ICOT Technical Memorandom: TM-1216

TM-1216

Proceedings of the FGCS
PROJECT EVALUATION WORKSHOP
第五世代コンピュータシステム
プロジェクト評価ワークショップ

内田　俊一

September, 1992

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

Proceedings of the
FGCS PROJECT EVALUATION WORKSHOP

第五世代コンピュータシステムプロジェクト評価ワークショップ

June 3, 1992
平成4年6月3日

Tokyo Prince Hotel, Magnoria Hall
東京プリンスホテル　マグノリアホール

ICOT
Institute for New Generation Computer Technology
財団法人　新世代コンピュータ技術開発機構

Table of Contents

Preface

It has been 10 years since the Fifth Generation Computer Systems (FGCS) project started in 1982. Throughout this period, we have been conducting research and development on hardware and software technologies, and basic theories for new computer systems, combining highly parallel processing and knowledge information processing using logic programming. Furthermore, we aimed at the development of a prototype of FGCS integrating hardware and software technologies to be able to objectively evaluate these newly developed technologies. The research and development contained many technical problems that were difficult to solve. However, rapid growth in the research fields of logic programming and knowledge information processing produced many distinguished researchers. Through cooperative research with these researchers, we were able to solve most of these problems and complete the prototype system.

At the FGCS'92 conference, we successfully demonstrated the major achievements of the project as well as presenting papers on many important research results. Furthermore, we introduced a new scheme that enables us to disclose the software developed in the project free of charge to contribute to the advancement of computer science and technology in the world. Taking this opportunity, we intended to hold a project evaluation workshop inviting leading researchers in the fields related to FGCS technologies to ask them to evaluate the software and hardware technologies and scientific and political contributions of the project.

In the workshop, we asked those of the invited researchers from overseas who have conducted collaborative research work to present their opinions on the evaluation of this project. This was followed by discussions with other workshop participants. This proceedings consists of the program of this workshop, which includes an explanation of the aim of the workshop and a list of participants, evaluation reports containing various opinions written by the researchers who made presentations at the workshop, and some additional papers contributed after the workshop.

Their opinions mentioned various aspects of the project; the impact of this project on government sponsored projects in many countries, contributions to computer science and technology, evaluation of technical achievements in logic programming and parallel processing, future directions of knowledge processing applications, disclosure of developed software, future role and extension of ICOT, and many others.

The various opinions summarized in this proceedings are not limited to the technical aspects of FGCS related technologies but extend to political aspects such as what Japanese national projects must do in connection with world research and development on advanced technologies. Thus, they indicate important conditions that Japan must consider in carrying out advanced research and development. All these opinions are very constructive and helpful. We, the organizers of this workshop, deeply thank the researchers who presented or joined the discussions at the workshop.

It is our sincere hope that this proceedings will serve as a valuable reference for the researchers, project managers, and government people who are interested in the evaluation of the FGCS project from technical, academic and political aspects and may be engaged in advanced research and development in the future.

August 1st, 1992 Shunichi Uchida and Kazuhiro Fuchi

まえがき

第五世代コンピュータプロジェクトが、1982年に開始されてから、10年が経過した。この間、私達は論理プログラミング言語を中核として、高度並列処理技術と知識処理技術を結び付けた新しいコンピュータ技術の創出をめざし、新しいコンピュータのハードウェアやソフトウェア技術、および、その基盤となる理論の研究開発を進めてきた。また、プロジェクトの最終目標として、これらのソフトウェアやハードウェアを一体化したプロトタイプシステムを試作し、新たに開発された技術の客観的な評価を行なうことを目指した。

その研究開発は、多くの技術的に困難な問題を含んでいた。しかし、幸いにも、本プロジェクトと共に、論理プログラミングや知識処理の研究分野は、大きな成長を遂げた結果、多くの研究者を生み出していた。私達は、これらの世界中の多くの研究者の協力を得て、それらの問題を解決することができ、その結果、プロトタイプシステムを完成することができた。そして、国際会議FGCS'92で、このような本プロジェクトの成果を発表するとともに、ソフトウェアやハードウェアのデモンストレーションを行なった。また、このプロジェクトで開発されたソフトウェアを無償で公開し、世界のコンピュータ科学技術の発展に貢献する道を開くことができた。

私達は、この機会を捉え、第五世代コンピュータ技術関連の研究分野の指導的研究者を海外より集めてワークショップを開催し、本プロジェクトの創出したソフトウェアやハードウェア技術、および、本プロジェクトの達成した学術的、政策的な貢献について、評価してもらうことを計画し、この評価ワークショップを開催した。このワークショップでは、海外より招いた研究者のうち、特にこれまで本プロジェクトにおいて実施した、いろいろな国際共同研究に協力してもらった研究者を選び、本プロジェクトの評価についての意見を発表してもらい、議論した。

このプロシーデングスは、このワークショップの趣旨説明を含むプログラムとこのワークショップでの発表内容を後から論文にしてもらったもの、さらに、本プロジェクトに関する意見を手紙でおくってくれたものなどを含んでいる。その内容は、本プロジェクトが海外の政府主導のプロジェクトに与えた影響や、コンピュータ科学技術の分野への貢献に関するもの、論理プログラミングや並列処理などの技術内容の評価に関するもの、知識処理の応用の将来に関するもの、ソフトウェア無償公開に関するもの、今後ICOTは何をなすべきかという将来展開に関するものなど、多岐に渡るものであった。ここにまとめられたさまざまな意見は、第五世代コンピュータに関連する技術的な面に留まらず、日本の国家プロジェクトが世界の先端技術開発の中で、何をなすべきかという政策的な面にも及んでおり、今後の日本が研究開発を行なう場合考慮すべき条件を示すものとなっている。

その意見は、極めて建設的なもので、私達、このワークショップの主催者は、これらの意見を寄せられた発表者と議論に加わった参加者に、深く感謝するものである。このプロシーディングスが、第五世代コンピュータプロジェクトの技術的、学術的、そして、政策的な評価に関心を持つ、今後、先端研究開発を担うであろう、研究者、技術管理者、政策担当者などの参考になれば、幸いである。

'92年8月1日　　　　内田 俊一、 淵 一博

Evaluation Workshop Program

Vth
ICOT

FGCS Project Evaluation Workshop Program

FGCSプロジェクト評価ワークショップ
プログラム

June 3, 1992
平成4年6月3日

Tokyo Prince Hotel, Magnoria Hall
東京プリンスホテル　マグノリアホール

Vth ICOT

Institute for New Generation Computer Technology
財団法人　新世代コンピュータ技術開発機構

Guide to the FGCS Project Evaluation Workshop

1 Aim of the workshop

The aim of this workshop is to collect opinions and comments on the FGCS project. We expect that many of the leading research scientists from overseas who have cooperated and encouraged this ten-year project will express their evaluations of this project, based on the final results of the FGCS project, through FGCS'92 and this workshop. At the same time, we would like to take advantage of this opportunity to express our appreciation of their cooperation and encouragement over the last ten years.

Domestic participants will consist of Japanese research scientists who have been charged with evaluating this project, and those who belong to organizations that have participated in this project, such as ETL, NTT, computer manufacturers, MITI, and ICOT. Domestic participants are willing to receive opinions and comments from overseas research scientists and to reflect these comments in the final evaluation report.

We also expect to have discussions on the issue of disclosing the project's achievements to the world and ICOT's further activities after March 1993, and to receive comments on such issue. The workshop should be closed so that participants can freely exchange their opinions with each other from various viewpoints.

Workshop Chair:	Kazuhiro Fuchi
Vice-chair :	Shunichi Uchida and Yoshihisa Ogawa

2 Date and Place

Date and time:	June 3, Wednesday 16:00~21:30
Place:	2F Magnoria Hall Tokyo Prince Hotel (FGCS'92 conference site) (Please take an escalator up to the demonstration site, and turn to the left)

FGCSプロジェクト評価ワークショップのご案内

1　開催の趣旨

このワークショプは、本プロジェクトの研究開発、および、種々の国際交流の実施に当たり、御支援と御協力を頂いた海外の研究者の方々に、本プロジェクトの最終成果について、いろいろな観点から、評価して頂き、御意見を頂くことを目的としております。また、同時に、これまでの十年間に、本プロジェクトに賜わった御支援、御協力に、謝意を表する機会とさせて頂くことを意図しています。

国内からは、本プロジェクトの評価作業を担当されている日本側研究者の方々、および、本プロジェクトを実施した、通産省、ICOT、および、関連研究機関やメーカ関係者が参加し、海外研究者の方々の評価に関する御意見を聞き、議論し、その結果を、評価作業に反映させることを予定しています。このほか、本プロジェクトのソフトウェア無償公開や、今後の展開に関しても、合わせて議論し、御意見を頂くことを意図しています。

尚、本ワークショップは、密な議論を行なうために、クローズド・ワークショップと致します。

ワークショップ委員長	淵 一博
代表幹事	内田 俊一
	小川 義久

2　開催日時／場所

日時:	6月3日(水) 16:00 ~ 21:30
場所:	東京プリンスホテル FGCS'92 会議会場内
会議室:	2F マグノリアホール
	(デモ会場行きのエスカレータに乗り、降りてから左へ)

3 Workshop program

1) Session-I	16:00 - 18:30	Chairman: S.Uchida and Y.Ogawa	
Greetings		H.Hiroshige	Executive director, ICOT
Aims of the workshop		K.Fuchi	Director, ICOT research center

10 presentations on the evaluation of the FGCS Project
by invited participants from overseas
(10 min. per presentation plus 5 min. Q&A)

1) D.Warren	Bristol Univ.
2) W.Bibel	Darmstadt Univ.
3) K.Clark	Imperial College
4) R.Feldmann	NIH
5) G.Kahn	INRIA
6) M.McRobbie	ANU
7) E.Shapiro	Weizmann Inst.
8) R.Stevens	ANL
9) S.Sundström	SICS
10) S.Tärnlund	Uppsala Univ.

2) Buffet Party 18:30 - 20:00 Hosted by K.Fuchi
Place: 3F Restaurant "Golden Cup"

3) Session-II	20:00 - 21:30	Chairman: K.Fuchi and S.Uchida	
Greetings		R. Hayashi	Director Electronics policy division, MITI
Presentation		T. Saeki	Deputy director Electronics policy division, MITI

"On the free distribution of the software and future direction of FGCS technology"

Q&A

* Consecutive interpretation between Japanese and English is provided.

3　ワークショップのプログラム

1) 第一部 (16:00 - 18:30)　　司会: 内田、小川

挨拶　　広重 専務

趣旨説明　　淵 所長

海外研究者よりの評価についての講演と質疑

、 講演予定者 10 名:(一人あたり: 10 分講演 + 質疑 5 分の予定です。)

1) D. Warren	Bristol Univ.
2) W. Bibel	Darmstadt Univ.
3) K. Clark	Imperial College
4) R. Feldmann	NIH
5) G. Kahn	INRIA
6) M. McRobbie	ANU
7) E. Shapiro	Weizmann Inst.
8) R. Stevens	ANL
9) S. Sundström	SICS
10) S. Tärnlund	Uppsala Univ.

2) 淵所長主催夕食会 (18:30 - 20:00)

場所: 3F レストラン: ゴールデンカップ

3) 第二部 (20:00 - 21:30)　　司会 淵、内田

挨拶　　林　電子政策課 課長

説明　　佐伯　電子政策課 技術班長

議題　　ソフトウェアの公開と今後の技術的展望

質疑応答

* 第二部は、日英の逐次通訳つきで行ないます。

4 Participants

4.1 Overseas

U.S.A.

J. Alan Robinson	Syracuse Univ.
Y. T. Chien	NSF
Richard J. Feldmann	NIH
Rick L. Stevens	ANL
Ross Alan Overbeek	ANL
Mark E. Stickel	SRI
Evan Tick	Univ. of Oregon

U.K.

Robert A. Kowalski	Imperial College
Keith Clark	Imperial College
David H.D. Warren	Univ. of Bristol

France

Gilles Kahn	INRIA
Alain Michard	INRIA
Hervé Gallaire	GSI (ex. ECRC)

Sweden

Sten-Åke Tärnlund	Uppsala Univ.
Siwert Sundström	SICS
Seif Haridi	SICS

Israel

Ehud Shapiro	The Weizmann Inst.of Science

Germany

Wolfgang Bibel	Univ. of Darmstadt

Australia

Michael A. McRobbie	ANU
Robin B. Stanton	ANU

EC

Jean-Marie Cadiou	ESPRIT

4　参加者リスト

4.1　海外参加者

アメリカ合衆国

J. Alan Robinson	Syracuse Univ.
Y. T. Chien	NSF
Richard J. Feldmann	NIH
Rick L. Stevens	ANL
Ross Alan Overbeek	ANL
Mark E. Stickel	SRI
Evan Tick	Univ. of Oregon

英国

Robert A. Kowalski	Imperial College
Keith Clark	Imperial College
David H.D. Warren	Univ. of Bristol

フランス

Gilles Kahn	INRIA
Alain Michard	INRIA
Hervé Gallaire	GSI (ex. ECRC)

スェーデン

Sten-Åke Tärnlund	Uppsala Univ.
Siwert Sundström	SICS
Seif Haridi	SICS

イスラエル

Ehud Shapiro	The Weizmann Inst. of Science

ドイツ

Wolfgang Bibel	Univ. of Darmstadt

オーストラリア

Michael A. McRobbie	ANU
Robin B. Stanton	ANU

EC

Jean-Marie Cadiou	ESPRIT

4.2 Domestic

Chair of MITI committee for development of basic computer technology

Hidehiko Tanaka	Univ. of Tokyo

MITI project evaluation working group

Hozumi Tanaka	Tokyo Institute of Technology
Fumio Mizoguchi	Science Univ. of Tokyo
Kokichi Futatsugi	ETL
Shigeki Goto	NTT basic research laboratories
Atsuhiro Goto	NTT software lab.
Hideharu Amano	Keio Univ.

ICOT project promotion committee

Hiroshi Kashiwagi	Director, ETL
Seishiro Tsuruho	Executive Manager, NTT software lab.

ICOT technical committee members
and invited participants from computer manufacturers

Nobuyoshi Miyazaki	Oki
Yasuji Obuchi	Sharp
Tsutomu Kawada	Toshiba
Mamoru Umemura	NEC
Kouichiro Ishihara	Hitachi
Junichi Tanahashi	Fujitsu
Masato Yamazaki	Matsushita
Masakazu Soga	Mitsubishi
Chiyoji Tanaka	Mitsubishi
Takao Uchara	Fujitsu
Shigeo Nagashima	Hitachi

4.2 国内参加者

MITI 諮問委員会委員長

田中 英彦	東京大学

MITI 諮問委員会評価 WG 委員

田中 穂積	東京工業大学
溝口 文雄	東京理科大学
二木 厚吉	電子技術総合研究所
後藤 滋樹	NTT 基礎研究所
後藤 厚宏	NTT ソフトウェア研究所
天野 英晴	慶應義塾大学

ICOT-PPC 委員

柏木 寛	電子技術総合研究所所長
鶴保 征城	NTT ソフトウェア研究所所長

メーカ技術委員 (代理含む) および招待者

宮崎 収兄	(沖) 総合システム研究所
小渕 保司	(シャープ) 情報技術開発センター
河田 勉	(東芝) 情報システム研究所 所長
梅村 護	(日電) C&C システム研究所
石原 孝一郎	(日立) システム開発研究所 主管研究員
棚橋 純一	(富士通) 研究所基礎システム研究部門 部門長
山崎 正人	(松下) 情報通信東京研究所 基礎研究部長
曽我 正和	(三菱) 情報電子研究所 所長
田中 千代治	(三菱) 開発本部 開発部 参与
上原 貴夫	(富士通) 研究所 情報処理研究部門 部門長
長島 重男	(日立) 中央研究所 第 6 部長

MITI

Ryozo Hayashi	Director, Electronics policy division, MITI
Toshinori Saeki	Deputy director, Electronics policy division, MITI
Hiroaki Uji	Electronics policy division, MITI

ICOT participants

Hiroichi Hiroshige	Executive director
Kazuhiro Fuchi	Director, ICOT research center
Takashi Kurozumi	Deputy director, research center
Koichi Furukawa	Deputy director, research center
Yoshihisa Ogawa	Manager, Research planning dpt.
Shunichi Uchida	Manager, Research dpt.
Shigeto Kitamura	Manager, Administration dpt.
Hiroshi Hara	Manager, International relations dpt.

ICOT deputy managers, chiefs, deputy chiefs, and leaders

Yoshinobu Murasawa	Ryuzo Hasegawa	Kazuo Taki
Keiji Hirata	Takashi Chikayama	Kazunori Ueda
Kazumasa Yokota	Hideki Yasukawa	Akira Aiba
Masayuki Fujita	Yuichi Tanaka	Katsumi Nitta
Nobuyuki Ichiyoshi		

ICOT Secretariate

Kazuhide Iwata	Hiroshi Sato	Kazuyuki Tani
Junzou Watanabe	Yasushi Kambayashi	Satoshi Tojo
Hisanori Tatae	Noriko Namikoshi	Emiko Higuchi
Kumiko Karakawa		

MITI 機情局

林 良造	電子政策課 課長
佐伯 俊則	電子政策課 技術班長
宇治 浩明	電子政策課

ICOT 参加者

広重 博一	専務理事
淵 一博	研究所長

黒住 恭司 (総括担当次長)	古川 康一 (研究担当次長)
内田 俊一 (研究部長)	小川 義久 (研究計画部長)
北村 重人 (総務部長)	原 弘 (調査国際部長)

ICOT 研究部長代理、研究室長、室長代理、テーマリーダ

村沢 好信	長谷川 隆三	瀧 和男
平田 圭二	近山 隆	上田 和紀
横田 一正	安川 秀樹	相場 亮
藤田 正幸	田中 裕一	新田 克己
市吉 伸行		

ICOT 事務局およびサポートスタッフ

岩田 和秀	佐藤 博	谷 和純
渡辺 淳三	神林 靖	東条 敏
湛 久徳	浪越 徳子	樋口 笑子
唐川 久美子		

Summary of the Workshop

in Japanese

Vth
ICOT

第五世代コンピュータ・プロジェクト評価ワークショップ概要
Summary of the Workshop

1 開催日時・場所

6月3日（水）午後4時－10時30分
東京プリンスホテル　マグノリアホール

2 議事概要

本ワークショップには海外から22名、日本側が約30名の参加者があった。海外からの参加者を代表する10名による本プロジェクトの評価に関する発表及びソフトウェアの公開と今後の技術的展望についての質疑が行われたが、その内容は、本プロジェクトに対して全体としてきわめて高い評価を与えるもので、今後、本プロジェクトで開発された技術を活用するための提案がなされた。その内容は、およそ次のようであった。

- 第五世代コンピュータ・プロジェクトの成否に関しては、技術的成果、および社会的なインパクトの両面において、高い評価が得られた。
- 技術的インパクトとしては、特に、知識情報処理を論理型言語という新しい言語によって並列マシン上で展開することにより、たとえば並列定理証明によって、これまでは解けていなかった定理の証明に成功したことなどの、さまざまな科学的・技術的進歩が得られた点が評価された。
- また、社会的インパクトとしては、人工知能や計算機科学の分野における研究促進、これまで日本では行われていなかった国際共同研究、研究者の育成といった点が高く評価された。
- また、並列アーキテクチャや、知識表現言語が、いずれもまだできあがったばかりであるため、それらの評価が未着手である等の指摘もあった。
- ソフトウエアの無償公開を評価する声も多かった。
- 今後の技術展開についても本プロジェクトを今年度で終了するのではなく、今後もフォローアップが必要であるとの見解が多く発表された。

今後の技術展開の内容としては、無償ソフトウエアのサポート、PIM上で動作するソフトウエアを市販の計算機に移植することでより広い利用を図るといった本プロジェクトで開発された技術の普及活動を必要とする声が多く、普及を促進するためのインパクトのある応用研究や、さらにこのような広範な利用の結果得られた知見を、さらに発展させるために基礎研究を後継プロジェクトとしてロングレンジで継続し、国際共同研究の輪を広げるべきであるとの結論であった。

第1部: 本プロジェクトの評価

第五世代コンピュータ国際会議1992に参加した上での、本プロジェクトの評価に関する発表

[発表要旨]

1) D. Warren (ブリストル大学・英国)

本プロジェクトは、全体として大きな社会的なインパクト、たとえば他の類似プロジェクト・組織等の促進と、科学的なインパクト、たとえば国際的に論理プログラムをプロモートした点があったと認められる。また、組織としてのICOTは3年で研究員が入れ替わるという点、研究員はメーカよりの出向という制約があったと思う。

プロジェクト全体としては、ハードウエアおよびオペレーティングシステムの研究に労力を割きすぎ、その分、知識処理や応用への展開が不足したように見受けられる。

結論としては、100MLIPSの推論性能を達成し、GHCというエレガントな言語を確立したほか、先端的な並列記号処理の応用ソフトウエアの開発に成功している。これらを見ても本プロジェクトは成功したと認められる。また、今後についてはより小さく、柔軟な形態で、ICOTは継続されるべきであると考える。

2) W. Bibel (ダルムシュタット大学・独)

プロジェクトの成果を評価する際には、予算担当者としての立場、ジャーナリストとしての立場、経済学者としての立場などがあるが、私は科学者としての立場から考えたい。

科学者の立場から見ると、一様な「論理」という枠組みを採用し、それに「並列」を組み合わせている点に特徴があった。また、プロジェクトの前提として、論理をベースにしたこと、システムを論理型マシンともいうべき専用機をもとに組み上げた点などがあった。

3) K. Clark　(インペリアル大学・英国)

本プロジェクトの持つインパクトとしては、国際的に人工知能や計算機科学の分野に対する、Alvey, Esprit, MCC, ECRC, SICS他の発足において見られたような政府の投資を引きだしたこと、私たちの研究分野全体の地位や、人工知能や論理型プログラミングの地位を高めたこと、日本において人工知能や論理型言語の研究者を育成した点が上げられる。

一方、昨日のヘラルドトリビューン紙によれば、プロジェクトは失敗であったとされているが、しかし、この記事の内容は誤ったものである。

本プロジェクトは、日本が誇るべきものであるし、ソフトウエアの無償化は良い考えであると思う。しかし、開発されたソフトウエアはより普及されている並列計算機に移植されるべきであるし、ヘラルドトリビューンのような誤った見方に対して、従来の言語よりも生産性、メンテナンスの容易さといった点を応用プログラムの開発手法を作り上げることによって示すべきであると考える。

4) R. Feldmann　(国立衛生研究所・米国)

米国国立衛生研究所 (NIH) は ICOT と 1988 年以来、コンタクトをとってきており、遺伝子情報処理とタンパク質の折り畳みに関する共同プロジェクトを行ってきた。その結果をもとにコメントをすると、国際的に用いられている言語は英語であり、プログラミング言語は C である。したがって、マニュアルは英語であるべきであるし、C と論理型言語とのリンクも考慮されるべきであると考える。現在、ICOT との研究は、KL1 言語で PIM を、C 言語でグラフィックインターフェース、さらにシリコングラフィックスを使って一体化しており、当初よりもスムーズに使えるようになった。いずれにせよ、このような共同研究プロジェクトは今後も継続されるべきものであると考える。

5) G. Kahn (INRIA)

本プロジェクトのスコープはたいへん広く、それを KL1 を通じて統合するものであったといえると思う。また、各室長発表やデモは良くできたものであったが、発表がうまくいった研究だけを集めており、やや政治的な感じがあった。うまくいかなかったものものもその原因を分析し、発表してもらえればより良かったと思う。結論としては、まず 10 年という年月はプロジェクトとしては特に長いものではなかったという点で、古川さんには賛成しない。また、本プロジェクトによって技術的には非常に多くのものを得ることができた。また、社会的には基礎研究を促進し、国際的な協調を進めてきた。このような観点からプロジェクトを突然に終了するべきではないと考える。また、このプロジェクト以前は日本の科学者とはまったく対話がなかったのだから、ここで生まれた友情も大事にすべきと思う。

6) M. McRobbie (オーストラリア国立大学)

われわれは特に定理証明の分野において共同研究を行ってきている。そのような経験から、次のように結論する。

まず、ソフトウェアの無償公開については、私の知るかぎり、初めてのことであり、大きく評価したい。また、四次元コンピュータ・プロジェクトは ICOT から多くのものを学ぶことができると考える。われわれは、定理証明の共同研究を行い、最近 ICOT の PIM/m を使って、これまで解けなかった群論の問題に関する定理を解くことに成功している。PIM/m 上の定理証明システムによって、並列マシンを使えばこの問題が高速に解けることがわかり、将来の研究方向が明確化してきた。共同研究の継続、ICOT の開発したソフトウエアの市販のマシンへの移植などを行い、さらに多くの人が利用できるようにするべきである。

7) E. Shapiro (ワイツマン研究所・イスラエル)

私は ICOT の招へい研究員の第 1 号であり、そのときこのプロジェクトの目標として説明を受けた並列知識処理技術や、その成果を世界の研究者に広く公開するといった方針は、今日、ほぼそのとおりに達成されていると思う。したがって、本プロジェクトは科学的技術的にはまちがいなく成功したと考える。しかし、国際レベルのインパクトという点においては、当初想定されていたレベルに達したとは思わない。すなわち、ICOT における応用研究はまさに始められたばかりであり、得られた科学的技術的成果を示す「これは絶対」といった決

定的な応用がまだ出来ていない。また、ICOTのソフトウェアはICOTの作ったハードウェアでしか走らせることが出来ないものである。したがって、このようなソフトウェアを公開し、普及させるためには、市販のマシンに移植し、メインテナンスを行い、ソフトウェアの信頼性を向上させるべきである。また、これらを利用しようとする国際研究グループのサポートを行うべきである。

8) R. Stevens　(アルゴンヌ国立研究所・米国)

本プロジェクトの評価をアメリカ流に行うとすれば、このプロジェクトが基礎研究のプロジェクトであったのか、あるいは開発研究プロジェクトであったのかをはっきりさせる必要がある。すなわち、基礎研究においては肯定的であれ否定的であれ明確な結果があれば成功といえる。開発研究プロジェクトにおいては開発成果が製品化一歩手前までまとまって、始めて成功したと言うことができる。さらに、どのような結果も得られなければ、これはどちらにおいても失敗である。

本プロジェクトは、この双方の面を持っており、また、米国では、日本の研究開発のプロセスが良く知られていないために本プロジェクトをどう評価するかについて混乱がある。

ICOTの今後について、以下を提言する。

並列処理、性能評価、知識ベースに基づいたプログラミングシステム、記号処理と数値計算の融合といった基礎研究について長期間のプロジェクトを行うべきであると考える。また、基礎研究の結果については、そのすべてをプロジェクトの初期段階から公開すべきである。さらに、研究目的・予算・マネージメントといった多くの面を共同で行う「本格的な共同研究」を推進すべきである。

その場合の研究開発の内容はソフトウエアに集中すべきであり、ハードウエアやOSは企業に開発させるべきである。

9) S. Sundström　(スウェーデン計算機科学研究所・スウェーデン)

本プロジェクトは論理プログラミングの技術が有望な市場を形成し得る研究分野であることを示し、多くの研究計画や研究機関がICOT設立の結果、生まれることになった。また、ICOTで生み出された多くの考え方や概念はICOT以外の研究機関でさらに研究開発されることとなった。

結論として、本プロジェクトは実際、価値のあるものであり、スウェーデンと日本との緊密な協力関係を確立することに成功した。さらに本プロジェクトは「スウェーデン計算機科学研究所(SICS))」の設立の動機となった。

ICOTのソフトウエアは、将来的には、広く使われている計算機に移植するべきであり、それによって普及をはかり、その結果を言語やソフトウエアにフィードバックするべきであると考える。

10) S. Tärnlund　(ウプサラ大学・スウェーデン)

ICOTの研究結果は世界的には最高水準にあると考える。また、まさにコンピュータ科学におけるブレークスルーを成し遂げた。PIM、PIMOS、KBMSそして、評価ソフトウエアが一体化してできており、プロジェクトは、大きな成功を収めた。特に、私は論理型言語の実行速度が、当初の40KLIPSが100MLIPSにま

で高速化されたのには驚嘆させられた。結論としては、第五世代コンピュータ技術の研究はさらに継続しなければならないと考える。

第2部：ソフトウエアの公開と今後の技術的展望に関する質疑応答

約2時間近く質疑応答が行われたが、その中心はソフトウエアの公開に関するサポート体制と、今後のフォローアップ・プロジェクトとの関連についての質問が多かった。質問およびコメントの趣旨としては、ソフトウエアは公開してもその後のメンテナンスが重要であり、さらに、これらのソフトウェアを広く利用してもらうためには、市販の計算機上への移植が不可欠であること、また、このようにして広く利用された結果をさらなる研究開発に反映させるためにも後継プロジェクトが必要であるというものであった。

また、海外からの参加者から、後継プロジェクトの実現が、ソフトウェア普及のみならず、国際協調、さらなる研究展開といった意味からも、是非とも必要であるとの声が強かった。また、この後継プロジェクトの性格にも質問はおよび、企業の参加の可能性の有無、年数などについての議論も行われた。

以上

(参考)

第五世代コンピュータ国際会議 1992 参加者数

平成4年6月18日

会議参加者数	約1,600名
(内海外26カ国	約170名)
デモンストレーション見学者数	約1,900名

ＦＧＣＳプロジェクトの評価に関する報告

Wolfgang Bibel
Technical University Darmstadt
Germany

１９９２年6月4日

要約

このレポートでは、第五世代コンピュータシステム（FGCS：Fifth Generation Computer System）プロジェクトの成果について簡潔に評価を試みる。まず最初に、私の評価の背景を明らかにするため、このプロジェクトに参加した研究者たちと私の関係について説明する。次に、評価の規準を明確にする。３章の「評価」では、技術的な成果について主に以下の３つの観点から記述する。

- マシンとソフトウェアを同時に設計することに関して、ロジックを、統一的かつ効率的な枠組みとして提示。
- 並列性の採用により示された非常に大きな進歩。
- ロジックでプログラムを作成することによる効率の向上を証明。

