「機能と構成」研究領域　領域活動・評価報告書

－ 平成 17 年度終了研究課題 －

研究総括　片山 卓也

1. 研究領域の概要

本領域は、"先進情報システムとその構成に向けて"、独創性に富んだ提案を積極的にとりあげようとするものです。具体的には、これからの社会を支える高度な機能をもった情報システムの構築を目指し、そのための構成や構築方法に関して、基本技術や先進応用事例および基礎となる理論の研究を行います。例えば、ソフトウェア、ネットワーク、プロセッサ、分散・実時間・埋め込みシステム、セキュリティ、設計・実装・進化方法論と環境、テスト・検証技術、形式的手法、高信頼性技術、ユーザインタフェースなどの研究を含みます。

2. 研究課題・研究者名

別紙一覧表参照

3. 選考方針

選考の基本的な考え方は下記の通り。

1）選考は、「機能と構成」領域に設けた 8 名のアドバイザーの協力を得て、研究総括が行う。

2）選考方法は、書類選考、面接選考、および総合選考とする。

- 書類選考においては、1 提案に付き研究総括と 2 名のアドバイザーが査読し、評価する。
- 面接選考では、可能な限り多くの提案者から直接説明を受け、質疑応答を行う。

3）個人レベルで行う研究提案であり、萌芽的でかつ独創性に富んでおり、3年間の研究により科学技術への大きな貢献が期待できそうなものを優先する。また、研究テーマや研究者の所属機関が特定のところに集中しないよう配慮する。

4）審査は書類選考結果、面接選考結果および研究実施の条件等を加味し、総合的観点から行う。

4. 選考の経緯

選　考	書類選考	面接選考	採用者
対象者数	42 人	20 人	7 人

5. 研究実施期間

平成 14 年 11 月～平成 18 年 3 月

6. 領域の活動状況

- 領域会議: 11 回

・ 研究報告会:3 回
・ 研究者訪問:研究総括が、研究開始後、全研究者を訪問した。その後、研究実施場所の移動時あるいは適宜、研究総括および／または技術参事が研究者を訪問した。

7. 評価の手続き

研究総括が、各研究者からの報告および自己評価を基に、アドバイザーの協力を得て行った。また研究終了時に当領域が開催する研究報告会(一般公開)等の参加者の意見を参考とした。

(評価の流れ)

平成 18 年 3 月	研究期間終了
平成 17 年 12 月	研究報告会を東京ガーデンパレスにて開催
平成 17 年 12 月	研究課題別評価提出
平成 18 年 1 月	研究総括による評価
平成 18 年 1月	研究報告書提出

8. 評価項目

(イ) 外部発表(論文、口頭発表等)、研究を通して得られた新しい知見などの研究成果
(ロ) 得られた研究成果の科学技術への貢献

9. 研究結果

平成 14 年度は 42 件の提案の中から、ソフトウェアの高信頼性化や検証技術に関わる研究(大崎研究者、長谷川研究者、結縁研究者)、大規模分散コンピュータに関わる技術の研究(デファゴ研究者、丹羽研究者)、 特にゲームに関わる人工知能の研究(飯田研究者)、およびソフトウェアを大規模 LSI 上に並列回路として自動展開する技術の研究(井口研究者)を採択した。応募数はやや少なめではあったが、採択した提案はいずれも優れたものであった。

3年間の研究期間を終了し、7名の研究者全員が当初の期待通りの研究成果を上げ、我が国の科学技術発展に大きな貢献ができたものと考える。平成 17 年 12 月 21 日に東京ガーデンパレスにて開催した、広く一般を対象とした研究報告会では、多くの好意的なコメントを頂いた。また、このうち2名の研究者が実装したプログラムは、近々公開を予定しており、ご利用頂きたい。

研究者毎に言えば、飯田研究者は、世界トップレベルのコンピュータ将棋プログラムの開発に成功し、各種コンピュータ将棋大会で優勝ないしは上位入賞を果たしたほか、プロ棋士との公開対戦を行った。また、ゲーム洗練度指標を導入しそれとゲームの進化との間に興味深い関係があることを見つけ, また, 競技性、遊戯性、哲学性といった観点から、三名人モデルを提唱した。

井口研究者は、ループの並列展開などの並列処理技術を応用し、ソフトウェアから演算並列性を引き出し、大規模 VLSI 上に専用並列化回路として展開する手法を開発した。また、実際に、実用的なアプリケーションの並列化合成を行い、その有効性を実証した。

大崎研究者は、等式つきツリーオートマトンの理論を提案し, その理論を発展させることによりリアクティブシステムの安全性を自動的に検証するための技術を開発し、自動検証ソフトウェア ACTAS を実

装した。また、これを用いて暗号通信プロトコルの安全性の確認を行い、その有効性を実証した。

デファゴ研究者は、同意問題や障害検出などに焦点を当て、フォールトトレラントな超大規模分散システム実現のための研究を行った。ここで開発したアクルーアル障害検出器は、性能および柔軟性の面で極めて優れており、特筆に値する。また、その評価のために NekoLS プラットフォームを開発し、その有効性を確認した。

丹羽研究者は、性能面ではクラスタキャッシュやプリフェッチ機構などの導入により、また耐故障機能面ではシステムレベルのチェックポインティングを用いて、広域環境に存在する計算機資源上で、共有メモリモデルに従って記述された並列プログラムを効率良く実行させるための最適化コンパイラを開発した。

長谷川研究者は、プログラムの動的振舞いを表す制御構造に関する先進的な数学的理論を構築し，この分野の研究に大きく貢献した。すなわち，継続、再帰、多相性をもつ高階関数型言語の振舞いを正確に表現できる意味論を数学的に展開し、プログラム意味論の研究を大きく前進させた。

結縁研究者は、Web アプリケーションの振舞いを Web オートマトンの合成として抽象化し，それからテスト系列を生成する方法や，非同期 π 計算への変換を行うことにより形式的解析を可能とすることにより，高信頼 Web アプリケーションプログラム開発の道を開いた。

10. 評価者

研究総括：片山 卓也 北陸先端科学技術大学院大学情報科学研究科 教授

領域アドバイザー（所属・役職は現職）

青山 幹雄	南山大学 数理情報学部 教授（平成 13 年 4 月～）
阿草 清滋	名古屋大学 大学院情報科学研究科 研究科長・教授
市川 晴久	日本電信電話(株) NTT 先端技術総合研究所長
岩野 和生	日本アイ・ビー・エム株式会社（～平成 13 年 3 月）
菊野 亨	大阪大学 中之島センター長・大学院情報科学研究科 教授
中島 秀之	公立はこだて未来大学 学長
南谷 崇	東京大学 先端科学技術研究センター 教授
湯淺 太一	京都大学 大学院情報学研究科 教授
米崎 直樹	東京工業大学 大学院情報理工学研究科 教授

（参考）

(1) 外部発表件数

	国　内	国　際	計
論 文	17	18	35
口 頭	62	85	147
その他	6	35	41
合 計	85	138	223

(2) 特許出願

国内	国際	計
7	1	8

（3）ソフトウェア公開（予定を含む）

- 大崎人士　ACTAS（Associative and Commutative Tree Automata Simulator）
- 丹羽純平　Optimizing compiler and runtime library for WDSM

(4) 受賞等

- 飯田弘之
  2004 年 5 月:　4th CSA 世界コンピュータ将棋選手権（木更津市）でタコス 7 位
  2005 年 5 月:　15th CSA 世界コンピュータ将棋選手権（木更津市）でタコス 6 位
  2005 年 9 月:　10th Computer Olympiad（台湾）将棋部門でタコス優勝（金メダル）
  2005 年 10 月:　第 3 回コンピュータ将棋王者決定戦（東京）でタコス 7 位
  2005 年 11 月:　コンピュータ将棋 2005 世界最強決定戦（能美市）でタコス 3 位

- 長谷川真人
  2005 年 11 月:　第 19 回日本 IBM 科学賞（コンピューターサイエンス分野）

(5) 招待講演

国際　13 件

国内　6 件

別紙

「機能と構成」領域　研究課題名および研究者氏名

研究者氏名 （参加形態）	研 究 課 題 名 （研究実施場所）	現　　職 （応募時所属）	研究費 （百万円）
飯田 弘之 （兼任）	探索と知識の融合がもたらす知能の創造と進化 （北陸先端科学技術大学院大学）	北陸先端科学技術大学院大学 情報科学研究科　教授 （静岡大学情報学部　助教授）	40
井口 寧 （兼任）	ハードウェア・プログラミングによる超細粒度並列処理 （北陸先端科学技術大学院大学）	北陸先端科学技術大学院大学 情報科学センター　助教授 （同　助手）	49
大崎 人士 （兼任）	刺激応答型実時間システムの自動検証技術：安全性・信頼性技術の開発 （産業技術総合研究所）	産業技術総合研究所システム検証研究センター　研究員 （同情報処理研究部門　研究員）	47
デファゴ クサビエ （兼任）	大規模分散アルゴリズム開発及び性能評価のツール構築 （北陸先端科学技術大学院大学）	北陸先端科学技術大学院大学 情報科学研究科　特任助教授 （同知識科学研究科　助手）	28
丹羽 純平 （専任）	広域分散共有メモリ機構を支援する最適化コンパイラ （東京大学）	科学技術振興機構 さきがけ研究者 （東京大学大学院情報理工学系研究科 助手）	28
長谷川真人 （兼任）	プログラミング言語の制御構造の意味論的分析 （京都大学）	京都大学数理解析研究所 助教授 （同上）	12
結縁 祥治 （兼任）	Web アプリケーション指向ソフトウェアモデリング （名古屋大学）	名古屋大学大学院 情報科学研究科　助教授 （同工学研究科　助教授）	24

研究課題別評価

1 研究課題名:

探索と知識の融合がもたらす知能の創造と進化

2 研究者氏名:

飯田 弘之

3 研究のねらい:

問題領域を限定するとき，探索と知識の融合によって超高度な知能を実現することが可能である。このようにして実現される機械的な知能は，人間の知能とは異なる特徴を持っている。本研究では，名人を超えるコンピュータ将棋の研究開発に取り組み，知能の創造と進化という観点から，そこに内在する機能と構成の美の科学的解明を目指した。さらに，意思決定システムの多様性や，知能とゲームの進化を探求した。

4 研究成果:

4.1 名人を目指すコンピュータ将棋の開発

一般に，コンピュータはゲームをプレイするとき，たくさんの局面を探索し，その中から一番良い手を選択する。将棋はチェスよりはるかに複雑なゲームであるため，たくさんの局面を網羅的に探索することは不可能である。それゆえ，本筋の手を深く読み，なおかつ，どの局面が自分にとって一番望ましいかを素早く判断することが重要である。

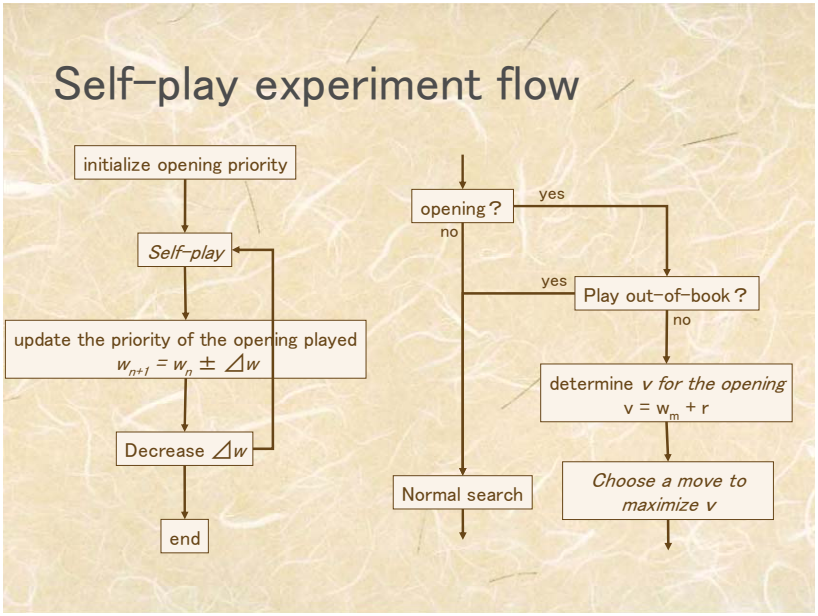
本筋の手を識別するために，指し手の必然性に着目した。指し手をいくつかのカテゴリに分類し，それぞれの指し手がどの程度の必然性をもっているかを計算して，先読み探索の方向性を制御する。必然性のなさそうな手は先読みを早く打ち切り，必然性の高い手はもっと先まで先読みする。必然性の計算のために，コンピュータの自動対戦を数多く実施し，それぞれのカテゴリの手がどの程度の割合で選択されているかを分析し，プログラムが最も安定したふるまいになるように自動調整した。

また，終盤の寄せ合いの場面で，高速に最善手を見出すことができるように，詰み探索のアルゴリズムを大幅に改良した。一般に，終盤の局面では，詰みがある場合よりも，詰みがない場合に不詰めを証明する探索コストが高くなる。証明数探索をベースに改良した詰め探索ルーチンは，不詰めであることを高速に識別できるようになり，複雑な場面でも最善手を発見できるようになった。しかし，先読み探索の末端周辺で詰みの有無を判定するにはまだ時間がかかり過ぎるという問題が残されている。末端ノードの局面で一手詰みの有無を判定するというわずかな工夫でかなり改善された。

コンピュータの最大の弱点は序盤だと言われている。定跡を抜けた後，あるいは，途中で定跡から抜けてしまった場合，定跡が意図するような戦略を継承できないという欠点である。全ての定跡変化に対してこの問題を解決するのは至難の業である。そこで，序盤戦術をいくつかの有力な戦型に絞り，かつ，それらの定跡変化に現れる全局面で定跡から外れた場合を想定し，コンピュータによる自己対戦を実施し，それぞれのふるまいを評価した。こうして，コンピュータは自分にとって得意な序盤作戦を識別できるようになった。結果として，序盤で相手の狙いを理解しつつ，自分の得意な戦型に誘導する

という高度な戦略が可能になった。

以上の研究開発を通して，コンピュータ将棋「タコス」は非常に強くなり、コンピュータ将棋世界選手権でも常時決勝に進出するようになった。また，コンピュータオリンピックでは優勝して金メダルを獲得した。さらに，プロ棋士との公開対局ではもう一歩というところまで追い詰めるという成果を上げることができた。

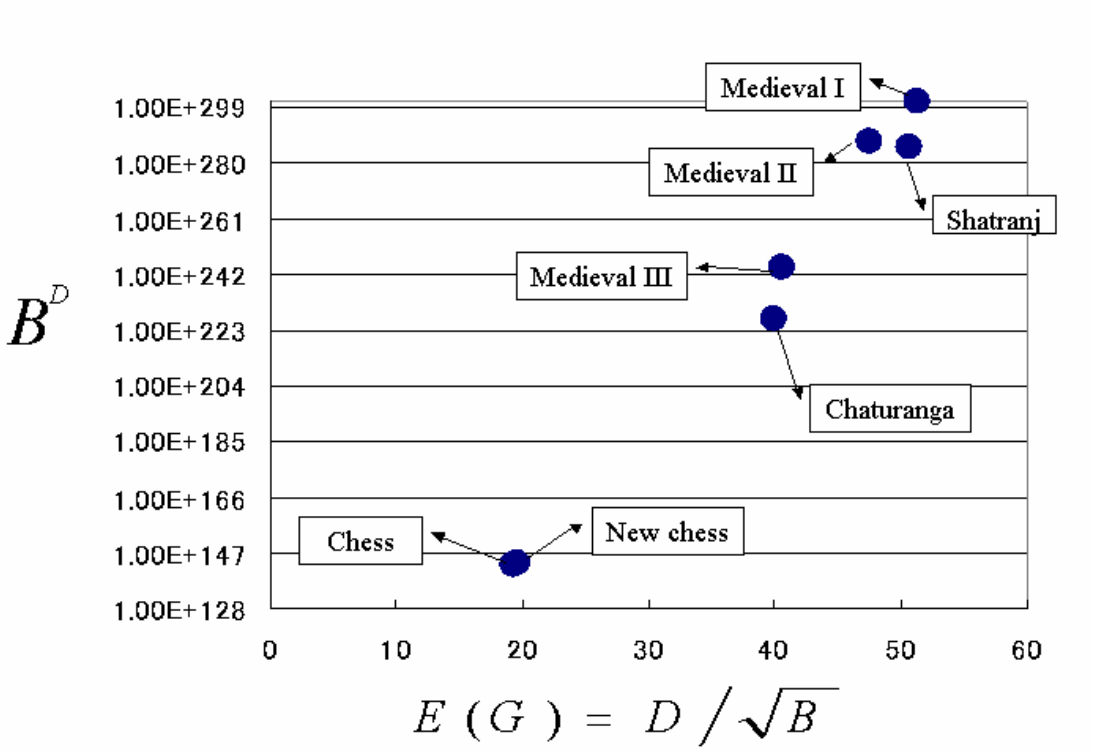


序盤定跡の途中変化や終了した局面からたくさんの自動対戦を実施する。自動対戦の勝率に基づいて定跡手順選択の重み付けに反映させる。

図1　序盤定跡チューニングのための自動対戦の概要

4.2　ゲーム洗練度の指標とゲームの進化

ゲームはシーソーゲームのときにエンタティンメント性が高まる。シーソーゲームとは試合の結果が終了間際まで判らないハラハラ，ドキドキするような試合プロセスのことである。このようなシーソーゲームの概念を定式化する数理モデルを考案した。具体的には，試合結果の情報量 x(t)を試合進行時間 t の関数として表す。シーソーゲームの特徴は，終了間際での情報量の変化率（微分）に現れる。この数理モデルから，ゲームの自由度 B（可能な手の数）および試合時間 D（終了手数）に対して$\sqrt{B/D}$の値をゲーム洗練度の指標として提案した。チェス，中国将棋，将棋，囲碁，麻雀など，長い歴史を経て洗練淘汰してきたゲームではこの指標の値がいずれも 0.07～0.08 の間になることがわかった。一方，歴史の浅いゲームのほとんどは，この指標の値が異なっている。このように，ゲーム木の探索空間の複雑さと洗練度の指標に基づいて，ゲームの進化を論じる枠組みをゲーム洗練度の理論として提唱した。ゲームでのスリル感という観点で，ゲームの洗練度を高めることがゲームの進化の原動力になってきたことが明らかになった。



