

私のソフトウェア研究

米澤 明憲

1 はじめに

このたび、個人の学術的な貢献を顕彰するフェロー制度が日本ソフトウェア科学会に創設され、その第一回の受賞者の一人としてお選びいただいたことは、誠に身に余る光栄と存じます。制度の創設・選考に携われた先生方のご尽力に、深い敬意と感謝の意を心から表したいと存じます。

本学会の理事の方からいただいたメールの中には、私が以前に「コンピュータソフトウェア」に書いた2つの論文が挙げられており、好きな方を1つ選ばばそれを近いうちに発行される同誌に再掲して下さるとのことであった。そして、その再掲される論文にあまりページ数にこだわらずに序文をつけてよい、またその序文のタイトルを何とするつもりか至急知らせよ、とあった（もちろん締切はタイトであるとも）。

苦し紛れにここにあるようなタイトルで返答してしまったので、本年57歳で（そのままいけば）あと7、8年はお勤めできる現在の所属大学を「定年」で退職する際の最終講義のタイトルを、あたかも早めに口走ってしまったような気がして、いまだにバツが悪い。しかし、あまり面白いことも書けないので、厚顔無恥なる年長者の昔話と意固地な信条の一端にお付き合いいただければ、誠に有難い次第である。長くなるかもしれないので、その信条の一部については、

あらかじめ、箇条書き的に要約しておく（もちろん、理想を述べているに過ぎず、私がすべてを実践できたものではない）。

1. 奥の深そう (pregnant) な着想を見つけて、長期研究する。
2. 研究の成果に対して明確な目的意識をもって研究する。
3. 理論的研究は、現実問題に有用であることを、その価値基準とする。
4. 力のありそうな人となるべく多く議論をする。
5. 学生は、できるだけ早くかつ多く欧米に行かせ、発表・議論をさせてくる。
6. 理論を研究する学生にも、必ず実装をきちんと経験させる。
7. システム実装が得意な学生には、自分の実装の特徴を概念化・明示化する訓練をさせ、かつ実装の重要性を自覚させる。
8. 各学生には、その人生に1つでよいから、他人が考えたことのない、概念・アイデア・プロダクトを、確立あるいは作成するという使命感をもたせる。

本稿の最後に、これまでの私の学術生活の上で、大変お世話になったり、強い影響を受けたりした先生方や皆様のお名前を順不同で列挙させていただきだけで謝辞に代えさせていただく。本来なら、その由来のようなものを申し添えるべきだが、紙面の制限もありご容赦いただくことにする。

2 なぜソフトウェアの研究へ

2.1 Noam Chomsky

幾分自叙伝じみているが、私の本当の興味は「言語・言葉」にある。言語一般について、今でも強い興

My Software Research.

Akinori Yonezawa, 東京大学情報理工学系研究科コンピュータ科学専攻, Department of Computer Science, Graduate School of Information Science and Technology, University of Tokyo.

コンピュータソフトウェア, Vol.21, No.4(2004), pp.??-??.

2004年8月17日受付.

味をもっている(といて、何ヶ国語も読み書きできるわけではない)。大学の教養学部では、第2外国語としてロシア語を選択した(1966年)。スプートニクが上がり、科学にはロシア語がいる、ドイツ語等はそのうち勉強するだろうという、安易な選択であった(我々の頃は、大学院の入試にも第2外国語があり、それをロシア語で受験せざるを得ず、まったく往生した)。しかし、我々のクラスのロシア語の先生の一人は若くて、学生に刺激を与えようとした。その先生が、あるとき1年生に向かって、ロシア語の文法を「生成文法」で記述するという話があるということ、講義の中でポロっと話した。その講義の直後、私は教師の後を追って彼の部屋に行き、「生成文法」って何ですかと訊ねた。そこで、その先生は、2つほど日本語の論文のコピーを下さし、「興味があるなら、Noam Chomsky の Syntactic Structures という本を読むといいですよ」と恥ずかしそうに言ってくれた。

さっそく丸善に行ってみたら、1冊残っていたので購入した。MITでのChomskyの学位論文である。吸い込まれるように読み進んでいくことができた。おそらく革新的な内容が存在したが、内容それ自体にはあまり感銘を受けなかった(むしろその後大学院で読んだ、言語のタイプ0、タイプ1、タイプ2...等の階層が議論されている、Information and Controlに発表された論文は強い印象を受けた)。強く打たれたのは、彼の明晰な言語表現であった。それは、ある意味で丸山真男の「現代政治の思想と行動」を読んだときの明晰な日本語への驚きと同じであった。論理実証主義の雄Wittgensteinが「およそ語れるものは明晰に語れる。語れないものについては沈黙を守らねばならない」と言い放った気分が小気味よく思い出された。

2.2 アセンブラの解読

教養学部の学生には、一般の教養学部の講義よりレベルの高い特別な講義群がボランティアの先生方によって用意されていて、私は1.5年の間に折原明先生の「Banach空間論」、齋藤正彦先生の「超関数論」と森口繁一先生の「電子計算機入門」をとることができた。前者の2つも大変面白く勉強になったが、この種のことを研究する職業はちょっと苦しいなと思っ

た。一方森口先生の講義は、十進の磁気コアメモリが64K桁ほど用意されていたOkitacという計算機の機械語とアセンブラを教えてくれるというものであった。

本郷へ進学する学科を選択する時期となり、チョイスが非常に沢山あり迷った。脳のことや人間の考えるメカニズムについては、難しくてなかなか解明できないであろうから研究するのに相当な根気がいると思ったが、自分自身の興味をそらすことは出来ず、そうしたことを比較的深く研究されていた本郷の南雲仁一先生を友人と二人でお訪ねした。「そういうことをここで勉強したいなら、エレクトロニクスと数学をよく勉強しておきなさい」という助言をいただいたが、数学はまだしも、私はエレクトロニクスは苦手であったので、この系統はあっさり諦めた。