これらの点を考慮し、またこのプロジェクトのその他の多くの成果を考慮した結果、このプロジェクトはたぐいまれな成功を納めたと判断できる。さらに4章では、このプロジェクトの基盤に横たわる主な仮説についていくつか考察し、結論としてそれらがすべて現実のものとなり、正しく証明されたことを述べる。最後の章では、ICOT の将来と FGCS の精神を受け継ぐ研究について多少の意見を記述する。ICOT は一定期間は存続させるべきであり、また情報技術の基礎研究のために日本が何らかの研究機関を設立するように提案する。

1　ＩＣＯＴの研究と私の関係

光栄なことに、私は FGCS プロジェクトの要人とその準備の段階から、もっと正確には 1979 年 8 月に東京で開かれた IJCAI-79 から接触を持つことができた。その会議で淵博士との議論を通して、プログラミング、問題解決、および知識工学のためのコンピュータを構築して使用するために、ロジックが統一的かつ包括的なアプローチとしての潜在的に利用できるとの共通の認識を持った。

1981 年に、私は、最初の FGCS 会議において６つの招待講演の１つを行う名誉を得た。この講演において、論理の視点から、ソフトウェア開発に対する私の見解を示した。この概念は、プログラム合成に MGTP などのツールを適用することなど（すなわち MENDELS ZONE）、現在 FGCS プロジェクトの中で開花しつつある。

1990年の始めに約2週間にわたってICOTを訪問し、FGCSプロジェクトの枠組みの中で遂行されている多くのプロジェクトの各側面についてより詳細に知ることができた。このときは、ちょうど、プロジェクトの最終段階の作業計画が最終的に決定されるときだった。これは千載一遇のチャンスであり、私は、計画作業に携わるスタッフに対して、演繹推論と自動的な定理証明が、プロジェクトの終了時にPIM上で動作する予定の基本ソフトウェアのための有望なアプリケーション領域であることを強調した。

この13年間に何度か日本を訪問したばかりでなく、ICOTから何人か訪問者を迎える機会も得た。1981年9月に淵博士と故元岡教授がミュンヘンのTechnical Universityを訪れ、第1回の会議の概要を説明してくれた。その後、ICOTから何人かの研究者（少なくとも10人）がミュンヘンの私の研究所を訪れ、また最近はやはりドイツのダルムシュタットの研究所を訪れ、成果、経験、および意見の交換を積極的に行った。その中には、古川博士、長谷川博士、藤田博士などがいる。

1991年にドイツのBirlinghofenのGMDで開催された「演繹推論に関するドイツ－日本ワークショップ」では、お互いの成果について特に詳細にわたって意見を交換することができた。このワークショップでは、私はコーディネータの役を担った。日本からは8人の研究者（大部分はICOTから）が参加し、ドイツからは12人の研究者が参加した。

さらに、IJCAI、AAAI、ロジックプログラミング、自動演繹推論などの会議では、ICOTから参加した研究者と出会う機会が数多くあった。

これらのことを述べたのは、このプロジェクトの最終成果に対する私の評価が、このプロジェクトのライフスパン全体の作業を私がかなり熟知していることに基づくものであると示すためである。また、まさにこのプロジェクトの開始時点から、私が大きな興味と共感を持ってこのプロジェクトを追跡していたことを示すためである。この点から、私の判断は片寄っているとも考えることでできるであろう。しかしながら、科学に対して同じ所見を持つことに問題はないはずである。

2 評価規準

FGCSプロジェクト程度の規模のプロジェクトを評価する方法にはいくつか考えられる。これらの方法のどれを適用するかによって、得られる評価結果も変わってくる。誤解の可能性を排除するため、まず最初に現在の状況を踏まえて、私がこれらの方法の内からどれを採用したいかを明らかにする。

1. 財政担当者の評価方法では、1981年に発行されたこのプロジェクトの当初のレポートに立ち返って、それをチェックリストとして使用し、現在実際に達成された目標のパーセンテージを計算することになるであろう。私は、FGCS程度の規模の基礎研究プロジェクトでは、このような評価はほとんど意味がないと強く信じている。したがって、このレポートではこの方法は採用しない。にもかかわらず、私の持っている感覚では、このプロジェクトは、その一里塚となる主な目標のすべてを実際に達成したということができる（ただし、あまり重要でない話題の中には、何らかの理由で途中で断念されたものもある）。

2. ジャーナリストの評価方法では、何らかの過程やできごとによって発生した社会的な期待に対する成功の度合いを計ることになる。FGCSプロジェクトは、たしかに人によってまったく異なる各種の期待を与えた。たとえば、日本の報道機関は、アメリカの報道機関とはかなり異なる理解を示した。（かなりの量の国家資金が投入されるた

め）このような規模のプロジェクトでは社会的な意見は重要であるが、技術的な評価という意味ではこの点にあまり留意する必要はないと思う。少しわき道にそれるが、FGCS プロジェクトは、アメリカおよび世界各地において以前ほど好意的に報道されているわけではない。これは、このプロジェクトに対する期待が誇張されていた結果であり、また広い意味での複雑な政治的理由によるものである。過剰な期待には、当初の FGCS レポートに政治的な理由から知識工学の夢が語られていたが、これがこのプロジェクトの最終目標と誤解されることとなったという事実も含まれる。

3. 経済学者の評価方法では、このプロジェクトの成果によってもたらされる経済的なインパクトの大きさによって成功の度合いを計ることになる。この方法は、期待されるインパクトが現れるまでに何年もかかる基礎研究プロジェクトでは、やはり意味を持たないものである。現時点における経済的なインパクトはたぶんほとんどゼロに近いため、この点からはこのプロジェクトは完全な失敗と評価しなければならないだろう。しかし、長い目で見れば、たぶん（そしてかなりの確率で）非常に大きな経済的な効果を持つこととなるであろう。

4. 私が採用するのは科学者としての視点である。この視点では、この計画の本当の効果は何であるかを評価する。つまり、この計画の効果と、この計画を遂行しなかった場合の状態を比較する。その効果として私が理解しているものの中には、科学的な成果、技術的な進歩、構築されたシステムやマシンなど、このプロジェクトがもたらしたすべての変化だけでなく、日本や国際的な研究コミュニティ、またその点において世界全体で引き起こされた変化が含まれる。このような評価に加えて、私は、このプロジェクトの状態の何かを変化させることによって、本当の効果を改善できるかどうかについても熟考している。

3 プロジェクトの本当の効果

このプロジェクトは、かなり異なる種類の成果および効果を生み出した。主な効果の 1 つは、政治的社会的な性格のものである。その他は、インフラストラクチャに関するものである。そしてもちろん、出版物、システム、マシンなどの形式の科学的な成果もある。これ以降、この順序でこれらのすべてについて議論したい。

私の知る限り、1981 年の FGCS 会議は、日本で開かれた会議のうち、高いレベルの国際的な知名度を得て世界的な注目を集めた最初のものであった。世界は初めて、日本が将来の鍵を握る技術の 1 つにおいてリードを奪うのではないかという感触を持った。明らかに、これらの感触は深刻な憂慮と重なっていた。人々の中には、技術的な戦争とまで過剰に反応し、また発言する人もいた。今日、ある人々は再び過剰に反応している。彼らは、自分たちの恐れが実現しなかったのを見て、このプロジェクトが失敗したものと理解している。

差し引きして、私は、政治的な本当の効果としては成功だったと判断している。日本は、世界をリードするビジョンを持っていることを証明した。一方、日本の行動は賢明であり、その成果を国際社会に対して無償で提供した。つまり、自分自身の利益のためだけでなく、人類の利益のためのリーダーとしての役割を果たした。しかし、この視点は、政治的には確固たる意見として定着したものではないことに注意する必要がある。将来、政治的な進展を誤れば、この肯定的な状況は簡単に壊され得るものである。

社会的には、日本がイニシアティブをとった効果として、情報処理技術が人類の福祉のために重要であることを世界中が認識した。FGCS プロジェクトの直接的な結果として、

アメリカ（MCCなど）、ヨーロッパ（ECRC、SICS、ESPRIT、Alveyなど）およびその他の地域に主要な研究機関やプロジェクトが存在する。これらの研究機関やプロジェクトは、すべて情報処理技術の発展に寄与している。

FGCSプロジェクトの大きな成果および成功の1つに、日本の情報処理技術の研究および開発のインフラストラクチャに及ぼした効果がある。非常に賢明な構成により、日本の企業および大学の数百人にものぼる若い研究者たちが、実際に最先端の情報処理技術を学ぶことができた。これは、ICOTと企業および大学とのつながり、さらに研究者は研究機関からICOTに出向して一定期間留まった後に、その研究機関に戻るというICOTの方針によるものである。これらの研究者たちは、単なる教育で得られる以上のものを学び得るばかりでなく、国際貢献の場に触れることができた。そして、現在はそれぞれの研究機関で、この交流を継続する可能性を享受している。このプロジェクト以前は、日本は、国際的な研究社会に参加するうえで問題を抱えていた。したがって、私は、この効果は、将来、科学的にもまた経済的にも情報技術において日本がリーダーなり続けるために非常に重要なものであると考える。ドイツ人として、私は、我国がこの点、特にマシンの設計とアーキテクチャの分野で同じような賢明な行動をとることを期待する。

このプロジェクトは、日本のインフラストラクチャを変革したばかりでなく、国際的な研究コミュニティのインフラストラクチャも変革した。かつて欧米の科学者たちは日本の科学者を仲間として真剣に考慮したことはなかった。現在日本の情報技術の科学者たちは、他の欧米の科学者と同様に対等のパートナーと見なされている。日本の科学者たちは、その成果を以前よりも活発に国際的なジャーナルや会議に発表している。逆に、日本のジャーナル（第五世代コンピュータシステムジャーナルなど）や日本の会議（FGCSなど）は、世界中の科学者たちから、科学的な成果を発表する名誉ある場と考えられている。1997年に日本がもう一度、最も影響と規模の大きい情報技術に関する会議IJCAIを開催するという事実は、世界中が日本の研究者たちとの関係の重大さ認識していることを示すものである。

最後に、そして最も重要なことであるが、私は、この傑出したプロジェクトの科学的な成果から本当に深い感銘を受けた。我々の分野で初めて、ハードウェアとソフトウェアに対して単一の言語KL1を通した統一的アプローチが存在するようになったのである。

一方、PIMの枠組みで構築されたすべてのマシンは、KL1プログラムの実行という特殊な目的で設計されており、このことがその実行を非常に効率的なものにしている。一方、ソフトウェアはすべてKL1上で構築されている。このことは多くの理由から非常に優れた成果であり、以下にそのいくつかを述べる。

KL1は論理型言語（の1種）であることを思い出してほしい。論理型以外の計算処理の世界では、2つの理由からロジックは計算のための有効な手段としては無視されている。それは、状態依存ソフトウェア（オペレーティングシステムなど）に対する不適合性が疑われていること、およびその非効率性である。FGCSプロジェクトは、この2つの心配が間違っていることを証明してみせた。まず最初に、PIMマシンのオペレーティングシステムのカーネルはKL1で実現されたものの1つである。そして、オペレーティングシステムの残りの部分はPIMOSと呼ばれる大規模なソフトウェアシステムとして構築されており、これはその中に包含するオペレーティングシステムの機能を使用してすべてKL1で書かれている（約133K行のコード）。PIMOSが証明したように、ロジックは、状態に依存するシステムに対処するための形式として十分に使用可能なものである。2番目に、アプリケーションソフトウェアシステム（MGTPおよびその他多く）が顕著な方法で証明したように、KL1の処理系は非常に効率的である。

KL1上に構築された基本ソフトウェアの1つに知識ベース管理システムKappa-Pがあり、Kappa-P上には知識表現言語Quixoteが構築されている。KL1のような論理型言語

が知識表現に適しているのは驚くに当たらない。しかし、注目に値すべきことは、その基本がオペレーティングシステムのそれとまったくに同じだということである（訳注：オペレーティングシステムと知識ベース管理システムがともにKL1上に構築されていること）。したがって、最適化の作業をマシン上のKL1の処理系に集約することが可能であり、PIMOSとKappa-Pを自由に切り離すことができるという利点がある。

したがって、統一的かつ効率的な枠組みとしてのロジックを、このプロジェクトの優れた成果の第1としてあげることができる。その側面としては以下のものがある。

- ハードウェアとソフトウェアの設計を、情報処理の問題全体の重要な部分としてみる視点。

- 知識処理に関して、推論と知識が同等な重要性を持つこと。

主な成果の2番目として、並列性の重要性がある。このプロジェクトで作成された多くのソフトウェアはまず最初に逐次的な方法でコード化されたため、並列化によって経験したスピードアップは明白かつ劇的なものがあった。多くの場合、線形に近いスピードアップが実現した。明らかにこれは偶然発生したものではない。国際的な研究コミュニティは、このような重要な実験を行い、またこの励みとなる成果を得たことについて、日本の研究者たちに感謝している。KL1の実行について、最終的にこのプロジェクトが性能目標の100MLIPS（1秒間の論理推論回数）を達成できたのは、この並列性のためである。

私の判断で3番目の主な点は、信頼性の高いソフトウェアの効率的作成のためのフォーマリズムとしてのロジックの容易性である。過去2年半に、KL1で直接または間接に書かれたソフトウェアの量は、信じられないほど大量である。デモンストレーションで見たように、これらの大規模システムの実行では何も問題が発生しなかった。この成果を公正な方法で評価するためには、すべてのソフトウェアが並列実行のために記述されていることを心に留めなければならない。我々は、皆、並列プログラムを作成する困難さを知っており、また私は、世界中でこのような規模で並列ソフトウェアを作成したプロジェクトを知らない。従来のソフトウェア作成の経験（逐次的であり、並列的なものは1つだけ）では、同じ機能を持つソフトウェアを作成するのに明らかに多くの時間が必要である。少なくとも私にとっては、このプロジェクトの成果の1つは、ソフトウェア開発はロジックによって桁違いに改善されるという主張を証明したことである。

これらの成果、およびその他の重要な成果に加えて、数多くの詳細な成果が存在するのは明白であり、それは発表された論文や動作中のシステムに見ることができる。その正確な数がいくつであろうと、我々は、日々の研究業務から、日本の研究者たちの多くの成果が我々の研究で重要な役割を果たしていることを知っており、これはFGCSプロジェクトなくしては考えられないことである。

4　プロジェクトの仮説の評価

プロジェクトの本当の成果がもっとより良いものになり得たのではないか、別の方法があったのではないかなどについて、熟考してみたくなる人もいると思う。このことについて、この節で簡潔に述べる。

まず最初に、ロジックだけに賭けることは、これまで説明した成果が示すように本当に有利な賭けである。また、ハードウェアから知的な機能およびプログラムまで、垂直に統合された方法で問題に対処するのも同様に正しいことである。

なかには、並列ロジックに基づくアプローチを、このようにプロジェクトの遅い段階で評価するのは間違いであるという人もいる。一方、この議論には指摘すべき点がある。それは、単に魅力的なアプリケーションからのみ強い印象を受けるという理由で、計算機科学界の人々が、これまでこのアプローチの詳細な点にほとんど興味を示さなかったことである。一方、マシンと基本ソフトウェアを完成させなければ、採用したアプローチの正しさをどうやって証明できるであろうか。私は、これは視野の狭い議論だと思う。このような長い期間の目標を定めてプロジェクトを遂行し、それを比較的長い期間にわたって変更せずに保つのは、日本的な方法の大きな長所である。

もう1つ問題となる可能性のある点は、特にKL1を効率的に実行するためにのみ構築された、PIMマシンの特殊化した性質である。同じような目的のためならば、汎用目的の並列マシン（この会議の招待講演で発表されたJ-machineなど）も同じように役に立つのではあるまいか。これは良い質問であるが、現時点では十分に満足のいく方法で答えることはできない。将来、正確にこの項目に焦点を当てた実験で得られた結果でのみ議論が可能である。特殊化が差を生まないとしたら、それは私にとって驚きであり、私は差が生じると信じている。特に論理型言語（手続き型言語や関数型言語ではない）の実装では、必要な論理演算を実行するために特殊化されたハードウェアを使用するのは非常に重大なことである。現在は、マシン開発の進歩がまだこの利点を先へ押しやっているようである。特殊化されたマシンが動作可能になるまでに、すでに汎用マシンが性能上で非常に高度化しており、特殊化されていない不利を十分に補うことができるほどになっている。しかし、私は、最終的にはプログラミングのスタイル（論理型、関数型、手続き型）に専用のマシンが、特にロジックプログラミングの場合には重要になると確信している。したがって、FGCSプロジェクトで遂行された実験は、将来のロジックベースのマシン設計にとって非常に重要なものとなるであろう。この評価を判断する場合には、信頼性の高いソフトウェアの迅速な作成に関するロジックの価値について、すでに前の項で私が説明したことを心に留めておかなければならない。なぜなら、マシンの比較を行う際に、多くの人はソフトウェアの作成にかかる工数という意味での投資を忘れているからである。

KL1の選択に関しては、これが純粋な形式での論理型言語ではないという点を考察しなければならない。この指摘は事実であるが、しかし私は、KL1が、この特定言語の設計を現時点で達成するための最善の妥協点であると信じている。この点については、将来のプロジェクトでは新しいより改善されたアプローチがとられるものと期待している。しかし、将来の科学的な進歩によってその成果がいくつかの点で改善されるのは、いつの時代でもすべてのプロジェクトの運命である。

5　将来の見通し

FGCSプロジェクトの傑出した成功を考えると、高い価値を持つICOTのインフラストラクチャや、現在達成されているマシンおよびソフトウェアに関する基盤を廃棄してしまうのは労力の無駄であると考える。言い替えれば、私は、ICOTが、限定された期間（たとえば5年）だけ何らかの形で存続するよう強くお勧めする。そして、以下の作業を遂行すべきである。

- その本質的な機能について、マシンとソフトウェアシステムを評価する。
- 多様なアプリケーションで成果を利用する。
- システムの保守管理を行う。

● 適切な新しい研究目標を達成する。

システムの保守管理は、MITIの政策ですべてのソフトウェアを自由に使用できるものとして開放するという点から非常に重要である。この政策は、国際協調にもたらす効果を考えると画期的な一歩である。もちろんこの政策は、そのソフトウェアがPIM以外の標準的なマシンで使用可能となる場合にのみ実を結ぶものである。しかし私の理解によれば、すでにソフトウェアをUNIX環境に移植する計画がある。

現在ICOTを指導しているスタッフに移動があるようである。ここで、このプロジェクトの成功に大きく寄与した研究所長の渕博士のことを特に強調しておきたい。私は、個人的な研究目標のために時間を取りたいという彼の望みを十分に理解している。にもかかわらず彼の才能を今以上に壮大な計画で役立てないとしたら、それは悲しむべきことであると思う。

基礎研究の遂行に成功したことで高い評価を得たことから、日本に基盤をおく基礎科学の研究機関の考えが私の心に浮かんでいる。この機関は、世界中の研究者を親密に結び付ける上で、現在ICOTが果たしている役割の一部を受け継ぐことができると思う。それは、世界中から第一線の研究者たちが集う場である。ついでに述べると、このような性格を持つ研究機関は、環境的に魅力のある地に置くのがよいと思う。

すでに指摘したように、私は、純粋に論理的なマシンおよびソフトウェアに向かって、重要なただし最終ではない一歩がKL1によって踏み出されたと確信している。この同じ方向で次の一歩を踏み出すには、ロジックにおける新しい方向（リニアロジックなど）を考慮に入れる必要がある。私は、従来のソフトウェア作成上の問題から、他のコンピュータ科学者たちも、最終的には論理を中心とした計算処理と知識工学分野というFGCSプロジェクトと同じ方向に向かって来ると確信している。

Reports and OHP for the Presentations

Vth ICOT

Evaluation Report

on the FGCS Project

Wolfgang Bibel
Technical University Darmstadt
Germany

4 June 1992

Summary

In this report I try to briefly evaluate the results of the Fifth Generation Computer Systems (FGCS) project. First, I describe my interactions with the researchers involved in the project in order to make the background of my judgment visible to the reader. Second, I clarify the criteria under which I undertake the evaluation. Within the evaluation itself given in section 3 my main points of appraisal of the technical achievements are

- the demonstration of logic as a uniform and efficient framework for designing machines and software at the same time;
- the enormous gains demonstrated through using parallelism; and
- the demonstration of the gains in efficiency by producing software in logic.

Considering these and many other achievements of the project I judge it as an outstanding success. In section 4 I then test some of the major hypotheses underlying the project and come to the conclusion that all of them were solid and proved successful. In the final section I have a few remarks for the future of ICOT and of research in the spirit of FGCS which include the suggestion to continue ICOT for a limited period of time and set up a Japanese Research Institute for basic research in Information technology.

1. My interactions with ICOT research

I am proud to be able to say that I have been in contact with the key persons of the FGCS project since its preparatory phase or, more precisely, since IJCAI-79 held in Tokyo in August 1979. In discussions with Dr. Fuchi during that conference it became clear to me that he and I shared the same vision of logic offering the potential for a uniform and comprehensive approach to building and using computers for programming, problem solving, and knowledge engineering.

In 1981, I was given the privilege to present one of the six invited lectures at the first FGCS conference. In this lecture I outlined my view of software development from a logical point of view, a view that is now beginning to emerge also within the FGCS project among the applications of tools such as MGTP to program synthesis (viz. the MENDELS ZONE system).

I visited ICOT early in 1990 for about two weeks and became even more intimately familiar with many facets of the projects carried out within the framework of the FGCS project. At that time the plans for the work in the final stage of the project were just finalized which gave me a unique chance to encourage those involved in the planning task to emphasize deduction and automated theorem proving as one of the promising application areas for the basic software planned to be operative on PIMs at the end of the project.

Not only did I visit Japan several times in these past thirteen years but also had I a chance to host a number of visitors from ICOT during that time period. Dr. Fuchi and the late Prof. Moto-Oka visited the Technical University in Munich in September 1981 in order to brief me for the conference. Afterwards several other researchers from ICOT (at least ten) visited my laboratory in Munich and later the one in Darmstadt, both in Germany, for an intensive exchange of results, experiences, and opinions. Among them are Dr. Furukawa, Dr. Hasegawa, Dr. Fujita, and others.

A particular extensive exchange of our respective work became possible by a German-Japanese workshop on deduction held in 1991 at the GMD in Birlinghofen, Germany, for which I was in charge of the coordination. Eight researchers from Japan (with a large proportion from ICOT) and about twenty researchers from Germany participated.

In addition there were many occasions for encounters with researchers from ICOT at conferences such as IJCAI, AAAI, Logic Programming, Automated Deduction, etc.

All this is meant to show that my evaluation of the results finally achieved in this project is based on a rather intimate knowledge of what was going on in the project over its entire life-span. It should also inform the reader that I followed this project with a great interest and sympathy from the very beginning. In this respect my judgment may be regarded as a biased one. On the other hand, what is wrong with sharing similar visions in science?

2. Evaluation criteria

There are various possible ways of judging a project the size of the FGCS project. Depending on which of these ways one applies one would get different evaluation results. In order to avoid any possible misunderstandings, I first want to clarify which among the following ways I prefer in the present context.

a. The *accountant's* way of judgment would be to go back to the original report of the project published in 1981 and, taking it as a checklist, find out the percentage of the targets that are now actually achieved. I strongly believe that in a basic research project the size of FGCS this kind of evaluation would be rather meaningless. Therefore I will not follow this way in this report. I might nevertheless mention that according to my feeling the project has indeed achieved all its core milestone targets (while some less central topics for good reasons have been dropped along the way).

b. The *journalist's* way would judge the success against the expectations which were generated in the public through whatever process or events. The FGCS project has indeed generated various different expectations depending which public we are looking at. For instance, the Japanese press understood it very differently in comparison with the press in the US. While the public opinion is important for a project this size (since public money is involved in substantial amounts), I do not think I should bother with this issue here as a technical reviewer. Just as an aside I mention that the FGCS project has now a less favorable press in the US and in other parts of the world due to exaggerated expectations associated with the project and caused by complex reasons of a political nature in a broad sense. This includes the fact that the initial FGCS report for political reasons contained a vision of knowledge engineering which by the public was misunderstood as the final targets of the project.

c. The *economist's* way would judge the success by the amount of economic impact caused by the results of the project. Again this way does not make sense for a basic research project for which the impact is to be expected not before many more years. In this respect the project would actually have to be judged a total failure since at present the economic impact is probably close to zero. In the long range, however, it might (and probably will) turn out to be enormous.

d. The viewpoint I take is that of a *scientist*.. It consists in estimating what the net-effect of the enterprise might be, i.e. the effect of the enterprise in comparison with the situation, were it not been undertaken. As effect I understand all the changes caused by the project including the scientific results, the technology evolved, the systems and machines built, but also the changes caused in the Japanese and the international research community, or in the entire world for that matter. In addition to such an estimation I speculate about whether the net effect could have been improved by changing some of the project's conditions.

3. The project's net effect

The project has produced results and effects of very different kinds. A predominent effect is political and social in nature; others are of an infrastructural kind; and of course there are the scientific results in form of publications, systems and machines. I will discuss them all in this order.

As far as I know, the FGCS conference in 1981 was the first conference held in Japan which attracted the worldwide interest at such a high level of international visibility. For the first time the world got the feeling that Japan is about to take the lead in one of the key technologies of the future. Clearly, these feelings were mixed with serious concerns. Some people overreacted and spoke even of a technological war. Today some people again overreact. As they see that their fears have not materialized, they regard the project as a failure.

On balance, I judge the political net effect as a success. Japan has indeed proved that it has the vision to take a lead for the rest of the world. On the other hand, it acted wisely and offered the results to the international public for free use, thus acting as a leader to the benefit of mankind and not only for its own self-interest. One must, however, be aware that politically the views have not settled down to a stable state of opinions. False political steps in the future could easily destroy the current positive state of affairs.

Socially the effect of Japan's initiative is that the rest of the world has recognized the importance of information technology for the well-being of mankind. The existence of major institutions and projects in the US (MCC and others), in Europe (ECRC, SICS, ESPRIT, Alvey, and others), and in other parts of the world is to be seen as a direct consequence of the FGCS project. They all have contributed to the advancement of information technology.

One of the major results and successes of the FGCS project is its effect on the infrastructure of Japanese research and development in information technology. By an extremely clever arrangement hundreds of young Japanese researchers in research institutions of industry or universities have actively learned about the latest state of the art in information technology. This is because of the many links of ICOT with companies and universities and its policy to exchange researchers in its laboratory after their temporary stay at ICOT. Not only have these researchers learned more than would have been possible by mere education, but they also were exposed to international cooperation and now enjoy the possibility to continue these contacts at their respective institutions. Since before this project Japan had some problems with opening up to the international research community, I regard this effect as one of extreme importance for the future prospects of Japanese ability to remain a leader in information technology scientifically as well as economically. As a German I wished my country would have taken similarly wise moves in this respect, especially in the area of machine design and architecture.

Not only has the project changed the infrastructure in Japan, but also the one of the international research community. While previously western scientists rarely took their Japanese colleagues into serious consideration, now Japanese scientists in information technology are considered as equal partners a par with any others. Japanese researchers present their results more than ever before in international journals and conferences. Vice versa, Japanese journals (like the Future Generation Computer Systems Journal) and Japanese conferences (like the FGCS) are regarded as esteemed stages for the presentation of scientific results for scientists from all-over the world. The fact that Japan will host again in 1997 one of the most influential and largest conferences in information technology, namely IJCAI, underlines the respect with which our Japanese colleagues. are regarded by the rest of the world.

Finally, and most importantly, I am genuinely impressed by the scientific achievements of this remarkable project. For the first time in our field, there is a uniform approach to both hardware and software design through a single language, viz. KL1.

On the one hand, the machines built under the framework named PIM all are designed for the special purpose of executing KL1 programs which makes this execution remarkably efficient. On the other hand, all software is built on top of KL1. This is an exciting achievement for a number of reasons, some of which I will mention in the sequel.

Remember that KL1 is (sort of) a logical language. The rest of the computing world ignored logic as useful vehicle for computation mainly for two reasons, namely for its alleged inappropriateness for state-dependent software (such as an operating system) and for its inefficiency. The FGCS project has given proof that both concerns are actually wrong. Firstly, the kernel of the operating system for the PIM machines is part of KL1's realization, while the rest of the operating system is built as a large software system, called PIMOS, which is all written in KL1 using the kernel operating system functions contained in it (with about 133K lines of code). Logic can well be used as a formalism to cope with systems which are state sensitive as PIMOS proves. Secondly, the realization of KL1 is extremely efficient as the application software systems (like MGTP and many others) demonstrate in a remarkable way.

The other part of the basic software built on top of KL1 is a knowledge base management system, Kappa-P, on top of which Quixote, a knowledge representation language is built. It is less surprising that a logical language like KL1 is suitable for knowledge representation. The remarkable feature, however, is that the basis is exactly the same as the one for the operating system. The optimization efforts could therefore be concentrated on the realization of KL1 on the machines with the benefits for PIMOS and Kappa-P falling out for free.

Logic as a uniform and efficient framework is thus one of the outstanding results of the project. Aspects of this are

- the view of hardware and software design as an integral part of the problem of

information processing as a whole and
- the equal importance of inference and knowledge for knowledge processing.

The other major result is the *importance of parallelism.*. Since a lot of software produced during the project was first coded in a sequential way, the speed-up by parallelization could be experienced in an explicit and dramatic way, i.e. near linear speed-up could be experienced in a number of cases. This was by far not happen to happen, so that the international research community is grateful to the Japanese researchers to carrying out this important experiment and achieve this encouraging result. It is parallelism which eventually enabled the project to meet the performance target of 100MLIPS (logical inferences per second) for execution of KL1.

A third major point in my judgment is the ease of logic as a formalism for *efficient production of reliable software*. It is nearly unbelievable how much software was produced in about two and a half years written directly or indirectly in KL1. As one could see in the demonstrations no problems arose running these large systems. In order to appreciate this achievement in a fair way, one has to keep in mind that all this software is written for parallel execution. We all know how hard it is to code parallel programs, and in fact I know of no project anywhere in the world which has produced parallel software at such a large scale. Given the experience with conventional software production (even sequential, let alone parallel one) which obviously requires much more time for producing software with the same functionality, it is obvious at least to me that one of the results of the project is a proof for the claim that software production is enhanced by logic by orders of magnitude.