横軸はゲーム洗練度の指標の逆数、縦軸は探索空間の複雑さの指標を表す。チャツランガ（紀元５世紀）から中世のチェス変種（１６世紀）を経て現代のチェス（１７世紀）に至る。最初は洗練度が低いまま探索空間の複雑さが増加し、後に探索空間の複雑さが減少し洗練度が高まるように進化してきたことがわかる。

図2　チェス種の進化

4.3　三名人モデルからゲーム場における知の力学へ

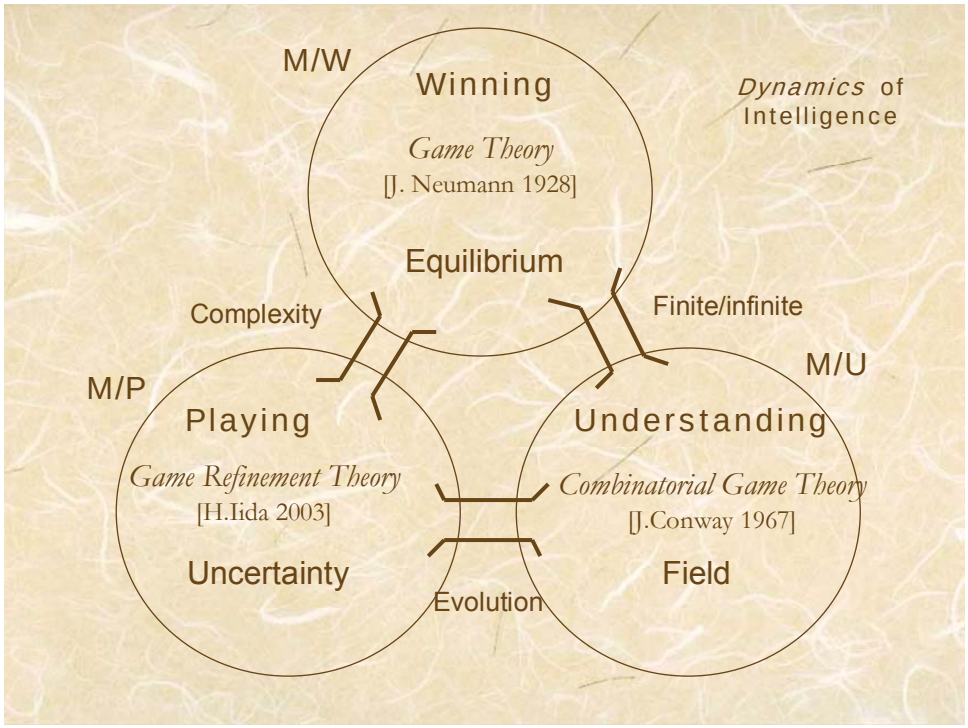
定性的な意思決定の理論として提唱されたゲーム理論では，ゲームの均衡点，つまり，ゲームの解を求める。ミニマックス均衡やナッシュ均衡などが有名である。競技性，遊戯性，哲学性，というゲームの3つの側面を考えるとき，従来のゲーム理論はゲームの競技性に焦点を当てた，ゲームに勝つための理論である。本研究では，ゲームの全体的な側面に焦点を当てるべく，三名人モデルを提唱した。上述した3つの側面に対応させて，名人（master of winning），達人（master of playing），鉄人（master of understanding）と呼ぶ。

名人の視点では，ゲーム理論[von Neumann 1928]が主要な役割を果たす。相手プレイヤの合理的なふるまいを想定した上で最善の選択を試みる。「ゲーム＝不確定性」という視点では，ゲーム理論は不確定性がゼロの状態を記述する表現法と言える。

達人の視点では，ゲーム洗練度の理論[Iida 2003]が本質的である。シーソーゲームによる場のテンションによってゲームの遊戯性（特にスリル感）が高まるからである。シーソーゲームを楽しむには，プレイヤの能力に応じた適切な場としてのゲームが必要である。「ゲーム＝不確定性」という視点では，ゲーム洗練度の理論は，どの程度の不確定性がヒトの知にほどよいスリル感を与えてきたかを示している。

鉄人の視点では，ゲームの不確定性が稠密になる場を求める。J.Conway は著書「On Numbers And Games」（1976）で，ゲームは可換な群であり，数はその部分群となることを示した。そして，数の集合から体（field）を構成する方法を提示し，ゲームに基づく新たな数論を展開した。これらの理論は組み合せゲーム理論と称されている。

三名人モデルでは，それぞれの理論が目指すものだけでなく，重なり合う部分での特質を考察することで，それぞれ異なる視点での理論が支配する世界同士の架け橋を探求することを可能にした。そのために，ゲーム場における知と知の相互作用に着目した。物体と物体の間に働く相互作用は質量の積に比例し，またその距離の二乗に反比例するが，知と知の間に働く相互作用はどのように定式化できるだろうか。「ゲーム場」および「テンション」の概念を導入してこの定式化に取り組んだ。これを「ゲーム場における知の力学」と名付けた。シーソーゲームによるゲーム場のテンションはその一例で，ゲーム洗練度の指標を導き，ゲームの進化を解き明かす扉を開けることに成功した。



ゲーム理論値の不確定性，ゲームの試合結果の不確定性，ゲーム設計上の不確定性に対峙するそれぞれを名人（M/W），達人（M/P），鉄人（M/U）と名づけた。不確定性がゼロから無限大へと変遷する中で，ヒトの知は適度な不確定性の中にスキルとチャンスの調和美を感じられるようにゲームを進化させてきた。それはゲーム場における知の相互作用の現れを示すものである。

図3　三名人モデル

5　自己評価:

チェスで主流となった単純な「探索」ベースの思考ルーチンに、局面評価や指し手の必然性に関する「知識」を融合することで、いわゆる本筋の変化を重点的に深く読む手法を開発した。この手法が功を奏し、非常に強いコンピュータ将棋を実現することができた。まだ名人レベルには至っていないが、

名人を目指すにあたりその方向性が見えてきた。当初目指していた「知能の創造」は達成したと思う。

コンピュータ将棋で開発した探索手法に機械学習のアイデアを織り混ぜる交ぜることで、ほぼ自動的にターゲットとするゲームの高品質なプログラムを開発できるようになった。その理由として、与えられたゲームのルールから、どのような指し手が本筋となるかを識別する技法の自動化があげられる。これを応用して、過去に廃れたゲームのプログラムを効率よく作成し、ゲームの進化論的変遷を調査した。さらには、ゲームの進化とそこに内在するプレイヤの知の嗜好との関連が浮かび上がってきた。ゲームを切り口として、その進化プロセスを追跡することで、ゲーム進化および知の多様性を考察することができた。特に、シーソーゲームの概念から、ゲーム洗練度の理論を導き出すことができたことは、予想以上の成果と言える。

公平性やスリル感などに着目しつつ、ゲームの持つ性質を深く考察することで、トーナメントあるいは一般的なランキングシステムのようなメタなゲームにおいても、そのような性質を適宜取り入れつつ進化してきたことがわかった。チェスのような古典的なボードゲームでは、公平かつ公正さを維持するために「引き分け」を導入し、一方、メタなゲームであるランキングシステムではランキングの「あいまいさ」を残すことで敗者へのいたわりとしてきた。まさに、ここにゲームにおける機能と構成の美を見出す。直感的に感じてきたこのような性質をゲーム情報学的な観点から少なからず明らかにできたことは望外の成果と言わざるを得ない。

本研究を通して新たな問題を発見する機会を得たことも望外の成果と言えるだろう。すなわち、名人同士の対戦で感じられる迫力、高揚感、エンタティンメント性などが、コンピュータ同士あるいは名人対コンピュータの対戦ではあまり感じられない。今後、この問題を探求するにあたり、ゲーム場における知と知の相互作用という、まさに情報科学的アプローチでしか迫れない新たな研究領域の発展につながるものと思う。

6　研究総括の見解:

飯田氏の研究は，コンピュータ将棋の研究開発を通して，探索と知識の融合により高度な知能を実現することを目的としたものである．これまでのコンピュータ将棋の開発と比較すると，飯田氏のアプローチは，科学的裏づけを重視したものであり，新たな探索法の提案などにより世界トップレベルの非常に強力なシステムが構築されている．さらに，飯田氏は一般のゲームについても研究を行い，ゲーム洗練度の指標の提案とそのゲーム進化に関する大変興味深い結果を得ている．また，三名人モデルの提案を行っている．飯田氏の研究は，ゲームを人間の最も基本的な知的活動と捉え，その科学的解明を目指すと同時に，知能に関する新たな研究領域を構築しようとするものである．今後の展開が大いに期待されるものであり，大変高く評価できる研究である．

7　主な論文等:

論文

1．　J.Nagashima, T.Hashimoto, H.Iida (2005). Master-like Opening Strategy in Computer Shogi, Proceedings of JCIS2005 (The 8th Joint Conference on Information Sciences) in CD-ROM, Salt Lake.

2．　H.Iida, K.Takahara, J.Yoshimura (2004). An Application of Game-Refinement Theory to

Mah Jong, Proceedings of ICEC2004, Eindhoven, pages 333-338, LNCS3166, Springer

3. P.Majek, H.Iida (2004). Uncertainty of Game Outcome, Proceedings of Inter Academia 2004, Hungary, pages 171-180
4. T.Hashimoto, H.Iida (2004). An innovative design of game position evaluation, Proceedings of JICAST 2004 (The 8th Joint International Conference on Advanced Science and Technology), China, pages 187-190
5. M.Sakuta, J.Nagashima, T.Hashimoto, H.Iida (2003). Application of the killer-tree heuristic and the lambda-search method to LOA, Information Sciences, 154:141-155

特許

特願 2003-091750「対戦組合せ装置、方法及びプログラム」
特願 2003-187016「実力評価システム、その方法およびプログラム」
特願 2003-350543「競技システムの洗練度の測定システムおよびその方法並びにプログラム」
特願 2004-122212「選抜システム」
特願 2004-358672「順位決定システム」

受賞

2004 年 5 月　14th CSA 世界コンピュータ将棋選手権（木更津市）でタコス 7 位
2005 年 5 月　15th CSA 世界コンピュータ将棋選手権（木更津市）でタコス 6 位
2005 年 9 月　10th Computer Olympiad（台湾）将棋部門でタコス優勝（金メダル）
2005 年 10 月　第 3 回コンピュータ将棋王者決定戦（東京）でタコス 7 位
2005 年 11 月　コンピュータ将棋 2005 世界最強決定戦（能美市）でタコス 3 位

招待講演（総数：国内3件、国際7件）

2005 年 9 月 6 日　“Towards Dynamics of Intelligence in the Field of Games”, The 11th Advances in Computer Games Conference, Taipei. Keynote lecture
2004 年 9 月 5 日　“The Attractiveness of Asian Games”, ICEC2004 (The 3rd International Conference on Entertainment Computing) Workshop on Modeling and Playing Computer Games, Eindhoven. Keynote lecture
2003 年 12 月 6 日　“The Art of Opponent Models: Uncertainty and Games”, NWO - SIKS Workshop "Opponent Models in Games", Universiteit Maastricht, Maastricht. Keynote lecture
2003 年 9 月 16 日　“Games and Evolution”, IBM Innovation Days Kick-off Meeting. Tokyo, Keynote lecture
2002 年 11 月 9 日　“Chips Challenging Champions: A model of three masters”, in the 8th Game Programming Workshop, Hakone, Keynote lecture

研究課題別評価

1 研究課題名:

ハードウェア・プログラミングによる超細粒度並列処理

2 研究者氏名:

井口 寧

3 研究のねらい:

近年, VLSI の集積度は急速に向上しており, 近い将来数十億ゲートのコンピュータチップが出現すると予想される. CPUチップ上の回路に占める演算器の回路量は, 現在では既に全体の数パーセントと非常に小さくなっていて, 今後は CPU チップの回路量が増加しても, それに比例する処理性能の向上は困難だと考えられる. 一方で, 多数のマイクロプロセッサを結合した超並列計算機は, 自然科学分野におけるシミュレーションなどの多くの分野で, 大規模な計算を高速に実行することができる計算機として盛んに研究され, 並列処理に関する技術が蓄積されてきた.

そこで本研究では, これまでの超並列処理技術を VLSI チップの回路設計に適用することによって, VLSI チップ内の大規模な論理回路上に高並列な専用演算回路をプログラムの都度ごとに合成し, 将来大容量の論理回路が与えられた時にも高い処理性能を実現するための手法について研究した.

研究の進め方として, 近年利用可能なCレベル設計ツールに対して, 超並列処理技術の助けを借りながら並列化解析を行い, 専用並列化回路を合成するアプローチをとった. C レベル設計ツールは C 言語に近い記述でハードウェア回路を設計可能なツールであり, 近年盛んに開発・利用されるようになってきている. しかし, これらの処理系はコストを重視しコンパクトな回路を合成するための最適化がなされているため, そのままでは高い並列性を持った回路を合成することは困難である. そこで, 元のプログラムに含まれる並列性を解析・抽出し, C レベル設計ツールが並列演算回路を合成できるように, ディレクティブ(並列化指示)を挿入したりループを展開することによって, ソフトウェア・アルゴリズムを高い並列性を持つ演算回路として展開するアプローチを採り, 研究を行った. プログラムは, ディレクティブやループ展開などによって並列性を陽に含む記述に変換された後, C レベル設計ツールや配置配線ツールによって並列演算器を含むハードウェア回路として展開できるので, VLSI チップ上に並列演算回路を容易に構築できる. 超並列計算機での並列性解析の知見を大規模 VLSI 上での回路設計に導入することによって, 将来の大規模 VLSI チップのための超細粒度並列処理の実現を試みた.

4 研究成果:

4.1 超細粒度並列処理のためのテストベット構築

まず最初に, 本研究で実験に用いるテストベットを構築した. 図1は, 構築したシステムによる超細粒度並列処理の仕組みである. 実験用VLSIとして, FPGA 素子を利用した. FPGA は, ユーザーが使用時に内部回路をプログラムできる素子であり, ユーザーは任意の回路を繰り返し LSI チップ内に構成することができる. 本研究では, C 言語で記述されたソフトウェアを入力とし, そのソフトウェアを並列化演算回路として展開し, FPGA 上に実装する. プログラムごとに要求される内部回路が異なるので,

プログラムの実行の都度回路合成を行い，FPGA の内部回路を指定するコンフィギュレーションファイルを得てFPGAにダウンロードし回路をスタートさせる．本テストベッドを用いて，提案スキームの評価などを行った．

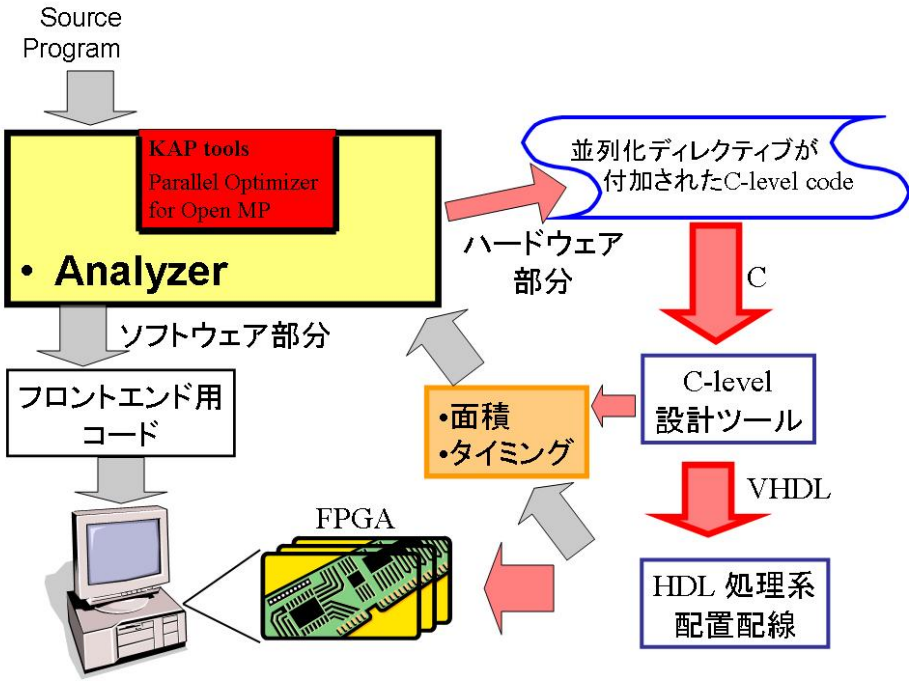


図 1　超細粒度並列処理のためのテストベット

4.2 並列化回路合成スキーム

次に C 言語のソースコードから並列演算器を合成するための手順を確立した．従来の C レベル設計ツールでは，C レベルのソースコードを与えると逐次演算回路を合成するが，逐次演算回路では利用可能なゲート数が増加しても処理速度の向上に結びつけることができない．一方，ハードウェア記述用の C レベル設計ツールは，並列な演算器を合成するための手段が用意されている．例えば，”par”ディレクティブを指定すると，par 文に続くブロック内のステートメントが並列化回路として合成される．そこで，ループを展開後，平行して実行可能な演算子群に par ディレクティブを挿入することによって，並列化演算回路を合成する手法を開発した．ディレクティブ挿入の際，従来のソフトウェア超並列処理で用いられてきた並列性解析ツールの出力を参照しながら，回路合成に必要なディレクティブを挿入する．

4.3 チップ面積を最大効率で使うための並列化最適化手法

プログラム内の各行が並列化可能だとしても，論理回路量には制限があるので，与えられた回路量の範囲内で最も効果的に並列化する必要がある．つまり，並列化しても回路量の増加が少なく処理速度向上の効果が大きい部分の並列度を高くする一方，並列化すると回路量が大きく増加する割には処理速度向上の効果が少ない部分は並列度を低くする．プログラムの各部分の並列度を決定するため

の手法を明らかにした. 並列度決定は以下の手順で行なう.

