法の並列化

同期部生成に用いる拡張されたタブロー法は，ペトリネットと時相論理式がある程度の大きさ以上になると計算コストの爆発を生じ現実的に同期部生成が不可能になる．しかし，並列マシン上で高速にタブロー法を実行することにより，現実的に生成可能な範囲を広げることができる．我々は，分散タブロー法を開発し，16台CPUの並列マシン上で実行することにより，2-8倍の高速化を実現した．

4.3 KL1コードの生成

本体部に同期部を結合してできたMENDELプログラムは，KL1コードにコンパイルされ並列マシン上で実行される．プログラムの実行過程はMENDELS ZONEの画面上でビジュアルに表示される．MENDELS ZONEで生成されたプログラムは形式仕様記述（等式論理，時相論理）を満たすことは保証されるが，それが真にユーザの要求にあっているか否かはユーザがビジュアルに表示される実行過程を見ながら確認する必要がある．

5．適用事例

MENDELS ZONEの適用事例として、あるプラントの系の進行を制御するシステムの開発を行っている。このシステムは、あらかじめ与えられた知識とプラントから送られてくる操作要求信号を基に、プラントが次にすべき動作を推論して指令を出力するシステムであり、実際に現存する。MENDLES ZONEを用いてその70%に相当する部分を構築した。その結果、MENDLES ZONEによって実システムを充分構築できることが判明した。特に、仕様記述レベルにおいて検証を実施できることは、高信頼性の保証されたプログラムを作成する上で、非常に有効であった。

6．あとがき

以上、高信頼性が保証された並列プログラムを作成するための手法、およびその手法に基づいたMENDELS ZONEについて述べた。

今後、様々なアプリケーションを構築することにより、MENDELS ZONEのソフト

ウェア開発環境としての洗練化を図っていく。

謝辞
本研究は第５世代コンピュータプロジェクトの一環である。研究の機会をいただいた新世代コンピュータ技術開発機構(ICOT)、長谷川部長代理をはじめ関係者に感謝の意を表する。

文献

(1)S.Honiden, et al.:An Application of Structural Modeling and Automated Reasoning to Real-Time Systems Design, The Journal of Real-Time systems, Vol.1, No.3, pp.313-331 (1990)
(2)本位田真一、他：代数的仕様と時制論理によるリアルタイムSAとオブジェクト指向設計の融合手法、情報処理学会論文誌、Vol.33, No.2 (1992)
(3)大須賀昭彦,他：E単一化子の完全集合を求める推論規則，コンピュータソフトウェア，Vol.8, No.3,pp.33-54 (1991).
(4)A.Ohsuga, et al.: Metis: A Term Rewriting System Generator, Software Science and Engineering (Nakata,I. and Hagiya,M. eds.),pp.1-15, World Scientific (1991).
(5)A.Ohsuga, et al.: Complete E-unification based on an extension of the Knuth-Bendix Completion Procedure, in Proc. Workshop on Word Equations and Related Topics, LNCS 572, pp.197-209,Springer-Verlag (1990).
(6)N.Uchihira, et al.:Concurrent Program Synthesis with Reusable Components Using Temporal Logic, Proc. 11th COMPSAC,pp.455-464(1987).
(7)N.Uchihira,et al.:Synthesis of Concurrent Programs: Autoamted Reasoning Complements Software Reuse, Proc. 23th HICSS,pp.64-73(1990).
(8)N.Uchihira,et al.: Verification and Synthesis of Concurrent Programs Using Petri Nets and Temporal Logic, TRANS.IEICE,Vol.E73,No.12,pp.2001-2010(1990)