In addition to these and many other important main results, there are obviously the many results of detail, available in many hundreds of published papers and operative systems. Whatever the exact number is, we all know from our daily scientic work how many of the results of the Japanese colleagues play an important role in our own research which would not be the case without the FGCS project.

4. Evaluation of the projects hypotheses

One might speculate whether the net results of the project could have been even better, would different routes be followed, a topic which I discuss briefly in the present section.

First of all, betting exclusively on *logic* has been a real bargain in all respects as the discussed results demonstrate. The same is true for dealing with the problem in a *vertically integral way*, from hardware all along through to intelligent functions and programs.

Some people argue that it has been a mistake to test the approach based on parallel logic only at such a late stage in the project. On the one hand, there is a point to this argument because so far the computing community became hardly interested in the details of the approach simply because they could be impressed only by attractive applications. On the other hand, how could one manage to demonstrate the taken approach without having completed the machines and the basic software? I think this is a shortsighted argument. It is one of the major virtues of the Japanese way of carrying out this project that such long-range goals were undertaken and kept unchanged for such a relatively long period.

Another issue of possible concern is the specialized nature of the PIM machines, built especially to run KL1 efficiently. Would not general purpose parallel machines (like the J-machine presented in an invited lecture at the conference) serve the same, if not a better purpose? I think this is a good question which cannot be answered at present in a fully satisfactory way. I only can speculate about the outcome of future experiments focussing on exactly this issue. It would be extremely surprising to me, if specialization would not make a difference, rather I believe it does make a difference. Especially for the implementation of a logical language (rather than an imperative or functional one) it may be crucial to use hardware specialized to carry out the required logical operations. At present the progress in machine development might still outway this advantage: until a specialized machine is ready for operation, the general purpose machines have become so advanced in performance that they easily compensate the disadvantage of being unspecialized. I am convinced, however, that ultimately machines dedicated to the style of programming (logical, functional, imperative) will become crucial, especially for logic programming. So the experiment carried out within the FGCS project will turn out to be extremely valuable for future logic-based machine design. In appreciating this judgment one should keep in mind what I said in the previous section concerning the value of logic in producing reliable software fast, since people tend to forget the investment in terms of people's time spent for software production in machine comparisons.

As to the choice of KL1 there might be the concern that it is not really a logical language inpure form. Although this is in fact true, I still believe that KL1 is the best compromise which could be achieved at the time of the design of this particular language. I do expect a new and improved approach in this respect in some future project; but it is the fate of any project that at some point in time its results will be improved by further scientific progress.

5. Perspectives for the future

Given the outstanding success of the FGCS project I think it would be a waste of efforts, would the valuable infrastructure of ICOT and the basis in terms of machinery and software achieved now be abandoned. In other words, I strongly recommend that ICOT shall continue to exist for alimited period of time (e.g. five years) in some form or another. Its tasks might be to

- evaluate the machines and software systems w.r.t. their crucial features;
- exploit the results in various applications;
- maintain the systems; and
- pursue appropriate new research goals.

Maintenance of the systems is especially important in view of the fact that MITI has adopted the policy to make all software available as free software which is a remarkable step which will have its effect in terms of international cooperation. Of course, this policy will bear its fruit only when the software will become available on standard machines other than PIMs; but as I understand there is anyway the plan to port it to a UNIX environment.

There will be changes in the personell currently leading ICOT. I want to emphasize the enormous influence on the success of the project which is due to its scientific director Dr. Fuchi. Although I fully understand his desire to take a rest for his own personal research goals, it would be a real pity if one would not use his talents for some other, perhaps even grander enterprise.

With having achieved such a high reputation for carrying out basic research in a successful way, the idea of a scientific institution for basic research based in Japan occurred to me. It could continue to play part of the role currently played by ICOT to maintain close links with researchers from all over the world. It could be a meeting place for first-rate researchers from all over the world. As an aside I mention that institutions of this nature would best be placed in an environmentally attractive area.

As I indicated in the previous section, I strongly believe that with KL1 an important, but not final step was taken towards a purely logical machine and software. New directions in logic (such as linear logic) will have to be taken into account for doing a next step in the same direction. In any case, I am convinced that the problems with conventional software production will bring the rest of computer scientists eventually towards the same line of a logically oriented computing and knowledge engineering discipline.

COMMENTS ON THE RESULTS OF THE FGCS project

W. BIBEL
Darmstadt, GERMANY

Personal and general backgrou

- Interaction since 1979
- Computing world in 1981
  - logic and inference for
    - = computing
    - = knowledge engineering
- Bold goals

Possible evaluation criteria

- Accountant's criteria
- Journalist's "
- Economist's "

My choice:

- Scientist's "
  - estimation of net effect
  - variation of hypotheses

Project's net effect

- Political and social
- Infrastructural
  - in Japan: generation of researchers
  - worldwide: Japanese apar (FGCS, IJCAI-97, etc)

- Scientific
  - uniform & efficient logical framework
  - demonstration of parallelism power
  - efficiency of producing reliable SW in logic
  - thousands of results in detail

Project hypotheses

- Logic as basis
- Vertically integral systems building
- Specialized logic machines
- KL1's non-logical features

Future

- Limited exploitation
- Elimination of non-logical

FGCS Project Evaluation Report

Keith L. Clark

Imperial College
London, UK
5th June 1992

This report is an expanded version of the short presentation that I gave at the FGCS Project Evaluation Workshop on June 3rd in Tokyo. It is based on knowledge of the project gained since my first visit to ICOT in 1983, on the ICOT reports and associated presentations of the first two days of the FGCS92 conference, and on the presentations and discussions of the evaluation workshop.

Interest and involvement with the FGCS project

Let me say at the outset that I was very pleased and honoured to be invited to take part in the evaluation workshop. As one of the originators of the concept of committed choice concurrent logic programming I have had a vested interest in the project, which in the latter stages became a huge experimental investigation into the utility of this variant of logic programing. Your adoption of this special form of logic programming has been criticized. It is said to be too low level, too far away from the ideal of high level declarative programming. The retort I have always given is that at the time you made the decision to fix on such a language as your kernel language for parallel machines, no other proposed type of logic programming language, which one could hope to implement efficiently on a multiprocessor machine, enabled one to program the concurrent communicating processes needed in an operating system for a parallel machine. Quite correctly, you had the programming of the operating system, in the LP language of the machine, as a major goal. In addition, and as you have shown, I have argued that more declarative logic program languages can be supported on top of such a language. (I am now inclined to agree with the view expressed at the evaluation workshop by David Warren that the recently proposed Andorra extension of Prolog, or the Pandora extension of Parlog, are higher level languages which today would be a better starting point for a PIM kernel language than FGHC. But that is with hindsight.)

Interestingly, three people (Giles Kahn, Alan Robinson and Tony Hoare), who had an influence on the conception of committed choice concurrent logic programming, are attending FGCS92. Giles Kahn, in his 1977 IFIP paper with Dave McQueen on a stream communication model of parallel programming, set me and Frank McCabe on the track of exploring coroutining and pseudo parallelism, with incremental communication via shared variables, in IC-Prolog. It was our inability to see how we might efficiently implement IC-Prolog on a multi-processor that was the motivation to find an alternative approach. Then, on a semester visit in 1980 to Syracuse University, at the invitation of Alan Robinson, Steve Gregory and I decided to look at Hoare's CSP for inspiration. The concept of guards and committed choice non-determinism of that language (an idea previously proposed by Dijkstra for his guarded command language) seemed to us just the concept needed to allow efficient implementation of and-parallelism with stream communication in a logic programming language. This lead to the so called Relational Language, which merged the committed choice, communication only on committment, concept of CSP with the equally elegant stream communication model of Kahn and McQueen.

Because of our early work on concurrent LP languages, Steve Gregory and I were invited to ICOT in 1983. (I was pleased to hear in his Monday conference presentation, that Koichi

Furukawa had read with interest our 1981 paper on the Relational Language and, even before the start of the FGCS project, had considered using a concurrent logic language rather than an or-parallel Prolog as the PIM kernel language.) Our 1983 visit coincided with the second ICOT visit of Ehud Shapiro, the originator of Concurrent Prolog, which was based on but significantly extended the Relational Language. I believe that between the three of us, we helped convince Koichi Furukawa and his colleagues that adopting a concurrent LP language as the PIM kernel language was a sound approach. During that visit Steve Gregory and I crystalized our views on the essential features of Parlog, our successor to the Relational Language.

Since 1983 I have briefly visited ICOT twice, in 1985 and 1990, and had papers in both the 1984 and 1988 FGCS conferences. Colleagues Ian Foster and Jim Crammond, working on programming environments and implementations for Parlog, have both been invited to ICOT. Over the years there has been much exchange of views between ICOT and the far smaller Parlog Group at Imperial. The meta call of Parlog, introduced into Parlog by Steve Gregory and I on our 1983 visit to ICOT, is very similar to the shoen of KL1. Both are used to support the programming of operating system functions. Hence my vested interest in the project, and my earnest wish that it be perceived to be the great achievement that I believe it is. If some of my following remarks appear to be critical, they are intended as constructive criticism. They represent what I consider needs to be done to convince a skeptical world that there are significant results and achievements in the FGCS project of which the world *had better take note*.

Impact of the FGCS project

Let me begin by saying some positive things about the impact which the project has had outside Japan.

Firstly, it made Japan pacemakers in logic programming research and a country whose research into LP and its AI applications had to be taken seriously by the international AI research community. In addition, by the spin offs and interest in computer science research that it has generated in Japan, it has also made the country a force in CS research. You have also, through rotating industry researchers through the hot house of ICOT, trained a new generation of computer scientists and engineers into techniques of advanced research. I and others have observed with pleasure the maturing of the young scientists that were nurtured by ICOT. They are now well able to hold their own in the international research community and to explain their ideas effectively and clearly. Many have remarked to me at this conference on the quality of the presentations, especially those from ICOT researchers. ICOT staff and associated researches have not only had an impact in the fields of LP language design, programming methodology and implementation, they have made significant contributions in all areas of logic programming.

Outside Japan the FGCS project stimulated a great deal of research activity by both universities and industry, and it unlocked significant government funds to support this research. The UK Alvey and EC ESPRIT programs almost certainly would not have started, or would have been funded at much lower levels, were it not for the FGCS project. Nor would the industry supported MCC, ECRC and SICS research institutes have been formed. For this stimulus to CS research, thank you. I personally owe my chair at Imperial, certainly the fact that I got it in 1987, to this increased activity and respect for LP research that followed the announcement of the project.

The FGCS project had a significant effect on the amount of research activity and perceived importance of both AI and LP research. The IJCAI 1986 conference in LA and the ICLP 1986 conference in London have not had higher attendance or greater interest from industry. This interest was a direct result of the excitement and interest that the FGCS project aroused.

Some comments on the "Achievements of the FGCS Project" report

I shall now offer what I hope is some constructive criticism via some comments on this short (two page) report that you gave me to look at as part of the evaluation.

In this report you make several claims which I believe to be true but as yet unproven in the eyes of the world outside Japan, a world that is perhaps uncharitably looking for any excuse to claim that the project was a failure.

"Thus, .. (KL1) makes it possible to quickly develop application programs which make full use of parallel machines with hundreds to thousands of processors."

To convince the rest of the skeptical world you have to properly document KL1 and the KL1 programming methodologies that you have developed. You have an impressive range of applications on display at the exhibition and described in the conference, but the program level anatomy of the applications is not adequately described. Programming a large application in Prolog is not easy for beginners, programming in any parallel language is worse. The LISP or C++ hacker building AI applications will believe that programming in a concurrent LP language must be near to impossible. Of course they are wrong, but convince these AI application developers that you do have an easy to use language and good application development support tools. Describe them much more fully, in clear English. Show step by step how to develop a model application. I know that writing such documentation is an onerous ask, for which researchers have no appetite. But it needs to be done, perhaps by writers skilled in the art of technical documentation who have been shown how to use the software, and who are helped in the task by its developers and expert users. Your experiments in quickly building highly parallel applications need to be repeatable, by people who have not helped develop the technology or been subcontracted by ICOT to do it, if the truth of the above claim is to have the impact that it should.

Doing such documentation is also necessary if the excellent policy of making the software freely available is to have any effect.

Under the time and resource constraints that you had, I do not believe that you could have done such documentation before now. Indeed, many of the tools and methodologies would only recently have been developed. But please seriously consider doing such documentation in the final nine months of the project, or in the first year of a follow up project.

"(Pim) is now providing the most powerful symbol processing capability in the world"

Again, to be really convincing on this claim you should compare the implementations of KL1 and PIMOS on your Pims with an implementation on a standard multiprocessor, ideally one that uses a RISC processor. Many are skeptical about the need for special purpose processors and language dedicated machines. The LISP machines failed because LISP was as fast, or nearly as fast, implemented via a good compiler on a general purpose machine. The PSI machines surely do not have a market because the latest Prolog compilers, compiling down to RISC instructions and using abstract interpretation to help optimize the code, deliver comparable performance. Such compilers run on $5000 workstations that offer all the other UNIX tools on which many have become to depend. Might not clever implementation on standard multiprocessors offer acceptable performance for parallel applications developed in KL1 and its extensions. If that is the case, the major result and impact of the FGCS project will be its software, and its radically different approach to developing parallel applications.

I believe that this will indeed be the lasting legacy of the project, rather than the features of the PIM machines that have been built. (Did you really need to build five? Or was their construction relatively inexpensive compared to the cost of building PIMOS, KL1 and its extensions, and the applications?) However, computer architecture is not my field. A comparative evaluation against stock hardware might prove the need for the specialized architectural support of KL1. I understand that you intend to do such a comparison as well as a thorough evaluation of the alternative Pims. A good result would be the identification of a few low cost features that significantly boost the performance of KL1 and PIMOS on a multiprocessor machine, as was mentioned by Takashi Chikayama in the workshop. Such features could then be incorporated onto a general purpose commercial machine which offered both UNIX and PIMOS. (Could PIMOS run as a subsystem of UNIX?) UNIX would ensure initial penetration of the market and the ICOT software should then ensure a runaway success for some ICOT licensed Japanese manufacturer.

Conclusions

The FGCS project is something of which Japan can be truly proud. It has had more impact than any other research project in computer science. It was magnificent and bold in conception, and has delivered much more than I expected it could achieve.

PIMOS, the Pim machines, KL1 and its extensions and the impressive range of initial applications are significant achievements that are testimony to the skill, dedication and single mindedness with which the goals of the project were pursued. I suspect that in most other countries such a project would have ended long before the 10 year deadline, either through withdrawal of government support or lack of stamina of the principal investigators. ICOT and its associated researchers have also done excellent research in other areas of LP, as evidenced by the many publications and the high quality ICOT research report series.

The decision to freely distribute ICOT software is excellent, but this distribution needs to be supported by good documentation of both the software and the methodologies of its use.

You should definitely port KL1 and PIMOS to existing commercial multiprocessor machines. In your achievements report you say "*...the technology of PIMOS as well as the KL1 language is .. applicable to most MIMD .. machines..* ". I agree. By proving this you will increase the impact of the project. It is also necessary if the freely distributed software is to be widely used for developing applications for parallel machines.

You still need to convince a skeptical outside world that KL1 provides "*..much higher productivity and parallel program maintainability than any conventional language*". **Document, perhaps also refine, your application development methodologies. Describe the program level structure of your applications.**

Develop more applications. Develop what Ehud Shapiro in his workshop presentation called a 'killer' application and which I referred to as a 'demonstrator'. At the outset of the FGCS project there was much talk of knowledge information processing as the key application area of FG computers. Why not build a huge information processing application to support management decision making? Such an application must be multi-user, perhaps using KL1 and PIMOS implemented on a distributed loosely coupled system. There is great potential for your technology to support distributed AI applications and CSCW (Computer Supported Cooperative Work).

Also **look at applications of your technology for numerical applications.** At a workshop in Syracuse in 1990, Geoffrey Fox complained that existing languages for parallel numerical applications only support homogeneous parallelism. Perhaps this is another application area for a suitably extended KL1, heterogenous numerical applications.

In some form or other the FGCS project must continue, or the achievements will have *far less* impact than they should.

The new project, run perhaps by a smaller ICOT, should support, maintain and continue to develop KL1, PIMOS and the application support tools. Its role should be to help others to use this software, by producing excellent documentation and assisting outside groups (in Japan and elsewhere) to develop applications. There is less need, now, for ICOT to develop complete applications, except perhaps the 'killer' application. **In addition, ICOT should be adequately funded to continue fundamental research into LP and its use on parallel machines.**

Finally, thank you for an exciting 10 years of excellent research into concurrent LP and its use. May your good work continue.

Keith Clark
Imperial College, UK

Interest & involvement in FGCS project

1. One of originators of concept of committed choice concurrent logic programming in 1980/81. (Debt to Professors Hoare & Robinson)

2. Periodic visits and early involvement

1983	1 month with Steve Gregory Same time as 2nd visit by Ehud Shapiro
1985/90	Brief visits
1984	Paper on OS in Parlog in FGCS84
1988	Invited lecture at FGCS88
1992	Invited to this workshop

3. Project is partly huge experiment investigating utility of concurrent LP

4. Hence, strong personal interest

Impact of FGCS Project

1. Stimulated much research activity & unlocked considerable government funds for AI & CS research in rest of the world
   - Alvey, Esprit, MCC, ECRC, SICS etc
2. Personal benefit – my chair + Alvey/Esprit projects
3. Significantly raised profile of AI and LP
   - 1985 IJCAI
   - 1986 ICLP conference
   - AI as IKBS becomes respectable in UK
4. Made Japan pacemakers in AI and LP (maybe even CS) research in world

But

Yesterday Herald Tribune can claim project has been a failure

— A view I do not share

— Oh..?

Some comments on "Achievements" report

"..... Thus (KL1) makes it possible to quickly develop application programs which make full use of parallel machines with hundreds to thousands of processors.... "

I believe this to be true.

But can the rest of the (skeptical) world?

1. Where is the methodology of KL1 programming?
2. What exactly is KL1 over and above well described FGHC
3. You have impressive range of applications but where is convincing demonstration that they were easier to develop using FGCS project software
4. Need anatomy of applications, not just performance statistics

"(Pim) is now providing the most powerful symbol processing capability in the world ... "

However ...

1. What is the gain as compared with an implementation of KL1 + Pimos on more conventional existing multiprocessors?
2. Compare PSI approach to good compilation technology for Prolog on RISC machines.
3. Is dedicated hardware necessary?
   - expensive for customers
   - lags behind latest techology
   - Why did LISP machines fail?
4. Perhaps it was a mistake to put so much emphasis on specialised machines
   - were 5 Pim's needed to validate approach?
5. Instead, more could have been spent on applications, application support software and investigation of application development methodologies

Concluding remarks

1. FGCS project is something Japan can be proud of.

2. Pimos, PIMs + impressive range of parallel and complex applications are significant achievements.

3. Free distribution of software a good idea.

4. But, need to port software to existing more widely available multiprocessor computers "..technology of Pimos as well as the KL1 language processors is ..applicable to most MIMD .. machines"

5. Need to convince a skeptical world (and HT) that there is "much higher productivity and parallel program maintainability than any conventional language" Do this by developing easy to use application development methodologies

6. Need also to develop more applications in new areas. Perhaps a huge knowledge information processing application for management decision making

Report on the collaborative project between ICOT and the NIH

Richard J. Feldmann
Division of Computer Research and Technology
National Institutes of Health
Bethesda, Maryland 20892

The collaborative project between ICOT and the NIH must be considered at several levels. At the highest level the project is meant to provide a vehicle for developing friendship and understanding between Japanese and American scientific workers. By means of visits to each others laboratories, by almost daily fax and e-mail messages we have begun to understand each other's ways of thinking.

Two specific scientific projects were used as the scientific substrate for the project: Genetic Information Processing and Protein Folding.

The Genetic Information Processing work is very much influenced by the ICOT style of logic programming. This work is being done also in collaboration with workers at the Argonne National Laboratory (ANL) and the Lawrence Berkeley National Laboratory (LBNL). Over the last year and a half, four workshops have been held. The emergence of the InterNet means that workers can come together in one physical place to meet and talk but still use the computers and databases in their own laboratories. A very graphical interface and database program, called GenoGraphics, has developed from these workshops. GenoGraphics which is the work of Ross Overbeek and Ray Hagstrom from ANL, started with the data representation of George Michaels (NIH) on the *E. coli* genome and the work of Kaoru Yoshida (ICOT and LBNL) and Cassandra Smith (LBNL) on human chromosome 21. The *E. coli* data was collected by Ken Rudd who is now a member of the National Center for Biomedical Communication and Information (NCBI) in the National Library of Medicine (NLM) of the NIH. Dr. Michaels has just recently held such a workshop at the NIH. Workers from all over the USA and from England came together to increase the range of genomes which can be handled by GenoGraphics. During this workshop the genome for yeast pombe collected by the workers at the Imperial Cancer Research Fund (ICRF) laboratory in London, England was introduced into the GenoGraphics logic programming data format. The ICRF workers had spent almost half a year developing programs and organizing their data. During the week-long workshop they were successful in transferring their data to logic programming format.

Dr. Michaels expects that in the next year there will be fragments of about 50 genomes for various small organisms entered into the GenoGraphics format. GenoGraphics as a result of Dr. Hagstrom's work is now written in C and runs on any PC compatible machine. We expect that GenoGraphics will become a world standard tool for the representation, manipulation and investigation of genomes.

The collaboratory model which George Michaels has developed is a powerful outcome of our interaction with ICOT and the other US national laboratories. Scientific workers can now come together from all over the world and using the InterNet can work together effectively for a short period of time.

In the Protein Folding portion of our collaboration with ICOT we have built and analyzed several models for the representation of protein structure. The collection and analysis of x-ray crystallographic data sets was begun in our laboratory almost 20 years ago. The relationships between protein structure, function and folding pathway have been very difficult to elucidate. The protein folding problem is the key technology which will enable biological system design. During the collaborative project workers in both countries engaged in the design and construction of both physical and computer models. Physical models provide simple, visual, trans-cultural vehicles for communication. Computer models can be constructed to represent salient features of the physical models. Using both logic programming and conventional machines we have investigated the statistics and dynamics of these models. The resolution of a protein model and its water environment is a critical determinant of the computer power required to simulate folding. Parallel computational techniques for simulating protein folding using logic programming machines have been developed by our ICOT collaborators, Makoto Hirosawa, Masato Ishikawa and Masaki Hoshida. Hirosawa-san spent his whole winter vacation programming and running the folding algorithm. At the NIH, David Rawn (Towson State University) and I have made progress towards finding a topological principle which unites the water seeking (hydrophilic) and water avoiding (hydrophobic) aspects of protein structure. A complete and simple topological model would reduce the N^2 portion calculations to the number of amino acids in a given protein. With such a model we would hope to be able to fold proteins on many different types of computers.

Discussion with the ICOT workers has also focused on computer languages, style of operating environment and network connectivity. Using the PSI II and III machines loaned to the NIH under the auspices of this collaboration, it has been possible to evaluate the state of development of the hardware and software produced by the Fifth Generation Project. Any user who decided to accept a research machine must know that it will be a lot of work. The FGCS conference shows that at the end of the project, much more of the potential of the hardware and software is now usable. During the continuation year we at the NIH would expect to make much greater use of the capabilities of the PIM machines at ICOT. Discussions during the conference brought out the problems which other collaborators have been experiencing in the early utilization of the ICOT hardware and software. The decision by MITI announced at the conference is a clear indication that the Japanese viewpoints of the utility of international collaboration are rapidly changing. The PIMOS operating system should be ported to world-standard machines so that scientists all over the world can begin to do program development in KL1.

During this collaborative project we have come to the opinion that even more than any single or parallel computer, the network is the most powerful artifact created by man. In this trans-global project we have experience the transition from paper letters to fax letters to fully electronic messages to interactive use of remote computers. In the beginning of the project the InterNet between Bethesda and Tokyo was rather slow and unreliable. Machines at either end had trouble talking to each other. As the project proceeded, the ability to communicate both electronically and intellectually rose to very high levels. The network gives us the ability to reason with each other about problems of mutual concern. The reasoning which we can do by sending messages is, however, rather limited. The InterNet is coming to the point where scientists can use databases and run processes on computers all over the world. New classes of tool for utilizing the network are being developed in many places in the world. We would hope to use these network tools to more strongly couple our collaborative research efforts.

We thank the administrators of ICOT for making such a strong and exciting collaboration possible. We expect to continue our collaborative work long after the formal end of the project.

The collaborative project between ICOT and the NIH

Richard J. Feldmann

Division of Computer Research
and Technology
National Institutes of Health
Bethesda, Maryland 20892

June 3, 1992

Our project started in year 9

October 1988	First contact
October 1989	First visit to NIH
July 1990	First visit to ICOT
October 1990	PSI II delivered
October 1991	PSI III delivered

Considered at several levels

Developing friendship and understanding between Japanese and American scientific workers.

Scientific projects

General conclusions

Two scientific projects

Genetic Information Processing

Protein Folding

Genetic Information Processing

Included other national laboratories

- Argonne National Laboratory (ANL)
- Lawrence Berkeley National Laboratory (LBNL)
- National Library of Medicine (NLM)
- Imperial Cancer Research Fund (ICRF)

Produced a new class of genetic information manipulation tool

- GenoGraphics
  - Prototyped in Prolog on Sun
  - Distributed in C on PC
- Whole genomes of Bacterium, Yeast and Human

Protein Folding

Built and analyzed several models for the representation of protein structure.

Used both logic programming and conventional machines

Made progress toward understanding protein folding

General Comments

Hardware is irrelevant

Software is everything

- Mass action of programmers
- Graphical User Interface
- International Interface Languages
  - English and C
- Balance between C and Logic Programming
- Data structures independent of processing language
- Still need highest level tools to help thinking

Communications defines the 1990s

- The network is the most powerful artifact built by man

Collaboration will continue

Report on FGCS Project

H. Gallaire

Surely, it is difficult for me to report on a project with which I have had many points in common. ECRC, the laboratory of Bull, ICL, and Siemens which I had the role to direct in its formation years had a lot to share with ICOT, in spirit if not in practice. I hope that my comments will not be distorted by what I would like others to think about ECRC's own role, achievements and problems.

First I want to command the ICOT management for getting external viewpoints, for opening wide their doors, and for letting so much of their work be discussed.

1 SUMMARY

The scientific success of the FGCS project is not to be disputed. There has been innovative work, there has been deep and thorough work yielding a better understanding of issues which would otherwise not have progressed as much.

The technical achievements of ICOT are impressive. Given the novelty of the approaches, the lack of background, the difficulties to be solved, the amount of work done which has delivered something of interest is purely amazing; this is true in hardware as well as in software. The number of alternatives explored must have generated a large body of knowledge which, if it can be shared will prove to be a major return for all of us. The steps taken to provide free software for all is definitely a positive one, even if it is currently hampered by the absence of hardware available to run it or by lack of specifications available to all, as appears to be the case for e.g. KL1. Whether many people will take advantage of this offer remains to be seen because it appears to be difficult to take only bits and pieces of it.

The scientific vision of the ICOT management and of their sponsors has been maintained all over these years, and it must have been a very challenging task; they must be complimented for holding firm to their technical vision. The enthusiasm behind this vision has had positive effect on the whole community across the world, leading others to take actions, if not to follow the approach that ICOT was pursuing.

The fulfillment of the vision, should I say working on the 'grand plan' and bringing benefits to the Society, is definitely not at the level that some people anticipated when the project was launched. This is not, to me, a surprise at all, i.e. I have never believed that very significant parts of this grand plan could be successfully tackled.

I was expecting however to see "actual use" of some of the technology at the end of the project. There are three ways in which this could have happened. The first way would have been to have real world applications, in user terms: only little of that can be seen at this stage, even though the efforts to develop demonstrators are not be be underestimated. The second would have been to the benefit of computer systems themselves (eg impacting the computer manufacturers); this does not appear to be directly happening, at least not now and this is disappointing if only because the Japanese manufacturers have been involved in the FGCS project, at least as providers of human resources and as subcontractors; whether this lack is due to the fact that not enough effort went into getting their true support (which may have been a tough issue after TRON and SIGMA) or not, is difficult to assess from a distance, and the responsibility for that state of affairs certainly lie in several hands. However, I firmly believe that when Japanese industry starts looking for engineers and designers for parallel systems (which may already be the case), they will draw heavily on the skills developed at ICOT in the FGCS project; indeed what has been learned through the development of many versions of machines, of parallel OS, of cache management and of load distribution algorithms, of distribution networks, etc will undoubtedly be useful to them. The third way would have been to impact computer science outside of the direct field in which this research takes place: for example to impact AI, to impact software engineering, etc; not a lot can yet be seen, but there are promising signs, eg the results on AI in legal reasoning or theorem proving; by the way there are again direct ways and indirect ways through which the project impacts these fields: by making sheer use of the powerful hardware technology and making practical what was known but was impractical on conventional hardware (the parallel theorem prover is one such example), or by true innovation using the new tools of the project ; there are certainly several examples of the latter (eg CAL, QUIXOTE, ...); it seems to me however that there has been more reliance on the use of the power of parallelism; this is probably natural since developing parallel systems was and still is the major technical thrust of the project; one can only wonder what are the limits of this position, as we were reminded by A. Robinson quoting M. Minsky during his invited talk at FGCS'92. More application work would have been needed to feel fully optimistic about the impact on the environment of this work.

The project made a choice of one approach of symbolic computing, namely logic programming (LP); it pursued it very consistently; this is a very wise behavior, and I did the same at ECRC. ICOT went very far, building many different pieces of hardware (convincing us if needed of the exceptional manner in which technology is mastered in Japan), building full operating systems with great success, investigating many solutions in parallel. If one wants to establish a new center, I would recommend to follow the same pattern, namely stick to one type of technology, especially when it is a new technology and when so little is known about it. This allows one to see a problem under different yet related views and helps progressing to solve it. For example the work on parallel implementation of KL1 is useful in its own right but it also provides insight for other problems as well, e.g. for KBMS implementation.