Step1: ハード化部分の選択

最初にソフトウェアとしてプログラムを実行し, 関数ごとに実行時間の割合を求める. この分析で処理時間を多く消費すると判定された関数が, ハードウェア並列化の対象となる.

Step2: 逐次版での実行速度/面積評価

行ごとに適切な並列度を決定する. コードプロファイラを用いて行ごとの実行時間を求める. 行 *i* の実行時間を *T(i,1)*と置く. *T*の最初の引数は行番号, 2つ目の引数は並列度で, 逐次版なので並列度は *1* となる. また, 対象となっている関数全体の実行時間を *Tseq* と置く. C レベル設計ツールを用いて, 逐次版の関数を論理回路に展開した時の使用論理回路量を求め, *Aseq* と置く.

Step3: 試行的並列化によるデータ収集

行 *i* をある並列度 Pt で並列化し, 使用回路量を評価する. この時の論理回路量を *A(i,P_t)* と置くと, 行 *i* の 1 並列当りに要する論理回路量は次式で記述できる.

$$\Delta A(i) = (A(i,Pt) - Aseq)/(Pt - 1)$$

Step4: 並列化後の実行時間/面積の推定

これらの情報を元に, 各行の並列化後の実行時間と面積を推定する. 行 *i* を *Pi* 並列で実装した時の推定実行時間 *T(i,Pi)*および面積 *A(i,Pi)*は次式で推定する.

$$T(i,Pi) = Tseq + T(i,1)(1-Pi)/Pi$$

$$A(i,Pi) = Aseq + \Delta A(i) \cdot Pi$$

Step5: 他の行の評価

Step3, Step4 の手順を他の行についても適用し, 各行ごとの並列度の効果を求める.

Step6: 行ごとの最適並列度の決定

計算時間を多く消費する各行に対する並列化後の実行時間と面積の推定結果より, どの行を何並列するのが最適なのか求める. これは, $\Sigma \Delta A(i) \cdot Pi + Aseq \leqq Achip$ の条件下で $\Sigma T(i,Pi)$ を最小にする *Pi* を求めることによって, 各行ごとの最適な並列度を算出できる.

4.4 応用として電子透かし検出アルゴリズムへの適用

本並列化実装手法の実用的な応用として音楽向け電子透かし検出プログラムに適用し, 性能向上を確かめた. 電子透かしは, デジタル化著作物の著作権保護や流通の制御を可能とする技術として, 近年注目されている技術であるが, CPU を用いた処理では十分な処理速度が得られない問題がある. そこで, 本研究の並列化手法を適用して, 電子透かし検出を並列化されたハードウェア回路として実装し, 電子透かし検出速度の大幅な向上を試みた.

電子透かし技術は, 著作者の情報などのビット列を人間が知覚できないようにデジタル化著作物に埋め込む方法である. 電子透かしの検出には様々なアルゴリズムが提案されているが, 本実験で用いた P. Bassia (2001)らによるアルゴリズムは, プログラム内の並列性が非常に高く, 本並列化手法

による大幅な高速化が期待できる.

表1に本手法によって合成した並列化検出回路の実行速度を示した. この例では, 44.1KHz で PCM 録音された 20.4MByte の wave ファイルの左チャネルに透かしを埋め込み, この音楽ファイルを入力とした時の検出速度を評価した. ソフトウェアによる検出では, 透かしの種類の数だけループを繰り返す必要があるため, 透かしの種類が増加すると検出速度が低下する. これに対してハードウェア並列化された検出回路では, 透かしの種類が増えても, それぞれの透かしのための検出回路は, チップ内に独立した並列化回路として合成され, これらが同時に動作するので, 透かしの種類が増えても検出速度は低下しない. 50 並列の透かし検出回路は, ソフトウェアに比べておよそ 150 倍の検出速度を達成できたことが分かる. 表2には, 合成された回路の使用ゲート量, チップ上の全ゲートに対する使用ゲート量の割合, およびクリティカルパス長を示す. 合成された回路は C 言語で記述されているが, 十分実用的な回路量で実装できていることが分かる. クリティカルパス長は, 回路の動作速度の指標である. 合成された回路の性能は, 従来の回路設計で行なわれている人間が最適化した回路に比べると決して高くはないが, ソフトウェアからの並列化回路合成の容易性という点で十分な成果だと言える.

表1　透かしの検出速度と回路規模

	ハードウェア検出		ソフトウェア検出	
	検出時間	検出速度	検出時間	検出速度
25-並列検出	500ms	326.4Mbps	38,181ms	4.2Mbps
50-並列検出	500ms	326.4Mbps	74,351ms	2.2Mbps

表2　回路量とクリティカルパス

	使用 LUTs 数	回路使用量割合	Critical Path
25-並列検出	29,485	31%	24.727ns
50-並列検出	53,660	57%	24.867ns

5　自己評価:

本研究では, ソフトウェア内の演算並列性を陽に引き出し, 大規模化する VLSI 内に専用並列化回路を合成し, 増大する論理回路を処理速度の向上に結びつけることを目標に研究を行なった. 研究成果として, これまでの超並列処理技術の助けを利用して, ループを独立した演算に並列展開する方法や並列展開されたステートメントを並列化演算回路として合成するための並列化指示行の挿入法, および並列度の決定法などを明らかにした. 更に, 実用的なアプリケーションの並列化合成を行なったところ, C 言語で記述された電子透かし検出アルゴリズムが容易に並列化回路として実装でき, その回路はソフトウェアに比べて十分な速度を得ることができた.

当初はこれらの処理過程を全自動で行なう予定であったが, 処理段階の一部で用いる並列化支援

ツールやFPGA実装ツールの動作環境の都合上，全自動合成が達成できていない．しかし，これらは手段の問題であり，知見としての並列化回路合成手法は当初の目標通り得られたものと考えられる．

今後の予定としては，現在回路の並列度を求めるために非常に長時間を要する問題があるので，予測に基づく並列度決定法の開発を予定している．更に，現在はプログラム単位の専用回路合成となっているが，近年開発されているリアルタイム書き換え可能デバイスを用いれば，ソフトウェアの関数単位ごとの専用回路にすることができるので，これらの新デバイスを用いて処理効率の一層の向上を試みる予定である．

6 研究総括の見解:

近年のハードウェア技術の進歩によりVLSIのゲート数は飛躍的に増大しているが，それらを有効に活用する技術が十分には確立されていない．井口氏の研究は，ソフトウェアの一部を直接ハードウェア化することにより，高速な実行を行わせようとするものである．C言語で記述されたソフトウェアを対象にして，並列化の効果の高いループ構造の展開や，コンパイラへの並列化ディレクティブ挿入による並列化の指示などにより，並列性の高いFPGAを合成する方式の研究を行ったものである．超細粒度並列処理のためのテストベッドの構築，並列化回路合成スキーム，チップ面積を有効活用する並列化手法などの研究を行うと同時に，電子透かし検出アルゴリズムなどの実例に対してその有効性を確認するなどの成果を上げている．各種組み込みシステムなどへの応用も十分に見込めるものであり，今後重要となる技術のさきがけとして大いに評価できる研究である．

7 主な論文等:

論文・口頭発表

1) 榊原 憲宏，井口 寧，“FPGA を用いたオーディオ電子透かしの超高速検出”，電子情報通信学会論文誌 D-II (条件付採録済み)
2) M.M. Hafizur RAHMAN, Yasushi Inoguchi and Susumu Horiguchi, “Modified Hierarchical 3D-Torus Network”, IEICE Transaction on Information System, Vol.E88-D, No.2, pp. 177-186, Feb. 2005
3) Y. Inoguchi, “Outline of the Ultra Fine Grained Parallel Processing by FPGA”, Proc. in IPSJ and IEEE, High Performance Computing in Asia Conference 2004, Workshop on Reconfigurable System for HPC, pp. 434-441, Jul., 2004
4) 井口 寧,”FPGA を用いた超細粒度並列処理の概要”，信学技報，第一回 リコンフィギャラブル研究会 論文集, pp.1-6, Sep.18, 2003

特許

1) 井口 寧，榊原 憲広，”電子透かし検出装置，それを内蔵する中継装置，及び，電子透かし検出方法”，特願 2004-230360, 2004

研究課題別評価

1 研究課題名:

刺激応答型実時間システムの自動検証技術:安全性・信頼性技術の開発

2 研究者氏名:

大崎 人士

3 研究のねらい:

インターネットのような, 公衆ネットワーク回線を通じての通信においては, 暗号化による通信の秘密の保護が必要だが, そこでは破られにくい暗号法の技術と共に, 暗号化法を通信網の中で正しく生かして使う技術が必要である. 例えば, 正規の(想定する)データ受取人になりすまして, 暗号を解くための鍵を不正に入手されるようでは, いかに破られにくい暗号法を使用していても, データの秘匿性を保つことはできない. つまり, 鍵を不正に入手して暗号化されたデータを入手されるなどという攻撃を避けるための技術が要請される. 通信手順の安全性を確保することは, 解読が困難な暗号化法を考案する技術とは独立のものである. なぜなら, 秘密のデータを暗号化して受信者に伝えるとしても, 暗号法には復号法が必ずあり(さもないと受信者はデータを読み取れない), 復号化の方法も受信者に伝える必要がある. このときに, 暗号化にもちいるトリック(多くの場合は, 鍵)を横取りされないような通信手順が必要となる. ここで問題となるのは, 「なりすまし」などの悪意の第三者からの攻撃が決して成功しないことをどのようにして確かめるのかである.

本研究では, 情報科学における基礎技術を応用して, 情報システムの安全性を検証するための技術について研究する. 書換系(かきかえけい)およびツリーオートマトンの理論による数理的基礎を発展させて, リアクティブシステムの安全性を自動的に検証するための技術を開発し, 実際に, 自動検証ソフトウエア ACTAS を作成し, ソフトウエアの性能評価をもとにアルゴリズムの改良, 実用化の模索を行うことが具体的な目標である. さらに, 具体的な研究のプロセスと経て, 一連の研究活動がスパイラル状に展開し, 新たな研究シーズの発掘が行えたのであれば, 理想的結果が得られたと考える.

4 研究成果:

本研究の第一の目的は, 「自動化の利点を損なわずに, 柔軟な表現力をそなえた普遍性のあるモデル化法とその自動検査技法を開発すること」である. 暗号プロトコルの例に対しては, より広いクラスの暗号通信プロトコルを自然な表現によりモデル化し, 安全性を自動的に検証する, ことが具体例となる. しかし, 形式言語理論の世界では一般的に, 「より強力な表現力を求めると, 自動化の利点は失われる」と言われており, 本研究の理論的基礎であるツリーオートマトンについても, この考察が当てはまると推察する. なぜなら, 従来のツリーオートマトンでは, 自動検証に適した演算を構成的に定義することや, 各種の決定判定問題を解消することはできるが, その表現力は必ずしも十分であるとは言えない. 例えば, $x = y$ のような, 線形方程式の(自然数上の)解空間を表現することは困難である. また, 交換則($x + y = y + x$)や結合則($(x + y) + z = x + (y + z)$)を仮定すると, 受理言語の閉包性

($t \in L$ かつ $t = s$ ならば, $s \in L$)が失われる. そこで本研究では, 従来のツリーオートマトンを拡張した新たな枠組みである等式付ツリーオートマトンを理論的背景にして,『表現力についての研究』と『自動計算についての研究』を行った. 前者の研究では, この新しい数理的なモデルについての閉包演算および決定問題の解決をした. 後者の研究では, 等式付ツリーオートマトンを基礎として, リアクティブシステムのための自動検証方法を考案し, 現実的な時間で安全性を検証するためのアルゴリズムの開発や, 近似計算アルゴリズムの開発などの研究を行った. 以降の節では, 本研究プロジェクトで得られた研究成果の概要について述べる.

4. 1 表現についての研究

先ず, 2001 年に, 世界にさきがけて「等式付ツリーオートマトン」という理論概念を導出した. 等式付ツリーオートマトンは, 交換則や結合則を仮定しても受理言語の閉包性を失うことはない. しかも従来のツリーオートマトンのように:

- 空(くう)判定問題が決定可能であること
- 受理言語上の集合演算について閉じていること

が多くの場合で成り立つだろうという予想があった. 本研究では, 形式表現としての性質を調べることにより, 等式付ツリーオートマトンの特性を明らかにすることを目指した.

本研究中に得られた研究成果の一覧を次頁の図1と図2にまとめる. (1)交換則・結合則を仮定するか, 結合則のみを仮定するか, (2)正則な遷移規則のみをもつと仮定するか, 正則な遷移規則に加えて単調な(monotone)イプシロン遷移規則をもつと仮定するか, の場合分けにより, 生じる表現力に違いを確かめた. 理論概念の導出以来の未解決の問題とされていた, 交換則結合則付き単調ツリーオートマトン(monotone AC-tree automata)についての以下の定理を導けたことが, 本テーマでの最大の研究成果であった.

交換則結合則付単調ツリーオートマトンの受理言語のクラスは,

1. 交換則結合則付正規ツリーオートマトンの受理言語のクラスを, 真に包含している,
2. 補集合の演算について閉じていない,
3. 包含関係(⊆)の判定問題が決定不可能である.

いっぽう, 等式付ツリーオートマトンによる自動検証の可能性を理論的に裏付けるためには, 空判定問題が決定可能になるための十分条件を調べることが, 一つの重要なカギとなる. アルゴリズム実装のためには, 空判定の計算量を測定することも必要である. 空判定が, 計算量的に実装がむずかしい場合であっても, 近似判定法があるかどうかを検討することにより, 多くの具体的な検証例を手がけることも可能になる. 本研究では, 上述(1)と(2)の場合分けのすべて対して, 空判定についての考察を行った. そしてこの考察をもとに, 結合則と交換則をうまく扱うことのできなかった従来の理論では秘密保持性の自動検証が難しいとされていた「Diffie-Hellman の鍵交換プロトコル」や「Shamir のスリーパス・プロトコル」を使う暗号通信手順が, 等式付ツリーオートマトンによる自動検証の対象に含められることを示すことができた.

Closure under Boolean operations

	regular	AC-regular	AC-monotone
closed under $\cup$	✓	✓	✓
closed under $\cap$	✓	✓	✓
closed under $(\ )^c$	✓	✓	×

regular TA $<$ regular AC-TA $<$ monotone AC-TA

	regular	A-regular	A-monotone
closed under $\cup$	✓	✓	✓
closed under $\cap$	✓	×	✓
closed under $(\ )^c$	✓	×	✓

regular TA $<$ regular A-TA $<$ monotone A-TA

図1 ブール閉包性とツリー言語階層 §

Decidability results

	regular	AC-regular	AC-monotone
$t \in \mathcal{L}(\mathcal{A}/\mathrm{AC})$?	✓ (LOGCFL)	✓ (NP-complete)	✓ (PSPACE-compl.)
$\mathcal{L}(\mathcal{A}/\mathrm{AC}) = \varnothing$?	✓	✓	✓
$\mathcal{L}(\mathcal{A}/\mathrm{AC}) \subseteq \mathcal{L}(\mathcal{B}/\mathrm{AC})$?	✓	✓	×

	regular	A-regular	A-monotone
$t \in \mathcal{L}(\mathcal{A}/\mathrm{A})$?	✓ (LOGCFL)	✓ (P-time)	✓ (PSPACE-compl.)
$\mathcal{L}(\mathcal{A}/\mathrm{A}) = \varnothing$?	✓	✓	×
$\mathcal{L}(\mathcal{A}/\mathrm{A}) \subseteq \mathcal{L}(\mathcal{B}/\mathrm{A})$?	✓	×	×

図2 決定可能性と計算量 §

§ PRESTO 実施中に得られた成果は**青色**で示します.

