


研 究 計 画 書

2004年 9月提出

学籍番号 3603U147-0 CD

氏 名	矢島 伸吾	専門分野	高度計算機構	指導 教員	上田 和紀 
		研究指導	並列知識情報処理研究		
研 究 題 目	小規模制御系における LMNtal 処理系の設計と実装				修 士 課 程
<p>1. 研 究 目 的</p> <p>LMNtal は、大規模計算から極小計算までを簡潔かつ統一的に扱うことを目的とした言語である。LMNtal 処理系の研究開発において大規模計算に対する議論は活発になされ、分散計算環境上の LMNtal 処理系の設計、実装も行われているが、小さな計算環境における LMNtal 処理系のあり方についてはほとんど議論されていない。一方、情報家電や産業機械、ロボットなど、小さな計算機の協調動作によるシステム制御が近年特に重要視されている。</p> <p>本研究では、小規模な計算環境上における LMNtal 処理系の設計と実装を行い、LMNtal によるシステム制御の可能性について模索する。対象システムとしては、1. カメラ、無線を始め、様々な入出力制御機構を備えている、2. 上記制御がライブラリとして提供され、C, C++ 言語による実装が可能である、3. 搭載 RAM が 1M と大きく、メモリ制約が緩いため実装が容易である、という点から、ロボット制御コントローラー eyebot を選んだ。</p> <p>LMNtal は、細粒度の並行性を持つ、分散処理を簡潔に記述できるなど、制御システムの構築に有利な特徴をもった言語である。しかし、現処理系においては、入力に対する即時応答性を得るのが難しいなどの問題がある。本研究を通じて LMNtal による制御システム記述の利点、欠点を洗い出し、制御システムに特化した LMNtal 言語のサブセットについても考える。</p>					
<p>2. 従 来 の 研 究</p> <p>LMNtal 処理系は、去年の卒業論文において基本的な計算機構を作成して以来、言語仕様の見直し、並列・分散までを考慮に入れた処理系の設計の見直し、ルールコンパイルにおける中間言語の作成などが開発チームにより行われた。その中で私は、処理系の見直し（実行スタックの扱い、自由リンクの扱いなどの変更）や言語仕様についての議論、モジュール仕様に関する考察、処理系コア部分の実装などを行った。なお、処理系の設計については JSSST にて発表を行った。</p> <p>また、LMNtal の研究とは別に、以前から興味があった脳に関する研究（脳神経科学、脳数理科学）について調査した。脳科学と自分の専門分野との接点を見つけるのが目的だったが、現在の脳研究は医学もしくは数学、心理学が主であり、脳科学に関連する研究を行っていくのは難しいという結論に至った。</p> <p>関連研究としては、LEGO mindstorms の制御コントローラー RCX 上における、Scheme 処理系の実装の研究がある。この研究では 32KB の空間に収まる処理系を実装しており、処理系の小規模化や処理系における入出力制御の扱いについて参考にすることができる。また、他にも KLIC や HCC など、よくカスタマイズされた処理系は数多く存在する。これらのソースやドキュメントを読むなどして実装方法を参考にすることも有効であると考えている。</p>					

3. 研究計画
1. eyebot の理解
- eyebot の提供する機能についてテストを行い、またモータやセンサなど周辺機器の購入、製作をとおしてマイコンシステムやハードウェアについての知識を得、理解を深める。また、提供ライブラリの仕様や使用方法について理解する。
- eyebot が提供する機能について理解し、それらの中で LMNtal 処理系上から扱えるようにする機能を選択し、またそれらの機能の LMNtal 上における表現方法について考える。
2. LMNtal 処理系の理解
- 現行の処理系のソースコードを読み、とくに実行時データ構造と中間コードの仕様と解釈について理解する。
- eyebot 上の処理系の実装にあたり、サポートする LMNtal の文法およびそこから生成される中間コードを決定する。
- 実装した処理系のテストには、最初はハンドコンパイルした中間コードを用いるが、最終的にコンパイラ側の改変利用もしくは新規実装を行うことが可能かを考え、今後の指針を決定する。
3. eyebot 上の LMNtal 処理系の設計
- 1. で決定した「LMNtal 上で扱える機能」を実現するために必要なシステム構造を設計する。とくに、センサ等の入力値の変化に即応できるようなシステムを目指す。
- 2. で決定した中間コードを解釈、実行できる実行時システムを設計、実装する。基本は Java 版の処理系を参考にし、リアルタイム制御、GC などを実現するためのデータ構造や必要となる機構を追加する。実装には C 言語（もしくは C++ 言語）を用いる。
- 使用する中間コードはハンドコンパイルにより生成する。
4. 関連研究のサーベイ
- 他の処理系の実装や組込み、RTOS、ロボット制御、空間量などの資料、論文を参考にし、3. の実装を見直し、改変する。
5. コンパイラの実装
- 実行時システムが解釈可能な中間コードを生成できる LMNtal コンパイラを実装する。
- 実行時システムとコンパイラを処理系として配布可能な形にまとめ、公開する。
6. 論文の執筆
- 上記研究成果をまとめる。LMNtal 処理系の eyebot への移植というだけでなく、LMNtal における即時処理、入出力処理の扱いかたを提案するとともに、C 言語による LMNtal 処理系のプロトタイプとなることを目指し、詳細なドキュメントの記述に努める。