I believe logic is the right choice that had to be made to investigate knowledge based systems; I have argued elsewhere that it does not mean that there is only logic in the practical world but there is no contradiction here. Perhaps this perspective should have guided ICOT more towards integration of logic to other environments than it did.

I will not criticise the choice of KL1 against the choice of logic languages a la Prolog; it is important to understand the limits of each approach; if the parallel implementations of Prolog a la Andorra work well (on truly large scale problems), fine; otherwise we know we have solutions a la KL1; it is early in the game to know for sure.

In general I am surprised positively by the speed at which the researchers have picked up the background that they encountered elsewhere during their research to make novel proposals; constraint logic programming is one such example where the progress made is significant, even though they were not the first players.

When it comes to discuss specific results, it is difficult to single out one of them, because the areas covered are so different. I only would like to mention again the fact that all the work done on parallel systems implementations will definitely bear fruit in a non distant future, directly or indirectly. I feel that the work on knowledge bases is not as foolproof as some of the other work done in the project and that the QUIXOTE environment, although its is appealing when one considers all the features it integrates, would need more testbench work before it can be adopted because I find it complex and lacking some of the features that such complex semantic representation systems need (see below). The work on parallel theorem provers embodies some nice results and has shown that it goes beyond state of the art; however I have some reservation due to the fact that parallelism cannot be the answer to all difficult problems. The work on constraint languages is very interesting and one of the very few to allow to use non linear constraints; this work shows the high level of skill with which the developments have taken place; there is also room for improvement here because again speed is probably not the only answer. I will not comment much further individual results except to say that case based reasoning may appear to be easier to do now than before (until we run into other speed bareers ...)

2 FUTURE WORK

There is a list of actions which could be mentioned here; I will only stress some of my main points.

There is a need to evaluate how standard technology will support the FGCS results, knowing that the standard technology itself is not stable, in terms of performance, in terms of features (distributed OS may appear soon for example). This is in my mind crucial to the future of the results of this project.

I believe that there is a need for ICOT to show the impact of the technology on classical problems. Just to give an example, why not try to develop a payroll package, where knowledge bases and rules could play an essential role for building easy to customise packages, where parallelism is clearly possible and interesting at the processing and database level, where constraints can be useful for a human resource package (eg for allocating people to tasks, ...) and where it is possible to enhance basic packages by AI: for example finding the best person qualifying for a task, based on a description of skills etc; this can be as sophisticated as one wants, but only as the icing on the cake. I am sure there other potential applications where such combinations are possible. What would be important would be to be able to compare development time, maintenance time, adaptation time, performance and costs of running systems, etc. In general I feel a need for more comparative work, taking into account costs which I admit may be difficult to do since the new hardware and software cost can hardly be compared to commercial one.

The results on parallel implemetations are impressive; however there is the need to work on automating the mapping between programs (in KL1) and processors; if this is not done, it may jeopardise building the higher level languages and applications which need to run efficiently in most cases and to exploit well the architectures.

There is a need to work on important issues related to the knowledge bases work, at least to simplify it and to address for example the problem of integrity constraints (which is not, as I understand it, what has been done under the constraints heading in QUIXOTE). I also believe that the work on constraints need more research and incorporation of more propagation-like techniques. I have no feeling about what's needed for the work on the Genome project.

There are definitely enough results obtained and enough good and important problems waiting for an answer, that there is no doubt that a follow up of the project is needed.

Hervé Gallaire - 2 June 92

Hervé Gallaire
GSI
25 Boulevard de l'Amiral Bruix
75782 Paris Cedex 16

FGCS Assessment

Gilles Kahn
INRIA Sophia Antipolis, FRANCE

Introduction

Assessing the FGCS project is a hard task, not only because there is little time and space to do so, but also because the scope of this project is very wide. I doubt if many people can read equally competently the work that has been performed in so many diverse areas: computer architecture, programming language design, database design, natural language analysis and generation, genome research. I for one have very little competence on natural language work, although I hear from a variety of sources that this may indeed be one of the strongest points of your work in the last phase of the project.

The FGCS project is very broad, but everyone can see its unity of purpose. When listening to all the presentations and reading some papers, I wonder whether there has been enough time for true integration of the many components that have been developed in the seven laboratories. I suggest the following problem, to see what I have in mind. You have developed a theorem prover, MGTP. On the other hand, you have sophisticated tools to generate sentences and paragraphs in Japanese. Assume that you would like to connect these two components, so that the theorem prover produces proofs in natural language, that a Japanese mathematician would like. In particular, these proofs should not be too verbose, concentrate on the real difficulties and be socially acceptable. Is this a problem that can be solved in a matter of weeks or months with the software that you have developed, or do we need to start a new project?

I have listened carefully to all the talks that reported the work of FGCS project, and I must say that they were all very high quality presentations. The laboratory chiefs show considerable experience, maturity in their scientific fields. Answers to questions are very direct and frank and do not try to cover up difficulties when there are some. As well, the demonstrations have shown strengths and weaknesses of the software. I appreciate the considerable amount of work needed in preparing such a thorough presentation of the FGCS achievements. This has confirmed an attitude that I have witnessed in the wonderful INRIA–ICOT meetings that I have attended: a completely frank exchange of views with scientists of high caliber, who are concentrating on basic research, and building software prototypes to demonstrate the validity of their fundamental ideas.

The Central Issue

As I see it now, the FGCS project has been attacking a fundamental problem in Computing, namely concurrency; and it has chosen a line of attack, Logic Programming. This is a priori how good projects get started: with a difficult and deep problem on the one hand, with an original idea on how to solve it on the other hand. Indeed, the problem of concurrency in computing, which has been with us practically since the invention of computers, has become ever more essential in the past 10 years. Let us review for a moment the positive and negative aspects of your approach.

On the plus side, we can see many advantages: first the attack is extremely original, almost far fetched for some. The Japanese effort appears immediately as a leader, because noone else is betting on this direction of work on the same scale, even though a number of very bright individual researchers around the world have had successes. Second, your approach is a software approach, i.e. you are concerned a priori with the intellectual control of parallel hardware, with putting into hardware mechanisms that will make it easier to program. Let us go more quickly through the other

advantages: the problem seems tractable, opens many questions, leads to basic research; it focusses on fine grain parallelism, which is a priori harder, on irregular problems that occur in symbolic computing –of course– , but even in numerical computing (finite elements), in geometry or in discrete events simulation.

On the negative side, the major problem is that the approach is

very difficult:

how can one have an efficient computing system that combines parallelism with non-determinism, which is implied by a declarative approach? Parallelism means that you parcel out work to remote computing units, non-determinism means that you may discover at any time that this work is useless and should be canceled immediately. In fact, the problem is so difficult that one of these two aspects runs the risk of being short-changed. Clearly, GHC and KL1 have shown their bias in favor of parallelism. The second problem with your approach is that people have misunderstood it: they considered that you worked on Logic Programming, using parallelism to compensate for its intrinsic inefficiencies. Another difficulty, linked to the originality of your approach, is that when you started, there were very few applications based on the logic programming paradigms, so that you ended up having to mount your own effort to build applications.

Let me add two remarks regarding the difficulty of your project. First I have stated in my talk that the Logic Programming community, as a scientific group, was "weird". As there were many proeminent members of that community in the audience, I received a large number of inquiries about what that statement really meant. First, Logic Programming, in 1981, was virtually unknown in the United States. The group of scientists who had been fighting for it was necessarily a bit paranoid about that fact. Second, there was, and unfortunately there still is, in this community –usually not among the top

leaders– a slightly sectarian attitude: they have seen the Truth revealed, nothing else deserves paying attention to.

The next remark has to do with something that has unfolded during

the project in the commercial world: artificial intelligence, as a business, has not exploded as expected. Progressively – to the dismay of the many gold seekers in Silicon Valley, but not to the surprise of true scientists – it has turned out that the limiting factor in the development of AI is not hardware, not even software, it is the capacity of human beings to model satisfactorily a larger and larger number of problems that were not previously solved on the computer. So AI, and expert systems have developed and matured, but not at the rythm of electronic circuit technology. As a result, many companies have dropped out of the field entirely and a company like Thinking Machines Corporation has fundamentally redirected its marketing efforts away from AI. So it is certainly the case that some AI applications are compute bound –and you have worked on them–, but the pressure to solve these problems is not drastically different of that of solving a number of other scientific problems.

In view of the remarks above, my assessment is as follows. The FGCS project has accumulated considerable experience on MIMD computing, in terms of programming and architecture as well. This experience is probably unique. All methods that would be applicable only for a small number of processors have been rejected off hand. This is a very sound approach for basic research. In terms of software, you have designed and implemented bold and elegant ideas. I believe that many of these ideas will be useful, and used, outside the ideological context of FGCS. The basic research that you have performed has been published in the open literature, it is deep and durable, it has earned you the esteem of many scientists around the world. Globally, my opinion is very positive.

Now, I will turn to the aspect of your work that is closest to my personal area of research.

Language Issues

A priori, all of the work of FGCS revolves around one language, KL1. KL1 is an original construction. Aspects of KL1 are described in many papers. The implementation of KL1 must be fairly similar on all of your hardware platforms, otherwise you would have a difficulty porting PIMOS, and applications. It would also be difficult to train new users. Yet, I see no single report which is "the definition of KL1", that I could read at leisure to form a precise opinion of the language. I think one of you wrote that KL1 = FGHC + meta – control + convenient-things. That leaves a bit too much room for my imagination. In fact, I am not totally certain, given my previous understanding of ESP, that the logic programming aspect of KL1 is so important in comparison with the Object-Oriented Methodology of using KL1, that Chikayama-san alluded to in his presentation.

In any case, even if KL1 is very well designed, it is not the only language that you have designed. I have heard of A'UM, AYA, MENDELS ZONE, GDCC, cu-Prolog. The dream of having one single language to implement everything, no matter whether your are a systems or an application programmer is long gone. In fact, facing a multiplicity of languages is unavoidable. But luckily, we know now that this diversity is tractable, thanks to the advent of distributed computing. So KL1, like all programming languages, needs

a detailed evaluation of its features: what is used by systems programmers; what is important for programs that generate KL1 code; what should be in libraries rather than as a primitive of the language; what are the protocols that should be used to interface to other languages, because in the past ten years, we have all learnt that there are very few "purely AI" applications. In the process of this analysis, you may also reexamine where hardware language support was essential. This was impossible to assume in 1981, but now we know that microprocessors supporting 64 bits of address space are here.

Basic research must elucidate, by analysis and experimentation, what the basic mechanisms and the basic protocols are. I feel that part of this remains to be done for KL1, although the really costly part, building an implementation and accumulating experience in building operational software with it, has been done thoroughly in the project.

Technical questions

I would like to list here a number of technical questions that have come to my mind during your presentations. The fact that I ask such questions show that I take extremely seriously the work of the project, and that I feel that it is necessary to understand your design decisions, to appreciate them fully. First, I would like to understand with greater precision the innovations of PIMOS, in comparison to other distributed operating systems such as AMOEBA, GUIDE or CHORUS. I am convinced that PIMOS's ideas are very general and quite unconnected to Logic Programming, frankly. My second question concerns memory management. As a Prolog user, I rarely have problems with speed, but I keep fighting with the memory management schemes of the various Prolog systems. I wonder whether you have looked at the remarkable work of Bekkers and his colleagues at IRISA, in France. On a similar line, I understand that you were focussed primarily on parallelism, but some schemes for extending the applicability of logic programming have appeared in the last ten years. In particular, given your interest in Theorem Proving, I would have thought that Lambda-Prolog, an extension of Prolog that includes terms with binders, should be of interest for you.

Several groups have designed languages at ICOT. Were certain principles of language design systematically used, did you design or use general tools for this task? Certainly the technology that you have developed can be useful there.

To conclude this paragraph, I believe that having unity of purpose is extremely useful; it gives everyone a sense of a global objective, a way of measuring progress. But basic research has its own logic as well. If you pass near an important scientific problem and do not treat it because it is not squarely in the direction of your project, this will be unfortunate. Because we are in 1992, I cannot avoid the banale remark that Columbus did not look for America.

Conclusions

In basic research, ten years is NOT a very long time. As I mentioned earlier, research on parallel processing has been with us at least since the mid-fifties. Many many subjects of Computer Science have taken longer than that to mature. In the case of an extremely original project like FGCS, everyone could –and did– predict that if you were serious about basic research, then it would take more than ten years to reach your objectives. It may be necessary to arrange the pursuit of your goals differently, but it is in the nature of good basic research to create constantly new and unforeseen problems.

In terms of technical achievements, the FGCS has produced many technical papers, it has accumulated considerable competence on software for MIMD machines, on how to harness the power of a large number of processors. It has defined and followed what I believe are fundamentally good strategic

directions of research: software for parallel processing, theorem proving, Object-oriented Operating systems, computing problems connected with the law or the understanding of the human genome.

In terms of social achievements, the success is truely remarkable. The project has developed basic research in computing on an unprecedented scale in Japan, supporting a number of activities outside ICOT as well. The Journal that it has fostered, New Generation Computing is a good scientific journal. ICOT has developed friendly and fruitful international contacts with many countries and scientific institutions. This cannot be overemphasized. For many of us, the project was absolutely crucial in opening and maintaining contact with Japan. Finally, the project has maintained faith in Artificial Intelligence as a fundamental research topic with a scientific basis.

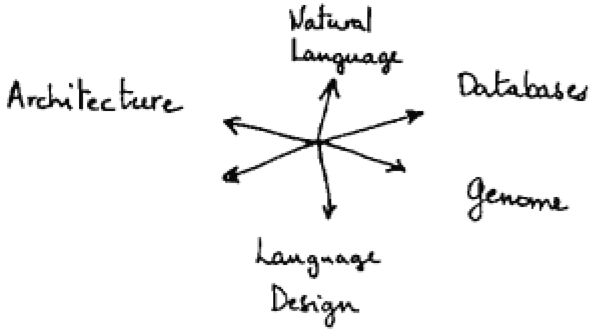
I cannot imagine, at this stage, that ICOT will stop brutally. I have understood that you plan to release software to the world, in a novel policy of your government. On the basis of the little experience I have in this area, I would say that few researchers in the world will want to use software if they know that no one is there to maintain, develop and improve it, as well as basing its own research on it. Software that has not changed for 18 months is considered dead. Of course, free software is not maintained like commercial software, but researchers must have the feeling that the authors of the software still care for it before they use it.

Tokyo, June 4 1992

G. KAHN

FGCS
assessment
—

Scope:



Broad, with unity of purpose

Question?: "integration" yes KL1, but..

Example: Build NL output for theorem provers

Reporting: **Very high quality talks**

Laboratory chiefs show experience, maturity, openness

Demos show strong & weak points

International Seminars: wonderful

Small objection: presentations too "politically correct"

A problem:

Concurrency

A line of attack

Logic Languages

+	−
original	difficult ***
Software view	misleading
tractable	applications
open	weird community
basic research	TMC
long range	world changes
fine grain //	
irregular	
TMC	

*** difficulty: N.D + //ism !

Assessment: MIMD, many processors } positive

Language issues

In principle :

KL1

? what is it really :

FGHC + x + y

\+ Methodology of OO programming

ONE language , but :

A'UM , AYA , MENDELS ZONE , GDCC , cu-Prolog.

Facing multiplicity of languages is <u>unavoidable</u>!
and <u>tractable</u>

Hardware language support:

Is it truely necessary (History!)

Research must elucidate : basic mechanisms, protocols

Technical questions :

PIMOS versus other distributed OS

say AMOEBA , GUIDE , etc..

Memory management
was the remarkable work of BEKKERS et al.
considered

L.P. Extensions
what about λ-Prolog ? Given accent
on Theorem Proving , should be taken seriously

Language Design
Principles , common tools ?

Z Single purpose is very useful as driving force but Basic Research has its own logic too!

(C. Colombus was not looking for America)

Conclusions

1. I disagree with Furukawa-san :

   10 years is NOT a very long time

2. Achievements of FGCS (technical)

   - many
   - Competence in software for MIMD
     large # of processors

   - good strategic directions

   software , theorem proving , OOOS , genome

3. Achievements of FGCS (social)

   - Basic Research , Journal
   - International Contacts ≈ Friends
   - Faith in AI

4. CANNOT STOP BRUTALLY

   But beware ICOT (Free Software)
   not (ICOT Free) Software

REPORT ON ANU/ICOT COOPERATION

Michael McRobbie
Executive Director
Centre for Information Science Research
The Australian National University

1 Introduction

Unlike many of the other participants in the Evaluation Workshop I do not feel competent to evaluate or comment in detail on the research activities of ICOT over the 10 years of the project though I will provide some general comments later in this report. ANU's position is different to that of many other organizations that have co-operated and collaborated with ICOT in that our interaction is almost solely in an application area - theorem proving - and not at the level of the fundamental research that has been until recently ICOT's main focus. Further our interaction only really commenced in 1992 though it has grow quickly into a valuable collaboration that has rapidly yielded significant results.

Thus I will restrict myself to describing this interaction and commenting on the quality of ICOT's work in theorem proving

2 History

ANU followed with great interest the MITI initiative that became ICOT from about 1980 and followed in broad outline the activities of ICOT ever since. ANU has had a strong logic research group since the early 1970s and this group started working in theorem proving about 10 years ago. The announcement in the early 1980s that Japan was to initiate a major well funded initiative in logic-based computer systems was an enormously exciting one to members of this group and it is difficult to convey now just how galvanizing an effect it had on researchers in foreign countries who had been working on computing and logic for years in relative obscurity. ICOT was a major stimulus not just to their morale but also to the establishment of large, well funded programs and laboratories in many foreign countries such as Alvey, Esprit, MCC and SICS.

These facts are well known and I will not dwell on them as others more knowledgeable than I will certainly do so in their reports. However the involvement of research institutions in my country, Australia, with ICOT, is less well known and is worth briefly putting on the record since this report is being written at the last FGCS Conference.

In the early 1980s there was an exceptionally strong research group in logic programming at the University of Melbourne led by John-Louis Lassez and containing John Lloyd, Rao, and others. As I understand it the Australian Minister for Science visited ICOT around 1983 or 84 and heard many favourable things about the research of the University of Melbourne group. He returned to Australia and made a decision to fund a "Machine Intelligence Project" based on the logic programming group at the University of Melbourne. The level of funding was good by the standards of the time, but unfortunately the huge amount of international interest that was generated by ICOT in logic programming and related areas caused talent in this area to be at an absolute premium. Hence it was not long before many members of this exceptional group had accepted much more attractive positions overseas.

I believe from informal discussions with a number of these people over the years, many of them would have quite happily stayed at the University of Melbourne and hence brought great distinction to the University of Melbourne and to Australia.[1] However a combination of an inflexible university system (since improved somewhat) and a lack of any sort of understanding of the emerging importance of the field of logic programming on the part of various government decision makers meant that a significant opportunity to establish a stable group of international stature in this area was unfortunately lost.

Researchers from a number of other Australian research organizations had various interactions over the years with ICOT but that with the University of Melbourne was the most substantial of them.

ANU took an initiative around 1987 to establish what is in effect the national centre for high performance computing and now has 4 supercomputers - 2 Japanese and 2 American. This initiative was numerically based but because of ANU's long-standing interest in logic and theorem proving as well as ICOT's announcement at the 1988 FGCS conference of the plan to build the PIM machines, ANU decided to explore with ICOT the possibility of establishing some sort of agreement on research cooperation.

Discussions commenced in 1989, initially with Dr Kurozumi and then also with Dr Uchida, in tandem with discussions to have ICOT participate in the 11th International Joint Conference on Artificial Intelligence which was to be held in Sydney in August, 1991. In 1991 a series of letters was exchanged between the Centre I head at ANU and ICOT to establish research cooperation in the field of theorem proving. Following ICOT's very successful participation in IJCAI-91, a party of about 30 ICOT scientists

[1] An example of what could have been is nicely seen in the fact that the 4th International Logic Programming Conference was held at the University of Melbourne in 1987. A major international event in logic and computing with a heavy emphasis on logic programming co-sponsored by the University of Melbourne was also held in Melbourne in early 1984.

travelled to ANU in September 1991 to install two Psi-2 workstations and to give a wide ranging series of lectures on ICOT's research. Dr Uchida has since indicated that the Psi-2 workstations which have been very difficult to maintain properly will be replaced by the more reliable, powerful and more standard Psi-3 workstations soon after FGCS-92 is finished.

3 ANU/ICOT Cooperation

For various reasons cooperation between ANU and ICOT only really began in early 1992. The cooperation has so far just been limited to theorem proving and has mainly involved Drs Slaney and Grundy at ANU and Drs Hasegawa, Fujita and Koshimura at ICOT. The cooperation has so far been in three different areas: (i) heterogeneous high performance theorem proving systems (ii) application of model generation techniques in theorem proving and (iii) system independent theorem prover performance analysis tools.

(i) Heterogeneous High Performance Theorem Proving Systems

In 1991 Dr John Slaney of ANU and Dr Ewing Lusk of Argonne National Laboratory (ANL) in the United States devised a new automated theorem proving system called SCOTT for Semantically Constrained Otter - Otter being a high performance theorem prover written by Dr Bill McCune at Argonne National Laboratory which is generally regarded as the fastest and best general purpose theorem prover presently in existence. On a wide variety of problems SCOTT is about twice as fast as Otter.

SCOTT builds into Otter the well known and venerable principal of semantic resolution which uses a model of the underlying theory with which the theorem prover is working to help guide the search for a proof. However semantic resolution is a static principal in that it uses a model, usually arrived at by somewhat arbitrary and non-systematic methods, and once incorporated into a theorem prover this model is not changed for the duration of the search for a proof.

The key new idea in SCOTT is that as in a standard resolution theorem prover it starts with a model of the underlying theory. However as more information is built up about this theory as the search for a proof progresses, this in turn is used to generate dynamically new and more precise models for the theory that both guide the search for a proof more effectively and which decrease the number of clauses that are discarded by expensive methods such a subsumption. The techniques used to dynamically generate such models systematically and efficiently given that the problem is exponential in its order of complexity in the number of values in the model, were developed over 10 years by Slaney and others mainly at ANU. Much of Slaney's work in this area is embodied in the program FINDER (available by anonymous ftp from arp.anu.edu.au).

SCOTT then in effect integrates together Otter, FINDER, a model testing module (in this case really just a module of FINDER) and a new performance monitoring tool called XSCGraph developed by Grundy which we discuss under (iii)below. All these programs are written in C.

Following the visit to ANU of the ICOT party in 1991, Slaney and Grundy noticed that ICOT' s grounded MGTP theorem prover and their non-ground version developed primarily by Drs Hasegawa and Fujita, could be combined together in the same way as Finder and Otter are combined respectively. What was necessary was to build the testing interface between them. This has been written in KL-1 by Grundy working at ICOT over the last 2 months.

The combined system called Semantically Constrained MGTP (SC/MGTP) is all written in KL 1 and runs on Psi-3 workstations. Work has begun to implement it on the 256 Processor Mitsubishi Electric Corporation PIM-m. SC/MGTP runs about 4 times faster than Otter running on a Sparcstation 1, and given that each processor in the PIM-m is effectively a Psi-3 processor, there is the potential for SC/MGTP to run hundreds of times faster than Otter. However I stress that this research has only just begun and that those working on it have not yet had time to analyse the comparative performance of the systems in any depth. Thus some of the figures just mentioned are to a certain extent conjectural.

(ii) Application of Model Generation Techniques in Theorem Proving

The topic here is rather technical but in brief it is a collaboration between ANU using the Finder system and ICOT using the ground-MGTP system to attack a series of problems posed by Bennett concerning certain finite algebraic structures in group theory. These problem have so far resisted solution by any automated theorem prover but a number of them have now been solved using Finder and ground-MGTP. This work was announced at FGCS-92 and will be reported on in detail in a joint paper by Hasegawa, Fujita and Slaney.

To paraphrase Lusk and Slaney from their CADE-92 tutorial on Finding Models, these result are more than a little encouraging and at the very least indicate that there is a rich vein of open mathematical problems concerning finite algebraic structures which can now be approached by means of automated reasoning techniques.

(iii) System Independent Theorem Prover Performance Analysis Tools

A regular criticism heard of some of ICOT's work is that without comparisons with other international research in the same fields it is difficult to form an appreciation of how good the work really is. As part of the collaboration to build SC/MGTP a system independent performance analysis tool XSCGraph has been written by Grundy specifically to allow the direct comparison of the performance of the C-based SCOTT system and the KL-1-based SC/MGTP system, as well as many other systems.

It is clear from (ii) that SC/MGTP compares very favourably with SCOTT and XSCGraph allows a more detailed comparison to be made on how they perform on the same problems.

4 Comments on ICOT Research on Theorem Proving

I hope that my discussion in the previous section indicates first that the collaboration between ANU and ICOT has grown in a short time to be a very fruitful one and one that parallels and complements in a close way the collaborative relationship between ANU and ANL.

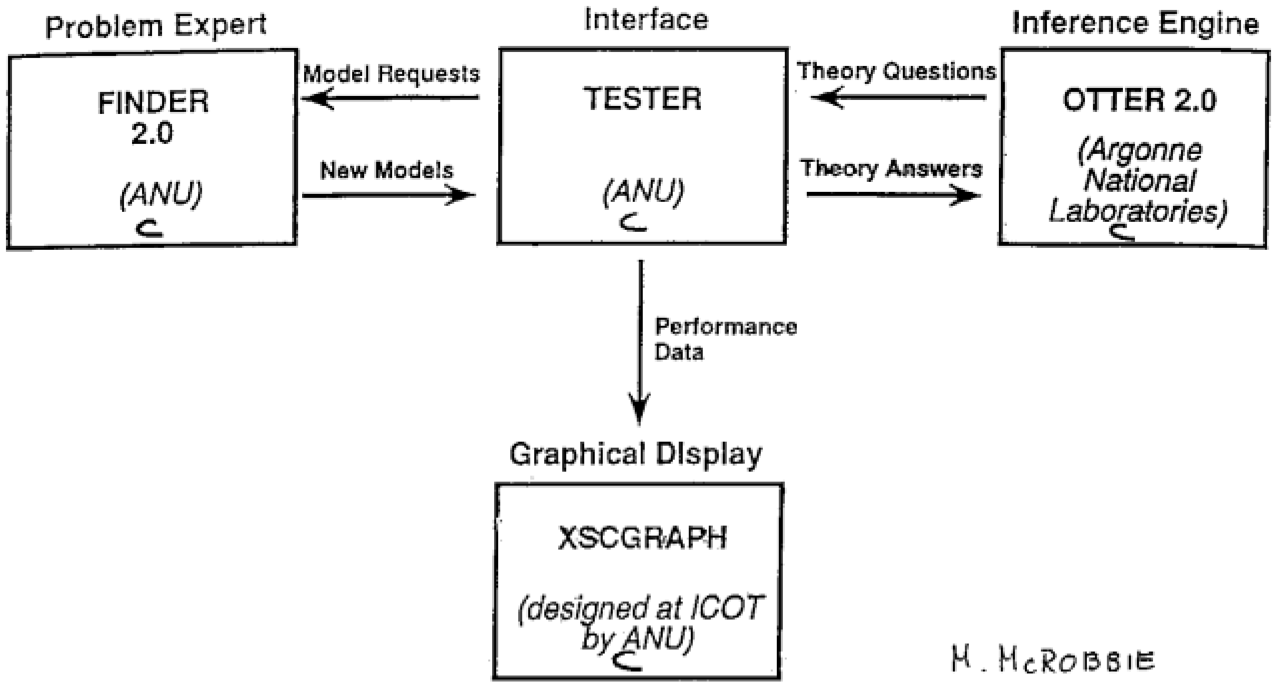
It should also be clear that the ICOT work on theorem proving stands favourable comparison with some of the world's best which can be seen from the fact that it has passed the hardest test of all - proving a significant open mathematical problem. The ANU researchers involved in the collaboration have also reported favourably on KL-1 as a prototyping language.

The ANU/ICOT achievements were cited prominently a number of times during FGCS-92 and all the research discussed in Section 3 was either demonstrated at the FGCS Demonstration or reported on at the Conference.

5 The Future

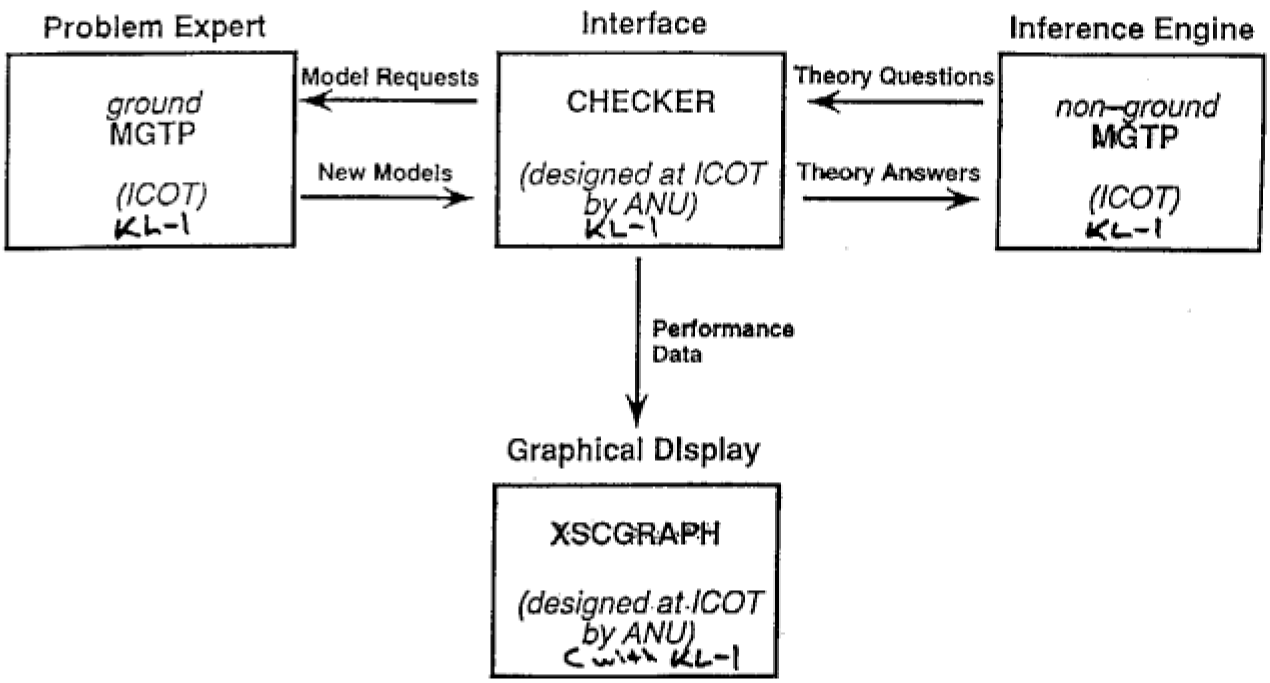
ANU has been very pleased with how its collaboration with ICOT has progressed in 1992. Because of this we will be employing another researcher to work on the joint research projects and to help maintain the Psi-3 workstations once they arrive. We certainly support the idea of the core activities in ICOT being extended beyond March, 1993. Making ICOT tools such as KL-1 available on "stock" Unix platforms would be a great advantage and ICOT should be encouraged to do so as soon as possible.