結局、計数工学数理コースに進学し、夏学期が終わるか終わらないかで全学ストライキに突入した(1968年)。夏学期の甘利俊一先生、伊理正夫先生、森口繁一先生の講義は、私がいよいよと思う講義のモデルとなった。学生ストライキによる混乱と混沌の中(本郷の総合図書館が学生によって閉鎖されるまで、そこでChomskyの著作を何冊か読んだが、Syntactic Structuresを超えるものはなかった)での3、4年生であったが、その森口先生の講義の演習で使われたAlgolのサブセットAlgolipが吐き出すアセンブリ言語がどのようにOkitacの機械語になるのかに興味を持ち、Okitacのオペレータの女性に頼んでアセンブリプログラムの部分を16進でdumpしてもらい、それを一生懸命に解読した。名前表や、新しい名前が出てきたときの処理、同じ名前の出現をポインタでリンクしておくなどの処理を知って、感心した。次にAlgolipのプログラムをアセンブリに変換する部分に挑戦しかかったが、16進でまた解読する気力もなく、逆アセンブラを自分で作ってから解読するという知恵もなかった。しかし、このアセンブラの解読は、言語処理系に強い興味を持つきっかけとなった。

また、伊理先生が、演習のような講義で、「Fortranでスタックをシミュレートしながら、ハノイの塔の解を再帰プログラムで書いて実行せよ」という問題を出され、それを解いたとき、気分が「すっと」したのを

今でも記憶している。

2.3 ALGOL N のコンパイラ作成

大学紛争も終わり、大学院の入試のロシア語はなんとかごまかし、森口繁一先生・和田英一先生の研究室に所属させていただいた。武市正人さんと同期・同門である。このころ、プログラミング言語研究の世界的な流れの一つに、拡張可能言語 (Extensible Languages) の設計・意味論・実装などがあり、Algol 系言語の拡張形式について数カ国が提案をし、その中の一つを IFIP による世界規格にするという競争が始まっていた。日本からは、故島内剛一先生を中心に、故米田信夫先生・和田英一先生が ALGOL N という言語の設計を行っていた。同様の目的で、当時日立中央研究所の中田育男先生を中心とし、別な言語名と方式での設計・実装も始まっていた。当時修士 2 年の鈴木則久さん・藤崎哲之助さんが中心となり ALGOL N のコンパイラの実装が始められ、修士 1 年の武市さんと私がその実装の手伝いをするようになった。これは、お手伝いであったが、コンパイラをどうやって作るかを知る上で、大いに勉強になった。ガーベジコレクションのことを学ぶ機会ともなった。ちなみに、拡張可能言語とは、一つの言語内でプログラマがいろいろなオペレータを定義することができ、そのオペレータ群を使えば Fortran でも Cobol でも PL/I でも、同じコンパイラでコードが生成できるようにすることを目指す言語であった。コンパイラ作成に使った言語は Fortran とアセンブリ語であった。

2.4 定理証明

先輩の修士論文が終わり、次は自分の番であった。コンパイラをもう一度作っても仕方ないので、修士 1 年のときから少しずつ勉強していた定理の自動証明のことを考えてみることにした。J. A. Robinson の導出原理やその変形・変種などたくさん論文を読んだが、ぴんと来るものはなかった。島内先生にご相談したら、G. Gentzen の 4 つの論理系と、そのなかの特に LK に基づく証明手続きなどについて、わざわざ 2, 3 回講義をして下さった。「後はなにか、これを使っておもしろいことを考えたら」と言われて一瞬

途方に暮れたが、ひと夏 on-and-off で考えていたら、修士論文の種が幸い見つかった。

一階述語論理の恒真性の判定は、一般には決定不能 (あるいは半決定可能) であるが、証明すべき論理式の形を制限すると決定可能になる場合がある。Ackermann 関数で有名な W. Ackermann の著による比較的厚い「Solvable Cases of the Decision Problem」という本のタイトルが目に入り、決定可能なケース (decidable cases) を調べてみる気になった。何週間かして、いくつかの決定可能なケースが証明できた。そのうちの 2 つは、幸か不幸か、1930 年代にすでに Jacques Herbrand が発見していたことが分かり落胆した。しかし、証明法も違うし、Herbrand が証明した定理の再発見なら、修士論文ぐらいには構わないだろうという評価にしてもらった。理論の追求はここまでにして、M2 の 9 月に入ってから、東京女子大にあった Tosbac3400 をお借りして実装を開始した。発見した決定可能なケースを前もってチェックする機構を内蔵した、シーケント形式で一階述語論理式を書けばその恒真性を (準) 自動的に判定する定理自動証明器を一人で実装した。まともな Lisp の言語処理系はまだ日本に無かったので、再びアセンブリ語でプログラミングした。約 6000 語ぐらいのプログラムになったように記憶している。I/O は Fortran で書きリンクした。この研究は、私の最初の論文となって「情報処理」に掲載された。

この実装に関連して、木 (tree) の中のポインタを反転させることで、スタックを用いずに木をトラバースする方法を考えついた。島内先生が「リカーションとスタックとは同相なんだ」とつぶやかれたのを小耳に挟み、何か旨い方法がありそうだと思っていたので、この方法を思いついたわけである。われながらこれは良いアイデアと思って、助手の人にレポートの形でお見せしたら、「これなら出来るね」とは言って下さったが、論文にしたらとまでは勧められなかった。これは、私がもう少し積極的であればよかったと悔やまれる経験であった。なぜなら、これと全く同一のポインタ反転アルゴリズムが、Schorr-Waite-Wegbreit のアルゴリズムとして米国の著名な教科書に掲載されていたからである。このようなことは、昔の日本人