理論的な研究成果のさらに詳しい解説は省略するが，個々の技術的な研究成果については，すでに研究論文としてまとめて，その多くは国際研究集会にて発表している．また近年，等式付ツリーオートマトンに関する研究が，世界的な広がりをみせており，最新の研究成果はおもに以下の国際研究集会などで報告されている：

- International Conference on Rewriting Techniques and Applications（RTA）
- International Conference on Logic for Programming, Artificial Intelligence and Reasoning（LPAR）
- International Conferences on Foundations of Software Sciences and Computer Structures（FOSSACS）

いずれの会議も，会議録が Springer-Verlag 社の LNCS シリーズとして出版されている．

4．2　自動計算についての研究

等式付ツリーオートマトンの理論を生かしたシステム開発をすることにより，検証の自動化についての検討を行った．本研究で開発したのは，結合則交換則付ツリーオートマトンを入力として，各種の演算や判定問題の解消を行うためのシステムである．オートマトンの基本的な演算である，共通集合や和集合（∪，∩），所属判定や空判定（∈，≠∅）の機能を備えていることから，ACTAS（Associative Commutative Tree Automata Simulator）と命名した．ACTAS では，結合則交換則付ツリーオートマトン A と結合則交換則書換系 R を与えて，A の受理言語の R による書換閉包を受理する結合則交換則付ツリーオートマトンを求めることができる．図 3 は，実際に ACTAS で書換閉包を計算したときに表示されるインターフェース画面である．

書換系とは，書換規則の有限集合で，書換規則は項の順序対で，$l \to r$ と書く．項 t と書換規則 $l \to r$ が与えられたときに，t の中に l のパターンにマッチする部分項が存在したとき，その部分項は r に置き換えられ，この関係を書換関係と呼ぶ．項 t から項 t' に項書換え系 R に含まれる規則で書換えられるとき，$t \to_R t'$ と書く．また，t から 0 回以上の書換えで t' に到達するとき，$t \to_R^* t$ と書く．結合則交換則付ツリーオートマトンは，結合則交換則付書換系の特殊なクラスとみなすことができることから，書換系とツリーオートマトンは，理論的な親和性がよく，書換系の研究成果をツリーオートマトンへ応用することも容易である．実際に，ツリーオートマトンの各種演算や問題解消系を利用して，書換閉包を計算することができた．ツリーオートマトン A の受理言語を L と表すとき，L の R による書換え閉包というのは，L に含まれる項から R によって書換えて得られる項全てからなる集合 $\{ t \mid s \to_R^* t, s \in L \}$ である．書換閉包の計算手続きは，一般に停止性を保証することはできないため，ACTAS では結合則交換則付ツリーオートマトン A と結合則交換則書換系 R を与えられたときに，(1) A の受理言語の R による書換閉包，(2) A の受理言語の R による書換閉包を含む集合（強近似書換閉包），(3) A の受理言語の R による書換閉包に含まれる集合（弱近似書換閉包）のいずれを計算するのか，を選択可能にした．特に，弱近似を行うアルゴリズムでは，いくつかのパラメータを指定することで，現実的な時間で計算の実行を終了させることが可能である．書換閉包の計算と同様に，空判定も弱近似判定や強近似判定が適用可能であるため，これらの機能を組み合わせて，モデル検査を実現した．暗号プロトコルの安全性自動検証への適用に関する考察は，論文 2（国際研究集会）にまとめた．現在は，研究交流のあるイリノイ大学計算機科学科研究者らと，検証用計算ライブラリを開発

し，ACTAS への統合を計画している．詳しくは，http://texas.cs.uiuc.edu/ceta/ を参照されたい．

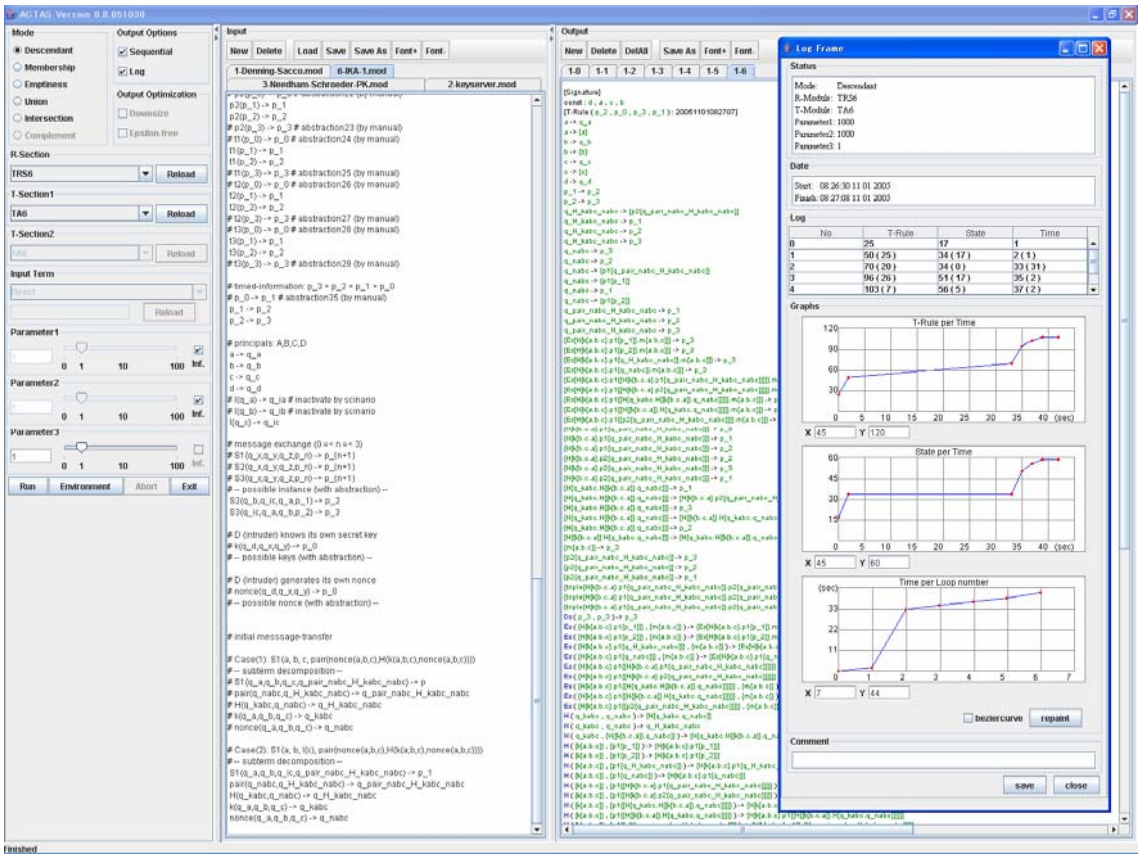


図3　ACTAS インターフェース

5　自己評価：

本研究中に開発した暗号通信手順の安全性自動検証法は，リアクティブシステムと呼ばれる「動作中に外界からの多様な刺激を受けて，その刺激と内部状態から応答を決定する」装置全般に適用可能である．線形等式による動作制約をもつシステムなどをモデル化することができ，近似アルゴリズムを用いれば，ある程度の規模のシステムであっても自動検証可能である．また，等式付ツリーオートマトンに関する最近の研究では，本研究者のアイデアとしてだけにとどまらず，現在では，構造化文書の検証，型推論エンジンの設計，プログラムの整合性検査など，様々な方面に活用する試みがなされている．本研究の成果は，情報科学の問題に端を発しながら，さらに今後は一般の情報システムの安全性や整合性検査へ適用することが検討されことになると考える．

リアクティブシステムの代表的な例である，銀行のオンラインシステムや携帯電話等の通信システムなどは，いちど稼働させてしまうと容易に停止させることが出来ない．安全で安定した情報システムの整備が社会的な急務となっている現在，稼働前に十分な安全性の検証をおこなうことへの社会的ニーズに応えることを，今後の研究目標としたい．また，大規模システムやマスプロダクトに対しては，検証で発見された誤りにより設計変更を余儀なくされた場合に，損失を極力小さく抑える必要があるため，

設計の初期段階で検証できなければならないという要求もあるため，業種を問わず適用可能な検証技術としていくための技術汎用化も今後の研究課題である.

6 研究総括の見解:

安心性や安全性は，今日の情報システムにとって最も重要な要件であるが，大崎氏の研究は情報科学の基礎技術である形式仕様化や形式検証技術によりこの問題の解決を図ろうとするものである. 大崎氏は，従来のツリーオートマトンを拡張した新しい等式つきツリーオートマトンの概念を独自に考案し，その理論展開を図ると同時に，それにもとづいて形式検証のためのツール群を構築し，リアクティブシステムのための自動検証システムを開発した. 等式つきツリーオートマトンは国際的にも高く評価され，新しい研究分野として認識されている. 国際的共同研究を活発に行い，成果を着実にあげると同時に，著名な国際会議での論文発表やチェアー，プログラム委員などを務め，国際的評価も高い. 非常に高く評価できる研究である.

7 主な論文等:

論 文(国際研究集会)

1. 大崎人士，Jean-Marc Talbot，Sophie Tison，Yves Roos:“Monotone AC-Tree Automata”. In proceedings of 12th International Conference on Logic for Programming, Artificial Intelligence and Reasoning(LPAR'05)，Montego Bay(Jamaica). Lecture Notes in Computer Science 3855 巻，337-351 頁，Springer-Verlag，2005.
2. 大崎人士，高井利憲:“ACTAS: A System Design for Associative and Commutative Tree Automata Theory”. In proceedings of 5th International Workshop on Rule-Based Programming(RULE'04)，Electronic Notes in Theoretical Computer Science，124 巻，97-111 頁，Elsevier Science，2005 .
3. 大崎人士，関浩之，高井利憲:“Recognizing Boolean Closed A-Tree Languages with Membership Conditional Rewriting Mechanism”. In proceedings of 14th International Conference on Rewriting Techniques and Applications (RTA'03)，Lecture Notes in Computer Science，2706 巻，483-498 頁，Springer-Verlag，2003.
4. 大崎人士，高井利憲:“Equational Tree Automata: Towards Automated Verification of Network Protocols”. 京都大学数理解析研究所講究録，1318 号，48-52 頁，京都大学，2003.
5. 大崎人士，高井利憲:“Decidability and Closure Properties of Equational Tree Languages”. In proceedings of 13th International Conference on Rewriting Techniques and Applications (RTA'02)，Lecture Notes in Computer Science，2378 巻，114-128 頁，Springer-Verlag，2002.
6. Joe Hendrix，大崎人士，Mahesh Viswanathan:“Propositional Tree Automata” . 論文草稿，2006.(国際会議投稿中)

論 文(その他)

7. 大崎人士，Joe Hendrix，José Meseguer:“Sufficient Completeness Checking with Propositional Tree Automata”.

産業技術総合研究所研究速報 AIST-PS-2005-013, 産業技術総合研究所, 2005.

システム紹介

8. 大崎人士, 高井利憲:“ACTAS: Associative and Commutative Tree Automata Simulator”. 4th International Conference on Application of Concurrency to System Design（ACSD’04), 2004.

研究交流

1. Université des Sciences et Technologies de Lille（リール・フランス;2005 年 5 月 21 日－6 月 15 日）.
2. École Normale Supérieure de Cachan（パリ・フランス;2004 年 8 月 28 日－9 月 30 日）.
3. University of Illinois at Urbana-Champaign（アーバナ・イリノイ州;2004 年 1 月 10 日－3 月 31 日）.
4. 京都大学数理解析研究所（京都;2004 年 7 月 26 日－7 月 30 日）.

外部委員

1. Program Committee Member:18th International Workshop on Unification（UNIF’04）, Cork（Ireland）,（2004 年 7 月開催）.
2. Program Committee Member:16th International Conference on Rewriting Techniques and Applications（RTA’05）, 奈良市,（2005 年 4 月開催）.
3. Organizing Chair:16th International Conference on Rewriting Techniques and Applications（RTA’05）, 奈良市,（2005 年 4 月開催）.
4. Conference Co-Chair:Federated Conference on Rewriting, Deduction and Programming（RDP’05）, 奈良市,（2005 年 4 月開催）.

など.

研究課題別評価

1 研究課題名:

大規模分散アルゴリズム開発及び性能評価のツール構築

2 研究者氏名:

Xavier DÉFAGO

3 研究のねらい:

The objective of the research is to better understand the performance tradeoffs associated with fault-tolerant mechanisms for distributed systems. In particular, group communication protocols, such as Total Order Broadcast, are key factors in determining the performance of the system in the absence of failures. While failure-free executions constitute the common case, the occurrence of failures should not affect system performance too drastically, or else failures risk being perceived by the users, thus defeating the objective of masking them. The performance in the face of failures depends mostly on the ability of the system to detect failures promptly and accurately, but this is made difficult by an inherent tradeoff between these two measures. Thus the second objective is to provide a generic failure detection service, the speed and accuracy of which can be best tuned to the specific needs of each part of the entire distributed system.

4 研究成果:

In this research, we have made three major contributions.

Firstly, we have studied several group communication protocols, with a particular focus on the problem of Total Order Broadcast (also called Atomic Broadcast) because it is an important component for many kinds of practical systems, including distributed databases, distributed shared memory, highly-available replicated services, etc.

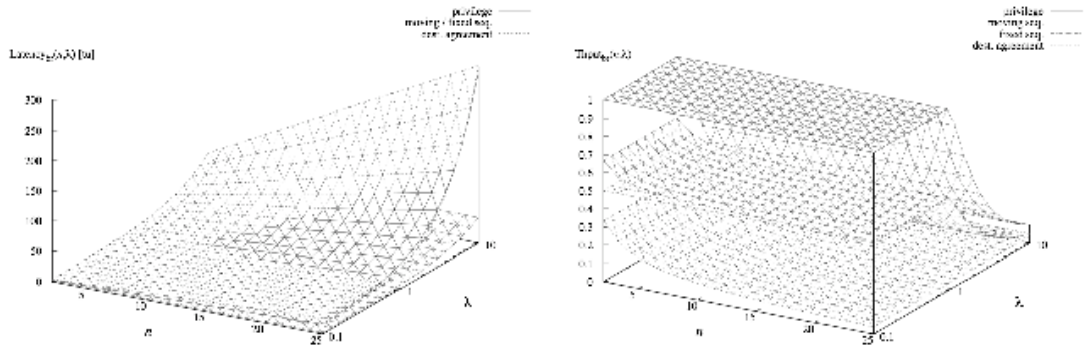
Secondly, we have proposed a novel failure detection method, called accrual failure detectors, with the ultimate goal of providing failure detection as a generic and highly-configurable service for distributed systems. Studying failure detection is also a prerequisite to understand the actual performance of group communication protocols in the face of failures.

Thirdly, we have developed a communication platform to provide a better support for the evaluation of distributed algorithms.

4.1 Group Communication and Distributed Agreement

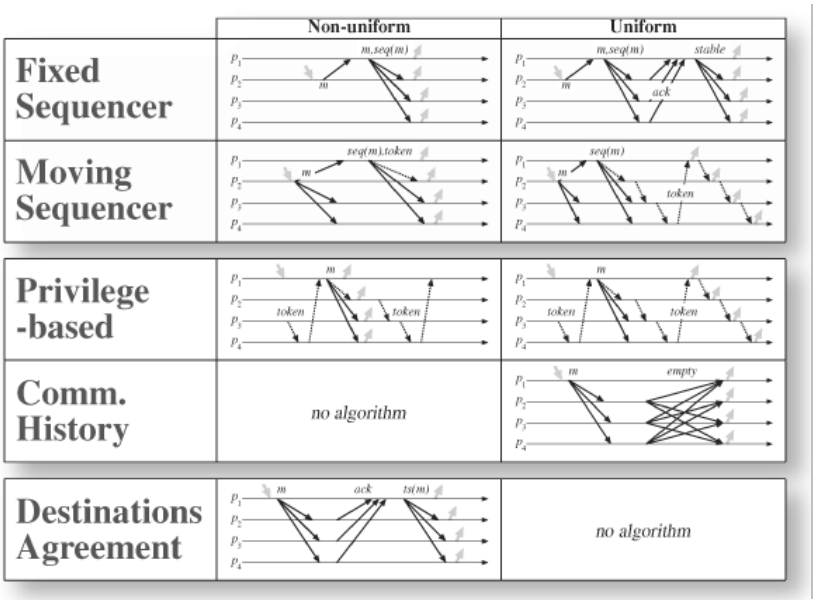
Group communication and distributed agreement are basic primitives to maintain the cohesion of a distributed system, by allowing the participating nodes to agree on some common issues. A very practical instance of agreement problems is a communication primitive called *Total Order Broadcast* (also Atomic Broadcast). In short, this primitive allow any of the processes to broadcast

messages any time, but guarantees that all destinations will always see (and process) the messages in the same exact order. Among other things, Total Order Broadcast is a key component for supporting the replication of running programs and services (e.g., web or Grid services). Indeed, assuming that all replicas have the same initial state (i.e., value of variables, etc.), then Total Order Broadcast is used to issue requests to the service. Because of the guarantees offered by the primitive, all replicas perform the same actions in the same sequence, and thus their state change in exactly the same way. The benefit is that the replicas remain exact copies of each others, and thus the service can remain operational even after the crash of some of the replicas, that is, provided that the Total Order Broadcast can actually tolerate the crash of some of the processes.



There exist many algorithms to solve Total Order Broadcast, most of which can tolerate failures. However, they do not offer exactly the same guarantees, and their respective performance can vary drastically. We have thus surveyed and analyzed about sixty different Total Order Broadcast algorithms [1], and identified five basic families and a total of thirteen subclasses. The basic families, defined on the decision process used to generate the delivery order, are called *fixed sequencer*, *moving sequencer*, *privilege-based*, *communication history*, and *destinations agreement*.

Using this classification, we have defined representative algorithms for each of the class, and compared their performance and scalability [3] in different network environment. We have also defined a novel replication technique using a variant of a destination agreement Total Order Broadcast algorithm [2].



4.2 Accrual Failure Detectors

Failure detection plays an essential role in ensuring fault tolerance in distributed systems. Recently, many people have come to realize that failure detection ought to be provided as some form of generic service.

