

ADVANCED RESEARCH PROJECTS AGENCY (高等研究計画局 / ARPA)

Washington 25, D.C. April 23, 1963

メモ： 銀河間計算機ネットワークの メンバーおよび関係者向け

Memorandum For Members and Affiliates of the Intergalactic Computer Network

FROM: J. C. R. リックライダー

Translated by: 山形浩生 (hiyori13@alum.mit.edu)

SUBJECT: 来る会議での討議内容

Executive Summary

インターネットの生みの親の一人とされるリックライダーが全地球的、宇宙的な壮大なネットワーク構想を抱いていたという根拠としてよく持ち出される文書。ただし実際の中身は、宇宙はおろか地球全体といった話はほぼ出てこない。計算機センターの相互運用を高めるための標準化や RFC 的なコンセンサスの必要性をめぐる漠然としたアイデア。言語の標準化、ネットワークのプロトコルめいたものの必要性、さらにネットワーク上のリソース共有の方法について議論しようという、会議のためのアイデアメモ。

まず、パロ・アルトでの 1963 年 5 月 3 日予定の会議を延期したことについて真摯にお詫びする。ARPA の指揮統制研究局は、ちょうど即座に開始すべき新たな責務を割り当てられたので、今後一週間丸ごとそれに専念しなくてはならなくなったためだ。この優先事項は外部

リックライダー「銀河間計算機ネットワークのメンバー向けメモ」(1963)

から強制されたものだった。5月3日を予定していた諸君に迷惑をかけたのは本当に申し訳ない。今週はこれからずっとケンブリッジにいることになるので、会議の日時と場所を5月10日にパロアルトにリスケするよう、この同僚たちにお願ひする。

この会議の必要性と目的は、私が直感的に感じたもので、何か明確な構造をもったものではない。この事実は、以下の文を読んでもらえれば露骨にわかってしまうだろう。それでも、全体としての事業と、その中の各種活動の相互作用の可能性について、多少の背景となる材料を提示してみようとは思ふ。その全体事業を何と呼ぶべきかは、上の題名からもご明察の通り、自分でもよくわからないのだが。

そもそも、我々の中には個別の(個人、そして/あるいは組織の)野心、努力、活動、プロジェクトがいろいろあるのは明らかだ。それらは、思うに、情報処理の技芸または技術、知的能力(人間、人間=機械、機械のもの)の発展、科学理論へのアプローチという点と、何らかの形で関連しているはずだ。個別の部分は、少なくともある程度は相互に依存し合っている。前進するためには、それぞれの活発な研究はソフトウェアの基盤と、ハードウェア設備が必要となるが、それはその人物が自分だけで、現実的な時間内に作り出せるものよりも複雑で広範なものとなる。

こうした個別の目標を追求するため、グループの各メンバーは監視ルーチン、言語、コンパイラ、デバッグシステム、文書化方式、重要な計算機プログラムを実効的に用意することになる。これらはおおむね、汎用的に役立つものだ。この会議の目的の一つ——おそらく主要な目的——はこうした活動でお互いに役に立つような可能性を検討することだ——だれが何について他のだれに依存しているかを見極め、グループ内で他のメンバーのどんな活動から、他の人がおまけの便益を得られるかを見極めることになる。もちろん、価値だけでなく費用も考慮することは必要だ。それでも、計画を完全に固める前に、お互いの暫定的な計画を観ておくのは、不利益よりも利益のほうがずっと多いように思える。これは別に、プログラムの互換性を最大化するために、何やら硬直したルールや制約の仕組みにみんなを従わせるべきだとか論じるつもりではない。

しかし、いくつか予想される活動の主要部分を全部黒板に書き出して、見てみるべきだとは思ふ。そうすれば、それをしない場合よりも、ネットワーク全体としての決まりがどこで役にたち、それぞれのグループの利益に個別に任せるのが最も重要なのはどこなのかについて、もっとはっきりするとは思ふ。

リックライダー「銀河間計算機ネットワークのメンバー向けメモ」(1963)

