

数値計算プログラムのビジュアルな構築環境

井上勝行* 小林康祐***
河合克哉** 梶 正弘***
北村操代*

A Visual Environment for Building Arithmetic Calculation Software

Katsuyuki Inoue, Katsuya Kawai, Misayo Kitamura, Yasumasa Kobayashi, Masahiro Kaji

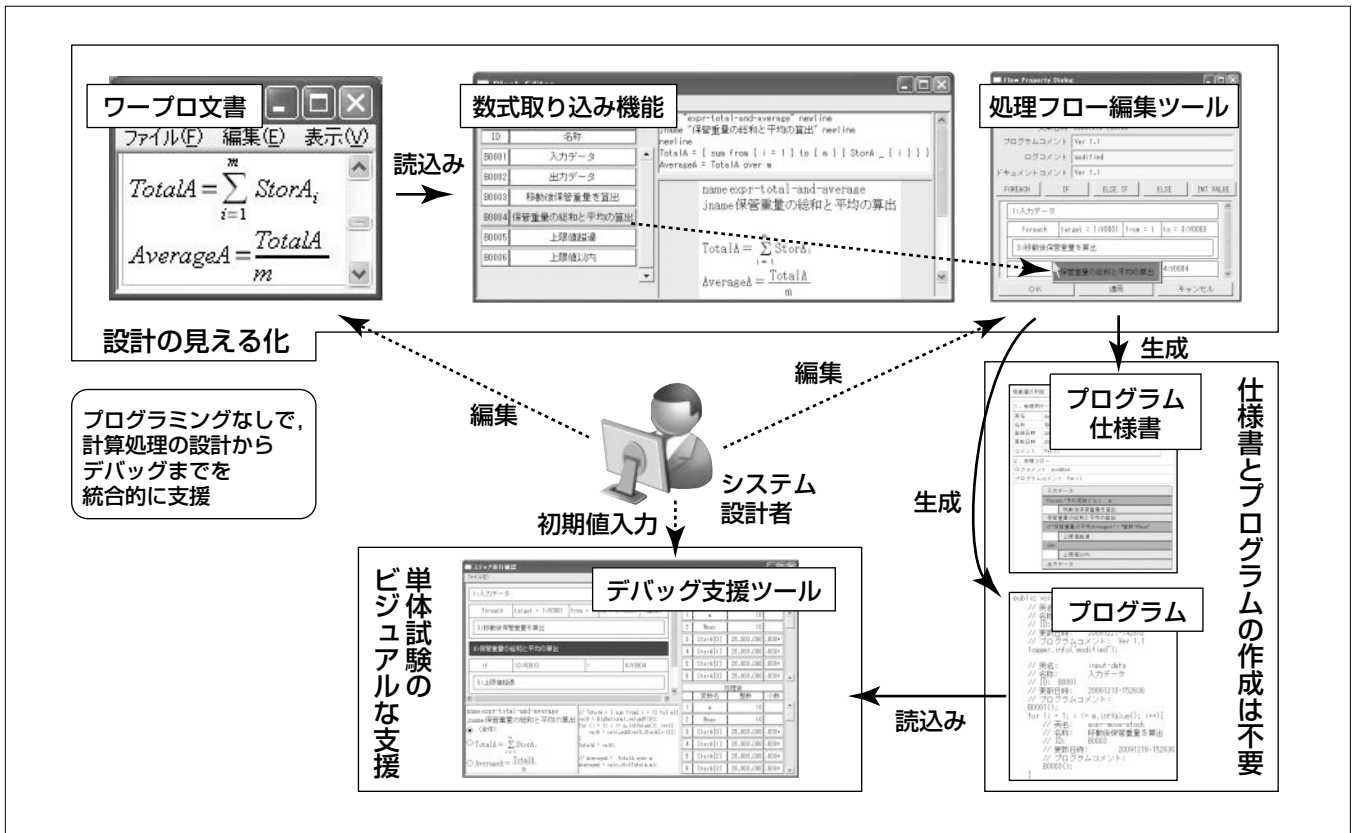
要 旨

一般に、監視制御システム、工場の運転管理システムなどの産業用システムは、センサなどで取得した数値データを用いて複雑な計算をする機能を備えていることが多い。こうした数値計算機能は、年々高度化・大規模化している。また計算の内容は、法令の改正などの外的要因によってもしばしば変更される。これらの理由から、数値計算機能の構築における生産性の向上は、今後ますます重要となる。

三菱電機では、産業用システムにおける数値計算機能をビジュアルに構築できる数値計算プログラム構築環境を開発した。この構築環境は、数値計算で用いる数式を汎用(はんよう)のワープロソフトウェアで作成した文書から取り込むことができ、それらの実行順序をユーザーがビジュアルに指定するだけで、数値計算プログラムとその仕様書