の研究にはたくさんあったのではないだろうか。

3 外国研究者の訪日

3.1 John McCarthy

私が修士1年のときに、StanfordのJohn McCarthyが大学を訪問し、彼のAI研究所の当時最新の成果をフィルムで見せていった。

Instant Insanityという名称のカメラ系に連動したマニピュレータシステムシステムのデモフィルムを講演中に見せてくれた。それは、6色塗りの立方体6個を縦に積み上げ、かつ全ての側面を同色にするという知能(?)を持ったロボットシステムのようなもので、McCarthyは、これをDARPAの連中に見せて研究資金を貰うんだと言っていた。当時、同じ講演で、Lisp 1.5のmeta-circularな操作的意味論の話もしたように記憶している。私は、meta-circular definitionを動的に変化させたらプログラムが進化しそうだと考えた。これが、後年自分が自己反映計算の研究に入っていたきっかけとなったことは間違いない。このとき彼を羽田まで送りに行った際に、自分の考えている定理証明器の話をしたら、自分の考えている定理証明器の話をしたら、「何に使うのか?目的がはっきりしていないならあまり意味がないよ」「導出原理に基づくものか?もしそうなら、長い証明/探索には、そのままでは旨くないから、巧みな証明制御が必要だ」とMcCarthyは言ってくれた。導出原理に基づく定理証明の場合、証明したい述語の記述されている形を全てconjunctive normal formに書き換えてから「導出」を始めるので、元の述語の記述構造が失われてしまうから、証明に問題領域の性質や構造が利用しにくくなるというのが、彼の見解であると私はその時推測した。

3.2 Rod Burstall

これも修士1年のころと記憶しているが、JIPDECが招いた多くのAI関係の欧米人のなかに、一人だけかなり若いEdinburgh大学のRod Burstallがいた。彼は、Donald Michieの研究所でやっている巨大で不恰好なロボットハンドのスライドをみせながら、この制御はPOP2という言葉で記述され、その言語は

パターンマッチングやある種のバックトラック機能を持っていると言っていたように記憶している。この言語は、ある意味で、Carl HewittのPlanner言語の前身に近いものであった。さらに、Burstall自身の研究として、様相論理を用いてロボットの制御を記述する構想を述べていった。私が講演直後に様相論理のことをいろいろ訊いたためか、「あなたはCresswellの“Modal Logic”という教科書を読むべきだ。私もその本で勉強した。」という内容の手紙が2週間後に英国から届いた。このときも早速丸善に行ったが、この本は見つからず、注文して2ヶ月後に届いた。本にはS1だのS5だのの様相論理のシステムが意味論と共に書いてあり、Saul Kripkeが15歳で作ったという意味構造(semantic structures)には感銘を受けた(BostonでKripkeの講演を聴くチャンスに恵まれたが、そのときは、彼の昔の天才ぶりはあまり余り感じられなかった)。

3.3 Andrei Ershov

これは私が修士2年生のときだと思うが、後藤英一先生が、ロシアの理論家Andrei Ershovを招待し、東大で彼の講演が行われた。いわゆるプログラムスキーマの話で、「フローチャート風の言語で書かれたプログラムの等価性をどのように論じるか」というもので、それ自身は決して悪くない講演であった。しかし、「再帰呼び出しのあるようなプログラムの等価性は、この方式でどうするのか」と手を上げて質問すると、Ershovはかなり長い返答をしたが、正面からの回答は出てこなかった。そのあと司会者が更なる質問を促して後藤英一先生を指名されたが、後藤先生は「いまの学生さんに先に質問をされてしまった」と頭をかきながら言われたので、なんだか嬉しくなったのをやはり記憶している。

3.4 Patrick Winston

やはり修士2年のときに、当時の電子技術総合研究所(今の産業技術総合研究所)がMIT人工知能研究所のPatrick Winstonを招き、MITでの最新の研究について、比較的詳細に日本で知ることのできる機会を作ってくれた。3日間、朝から夕方までWinstonは

ずっと講義をした。そこでは、彼の概念学習プログラムの詳細、Terry Winograd の自然言語理解システム SHRDLU のかなりの詳細、それに、このシステムを大成功に導いた、Hewitt が考案し Gerald Sussman らが Lisp で実装した micro-Planner という言語が手際よく紹介された。Winograd のシステムは当時まさに画期的であり、AI (人工知能) の更なる大きな可能性を世に訴えるうってつけの強力さとアピール度をもっていた。一方、micro-Planner は、プログラミング言語として大変興味深いもので、バックトラック機能と、表明データの宣言、削除、パターンマッチング、パターン主導型手続き起動などの新しい機能を備えた、これまた画期的な言語であった。そして、述語論理式の「手続き的な解釈」を与える言語としても認識されていた。micro-Planner については、東大の研究室での和田英一先生のゼミで、この言語のマニュアルを読み尽くした。

4 MIT の AI Lab と LCS で

4.1 留学スカラーシップ

人工知能と micro-Planner への強い興味で、MIT に長期間行ってみたいと思い、森口・和田両先生にご相談したところ、是非行ってみるといいと言われた。その前に、和田先生からお借りした、Multics のマニュアルや Jerry Saltzer の学位論文を読んでいたのも大きな理由かもしれない。日本学術振興会は当時、米国に9ヶ月留学するためのスカラーシップを毎年募集していたので、これに応募した。受験に際し、以前そのような面接試験官をしたことがあるという方をご存知の友人のお母さまに、一般にどんなことが質問されるか伺ってもらった。また、TOEFL と GRE を受けて書類を整え、同時に米国の大学院博士課程6つに出願書類を送った。数人の先生方に推薦状をお願いしたが、どの先生も私の実力以上のまさに身に余る良い推薦状を出して下さったようである。5つの課程からアクセプトがきたので、MIT に行くことにした。アクセプトがきたのが4月のはじめで、7月には学術振興会がフルブライト委員会に委託している6週間の語学研修のために Texas 州 Austin に行かされた。それを終えると、New Orleans を回ってそのまま東