もちろん「グループの利益」とは何かを決めるのはむずかしい。が、私自身(あるいは ARPA の) 目的と「グループ」の目的をごっちゃにしかねないとはいえ、ある意味で、グループやシステムやネットワークの必要としているものを、いくつか挙げてみよう。

プログラミング言語、デバッグ言語、TSS 制御言語、計算機ネットワーク言語、データ・ベース (またはファイル保存抽出言語)、さらに他の言語もあるだろう。こうしたものが乱立してしまうのを禁止したり制約したりするのは、いい考えかどうかはわからない。だがこれらの言語の間で、「訓練の移転」を促進するのが望ましいという点については、疑問の余地はあまりないように思う。こうした移転を促進する一つの方法は、各種言語の設計と実装において登場する、恣意的またはかなり恣意性の強い決断を行うときに、グループのコンセンサスに従うことだ。例えば、「何かの内容」「何かの内容の種類」を示すシンボルが、個人ごとあるいはセンター毎にちがうのは、まるで意味がない。ある言語システムのサブ言語集合において、現実的に実現可能な限りの統一性を維持するのは望ましいように思う。たとえば Q-32 上の JOVIAL に関連したプログラミング、デバッグ、TSS 制御言語、あるいは Q-32 計算機用の Algol (それが開発され、JOVIAL セットとはちがったものになったとするなら) に関連した各種システムや、7090 か 7094 用 FORTRAN 関連の言語セットなどは統一性を持っていいはずだ。

いまの段落を口述したことで、以前よりもはっきり認識するようになったことだが、関連する言語セットの中で統一性を実現するという問題を困難にしているのは、各時点において、ある計算機上で稼働している TSS はたった一つしかないが、その上で同時に使われているプログラミング言語は、それに関連するデバッグ言語とともに、複数あるということだ。TSS 制御言語は、どれか一つのプログラミング言語/デバッグ言語のペアとしか強い相関を持ってない。だからシンタックスに関する限り、それぞれの計算機設備やシステムについては「推奨」言語を決めて、TSS 言語はその推奨言語と整合させるべきかもしれないようだ。意味のほうを考えるなら——少なくともある記号をある制御機能と関連させるという話であるなら——複数のちがった個別語彙を使う、複数のオペレータの用途に供するのは、可能ではあるが、不便かもしれないな。いずれにしても、この領域では問題、あるいは問題群があるように思う。

ネットワーク化された計算機の制御の問題においても、似たような問題がある。おそらくこちらのほうがむずかしいだろう。複数のセンターがネットでつながれ、各センターがきわめ

リックライダー 「銀河間計算機ネットワークのメンバー向けメモ」(1963)

て独自化されていて、独自の特別な言語と、独自の特別なやり方を持っている場合を考えよう。すべてのセンターが、「どんな言語を話しますか？」といった質問をするときに、何か言語、あるいは少なくとも慣行について合意しておくのが望ましいか、必要ですらあるのでは？ 極端な話として、この問題は基本的にSF作家が論じている次のようなものと同じだ。

「まったく相関性のない『知能を持つ』存在同士でそのようにコミュニケーションを開始しようか？」 だがここでは、相関性のなさについてあまり極端な想定をしたくはない（知能についてはいくらでも極端な想定をしよう）。もっと実務的な問題は次の通り：ネットワーク制御言語は、TSS 制御言語と同じものなのか？（もしそうなら、共通の TSS 制御言語があることになる）。ネットワーク制御言語は、TSS 制御言語とはちがっているのだろうか、そしてネットワーク制御言語は、複数のネット接続された設備の間で共通なのか？ ネットワーク制御言語などというものはあり得ないのか？（たとえば、すでに稼働しているネットワークのどんな部分にでも好き勝手に接続するよう、各人が単純に自分の計算機を制御して、つないだら適切なモードにシフトすればいいのか？）