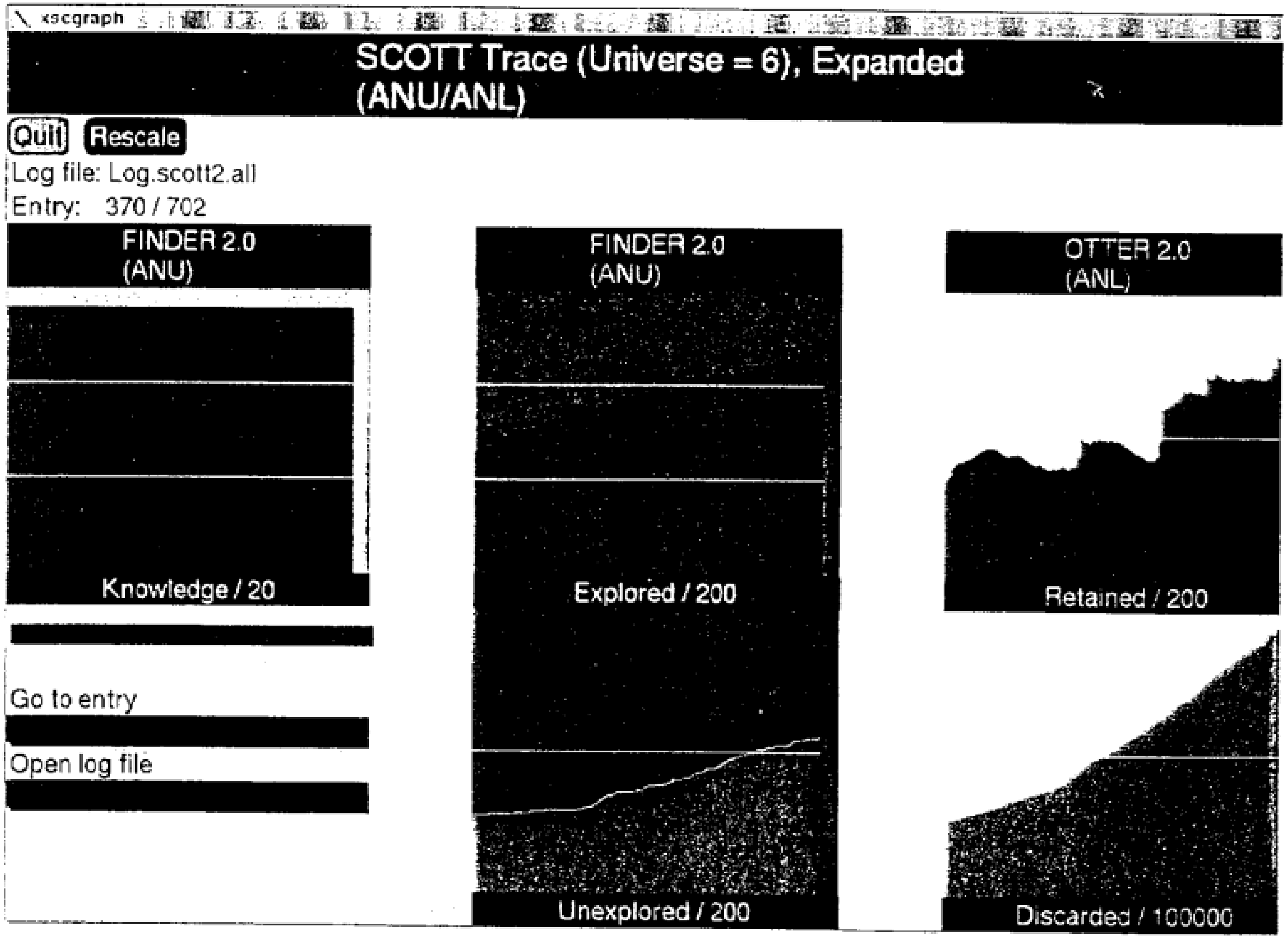
SCOTT Realization

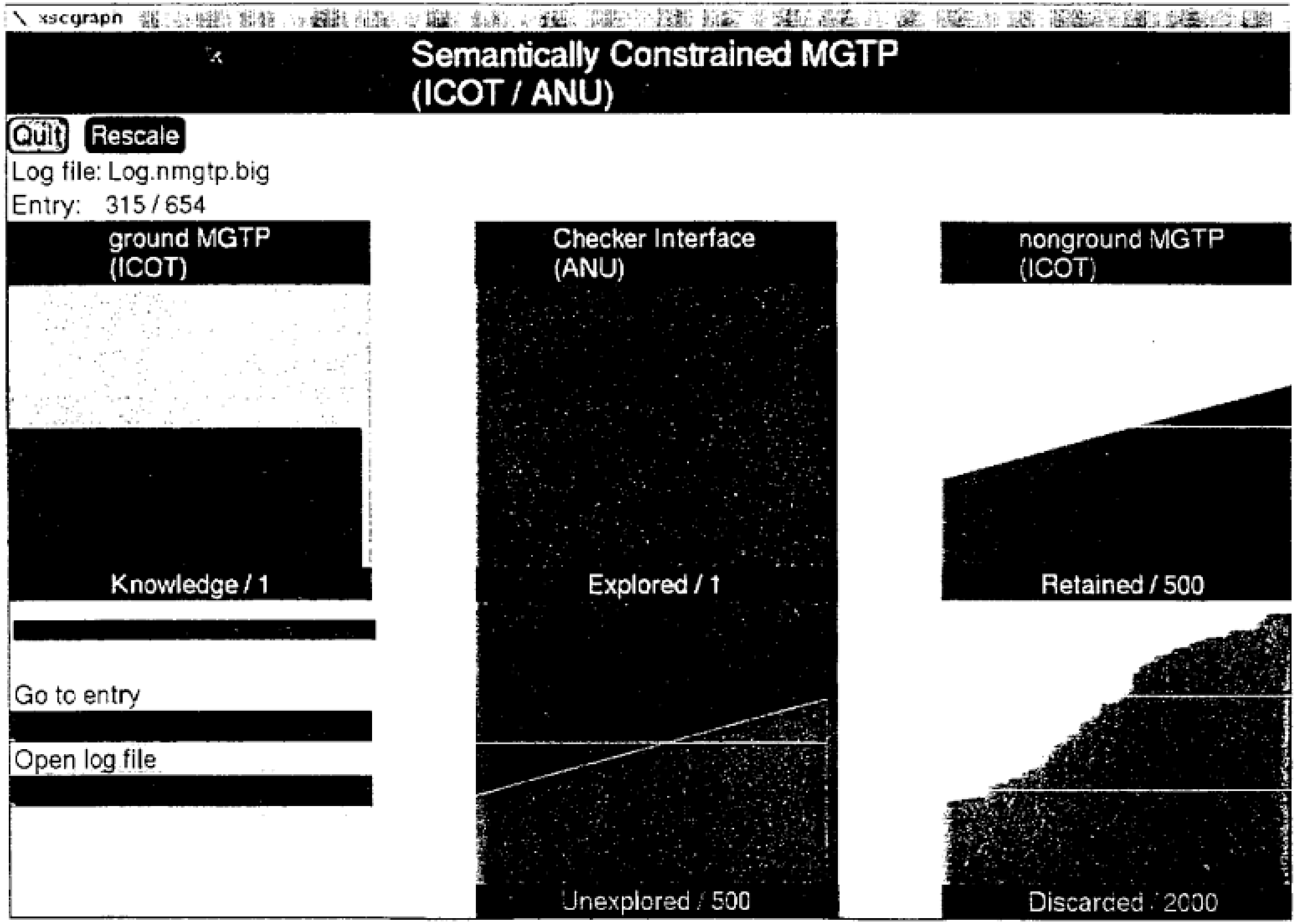


M. McROBBIE
ANU

SC/MGTP Realization

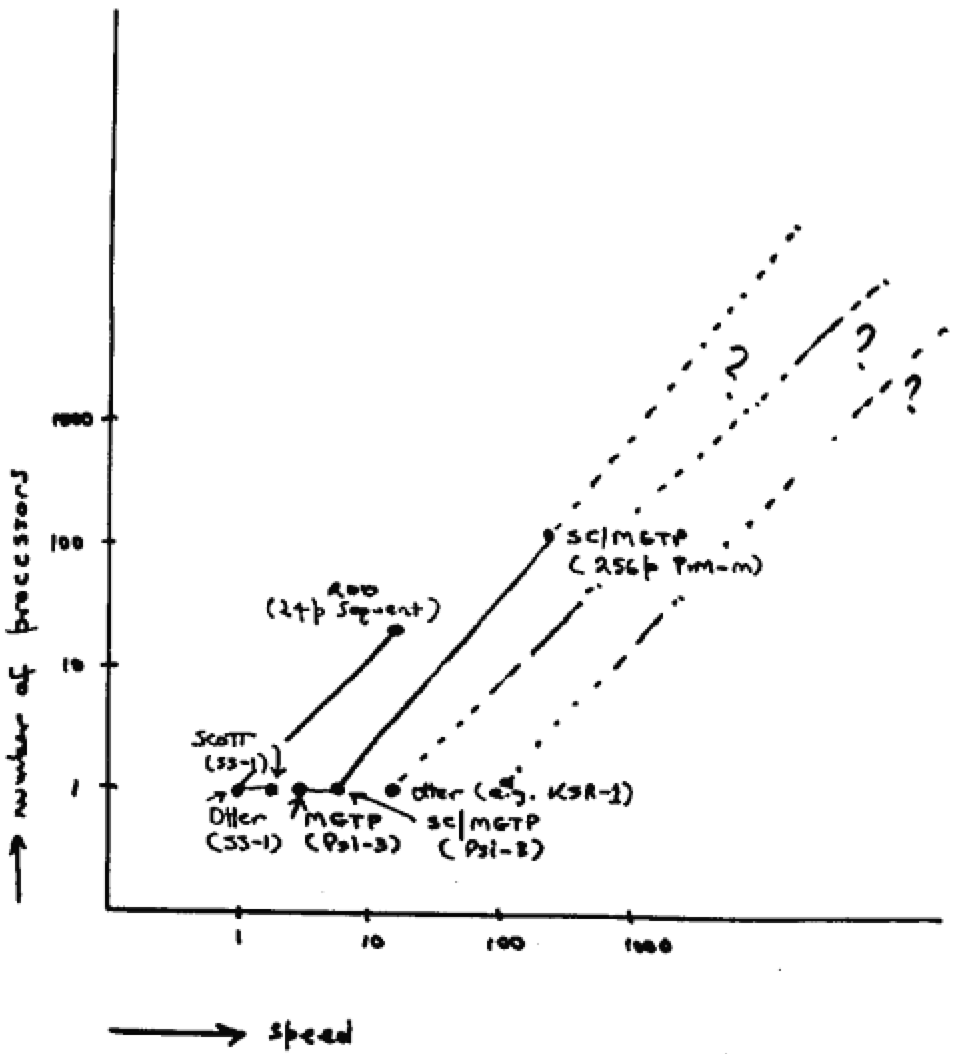



xscgraph
SCOTT Trace (Universe = 6), Expanded
(ANU/ANL)
Quit
Rescale
Log file: Log.scott2.all
Entry: 370 / 702
FINDER 2.0
(ANU)
Knowledge / 20
Go to entry
Open log file
FINDER 2.0
(ANU)
Explored / 200
Unexplored / 200
OTTER 2.0
(ANL)
Retained / 200
Discarded / 100000


xscgraph
Semantically Constrained MGTP
(ICOT / ANU)
Quit
Rescale
Log file: Log.nmgtp.big
Entry: 315 / 654
ground MGTP
(ICOT)
Knowledge / 1
Go to entry
Open log file
Checker Interface
(ANU)
Explored / 1
Unexplored / 500
nonground MGTP
(ICOT)
Retained / 500
Discarded / 2000

ANU/ICOT & ANU/ANL Collaboration in

Heterogeneous High Performance Theorem Proving Systems



ANU/ICOT Collaboration in Theorem Proving

- Heterogeneous High Performance Theorem Proving Systems (eg SC/MGTP)
- Application of Model generation Techniques in Theorem Proving (eg. Solutions using MGTP on Bennett's problems)
- Development of system–independent Prover Performance Analysis Tools (eg. XSCGraph)

Some Points

- ICOT software on emerging Japanese parallel machines eg. AP-1000 (Overbeek)

- MITI's policy on making ICOT software freely available
  - major change as I understand it
  - implications for RWC project

Fuchi : "best efforts in the context of Japanese society"
  - RWC Project should learn from ICOT

- Tradgedy if ICOT not extended in some form to:
  - continue collaborations already in place
  - implement ICOT software on "stock" machines & under "stock" OSs
  - fully explore the capabilities of the PIM machines

Report for the Evaluation of the FGCS Project

Alain Michard

INRIA
Domaine de Voluceau
78153 Le Chesnay
France

June 4-5, 1992

Introduction

I have attended the FGCS' 92 Conference and have observed most of the demonstrations which were presented. I also attended the FGCS evaluation workshop on June 3. It was my first direct contact with ICOT researchers and with the FGCS Project, although I was somewhat aware of the main research trends and achievements of the project, mainly through reports or published papers.

As my personal background is not directly in the domains of interest of FGCS, I shall focus my comments on the issues related to the exploitation of the results (e.g.: distribution of software, follow-up to the program) rather than on the scientific results themselves.

During its 10 years life-time, ICOT has succeeded into building a very original and efficient *integrated* system which can be used as a basis for intensive symbolic processing applications. Doing this, ICOT has obviously made some highly valuable scientific breakthroughs, recognized as such by the international scientific community. I feel that strong efforts should now be made to capitalize on these scientific and technological results.

Main achievements

As it has been pointed out by most participants to the evaluation workshop, the achievements of the projects are both technical and sociological.

In my opinion the most salient technical results are:

- a very high-performance parallel machine, designed for and around a concurrent logic-based language (KL1), and demonstrated with its complete Operating System, PIMOS.

- an elegant Guarded Horn Clause language, with an efficient implementation.

I must say I have not been so much impressed by the demonstrated applications: even if they are of very high quality, I feel that similar experiments are (or have been) developped elsewhere, on more traditional plateforms. Even if these applications show that the FGCS environment is truly operational and usable, they do not clearly demonstrate the *specific added-value* of the "PIM+PIMOS+KL1+KBMS+GDCC" integrated system. A specific effort should therefore be made to make visible the elegance of code, ease-of-programming, and resulting efficiency obtained through the use of the FGCS development plateform and environment.

The main so-called "sociological" achievements of FGCS Project are:

- ICOT is internationally recognized as a strong research point in Logic Programming, Concurrency and Parallel Architectures, and more generally, Japan is now considered as an important country in computer science. Ten to fifteen years ago, Japan was hardly recognized as a partner in the research community in Computer Science (although some individuals were already quite well-known). It is now obvious that Japan plays a major role in CS-research : ETL, ICOT, NTT Laboratories, Tokyo University, among some others, are well-known abroad. This is probably due, up to a large extent, to the fact that the FGCS Project drew attention of the scientific community to what was happening here.

- FGCS has given credibility to Logic Programming and to artificial intelligence. Before the announcement of FGCS, these domains and the corresponding technologies were considered as pure "research toys" by policy-makers and by the industry.

- FGCS had a major political impact in all western countries. I think for instance that the launching of the ESPRIT program was up to a large extent a European political answer to FGCS. In this respect, FGCS has strongly boosted I.T. R&D all over the world.

- Japan has now some highly-skilled researchers and engineers, who can be considered as top-level experts in the domains of machine architecture, parallel operating systems, logic programming and knowledge-based systems.

Maximizing the benefits of the Project

FGCS has been a major basic research effort. It is the author's view that some decisions have to be taken and some actions have to be launched in order to maximize the benefits of the Project, both for Japan and for the whole scientific community.

The decision to make all the resulting software freely available to third-parties is, of course, a very positive point. It is a clear sign showing the willingness of the Japanese government to support full participation of its research laboratories to the international effort for the progress of knowledge. But in my view, this decision is not sufficient and some additional decisions should be taken to obtain full benefit from the FGCS effort.

This decision will take its full effect only if a large community of users have the possibility to develop applications with the FGCS environment and tools. Existence of a large community of "advanced-users", developing original applications, exchanging information on their experiments, and reporting to ICOT (or its successor) any encountered problem (bug, functional limitation, etc.) will be the most powerful way to convince the I.T. community that FGCS has been a true success.

This will happen only if MITI decides to maintain a strong research team active in the domains of logic programming, parallel architectures and concurrency, theorem proving, etc. The role of this team will be to keep the FGCS environment (KL1+PIMOS+KBMS+-

MGTP+GDCC) alive. To convince foreign organizations to build upon this environment and its associated tools, clear guarantee should be offered that this software will evolve, become more portable, will be ported on various hardware platforms, and will progressively be enriched and modified to stay at the upper level of the state-of-the-art.

The team in charge of these tasks might be located within ICOT or elsewhere (ETL?). This is a purely domestic issue. But it should be clear for all external partners that this team shall exist for at least several years (ten years for instance). The number of external users of FGCS software will be a very clear sign of the success of this team.

I think that the first effort of this team should be to port the "FGCS environment" (KL1, PIMOS,..) on a parallel architecture based on standard off-the-shelf RISC processors. This porting should be used as an opportunity to try to design a portable version of KL1, isolating the machine-dependent part of the compiler. The method for porting KL1 on other machines should be documented in English, to make as easy as possible for other organizations (manufacturers, research organizations) to port KL1. *A formal specification of the semantics of KL1 should also be delivered.* All the efforts should be done to make the porting of KL1 to other machines by external organizations, not only feasible, but even easy.

A second action of this team should be to document in English the higher-level components of the FGCS environment: KBMS, MGTP, GDCC, ... A programmer's guide to KL1 should also be prepared and published. All this documentation should be permanently updated as the software evolves.

If MITI clearly demonstrates its commitment to support the distribution, porting, *and evolution* of the FGCS environment, this software has a reasonable chance to be, in five to ten years from now, a success-story similar to the Unix story of the eighties. Such a thing would definitely make of Tokyo one of the major place of the I.T. R&D for the next decades.

Furthermore, as the basic software could be ported on any machine by any foreign manufacturer, and that applications could be developed by any public or private organization, such a success would not create any political misunderstanding or industrial conflict.

Short Resume of the author

After obtaining master degrees in Data Processing and in Experimental Psychology at Paris University (1972-1973) Alain Michard entered IRIA (former name of INRIA) in 1973, as researcher. His first research activities were in the domain of cognitive science (modeling of reasoning) and of human-factors of user-interfaces.

His PhD Thesis in 1976 was entitled "Analysis and Formal Modeling of Diagnosis Tasks".

Nominated Research Director in 1980, he launched a Project in 1981, dedicated to the study of design methods for user-interface and interactive systems. The author was head of this Project from 1981 to 1990. Main research topics were in the domains of :

- design of on-line context-dependent help systems, with natural-language interface;
- design methodology for graphical user-interface;
- user-interface management systems, and dialog managers.

In 1990, the author joined INRIA's headquarters where he's now in charge of the International Scientific Relations Office for Western Countries. He is also the INRIA correspondent for the European Community R&D Programs.

A Report on My Visit to FGCS'92

Ross Overbeek

Summary:

I attended FGCS'92 at the gracious invitation of representatives of ICOT. I was invited to participate in a panel on the topic of computing in the 21st century, and then later I was invited to also participate in a workshop to evaluate the FGCS project.

Since I have for a number of years admired the vision and goals of the FGCS project, I felt honored by the invitation. In this report, I will try to honestly and constructively report on my view of the achievements of the project, along with some comments on the future direction of the project.

Achievements:

The project's achievements can be grouped into three categories:

1. Those relating to the general advancement of logic programming.

2. Those relating to the specific advances in software relating to an environment to support parallel processing.

3. Specific applications built upon the systems.

General Advancement of Logic Programming:

By committing such a sizable project to a technology based on logic programming, the Japanese immediately had a serious impact on the computer science research community. Their commitment to this technology elevated interest and a general recognition of its role in knowledge representation, parallel processing, and database technology. This produced sizable research projects in the USA, Europe, Korea and elsewhere to explore the technology.

My contacts with the Japanese have been far more limited than with Europeans, so my impressions may be inaccurate. However, I believe that the attempt to coordinate efforts on the project with foreign researchers has benefited both the Japanese effort and the other projects worldwide. It has, in my case, started a number of interactions that will continue well past the end of the project.

I have found it a difficult process to establish meaningful research relationships with Japanese logic programmers, and I am sure that they have found it even more difficult (the asymmetry of the relationship, with Japanese being willing to learn English, to understand our culture, etc., while people like myself take only a limited amount of time to reciprocate, must surely cause some problems). Yet, I cannot help but believe that the contacts initiated by the FGCS project will gradually produce a lasting benefit.

Specific Advances in the Area of Parallel Processing:

The development of PIMOS, the five PIMS, and the associated applications environment is a major achievement. In the case of parallel processing, I believe that it is likely (but far from certain) that the ideas pioneered by ICOT will play a central role in the eventual software environments to support applications development. In Europe and America, these ideas have emerged in the Strand and PCN efforts.

The development of PIMOS could lead to a commercial success, if it is successfully moved to general-purpose MIMD machines, and if it widely adopted.

In an incident that I deeply regret, I was quoted out of context by a reporter, and the quote was used in an article critical of the ICOT effort. What I regret is speaking to the reporter; he chose to completely misrepresent the basic intent of my remarks that evening. During the day, MITI had announced that it would make the software developed at ICOT freely available. I have consistently taken the position that ICOT should make the software available, that it must be ported to general-purpose machines before it is could possibly be of any commercial significance, and that it would benefit the Japanese to get the software in use. It should be released free of charge as the first step in attempting to build an effective solution to the problem of applications development for parallel processors; the commercial payout of such software will never be achieved, if it is not first widely adopted. I pointed out the sequence of events leading to the gradual adoption of UNIX to illustrate what I believe would be the correct strategy for allowing the software to attain its potential value. The tone and content of my remarks, as they were reported, were both rude and inappropriate. I do not believe that I ever spoke them (although I do believe that I expressed the opinion represented by the quote). In any event it was very foolish to place myself in a position where I might cause such misunderstandings, even unintentionally.

I do believe that the software and pool of experience represented by PIMOS is a substantial asset developed by the project.

Advances in Application Areas:

In general, I believe that too little emphasis was placed on building the best versions of applications on the machines (as opposed to demonstration versions).

However, I have been quite impressed with several specific areas. First, in the theorem-proving work (which I did not expect to result in substantial advances), the team did develop a system that proved an open theorem and another system that could prove a set of difficult theorems that clearly established the group as very serious indeed. For a group of relatively inexperienced young researchers to have attained this level of achievement in such a short time is quite remarkable. They have done more in a short time than many larger international groups have achieved over much longer periods.

In addition, I noted that the effort to integrate data from a variety of biological databases could offer a foundation for a serious advance in science (but has not yet progressed to that point). When I reflect on what has been done in a fairly short time, and upon the difficulties in communication between computer scientists and biologists, I cannot help but believe that a great opportunity still exists in this area. The Japanese are in a position to play a significant role in what will certainly become one of the most significant areas of scientific research during the coming decades.

General Reflections:

I have always viewed the Fifth Generation Project as fundamentally heroic. I am astounded that five distinct PIMs were actually produced, that a complete operating system with associated tools was completed, and that a serious attempt was made to apply this technology. I vividly remember being in an audience when an American researcher lectured an audience that included members of ICOT on the topic of software engineering, basing his comments on experiences writing programs of 2000 lines or less. A number of us found this truly strange, given the enormous effort that went into the large systems developed at ICOT, yet unfortunately typical.

Report
by Ehud Shapiro,
June, 1992.

I was invited to attend FGCS'92 in order to evaluate its progress and present a report in the "Project Evaluation Workshop". Hence I will focus in my report on the workshop.

The workshop seemed like a well-orchestrated psychological warfare against the two MITI officials attending, with side-effects for the representatives of the "private sector", i.e. lab chiefs from the participating companies. The troops used in this warfare were the "Logic Programming Mafia", who stood up, one after another, praised the Fifth Generation Project, and urged the MITI officials to extend the lifespan of ICOT.

Even though the different presentations were not coordinated (as far as I know), they showed remarkable uniformity of opinion. So perhaps the opinions expressed were sensible after all. All speakers praised the project for its achievements, noting especially (F)GHC and PIMOS as the notable achievements. GHC was praised for being an innovative and elegant concurrent logic programming language. PIMOS was praised for being a complete operating system built from the ground up using FGHC/KL1.

Most of the speakers also claimed that the hardware developed by the project was of a lesser long-term significance. Some went as far as saying that investing so much in hardware development was a mistake. All seem to agree that the fact that the software developed by ICOT was available only on proprietary machines diminished its impact.

As for the future, everyone stressed that without a continued presence of ICOT, the research results produced by the project would vanish into thin air. One of the main functions ICOT should play is to make the software technology available on stock hardware, under stock operating systems (Unix), provide documentation and support for the software, and integrate the various changes and improvements the users of the software, who will have access to its source code, are bound to make.

Every workshop participant got a collection of reports by ICOT visitors. Ogawa-san showed me my 1982 report, which appears there. I must admit I read it with great interest, not remembering at all what I wrote there 10 years ago. Reading the report caused me to change my presentation. I started by quoting the goals of the project, as I saw them then: "[The Fifth Generation Project's] ultimate goal is to develop integrated systems — both hardware and software — suitable for the major computer applications of the next decade, identified by the Japanese as 'Knowledge Information Processing'."

"In addition ... the project is expected to elevate Japan's prestige in the world. It will refute accusations that Japan is only exploiting knowledge imported from abroad, without contributing any of its own to benefit the rest of the world. Hence the project aims at original research, and plans to make its results available to the international community."

In retrospect, I think that both of these goals — realizing an integrated, innovate, and useful system, and elevating Japan's in the world — have been fully achieved. MITI's recent decision to allow free distribution of software is consistent with the plan of making the results available to the international community. However, I agree with the other speakers in the workshop, that without continuing support from ICOT, that software will simply die.

I agreed with some of the previous speakers, that the international impact of the project was not as large as one hoped for in the beginning. I think all of us who believed in the direction taken by the project, i.e. developing integrated parallel computer systems based on logic programming, hoped that by the end of the 10 years period the superiority of the logic programming approach will be demonstrated beyond doubt, and that commercial applications of this technology will be well on their way. Unfortunately, this has not been the case. Although ICOT has reached its technological goals, the applications it has developed were sufficient to demonstrate the practicality of the approach, but not its conclusive superiority.

This can be partly attributed to the short period available for application development, give the software and hardware development schedule. But, more importantly, I think that this was the result of the applications being developed in an artificial setup. I believe applications should be developed by people who need them, and in the context where they are needed. The suitability of the software technology developed by ICOT cannot be fully evaluated until such applications are attempted.

Therefore I made some concrete suggestions for the future direction of ICOT. Some of them are in line with what is already planned. I suggest that ICOT now focus on making its concurrent logic programming software technology widely available, and actively encourage and support research and development groups who will use it in real-life applications. Specifically, I suggest that ICOT will:

- Port KL1/Pimos to stock hardware (both Unix workstations and commercially available parallel computers, including the recently emerging small-scale symmetric multiprocessors. I believe even a PC version of a mini-system can be quite useful for teaching and exploratory purposes.)

- Initiate some standardization effort for concurrent logic programming languages and systems.

- Provide specification and documentation for the software.

- Make the software widely available, with a GNU-like distribution policy.

- Provide teaching material, tutorials, and consulting services for users who wish to use the software.

- Provide research grants for research groups who are interested in applying and/or improving ICOT's concurrent logic programming system.

Since the last point has raised considerable interest, I would like to elaborate on it.

I think that applications are best developed by people who care about them. There are research groups around the world who are interested in developing applications using parallel computers. ICOT can more fully achieve its goal of international contribution and impact by supporting such groups and encouraging them to use ICOT's concurrent logic programming system. Fostering an active user's community for ICOT's system will have many positive impacts. First, it will require ICOT to clean-up, fully specify, and document its system. User's feedback will suggest improvements, and the body of knowledge accumulated by users of the system will help the development of future applications, and suggest ways to evaluate and improve the system.

I suggest that research groups involved in such activity will be offered grants for periods of 2-3 years each, for the sum of about ¥10M a year. Grants should be offered to academic and research institutions on the basis of the scientific merit of the research proposal. The fact that such grants are available can be easily publicized to the international research community through electronic bulletin boards and other means, inviting grant applications, which include a research proposal and a budget proposal. The applications can be ranked using external referees, and the best ones will be selected by a committee. The research agreement could either state the results of the research be put in the public domain, or that they belong, independently, to the funding agency (ICOT), and the institute carrying out the project. Under such an agreement ICOT can, if it so chooses, put the results in the public domain.

The return on investment for say, 50 such grants, meaning ¥500M a year, will be much higher than any investment in an application development team assembled inside ICOT for such a task.