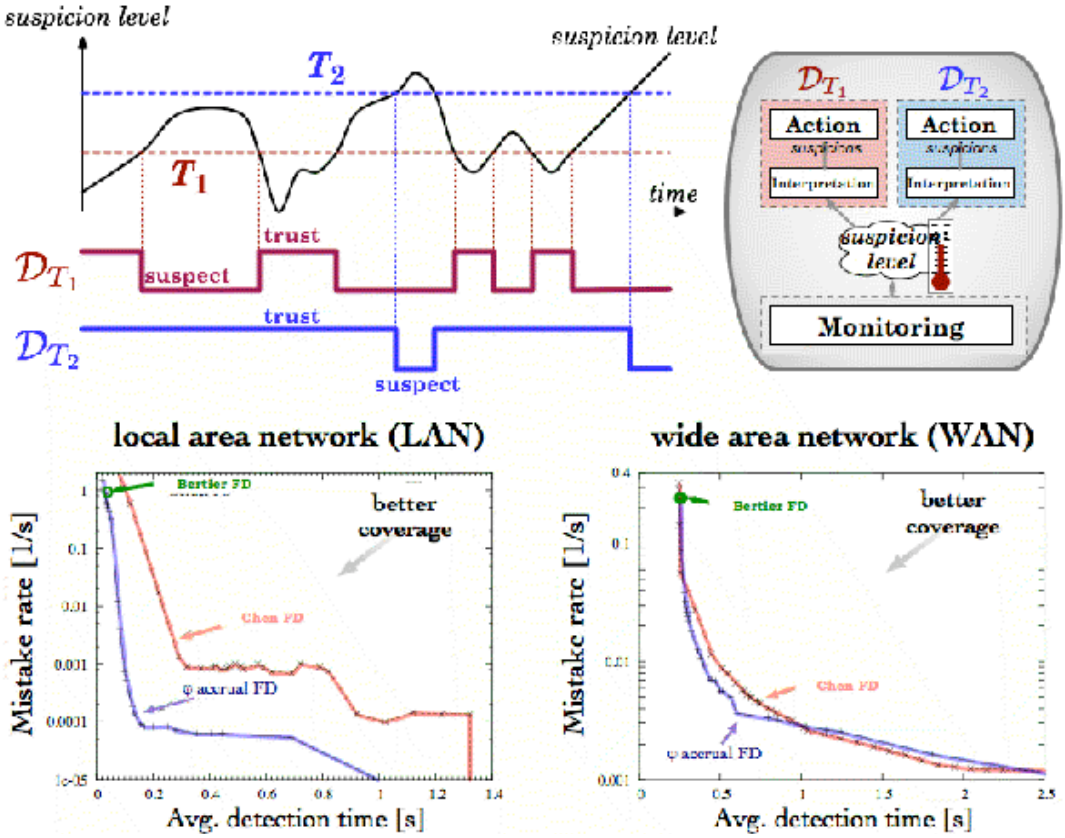
The performance of fault-tolerant distributed systems, and their ability to mask failures from the viewpoint of their users, depend greatly on the characteristics of the failure detection. This is especially true when it comes to group communication algorithms, such as the Total Order Broadcast discussed in the previous. When considering a complete system, it turns out that failure detection is required at many different levels, but often with very different performance requirements. Roughly speaking, the performance of a failure detector can be expressed by two measures: *detection latency* (i.e., how long it takes until a real failure is detected) and *accuracy* (i.e., how often a running process is erroneously suspected).

In conventional systems, there is always a tradeoff between detection latency (also called detection time) and accuracy, but different operations will benefit from very different settings. For instance, many Total Order Broadcast algorithms will have best overall efficiency with a failure detection latency in the order of a hundred milliseconds, even if the detection is often erroneous. In contrast, a global system reconfiguration will benefit from more accurate detection, even if this leads to a considerable detection latency. It is thus difficult to provide a failure detection scheme that simultaneously provides ideal performance for both.

To address this problem, we have developed the notion of accrual failure detector, promoting a clean decomposition of roles and formally establishing the link with the basic theory on failure detection [4]. Instead of a binary value (trust or suspect), accrual failure detectors associate to each process a real value representing a *suspicion level*. By establishing the clear link with the theory on failure detection, we have identified minimal properties whereby the failure detection

scheme can be used for solving distributed agreement problems, such as Total Order Broadcast and Consensus.

We have developed several implementations of accrual detectors and, in particular, a highly adaptive one called φ accrual failure detector [7]. We have conducted extensive performance measurements, comparing the performance of our φ accrual failure detector with that of other state-of-the-art failure detection schemes. Our results have shown that, for the same latency, our scheme could perform up to ten times more accurately in a local network. Our experiments on an intercontinental network (between Japan and Switzerland) have shown comparable performance in spite of the change of interaction scheme, thus effectively showing the practicality of our approach.



4.3 NekoLS Prototyping Platform

To conduct our experiments more efficiently, we have developed a communication platform called NekoLS. This platform is an extension of an earlier system of us, called Neko, that allowed an easier development of distributed algorithms, and with which the same code can be executed either in a real network environment or on a single machine, within a simulated network. We have made many improvements of this system, the most notable of which is a seamless integration with the SSFNet project; a network simulator aimed for describing large and complex network topologies.

For describing algorithms, the Neko platform is based on a simple layered architecture. While this choice was good for describing simple protocols, it turned out that even moderately complex protocols were very difficult to design elegantly, due for a large part to the difficulty to prevent deadlocks in the protocol. To address this issue, we have been working on a novel mechanism for the composition of micro-protocols.

5 自己評価:

Our research focused on ensuring fault-tolerance in large-scale distributed systems. In particular, we examined three fundamental aspects in relation to this goal: fault-tolerant agreement, failure detection, evaluation/prototyping.

Fault-tolerant agreement is a fundamental component in distributed systems. In particular, we have studied closely the problem of Total Order Broadcast (i.e., broadcast with the guarantee that all messages are received by all destinations in the same global order). We have analyzed about 60 algorithms; identifying their exact guarantees, and defining classes of algorithms [1]. Based on this classification, we have analyzed the performance tradeoffs of these algorithms [3]. In parallel, we have used our understanding of the mechanisms for reaching agreement and developed a novel replication algorithm called semi-passive replication, that is uses less resources than active replication schemes while being more robust than passive ones [2].

When analyzing the performance of agreement problems, it turned out that their performance in the face of computer failures is completely dependent on the performance of an underlying failure detection mechanism. While most systems use simple timeouts mechanisms and detect failures within a few minutes, our research has lead to failure detection that can detect failures in less than a second, even with a world-wide system. In particular, we have developed several failure detection mechanisms of which we have analyzed the performance in local, as well as wide area networks. One such mechanism is the ϕ accrual failure detector [5]. Realizing that fault-tolerant systems actually need to rely on several failure detectors with different performance guarantees, we have defined formally the notion of accrual failure detectors, about which we could derive interesting properties regarding the actual quality of service (i.e., the performance) of the failure detection [4]. These results will provide the basis for building a generic failure detection and monitoring system for large-scale systems.

The NekoLS platform has been very useful for running our experiments and performance analyses, thus meeting its initially intended purpose. In addition, with the experience obtained when building the tool and running the experiments, we have been able to find a much better approach to express distributed protocols in general, so that they can be reusable. That approach consists of an advanced protocol composition mechanism. We have obtained very interesting results about this recently, that are currently under submission to a major international conference.

To sum up, this research project has been very productive in terms of scientific results, with many important contributions to the field of fault-tolerant distributed systems. Most of these

contributions have been published in major international conferences or journals.

6 研究総括の見解:

Defago 氏の研究は，大規模分散システムの高信頼，耐故障性に関するものである．このようなシステムでは，それを構成する計算ノードやネットワークが故障することは日常的であり，このような故障に際してもシステム全体が動作し続けることが重要である．Defago 氏は，最も基本的な全順序ブロードキャスト通信プロトコルのためのアルゴリズムの詳細な検討と分類を行い，新しいアルゴリズムを得ている．また，誤り検出方式に関しては，従来よりも高性能で応用の広い Accrual 誤り検出器を発明した．さらに，これらの成果を実験的に検証するためのシミュレーション環境 NekoLS の開発を行い，これらの方式の有効性を確認している．これらは，国際的評価の高い論文誌や専門家会議の招待講演などで発表されている．高く評価できる研究である．

7 主な論文等:

International Journals

1. X. Défago, A. Schiper, and P. Urbán, Total order broadcast and multicast algorithms: Taxonomy and survey, *ACM Computing Surveys*, 36(4):372-421, December 2004. ACM Press.
2. X. Défago and A. Schiper, Semi-passive replication and Lazy Consensus. *Journal of Parallel and Distributed Computing*, 64(12):1380-1398, December 2004. Elsevier.
3. X. Défago, A. Schiper, and P. Urbán. Comparative performance analysis of ordering strategies in atomic broadcast algorithms. *IEICE Trans. on Information and Systems*, Vol.E86-D, No.12, pp.2698-2709, December 2003.

Refereed International Conferences

4. X. Défago, P. Urbán, N. Hayashibara, T. Katayama. Definition and specification of accrual failure detectors. In *Proc. IEEE/IFIP Intl. Conf. on Dependable Systems and Networks,* pp. 206-215, June 2005. IEEE CS Press.
5. N. Hayashibara, X. Défago, R. Yared, and T. Katayama. The ϕ accrual failure detector. In *Proc. 23rd IEEE Intl. Symp. on Reliable Distributed Systems*, pp. 66-78, October 2004. IEEE CS Press.
6. M. Wiesmann, X. Défago, and A. Schiper. Group communication based on standard interfaces. In *Proc. 2nd IEEE Intl. Symp. on Network Computing and Applications*, pp.140-147, April 2003.

(and many other...)

Important Invited Presentations

2005 年 12 月 5 日 "*Failure Detection in Distributed Systems: Retrospective and recent advances*." **Tutorial**. 6th Intl. Conf. on Parallel and Distributed Computing, Applications and Technologies.

2005 年 7 月 2 日 "*Revisiting Failure Detection for Grid Systems.*." **Invited talk.** 48th meeting IFIP working group 10.4 (dependable computing & fault-tolerance).

研究課題別評価

1 研究課題名:

広域分散共有メモリ機構を支援する最適化コンパイラ

2 研究者氏名:

丹羽 純平

3 研究のねらい:

近年急増している新世代 E-Science アプリケーションは、大量の計算機パワーを必要としており、従来のスーパーコンピュータを多少増強した程度では扱いが困難であり、コストパフォーマンスの点からも非現実的であるという点があげられます。コストパフォーマンスの面では、研究機関内にある計算機資源をネットワークで接続した LAN クラスタが優れているものの、如何せん、計算機パワーの不足は否めません。一方、ネットワーク技術の進歩やスイッチング技術の向上により、SuperSINET を初めとする国内超高速バックボーンネットワークが急速に整備されてきました。そして、国際間の超高速バックボーンネットワークも整備されつつあります。その結果、研究機関の計算資源が超高速でつながれ、計算資源を広域的に利用した"次世代実験科学のための計算プラットフォーム" の構成(図1)がコストパフォーマンスの面からも現実味を帯びたものとなりつつあります。

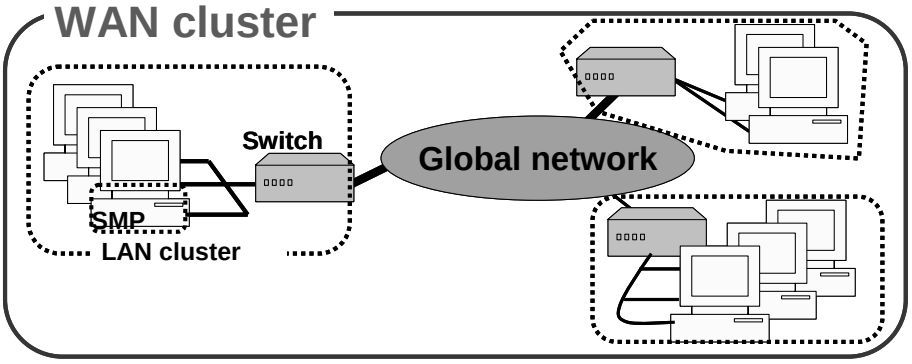


図1 WAN クラスタの構成

広域環境にある計算機資源をフルに活用するには、ユーザ自ら並列プログラムを記述することが求められます。共有メモリモデルは従来のスタンドアローンなプログラミングの自然な拡張であり、メッセージパッシングモデルに比べて、並列プログラムを記述しやすいという利点を持っています。もちろん、共有メモリモデルに従って記述された並列プログラム(例:OpenMP や PARMACS で拡張された C のプログラム)を、直接メッセージパッシングコード(分散計算機上のコード) に変換することは可能です。しかし、幅広いクラスのアプリケーションを効率良く扱うためには、アプリケーションが実行時に直接共有アドレスを扱えること(例:タスクキューを用いた動的負荷分散)が求められます。すなわち、実行時にシステム全体で仮想的に共有メモリを提供する機構:ソフトウェア分散共有メモリ機構(S-DSM)が必要になります。

そこで、効率の良いソフトウェア分散共有メモリ機構を構築できるかどうかが鍵になってきます。これまで、LAN 上の分散環境では、オペレーティングシステムの支援と最適化コンパイラの支援とランタイムの支援があれば、高性能な S-DSM を構築することが可能であることが示されてきました。

この結果から、超高速 WAN 上であっても、最適化コンパイラとランタイムの支援があれば、共有メモリモデルに従って記述された並列プログラムを効率良く実行できるのではないかと考察しました。もちろん、WAN は LAN と比較して様々な問題を抱えています。例えば、通信の遅延(レイテンシ)が大きいといった点が挙げられます。また、大量の共有メモリを必要とするアプリケーションが多いという点も挙げられます。更に、計算機の台数が多くなるために、一部の計算機が故障する可能性が増大するので、効率的な耐故障機能が必要となります。

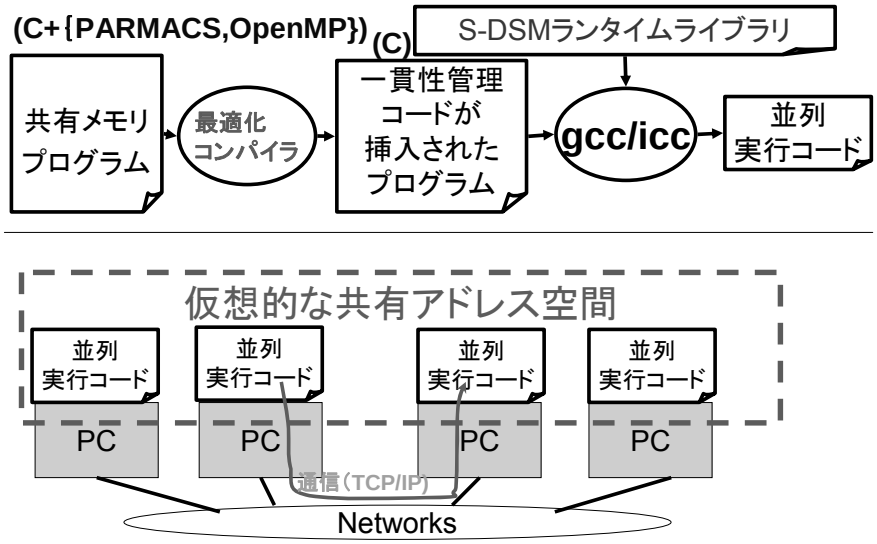


図2 本研究が提案する広域分散共有メモリ機構のイメージ

本研究では、こういった問題点を克服するコンパイル時最適化と実行時最適化を提案し、それらの最適化を可能とするインタフェイス(WDSM)を導入しました。実際にシステムを構築し、本手法の有効性を評価・検証してきました。図2は、本研究が提案するコンパイラが支援する広域分散共有メモリ機構のシステムを簡単に表しております。

4 研究成果:

4. 1 遠隔アクセスのレイテンシ削減

1)クラスタキャッシュ

共有メモリ機構の性能を向上させるためには、良く使用するデータは手元にキャッシュしておく必要があります。図1にあるように、WAN クラスタには3つの階層があり、PC 内部/LAN 内部/WAN 内部と分類されます。PC 内部では、CPU とメモリの速度のギャップを埋めるために、ハードウェアによる1次/2次キャッシュがあります。LAN 内部では、LAN アクセスとメモリアクセスの速度のギャップを埋めるために、ソフトウェアキャッシュがあります。本研究では、WAN 内部において、WAN のアクセス速度と LAN のアクセス速度のギャップを埋めるために、クラスタキャッシュを提案しました。

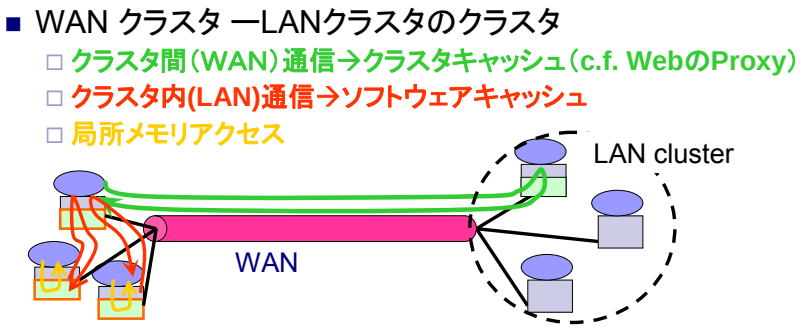


図3　クラスタキャッシュ

図3にあるように、LAN クラスタ（図左）は遠隔 LAN クラスタ（図右の点線で囲まれた部分）にあるデータを自分のクラスタにキャッシュすることで、できるだけ WAN 上の通信（図の緑色の部分）を起こさないようにします。以下に簡単に説明します。ある PC が遠隔クラスタにあるデータに初めてアクセスする場合、WAN 上の通信によりそれを入手します。しかし、それ以降、その PC と同じクラスタ内にある別の PC が同じデータにアクセスする場合には、遠隔クラスタから取ってくるのではなく、同一クラスタ内のクラスタキャッシュから取ってきます。全ての PC が同一データにアクセスするというのは並列計算では良く見受けられるので、この最適化は非常に効果的であるといえます。バリア同期に関しても、クラスタキャッシュと同様の最適化を適用しました。システムの階層性に着目して、LAN 内部でバリア同期を取ってから、WAN 内部でバリア同期を取るようにすることで、WAN 上の通信の回数を減少させました。

2）プリフェッチ機構

クラスタキャッシュ最適化は無駄な WAN 上の通信を削減する手法とみなすことができます。どうしても必要な一回目の WAN 上の通信（図3の緑色の通信）の遅延を削減するために、本研究ではアグレッシブなプリフェッチ機構を提案しました。プリフェッチは本来 CPU とメモリのスピード差を埋めるために開発された手法で、データの使用前にノンブロッキングなメモリアクセス要求を発行します。

WAN の通信遅延を考えると、少し先読みするだけでは不十分であるといえます。そこで、プログラムの意味を変えることなくできるだけ早くかつ無駄なくデータの要求を行う手法を提案しました。

1つ目は、部分冗長性削除のフレームワークを活用して、if 文や case 文のような条件節があってもプリフェッチをさかのぼって発行できるようにしました。図4はコンパイラによるコード生成例です。配列 a への読み出しは、従来の手法だと、if 節の先頭に挿入されていましたが、今回提案した手法では、更に遡ってバリア同期の直後に挿入されます。無駄な通信を誘発しないように、条件節つきのプリフェッチとなります。