いまのいくつかの段落で、ややこしい話のど真ん中にいきなり飛び込んでしまったようだ。別の出発点からアプローチしてみよう。明らかに、この事業のメンバーの中には、FORTAN [ママ]、JOVIAL、ALGOL、LISP、IPL-V（または V-I, V-II）をコンパイルする既存プログラムを改変するようなコンパイラ（複数かもしれない）を用意しているところがあるだろう。いま挙げたそれぞれや、私が予想もしなかった何かについて複数のものがあれば、そうした予想されるプロジェクトの互換性を検討しておくといはずだ。さらには、想定されている活動を見て、その固有の特徴が何かを調べ、望ましい特徴の集まりを定義して、それを一つの言語や一つのコンパイラシステムに入れ込むのに何か意味があるかを考えるのは、望ましいんじゃないかと私は少なくとも思う。ALGOL や JOVIAL の要素の可能性として、リスト構造の特性が重要なのだという議論には感心している。リスト構造を核とした言語構築をやるのと同じくらい、リスト構造的な特徴を既存の言語に組み込む活動もすべきだという議論は一理ありそうだ。

システム全体の計算機がすべて、あるいはそのほとんどが統合ネットワークの中で協働で動く場合はごくまれだということになるかもしれない、というのはわかっている。それでも、統合ネットワーク運用能力を開発するのは、おもしろいし重要に思える。私が漠然と考えているそんなネットワークが運用にこぎつけたら、少なくとも大型計算機 4 台、小型計算機が 6 台か 8 台かな、さらに大量のディスクファイルや磁気テープユニット——さらにはリモートコンソールやテレタイプ局が山ほど——がすべて同時に稼働することになる。

リックライダー「銀河間計算機ネットワークのメンバー向けメモ」(1963)

この問題へのアプローチは、個別利用者の観点からやるのがいちばんよさそうだ——その人が何を持っていそうで、何をやりたそうかを考え、その要件が満たされるようなシステムをどう作るかを考案しようというわけだ。利用者が持ちたい、またはやりたそうなこととしては、こんなものがある:

(仮に、私が CRT ディスプレイとライトペンとタイプライターを持つコンソールの前にすわっているとしよう)。「リスニングテスト」というテープにある実験データ集合を読み出したい。そのデータは「実験3」という名前だ。このデータは基本的に、各種の S/N 比の比率だ。こうした実証関数はいろいろある。この実験はマトリックス型で、聞き手は数人、提示方法は数種類、信号周波数も複数、その期間も複数だった。

まず、「理論的」な曲線を実測データにフィットさせたい。これを予備的にやって、割合と S/N 比の理論的な関係についてどの基本関数を選ぶべきかを調べたいわけだ。「カーブフィッティング」と名付けた別のテープには、直線やべき乗関数、累積正規曲線にフィッティングを行うルーチンを保存してある。だが他のものも試してみたい。最初は、私がプログラムを持っている関数から始めてみよう。困ったことに、私はよい散布プロットプログラムを持っていない。それを拝借したい。単純な直行座標で十分だが、軸の目盛の細かさや、その凡例は指定したい。その情報をタイプライターから入力したい。システム内のどこかに、これに適した散布プロットプログラムはあるだろうか?

現状のネットワークのドクトリンを使うと、まずは自分のいる設備を調べて、それから他のセンターを調べることになる。たとえば私が SDC で働いていて、使えそうなプログラムがバークレーのディスクファイルで見つかったとしよう。私のプログラムは JOVIAL で書かれている。だがシステムを通じて見つけたシステムは FORTRAN で書かれている。それを移植可能なバイナリプログラムとして持ってきて、それを「導入時点」または「実行時」に、自分のカーブフィッティングプログラムのサブルーチンとして使いたい。

いま述べたステップを実現できたとして話を進めよう。直線、二次曲線、四時曲線などではデータがうまくフィットしないことがわかった。最高のフィットでもオシロスコープで見たらかなりひどい。

計測データを累積正規曲線にフィッティングさせると、どうしようもないほどひどくはない。検出プロセスについての何かの理論と整合させるよりも、少数のパラメータで制御でき

リックライダー「銀河間計算機ネットワークのメンバー向けメモ」(1963)

