

「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、迅速で確か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なと認められれば、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も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2~3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。申し込みの際には、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3週間前までに、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

プログラミング研究会では、発表会ごとに特集テーマを設けている。ただし各発表会では、特集以外の一般の発表もつねに受け付けている。

2000年度の発表会予定は次のとおりであり、各発表会の特集テーマは、今後数年間はそのまま維持する予定である。

6月15～16日[プログラミング言語の設計と実装]

8月2～5日[SWoPP—並列/分散/協調プログラミング言語と処理系]

11月16～18日[理論]

1月22～23日[並列・分散処理]

3月26～27日[プログラミング言語一般]

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。2000年度より論文誌編集委員会メンバを増やし、各研究発表会を2名の編集委員で担当している。

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)

伊知地宏 ()

岩崎英哉 (東京大学)

上田和紀 (早稲田大学)

小川瑞史 (NTT)

小野寺民也 (日本IBM)

久野 靖 (筑波大学)

高木浩光 (電総研)

寺田 実 (東京大学)

富樫 敦 (静岡大学)

西崎真也 (東京工業大学)

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)

本号の編集にあたって

八杉昌宏, 高木浩光

2000年度第2回プログラミング研究会は、2000年8月2,3日に松山市総合コミュニティセンターにおいて、『2000年並列/分散/協調処理に関する『松山』サマー・ワークショップ(SWoPP松山2000)』(8月2日～5日開催)の一環として開催された。

昨年度に引き続き、発表予定者には、従来のSWoPPへの発表申し込みに加えて、プログラミング研究会への発表申し込みも行っていただいた。論文投稿の有無などの指定はプログラミング研究会への申し込みの方にのみ明記することになる。発表申し込みの時期はSWoPPへの発表申し込みの時期に合わせているので、やや早いスケジュールとなっている。

発表枠としては最大14件を想定していた。SWoPPでの他の研究会と比べて件数が少ないのは、発表と質疑の時間を合計45分(発表25分,質疑20分)としているためである。会場や発表スケジュールの調整ではSWoPP幹事のお世話になっている。発表募集を行ったところ、予定していた枠とほぼ同数の申し込みがあり、最終的には13件の発表を行うことになった。方式が変更された1998年度以来、申し込みに対して発表枠が不足することはなく、当初懸念されていた大幅な発表枠不足の問題は生じていない。

研究会は8月2日と3日に開催され、SWoPPの他の参加者も交えた活発な質疑討論が行われた。また、論文誌の創刊にともなう変更として従来の研究報告が

廃止されているため、発表者が発表資料を必要部数、持参する形となっている。このため、SWoPP 参加者は自由に発表資料を入手できるようになっている。

投稿原稿の査読を議論する編集委員会会合は、編集委員ならびに編集委員会が出席を依頼したメンバで現地にて開催した。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。各論文を議論するのに十分な時間を確保するために、会合は3回に分け、昼休みの時間などのまとまった空き時間のほとんどを利用した。十分な議論の後、各投稿論文について担当の査読者を決定し、査読を行ってもらった。

最終的に、投稿を希望したうちの6件の論文(すべて通常論文)が採録となった。これらの論文の掲載に続き、それ以外の発表については1ページの概要を掲載してある。掲載順序は、論文、概要それぞれについて当日の発表順に従うことにした。

掲載した論文について、以下、簡単に紹介する。

「並列 KLIC 処理系上での配列演算の最適化」で、著者らは、並行論理型言語 KL1 の UNIX 上の処理系 KLIC 上で、単一代入変数の特徴を生かしたまま効率の良い並列配列計算を実現することを目的としている。この論文では制約ベースの静的解析系による変数のモード/型/参照数情報と具体化状態に関する性質を併用して、破壊的代入、タグ判別の省略等で高速に読み書きできる数値演算用の配列の実装について述べている。

「Java 向けソフトウェア分散共有メモリの実現」では、Java 言語の実行系である JVM が共有メモリ型並列計算機での動作を前提として PC クラスタなどの分散メモリ型並列計算機上でマルチスレッドで動作させることができないことから、ソフトウェア分散共有メモリシステムの実現を目的としている。この論文では、科学技術計算などで用いられる SPMD スタイルのマルチスレッドプログラムを対象としたソフトウェア分散共有メモリの、JVM の改変を必要としない実現について述べている。

「アセンブリ言語レベルでの異種計算機間のヒープとスタックの共有機構」では、ソフトウェア分散共有

メモリを用いた異種計算機間のメモリ上のデータの共有を特定の言語処理系に依存せずに実現する手法を示している。この論文では、メモリ上のデータを計算機固有のデータ表現形式ではなく計算機間で共通のデータ表現形式を用いて表現し、また、その手法をスレッドのスタックにも適用することで、計算機間のスレッドの移動に必要なスタックの共有も可能な手法について述べている。

「オブジェクト指向スクリプト言語 Ruby への世代別ごみ集めの実装と評価」では、オブジェクト指向スクリプト言語 Ruby に世代別ごみ集めを実装する場合の方法を示すとともに、評価を行っており、世代別 GC にすることによる GC 処理時間の大幅な短縮を確認している。

「高度な問題領域依存チューニングを許す並列組合せ最適化ライブラリ PopKern」では、問題領域依存知識の活用に重点をおいた汎用並列組合せ最適化ライブラリ・パッケージである PopKern について述べている。アルゴリズムの問題領域依存部分をすべて利用者定義とし、問題領域に独立なアルゴリズムの枠組みのみをライブラリとして提供して、利用者定義部分をプラグインすることによって探索を行うことで、利用者は高度な高速化技法を自由に記述することができ、かつ並列処理特有の専門知識を必要とする記述を行うことなく並列計算ができると述べている。この論文では、PopKern の設計、実装、大規模問題への適用による評価を述べている。

「資源情報サーバにおける資源情報予測の評価」では、分散コンピューティング環境における効率的なタスクスケジューリング実現の一環として、分散コンピューティング環境上のプロセッサやネットワークの負荷など計算資源に関する情報(資源情報)を収集し、それをもとに将来の値を予測する資源情報サーバ(Resource Information Server; RIS)を開発している。この論文では、資源情報の計測と予測方法、RIS のシステム構成、予測時間と予測精度に関する評価について述べている。

最後に、研究会開催、論文誌編集にご協力を賜りました皆様に感謝いたします。