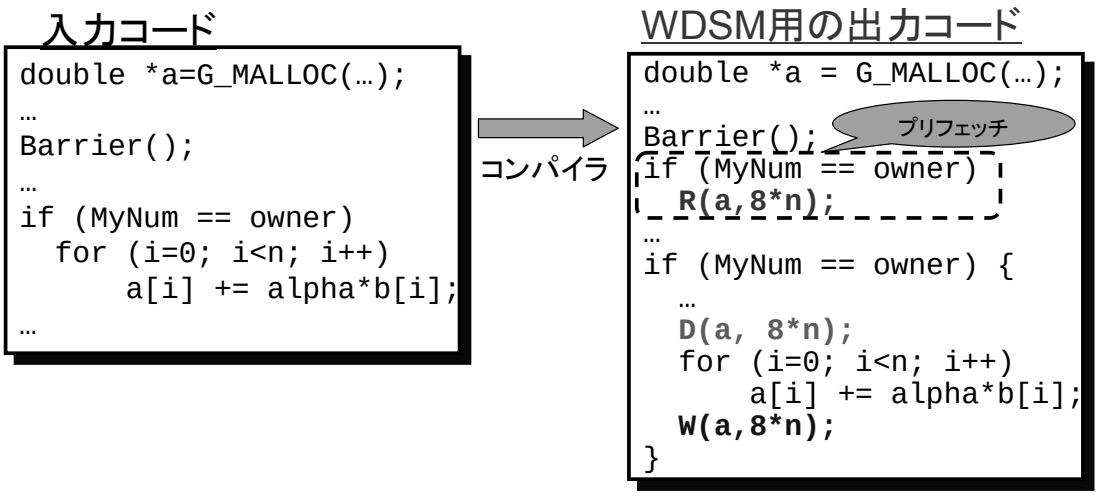
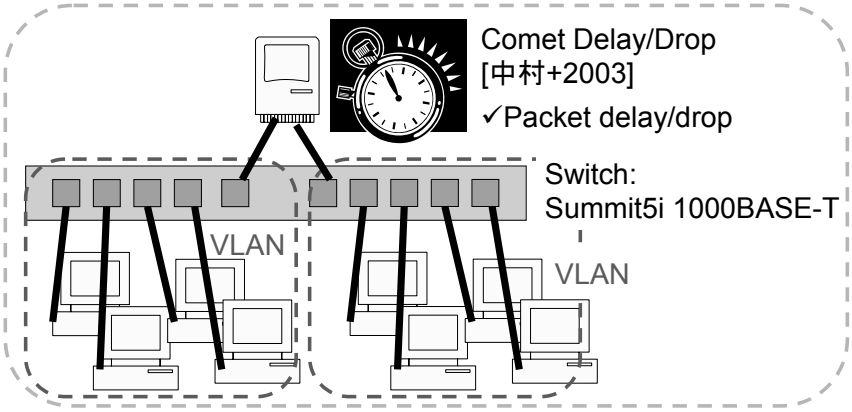


図4 コード生成例

2つ目は履歴の活用です。WDSM ではキャッシュミスを起こしたページ番号の履歴を残すことが可能です。そこで、バリア同期を使用して毎回同じ領域にアクセスするような反復計算(いわゆるループ)に関しては、履歴を活用することで更に通信の遅延を削減することができます。すなわち、最初の一回目のイテレーションでキャッシュミスを発行したページ番号を記録しておき、二回目以降のイテレーションでは、その情報を使用して、バリア同期の際にキャッシュブロックリクエストをまとめて発行することで通信の遅延を削減します。二回目以降のイテレーションでは、キャッシュのミス／ヒット判定も不要となるので，そのオーバヘッドも削減できます。

実 験

本方式の有効性を確認するために、以下の擬似広域環境で実験を行いました。図5に示してあるように、Summit5i を利用して VLAN を作成し、VLAN 間同士で通信する場合には、Comet Delay./Drop を経由するように設定しました。この Comet Delay/Drop が、ユーザが指定した間隔でパケットを遅延させたり、落としたり、バンド幅を制限したりします。WAN クラスタ(8台)は、2つの LAN クラスタ(4台)から構成されます。実行されるアプリケーションのサイズを考慮して、メトロポリタン地域で想定される通信遅延(0～10ミリ秒)を入れて実験を行いました。



Node:Dell PowerEdge1650, 1.26GHz Pentium III (512KB cache)
2GB Memory, 1000BASE-T

図5　仮想広域環境

使用したアプリケーションは SPLASH2 の密行列の LU 分解です（行列のサイズは 4096×4096）です。図6が実験結果を示しています。WDSM というのが今回提案した方式で、ADSM というのが従来の LAN で使用されてきた方式です。

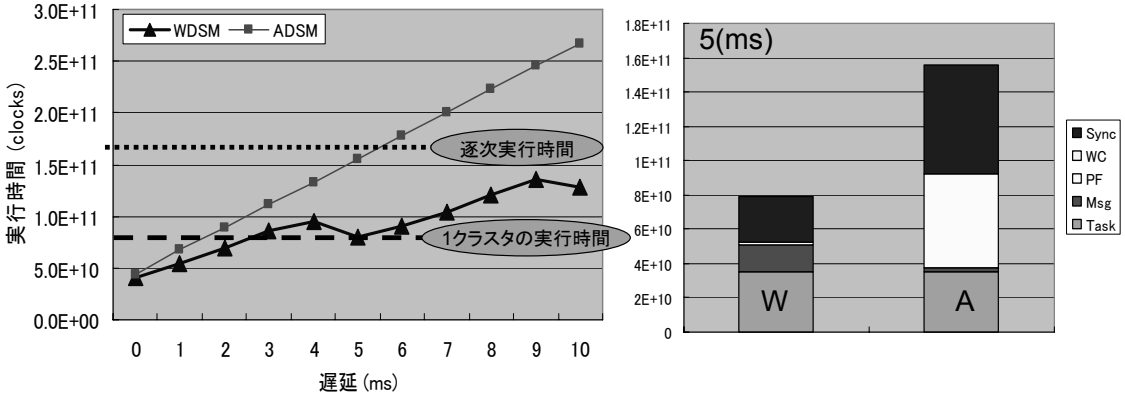


図6　遅延と実行時間の関係（左）と 遅延が 5（ms）の時の実行時間の内訳（右）

図6左から、WDSM は ADSM よりも遅延に対してロバストであることが分かります。図6右の棒グラフの各成分について簡単に説明します。Msg はメッセージの処理時間、PF はキャッシュミスの待ち時間、WC は書き込み後のコヒーレンス管理コードの実行時間、Sync はバリア同期の実行時間を表しています。残りの時間が Task になります。すなわち本来の計算時間と、プリフェッチの実行時間とを合わせたものです。左側の棒グラフが WDSM の結果で、右側のそれが ADSM の結果となります。WDSM は ADSM と比べて PF（キャッシュミスの待ち時間）が大幅に削減されていることが分かります。以上の結果から本研究で提案した最適化方式は効果的であるということができます。

4．2　スケーラブルな共有メモリ

32 ビットアーキテクチャの場合には、論理アドレスが 32 ビットしかないので、アドレス空間として 4 GB しか使用できません。すなわち、計算機を何台接続させたとしても、共有アドレスは最大 4 GB まで

しか確保できません。広域環境上で動作させたいアプリケーションの性質を考えると、4 GB では十分であるとはいえません。近年、64 ビットアーキテクチャ（例：AMD64, EMT64, Potwer5）の低価格化と高性能化が進んだことを受けて、本研究では、64 ビットアーキテクチャを計算機として仮定しました。64 ビットアーキテクチャの場合、仮想メモリサイズはほぼ無限大なので、計算機の台数に比例してスケーラブルに共有メモリを確保することができます。しかしながら、共有メモリサイズが物理メモリサイズを超える場合には、適宜（ソフトウェア／クラスタ）キャッシュをリプレイスする必要が生じます。

本研究では、キャッシュをロバストにリプレイスする手法を提案し、実装しました。ホーム（オリジナル）はリプレイスの対象にはなりません。無効なキャッシュは同期ポイントですでにアンマップされているので、あらためてリプレイスする必要はありません。有効なキャッシュがリプレイスの対象となります。その際、読み出しのみのキャッシュは、ただアンマップするだけで良いのですが、書き込まれた（る）キャッシュは、ただアンマップするだけではプログラムの正確性が失われる危険があります。すなわち、書き込んだ(書き込む)データだけ、ホームに書き戻してから、アンマップするようにします。

そのために、本研究では、書き込み前のコヒーレンス管理コードを導入しました。図4のコード生成例に示してあるように、配列 a に実際に書き込む前に、これから書くというコード（D(a, 8*n)）を挿入します。

書き込み前のコヒーレンス管理コード D は書き込まれるブロックのアドレスとサイズ（図4の例では a と 8*n）をリストに登録します。書き込み後のコヒーレンス管理コード W は書き込まれたブロックをホームに転送します。W はその際に書き込まれたブロックのアドレスとサイズの組をリストから削除します。キャッシュリプレイスが発生した場合には、リストに残っている組に関して、対応するブロックをホームに書き戻すことで、プログラムの正しさが保障されることになります。

実 験

本方式の有効性を確認するために以下の環境で実験を行いました。

1）PC：AMD Opteron242 (物理メモリ4GB) 8 台

2）ネットワーク：ギガビットネットワーク（NetGear GS524T）

3）OS：SUSE Linux9.0

各 PC ですが、ホームとして 2 GB 使用し、キャッシュとして 2 GB 使用するように設定しました。その結果、システム全体として、16 GB の共有メモリを扱うことができました。その上で、N 体問題のシミュレーションを行うツリーコードを動作させました。粒子数は6億 4 千万で、ステップ数は 32 です。必要とされる共有メモリは 10 GB で、一台の PC では実行不可能な問題サイズであるといえます。その実行結果が表1に示されています。Msg はメッセージの処理時間、PF はキャッシュミスの待ち時間、WC は書き込み前／後のコヒーレンス管理コードの実行時間、Sync はバリア同期の実行時間を表しています。残りの時間が Task になります。すなわち本来の計算時間に、プリフェッチの実行時間を合わせたものです。

表1　ツリーコードの実行時間とそのブレイクダウン

Total	Task	Msg	PF	WC	Sync
865(分)	769(分)	1(分)	52(分)	5(分)	38(分)

このアプリケーションでは、参照の局所性が高いためにキャッシュのリプレイスは発生しませんでした。この表から、WC の実行時間は非常に小さいということがわかります。従来のシステム(ADSM)で問題の規模が 8 分の1(粒子数 8 千万)の問題を解いた所、実行時間は 93.4 分かかりました。ツリー法の計算量は O(*nlogn*)であることが知られているので、計算量の比と実際の実行時間の比を比べると、今回導入した書き込み前のコヒーレンス管理コードのオーバヘッドの大小の判別の指標となります。結果は両者ともほぼ9となり、書き込み前のコヒーレンス管理コードのオーバヘッドは十分小さいものであるということがわかります。

4．3　効率的な耐故障機能

全体の PC の台数が増加すると、それだけ故障する台数も増加します。もし、何も対策を講じなければ、計算の途中で一台の計算機に不具合が発生した場合には、全ての計算機で、最初から再計算することとなります。それを防ぐために、今まで計算しておいた結果を保存する必要が発生します。すなわち、ある時点でのプログラムの実行状態を保存しておいて(checkpointing)、後になって、そこから、プログラムを再実行可能なようにしておく必要があります。Checkpointing に関する研究は長年に渡って行われていますが、本研究では、できるだけ効率良く実装するということに力点を置きました。

本研究では、既存の逐次計算に対する耐故障ライブラリ:libckpt を改良しました。まず、Dynamic library に対応できるようにしました。また、ソケットやファイルも保存することができるようにしました。状態管理の容易性から、Coordinated checkpointing を採用しました。すなわち、バリア同期の時に、全ノードが同時に checkpointing を行うようにしました。ただし、バリア同期の都度、checkpointing していてはオーバヘッドが増大する危険があります。そこで、タイマー機能を付加することにより、一定の時間間隔があかないと checkpointing は行わないように制御しています。

実 験

本方式の有効性を確認するために以下の環境で実験を行いました。

1)PC:Dell PoweEdge1650　(P-III 1.26GHz, 2GB メモリ)　4 台

2)ネットワーク:ギガビットネットワーク(Summit 5i)

3)OS:FreeBSD5 .1

4)アプリケーション:SPLASH2 の LU 分解(行列のサイズは 2048×2048)

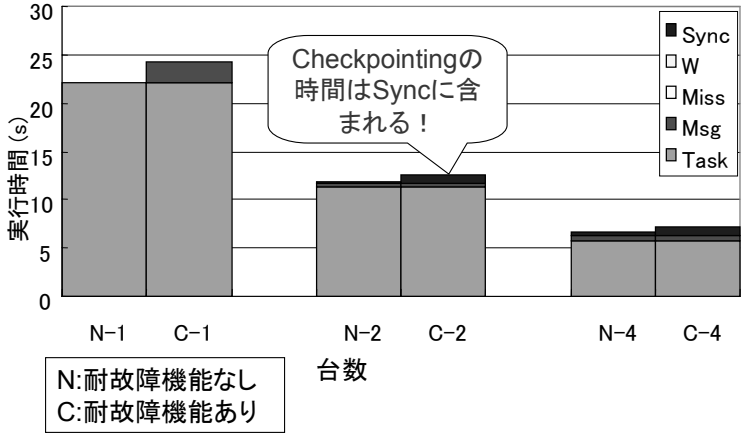


図7　Checkpointing のオーバヘッド測定

結果が図7です。横軸は台数、縦軸は実行時間を表します。2つの隣接している棒グラフの左(N-＊)が耐故障機能のない場合の結果で、右(C-＊)が耐故障機能のある場合の結果です。Msg はメッセージの処理時間、PF はキャッシュミスの待ち時間、WC は書き込み前／後のコヒーレンス管理コードの実行時間、Sync はバリア同期の実行時間を表しています。耐故障機能がついている方(右側の棒グラフ)では、checkpointing の時間は Sync に含まれています。残りの時間が Task になります。台数によらず、checkpointing のオーバヘッドは全実行時間の10%程度になっていることが分かります。

5　自己評価:

遠隔アクセスのレイテンシ削減に関しては、クラスタキャッシュとプリフェッチ機構の導入により、実際のアプリケーションに対して当初の予想以上の成果を収めることができました。プリフェッチに関しては、厳密にループ不変でなくても再帰的なアクセスをしている場合（例:N 体問題におけるツリー法でツリーをたどる場合）であっても、履歴の活用が有効であるということが分かりました。

今後は Grid 技術と融合して実際の広域環境上で実験を進めていきたいと思います。

スケーラブルな共有メモリの適用に関しては、当初は、低価格高性能な 64 ビットアーキテクチャが普及していなかったので、ソフトウェアによる64ビット共有アドレスから32ビット論理アドレスへの変換を想定していました。それに比べると64ビットアーキテクチャを使用することで、自然にオーバヘッドを削減できたと思います。どれだけ、キャッシュを確保するのがよいか？という点に関しては、論理的に最適な値を発見できませんでした。環境依存なので、実験によって求めていくしかないと思われます。

効率的な耐故障機能の提供に関しては、当初の予定通り、システムレベルのチェックポインティング（コアダンプ方式）の実装を行いました。どんなアプリケーションであれ、中断、再開することが可能になりました。実験を行うことで、実際のアプリケーションで鍵となるデータ構造の数はそんなに多くないということが分かりました。アプリケーションレベルのチェックポインティングの方が低オーバヘッドで実現できるのではないかと思われます。ただし、自動的に鍵となるデータ構造を発見するのは困難であるという問題点があります。そこで、ユーザからの指示を与えるという形を取ることで問題点を克服できるのではないか？と思います。 今後はシステムレベルとユーザレベルを融合したチェックポインティングを目指したいと思います。

6 研究総括の見解:

近年，大規模科学計算のために大量の計算機パワーが必要とされ，グリッド計算システムが注目されている．通常は高速LANで結合された計算機群が利用されるが，超高速広域ネットワークで結ばれた研究機関の計算資源を利用する次世代実験科学のための計算プラットフォームが今後重要になると考えられる．丹羽氏の研究は，広域環境中の共有メモリ並列プログラムを高速に動作させるための，最適化コンパイラやランタイムシステムの構築法に関する研究である．クラスタキャッシュやプリフェッチ機構，スケーラブル共有メモリ機構，効率的耐故障機能などの方式を考案，実装と評価を行い，それらの有効性を実証した研究である．このような研究は今後のさきがけとなるものであり，丹羽氏の研究は高く評価できるものである．

7 主な論文等:

論文

1. 丹羽純平，広域ソフトウェア分散共有メモリ機構を支援する最適化手法，
情報処理学会論文誌:コンピューティングシステム Vol.45 No. SIG 11(ACS 7) 36—49, 2004.年.
2. J. Niwa, Prefetch Mechanism in Compiler-Assisted S-DSM System,
CRTPC-04/ICPP-04, Montreal, Quebec, Canada, pp. 520—529, 2004/08/15--18.
3. J. Niwa, Compiler-Assisted Software DSM on a WAN Cluster,
Parallel and Distributed Computing: Applications and Technologies, LNCS vol.3320, pp. 815—828, (PDCAT 2004, Singapore, 2004).
4. 丹羽純平，コンパイラが支援するソフトウェア DSM におけるレイテンシ削減技法，
情報処理学会 ハイパフォーマンスと計算科学シンポジウム pp 81—88, 2005 年.
5. 丹羽純平，コンパイラが支援するソフトウェア DSM におけるレイテンシ削減技法，
情報処理学会論文誌:コンピューティングシステム Vol.46 No. SIG7 (ACS 10) 74—84, 2005 年