を生成する。またこの構築環境は、数値計算のプログラムコードを見なくても数式のレベルでデバッグできるようにするビジュアルなツールも提供しており、数値計算の設計からデバッグまでを統合的に支援する環境となっている。この構築環境を利用することで、プログラミングの知識がほとんどない技術者でも数値計算機能を構築できるようになり、生産性の大きな向上を実現できる。また同様に、この構築環境が生成する仕様書を見ることで、プログラムで実際に実行される計算の内容を詳細に把握できるようになる。

本稿では、数値計算プログラム構築環境の開発コンセプトと、おもな機能・ツールについて述べる。



数値計算プログラム構築環境による数値計算機能の構築

ユーザーが、汎用のワープロソフトウェア上で入力した数式をこの構築環境へ取り込み、それらの実行順序をこの構築環境上でビジュアルに指定するだけで、数値計算プログラムとその仕様書を生成させることができる。また試験・検証工程用に、数値計算の外部仕様をバッチ処理的に確認するツール、及び数値計算の実行をビジュアルに再現するツールも提供している。

1. ま え が き

一般に、監視制御システム、工場の運転・管理システムなどの産業用システムは、センサなどによって取得する数値データを用いて複雑な計算をする機能を備えていることが多い。当社では、こうした数値計算の機能をビジュアルに構築できるプログラム構築環境を開発した。この構築環境を利用すると、汎用のワープロソフトウェアで入力した数式に、実行順序をビジュアルに指定するだけで、数値計算プログラムとプログラム仕様書を生成できる。

本稿では、数値計算プログラム構築環境の開発コンセプトと、おもな機能・ツールについて述べる。

2. 数値計算機能の構築における課題

産業用システムにおける数値計算の内容は、システムの運用が始まってからもしばしば改修される。その原因として、関連する法令の改正、プラントの設備・機器の更新、仕様についてのプログラムの誤解、仕様の漏れなどが挙げられる。そして改修の際には、システム設計者(以下、設計者という。)が仕様書を正しく改訂し、それをプログラムが適切に解釈してプログラムを正確に書き換え、試験技術者が試験の必要な範囲を特定して試験を実施する必要がある。このように、数値計算の改修には工数がかかる。このため、数値計算機能をより容易に構築・改修できることが求められる。

また、プラント技術者、設計者などシステム関係者のほとんどは、プログラミングの知識を持たないのが普通である。このため、これらの関係者が数値計算の詳細を確認する際には、プログラム仕様書に頼ることになる。したがって、プログラムで実行する計算の内容が細大漏らさずプログラム仕様書に記述されていることを保証する必要がある。

3. 数値計算プログラム構築環境

3.1 開発コンセプト

当社では、数値計算の内容をプログラミングしなくても、設計者がビジュアルに構築できるプログラム構築環境が必要と考えた。その構想は、数値計算を構成する数式を汎用のワープロソフトウェアで入力し、それらの実行順序をユーザーがビジュアルに指定することで、プログラム構築環境がプログラムとプログラム仕様書を生成するというものである。この構想によれば、数値計算をプログラミングしなくても、汎用ワープロ上でのビジュアルな数式のレベルで数値計算プログラムを作成でき、さらにプログラムと仕様書の内容が完全に一致することを保証できる。

この構想に基づいて、数値計算プログラムのビジュアルな構築環境を開発・実用化した。この構築環境では、数値計算の処理を処理フローと呼ぶモデルで表現した。図1に

示すように、処理フローは、処理ブロックを要素とする構造化フローチャートとして数値計算処理を表現したものである。また処理ブロックは、任意個の数式を一つのかたまりとして表現する。処理フローは構造化に対応しているので、処理ブロックの逐次実行と条件分岐だけでなく、数列の各要素に同じ処理ブロックを適用するなどの処理を、繰り返し構造を用いて簡潔に表現できる。

この構築環境を用いて数値計算機能を構築することによって、2章で述べた二つの課題を解決できるようになる。

3.2 数値計算のビジュアルな定義

3.2.1 ワープロ文書からの数式の取り込み

この構築環境は現在、OpenOffice.org Writer(以下“Writer”という。)文書からの数式の取り込みに対応している。Writerは、近年急速に普及が進んでいるフリーなオフィススイートOpenOffice.org⁽¹⁾のワープロソフトウェアで、その標準文書形式は、ISO/IEC 26300規格のODF(Open Document Format)⁽²⁾に準拠している。また、Writerの数式入力機能はテキスト形式での入力に対応しており、例えば図2でTotalAを算出している数式のテキスト表現は、“TotalA = Sum from {i=1} to {m} {StorA_i}”である。