海岸の Cambridge に向かった。

4.2 セミナーと講義

1973年の9月に始まった大学院の講義は、秋学期、春学期ごとに週2,3回の講義とゼミがあり、新入生には学生個別に、どの講義を取るべきかの相談と助言をするカウンセラがついた。Winston が私のカウンセラで、彼はどの講義をとってもいいと言ったが、今度新設された Albert Meyer の“Algorithms”という講義は、是非取るようにと強く勧めた。実際、Meyer の講義は、その前夜はいつも徹夜だったと彼が後に述懐していたように、彼にとっても初めての、大変“力”の入ったものであった。本質だけを非常に分かりやすく説明してくれ、かつ毎回宿題が出た。それを TA が採点して翌週には返してくれた。

セミナーは、多くの場合若手の先生が自分のリサーチトピックや得意分野を中心に、学生の人数をせいぜい7,8名ぐらいに限って出席させて、講義や学生の発表と徹底した議論を行うもので、私は、はじめの年の9月から Carl Hewitt のセミナーに毎回出席することとなった。Brian Smith, Henry Baker, Ken Kahn などの学生と Hewitt が喧々諤々、私が議論に口を挟む余地などなかったが、時々私が何かを言いかけて口をパクパクしていると、Hewitt が「Aki が何か言いそうだ」と言って皆の発言を制止してくれた。議論はもちろん Actor の揺籃期の混沌たるものであり、Hewitt が新しい操作的意味論を提案するたびに、セミナー参加者のブーイングをうけた。しかし、彼の考えはいつも斬新で直感と洞察に満ちたものであり、学生の教師評価では most brilliant professor という評価を受けていた(もちろん、1973年の時点で micro-Planner や Planner の研究は完全に終わっていた)。

Barbara Liskov の CLU 言語設計のセミナーにも参加していた。このセミナーはより整然としたものであったが、Liskov の冷静な判断と説得力で、CLU を着実に設計していき、実装も学生を雇いながら上手にやっていくのをつぶさに見た。Alan Snyder と Liskov とによる、抽象データ型の定義モジュールや、基本手続きのパラメタに型の制限をいれる where-clause の

宣言導入に関する議論は、幾度となく激しいものがあった。

Lisp 1.5 のマニュアルの著者の一人である Michael Levin による「Logic for Computer Science」という講義では、講義の時間が余ったので、私が東京で研究した「述語論理の証明手続き」について、3 回連続で 20 人ぐらいの大学院生の前で発表させてくれた。Gentzen による LK に基づくシーケントベースの形式なので、米国ではあまり知られていないようで、興味をもって皆が聞いてくれたのは嬉しかった。

4.3 RA をもらう

私の日本からのスカラシップは 1 年間だけであった。その先ここに留まるためには、来年の RA (Research Assistantship) をなんとか手に入れなければならなかった。セミナーでの学生の点数は、term paper (日本のレポートよりも少しプロジェクト性が高いもの) とセミナーでの発言等で決められる。私の場合、発言はあまり勘定にいれてもらえなそうであったので、Hewitt のセミナーの term paper に焦点をあてて、彼を impress するしか方法はなかった。そのとき、どんな term paper を書いたか記憶にないが、かなり大部のものだったことを記憶している。クリスマスごろ、来年度 RA が欲しいと Hewitt に頼むと、1 月の終わりに返事をすると言われた。MIT の EE&CS 学科では Ph.D. candidate になるための資格試験は 3 回合格する必要があった。1974 年の 1 月の初めに第 1 回目 (PWE: Preliminary Written Examination) を受け、2, 3 科目においてクラスで 1, 2 番の成績であったらしい。それとどう関係があるかわからないが、1 月の終わりには、Hewitt から、来年度から君に RA をやるというメールを受け取った。その直後、いつまでもらえるのかと訊くと、「Ph.D. が取れるまで」とウインクされた。ARPA からの研究資金を AI Lab. と LCS (Laboratory for Computer Science) がちょうど獲得した時期で、私は両研究所から 50/50 の共同サポートを受けることになった。このため、所属もこの両研究所ということになった。

4.4 学位のための研究

2 年目からは RA になったので、なにか特別なことをやらされるのかと思って、Hewitt に何をすべきかと尋ねると、「Actor 形式の中でなら好きなことを研究してればいいよ」という返答が戻ってきた。そこで、前年度 Hewitt が出していた Actor 形式の操作的意味論、すなわち meta-circular interpreter を 2, 3 週間眺めていて、これを使って Actor 形式 (まだ並列性を導入していなかったもの) で書かれたプログラムをデータなしで抽象的に評価する方式を考えついた。まずはキューに相当する Actor プログラムを、具体的なデータなしで評価するというところを行った。これを Hewitt は meta-evaluation と呼びたが、実は後のいわゆる symbolic evaluation と呼ばれるものと、同一のものであった。この話は AI Memo として公表することができ、これを読んでいた古いフランスの研究者からは、symbolic evaluation のパイオニア的仕事と言われることが何度もあった。もう少し別の見方をすると、状態を持つ (stateful あるいは side-effect を持つ) 抽象データ型 (ADT) の実装が仕様を満たすか否かのチェックをしていることに対応していた。状態を保持しうる抽象データ型に並列性を導入すると、後に私が「並列オブジェクト指向」と呼んだ計算モデルに自然と発展することになる。

Hewitt はこの研究を比較的気に入ってくれたらしく、この方向で学位論文を書くことを勧めだしてくれた。まだ Actor 形式自身もしっかりできていない上に、並列性を扱うモデルも曖昧であったので、道は遠いとは思ったが、この方向で進めていけばなんとかなりそうだという気もしてきた。この間の Hewitt との議論は e-mail を使って行うことが多かった。彼は何しろ夜中の 10 時ごろオフィスに出てきて朝帰宅することが多かったので、彼に会って話をするタイミングを見つけることになり苦労した。

学位のための研究は、並列性を明示的に入れた Actor 計算モデルをまず明確に定義することであった。これを表示の意味論として書き下す研究も考えたが、良い表示の意味領域を考案するような数学的な力もないので試みなかった。学位が終了したあと D. Scott からメールが来て、意味領域を考えたいから君の学位