研究課題別評価

1　研究課題名：

プログラミング言語の制御構造の意味論的分析

2　研究者氏名：

長谷川 真人

3　研究のねらい：

現在用いられているプログラムおよびプログラムによって表現されたアルゴリズムは、一般に、とても複雑な構造をもっています。その構造は、多くの場合、プログラムの中で用いられるデータの構造、そしてプログラムの実行過程をつかさどる制御の構造の、巧妙な組み合わせから生みだされます。前者の、データ構造の特性を生かした効果的なアルゴリズム設計は、古くから計算機科学の中心的な話題のひとつでした。一方、プログラミング言語の制御構造の活用も、また古典的な問題ですが、高度な経験則や抽象的な思考を必要とするため、必ずしも広く深く理解されてきたとは言いがたいのが実情です。古くは goto 文のようなジャンプの使用の是非に関する議論が有名ですが、その一般化である継続（コンティニュエーション）、自己言及的なプログラミングを許す再帰、あるいはプログラムをデータとしてやりとりする高階のプログラムなど、いずれも表現力に富み、プログラムに明快な構造を与えうる高度な制御構造が、実際のプログラミングの現場では、むしろいたずらにプログラムをわかりにくくするものとして敬遠されたり、理論屋のおもちゃに過ぎないと揶揄されたりしているのが実情です。

確かに、高度な制御構造は、習得に時間を要します。理解不足のまま用いれば、いともたやすく「スパゲッティプログラム」の出来上がりです。けれども、正しく用いれば、プログラムの開発・保守・改良に際し、大変有効な道具です。使わないのはもったいない。問題は、その「正しく使う」ための指針が、必ずしも誰にでもわかるかたちで与えられていないことです。これは、単に良い教科書がないとか良い先生がいないなどということではありません。実のところ、プログラムの制御構造の相互作用をきちんと解き明かした理論がまだないのです。個々の制御構造を取り出してきちんと調べたり解説した例は数多くありますが、現実のプログラミング言語でおきている、様々な制御構造がお互いに干渉しあって生じる現象は、まだ理論的に十分には解明されてはいないのです。

本研究は、そのような、制御構造の間の関係を、プログラミング言語の基礎理論の最新の成果を駆使して、明らかにしようというものです。特に、継続、再帰、高階プログラム、そして多相型プログラムの関係に焦点を当て、プログラム間に成り立つ等式の理論を導きました。

4　研究成果：

4．1　継続の意味論

継続は、もともと、代表的な制御構造のひとつであるジャンプを意味論的にとらえるために考えられたものですが、近年、さまざまなプログラミング言語で、一般化されたジャンプとして「（第一級の）継続」が用いられるようになりました。「継続」とは、「これから継続して行う残りの計算」のことです。プログラムの実行中に、その時点での継続に適当なラベル L をつけて覚えておき、しばらく計算がすすん

でからＬのあらわす継続を呼び出して使うことにより、ラベルＬへのジャンプを表現することができます。そのほか、コルーティンなど、多様な制御を、継続を用いて表現することができます。

しかし、継続を多用したプログラムの意味を理解することは、大変困難です。goto 文を濫用して不要に複雑になったプログラムのことを、よく「スパゲッティ」などと呼びますが、継続を安直に使うと、さらにたちの悪い「高階のスパゲッティ」（高階、というのは、継続を引数として用いるプログラムは高階の汎関数とみなせるからです）が簡単にできてしまうのです。一方で、継続の現実的な使い方の多くでは、継続は「線形」に、つまりつねに一回しか用いられないことが指摘されています。本研究では、この、線形にもちいられる、いわば「筋の良い」継続のための意味論を整備しました。また、その重要な応用として、亀山幸義氏（つくば大学、当領域 2 期生）とともに、「限定継続」とよばれる精密な継続を扱う機構についてなりたつ等式理論を導きました。これは、限定継続では、メタ継続という常に線形に用いられる継続があることに着目したことが大きな鍵となりました。以下に挙げるのは、この研究によってつきとめられた、限定継続を持つ（値呼び）ラムダ計算の等式理論の公理系です。

$$\langle F[\mathrm{shift}\ M]\rangle = \langle M\ (\lambda x.\langle F[x]\rangle)\rangle \quad (1)$$
$$\mathrm{shift}\ (\lambda k.k\ M) = M\ (k\text{はMに自由に出現しない}) \quad (2)$$
$$\mathrm{shift}\ (\lambda k.\langle M\rangle) = \mathrm{shift}\ (\lambda k.M) \quad (3)$$
$$\langle \mathrm{let}\ x\ \mathrm{be}\ \langle M\rangle\ \mathrm{in}\ N\rangle = \mathrm{let}\ x\ \mathrm{be}\ \langle M\rangle\ \mathrm{in}\ \langle N\rangle \quad (4)$$
$$\langle V\rangle = V\ (V\text{は値}) \quad (5)$$

これらを用いて、限定継続を用いたプログラムについて、たとえば以下のような等式を用いた推論をすることが可能です。

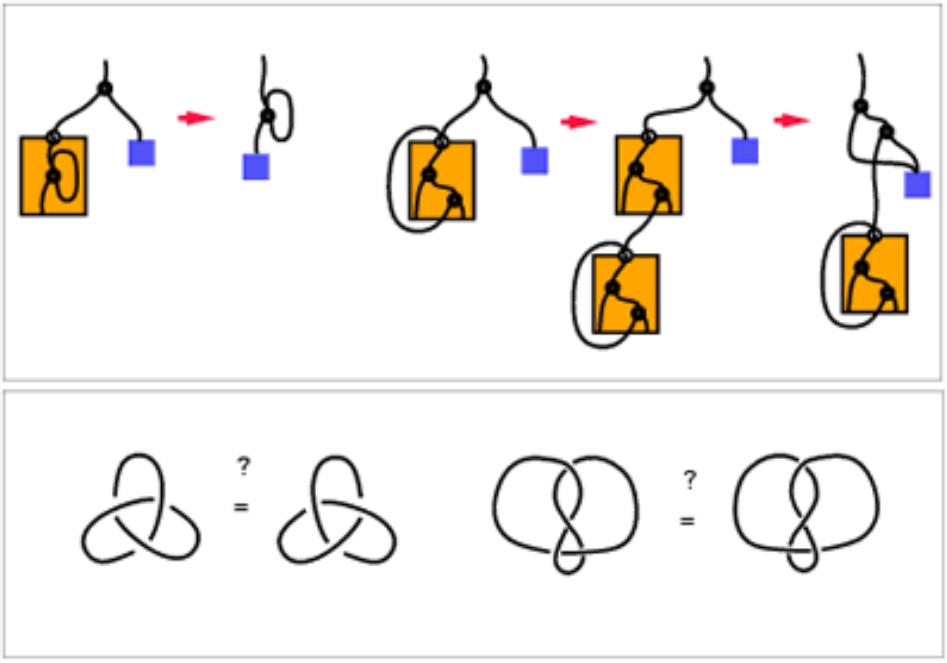
$$\begin{aligned}
& 1+\langle 2*(\mathrm{shift}\ (\lambda k.3+(k\ 4)))\rangle \\
= \ & 1+\langle (\lambda k.3+(k\ 4))\ (\lambda x.\langle 2*x\rangle)\rangle && (1)\text{より} \\
= \ & 1+\langle 3+((\lambda x.\langle 2*x\rangle)\ 4)\rangle && \text{値呼び}\beta\text{簡約} \\
= \ & 1+\langle 3+\langle 2*4\rangle\rangle && \text{値呼び}\beta\text{簡約} \\
= \ & \cdots && (5)\text{などより} \\
= \ & 12
\end{aligned}$$

プログラム中、kは、xから２＊xを求める計算（２＊[-]）につけられたラベルです。この等式推論により、プログラムの実行結果を正しく、また明快に求めることができます。

また、この「継続の線形な利用」は、より一般の制御構造の線形な利用のための、一般化された意味論の枠組みに一般化して議論できることも示しました。

４．２　再帰法の意味論

再帰法は、自己言及的なプログラムを書くことを可能にする、代表的な制御構造です。本研究に先立つ研究で、巡回的なデータ構造から生み出される再帰計算について分析するための圏論的なモデルの理論を展開しており、本研究では、この理論の発展・応用のために欠かせない、「よい性質を持つモデルを構成する技法」について成果をあげました。



この研究の核心となったアイデアは、再帰計算の実装（上図）の違いを適切に捉えるために、絡み目（下図）を分類する不変量のための圏論的構造を応用するというものです。特に、扱いやすいモデルからは、プログラム間の関係が自然に導き出せるので、現在では、そのようなモデルを、すでに知られているモデルをもとに構成する方法を見つけることが大きな問題になっています。

具体的には、この理論で中心的な役割を果たす、トレースつきモノイダルカテゴリとよばれる数学構造に関して、巡回構造を生み出すトレースとよばれる演算が満たす一様性原理の理論を構築しました。これは、古典的な表示的意味論（領域理論）において最小不動点演算子がみたすプロトキンの一様性原理と呼ばれるものの、一般のトレースつきモノイダルカテゴリへの拡張と考えることができます。本研究では、この一様性条件による制約を受けた要素だけからなる「使いやすい」モデルを構成する一般的な技法を与えました。さらに、この方法が自然なものであることを示す例として、古典的な意味論において再帰法について推論するための基本的なテクニックであるスコットの不動点帰納法が、このトレースつきモノイダルカテゴリの構成技法から自然に導かれることを示しました。

4．3　制御構造の相互作用

現実のプログラミングにおいては、複数の制御構造が、巧妙に組み合わさって用いられることがよく見られます。しかし、はじめに述べたように、そのような状況を明快に説明できる理論は、まだ存在しません。この方面では、本研究に先立ち、継続、繰り返しと再帰の組み合わせに関する状況を明快に説明する圏論的な枠組みを与えていました。以下に挙げるのは、その理論的考察から得られた、第一級継続と繰り返しの組み合わせによる再帰のプログラム例です。

```
(* an empty type "bot" with an initial map "abort" A : bot -> 'a *)
datatype bot = VOID of bot;
fun A (VOID v) = A v;
(* the C operator, C : (('a -> bot) -> bot) -> 'a *)
fun C f = SMLofNJ.Cont.callcc (fn k => A (f (fn x => (SMLofNJ.Cont.throw k x) : bot)));

(* basic combinators *)
fun step F x = C (fn k => F k x);          (* step : (('a -> bot) -> 'b -> bot) -> 'b -> 'a *)
fun pets f k x = k (f x) : bot;            (* pets : ('a -> 'b) -> ('b -> bot) -> 'a -> bot *)
fun switch l x = C (fn q => l (x,q));      (* switch : ('a * ('b -> bot) -> bot) -> 'a -> 'b *)
fun switch_inv f (x, k) = k (f x) : bot; (* switch_inv : ('a -> 'b) -> 'a * ('b -> bot) -> bot *)

(* an iterator, loop : ('a -> 'a) -> 'a -> bot *)
fun loop f x = loop f (f x) : bot;

(* recursion from iteration *)
fun fix F = switch (loop (step (switch_inv o F o switch)));
(* fix : (('a -> 'b) -> 'a -> 'b) -> 'a -> 'b *)
```

意味モデルの分析から得られた等式理論を用いて、このプログラム例が、再帰と繰り返しの間の一対一対応を与えていることが（限定継続の例と同様の、等式を使った推論により）証明できます。

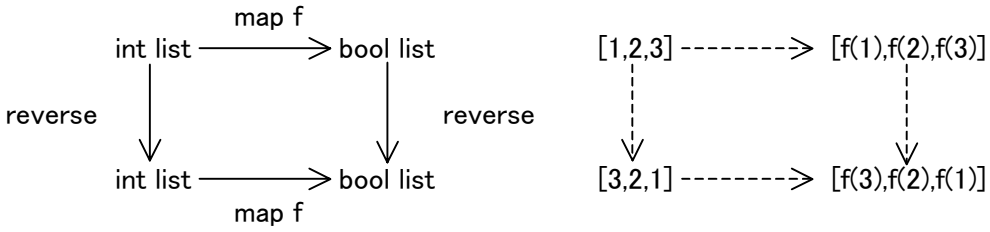
本研究では、角谷氏とともに、この枠組みでの再帰法、継続とパラメータの間に成り立つ関係を分析し、「関数完全性」とよばれる、パラメータの取り扱いに関する基本的な性質が成り立つことを示しました。また、再帰法がこの枠組みでみたすべき一様性原理を調べ、簡単な特徴づけを与えました。その結果、適切な一様性原理のもとでは、上に述べた圏論的な意味モデルはパラメータに関する必要な性質を常に満たしていることがわかり、意味モデルの満たすべき条件を、以前の研究で与えていたものから、大幅に単純化することができました。

４．４　多相型プログラムと制御構造

以上のテーマについて研究を進めていく過程で、プログラミング言語の制御構造は、その中に（たいてい暗黙のうちに）用いられている、ある種の「一様性」に注目して分析を行うのがよい、ということがわかってきました。つまり、制御構造を用いたプログラムが、様々な状況で示す一様な振る舞いに、その制御構造の本質が現れている、と気がついたのです。そのことを端的に示すものとして、本研究の後半で、制御構造を用いる多相型プログラムが満たす、「パラメトリシティ原理」と呼ばれる一様性原理について、集中的に調べました。

多相型プログラムとは、様々なデータ型に対して動くように書かれたプログラムです。ここでは、それらのデータ型の内容に一切関わりなく動くように書かれた（パラメトリックな）多相型プログラムだけを考えることにします。たとえば、リストの順番をひっくり返すプログラムは、整数型のリスト、真偽値型

のリスト、文字列型のリスト、など、どんなデータ型のリストについても使うことができる多相型プログラムです。このようなプログラムは、データ型のとりかたに関係なく一様にふるまうのですが、その一様性を数学的に定式化したのが「パラメトリシティ原理」と呼ばれるものです。リストの順番をひっくり返すプログラム reverse は、パラメトリシティ原理によって以下の可換性を満たすことがわかります。



ここで、f は、整数を入力に取り真偽値を出力する任意のプログラムです。

このような、プログラムに関する代数的な法則が、パラメトリシティ原理から自然に導かれます。これらは、プログラムの性質に関する推論や、プログラムの最適化などのために、基礎的な役割を果たします。

ところで、パラメトリシティ原理は、制御構造を全く含まない、いわゆる純関数型プログラミング言語とそのモデルの理論に関してよく調べられ、その正当性もわかっていますが、再帰や継続のような制御構造が存在する場合には、実はそのままでは正しくないのです。一般に、多相型プログラムは、制御構造を含むプログラムについては、一様に動く保証がないのです。この問題は、再帰の場合には以前から知られていました。たとえば、再帰を用いた（止まらないかもしれない）多相型プログラムは、常に定数を返すようなプログラムに関しては、先ほどの可換性を満たしません。

だからといって、パラメトリシティ原理は、制御構造を持つプログラミング言語について何も言えないか、というと、決してそうではありません。「一様に動く」ことを保証するための、よい条件を見つけることで、制御構造を用いた多相型プログラムのための、一般化されたパラメトリシティ原理を展開することができるはずです。また、その「よい条件」は、用いる制御構造によって定まる本質的な性質のはずです。実際、再帰に限れば、すでに「線型性」という、プログラムの停止性に関する本質的な条件を用いた、「線型パラメトリシティ原理」の理論が、すでに考えられていました。（この線型性は、再帰とトレースつきモノイダルカテゴリの研究で出てきた一様性の定義に用いられるものと、同一のものと思って差し支えありません）。本研究では、線型性に代わり、ジャンプと可換であるという「焦点の合った性質」を用いることで、継続を用いても成り立つ「フォーカル（focal）パラメトリシティ原理」を提案し、その正しさを証明しました。

さらに、その後、「線型パラメトリシティ原理」「フォーカルパラメトリシティ原理」は、より一般的な制御構造に関する理論として、大きなひとつの枠組みのなかで統一的に議論をすることができることがわかってきました。この、多相性と制御構造の一般論に関する理解は、本研究で得られた最も深い成果だと考えています。

5　自己評価:

さきがけ研究における私の仕事は、およそ10年前からとりくんできた、プログラミング言語における

制御構造に関する基礎研究の、自然な継続です。プログラム意味論の最新の成果を利用してプログラムの制御構造に関する、正確かつ有用な原則を抽出する、という当初の目的を十分に達成したのみならず、そのプログラム意味論そのものを大きく前進させる結果を出せたと考えています。長く取り組んできた再帰の意味論についてはその掘り下げを行い、また、比較的最近になって着手した継続の意味論については、亀山氏との共同研究の成功を含め予想以上の成果をあげることができました。そしてなにより、再帰、継続、線形性、一様性といったこれまでに取り組んできた問題を自然に総括するかたちで、プログラムの多相性と制御構造に関する、本質的なアイデアにたどりつくことができました。この、本研究の到達点ともいえる、多相型プログラムと制御構造のパラメトリシティ原理の理論は、今後、この分野の中心的な話題になっていく可能性があると考えています。

6 研究総括の見解:

長谷川氏の研究は，プログラムの振舞いを表す制御構造を数学的に解明することを目的としたものである．現在のプログラム言語は種々の制御構造を持ち，それによって高度な計算を表現できる一方，その組み合わせによって複雑な振舞いが生じ，プログラムの理解を妨げ，誤りの原因になることが指摘されている．長谷川氏は，再帰，継続，多相性をもった高階関数型言語の振舞いを正確に表現するプログラム意味論を，カテゴリ理論などを用いて数学的に展開し，これまでのプログラム意味論を大きく前進させたものであり，この研究から生まれたパラメトリシティ原理は，今後の意味論研究で重要な位置を占めるものと考えられる．国際的な評価も高く，IBM科学賞を受賞するなど，非常に高く評価できる研究である．

7 主な論文等:

（1） 学術雑誌

1. M. Hasegawa, The Uniformity Principle on Traced Monoidal Categories. Publications of the Research Institute for Mathematical Sciences 40(3): 991-1014, September 2004.
2. Y. Kakutani and M. Hasegawa, Parametrizations and Fixed-point Operators on Control Categories. Fundamenta Informaticae 65(1-2): 153-172, March 2005.
3. M. Hasegawa, Classical Linear Logic of Implications. Mathematical Structures in Computer Science 15(2):323-342, April 2005.
4. J.R.B. Cockett, M. Hasegawa and R.A.G. Seely, Coherence of the Double Involution on *-Autonomous Categories. To appear in Theory and Applications of Categories.