I ended my presentation with another quotation from my 1982 report: "The eventual success of the project will follow not from the amount of money invested in it, nor from the number of people working on it, nor even from the individual excellence of these people. It will follow from the coherent vision of its leaders, the genuine enthusiasm that they generate, and from the promising path of research that they chose."

I believe the statement to be as true today as it was 10 years ago. I thank ICOT and its leaders for these most exciting and rewarding 10 years.

CACM 1982

8. Conclusion

People who believe in the unpredictability of scientific progress and revolutions find
a planned revolutionary project to be almost a contradiction in terms. But sometimes [1]
ideology has to give way to reality: the Japanese project is both well planned and
revolutionary. It did not invent the concepts of logic programming, but it is certainly
the first, and perhaps today the only one, which grasped the immense potential of this
approach, and gathered the critical mass of resources necessary to utilize it on a large [2]
scale.

There are thoughts and attempts throughout the world at responding to the Fifth
Generation project, but I suspect that this battle is already won. The eventual success [3]
of the project will follow not from the amount of money invested in it, nor from the
number of people working on it, nor even from the individual excellence of these people.
It will follow from the coherent vision of its leaders, the genuine enthusiasm that they
generate, and from the promising path of research they chose. [4]

Any response to the project may match the amount of money or other resources
invested in it, but will fail to to come up with the same sense of direction and devotion
that holds the Fifth Generation project together. One such example is the British
response, which basically says: Let's keep doing what we do today, but with more [5]
money. Money will increase the progress of research, but by itself will not result in a
new generation of computers.

The Fifth Generation project faces two dangers: one is that it will succeed too late; [6]
the other is that it will succeed too early. If several years pass before any applicable

E. SHAPIRO

CACM 1982

Last April Japan's Ministry of International Trade and Administration (MITI), in cooperation with eight leading computer companies, launched a research project to develop computer systems for the 1990's. The project, called the Fifth Generation Computers Project, will span 10 years. Its ultimate goal is to develop integrated systems — both hardware and software — suitable for the major computer application for the next decade, identified by the Japanese as "Knowledge Information Processing". Even though it may ultimately have applicable results, the current focus of the project is basic research, rather than the development of commercial products.

In addition to bringing Japan into a leading position in the computer industry, the project is expected to elevate Japan's prestige in the world. It will refute accusations that Japan is only exploiting knowledge imported from abroad, without contributing any of its own to benefit the rest of the world. Hence the project aims at original research, and plans to make its results available to the international research community.

I was the first non-Japanese researcher invited for a working visit to ICOT, the Institute for New Generation Computer Technology, which conducts the project. Due to the nature of the project I was given explicit permission, even encouragement, to report on everything I saw and heard during my visit; hence this report.

ICOT is located on the 21st floor of an office building in central Tokyo. It currently hosts around 40 researchers, most of them on a 3-year "loan" from their industry-based research laboratories at Fujitsu, Hitachi, NEC, Matsushita, Mitsubishi, Toshiba, Oki, and Sharp.

The institute is divided into three research labs, responsible for research in hardware, basic software and applications software. The leaders of the laboratories are Drs. Munakami of Nippon T...

- The FGCS project is certainly a scientific & technological success!
- Yet, it hasn't ~~reached~~ achieved the desired level of international impact.
- Why? What can be done about it?

Scientific success:

- Contributions to programming language & system design, analysis, & implementation.
- Contributions in various application areas (NLP, KB).
- Stimulated basic research around the world, in LP & related / competitive areas.

Technological success:

- Very impressive, complete, (HW+SW) usable parallel computer system, with very clean design
- A result of good leadership with coherent & consistent philosophy.
- A large set of applications.

Additional positive side-effects.

- Greatly improved contact with international computer-science community.
- Exposed industry R&D staff to international standards & competition
- Provided stimulation & focus for Japanese Computer-Science community.

Why disappointing international impact?

- Radical software technology
- Needs "killer applications" to convince the mainstream
- ICOT applications only now emerging
- Available only on special ICOT hardware

⇒ Mainstream C.S. community is not convinced of this direction, yet...

Even though no viable alternative.

What can be done?

Keep the momentum,

Encourage larger circles to use the technology.

How?

- Make software publicly available
- Port to stock hardware (V?)
- Provide maintenance + upgrades (?)
- Provide support for international research groups that apply and/or improve the software technology.

(e.g. 50 × 10M ¥ /yr.)
= 500M ¥

Some Thoughts on the FGCS Project*

Rick Stevens
Argonne National Laboratory

I am pleased to attend the 1992 FGCS international conference. In particular, I am happy to have had the chance to visit with many members of ICOT and discuss the evaluation of the FGCS project with various industrial representatives. I thank Iwata-san for making local arrangements and Uchida-san for his continued interest in Argonne and its research programs. Nitta-san was very helpful in explaining the demonstration programs, and I am pleased to have had his assistance. Yamazaki-san and Ishihara-san, both Japanese industry representatives on the ICOT technical board and ICOT staff, have been open about discussing the future of ICOT and the role of basic research in Japan.

My involvement with the FGCS project began in 1986 with activities and participation in the Gigalips project, which was organized by Argonne and inspired by the FGCS project. In 1988, I participated in the joint ANL/ICOT NSF workshop in AI held at Argonne. This workshop gave me the opportunity to begin to understand the hardware and software aspects of the FGCS project. Later, as part of an ANL and ICOT joint project, I visited ICOT several times and was involved in installing PSI-II workstations and network connections at Argonne and in developing programs in KL1.

Too many people were involved in my various visits to mention them all. However, I mention especially Ichiyoshi-san and Susaki-san for their friendship and hospitality, and Furuichi san and Minami-san for offering to let me visit their homes. All of my interactions with ICOT staff have been highly positive. I have enjoyed my interactions immensely and wish in some fashion to continue these personal relationships.

In the remainder of this report I focus my comments on the topics raised in my evaluation presentation.

Evaluation of the FGCS Project.

First I want to make the point that the Western view of computer research and development processes is possibly quite different from that in Japan and this difference in view is largely responsible for the difficulty in assessing the significance of the FGCS project accomplishments. I believe that there is confusion about whether the FGCS project was a basic research project or an advanced development project.

I believe that the evaluation of the FGCS project will take considerable time and effort, and I also fear that the international community will not fully understand the impact of the FGCS on Japan and even on the world computer science community. I firmly believe that Japan has become a significant force in the computer science community and that the important point is that this position was achieved during a short-term project and for modest cost. Japan should not waste this opportunity to remain actively and productively engaged in a core area of basic computer science research. FGCS has, to a large extent, decided the directions of the logic programming community and heavily influenced parallel processing projects around the world.

Many in the United States are confused about how to evaluate the FGCS because the Japanese R&D process is not well understood. However, the average person in the United States does not fully understand the R&D process in the United States either! What is important is that the process of becoming open–the distribution of software and the evaluation of progress–be continued.

It is also important to remember that in basic research a negative result is not a failure but that the process of uncovering truth is pursued despite setbacks from time to time.

Many people, I think, desired to evaluate the FGCS project as an advanced development project, where an inability to get to product development is considered a failure. What many do not understand is exactly what the goals really were. Did Japan really want to develop prototypes for products? Was there a hope that industrial companies would adopt the technology and revolutionize the computer industry?

The most difficult point for the outside community to consider is what specific problems have been solved and what technological breakthroughs have occurred. The lack of clearly showing these things has caused many to discount the accomplishments.

The United States was evaluating the FGCS as both a basic research project and as an advanced development project. As a basic research project and as an advanced development project, therefore, it could have been considered a success if one or more hard problems in AI or CS had been solved or if a commercial company had committed to producing products based on the results of development. Have these things happened but not been revealed?

When the FGCS project was first announced, it created a storm of controversy in the United States and Europe. I think that both countries feared the project for two main reasons.

1. It fundamentally challenged their notions of preeminence in basic research.

2. If commercial products resulted from the project, Japan would have taken a

lead in knowledge-based systems–an important new paradigm–with little Western response possible in the short term.

This shakeup caused many government-sponsored projects to be created in the West (MCC, ECRC, Alvey, SICS), and even now we see the United States federal High Performance Computing and Communications project to have been influenced by the FGCS project. In this initiative government and industrial firms are teaming to develop systems and software.

Lessons Learned.

What lessons have I learned from the FGCS project?

1. Be aware that government-supported industrial consortia may not be able to "read the market," particularly over the long term. This limitation probably means that joint government-industry projects should be short term.

2. Do not confuse basic research and advanced development (i.e., know what you are doing, and don't confuse the evaluation criteria for the two). It is important that funding agencies and the community know what type of project one is working on and how that project will be evaluated.

3. Expect negative results but hope for positive. Mid-course corrections are a good thing. Assessing the direction and expecting that research may change direction are key to keeping projects relevant to the goals and to changes in the "real world."

4. Ensure that the basic research infrastructure has stability, a strong sense of the important core problems, flexibility, and an evaluation mechanism that can distinguish between negative results and incompetence.

5. Have vision. The vision is critical: people need a big dream to make it worthwhile to get up in the morning. The most important role of a project leader is to focus energy and attention on maintaining the vision and direction of large projects. The vision has the power to unify a group and motivate them to work through hard problems. "Make no small plans, for small plans have no power to stir men's souls."

Impact and Accomplishments of ICOT.

I've been thinking about the impact and accomplishments of ICOT since my first interactions in 1988. I have included here a specific list of accomplishments of ICOT based on my discussions and experience during the past four years.

+ Can one built a whole computing system based on logic programming and provide a useful tool for applications use? Answer: YES

+ Are the resulting systems so much easier to use that people will immediately switch from conventional computing systems? Answer: NO

+ Does special-purpose hardware give KBS a performance advantage over general-purpose hardware? Answer: NO

+ Can logic programming and KBS be applied to variety of applications areas? Answer: PROBABLY YES

+ Is the world likely to adopt KBS systems as a major alternative to object-oriented systems development environments for non-numerical computing? Answer: PROBABLY NO

+ Can logic programming and KBS open a new world of applications areas with the same effect on society (and markets) as numerical computation did in the 1950s and 1960s? Answer: TOO EARLY TO TELL

+ Did the FGCS project succeed in giving Japan new visibility in the world computer science community? Answer: ABSOLUTELY YES

To get the answers to these questions required much effort and resources. Japan was the only country willing to take the risk and to invest in obtaining these answers. The need to take risks and to try to do something new is essential. The United States and Europe have in many respects lost the ability to take these risks as a normal part of doing research. Perhaps as a result of economic decline or the collective loss of imagination, U.S. companies and government have failed to remain on the leading edge of risk-taking in large projects. I hope that Japan does not get discouraged by the international criticism of FGCS to abandon risky projects. Perhaps the RWC project is a step in the direction away from risk taking. I don't know for sure, however.

Recommendations.

I would like to make a few specific recommendations regarding the future of ICOT and the basic research agenda developed during the past ten years.

First, I think Japan should establish long-term funding for basic research in computer science and focus this work on three areas:

1. Parallel processing, performance evaluation, etc.
2. Knowledge-based programming systems
3. Combination of symbolic and numerical computation

Second, MITI–with the new policy for software distribution–can alter the view of U.S. and European governments by making all basic research results publicly available from the very beginning of any new project.

Third, research leaders in Japan need to encourage true collaborations that involve the setting of joint research objectives, joint funding, and joint management of the projects.

Fourth, basic research efforts should concentrate on software for general-purpose machines and should let industry develop the hardware and operating systems software.

Fifth, Japan should encourage smaller, more independent research groups that may be distributed with less central control, perhaps some in universities.

Acknowledgments.

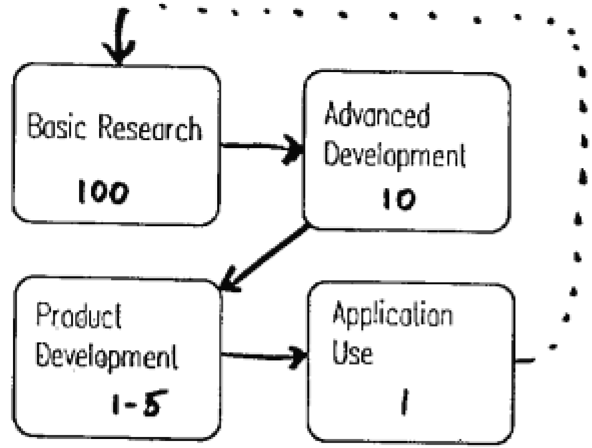
I thank all the wonderful people at ICOT who have made the interactions so pleasant during the past four years. I also wish to continue my relationships with my various Japanese colleagues in whatever way possible.

*This work was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

FGCS 92 Evaluation Workshop

Rick Stevens
Argonne National Laboratory

Western View of Computer Research and Development Process



Was the FGCS project a Basic Research Project or an Advanced Development ? **How To evaluate it**

Many in the US are confused about how to evaluate the FGCS because Japanese R&D process is not well understood.

In Basic Research a negative result is not a failure, in Advanced Development inability to get to product development is considered a failure, however not having any results would be considered failures in both cases.

Basic Research

Goal: The creation of new knowledge, both affirmative knowledge and negative knowledge.

- 1. what works
- 2. what doesn't work
- 3. what is true and what is false

Evaluation: Are important questions getting answered ?

Advanced Development

Goal: Engineering of prototype systems, determining feasibility, uncovering gaps in technical knowledge.

1. what processes work ?
2. testing for adoption, (i.e. given a choice do people use it ?)
3. application of an idea to a product

Evaluation: Are new technologies ready for product development ?

- The US was evaluating the FGCS in both categories, as a basic research project and as an advanced development project, therefore it could have been considered a success if one more or more *hard problems in AI or CS had been solved* or if a commercial *company had committed to producing products* based on the results of development.

- USA feared the project for two main reasons.

  1. It fundamentally *challenged their notions* of preeminence in basic research.

  2. If commercial products resulted from the project, Japan would have taken a lead in knowledge based systems-- an important new paradigm-- with *little western response possible* in the short term.

So what went wrong ?

I THINK it will take many YEARS To fully understand what has happened and what it means.

Lessons I have learned from the FGCS project

USA is adopting Similar STYLE Projects.

- government supported industrial consortia may not be able to "read the market", particularly over the long term.
- Don't confuse BASIC RESEARCH and ADVANCED DEVELOPMENT. (i.e. know which you are doing and don't confuse the evaluation criteria for the two.)
- Expect negative results but hope for positive. Mid-course corrections are a good thing.
- BASIC RESEARCH infrastructure needs stability, strong sense of the important core problems, flexibility and an evaluation mechanism that can distinguish between negative results and incompetence.
- VISION: The vision is critical, the most important thing people need is a BIG DREAM to make it worthwhile to get up in the morning.

"Make no small plans for small plans have no power to stir men's souls."

IT IS IMPORTANT TO HAVE VISION.

Specific Accomplishments

- Can one built a whole computing systems based on logic programming and provide a useful tool for applications uses? Answer: YES
- Are the resulting systems so much easier to use that people will immediately switch from conventional computing systems ? Answer: NO
- Does special purpose hardware give KBS a performance advantage over general purpose hardware ? Answer : NO
- Can logic programming and KBS be applied to a wide variety of applications areas ? Answer: Probably Yes
- Is the world likely to adopt KBS systems as a major alternative to Object-Oriented systems development environments for non-numerical computing ? ANSWER: Probably Not
- Can logic programming and KBS open up a whole new world of applications areas with the same effect on society (and markets) as numerical computation did in the 1950's and 1960's ? ANSWER: Too Early to Tell
- Did the FGCS project succeed in giving Japan new visibility in the world computer science community? ANSWER: Absolutely YES

Many of these problems are common to many areas of computer science Research.

What did Japan get for $US 430 Million Investment?

- Estimated 2.25 Million Lines of KL1 and ESP code ($US 200 Million to produce at commercial rates)
- Over 700 reports and papers ($US 35 Million to produce at National Lab Rates)
- New Generation of Symbolic Computing Architectures (about 8 or 9 architectures $US 20 Million each = $US180 Million)
- Three international Computer science conferences ($US2 Million each (?) or $6 Million)
- Training a new generation of computer architects and scientists. ($US 20 Million)

Specific Recommendations for the follow-on of ICOT

- Establish LONG TERM funding for BASIC RESEARCH and work in:

  1. Parallel Processing, performance evaluation, etc.
  2. Knowledge Based Programming systems
  3. Combination of Symbolic and Numerical Computation

- Make all BASIC RESEARCH results publically available from the very beginning of any new project.

- Encourage TRUE COLLABORATIONS which involve the setting of joint research objectives, joint funding and joint management of the projects.

- Concentrate on software for general purpose machines and let INDUSTRY develop the hardware and operating systems software.

- Encourage smaller more INDEPENDENT research groups that may be distributed with less central control, perhaps some in universities.

Report on ICOT

Mark E. Stickel

June 9, 1992

With the generous sponsorship of the U.S. National Science Foundation, I had the privilege of visiting the ICOT Research Center and participating in its research program for nine months during September-November 1990, March-May 1991, and November 1991-February 1992. It was a wonderful personal and professional experience. The people at ICOT were unfailingly kind and friendly. I would especially like to thank the people I worked most closely with: Masayuki Fujita and Ryuzo Hasegawa. I had many interesting technical and personal discussions with them and saw in them, and their colleagues, the time, energy, interest, and intelligence they invested in their research. Fujita and Hasegawa shared my love of theorem proving research. I would also like to thank Koichi Furukawa, whom I met before coming to ICOT, who stimulated my interest in visiting ICOT, and with whom I also had many technical and personal discussions while at ICOT. He and Dr. Fuchi created the excellent research climate at ICOT and fostered the work on theorem proving there. I would like to thank Kazuhide Iwata for his friendship and help with the daily details of living in Tokyo, including arranging for an apartment. I had good personal and professional interactions with too many other people at ICOT to name here.

I was briefed on the research activities of the various laboratories at ICOT and found much high quality research that was of interest to me: constraint logic programming, knowledge representation, natural language processing, parallel software and hardware, and, of course, theorem proving. In the end, though I would have enjoyed working on any number of activities at ICOT, I concentrated on theorem proving, the area of strongest personal interest and an area where I thought I could contribute the most while also demanding the least time of ICOT researchers to educate me in what they were doing.

On various occasions, I lectured on the Prolog Technology Theorem Prover (PTTP), cost-based abductive inference, equality theorem proving, theory resolution, upside-down meta-interpretation of model elimination, and theorem proving in general. While at ICOT, I researched and wrote about upside-down meta-interpretation of model elimination, a unit-resulting extension of PTTP, and function and relation matching rules for building in theories. I discussed ICOT's MGTP, the earliest versions of which had already been written. I discussed the value and importance of term indexing to improve efficiency of theorem proving and described Argonne's approach to theorem proving. I performed many experiments with theorems ICOT was working on using Argonne's OTTER and my own PTTP. I implemented discrimination-net-based term indexing in KL1; this was used in some versions of MGTP. Aspects of the Argonne approach that I described and praised to them also found their way into versions of MGTP.

A focus of theorem proving research at ICOT was to prove a set of theorems offered by Argonne's Ross Overbeek as challenge problems to test the performance of theorem proving programs. These are difficult problems solved by few programs. During and between my visits to ICOT, I saw steady progress being made on solving these problems. They were first solved by use of many heuristics, then without heuristics but with very poor parallel speedup, and finally solved with near-linear speedups without heuristics. A proof "look-ahead" capability often allows the ICOT prover to solve problems after generating many fewer clauses than Argonne's OTTER.

These problems were solved by a "nonground" MGTP designed to solve such non-range-restricted Horn problems. The other, original "ground" MGTP is designed for range-restricted non-Horn problems. Ground MGTP is based on Manthey and Bry's SATCHMO theorem prover and is basically a hyperresolution theorem prover that performs case-splitting on non-unit, positive derived clauses. Case splitting is feasible because the range restriction ensures that derived clauses are ground, so there is no problem with variable sharing between cases. Ground MGTP is especially well suited to implementation in KL1 on PIMs. The range restriction implies that whenever a pair of terms is unified, at least one of those terms will be ground (variable free). This permits the efficient implementation of theorem-proving variables by KL1 logical variables and the use of KL1's one-way, single assignment unification. By contrast, nonground MGTP required a unification algorithm, with theorem-proving variables represented by integers and substitutions represented by vectors indexed by these integers, be written in KL1, resulting in a costly slowdown. Case-splitting in ground MGTP provides lots of work for many processing elements, with low communication requirements, thus making it easy to achieve high speedup factors. In contrast, the successful achievement of high speedup factors for the nonground MGTP required much experimentation and refinement of work distribution schemes.

Ground MGTP is still a niche theorem prover, well-suited only for range-restricted, non-Horn problems. It is motivated by the case-splitting possibilities of non-Horn problems, so it offers nothing extra for Horn problems. Although any problem that is not initially range-restricted (every variable of a positive literal of a clause must also appear in a negative literal of the clause) can be translated into one that is by adding a "dom" predicate that inductively defines ground terms of the domain and qualifying non-range-restricted clauses with "dom" literals, this is usually not very effective. Nevertheless, range-restricted non-Horn problems appear to be a useful niche. Evidence of this comes in the form of ICOT's recent solution of an open problem in mathematics. MGTP thus joins the very short list of theorem provers that have solved open problems. The contribution of ICOT's parallel hardware to this proof is noteworthy as well. One of these open problems was solved in 3 hours on a 256-processor PIM; on a single processor of the same power, this would have required waiting a month for the solution. This represents a qualitative difference in the theorem proving process that makes otherwise nearly unthinkable tasks doable. Inoue and others have also been doing excellent work in demonstrating the usefulness of MGTP-style reasoning for nonmonotonic reasoning, diagnosis, etc. Many AI reasoning problems seem naturally formulatable for execution by ground MGTP. Upside-down meta-interpretation permits the bottom-up MGTP to be imbued with goal-directedness.

Outside of theorem proving, I think ICOT's contributions are many. ICOT's scientific contributions, particularly in the area of logic programming languages, practice, and theory, are competitive with that of other research institutions around the world. ICOT is an internationally recognized research center. Through foreign visitors coming to ICOT, ICOT researchers visiting overseas, ICOT's organization and participation in conferences and workshops, publication of research results in technical reports, conference proceedings, and journals, often in English, ICOT and Japan are participating strongly in the international computer science community. ICOT's making the software developed in the Fifth Generation Project freely available is an extension of ICOT's deliberate policy of openness and is a noteworthy and generous contribution to the scientific community.

I think ICOT has been very successful in training its researchers in logic programming, parallel processing, and the methods and values of computer science research. Perhaps they can bring about greater commercial use of logic programming when they return to their companies. Still, the barriers to new programming methodologies in industry seem high. With my long experience programming in LISP as well as Prolog, industry's failure to recognize the value of alternative methods has long been obvious and disappointing. Considerations other than technical merit often determine what is successful, such as MS-DOS and C.

ICOT's objectives seem fundamentally right. Parallel processing is the right way to provide lots of computing power cost-effectively. MIMD architectures are more easily used for a variety of applications than more restrictive computational approaches. A focus on symbolic computing applications is a needed counterpoint to the usual emphasis on supercomputing for numeric, scientific applications. The KL1 programming language is a major accomplishment of ICOT. It is an elegant parallel logic programming language that facilitates writing parallel programs while easily avoiding synchronization errors. Writing PIMOS and earlier operating systems entirely in logic programming languages is an massive demonstration of the suitability of such languages for low-level systems programming as well as high level applications programming.

Full exploration of the capabilities and limitations of a concurrent logic programming approach demanded that everything be written in KL1 as a research methodology. After writing out as much information and developing as many programming techniques as we can this way, I think that acceptance of KL1 in the marketplace will be enhanced if KL1 procedures can be combined with procedures in other languages. This could save the cost of rewriting in KL1 procedures already available in other languages. Also, sometimes performance can be significantly improved by rewriting small portions of systems in a lower-level language. The performance loss of the heavily used unification algorithm written in KL1 in nonground MGTP suggests a case in which performance could be improved substantially.

ICOT also created experimental hardware to support parallel symbolic processing. However, the wave of language-specific processors seems to have crested and past. In the LISP world as well as the Prolog and logic programming world, there now seems to be little interest in special processors. While language-specific processors can certainly deliver superior performance compared to general-purpose processors, less money and resources are available for their development than for general-purpose processors. Development costs of specialized processors must be spread over a much smaller market, often rendering them uncompetitively expensive. The revenues generated by a mass-marketed general-purpose processor can provide funds to improve its performance even on tasks for which it somewhat ill-suited, enough to ultimately become competitive with specialized processors.

The lack of commercial appeal of ICOT's prototype Parallel Inference Machines (PIMs) is a direct result of the decision to use specialized processors. In this respect, the hardware group was asked to play a supporting role in the project by providing hardware designed around ICOT's software research effort instead of designing machines with wider appeal by using more standard processors. ICOT succeeded in building machines with hundreds of powerful processors and achieved the goal of building a machine delivering 100 megalips of performance. No more suitable machine for ICOT's work is available: the specialized processors do provide a performance gain for KL1 over standard processors, and other large MIMD processors aren't really quite available yet. PIMs and the earlier Multi-PSI machines provided the necessary testbed for ICOT's research, providing high performance, reliability, and availability.

I think the widespread propagation of the technology that ICOT has developed depends on porting it to commercially popular architectures. KL1 should be ported to standard processors. Standard, commercially available large multiprocessors don't exist yet, but ICOT's system and application software should be ported to run on them when they do. Besides providing the software ICOT developed, ICOT should find some means of instructing others in the programming methodologies they used to write huge systems using KL1.

The inception of ICOT was accompanied by great expectations. ICOT certainly failed to solve "the AI problem" and thus be viewed as an unqualified success by the world press. But neither has anyone else, and ICOT has contributed as much as other research centers. In its research approach, ICOT was always willing to build things: applications, languages, operating systems, processors, multiprocessors. They did not restrict themselves to developing paper theories, but realized them in hardware and software. Implementation is a good test of value of ideas, and I think ICOT's willingness to experiment with the technologies they devised is very healthy.

I hope that ICOT continues. Establishing a research center with an international reputation is no small task. The investment to develop the research center, to establish a core of researchers and managers, a set of operating procedures, and a culture, has been made and should be preserved. The PIM multiprocessors have only recently been completed, so there has been little opportunity to experiment with or evaluate them yet. More effort is required to propagate ICOT's ideas and software to the world.

Some Reflections on the Fifth Generation Project

Seif Haridi & Siwert Sundström
Swedish Institute of Computer Science

This report represents the personal opinions of the authors.

Summary

The Fifth generation project has been a very valuable project. ICOT has served as a source of inspiration to international research in the area. It is not unfair to say that many research programmes and institutes were conceived in reaction to the establishment of ICOT. ICOT has made a solid progress in applying parallel processing to symbolic computing and knowledge information processing. Many of the ideas and concepts developed by ICOT have been further developed and modified at other research centers. We believe that ICOT has contributed to and affected other research more than outside research has affected and influenced ICOT. This may have been inevitable due to the fixed nature of ICOT's long-term plan. Therefore, we must also look outside ICOT to judge the results of the project.

The internal technical results, given the constraint of a fixed time period, and the temporary nature of its personnel, surpass expectations. However, we find that at this stage the project remains unfinished.

In order to fully achieve the goals of the project, research should continue on improvement of the software generated by the project and on careful evaluation of the software and the parallel architecture prototypes. It is also of extreme importance to port the valuable software generated by the project to widely accessible general purpose machines, both parallel and sequential. This is necessary in order to disseminate the results of the project to the international research community and to create a basis for possible commercial exploitation. Experience from developing knowledge processing tools and databases has to be documented and fed back to improve the design of KL1. In particular, it seems that there is a need to improve the process communication abilities of KL1. There is also a need to incorporate recent research results on integrating the programming paradigms of Prolog into the language as shown in the Andorra family of languages. This is vital to facilitate the development of knowledge processing tools. Finally there is a need for research on software tools that integrate

both knowledge information processing and numeric computations.

The Goals of the Project

It is the understanding of the authors that the ICOT project was set up with the following combined goals:

1. To develop the knowledge processing technology and the basic tools needed for advanced applications that require both deductive and intensive symbolic processing capabilities. These applications were predicted to be increasingly in demand during the nineties and early 21st Century.

2. To exploit and develop parallel processing technology and highly parallel architectures in order to meet the performance requirements expected for the above class of applications.

3. As the link or glue between the higher level knowledge-based applications and the highly parallel machine at the lower level, an intermediate programming language based on logic programming was to be developed. This language, on one hand, would be flexible enough to implement the high level tools needed for the knowledge intensive applications, as stated in point 1, and, on the other hand, be suitable for exploiting the resources of parallel machines and making parallel programming easy as stated in point 2.

Needless to say the goals of the project at the time of its conception were very ambitious, since they required breakthroughs in many of the technological areas outlined above. However, that is not to say that they were not viable. In fact, with the current knowledge of the state of art, these goals seem achievable, and we certainly admire the courage of the project leaders to attack such challenging technological problems at such an early stage.

Organisational Structure Tradeoffs

The project leadership was given ten years to achieve these goals, and the majority of the staff were recruited from the Japanese computer industry on a temporary basis. Under these constraints, the ICOT project can be evaluated from two different perspectives, one w.r.t. its scientific output, and the other w.r.t. its industrial output. It is, however, important to notice that under the premises of the project, i.e fixed time and temporary personnel, the goals of maximising both the scientific and industrial outputs are largely incompatible and some compromises had to be made.

Just to illustrate some of the trade-offs, here are some issues:

1– To maximise the scientific output of the project, it is necessary to have well trained key persons stay on after spending their assigned period in the project (approximately 4 years) while, in contrast, to maximise the industrial benefits of the associated companies technology transfer, through experienced personnel transfer, should occur regularly. The policy used in the project, as far as the authors are aware, was something in between; most of the key persons stayed in the project while some left the project after their assigned period.