論文のコピーを送ってこないかと言われたので喜んで送ったが、その後彼自身で Actor の意味論のための領域を考察した様子はなかった。私の後、MIT の Ph.D. の学生 Will Clinger がこの分野で論文を書いたが、数学的な欠陥があったようである。

そこで、むしろ私は、かなり精密に定義された仕様言語を考案し、並行性をもつ Actor プログラムがその言語で書かれた仕様を満たすか否かをチェックする、symbolic evaluation に基づくかなり形式的方法を考案した。この仕事が学位論文の主要部分になると思ったが、指導教授である Hewitt は、形式的な部分はいいが何か分かりやすい appealing な例がないとだめだと言った。彼を満足させる例を考え付くのに3ヶ月ぐらいかかったように記憶している。

4.5 よい例題 (モバイルオブジェクト・モバイルエージェント)

そのころ、ノルウェーのオスロ計算センターにいた Kristen Nygaard (2001 年 Turing 賞受賞) が、最初のオブジェクト指向言語である Simula67 の後継言語を設計していて、我々と多少の交流があった。そこで、Hewitt の要求する良い例として、Actor 形式で何らかの社会システムあるいは離散事象のシミュレーションを書き下し、それについての正しさを形式的に検証するのはどうかと考えた。離散事象の例といっても、ガソリンスタンドへの車の出入りのようなものは退屈なので、もっと良いものはないかと考えめぐっていた。ある日、アエログラムを買うのに Cambridge 市の中央郵便局に行く機会があった。これを機に、少し大きめの郵便局での訪問者・局員・カウンター・ドア・局内ポスト・切手・お金などを、actor で表現することを思いついた。もちろん大部分の個々の actor は並列に挙動するわけで、並列性は十分あり、かつ分かりやすい例となった。図示すると図 1 のようなものだが、これを見て指導教授は OK を出してくれた。学位論文全体を書き上げるには、その後夏をはさんで半年近くかかった。学位の審査委員は3人で、指導教授の Hewitt と、Thesis Reader である B. Liskov と J.C.R. Licklider の両教授であった。Liskov は細かく提出論文を読んでくれて、コメントも沢山くれ、かな

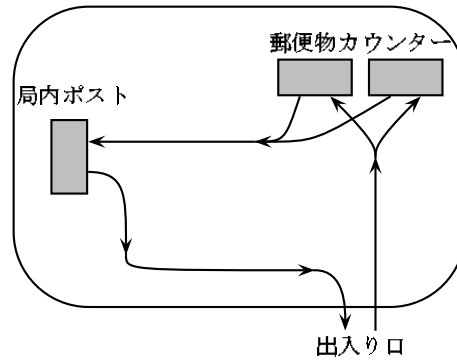


図 1 Cambridge Post Office

り英語も直してくれた。彼女にも大変感謝している。

この郵便局モデルは、後に「モバイルオブジェクト」や「モバイルエージェント」と呼ばれるものを考える上で、大変役に立った。私としては、これは、この種の話の先駆的な仕事と思っている。局内を移動する訪問者も actor であり、それらは物理的に移動しながらメッセージを授受できるようにモデル化した。また、数年前、Boston であった理論オブジェクト指向ワークショップの招待講演でこの話をしたとき、Ambient Calculus を考案した Luca Cardelli は、Ambient と例えばカウンター actor の類似性を指摘して、Ambient のような階層・入れ子構造をカウンターは持てるのかと質問してきた。答えはもちろん yes である。上の郵便局モデルで、郵便局に出入りする訪問者をモデル化する actor は多数郵便局に存在し、かつ待ち行列をつくるなどしてカウンターからのサービスをうける。また、訪問者の状態 (state) によって挙動が変わってくる。1990 年代の後半に我々の研究室で、関口龍郎氏が JavaGo を、橋本政朋氏が MobileML を設計・実装したのは、このときの郵便局モデルに対する私の思い入れを投影したものである。

5 日本に戻って

5.1 「算法表現論」

米国留学の最初 1 年間は、日本学術振興会からお金をもらっていたので、ビザのタイプで、2 年間は日本に返ることが決められていた。和田先生のお世話で東京工業大学の木村泉先生の研究室の助手として就職さ

せていただいた(1978年1月)。日本の大学の計算機環境は、カードによるバッチ計算方式で、TSSは実に300bit/secのモデムでつなげて使用するものであった。メールなど及びもつかない状況であった。

ここで助手を5年近くしていたが、その間英文の論文を5つほどしか書かなかった。たとえば、関数型言語のちょっとした話や、簡単なデータベースへの問い合わせプログラムの自動合成法の話をつづめた論文のほか、学位論文の内容の一部を「A Formal Specification Technique for Abstract Data Types with Parallelism」という標題でまとめた論文などである。助手であったので研究室の学生さんと一緒に仕事ができず、大変苦しい時代であった。しかし、岩波講座情報科学の12巻「算法表現論」の第2章から終章の第12章までを書かせていただいたので、精神的には平衡が保てた。また、体系的に新しいものを比較的易しく書くという点で、自分自身の勉強にも大いになった。この間、Oxfordに移ったC.A.R. Hoareに手紙を書いて、ポストクのポジションはないかと訊くと、あるから来てよといわれたが、手続きの最後の段階でうまくいかなかった。

述懐すれば、この時点で日本に留まって良かったような気がする。この時代に京大数理解析研究所の中島玲二さんや、当時電総研にいた古川康一さんとソフトウェアの学会を日本でつくる計画を練ったりもしていた。その後、多くの方々のご賛同とご努力の結果、「日本ソフトウェア科学会」が設立された。

5.2 自分の研究室を持って

東工大で助手から助教授にしてもらい、自分の研究室を作ってもよいということになった。35歳のときである。始めにしたことは、研究室助手になってくれる人を探すことであった。大変幸運なことに、京大数理解析研究所の中島さんのところで修士を取った柴山悦哉ということができる学生がいるというので、京都まで会いに行き、その場で彼に来てもらうことにした。数理解析研究所の難しい数学の入試をこなしているし、Prolog-KABAの主たる実装者の一人で、これに関し本も出版する予定であった。最初から大変よい片腕に恵まれたのは、実に幸運であった。