る基本関数を見つけるのがこちらの眼目なので、システム内のだれかが、利用者の提供する曲線にデータをあてはめたり、たまたま累積正規曲線っぽい(だが非対称)関数が組み込まれたりしているような、カーブフィッティングプログラムを持っていないか知りたい。仮にいろいろファイルを調べたり、あるいはマスター統合ネットワークファイルがあってそれを調べたりして、そんなプログラムはないことがわかったとしよう。したがって、正規曲線で行くことにする。

こうなると、プログラミングをしないとイケない。自分のデータは手元において、正規曲線にフィッティングするプログラムも手元に置き、拝借したプログラムを表示したい。実験の順序尺度または比率スケールの次元のそれぞれを進めつつ、各種の対象について少しずつちがった境界値群を与え、平均と分散がゆっくり変化するよう制約をかけながら、自分のデータの各種サブセットを累積正規曲線にフィッティングさせたいわけだ。そこで次にやりたいのは、カーブフィッティングのルーチンに境界値を設定するような、一種のマスタープログラムを作り、そのフィットがどれほど高いかを、図化すると同時に数値でも表示させ、ライトペンと境界値の図示と独立変数の対比をオシロスコープの画面上に表示し、いろいろやってみて、各種の(私から見れば)無理のない設定を試してみる。実データに対して、すでに述べたサブプログラムを使い、繰り返しプログラムを行い、うまく動くようにしたい。

仮にやっとこれが成功して、まあまあの結果が出て、実証データと「理論」曲線の両方を示すグラフを写真に撮って、新しいプログラムを将来も使えるように保存したとしよう。このプログラム群全体のシステムを作り、それを「制約境界正規曲線フィッティングシステム」と題して保存しておきたい。

が、そこで仮に、私の直感的には自然なシステム命名方法が、ネットワークのプログラム命名に関する一般指針に適合していないとしよう。そうした慣行との不一致については教えて欲しいと思う。だって私はプログラムライブラリや有用なデータの公開ファイルの話となると、良心的な「組織的人間」なんだから。

いまの話で、私はいくつかのネットワークの特徴を活用しているはずだ。念頭にある要件を満たすようなプログラムを探すシステムを通じて、情報の抽出を行った。おそらくこれは、記述子か、それにかなり似たものに基づくシステムで、遠からぬ将来には、自然言語に対する計算機適用に基づくものになるはずだ。だが前衛的な言語学の能力をある程度利用できる

リックライダー「銀河間計算機ネットワークのメンバー向けメモ」(1963)

たら嬉しいだろう。拝借したプログラムを使うにあたり、自分のプログラムと拝借したものとの間に、何らかのリンクを行った。できれば、これはあまり手間をかけずにできてほしい——願わくば、リンクまたはそのリンクを行うための基盤は、そのプログラムが使っているシステム[ママ]の一部として持ってこられたときに設定されていてほしい。データは何も拝借していないが、それは自分の実験データで作業をしていたからだ。何か理論を試そうとしていたなら、プログラムだけでなく、理論も拝借したかったはずだ。

計算機が私のプログラムを処理するとき、その処理は私の居場所だと想定した SDC の計算機で行われただろう。だがそれは単なる憶測レベルにとどめてもいいはずだ。高度なネットワーク制御システムがあれば、そのデータを送信して別のところのプログラムに処理させるのか、あるいはプログラムを読み込んでデータを処理させるのかという選択を自分ですることはない。自分でその決断を下すのに、少なくともいまのところは特に反対する理由もないが、原理的には、その判断を計算機またはネットワークが下したほうがいいように思う。自分の作業が終わったら、いくつかをファイルにしてしまい、それを他の人にも有用となる形で行った。これはおそらく、何か慣習をモニタリングするシステムが関与することになる。そうしたシステムは初期段階には、ほぼまちががなく機会[ママ] 処理に加えて人間の基準判定者も入っているはずだ。

いま挙げた(残念ながら長ったらしい)例は、言わば例のそのまた例となるよう意図したものだ。こうした例を大量に集めたい、あるいはだれかに集めてほしい。そしてそれがどんなソフトウェアやハードウェア設備を含意しているかを検討したい。こうした例の相当部分から出てくる含意の一つは、きわめて大量のランダムアクセスメモリとなることは、十分に承知している。