2– To fix a project period is extremely counter-creative and correspondingly counter productive as regards to the scientific output of the project. It is better in scientific endeavours to organise projects into stages, where the financing of the next stage is conditional on the progress made in the previous one, and to set the goals of the next period depending on the progress of the previous one, and possible changes in the environment of the project. This would make projects more flexible, and responsive to unexpected changes within the project and its surroundings. Moreover, if a project, under a fixed time constraint, has some concrete technological goals to achieve, a sizeable portion of its late period will inevitably be spent in pure development efforts, thus missing some opportunities of fundamental advances in the state of art during this late period. It is again the view of the authors, that the ICOT project, during its initial and intermediate period, played an important role in advancing the state of art towards achieving the goals of the project, while the last 4 years were mostly spent in developing the prototypes of the technology envisioned at the intermediate phase.

3– A scientific research project would benefit from open cooperation with other researchers and research organisations, while an industrial project is normally closed for competitiveness reasons. The ICOT project was more a scientific project than an industrial one in this respect, but not entirely. One important obstacle in this case is that most of the software developed by the project was on proprietary hardware which effectively prevented other researchers from sharing the results of the project, and resulted in the missing of opportunities that would otherwise have occurred through early feedback and valuable evaluation reports from other researchers. ICOT tried to remedy this problem by distributing some of its hardware to other research organisations but such hardware was not in fact effectively used for different reasons. Exchange of software among researchers is a very important factor in improving the quality of the research produced by any computer science research organisation. MITI lately released the software of the project in the public domain, but this should have happened earlier.

Scientific achievements of ICOT

On the positive side

Initially at the start of the project there was a clear world-wide trend to use dedicated machines for AI, and knowledge intensive applications. ICOT responded successfully to this trend by developing the PSI machine, and most importantly by developing ESP. ESP was the first programming language that combined both the logic programming paradigm as embodied in Prolog and the object oriented programming paradigm. The flexibility of the language was clearly demonstrated by programming the operating system of PSI and developing a number of tools for knowledge representation and natural language processing.

ICOT played a central role in the development of programming languages for parallel symbolic computing. The design of GHC was certainly a very important step in the development of concurrent logic programming languages.

ICOT has contributed considerably to the implementation technology of programming languages on parallel architectures. Techniques for memory management like the MRB scheme and parallel garbage collection on cluster basis, were certainly innovative techniques developed by ICOT.

ICOT has also contributed considerably to the development of constraint logic programming by for example, the work on CIL and CAL.

The development of PIMOS is the first large scale operating system for parallel computers.

ICOT demonestrated, beyond doubt, the usefulness of parallel processing in symbolic and highly irregular computations.

On the negative side
ICOT invested a lot of effort trying to compile the programming paradigms of PROLOG to KL1, but one must say that these efforts have only partially succeeded. Research done by outside researchers has shown that it is possible to combine Prolog and GHC (a good example is the work done at SICS on Andorra). However, the results came too late to be exploited within the remaining time allocated to the project. This is very unfortunate since a lot of ICOT's early research on natural language processing and constraints was based on Prolog and could not be exploited effectively when the project moved from ESP to KL1. Moreover there is some doubt about the flexibility of the communication structures of KL1.

Due to the lack of time the results of ICOT on knowledge representation and knowledge bases remain unevaluated by the outside research community.

It is questionable whether the architectures produced by the project are adequate for efficiently implementing languages like KL1. The most important issue is the lack of support of global address spaces, and multiple-users address spaces in the hardware. It is widely accepted by researchers in computer architecture that global address space is a very important factor in reducing the problems associated with dynamic data partitioning and dynamic load distribution.

Recommendation

Most of the problems mentioned above can be remedied if ICOT's research activities will continue. Therefore we recommend that ICOT:

- should perform research on improving KL1 to integerate don't care and encapsulted don't know nondeterminism.
- should carefully evaluate and improve the knowledge representation tools through feedback from sizable applications.
- should evaluate the parallel PIM machines to identify the minimal hardware requirements needed to efficiently execute parallel symbolic languages like KL1.
- should investigate the possibility to integrate more flexible communication facilities in KL1 (e.g. ports in the Andorra Kernel language), and/or the ability to handle distributed arrays efficiently (e.g. M-structures in dataflow languages).

Industrial achievements of ICOT

On the positive side

It is clear that a large number of young engineers and scientists from the associated companies have been trained in advanced technology, which would not have been possible without the ICOT project; in particular in areas of, (i) design, development and implementation of high-level programming languages,

1. knowledge representation languages, (iii) scalable operating systems and
2. design and implementation of multiprocessor architectures.

The knowledge and the skills developed while working on the cache coherent clusters of PIM/p, PIM/c and the two level cache coherent system of PIM/k must have been very important since the current commercial trend of successful parallel machines is cache-coherent ones. We congratulate the project leadership on producing the hardware prototypes on that scale given the tight time constraints.

It is a great achievement to be able to deliver on time, nearly all what has been promised when the project started 10 years ago, together with impressive number of demo applications running on parallel machines.

On the negative side

Since 1985 there was a clear shift from dedicated hardware to general purpose hardware. ICOT was not flexible enough, perhaps due to the overall project structure. A very valuable possible industrial output would have been setting standards for the design of general purpose RISC chip for symbolic languages. This might have been just adding tag support for conventional RISC chips, and/or support for multiprocessing. Another issue is the lack of active and incremental migration of ICOT's software to commercially available machines in general, and to machines belonging to the associated companies in particular. This is crucial for any commercial exploitation, and is a prerequisite for more eager adoption of new technology within the companies engaged in the project.

Recommendations

ICOT software should be ported to a wide class of commercially available machines in order to allow for adoption of the technology produced within the associated companies and the wide research communities.

Minimal extensions to existing standard hardware to efficiently support parallel symbolic computing should be identified, and reported to the associated companies to be taken into account when new generation of commercial hardware is designed by the associated companies.

Conclusions

The Fifth generation project has indeed been a very valuable project. From a Swedish perspective ICOT has succeeded in establishing close cooperation between Swedish and Japanese researchers both on the academic and industrial level. ICOT has also influenced the establishment of the Swedish Institute of Computer Science.

The internal technical results, given the constraint of a fixed time period, and the temporary nature of its personnel, surpass expectation. However, it is clear that at this stage the project remains unfinished.

In order to fully achieve the goals of the project, research should continue on improvement of the software generated by the project and on careful evaluation of the parallel architecture prototypes. It is also of extreme importance to port the valuable software generated by the project to widely accessible general purpose machines, both parallel and sequential. This is necessary in order to disseminate the results of the project in the international research community, to stimulate the adoption of this new technology within the associated companies, and to create a basis for possible commercial exploitation. Experience from developing knowledge processing tools and databases has to be documented and fed back to improve the design of KL1. In particular, it seems that there is a need to improve the process communication abilities of KL1. There is also a need to incorporate recent research results on integrating the programming paradigms of Prolog into the language. This is vital to facilitate the development of knowledge processing tools.

Some Reflections on the FGCS Project

Seif Haridi & Siwert Sundström

Swedish Institute of Computer Science

Summary

- The FGCS project has given credibility to Logic Programming research.
- The project has been a valuable source of inspiration to international research.
- Many research programmes and institutes were conceived in reaction to the establishment of ICOT.
- Many of the ideas and concepts developed by ICOT have been further developed and modified at other research centres
- The technical results surpass expectations

Our understanding of the Goals of the Project

1. To develop the knowledge processing technology and the basic tools for advanced applications that require both deductive and intensive symbolic processing capabilities.

2. To exploit and develop parallel processing and highly parallel architectures

3. To develop an intermediate programming language based on logic programming as link between knowledge-based applications and parallel machine,
   - flexible enough for knowledge intensive applications (1 above)
   - suitabel for using parallel machines (2 above)

Extremely ambitious goals reflecting admirable courage.

An Industrial or a Research project ?

Mainly a Research project, but with fixed time limit!

Some trade-offs:

1
- Scientific output needs well trained key persons to stay on,
- Industrial benefits needs regular experienced personnel transfer.
- ICOT policy, a sensible compromise

2 Fix project time, normally counter-creative, correspondingly scientifically counter productive.

Thus:

The FGCS project played a more important role in advancing the state-of-the-art during its initial and intermediate periods than during the last 4 years.

Trade-offs (cont.)

3 - A scientific research project needs to be open,
- An industrial project is normally closed

The FGCS project has indeed been very open.

However,
most of the software was developed on proprietary hardware.
ICOT tried to remedy this problem by distributing some of its hardware.

Scientific achievements of FGCS

On the positive side:

- Special purpose hardware for AI: PSI machine
- First programming language combining logic programming and object orientation: ESP
- Concurrent logic programming language:GHC
- Programming parallel architectures: memory management and parallel garbage collection
- Constraint logic programming: CIL and CAL
- First large scale OS for parallel computers: PIMOS

On the negative side

- Big efforts in converting Prolog paradigms to KL1, rather than combining Prolog and GHC.
- Knowledge representation and knowledge base techniques unevaluated.
- Lack of support for global address space and multiple-users address spaces.

Industrial impact of FGCS

On the positive side

- Large number of engineers and scientists trained in advanced technology:
  - design, development and implementation of high-level programming languages
  - knowledge representation languages
  - scalable operating systems
  - design and implementation of multiprocessor architectures
  - cache coherent systems, (successful commercial trend today)

On the negative side

- Commitment to special hardware, against world-wide trends, caused limited industrial interest.

- The project could have been setting standards for RISC chips for symbolic processing and hardware for multiprocessing.

The future:

To fully achieve the goals,
research should continue on:

- porting the valuable software to widely accessible machines

- improvement of languages and software based on feed-back from massive use

- careful evaluation of the parallel architecture prototypes

- incorporating recent research results regarding Prolog programming paradigms

Conclusions

- The FFGCS project has indeed been very valuable.
- ICOT has succeeded in establishing close cooperation between Swedish and Japanese researchers.
- FGCS influenced the establishment of the Swedish Institute of Computer Science.

FGCS Project Evaluation Report

Sten-Ake Tärnlund

1 VISION AND PLANS

The research paradigm of the FGCS project is logic programming [2], a major breakthrough in computer science and artificial intelligence from the early seventies going back to some principal notions of resolution logic [3] in the sixties, see professor Robinson's historical account [4] in the FGCS-92 proceedings. Dr. Fuchi, early to envision the potential of logic programming, gave together with several colleagues a research vision [5] based on this paradigm that enthused an entire world. He also worked out a practical research plan [6] for the FGCS project itself focusing on parallel inference computing. I shall do my best to evaluate the FGCS project according to this plan and not the vision that I, however, expect will inspire researchers for many years to come, as the Hilbert program [7], from almost hundred years ago, still inspires mathematicians.

2 MAJOR ACHIEVEMENTS

My evaluation will follow the presentations of the FGCS project given at the conference and in the proceedings.

2.1 Parallel Inference Machine (PIM)

The various PIM machines, are the centerpiece of the FGCS project. They differ somewhat in architecture, which opens up possibilities for experiments. They are all MIMD machines. A PIM has clusters of processing elements connected in a network of clusters. It is scalable at the three levels. A prototype machine PIM-m is used for demonstrations. They are scalable farther than thousand processors. They run the PIMOS operating system and the KL1 language. The performances of a PIM is about 100 MLIPS, although these LIPS may not quite be the original LIPS, they show that the FGCS project has more than reached the original bold performance target. More to the point, we can expect performances beyond the 100 MLIPS up to a GLIPS and beyond. This performance and these machines are brilliant results and could not have been taken for granted when the project started. On the contrary, after this project started other machine ideas have been proposed e.g., the Connection machine [8] in the US, In contrast, it is a SIMD machine, its network is less scalable and more limited by communication speed, these factors make it a more special machine for particular problems. So, the project could have gone wrong, but it didn't and it produced a unique general machine that cannot be seen anywhere else in the world. This is an outstanding result of its designers. In fact, the PIM itself justifies the focus of the FGCS project on parallel inference machines for logic programming.

One could, of course, argue whether or not the project itself should have used more hardware and software available on the market or develop its own as it mostly did. In general, I think both alternatives are justifiable. For Japan, being so strong on hardware, this project with logic programming as its key idea, is just an excellent place for various advanced hardware studies and experiments on machine prototypes. On the other hand, it would be possible to obtain several of the interesting results with more market hardware and software. This would show that these results are repeatable - a good scientific criterion. This line of research may lead to more procedural software, and that could impose a constraint on massive parallel processing, leading to more complicated programming and thus more expensive software development.

In general, machines of this structure [9], but with more market hardware and software, is likely to become general purpose parallel machines on the market in the near future. So as a bonus from this project, Japanese manufacturers could go from these PIM prototypes to competitive massive parallel computers with market software - a pleasant spin-off from the FGCS project.

2.2 Parallel Inference Software (PIMOS)

Pimos is a parallel operating system for a PIM and is written in the concurrent language KL-1 that includes control facilities e.g., for resource management. The concurrent languages GHC [10] and KL1 [11], are among the most prominent results of the FGCS project. GHC is an elegant successor of Parlog [12] and Concurrent Prolog [13]. It is famous in the world. KL1 has clearly been very useful for developing most of the software for the PIM:s. Nobody has done an operating systems project like this elsewhere, and it adds significantly to the success of the FGCS project. The operating systems principles of the PIMOS experience and the methodology of KL-1 comprehend new interesting concurrent programming ideas. We are keen to learn more about them and they deserve wider publicity.

At this point, a comment on the concurrent versus declarative logic programming paradigms [14] may be appropriate. The invention of concurrent languages by Clark & Gregory, Shapiro and Ueda & Chikayama has stimulated a great interest in solving concurrent problems that occur frequently in operating systems. Although, this approach departs from the declarative (semantic) idea of logic programming, these languages provide interesting techniques for concurrent problems e.g., processes, committed-choice, stream and-parallelism, and communications with terms containing variables and bounded-buffers. Taking up Quine's point [15] a solution in a concurrent language can, however, be regimented to a solution in logic. Futhermore, concurrent computations are not identical to parallel computations or always related to better performance of a parallel inference machine. In general, I view the ideas and the experiments on concurrency as brilliant and a very successful part of the FGCS project. The PIMOS system itself is unique!

2.3 Knowledge Base Management Software (KBMS) and Basic Artificial Intelligence

The methodology and general principles as well as the application experiments used in the FGCS project are very interesting. In particular, the application experiments have been useful in the field of artificial intelligence. Several special purpose languages have been developed for various applications e.g., CIL for natural language processing. There is also more basic AI research e.g., on theorem proving, hypothetical reasoning, analogy and non-monotonic reasoning. These are areas where results from logic programming have made several interesting contributions to the AI field recently. The results of the FGCS project make this progress even stronger.

2.4 Knowledge Programming Software and Programming Methodology

Beside the development of the concurrent languages there are also several other important results on programming methodology mainly building on results in the areas of: constraint programming [16], partial evaluation [17], meta programming, and program transformation. Several of these research results are superb, and have positioned Japanese research at the frontier. Some of these results could also play an important role for future software engineering. In fact, this methodology is particularly significant for the FGCS project since it could become a bridge between (declarative) logic programming and efficient parallel inference machines e.g., by automatically transforming logic specifications into parallel logic programs.

2.5 Experimental Parallel Inference Software

The project shows several fascinating choices of applications e.g., in legal reasoning that are impressing. The applications often demonstrate the parallel power of a PIM. Some of them have good potential of becoming blossom applications. This would be an interesting result in itself, but the methodology of developing such applications is also very interesting. With the PIM: s and the craftsmanship of logic programming methodology at hand the researchers at ICOT are well placed to develop extraordinary applications. In fact, to take advantage of logic programming and its problem solving competence, and the parallel inference machines for efficient computations, may also be a good subject for a sequel project.

3 SUMMARY OF RESULTS

I will sum up my evaluation in six parts.

3.1 Research

Principal results:

- the PIM prototypes and their architectures;
- performance of PIM of 100 MLIPS and potential for more;
- concurrent languages, GHC and KL1, and their methodology;
- logic programming methodology on: concurrent software, constraint
- programming, partial evaluation, meta programming, and program transformation;
- several interesting applications e.g., legal reasoning.

A superb result, and an outstanding achievement. Consequently, I would rank the ICOT work on par with the best in the world.

The successful advance and experiments of the FGCS project validates some essential aspects of the logic programming paradigm - declarative logic programming for (economical) software development and parallel inference machines for efficient computing.

3.2 Research education and training

Several of the researchers work for Japanese companies. They have been visiting researchers at ICOT and will return to their companies. This leads to a large scale training of researchers at the Japanese companies that should yield significant spin-off effects in the future.

3.3 Project planning

ICOT has had a 10-year working period. This may impose an earlier commitment to a research idea than some researchers may want, on the other hand a commitment gives the opportunity to come up with concrete prototypes e.g., a PIM.

3.4 Scientific responses

We all know the impact of the FGCS project around the world. Japan has given a distinguished mark on computer science research in general and logic programming in particular. I shall only mention a few responses that I have seen: ESPRIT in Europe; MCC and the strategic computer project in the US; and in Sweden research programs by STU and the creation of SICS. In addition, private companies have made significant efforts e.g., IBM.

3.5 Economics

As pointed out above, Japanese manufacturers have got the opportunity to go from the PIM prototypes to competitive massive parallel computers with more market hardware and software. Although such a move involves more than technical aspects, this economical opportunity is a pleasant spin-off from the FGCS project.

3.6 Public relations

We have seen and will see many press reports on the FGCS project. Some may be somewhat superficial as the article in the International Herald Tribune, the other day. Some will be more sophisticated as the two consecutive editorials in the New York Times, some time ago, interpreting the FGCS project not only as a success but recommending that the US should learn from this project and launch a new industrial policy for its computer industry. This recommendation is, of course, equally valid for Europe. In general, the results of the FGCS project are not yet fully understood and appreciated, and we would benefit from a better understanding of the achievements of the FGCS project.

4 A FEW RECOMMENDATIONS

Japan has set the stage for future research based on the success of the FGCS project. I shall only briefly mention some potentials for an international cooperation based on the Japanese results and the international results, in particular, from the many responses to the FGCS project around the world.

There are several interesting subject matters e.g., from parallel inference machines to advanced AI via programming methodology. Logic programming would be helpful to strengthen the coherence of such an effort. This research paradigm was considered the most powerful for the FGCD project. I do not think it has been replaced, new projects would energize the original vision and be a challenge for the new millenium.

Some principal areas would be:

- parallel inference machines;
- software methodology e.g., specifications, transformations and verifications;
- software development methodology;
- artificial intelligence methodology;
- advanced applications.

5 ACKNOWLEDGMENTS

I would like to thank the leadership of the FGCS project and all its members I have had the pleasure to meet and work with, for example, in several enjoyable workshops. In particular, I admire the works and monumental effort by Dr. Fuchi, Dr. Furukawa and Dr. Uchida to make this project such an outstanding success.

NOTES:

[1] Address: Computing Science Department, Uppsala University,
P.O. Box 520, S-751 20 Uppsala Sweden.
Phone: + 46 010 222 4002.
Fax: +46 18 521270.
Email: stenake@csd.uu.se.

[2] R. Kowalski, Predicate Logic as Programming Language, Proc. IFIP-74, Stockholm, North-Holland. Colmerauer, A., Kanoui, H., Pasero, R., & Roussel. P., 1972, Un Systeme de Communication Homme-Machine en Francais, Univ. d'Aix-Marseille, Luminy, Marseille.

[3] J., A., Robinson, 1965, A Machine-oriented Logic Based on the Resolution Principle, J. ACM, New York.

[4] J., A., Robinson, The Role of Logic in Computer Science and Artificial Intelligence, Proc., FGCS-92, Tokyo.

[5] T., Moto-oka, (ed.,), Fifth Generation Computer Systems, JIPDEC: North-Holland, 1982.

[6] K., Fuchi., in T., Moto-oka, (ed.,), Fifth Generation Computer Systems, JIPDEC: North-Holland, 1982.

[7] Hilbert, D., Mathematical problems. Lecture delivered before the International Congress of Mathematicians at Paris in 1900. Bull. Amer. Math. Soc., vol 8, 1901-2.

[8] Hillis, W.,D., The Connection Machine, MIT Press, Cambridge, 1985.

[9] CM-5, iPSC/Delta, nCUBE etc.,

[10] Ueda, K., Guarded Horn Clauses, in Logic Programming 85, E., Weda (ed.), Lecture Notes in Computer Science 221, Springer-Verlag, 1986.

[11] Ueda, K., and Chikayama, T., Design of the Kernel Language for the Parallel Inference Machine, The Computer Journal, Vol 33, No 6, 1990.

[12] Clark, K., and Gregory, S., PARLOG: A Parallel Logic Programming Language. Research Report DOC 83/5, Department of Computing, Imperial College of Science and Technology, 1984.

[13] Shapiro, Y., A Subset of Concurrent Prolog and its Interpreter. Tech. Report TR-003, ICOT, 1983.

[14] The concurrent versus the logic programming issue is an instance of an issue of philosophy of logics i.e., whether or not a problem e.g., a concurrent, modal or temporal problem should have its own logic or should be solved in standard predicate logic as argued by Quine see for example Word and Object, Wiley, 1960.

[15] See the preceding footnote.

[16] Colmerauer, A., Opening the Prolog III Universe: A new generation of Prolog promises some powerful capabilities, BYTE, August 1987. Jaffar, J., and Lassez J-L, Constraint Logic Programming, In 4th IEEE Symposium on Logic Programming, 1987.

[17] Futamura, Y., Partial Evaluation of Computation Process: An Approach to a Compiler-Compiler. Systems, Computers, Controls 2,1971.

Sten-Åke Tärnlund
Uppsala Univ.

Fifth Generation Computer Systems (FGCS)

Logic programming
A major brokethrough in computer science!
Parallel inference
A very important computation method for speed.

Project program
Vision - plans

Cf The Hilbert program for mathematics.

Major Achievements

1. Parallel Inference Machine (PIM)
A distributed cluster MIMD machine. Unique!
100 MLIPS
Cf. Connection machine
Comments: General vs special hardware.

2. Parallel Inference Software (PIMOS)
Nobody has done this elsewhere

3. Knowledge Base Management Software (KBMS)
See my report

4. Knowledge Programming Software
GHC KL1
Experiment in CLP. concurrent versus declarative LP

5. Experimental software
Many interesting applications e.g. legal reasoning

Suggestions & Conclusions

Research results.

I would rank the best of the work on par with the best in the world e.g. PIM, GHC...

* performance 100 MLIPS

* implicit methodology for software development.
* logic programming paradigms are successful

Scientific politics

Successful
Planning of the project good for concrete results.

Economical

Probably good prospects.

Politics

Herald Tribune may view the project as a failure but New York times viewed it successful!

Finally, the FGCS results should be continued!

ICOT Evaluation Report

Evan Tick
Department of Computer Science
University of Oregon
Eugene, OR 97403 USA

June 16, 1992

1 Introduction

This report summarizes my views of the Fifth Generation Project (FGCS) conducted at ICOT over the period of 1982–1992. My participation is somewhat unique because I was both an ICOT visitor in February 1987 and then a recipient of the first NSF-ICOT Visitors Program grant, from September 1987–September 1988. I joined the University of Tokyo, in the Research Center for Advanced Science and Technology (RCAST) at that time, with a visiting chair in Information Science donated by the CSK Corp. Thus over the period of 1986–1989 I had an "insider's view" of the FGCS Project. My area of research concerned performance evaluation of parallel logic programming paradigms [12, 15, 8, 11, 9, 10, 17], primarily working with M. Sato. I also worked on multiprocessor cache protocols with A. Goto and A. Matsumoto [3] and compile time estimation of task granularity in concurrent languages [7, 13]. During my stay at ICOT I had the opportunity to begin writing a book describing then state-of-the-art approaches to parallelization of logic programs [14].

Furthermore, my collaboration with ICOT researchers continued after my return to the U.S. During that period, the primary research was in the area of parallel garbage collection (with A. Imai [5, 4, 6]) and continued shared interest in distributed parallel algorithms, specifically for the best path problem (with N. Ichiyoshi [16, 1, 2]). (I included a bibliography of my own work associated with ICOT in order to convey the impact and importance on my own line of research. Of course, ICOT members have published extensively and produced results in numerous areas.)

My experience as a Stanford post-doctoral research associate, working full-time at ICOT in 1988, was very rewarding. I had previously worked at IBM Yorktown Heights (with T. Agerwala and D. DeGroot), and at SRI Menlo Park (with D. H. D. Warren), and the ICOT research environment and academic comraderie was on par with these institutions. Summaries of the FGCS Project successes and failures by most foreign researchers tend to categorize the abstract vision (of knowledge engineering and a focus on logic programming) as a great success and the lack of commercially competitive hardware as the main failure. I would like to comment on more subtle successes that fewer observers had a chance to evaluate. These successes, as alluded to by Dr. Fuchi in some of his remarks in the Evaluation Workshop, involve the training of a generation of computer scientists.

2 A New Generation

In these preliminary remarks, I will not go into the numerous details and implications of this "side effect" of the FGCS Project (I leave that detailed analysis for a journal article I am preparing on the subject). I will summarize the main points as I see them from the vantage point of working shoulder-to-shoulder with ICOT members.

1. increased communication culture — ICOT infrastructure was unique for Japanese research groups in the early 1980's in that it supplied researchers with various communication channels that normally did not exist in the corporate culture:
    - company-to-company interaction engendered by the cooperative efforts of engineers from several companies.
    - electronic mail, increasing international as well as local information flow.
    - company-to-university interaction engendered by the Working Groups (initiated at the inception of the project). A related point is PhD production, discussed below. I know best that Prof. Tanaka of the University of Tokyo has a close research relationship to ICOT, working on design of inference multiprocessors and languages throughout these past ten years. Other professors have also influenced and are influenced by ICOT research.
    - Japan-to-international research community interaction engendered by the high value placed on the publication and presentation of research results.
2. post-graduate education — ICOT served as a substitute for OJT ("on-the-job training"), and in doing so, graduated a generation of engineer/managers well educated in advanced areas of computer science and better able to manage their own groups in the future. The latter point applies to both the management of engineering groups as well as political management, learned by a close relationship with MITI.
3. "corporate culture explosion" — I know of no other words to aptly describe the movement away from the culture of lifetime employment. I believe that ICOT coincided with greater forces within Japan causing this revolution; however, the revolution was certainly felt within the FGCS Project. Several, not a few, ICOT members switched their affiliations between companies and from companies to universities. If there was ever an industry to foster such a movement, it would certainly be a high-technology area such as computer engineering, therefore this should not come as a surprise. However, I think it did catch some of the companies by surprise.

4. PhD generation — ICOT generated an abnormally high (compared to comparable research organizations within Japan, which is a difficult comparison to make) percentage of PhDs from work conducted. Furthermore, it generated several professors, going to most of the major Japanese universities, as well as others.

Complete analysis of these four points could take many pages, so I will only give my (unsubstantiated) opinions here. I think all these natural results of the "market forces" acting on young ICOT researchers are positive. Increased communication between engineers, managers, professors and students will lead to more rapid progress in developing basic research ideas into successful commercial products. The question remains in my mind as to whether a National Project of this magnitude is necessary to create these human networks each generation, or if this first network will propagate itself without help from another project? It is reasonable to assume a mixed success, i.e., the networks will weaken with age, but will remain in place. Thus in the future, it may not require such a grand-scale project to strengthen ties. For example, current ICOT graduates, understanding the importance of free and flexible discussion of results at national conferences, will increase the participation of the researchers in their care, thus enabling the next generation to form their own friendships and working relationships.

3 About Technology

I will exploit this opportunity to discuss the validity or commercialization of the processing technologies developed by ICOT, specifically the idea of building a special-purpose multiprocessor to execute a fine-grain concurrent language. This seems to be the main concern of the press, and perhaps the key point upon which ICOT is being evaluated. One could criticize ICOT for attempting to naively leapfrog "fourth generation" RISC-based microprocessor technologies, which continue yearly to grow in performance. Ten years ago, Japanese companies did not have experience developing microprocessor architectures, much less second-generation RISC designs (superscalar), nor MIMD multiprocessor designs. Building the various PIM machines gave the hardware manufactures some limited experience in microprocessor design, although presumably this experience could have been had with a more conventional target.

On one level, however, the unique experience that *was* attained, i.e., that of fabricating tagged symbolic architectures, contains much of the structure needed to tackle the problem from the bottom-up, as being done by conventional multiprocessor vendors. It is not surprising that operating systems are now developing light-weight threads, and that languages such as object-oriented Smalltalk and tuple-based Linda form the cores of recent distributed processing efforts. My contention is that these efforts are climbing from the bottom-up, whereas ICOT had a top-down approach to the same problem (of massively parallel symbolic computation). Furthermore, because I know firsthand of the quick responsiveness of Japanese research and development in this area, I have little doubt that these two methods will be bridged. If the performance gap is bridged, the key question is *who will be in the better position*?

The top-down approach has advantage of programming and application experience in concurrent and symbolic, high-level languages. The bottom-up approach has the advantage of using imperative languages that evolved slowly, thus retaining market share. There is no clear answer to this question, but let me rephrase it in terms of two specific technologies: wormhole-routed distributed networks and concurrent constraint languages. I believe both required significant intellectual efforts to conceptualize, design, implement, and apply in real systems. The former represents a bottom-up technology and the latter a top-down technology. Bottom-up technologies are easier to introduce into designs, e.g., PIM/m incorporates wormhole routing (and can execute GDCC, a constraint language), whereas the Intel machines do not yet have implementations of constraint languages. Perhaps GDCC can be ported to general-purpose multiprocessors, but that is *not* the issue. Where GDCC came from, and where it is going, can only be determined from the foundation of the research expertise gained in its development. This is of course true about routing technologies, but again, bottom-up technologies are more easily imported (and more easily sold — they translate more directly to FLOPS).