大学の教師になると講義という当然の義務が生じ、おそらくどなたでも同じであろうが、はじめの1年は、担当の講義の準備に忙殺されて、研究などは何も出来ないのが実情である。そんな中、学部4年生の大沢一郎君1人だけが私の研究室に志願してくれ、学生として配属されてきた。私は助手のときから彼の演習のレポートなどをよく見ていたが、どのレポートもきれいな字で明晰に記述され、内容的にも100点以外につけようのないものであった。自分は講義の準備のためなどで結構忙しいので、彼にTerry Winogradの学位論文と自然言語処理の適当な文献を渡し、時々議論をする程度であった。確か、私から彼には、自然言語で新幹線の予約を対話的にできるシステムを出るところまで作ってみたら、とまで言っただけであったと思う。彼は、卒論終了の段階でかなりよく動くシステムを構築し、さらに修士課程に入り、当時として画期的な自然言語対話システムを構築してくれた。このシステムのお陰で、NHK(7時のニュースを含む)に何度か出してもらったり、日経産業新聞や科学朝日などに記事を書いてもらった。

5.3 並列オブジェクトの研究(ABCL/1の研究)をはじめ

1983年に入って、日本に戻って初めて自分の研究グループが作れるようになり、本格的に研究のできる環境が整ったとき、なにを本格的に研究するか熟考した。私が学位論文を終えるころ、HewittはActorの理論形式からなんとか状態概念を消し去ろうとしていた。もっと関数的(functional)なものにしたがって、いろいろ工夫をしていた。しかし、私はむしろ、様々な並列・並行現象のモデル化には状態がある方がより適切だと反論していた。そこで、学位論文の延長にあり、かつHewittの進みつつある路線とは異なる自分の信条に沿った、状態を持つことを基本とする並列・並行モジュール群についてじっくり研究していこうと決意した。このモジュールは後に「並列オブジェクト」と呼ばれるものである。

5.4 言語研究の三位一体あるいは正四面体と再掲

論文

この並列オブジェクトが有効であることを示すためには、(1)「並列オブジェクト」群に基づいた計算・情報処理のモデルの確立、(2) そのモデルに出来るだけ忠実なプログラムを記述するプログラミング言語の設計、(3) その言語の厳密な意味論の確立、(4) その言語の効率のよい実装（コンパイラ等）の実現、(5) その言語の実用、を行わなければならないと私は思っていた。また、言語を使用しやすくするための諸ツールの工夫も必要である。これら5つのことを、これから一つ一つ実現していこうと思ったわけである。また、これらのアスペクトを考えていく中で新しい着想に遭遇し、さらなる様々な発展が期待できると、楽観していた。

実際、東工大の渡部卓雄氏と一緒に考えた並列オブジェクト群における自己反映計算 (reflection) などは、その顕著な例である。MOP (MetaObject Protocol) の後を追って現在注目を受けている AOP (Aspect Oriented Programming) というモジュール分割は、並列オブジェクト群の場合どのようになるか興味をもっている。また、並列オブジェクトにどのような継承機構をいれるかという議論は、Jean-Pierre Briot 氏との議論からはじまり、松岡聡氏がこの問題を「Inheritance Anomaly」と命名した。彼が主として多様な議論を展開し、国際的にも大きな注目を浴びた。松岡氏とこれに関して論文を書いたときは、この込み入った問題をこれ以上追求してもあまり実りの多い結果は出ない、という警鐘を鳴らすことを意図したつもりであった。しかし、論文が刊行されてから、「Inheritance Anomaly」を標題にした論文を数多く見るようになり、またこの問題の研究のために研究室に来たいという外国人学生が続出したのには閉口した。

2つ上のパラグラフで言及したプログラム言語研究へのアプローチの(2)、(3)、(4)は、言語研究の「三位一体 (trinity)」と勝手に命名しており、その正三角形に(5)という頂点ノードを加えて立体化したものを、これも勝手に言語研究の「正四面体」と呼んでいる。なんとなくこれらの感じを表現する単純な図をつけておいた(図2)。

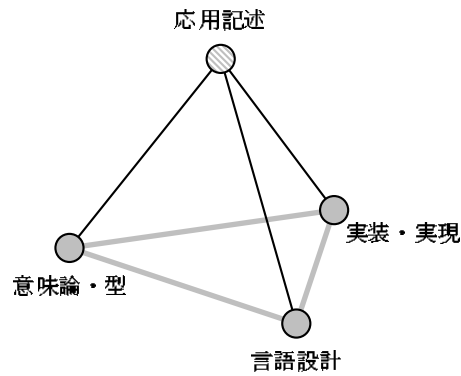


図2 言語研究の三位一体と正四面体

さて、日本ソフトウェア科学会のフェロー受賞のご褒美として、昔の論文を再掲してよいということで、選ばせてもらった論文「オブジェクト指向に基づく並列情報処理モデル ABCM/1 とその記述言語 ABCL/1」は、上で述べた(1)の確立を目指した論文である。東工大での研究室の第2期生にあたる、この年次の最も優秀な学生さんである渡部卓雄氏、本田康晃氏、高田敏弘氏が、私の研究室に配属されてきて、さらに柴山助手とフランスからの留学生 J.-P. Briot 氏が加わった大変強力なグループで日夜議論を重ねながら、私が纏め上げたモデルと言語を論文としたものである。これは1986年の4月に書いたものであるが、ほとんど同時に内容を短くして ACM の第1回 OOPSLA (Object-Oriented Programming, Systems, Languages, and Applications) に投稿し、うまくアクセプトされた。これで国際舞台に復帰した気分がしたのを覚えている。もっとも、前年にはパリとベルリンで招待講演を頼まれ、ABCL/1の部分的な話をする機会を得ていたが、米国での最初の第一線の舞台となると、その感慨はひとしおであった。