この構築環境は、このテキスト形式で表現された数式をWriterの文書ファイルから抽出・解析して、プログラムを生成するべき数式と変数の情報を取得する。例えば、図2に示すWriter文書を取り込んだ場合、図3に示す処理ブロックエディタに取り込んだ数式の内容が表示される。

処理ブロックエディタの左側には、取り込み済みの処理ブロックのIDと名称が列挙され、右側にはユーザーが選択した処理ブロックの内容が、テキスト表現(上段)とワープロ表現(下段)で表示される。一方、変数ビューアには、

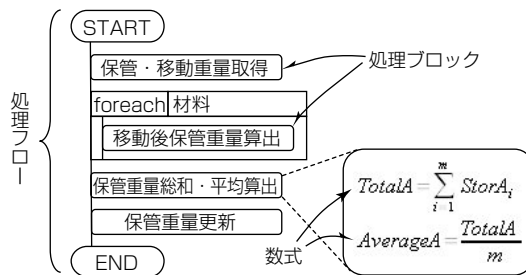


図1. 数値計算処理のモデル

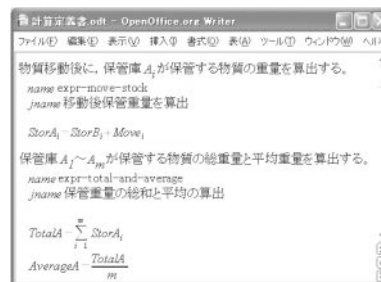


図2. OpenOffice.org Writer文書の例

変数名のテキスト表現，生成するプログラムでの変数名，変数の型（整数，又は有理数）などが一覧表示される。

3.2.2 数式実行順序のビジュアルな指定

Writer文書から数式と変数を取り込んで処理ブロックと変数を構築環境へ登録したのち，処理ブロックの実行手順を処理フローとして定義することによって，この構築環境がプログラムとプログラム仕様書を生成する準備が整う。

処理フローを編集するためのビジュアルなツールの画面を図4に示す。この画面下半分の領域へ，処理フローの構成要素をドラッグ・ドロップすることで，処理フローを構造化フローチャートとして作成・編集できる。処理フローの構成要素とは，処理ブロックエディタ上の処理ブロック，図4中段にボタンが並んでいる繰り返し構造(Foreach)，条件分岐(IF, ELSE-IF, ELSE)，及び変数ビューア上の変数と整数定数(INT-VALUE)である。例えば図4は，処理ブロックエディタから処理ブロック“保管重量の総和と平均の算出”をドラッグして，“移動後保管重量を算出”とIF要素の間にドロップ(挿入)しようとしている場面である。なお，矩形(くけい)“foreach”の右方にある“1:V0001”，“3:V0003”などは，変数ビューアからドロップした変数のIDである。

3.3 数値計算プログラムの生成

この構築環境は，処理フローを単位として数値計算のプログラムを生成できる。現在は，Java^(注1)言語に対応しており，各処理フローは一つのJavaクラスとして生成される。このクラスでは，長けた計算ライブラリを用いて計算を行うため，けたあふれが発生することはない。

(注1) Javaは，Sun Microsystems, Inc. の登録商標である。

3.4 プログラム仕様書の生成

この構築環境は，処理フローの詳細な仕様を記述したプログラム仕様書も生成できる。現時点では，Writer文書形式によるプログラム仕様書の生成に対応している。

プログラム仕様書には，処理フローの名称，更新日時などの管理用データ，処理フローの構造化フローチャート，使用する変数の一覧，各処理ブロックで実行される数式などが記述される。プログラム仕様書の生成例を図5に示す。

3.5 設計データの管理

この構築環境では，ユーザーが登録した処理フロー，処理ブロック，変数などのデータを，設計データと総称している。ユーザーは，これら設計データを保存するディレクトリを指定・切替えできる。これによって，数値計算を適用するシステムを随時切り替えて，数値計算機能を構築できる。

また，処理フロー中で使用する処理ブロック，処理ブロック中で使用する変数といった，設計データの使用関係を抽出する機能を備えている。これによって，ほかの設計データが使用している設計データをユーザーが削除できないようにして，設計データの整合性を維持する。

なお設計データは，名称，ID，最終更新日時などの管理用データを埋め込んだテキスト形式のファイルである。このため，設計データのバージョン管理に使用するツールを，様々な汎用ツールの中からユーザーが自由に選択できる。

4. デバッグ支援ツール

数値計算の構築では，ユーザーがその動作を詳細に確認できることが重要である。この構築環境では，単体試験をバッチ処理として実行する動作確認ツールと，処理フローの任意の時点での状況を再現するステップ実行ツールを提供しており，動作確認とデバッグの作業を強力に支援する。

