

# 「情報処理学会論文誌：プログラミング」の編集について

## プログラミング研究会論文誌編集委員会

情報処理学会では、研究会の活性化を目指して様々な改革を進めている。プログラミング研究会はこの流れを受けて、研究会のあるべき姿について徹底的な討論を行ってきた。その帰結として、研究会独自の論文誌の編集にいち早く踏み切ることを決定した。

研究会論文誌「情報処理学会論文誌：プログラミング」の特徴と意義は大きく3つある。第1は、従来の「論文」に対して想定されてきた対象分野や査読基準では必ずしもカバーしきれない、多様な成果の公表の場を提供することである。第2は、投稿論文の内容を研究会で発表することを義務づけることによって、迅速で的確な査読を実現するとともに、議論の結果の最終稿へのフィードバックを可能にすることである。第3は、研究内容の表現に必要であると認められれば、長大な論文も採録可能としている点である。

本論文誌を通じて、日本のプログラミング分野の研究活動を盛り上げていきたい。読者諸氏からの多くの論文投稿を期待する。

### 1. 対象分野

プログラミングは、コンピュータの誕生と同時に生まれた伝統的な分野であるが、コンピュータがある限り不可欠な技術である。並列分散処理やマルチメディア応用など処理内容が高度になるにつれて、プログラミングの重要性は増すことがあっても減ることはないであろう。

「情報処理学会論文誌：プログラミング」は、プログラミングに関するテーマ全般を専門に扱う論文誌である。具体例として次のようなテーマがあげられる。

- プログラミング言語の設計，処理系の実装
- プログラミングの理論，基本概念
- プログラミング環境，支援システム
- プログラミング方法論，パラダイム

これらを応用したシステムの開発事例も対象に含まれる。また、上記以外でも、プログラミングに関する面白い話題であれば対象となる。

### 2. 編集方針

本論文誌は、プログラミング研究会における発表と論文誌投稿が密接にリンクされている点に特徴がある。

論文誌への投稿者が用意する研究会発表用の資料が、そのまま本論文誌への投稿論文となる。

研究会発表をせずに本論文誌に投稿することはできないが、逆に、本論文誌への投稿をとまなわない研究会発表は可能である。そのような発表や、論文が不採録となった発表については、アブストラクトが本論文誌に掲載される。従来のプログラミング研究会の研究報告は廃止し、その代わりとして、研究会登録者には本論文誌が配布される。

本論文誌に掲載する論文は、通常のオリジナル論文と、サーベイ論文の2種類とする。どちらの種類であるかは、著者自身の指定によって決まる。論文の記述言語は日本語、英語のいずれかとする。論文の長さには制限は設けない。

### 3. 査読基準

基本的に、減点法に陥ることを避け、論文の良い点を積極的に評価するという方針を貫く。具体的には、新規性、有効性などの評価項目のうち、どれか1つの点で特に優れていると認められれば採録する。体裁のみが整った論文より、若干の不備はあっても技術的な貢献の大きい論文を積極的に受け入れる。

このような観点から、たとえば次にあげるような、従来は論文としてまとめることが難しかった内容について論じた論文もできるだけ受け入れる。

- プログラミング言語の設計論
- システムの開発経験に関する報告
- 斬新なアイデアの提案
- 概念の整理，分類法，尺度の提案
- 複数のシステムその他の比較

### 4. 投稿から掲載までの流れ

本論文誌への投稿希望者、および研究会での発表希望者は、発表会開催日の約2カ月前までに発表申込みをする。具体的な方法は研究会ホームページ <http://www.ipsj.or.jp/sig/pro/> を参照していただきたい。申込みの際には、本論文誌への投稿の有無、オリジナル論文とサーベイ論文の種別指定を明記する。また、アブストラクト（和英両方、和文は600字程度）を添付する。

論文投稿を希望した場合は、研究発表会の約1カ月前までに、別に定めるスタイル基準に従ったカメラレディ形式で論文を提出する。

毎回の研究発表会の直後、編集委員会が開催され、各論文について1名の査読者が決定される。査読報告をもとに、編集委員会は採録、条件付き採録、不採録のいずれかの判定を行い、発表会開催後3週間程度で発表者に採否通知を行う。照会の手続きはないが、論文改善のための付帯意見が添付される場合がある。この場合は、3週間以内に改良版を作成する。