さてこの問題全体にさらに別のアプローチをするなら、いくつか思いつきをつれづれなるままに述べていこう(ここでちょっと邪魔が入ったので、議論はそろそろ切り替えなくてはならない)。まずは「純粋プロシージャー」の問題がある。JOVIAL の新バージョンはプログラムを「純粋プロシージャ」の形でコンパイルするとのこと。

他のセンターの他のコンパイラもそうやってくれるだろうか? 第2に、別のセンターからそのセンターに向けられた要求の解釈の問題がある。入ってくる言語を、その受け手側のセンターが運用する形式の命令や質問に翻訳する、何か通訳システムみたいなものを漠然と思い描いている。あるいはもちろん翻訳は送り手側でやってもいい。また別のやり方は、協

リックライダー 「銀河間計算機ネットワークのメンバー向けメモ」(1963)

調が実に見事に行われてみんな共通の言語をしゃべり、同じ形式群を使うことになるというものだ。第3に、公開ファイルの保護と更新の問題がある。だれかが改変途中のファイルからの材料は使いたくない。我々相互の活動の中で、何か軍のセキュリティ区分と似たようなものがあるかもしれない。その場合にはどう扱おうか？

次に、インクリメンタルなコンパイルの問題がある。パーリスは、「スレッド化したリスト」の話で、その問題やそれに関するコンパイル-試験-再コンパイル問題を基本的に解決したんだっけ？

ハードウェア側では境界レジスタ型問題、あるいはもっと一般的には目盛保護問題が Q-32 では解決が高価となり、他のマシンでは高価な上に解決困難となるのではと懸念しているし、コアと二次記憶装置との間で情報をスワップまたは転送する問題が、7090 や 7094 では高価で困難になるのが心配だ——そして高速スワップや転送なしには TSS があまり役に立たないのではと心配している。この問題についての最高の考えとは何だろう？ 我々の各種または集会的な計画はこれについてどうなっている？

この長ったらしい例で含意されているのは、ランタイムでサブルーチンをリンクするという問題だ。呼び出しそのものは、単純なディレクトリを通じて簡単に行えるが、システム変数を扱うのはそれほど簡単ではなさそうだ。ひょっとすると原理的には簡単でも、テーブルや単純なアドレッシング方式を使ってシステム変数を実行時にリンクさせるのは、ひょっとすると実現困難なのだとするべきかもしれないな。

この大作をそろそろ終えねばならない。飛行機の時間に遅れそうだからね。ARPA の指揮統制局が持っている、人間=計算機のインタラクションや TSS や計算機ネットワークの改善についておさらいするつもりだった。だがたぶんみんな、こうした問題についての ARPA の基本的な関心はわかっている[ママ]と思うし、必要なら会議の席でそれを手短におさらいしてもいい。実際問題として、私の見たところ、軍はいま作られつつある設備を我々が活用しようとしたときに生じる多くの、いやほとんどの問題について、解決策を大いに必要としている。

我々の個別の活動の中で、協調的なプログラミングと運用に十分な利点が示されて問題が解決され、軍の必要とする技術を我々が生み出せるようになってほしいと思う。問題が軍事の文脈でははっきり登場するのに、研究の文脈では出てこないなら、ARPA はそれを個別に

リックライダー「銀河間計算機ネットワークのメンバー向けメモ」(1963)

処理するような手を講じればいい。だがすでに述べたように、願わくば問題の多くは軍事の文脈だけでなく研究の文脈でも本質的な重要性を持つはずだ。

まとめると、繰り返すがこれまでの議論で指摘しようとした、セットの中の問題や課題についてかなりしっかりと議論すべきだと私は思っている。すべての問題は指摘できていないかもしれない。願わくば、会議での議論はここで終えようとしている試みよりも多少はまとまりのあるものとなりますように。

付記

原文は以下：

<https://www.thekurzweillibrary.com/memorandum-for-members-and-affiliates-of-the-intergalactic-computer-network>