4.1 動作確認ツール

動作確認ツールは，処理フローの動作をバッチ的に確認するためのツールである。ユーザーが処理フローを指定して動作確認ツールを起動し，入出力定義ファイルを指定す

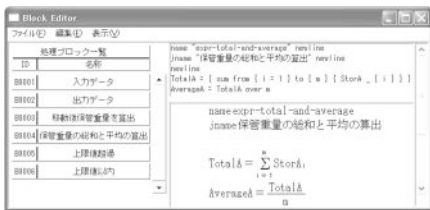


図3. 処理ブロックエディタの画面例



図4. 処理フロー編集ツールの画面例

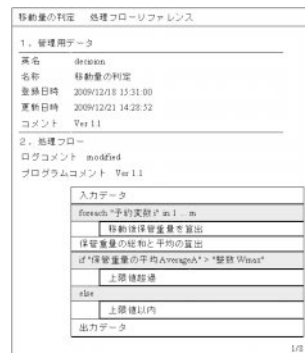


図5. 生成したプログラム仕様書の例

ると、動作確認を実行できるようになる。入出力定義ファイルとは、動作を確認するケース(以下、テストケースという。)ごとに、処理フローの実行開始時と正常終了時の変数値を、ユーザーが事前に記述しておいたものである。ユーザーは、任意の数のテストケースを入出力定義ファイルに記述できる。

動作確認ツールは、処理フローからプログラムを生成して、テストケースごとに変数値に初期値を設定して実行し、実行終了時の変数値を正常終了時のものと照合する。照合した結果は、テストケースごとに図6に示す動作確認結果詳細ウィンドウに表示される。このウィンドウには、正解値(正常終了時の値)と出力値(実行終了時の値)が、変数ごとに一覧表示され、出力値が正解値と異なる変数(図6では、変数StorA[10])の情報がハイライト表示される。

動作確認に用いる入出力定義ファイルは、一つの処理フローに対して必要なだけ作成・保存できる。このため、ユーザーは処理フロー、又は処理ブロックを更新するごとに、過去の動作確認で使用した入出力定義ファイルを用いて、機械的に回帰テストを実施できる。

4.2 ステップ実行ツール

動作確認ツールが実行結果に注目するのに対し、ステップ実行ツールは、処理フロー実行の任意の瞬間を再現するツールである。ステップ実行ツールは、処理フローの実行時に記録しておいた詳細な履歴情報を基に、ユーザーが指定する処理ブロックなどの実行前後での変数の値を表示する。

例えば図7左上部に表示している処理フローでは、ユーザーが指定した処理ブロックである“保管重量の総和と平均の算出”を反転表示している。そして、この処理ブロックの内容のワープロ表現が図7左下部に、中央下部には、生成したプログラムが表示される。さらに、図7右上部には、処理ブロック“保管重量の総和と平均の算出”実行前の変数の状態が一覧表示され、また右下部には実行後の状態が表示される。

再現する瞬間の指定では、処理ブロック以外にも、処理ブロック内の数式を選択したり、繰り返し構造の何回目の繰り返しを再現するかを指定したりもできる。さらに、繰り返し構造全体を選択して、その実行前後の状態を表示させることもできる。このように、ステップ実行ツールを利用すれば、処理フローの実行を詳細に追跡することも、大きな粒度で俯瞰(ふかん)的に確認することも容易である。

5. む す び

産業用システムの発達とともに、その数値計算機能も高度化・大規模化している。また数値計算の内容は、法令の改正などの外的要因によってもしばしば変更される。これ



図6. 動作確認結果詳細ウィンドウの表示例



図7. ステップ実行ツールの画面例

らのことから、数値計算機能の構築における生産性向上は、今後ますます重要となる。

本稿で述べた数値計算プログラム構築環境は、数値計算で用いる数式をワープロ文書から取り込み、プログラムとプログラム仕様書を生成する。また、数値計算のデバッグをビジュアルに支援する強力なツールも提供しており、数値計算の設計からデバッグまでの作業全体を統合的に支援するビジュアルな環境となっている。この構築環境を利用することによって、プログラミングの知識がない技術者でも数値計算機能を構築できるようになり、生産性の大きな向上を実現できる。また同様に、プログラムを見なくても、この構築環境が生成する仕様書を見るだけで、プログラムで実際に実行される計算の内容を詳細に把握できるようになる。

この構築環境は、エネルギー関連分野の工場運用支援システムへ適用される予定である。今後は、この構築環境の適用によって削減できる開発量の検証を進めていく。

参考文献

- (1) OpenOffice.org—the free and open productivity suite,
http://www.OpenOffice.org/
- (2) OASIS Open Document Format for Office Applications (OpenDocument),
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office