## 5. 研究発表会

2002年度の発表会予定は次のとおりである。

6月17～18日	[プログラミング言語の設計と実装]
8月21日	[SWoPP—並列/分散/協調プログラミング言語と処理系]
10月18～19日	[特集テーマは設けない]
1月	[同上]
3月	[同上]

## 6. 編集母体

本論文誌は、下記のプログラミング研究会論文誌編集委員会の責任で編集を行う。各研究発表会ごとに2名の担当編集委員が割り当てられ、投稿論文の査読プロセスを主導する。

### 2001年度プログラミング研究会論文誌編集委員会

委員長	柴山悦哉 (東京工業大学)
委員	天海良治 (NTT)
	石畑 清 (明治大学)
	岩崎英哉 (電気通信大学)
	上田和紀 (早稲田大学)
	小川瑞史 (科学技術振興事業団・NTT)
	小野寺民也 (日本アイ・ピー・エム)
	久野 靖 (筑波大学)
	高木浩光 (産業技術総合研究所)
	高橋和子 (関西学院大学)
	寺田 実 (東京大学)
	富樫 敦 (静岡大学)
	西崎真也 (東京工業大学)
	原田康德 (科学技術振興事業団・NTT)
	前田敦司 (筑波大学)
	松岡 聡 (東京工業大学)
	村上昌己 (岡山大学)
	八杉昌宏 (京都大学)
	結縁祥治 (名古屋大学)

本号は、編集時期が年度をまたがったため、2002年度の論文誌編集委員会メンバが主になって行った。2002年度の編集委員会は以下のメンバにより構成される。

### 2002年度プログラミング研究会論文誌編集委員会

委員長	村上昌己 (岡山大学)
委員	岩崎英哉 (電気通信大学)
	上田和紀 (早稲田大学)
	小川瑞史 (科学技術振興事業団)
	小野寺民也 (日本アイ・ピー・エム)
	久野 靖 (筑波大学)
	田浦健次朗 (東京大学)
	高木浩光 (産業総合技術研究所)
	高橋和子 (関西学院大学)
	富樫 敦 (静岡大学)
	原田康德 (科学技術振興事業団・NTT)
	前田敦司 (筑波大学)
	松岡 聡 (東京工業大学)
	結縁祥治 (名古屋大学)
	渡部卓雄 (東京工業大学)

## 本号の編集にあたって

2001年度第4回研究発表会  
担当編集委員 富樫 敦, 結縁祥治

2001年度第5回研究発表会  
担当編集委員 岩崎英哉, 原田康德

本号は、2001年度第4回プログラミング研究会(通算第37回)からの採録論文2件と2001年度第5回プログラミング研究会(通算第38回)からの採録論文7件からなる。

2001年度第4回プログラミング研究会は、2002年1月29, 30日に、工学院大学にて、電子情報通信学会コンピュテーション/ソフトウェアサイエンス/コンカレント工学の各研究会と同時連続で開催された。第5回研究会は、2002年3月15, 16日に電気通信大学で開催された。

研究会のテーマは、第4回が「並列・分散処理」、第5回が「プログラミング言語一般」とし、幅広く論文を募集した。研究会論文誌への投稿をともなう発表のほかに、論文投稿をともなわない発表を歓迎したことも、これまでと同様である。その結果、第4回では12件、第5回は14件の発表が行われた。

第4回、第5回研究会とも、研究会当日の昼休みや発表終了後に編集委員ならびに編集委員会が出席を依頼したメンバが集まって編集委員会を複数回にわたって

て開催した。編集委員会では、その委員会直前またはその前のセッションで発表された各論文について、発表から時間をおくことなく議論を行った。ただし、投稿論文の共著者となっているメンバは、その論文についての議論の間は退席している。委員会では先に記した対象分野、編集方針および査読基準に従って、各論文の評価できる点について意見が交され、その場で可能なかぎり査読者の選定を行うようにした。各査読者は、編集委員会での議論をふまえて査読を行った。