4 Conclusions

I would like to finish this essay with a prescription for guaranteed success at generating the positive "social" results of the FGCS experience. Much discussion at the Evaluation Workshop concerned the issue of software. I think the distribution and availability of the great body of work is valuable, but not the main issue. The main issue to making the FGCS Project a "success" is to guarantee the high-level of computer science research initiated by ICOT. Shapiro suggested giving grants to foreign institutes for collaborative efforts. A better idea, in my view, would be to create a permanent Japanese Institute of Computer Science, which could accept international visitors, company trainees, and hire directly from the universities. There may be few proponents of this idea, primarily because of the game-theory effect that it appears to be in no individual's best interest, i.e., not the universities (who have their own institutes), the national labs, or the companies. However, taken in total, it would in fact benefit Japan because a truly first-rate Institute, with an international reputation, would produce all the beneficial effects of ICOT, without the pressure of producing advanced technology products in limited time periods. The Institute would allow a unique opportunity for fresh university graduates to tackle advanced problems in computer science with the support of industrial technologies. For example, next-generation CPU and network designs are best produced with combined resources, allowing the companies to develop their own current-generation designs.

References

[1] P. Adamson and E. Tick. Greedy Partitioned Algorithms for the Shortest-Path Problem. *International Journal of Parallel Programming*, 20(4), August 1991.

[2] P. Adamson and E. Tick. Parallel Algorithms for the Single-Source Shortest-Path Problem. In *International Conference on Parallel Processing*, Penn State, August 1992.

[3] A. Goto, A. Matsumoto, and E. Tick. Design and Performance of a Coherent Cache for Parallel Logic Programming Architectures. In *International Symposium on Computer Architecture*, pages 25–33. Jerusalem, IEEE Computer Society, May 1989.

[4] A. Imai and E. Tick. A Shared-Memory Multiprocessor Garbage Collector and its Evaluation for Committed-Choice Logic Programs. In *Symposium on Parallel and Distributed Processing*, pages 870–877. Dallas, IEEE Computer Society, December 1991.

[5] A. Imai and E. Tick. Evaluation of Parallel Copying Garbage Collection on a Shared-Memory Multiprocessor. In *Proceedings Japanese Symposium of Parallel Processing*. Tokyo, May 1991.

[6] A. Imai and E. Tick. Evaluation of Parallel Copying Garbage Collection on a Shared-Memory Multiprocessor. *IEEE Transactions on Parallel and Distributed Computing*, 1992.

[7] E. Tick. Compile-Time Granularity Analysis of Parallel Logic Programming Languages. In *International Conference on Fifth Generation Computer Systems*, pages 994–1000, Tokyo, November 1988. ICOT.

[8] E. Tick. Performance of Parallel Logic Programming Architectures. Technical Report TR-421, ICOT, Tokyo, September 1988.

[9] E. Tick. A Performance Comparison of AND- and OR-Parallel Logic Programming Architectures. In *International Conference on Logic Programming*, pages 452–470. Lisbon, MIT Press, June 1989.

[10] E. Tick. Comparing Two Parallel Logic-Programming Architectures. *IEEE Software*, 6(4), July 1989.

[11] E. Tick. Parallel Logic Programming on Shared-Memory Multiprocessors: A Tale of N-Queens. In *Proceedings Japanese Symposium of Parallel Processing*. Atami, February 1989.

[12] E. Tick. A Performance Comparison of Shared-Memory OR- and AND-Parallel Logic Programming Architectures for a Common Benchmark. *Journal of Information Processing*, 13(1), 1990.

[13] E. Tick. Compile-Time Granularity Analysis of Parallel Logic Programming Languages. *New Generation Computing*, 7(2):325–337, January 1990.

[14] E. Tick. *Parallel Logic Programming*. MIT Press, Cambridge MA, 1991.

[15] E. Tick and J. A. Crammond. Comparison of Two Shared-Memory Emulators for Flat Committed-Choice Logic Programs. In *International Conference on Parallel Processing*, volume 2, pages 236–242, Penn State, August 1990.

[16] E. Tick and N. Ichiyoshi. Programming Techniques for Efficiently Exploiting Parallelism in Logic Programming Languages. In *SIGPLAN Symposium on Principles and Practices of Parallel Programming*, pages 31–39, Seattle, March 1990. ACM Press.

[17] E. Tick and K. Susaki. TOESP: A Prolog Benchmark. Technical Memo TM-451, ICOT, 1-4-28 Mita, Minato-ku Tokyo 108, Japan, January 1988.

Evaluation of the FGCS Project

David H. D. Warren
Department of Computer Science
University of Bristol

General Impact

The FGCS project had a major political impact from the time it was first announced. The originally described plan was rather broad and fuzzy, with some apparently grandiose objectives, and its announcement generated a lot of hype. It was some time before Fuchi's clear and far-sighted vision of future computer systems, in which logic programming would provide the central link between parallel architectures and knowledge processing applications, became widely understood. Many international developments were stimulated by the perceived "threat" of the FGCS project, including Alvey in the U.K. and MCC in the U.S.A. Other international developments were more directly inspired by the scientific vision of the project, and included the setting up of institutes such as SICS in Sweden and ECRC in Europe which were very much overseas counterparts of ICOT with very similar research directions.

Overall, the project has had a major scientific impact, in furthering knowledge throughout the world of how to build advanced computing systems. It certainly provided a tremendous boost to research in logic programming. In a real sense, FGCS has become an international research effort. This clearly has enhanced Japan's international prestige. The project has also led to Japanese researchers becoming far more "plugged in" to the international research community than they were at the time of the project's announcement. A further general benefit of the project to Japan must surely be the transfer to Japanese industry of research ethos and experience, provided by staff returning to their home companies after their three-year assignments to ICOT.

Organisational Issues

The project appears to have been handicapped, in tackling its very ambitious research goals, by being set in a framework more suited to an industrial development project. Ten years of basic research cannot be tightly laid down in advance, as much of ICOT's programme seems to have been, with its predetermined duration, phases, milestones and hardware deliverables. The inflexibility of ICOT's programme seems to have prevented the possibility of changes of direction and reevaluation that are necessary in an advanced research project.

The research leaders of ICOT are scientists of very high calibre much respected by their international colleagues. Most of them have been with the project for its duration and have provided continuity of direction. Most of the other ICOT staff have been working on three-year assignments from the companies. The resulting lack of long-term continuity of ICOT staff, and the fact, as I understand it, that ICOT could not hand-pick the majority of its staff, are additional handicaps to pursuing advanced research not shared by comparable institutions such as ECRC, MCC and SICS.

Major Technical Achievements

The FGCS project has produced many significant technical achievements. Some of the major accomplishments which are of particular interest to me and which I would highlight are the following.

First, ICOT has achieved its foremost concrete objective of building a parallel inference machine with a performance exceeding 100 megalips. Given the state of the art at the time the project was announced, when Prolog performance was at best 40,000 lips and large-scale parallel machines hardly existed, this achievement is quite remarkable and should not be underestimated. Although KL1 lips are not quite as powerful as Prolog lips, ICOT's achievement still represents a leap forward by more than three orders of magnitude.

On the language side, I consider GHC to be a most significant contribution. It embodies, in my opinion, the most elegant encapsulation of the committed-choice language (CCL) concept, simplifying and clarifying what was introduced by Parlog and Concurrent Prolog.

In its parallel implementations of KL1, ICOT has significantly advanced the implementation technology for CCLs. My own group has drawn on this work, and on the key idea of GHC, in our implementation of Andorra-I.

Although I have some reservations about that ICOT has committed itself entirely to CCLs and the concurrent logic paradigm, it cannot be denied that ICOT's PIM machine and operating system PIMOS are a powerful demonstration of what is possible in terms of building a machine and operating system entirely based on a CCL. It strikes me as something of an heroic feat, akin to climbing Everest or putting a man on the Moon, which opens our minds to future possibilities while perhaps not bringing immediate economic benefit.

As part of its programme for producing demonstrations of KL1 and PIM, ICOT has created a number of innovative parallel symbolic applications, notably in the areas of VLSI CAD, molecular biology, natural language analysis, and theorem proving. For me, they are particularly interesting in showing the potential for parallelism in algorithms very different from the kind of regular and repetitive numeric computations which are typical of parallel computing today.

Technical Issues

There are some specific technical issues on which I would criticise the approach taken by ICOT. While I can appreciate some of the reasons why ICOT took the path it did, I feel the project might have achieved more, and remained closer to its original vision, if certain key decisions had been made differently.

Perhaps the most important issue is the decision (or assumption?) that parallelism has to be expressed explicitly in user programs, rather than designing systems to exploit parallelism automatically (taking advantage of the fact that logic programming, as a declarative formalism, allows parallelism to be expressed implicitly). Requiring the user to take direct responsibility for expressing parallel algorithms adds greatly to the programming burden, especially for the kind of complex knowledge processing applications which are the main target of FGCS. This route is only appropriate for problems which are computationally very intensive and where adequate performance cannot be achieved by other means. But for such problems, the first priority before tackling parallelism is probably to ensure that the sequential algorithm is as fast as it possibly can be, using as low-level a language as is necessary. This tends to argue against using a high-level approach such as logic programming.

On the other hand, there are many problems which may be potentially speeded up by exploiting implicit parallelism automatically, and where logic programming may provide reasonable performance (perhaps via the parallelism) in relation to software development cost. If parallel computers become the norm, as seems technologically inevitable in the near future, software systems which can exploit parallelism automatically will have a major role to play. It is a pity ICOT didn't take the opportunity to pursue this direction, which is being actively explored by other research groups (including my own).

The decision to go for explicit parallelism was linked with the decision to adopt the concurrent logic programming paradigm as central to all aspects of the project. In particular, all user programs in practice have to be expressed in, or implemented via, the concurrent LP paradigm, by means of the kernel language KL1. While the concurrent LP paradigm is of considerable interest in its potential for formalising interactive systems, and may be appropriate for many purposes including implementing operating systems, it is not, in my opinion, suitable for most user programs.

For most user programs, a much more high-level approach is needed, and ideally one would like to use declarative logic programming, i.e. logic programming as it was originally conceived. In declarative logic programming, the program expresses a declarative view of the problem as well as providing an operational solution to the problem. By contrast, the concurrent LP paradigm provides no declarative view of the problem. At best, it can be said to consist of a declarative description of a concurrent algorithm for solving the problem. In practice, users of the paradigm take an exclusively operational view. Without the declarative underpinning, there is no particular reason to maintain the original connection with logic, and every reason to modify the formalism to make it better fit its operational purpose. For these reasons, it is arguable whether concurrent LP is indeed logic programming in its original sense.

Be that as it may, the present situation with ICOT systems is that the main user language, KL1, is considerably lower level than traditional logic programming languages such as Prolog. Other, more high-level, user languages have been provided, but have had to be implemented on top of KL1. Although ICOT believes the use of KL1 as an intermediate language does not entail any unacceptable overhead, there seems good reason to believe that higher level languages and inference systems (including Prolog for example) could be implemented much more efficiently if a lower level implementation language than KL1 were used. In my view, KL1 is too low-level as a user language for most purposes, but too high-level to serve as the lowest level implementation language.

For a kernel language based on logic programming to be acceptable as a general user language it must, in my view, provide at least the basic capabilities of Prolog. This certainly seemed to be the view in the original FGCS proposal and in the early stages of ICOT's work. KL1, however, is considerably weaker than Prolog in that it does not provide a builtin search mechanism for finding at least one (and possibly all) solutions to a problem, although it is more powerful than Prolog in that it provides builtin coroutining (necessary, amongst other things, to support the concurrent LP paradigm).

It should be noted that it would be possible to have a kernel language providing all the capabilities of Prolog together with all the essential features of KL1 (including at least all of flat GHC which is the heart of KL1). Such a language would be quite acceptable as a user language, while providing the necessary basis to implement an operating system according to the ICOT approach. Such a language is provided by the Andorra-I system implemented by my group at Bristol. This language is viewed primarily as a high-level extension of Prolog. However, since it includes flat GHC as a subset, it is capable of supporting the concurrent LP paradigm.

Another most important issue, of a completely different nature, is the question of whether ICOT was wise to concentrate so much effort on building specialised hardware for logic programming, as opposed to building, or using off the shelf, more general purpose hardware not targeted at any particular language or programming paradigm. The problem with designing specialised experimental hardware is that any performance advantage that can be gained is likely to be rapidly overtaken by the ever continuing rapid advance of commercially available machines, both sequentiual and parallel. ICOT's PSI machines are now equalled if not bettered for Prolog and CCL performance by advanced RISC processors. And it seems very possible that commercial multiprocessors such as Sequent Symmetry, the new Butterfly, and other recent machines could come close to equaling the PIM performance if ICOT's software technology were ported to those machines.

A subsidiary issue is whether it was necessary to target KL1 so much at distributed memory hardware, with all the attendant problems of achieving good locality of communication and good load balancing, rather than adopting a virtual shared memory approach, for which scalable solutions are becoming increasingly well developed, including ones supporting a quasi-UMA (uniform memory access) model (c.f. KSR-1 and the closely similar work on DDM that I have been involved in). In general, I feel that ICOT perhaps devoted too great a proportion of its effort to developing hardware and operating systems, and could perhaps have focussed its efforts more on the knowledge processing software and applications which were central to the original conception of the project.

This section of my report is rather long! Its length should be interpreted not so much as a measure of criticism of ICOT's approach, which given the many constraints they were operating under has been very productive I believe, but rather as a measure of the complexity of the issues that I felt needed to be mentioned.

Overall Evaluation

The nature of the original FGCS announcement raised a lot of expectations that the project could never have satisfied and certainly have not been satisfied. Unfortunately, this makes it difficult for the project to be judged a success by the world at large, which includes most of the media. However, I strongly believe that overall the project has been a considerable success, and I think most fair-minded and properly informed observers will share my view.

The project was a major success in galvanising worldwide activity and more importantly for its scientific impact in stimulating worldwide research in new directions inspired directly by the FGCS vision and ICOT's work. The project has also succeeded in achieving its main concrete target of 100 megalips plus, an outstanding accomplishment that shouldn't be diminished with the benefit of hindsight.

But above all, any research project such as FGCS should be judged in comparison with comparable efforts by comparable institutions elsewhere. I believe the specific research and development achievements of ICOT are on a par with the three institutions, MCC, ECRC and SICS, which are most comparable with ICOT and which are representative of the very highest level of computing research in the world. Moreover it should not be forgotten that those three institutions came into being largely following in the footsteps of ICOT and the FGCS project.

Recommended Future Steps

I strongly recommend that ICOT's work should be continued in some form beyond the 1993 official end date of the FGCS project. The nucleus of highly gifted people and expertise built up at ICOT should not be allowed to evaporate, but should be continued within a smaller and more flexible framework. The KL1 software should be made available on widely available standard hardware, including Unix uniprocessors and multiprocessors such as Sequent Symmetry and perhaps BBN Butterfly. The PIM hardware should be examined to see whether it might potentially form the basis for commercial products if standard languages and operating system were supported. More effort should be put into evaluating the FGCS results, and especially in comparing the performance and usability with the best conventional alternatives. Speedups and good load balancing are not enough by themselves; one needs to show that applications perform better than they would by other approaches with comparable implementation effort. There should also be continuing research, especially in the areas of knowledge processing and applications. I would suggest that all this would best be done within a much smaller research institute, with selected long-term staff, and a focussed but flexible ongoing research programme (c.f. for example SICS).

It is understood that MITI is anxious to have official overseas collaboration in any extension of the FGCS work. My own group would be interested in collaborating with ICOT (or its successor) in evaluating ICOT's parallel applications developed in KL1, to see to what extent the same problems can be solved through more directly declarative logic programs, and whether comparable performance and parallelism can be obtained from logic programming implementations supporting implicit parallelism (such as Andorra I). Unfortunately, DTI (the UK counterpart of MITI) requires 50% funding from UK industry for any research it supports. So long as the ICOT work is only available on custom hardware, it is unlikely that UK industry would be interested. And even if the ICOT software were ported to standard hardware, the likely payoff from such research is too long-term for most UK industry (with its rather short-term horizons). Therefore, I am afraid the chances of official UK involvement, through DTI, in continuations of the FGCS work seem poor, for the near term at least.

Evaluation of FGCS Project

David H.D. Warren
University of Bristol

1. General

- Major political impact
  (plan fuzzy, grandiose, hyped)
  – Alvey, MCC, ...
- Major scientific impact
  (Fuchi's vision: FGCS= PP-LP-KP)
  – ECRC, SICS, ...
  → international FGCS effort
- Boosted LP
- Plugged Japan more into international research community
- Transfer of research ethos to Japanese industry (?)

2. Organisational

- Project overplanned, inflexible, too large
  – attacking v. ambitious research goals in industrial development framework
- Research leaders v. high calibre
- Personnel not selected by ICOT ?
- Staff on 3-year assignments

3. Some Major Technical Achievements

- Achieved > 100 Mlips (but rps ≠ lips)
- GHC — most elegant CCL
- Implementation techniques for CCLs
- Demonstration of what it means to base a complete OS and parallel machine around a CCL
- Innovative parallel symbolic applications
- Many other valuable contributions to LP research

4. Technical Issues / Criticisms

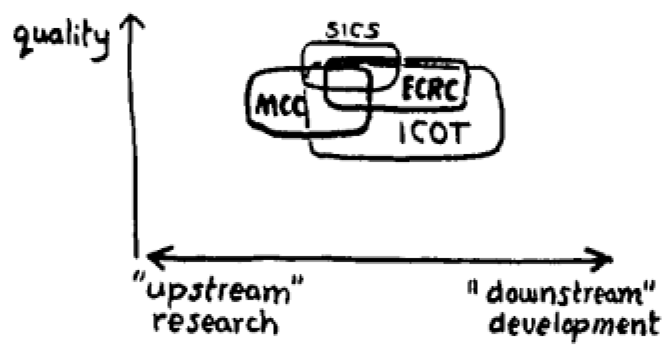
- •• Explicit versus implicit parallelism
- • Concurrent LP paradigm versus declarative LP paradigm
- • KL1 < Prolog

  versus KL ≥ Prolog (e.g. Andorra-I = Prolog + FGHC)
- •• Specialised versus general-purpose hardware
- • Distributed memory versus virtual shared memory (scalable, UMA) e.g. DDM, KSR
- • ? Too much effort concentrated on hardware and OS versus knowledge processing & applic^ns

5. Is FGCS a Success?

- • In satisfying expectations raised by FGCS announcement: no
- • In international impact: (resounding) yes
- • In achieving major target of > 100 Mlips: yes (but)
- •• In achievements on a par with institutions comparable with ICOT (MCC, ECRC, SICS): yes

Very subjective rating of R&D output



6. Conclusion

- • FGCS has been a success
- • ICOT should continue in some form (smaller, more flexible)

Remarks at the FWorkshop Reception

Y. T. Chien

6/3/92

It is a great honor and pleasure to be asked by Dr. Fuchi to say a few words in front of such a distinguished group of scholars, government officials and representatives from the industry. In behalf of myself and all of the invited guests, I want to thank Dr. Fuchi and his staff for this splendid opportunity to come here and take part in this important event.

I of course also want to extend my congratulations to ICOT on their 10 years of accomplishments and contributions to knowledge processing and to a new generation of computer technologies. But even more importantly, it is ICOT's relentless pursuit of an ambitious vision and their determined efforts to share that vision and the fruits of their labor with the international community that has captured most of our admiration. In the United States, the computer research community has benefitted greatly from a number of collaborative arrangements under the joint sponsorship of ICOT and the National Science Foundation. Early on, we instituted a formal Visitors Program, which made it possible for the U.S. scientists to conduct long term research at ICOT. We have also held regular joint technical workshops and symposia and have seen many informal scientific visits and exchanges among the researchers from both countries.

I am aware that other countries - France, Great Britain, Germany, and Sweden for example - have had similar arrangements with ICOT and participated, perhaps even more successfully in some cases, in its activities over the years.

In a report of his recent visit to ICOT, Professor Woody Bledsoe of the University of Texas pointed out that one of ICOT's valuable assets is its willingness to listen to and make use of the scientific expertise from over the world. I believe that this asset is an important hallmark of this institution and it has indirectly contributed to the many successes that ICOT has enjoyed in the course of their 10-year project. As an NSF officer and a member of the scientific community, I wish to express our sincere appreciations to ICOT and to the architect behind it - the Ministry of International Trade and Industry, for their leadership, generous support and their commitment to the advancement of science on a global level.

With Dr. Fuchi's permission, I'd like now to ask my colleagues to join me in proposing a toast to our Japanese hosts and friends: We wish the 5th Generation Computer Project continued success and, to borrow Dr. Fuchi's own words, ICOT to take their 10 years of labor of love into the launching of a even more exciting era ahead. Best wishes from all of us and peace, health and prosperity to everyone.

Report on the Success of Japan's Fifth Generation Project

Philip Treleaven

I would like to take this opportunity to give my views on the success of the Japanese Fifth Generation Project. Rather than restrict my comments solely to the technologies developed, I would like to discuss the important contributions that the FGCS Project has made to Japanese industry and to the organization of research in Europe.

To give you some background, I have been an Advisers to Government Ministers in a number of countries on the organization of industrial research programs such as the FGCS Project. I am currently an Adviser on industrial policy to the European Commission. And in addition, I am the Chairman for the British Conservative Party for the Thames Valley region; Britain's silicon valley. I also had the great fortune to be at the launch of ICOT, to see the foundation of the FGCS Project and to work at ICOT for a short period.

The FGCS project has three major achievements:

1. It has given a major boost to the state of the art of software technologies in Japanese industry.
2. It has changed the way that Information Technology research is conducted in Europe and other countries.
3. The FGCS Project plan provided a "Road Map" for future computer research, one that remains still relevant today.

I will discuss these three achievements further.

My observation of the Japanese software industry in the early 1980s, when I attended the conference that launched the FGCS Project, was that it was a long way behind the excellent Japanese hardware industry and was way behind the American and European software industries. With the launch of the FGCS Project, Japanese companies immediately acknowledged the growing importance of software and especially of artificial intelligence techniques for robotics, image processing and knowledge-based systems. The launch therefore spurred Japanese companies to switch major resources to software. The result is that during the 1980s the Japanese software industry overtook the Europeans and made very significant progress in catching up with the Americans. In addition, throughout the pasted 10 years of its operation the FGCS Project through its committee structure and conferences has provided a major vehicle for information to be gathered from around the world and to be quickly disseminated to Japanese companies.

Outside of Japan, the major impact of the FGCS Project was to change the way that Government sponsored research in Information Technology, and also other areas, was organized. In Europe prior to the FGCS conference:

1. Industry and universities had very little contact even in a single country.
2. Industry often had little idea of new concepts and universities undertook very pure, often only theoretical, research.
3. There was almost no contact between companies and universities from different countries in Europe.
4. Most researchers looked to the United State for technical leadership.

The FGCS Conference had a major impact on the government delegations attending. The two main conclusions were firstly that the Japanese Gevernment had a fundamentally better way to organize industrial research and secondly that the FGCS Project could lead to Japan becoming the world leader in computer products. In response the European Commission established the ESPRIT Programme and individual governments set up national programs to coordinate and fund IT research. For example Britain established the ALVEY Programme. The FGCS Project had the following impact in Europe:

1. Pre-competitive, collaborative research programs became the standard method of organizing government funded research.
2. Compaines and unversities started to undertake applied research. In fact, the research is becoming increasingly applied, moving towards development of products, and working with large user companies.
3. Strong links were build between compaines and unversities.
4. In the European Community these research programme have built a single research community. UCL now works close with PHLIPS(N), SIEMENS (G) and THOMSON (F) than with any British company.

Notably a large number of similar collaborative research programs have been established especially by the European Commission. Therefore the effect of the FGCS Project on the way that research is organized in Europe cannot be over emphasized.

The third achievement of the FGCS Project is the actual Project Plan. I well remember reading an early draft of the Plan and feeling I had a "Road Map" setting out future computer research. When I have asked to identify what are the most important aspects of the FGCS Project and the way Japanese research is organized, I always refer to the putting together of the Plan as the most valuable part. I describe it as; bringing together the leading experts in Japan, then gathering all the information available from the best people world-wide, collating this information, building a consensus on the future, and then distributing the resulting plan, especially to Japanese industry. The FGCS Plan is as relevant today as it was in 1982. During the past year I have been reading the NIPT or Real-World Computing documents and I have been struck by how similar they were to the FGCS Plan. In fact, if one took the FGCS Plan and changed all the references from Logic to Neural Networks the two plans would be almost the identical.

It is easy today to dismiss the impact of the FGCS Project saying that it was a mistake to base it solely on Logic and that some of the estimates for breakthroughs in Speech Processing were over ambitious, but, this is to ignore the wider impact that the FGCS Project has had particularly on software research in Japan and on the way governments' organize collaborative research world-wide. We must remember that at the time the decision was made to base the project on Logic, there were no other alternative choice. For example, Neural Networks were a good seven years away.

Therefore I remain convinced that the FGCS Project has been major success, and is a credit to the Japanese Government.

16th June 1992

Professor Philip C. Treleaven
Department of Computer Science
University College London

日本の第五世代コンピュータ・プロジェクトの成功に関するレポート

Philip Treleaven

私は、日本の第五世代コンピュータ・プロジェクト(以下FGCSプロジェクト)が終了するこの機会に、その成功について、考えるところをお話ししたい。このレポートでは、FGCSプロジェクトで開発された技術の面からだけでなく、このプロジェクトが日本の産業及び欧州の研究組織に重要な貢献を果たしたことにまで触れたいと思う。

本題に入る前に、私のこれまでの略歴についてまず触れておきたい。まず、私は多くの国でFGCSプロジェクトのような研究開発プロジェクトについて、大臣の顧問を務めてきた。現在では、ECの産業政策顧問をしている。また、英国のシリコンバレーと呼ばれるThames Valley地方の英国保守党の議長もしている。さらに、私にとって幸運だったことには、ICOTの発足、FGCSプロジェクトの開始に立ち会い、短期間だがICOTの研究に加えさせていただいている。

さて、FGCSプロジェクトが果たした三つの大きな業績をあげると次の通りである。

1. 日本のソフトウェア技術の水準を大きく引き上げた。

2. 欧州やそれ以外の国での情報技術研究の実施方法を変えた。

3. 未来のコンピュータの研究のための、今日においてすら適切な「道案内図」となった。

これらの三つの業績についてもう少し詳しく述べることとする。

私がちょうどこのプロジェクトの第1回目の国際会議に参加した1980年代初頭は、日本のソフトウェア産業は日本の優れたハードウェア技術に比べ大きく遅れており、欧米のソフトウェア産業に水をあけられていた頃であった。それがFGCSプロジェクトの開始によって、日本の企業はソフトウェアの重要性が高まっていることを認識するようになった。こうした動きは、特にロボティクス、画像処理、知識処理システムのためのAI技術において顕著であった。すなわち、FGCSプロジェクトの立ち上げが日本の企業をソフトウェアへリソースの投入を行うことを踏み切らせたのである。その結果、1980年代において日本のソフトウェア産業はヨーロッパを追い越し、アメリカに追いつく上での重要な発展を遂げている。さらにつけ加えれば、過去10年間、FGCSプロジェクトの組織や国際会議は、世界の重要な情報を収集し、それをただちに日本の企業に広める役割を果たしてきたのである。

日本国外でのFGCSプロジェクトのインパクトは、情報技術を始めとする分野の政府の出資する研究の実施方法を変えたことである。FGCS国際会議以前の欧州の状況は以下のような状況であった。

1. 産業と大学の間の交流がたとえ同じ国内であってもほとんどなかった。

2. 産業には新しいコンセプトはほとんどなく、大学における研究は本当に 純粋な理論研究しかしていなかった。

3. 同じ欧州の中でも、国が違えば企業と大学の交流はほとんどなかった。

4. たいていの研究者は技術面のリーダーシップは、アメリカをあてに していた。

FGCS国際会議は、出席した各国の代表に多大なインパクトを与えた。彼らの下した２つの結論は、第１には、日本の政府が産業技術研究を組織するよりよい方法をとっているということ、第２には、FGCSプロジェクトが日本を世界のコンピュータ分野のリーダーにできるであろうというものであった。このプロジェクトに呼応する形で、ECではESPRITプロジェクトが始まり、各国が同様の国家プロジェクトを打ち立て、情報技術の研究に資金を供出するようになった。例えば、イギリスはALVEYプロジェクトを立ち上げている。さらに、FGCSプロジェクトは欧州に以下のようなインパクトを与えた。

1. precompetitiveな共同研究開発が政府の出資するプロジェクトの標準となった。

2. 実際、研究はますます応用指向を強めており、製品開発に向かって研究成果を利用する大企業と共同で行われている。

3. 企業と大学の強い絆が作られた。

4. ECでは、このような研究プログラムにより研究コミュニテイが作られた。ロンドン大学は、現在、英国国内の企業よりもPHILIPS(蘭)、SIEMENS(独)、THOMSON(仏)といった海外の企業と密接な関係にある。

さらに、特にECの手によって同様の共同研究プログラムが数多く立ち上げられることとなった。

このように、FGCSプロジェクトの欧州における研究の組織作りの手法に関する業績は、いくら強調してもし過ぎることはないのである。

FGCSプロジェクトの第３の成果は、このプロジェクトの計画そのものである。私は初期の計画案を読んだ時のこと、また、その時に未来のコンピュータの研究のための道案内図を手にしているかのように感じたことを未だに覚えている。

私はFGCSプロジェクトの最も重要な面は何かとか、日本が研究を組織立てる際の最も重要な点は何かと問われた場合、計画の立て方自体がもっとも価値があるものだと答えている。すなわち、日本の指導的立場にある主要な専門家を集め、世界の最も優秀な人達から利用可能な情報をできる限り集め、この情報を付き合わせ、未来像のコンセンサスを打ち立て、その計画を広めた、特に日本の産業界に広めたことに価値があるの

X

だと。

今、論理だけに基礎を置こうとしたことが誤りであったとか、談話処理のブレークスルーへの期待は野心的であったといってFGCSプロジェクトの与えたインパクトを評価しないのは簡単である。しかし、これはFGCSプロジェクトが特に日本のソフトウェア研究、共同研究を政府が組織するやり方に世界規模で大きなインパクトを与えたことを無視している。我々は、このプロジェクトがロジックをベースにすると決定された時、別の選択がなかったことを思い出さなければならない。例えばニューラルネットなどは、その後7年経ってから良くなった技術である。

以上のようなことから、私はFGCSプロジェクトが大いなる成功を収め、日本の誉れとすべきところであると、以然として確信しているのである。

以上

1992年6月16日

ロンドン大学計算機科学部教授
Philip Treleaven