5.5 OOPSLA86 (Portland, Oregon)

再掲の論文にあるように、(1)の基本的な計算モデルが確立された。三位一体のアプローチとしては、(3)意味論と(4)言語処理系の実装・実現をなんとかやっつけてしまわなければならなかった。意味論に関しては、Hewittの学生で自分の後輩にあたる Gul Agha が、彼の学位論文として、大域的名前の無限生成を許

す遷移システムを基礎として、ACTOR というタイトルの成書を公刊する予定になっていた。少し焦りを感じたが、彼の意味論は状態を持たないモジュールを基礎にしているので競争しないことにし、人間的にも「うま」があったので、むしろ相互訪問や論文の相互参照を繰り返して協調していくことになった。これに加えて、比較的古い年代層には名前が知られていたブラウン大学の Peter Wegner が、なんだか私を気に入ってくれたので、3人で「Concurrent Objects」の国際的なスクラムを組んで進んでいくことになった。また、日本では偶然、当時慶応大学の所真理雄氏がやはり、ある種の並行オブジェクトをアクティブに研究されており、ECOOP (European Conf. on OOP) や OOPSLA のプログラム委員会でも共闘ができ、オブジェクト指向における並行性の問題は、国際的に常に大きなトピックを占めることができた。

設計した言語 ABCL/1 の実装は、始めた当時、東工大の一杉裕志氏と柴山悦哉氏が Lisp を用いて擬似並列で実行される版を製作してくれて、柴山氏が英文でうまくマニュアルを書いてくれた。このマニュアルを 30 冊ほど持って、Oregon 州 Portland で開催される第 1 回の ACM OOPSLA に向かった。私の発表は初日の午後で、Ken Kahn の講演の次にあたり、セッションチェアは Daniel Bobrow であった。発表は、自分では大変緊張したが、今までの自分の英語の講演では一番よいものであった。内容がはっきりしていたので、しゃべりやすかった。発表後、真っ先に Kristen Nygaard (この日の Keynote Speaker であった) が大仰な拍手をしながら質問者用のマイクに来て、Great work! Great work! と言いつつ大変褒めてくれた。これは ABCL の研究が後によく参照される起因になったと今でも思っている。また、いくつか容易な質問もしてくれた。Bobrow も何か簡単な質問をしてくれた。私の講演の終わりに「ABCL/1 の言語マニュアルを 30 冊ほど持ってきたから、欲しい方にはある分だけお渡しします」と叫んでいた。私の講演がセッションの最後で、すぐに休憩に入ったが、20 人ほどの質問者が壇上の私の前に列を作って、質問やコメントをくれたり、マニュアルをもらいに来てくれたりした。これで、成功を実感した。

第 3 回の OOPSLA には渡部卓雄氏と共著の「Reflection in an Object-Oriented Concurrent Language」という論文もアクセプトされた。いまでも、この論文はかなり creative なものだと思っている。以後 ECOOP や OOPSLA にはほとんど毎回論文がアクセプトされ、1990 年にカナダの Ottawa で開催された ECOOP と OOPSLA の合同会議では、私はプログラム委員長に選出された (プログラム委員会は、パリの Rank Xerox 社で行った)。さらに、1993 年までに、自分たちの論文だけを集めた論文集を 1 冊、Concurrent Objects の論文だけを編集した論文集を 2 冊、全て MIT Press から出版することができた。

5.6 三位一体の行方

一杉・柴山両氏の実装は比較的簡単な擬似並列でプログラム実行をさせるものであったが、本格的な並列計算機上で、大きな問題を効率よく処理する実装ではなかった。この間、私は東京大学理学部情報科学科に移り、松岡氏という新たな強力な助手を得た。彼のお蔭で、より広範な研究分野に手を染めることになり、大いに勉強にもなった。並列計算機上の本格的実装であるが、はじめ学生で後に助手になってもらった田浦健次朗氏によって、国際的に新規性の高いテクニックが考案され、それらが、AP1000 という富士通製の分散メモリ・マルチコンピュータ上で稼動するコードを生成する田浦氏のコンパイラの実現に盛り込まれた。マルチコンピュータの CPU 数は最高 512 個で、この点でも当時世界一の記録をもっていた。田浦氏が主著者でアクセプトされた米国の並列コンピュータや並列ソフトウェアの学会のうち 4, 5 個は、日本人として初めてアクセプトされたものである。

一方、八杉昌宏氏は電気工学科から情報科学の博士課程に入学してきた学生で、非常にユニークな考えをする人物であった。当時電総研にいた坂井修一氏が設計した EM4 という並列コンピュータの特徴の一つに、継続 (continuation) をハードウェアでうまくサポートする機構があった。八杉氏はこれに目をつけ、この機構をうまく用いて異なるプロセッサにある異なるプロセス/オブジェクト間のメッセージ送信やメモリ起動を大変速くする研究をしてくれた。

5.7 Linear Logic

言語研究の三位一体の最後である意味論(+型)であるが、すでに述べたように、Gul Agha が状態をもたないモジュールを仮定して、遷移システムによりその意味論を与えていた。私は、遷移システムではなく、並列オブジェクト群が状態をもち、それがメッセージを授受しながら状態を変えてゆく現象を、もっと自然に形式化したかった。当時は Robin Milner が π 計算を提唱し、英国を中心にヨーロッパでは、並列計算のモデルは π 計算がまさに帝国主義的な席卷状態であった。 π 計算を用いれば並列オブジェクトの挙動を記述できるような気もしたが、なんとなく π 計算の bandwagon に乗る気がせず、別の方法を模索していた。そこで遭遇したのが、フランスの Girard の Linear Logic であった。これは論理オペレータの種類が多いから、うまくいかもしれないと直感した。このことを、当時助手をしてもらっていた小林直樹氏に話したら、10日もかからないうちに、Linear Logic の fragment をうまく使って、巧妙かつ自然な論理システムを作ってくれた。彼は、その後この論理システムに phase semantics を付けて、その完全性を証明した。