結果として、第4回研究会からは2件、第5回研究会からは7件の通常論文が採録された。これ以外の発表については、各々について1ページの概要を掲載した。

以下、掲載論文について、簡単に紹介する。

「PowerPC プロセッサの特性を考慮した高速 Java バイトコードインタプリタの構成法」(Constructing a Fast Java Bytecode Interpreter for PowerPC Processors)では、冗長なメモリロードと高頻度な間接分岐のオーバヘッドの問題を、PowerPC プロセッサ用の Java バイトコードインタプリタにおいて解決する手法を提案している。さらに、それぞれの実行速度に対する効果を IBM POWER3 プロセッサ上で主要な Java ベンチマークプログラムを用いて示している。

「May 別名除去」(Eliminating May-aliases)では、部分 Must 別名をコード巻上げの手法によって Must 別名に変換し、従来法よりもさらに正確な定義-使用関係を明らかにする手法を提案している。この手法は、別名解析が終了していることを前提として、部分冗長除去に用いられるデータフロー解析と類似の解析法によって、まず部分 Must 別名を見つけ出し、次にそれらが Must 別名となるプログラム点を求めている。これらの解析は、ビットベクタ表現を用いた単方向データフロー解析によって実現でき、プログラムサイズを  $n$  とした計算量は、 $O(n^2)$  で抑えられることを示している。

「Java2C トランスレータにおける可搬性のオーバヘッド」(Overheads for Portability in a Java2C Translator)では、著者は、Java から C への変換系と C コンパイラの組合せで構成される Java コンパイラにおいて、null 検査、スタックオーバーフロー検査など Java 処理系が暗黙に実施する処理のオーバヘッドが、プラットフォーム依存の技法を使わなかった場合にどの程度になるかを定量的に見積もっている。著者が開発した Java 実行環境においては、暗黙の処理のオーバヘッドは、コード量にして約 40%、実行時間にして約 5~10% となることを報告している。

「家電向け Java JIT コンパイラの構成方法とその評価」(A Construction of Java JIT Compiler for Consumer Electronics, and Its Evaluation)では、著者らは、携帯電話を含む家電機器向けの Java Just-In-Time コンパイラの構成方法について述べている。家電機器では、メモリ資源や移植性から来る制約から、一般の Java JIT コンパイラが行うような、対象アーキテクチャに特化した高速化技法をとることはできない。著者らは、コンパイラの大きさを縮小するために、最適化を行わないコード生成器のみから処理系を構成し、移植性を実現するためにそのコード生成器をインタプリタループのソースコードから自動的に合成する方式を採用している。さらに、この JIT コンパイラを複数のアーキテクチャ上に実装し、定量的な性能評価結果を示している。

「別名情報に基づくレジスタ促進」(Register Promotion Based on Alias Information)は、先の論文「May 別名除去」の続編ともいえる論文である。この論文で著者らは、最適化コンパイラにおける重要な処理の1つであるレジスタ割付(レジスタ促進)において、仮想レジスタへのロード命令の冗長性を除去する手法を提案している。従来のロード命令の除去では、ロード対象の記憶場所への参照式が同じものだけを扱ってきたが、提案手法では“部分 Must 別名”に基づくことによって、同じ記憶場所を表すすべての別名を除去対象とすることができるとしている。

「述語付きソフトウェア・パイプラインへの Spiral Graph によるレジスタ割付」(Register Allocation for Predicated Software Pipelining Using Spiral Graph)では、著者らは、ソフトウェア・パイプラインにおいて、条件分岐を含むプログラムに対して、最小またはそれより1多いだけの個数のレジスタ割付を行う手法を述べている。これは、IA-64のようなレジスタ改名機構と述語付き命令実行機構を備えたアーキテクチャに対するレジスタ割付に利用することができる。著者らは、述語付き命令において述語の値が異なるために別々に定義される生存区間を適切に表現するために、従来の Spiral Graph の拡張である述語付き Spiral Graph を提案している。