（2） 査読つき国際会議

5. Y. Kakutani and M. Hasegawa, Parametrizations and Fixed-point Operators on Control Categories. Proc. 6th International Conference on Typed Lambda Calculi and Applications (TLCA'03), Springer Lecture Notes in Computer Science 2701, pages 180-194, June 2003.

6. Y. Kameyama and M. Hasegawa, A Sound and Complete Axiomatization of Delimited

Continuations. Proc. 8th ACM SIGPLAN International Conference on Functional Programming (ICFP2003), pages 177-188, August 2003.

7. M. Hasegawa, Semantics of Linear-Continuation Passing in Call-by-name. Proc. 7th International Symposium on Functional and Logic Programming (FLOPS2004), Springer Lecture Notes in Computer Science 2998, pages 229-243, April 2004.
8. M. Hasegawa, Relational Parametricity and Control (Extended Abstract). Proc. 20th Annual IEEE Symposium on Logic in Computer Science (LICS2005), pages 72-81, June 2005.

（3） 口頭発表

9. 長谷川真人、Traces in Computer Science、トポロジーとコンピュータ研究集会、2002 年 11 月
10. 長谷川真人、Recursive Programs in the Abstract、第 6 回プログラミング及びプログラミング言語ワークショップ（PPL2004）、2004 年 3 月
11. M. Hasegawa, On a Complete Axiomatization of Delimited Continuations, Joint Queen Mary and Imperial College Theory Seminar, June 2004
12. 長谷川真人、Geometry of Non-deterministic Interaction、第 8 回シンポジウム「代数・言語・計算」、2005 年 2 月

（4） 受賞

2005 年 11 月　第 19 回日本 IBM 科学賞（コンピューターサイエンス分野）

研究課題別評価

1 研究課題名:

Web アプリケーション指向ソフトウェアモデリング

2 研究者氏名:

結縁 祥治

3 研究のねらい:

本研究では，Web アプリケーションの記述と振舞いを結びつけるソフトウェアモデリング手法の確立を目指した．Web アプリケーションは，従来の計算機プログラムとは以下の2点で大きく異なる．

(1) 単一の言語で記述されないこと:Web アプリケーションにはさまざまなプログラム言語やスクリプトが混在する．このため，各々のプログラム言語から統合可能な形で相互作用を取り出せるようにする必要がある．
(2) 制御フローが単一プログラム上で完結しないこと:Web アプリケーションはイベント駆動のメカニズムで実行される．このため，従来の関数的な概念に基づく入出力関係ではモデル化が不十分である．

このような異なったパラダイムのアプリケーション構築の技法に対しては，新たな概念に基づく抽象的ソフトウェアモデリングが必要である．このためのキーとなる概念として，同期イベント通信と振舞い合成に基づいて振舞いモデルを構築する．具体的手法として Milner, Parrow および Walker の π 計算に基づいて，Web アプリケーションの動作モデルを開発する．この計算モデルでは Web アプリケーションの記述の構造から直接的に振舞いをモデル化することが可能になる．

数学的な手法に基づいた形式的意味論によるソフトウェアモデリングを行うことで，既存の言語で記述された Web アプリケーションに対してソフトウェアモデルを構築する．形式的意味に基づくことは，並行的に相互作用を及ぼすコンポーネントの振舞いを定式化する上で重要な意味を持つ．逐次的な実行に比べて，並行的な計算では実行状態を正確に把握することが難しいため，形式的意味に基づいてすべての状態をもれなく把握可能なことは基礎的な振舞いの意味づけとして有用な性質である．

このモデリングに基づいて Web アプリケーションに対してソフトウェアとしての信頼性の向上を図る．さらに，振舞いの明確化による振る舞いの安定性の向上のための具体的技法を提案し，実際の構築環境のプロトタイプを示す．

4 研究成果:

Web オートマトン:Web アプリケーションの振舞いモデル

Web アプリケーションの振舞いを抽象的に意味づけるためのモデルとして，Web オートマトンを提案した．Web オートマトンは，ページ間遷移を状態遷移とみなす静的な Web システムのモデルである Link オートマトンを MVC[1]にモデルに基づく Web アプリケーションに拡張した振舞いモデルである．本

[1] MVC=Model View Control

研究においてはMVCアーキテクチャに基づくフレームワークとしてよく用いられているApache Struts（アパッチ・ストラッツ）フレームワークを抽象的な振舞いモデルとして表現する．Webオートマトンは Webアプリケーションの抽象的な実行の概念を表現する．Webオートマトンの応用としてWebアプリケーションのテスト系列の生成手法について研究した．

Strutsフレームワークの基本動作の概略は以下の通りである．requestチャネルを通して外部からHTTPリクエストを受け取り，ActionFormオブジェクトを作成し，Actionオブジェクトを発行する．Actionオブジェクトはこの際にデータベースなどのバックエンドのビジネスロジックにアクセスし、Actionオブジェクトの結果はActionForwardオブジェクトで返される．その結果を構成ファイル(struts-config.xml)を参照して得られるActionMapping記述に従ってJSPを発行し，ブラウザなどのクライアントに結果を返す．このWebアプリケーションの振舞いをオートマトンの状態遷移としてモデル化した．

Webオートマトンは$\langle View, \delta, \iota, iv, Trans, FV \rangle$という6つ組みで表現される．$View$はJSPによって生成される画面の集合，$\delta$は$View$の各要素を生成するJSPに含まれる変数の集合，$\iota$は$View$の各要素に含まれる変数に成立する条件の集合，$iv \in View$は初期ビュー，$Trans$はビュー間の遷移関係，$FV \subseteq View$は最終状態の集合を表す[2]．Webオートマトンは EFA(Extended Finite Automata)の一種であり，URLリクエストによる変数の更新とページ遷移をモデル化する．Webオートマトンにおける$View$はJSPの変数に代入される値によって無限になる．しかし，遷移のパターンはその構成から本質的には有限であり、状態遷移のパターンによって値を有限的に分類することによって，有限の$View$によって振舞いを有限的に特徴づけることができる．

Strutsフレームワークで構成された図書管理システムのWebオートマトンの遷移グラフを図1に示す．このシステムは，図書データの検索と更新を行う．更新をするためにはあらかじめ登録されたアカウントでログインしなければならない．一回の検索，更新が終了するViewを最終状態として定義する．

Webオートマトンによるテスト系列生成

Webオートマトンの応用として，Webオートマトンの表す受理系列をWebアプリケーションのテスト系列として有効なテストを生成する手法を提案した．Webオートマトンにループが存在して初期Viewから到達可能である場合，Webオートマトンの生成する系列は無限になる．ここで，テストという側面から見ると同じループを回る場合はテストとしてはよく似たテストであるという観測から，ループを回る回数で系列を分類し，Webアプリケーションに対するテスト基準を提案した．通常のテストでは，ループを2回以上回らない系列が用いられる．図1のWebオートマトンの場合のテスト系列の数を表1に示す．ここでループ基準T^nはループをn回実行する系列の集合を意味する．

表1：生成されるテスト系列数

ループ基準	T^0	T^1	T^2
系列の数	9	132	1947

通信プロセスモデルによるモデル化へ

[2] ここで最終状態を定義に含めたのは，このモデルの能力を示すために，テスト系列の生成に応用するためです．通信プロセスモデルとしてモデル化する際には最終状態の区別は本質的に必要ありません．

Web オートマトンのような抽象的な振舞いを Web アプリケーションから直接生成するために，本研究では，Web アプリケーションを表す通信プロセスモデルとして非同期π計算という体系を対象とする．ここでの目的は，Web アプリケーションの振舞いモデルをプログラムから直接生成することである．

MVC のアーキテクチャに基づいてそれぞれの記述ごとに通信プロセス項に翻訳し，そのまま振舞い合成して全体の振舞いを定義する．この変換は以下の手順で行う．

(1) ビューに対する翻訳：各 JSP からアンカータグ，リンクタグおよびフォームタグを構文的に抽出し，アンカー，リンク，フォームの各パーツを振舞い合成する．

(2) モデルに対する翻訳：FormBean をオブジェクトとして変換し，setter メソッドおよび getter メソッドを実装する．

(3) コントロールに対する翻訳：Action オブジェクトを Struts-config.xml にしたがって変換する．Action オブジェクトは，ビューの中のフォームタグからの通信によって起動され，結果によってビューを起動する．

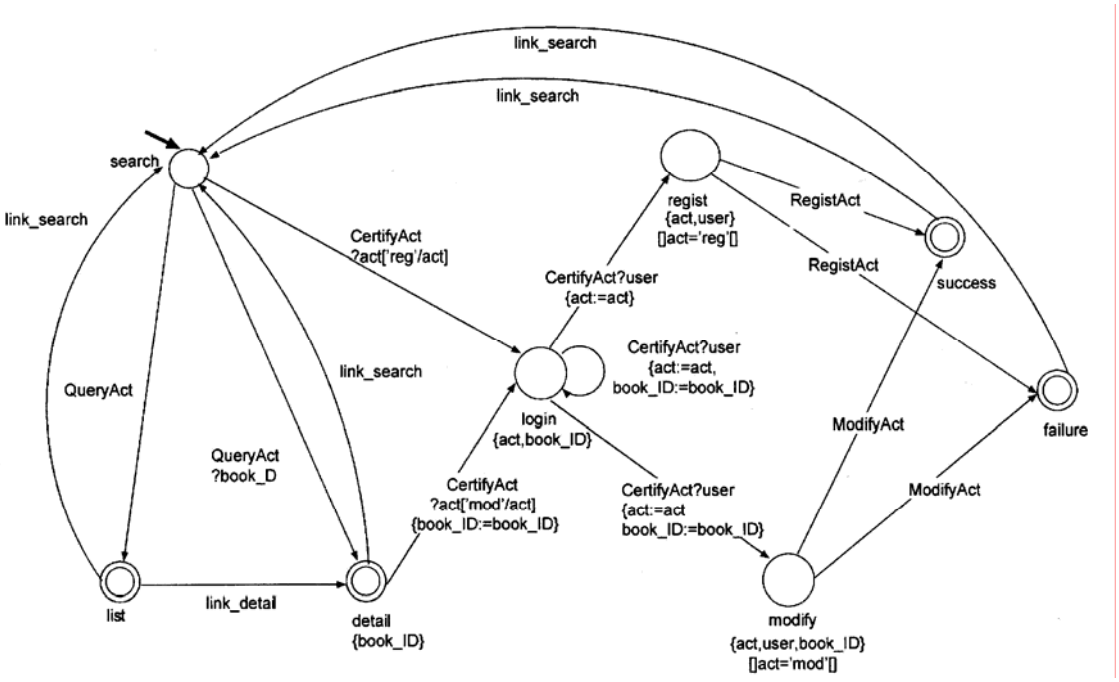


図1 Web オートマトン

それぞれの変換ではコンポーネント単位の Web オートマトンを構成し，非同期π計算の項に変換する．変換ターゲットとして Pierce と Turner によって提案されている非同期π計算言語 Pict によった．Pict 処理系によって抽象的に実行することができる．

ここでは，ログインページにおけるリンクの振舞い，ページ遷移，データベースとの通信をπ計算の同期通信としてモデル化した．さらに，Pict 言語のタイプシステムを用いて，通信内容を表すタイプづけを行った．

5 自己評価：

本研究では、Web アプリケーションの持つ特性を通信プロセスの振舞いとしてモデル化することを目標とした。モデル化を通じて Web アプリケーションプログラムが従来のプログラムとは異なるスタイルで記述され、異なった制御メカニズムで動作することを表す基盤を示すことができた。研究開始当初は、Web アプリケーションの構築においてその振舞いが従来のプログラムのモデルでは理解しづらく、ソースプログラム以外の抽象的な上位概念がなければ全体を構成できないという特徴に注目して、より柔軟な通信プロセスモデルを基礎モデルとすれば明確な特徴化ができると考えた。特徴化のアプローチとして最初は通信プロセスモデルのさまざまな振舞い等価性とその等価性に基づく解析手法を研究した。研究を進める過程で、振舞いを表現するには通信プロセスモデルでは細かすぎるために、まず、抽象度の高いオートマトンモデルを中間的モデルとして定義した。これは、当初の研究予定にはなかった視点であった。このモデルをさらにコーディングする過程で、Web アプリケーションの振舞いをチャネルタイプによって特徴づけを行えば当初の目的をより的確に表現できるという着想に至った。この点について、既存の研究についても詳細なサーベイを行った。Web アプリケーションはプログラムより細かい逐次的実行単位の合成として表現すればよいことが発見でき、その組み合わせにおける制約条件として、タイプ付けが有効であることが発見できた。

全体として、ソフトウェアの解析に対して、Web アプリケーションの進歩が非常に高速であり、基礎的なモデル化がごく一部のフレームワークについてしかできなかった点については課題が残った。本プログラムに基づく代数的な性質およびタイプ情報を利用した解析ツールの実装は引き続き行っていく必要がある。解析の目的として、Web アプリケーションのセキュリティモデルへの応用が有用であると考えている。

6 研究総括の見解:

現在の情報システムは，Web システムとして構築されることが多いが，この様なシステムは制御のフローが単一のプログラム上で完結しないことや，さまざまなプログラム言語で記述されことにより，そのテストや解析が困難であることが知られている．結縁氏は，Web アプリケーションの振舞いを抽象的な Web オートマトンとして形式化し，これをもとに高信頼 Web アプリケーションを開発する研究を行った．Web オートマトンの定義，Web オートマトンからのテスト系列の生成，Web オートマトンから非同期π計算への変換，それらの現実システムへの適用の研究を行い，このような形式的方法論が高信頼 Web アプリケーションの開発に有効なこと示したものであり，高く評価できる研究である．

7 主な論文等:

論文誌

1. Atsushi Mizuno, Ken Mano, Yoshinobu Kawabe, Hiroaki Kuwabara, Shoji Yuen, Kiyoshi Agusa, "Name-passing style GUI Programming in the π-calculus-base language Nepi" Electric Notes of Theoretical Computer Science (Elsevier) , Volume 139 Issue 1, pages 145-168, 2005 (ARTS'04)

2. Mohammad Sharaf Aun, Shoji Yuen, Kiyoshi Agusa,"Towards Assuring Quality Attributes of Client Dynamic Web Applications: Identifying and Addressing the Challenges"

Journal of Web Engineering , Vol.4 , 2005 , pages 144-164

3. Shoji Yuen, Keishi Kato, Kiyoshi Agusa, “Web Automata: A Behabioral Model of Web Applications based on the MVC”, コンピュータソフトウェア , 22 巻 , 2005 , pages 44-57
4. Irek Ulidowski, Shoji Yuen ,"Process languages with discrete relative time based on the Ordered SOS format and rooted eager bisimulation", The Journal of Logic and Algebraic Programming (Elsevier) , Vol.60-61 , 2004 , pages 401-460
5. 桑原 寛明,結縁 祥治,阿草 清滋, “時間付きπ計算によるリアルタイムオブジェクト指向言語の形式的記述”, 情報処理学会論文誌 , 45 巻 , 2004 , pages 1498-1507
6. Mohammad Sharaf Aun, Shoji Yuen, Kiyoshi Agusa, “An Approach for Debugging Client Dynamic Web Applications”, 情報処理学会論文誌 , 45 巻 , 2004 , pages 2373-2383
7. 渥美紀寿,山本晋一郎,結縁 祥治,阿草清滋, “FCDG に基づいたコーディングパターン”, コンピュータソフトウェア , 21 巻 , 2004 , pages 27-36

解説論文

8. 結縁祥治, “通信プロセスモデルと形式意味論に基づくソフトウェアのモデル化”, コンピュータソフトウェア , 22 巻 , 2005 , pages 22-43

会議(査読つき)

9. H. Kuwabara, S. Yuen, and K.Agusa, “Congruence properties for a Timed Extension of the π-calculus”, in Supplemental Volume of DSN2005, 2005, pages 207-214
10. Atsushi Mizuno, Ken Mano, Yoshinobu Kawabe, Hiroaki Kuwabara, Shoji Yuen, Kiyoshi Agusa, "Name-passing style GUI Programming in the π-calculus-base language Nepi", in Preliminary Proceedings of ARTS2004, Technical Report No.2004/28, 2004, pages49-66
11. S. Yuen, K.Kato, D.Kato, S.Yamamoto, K.Agusa: “Testing Framework for Web Applications based on the MVC model with Behavioral Descriptions”, ICITA04, 2004, 11-4:pages 1-6

口頭発表

12. 加藤大樹, 結縁祥治, 阿草清滋, “高信頼性 Web アプリケーションのための振舞い合成手法”, 第2回ディペンダブルソフトウェアワークショップ, DSW05, 2005, pages 41-50
13. 桑原寛明, 結縁祥治, 阿草清滋:”π計算に対する時間拡張と代数的意味論”, ソフトウェア工学の基礎 XI, FOSE2004, 2004, pages 97-108
14. 末次亮, 結縁祥治, 阿草清滋:”時間オートマトンの遷移制約記述に基づく AIBO プログラムスケルトンコードの生成手法”, 組込みソフトウェアシンポジウム 2004, 2004, pages126-133
15. 深谷直彦,結縁祥治,阿草清滋, “実行可能なメモリモデルに基づく Java 並行プログラムのモデル検査”,ソフトウェア工学の基礎ワークショップ FOSE2003 , 2003 , pages 227-238

16. 桑原寛明, 結縁祥治, 阿草清滋, ”時間付きπ計算によるリアルタイムオブジェクト指向言語の形式的記述”, オブジェクト指向最前線〈2003〉情報処理学会 OO2003 シンポジウム, 2003, pages

67-73

特 許

1. 特願 2004-165578 “ウィジェット操作方法，装置，プログラムおよびこのプログラムを記録した記録媒体”，出願日：平成16年6月3日