これは良い論文になると思い、早速ジャーナル用の論文を書き、Formal Aspects of Computing に投稿した。この間、当時まだ Imperial College にいた game semantics で知られる Samson Abramsky が我々のグループを来訪する機会があったので、この仕事をキチンと彼に説明して、納得してもらった。投稿した論文の査読者の一人に必ず Abramsky がなると予想されたからである。この甲斐あってか、投稿論文は1回の照会でアクセプトされた。驚いたことに、我々の論文がジャーナルにプリントされると、最後のページの終わりに「Recommended by Robin Milner」と記されていた。小林氏はその後、多様な型の導入とその応用に研究をシフトしていった。これで、三位一体の3つ目も達成したことになった。

5.8 応用プログラム

言語研究の正四面体を考えると、三位一体の上に、その言語を用いて問題解決を行う、あるいは応用プロ

グラムを書くという問題が残っていた。これについては、大きな問題として、N-body 問題を Barnes-Hut の方法で計算するということと、RNA の準安定な 2 次構造を全て構成するというある種の探索問題を解くということを行なった。N-body 問題では、質点の数が数万個程度のもので相当よい結果がでたが、N-body 問題専用の計算機の出現などで顕著な結果とはならなかった。RNA の 2 次構造の発見については、Computational Biology 誌に 3 編の論文を公刊することができた。

さらに、Maryland 大学のシステム研究所に招聘されて、将来宇宙での長期滞在に利用されることとなる宇宙ステーションの動力学研究(姿勢・安定制御)に取り組んだ。宇宙ステーションを構成する物理モジュールと、その物理モジュール間を結合するコネクタとを並列オブジェクトで表現し、モジュールに動的に力を加えて姿勢制御する問題をモデル化する。それによって、最終的には解析的ではなく、並列オブジェクト群によるシミュレーションで解を求めるといった問題に挑戦した。この問題は 2ヶ月の招聘期間では終わらず、また招聘してくれた Krishna Prasad 教授の大学院生が急に大学をやめて就職することになり、このプロジェクトは立ち消えとなってしまった。いまでも、再開したいプロジェクトである。

6 「並列オブジェクト」の次は?

6.1 型・型推論

我々の学科では、演習(必修)でまず Scheme と ML 言語(あるいは OCaml)を教える上に、私の 3 年生への講義「言語モデル論」では ML の型推論や多相型の理論を、加えてコンパイラの演習では ML 風の言語のコンパイラを作らせるところまで教える。当然の結果として、型について比較的強い興味を持って研究室にやってくる学生が多い。一方、型システムの研究は近年相当に進歩している。通常の型のみならず、実行時のメモリ管理・変数スコープ・オペレーション適用順序などの分析まで型システムで表現し、それをソースコード分析に使い、メモリ使用の最適化・不正使用の防止などに使おうとする研究が行われている。有用な理論は、歓迎すべきものである。しか

しながら、自分自身でそのような型理論を展開していく力をもち合わせないので、この方面では、小林直樹氏や住井英二郎氏が去った今、創造的な研究の継続には多少困難を感じているのが実情である。

6.2 ソフトウェア・セキュリティと実用研究

平成 11 年 11 月、文部科学省の科学研究費補助金に、特定領域「社会基盤としてのセキュアコンピューティングの実現方式の研究」の代表者として申請し、幸い平成 12 年 9 月から 3 年半の期間、6 大学 9 つの研究グループの構成で、年間 1.2 億円の研究資金を得ることができた。この領域研究は、基本的には、ネットワークでつながったサーバーに存在する基本ソフトウェアの安全性を保証するための方法・手段を研究・開発するものである。この特定領域の代表者をさせていただき、またこのグループ研究の参加者の研究成果を勉強させていただき間に、ソフトウェア・セキュリティについて私自身比較的詳しくなってきたので、今後はソフトウェア・セキュリティの研究を発展させたいと思っている。

最後に、これまで長年多くの論文や著書を発表してきたが、この年齢になると現実の世界にもっともっとインパクトのある研究開発に携わりたいという欲求が強くなっている。もちろん論文のための論文を書いてきたつもりはないが、ある意味で論文以上の新しい概念と技法を体現した、現実の社会にインパクトの強いソフトウェアを構築する研究により大きく注力していきたいと思っている。

7 おわりに代えて

以上、私事を多少交えながら、私の研究を省みさせていただいた。古い記憶にたよった部分も多く、多少の勘違いもあると思うので、それについてはご容赦をお願いしたい。

謝辞

つぎに列挙させていただきます先生方や皆様（敬称略）は、私のこれまでの研究生活や学術的な方向付けに大変貴重な影響を与えてくださいました。ここに記して、心底より感謝申し上げます。菊池誠、森口繁一、和田英一、島内剛一、甘利俊一、伊理正夫、五十嵐滋、長尾真、淵一博、片山卓也、森村良典、Carl Hewitt、Kristen Nygaard、Barbara Liskov、Albert Meyer、Gerald Sussman、Vaughan Pratt、J.C.R. Licklider、井上博允、Patrick Winston、Hilary Putnam、Mike Gordon、Gul Agha、Alan Schnieder、Henry Baker、Eliot Moss、Rod Burstall、C.A.R. Hoare、Robin Milner、Krishna Prasad、伊藤貴康、鈴木則久、大堀淳、宮内久男、Dennis Tsichritzis、Brian Smith、Gregor Kiczales、Guy Steele、Luc Steels、中島玲二、古川康一、柴山悦哉、竹内彰一、上田和紀、松岡聡、小林直樹、田浦健次朗、増原英彦、加藤和彦、渡部卓雄、関口龍郎、住井英二郎、大山恵弘（順不同）。本稿を綿密に読まれ文章上の改良点を多数ご指摘いただいた上田和紀先生に深謝する。