「非可約な制御フローグラフのための簡潔で高速な支配木と支配辺境の検出算法」(A Concise and Fast Algorithm for Irreducible Control Flow Graphs to Identify Immediate Dominators and Dominance Frontiers)では、著者らは、非可約(複数のヘッダを持つループを含む)を含む任意の制御フローグラフを対象に、支配木と支配辺境を求める算法を提案し、

その正しさを証明している。基本的なアイデアは、複数のプリヘッドとヘッドを結ぶ頂点（仮想頂点）を制御フローグラフに追加したうえで、その有向非循環グラフ（DAG）部分について、著者らがすでに発表している縮約アルゴリズムを適用することによって支配木と支配境界を計算することにある。提案算法では、複雑なデータ構造などを用いずに支配木と支配境界が同時に得られ、その計算量はグラフの辺数を  $M$  とすると  $O(M)$  であるとしている。

「Java プログラムにおいて例外による順序制約を投機的命令移動を用いて除去する方法」(Eliminating Exception Constraints in Java Programs Using a Speculation Technique)では、著者らは、Java のような型安全な言語のプログラムにおける実行時例外検査によって課せられる命令間順序制約を除去する方法について述べている。提案手法では、対象を基本ブロック内に限定し、命令のデータ依存、例外依存、順序依存の関係を辺とする有向非循環グラフ（DAG）を用いて、ハードウェアに関する例外を発生する命令に投機的命令移動を適用する。著者らは、この方法を Java Just-In-Time コンパイラに実装し、いくつかのプログラムを IA-64 プロセッサ上で実行した結果、小規模なプログラムで最大 31%、SPECjvm98 ベンチマークで最大 11% の性能向上を得たとしている。

「部分評価を応用した動的 Web ページのキャッシュ機構」(A Caching System for Dynamic Web Pages Using Partial Evaluation)は、データベースへの問合せを行い動的にページを生成する処理において、PHP という言語で記述された Web ページの動的生成スクリプトを部分評価することで、データベースへの問合せを減らし、ページ生成の負荷を軽減するシステムを提案している。提案システムでは、データベースの更新に合わせてスクリプトを部分評価し結果を Web サー

バに配置する。ページを生成する際には、この部分評価されたスクリプトが実行されるため、更新の頻度の低いデータへの問合せは行われなくなり、データベースサーバの負荷が低減される。著者らの提案手法は、実時間に変更される情報を含むページやパーソナライズ機能を持つページに対しても適用が可能であるとしている。

最後に 2001 年度の活動についてまとめておく。

2001 年度は、5 回の発表会で 69 件の発表があった。前年度は 63 件、さらにその前年度は 62 件であったので、発表件数はゆるやかながらも増加傾向にある。発表のうち論文誌へ投稿された論文の中から合計 26 編が採録された。また、発表会では 50 名近い参加者が集まることもあり、発表後の質疑応答も大変に活発であった。発表者は、学生、研究者をはじめ、企業で開発に携わっている方も多く発表され、いろいろな分野のプログラミングの研究活動を盛り上げるという本論文誌の活動が評価されているといえよう。

最後に、活発な研究会活動を支えていただいた、発表者、発表会参加者、論文投稿者、査読者の方々へ感謝の意を表したい。大変短い査読期間にもかかわらず論文査読の労をとっていただいた方々の氏名を掲げる。

#### 2001 年度査読者

合田 憲人, 天海 良治, 石畑 清, 伊知地 宏,  
 稲垣 達氏, 上田 和紀, 大須賀 昭彦, 小川 宏高,  
 小川 瑞史, 小野寺 民也, 角田 博保, 木下 佳樹,  
 久野 靖, 胡 振江, 小出 洋, 古関 聡,  
 佐々 政孝, 柴山 悦哉, 首藤 一幸, 田浦 健次朗,  
 滝本 宗宏, 田中 哲, 田中 良夫, 近山 隆,  
 千葉 滋, 寺田 実, 中田 秀基, 萩谷 昌己,  
 長谷川 立, 藤波 順久, 松本 尚, 村上 昌己,  
 八杉 昌宏, 山名 早人, 脇田 建, 渡部 卓雄