

Looking Back for Future

SS30年の歩みと
ソフトウェアのこれから

岸田 孝一

Kiss cedar, Call witch!

SS2010 in 横浜

Closing Keynote

われわれはどこから来たのか？
われわれは何者か？
われわれはどこへ行くのか？

ポール・ゴーギャン



Encounter with Machines in 1950s

Tiger Calculator

大学の天文学教室で天体力学の計算に使った



承前

UNIVAC120

あるメーカーに就職した小学校時代の同級生のために
英語で書かれた保守マニュアルを翻訳した



承前

岩波講座「現代応用数学」

このアカデミックな講座に似合わない分冊を発見した:

「パンチカード計算機械」

筆者は東大教授・森口繁一.

わたしが訳した保守マニュアルとほぼ同じ内容でしかなかった.

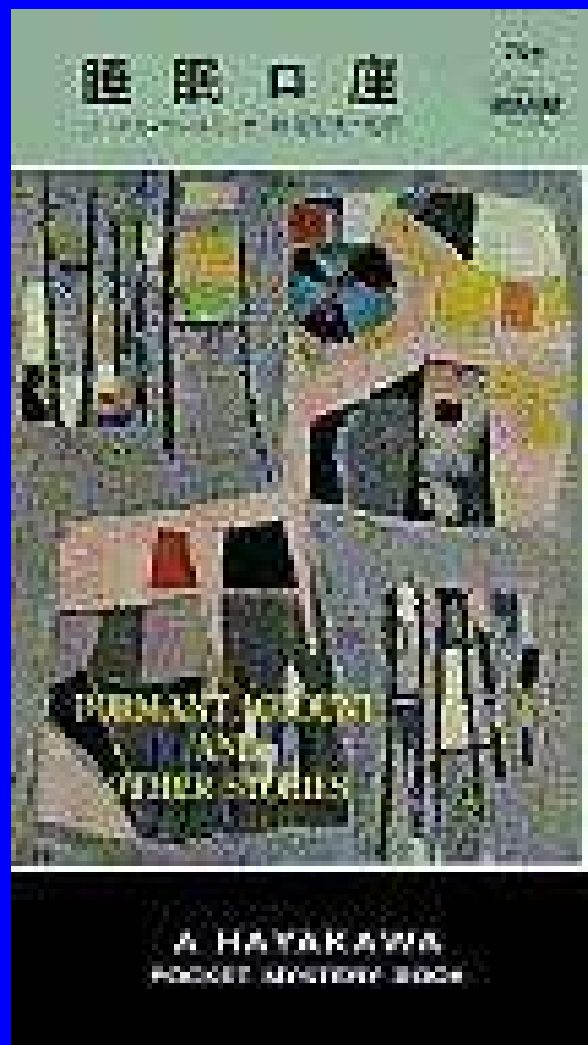
なぜ, こんなものがここに?

Memoir of 1960s

- コンピューティングの世界への扉が思いがけずに開いた
- きっかけは：
 - 都立日比谷図書館開架閲覧室
 - 日比谷公園のベンチで拾った1枚の新聞
 - 日活アクション映画

Gate to Computing World

@ the Library in Hibiya Park



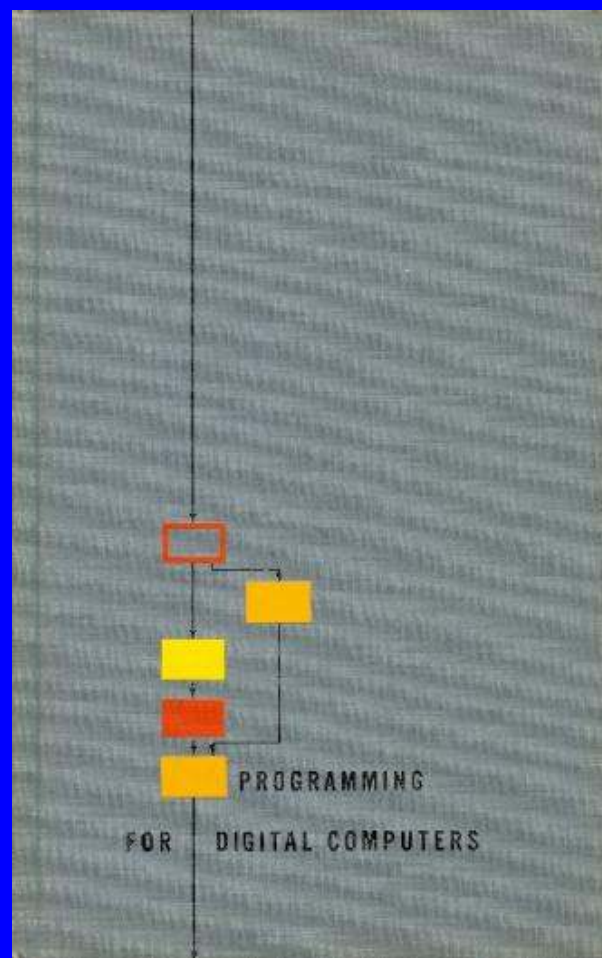
ニューヨークのセントラルパークのベンチに落ちていた新聞の広告欄から話が始まるミステリ。それを真似て日比谷公園のベンチで拾った新聞で見つけた翻訳者募集の広告に応募した。相手は電子計算機関係のセミナー屋さんだった。

承前

そのセミナー屋さんは日活映画封切館と同じ経営で、毎週新しい映画が無料で見られた。



翻訳したのは次の本
Text Book by Joachim Jeenel



それは Columbia Univ に導入された最新鋭
の大型計算機のプログラミング教科書だった。



Freelance Programmer としての出発

- 大学を中退し、セミナー屋で知り合った経営学の先生をお手伝いすることから、わたしのプログラマ生活は始まった。いまから考えるとおかしなことがいろいろあった：
 - パンチカード vs 紙テープ の論争 ☺
 - パンチカードの孔のかたち 論争 ☺
 - 磁気ドラム主記憶との格闘
 - アドレス方式磁気テープとの格闘

Strangers in the New World of Computing

あの時代，新しいコンピュータの世界では，
だれもが異邦人だったように思える。



たとえば

- Edgar Dijkstra
 - 物理学専攻
- Michael Jackson
 - 古典哲学専攻
- Barry Boehm
 - 天文学専攻
- そして, わたし
 - 芸術理論専攻

人間の成長モデル

オルテガ・イ・ガセット

ひとは人間として生まれてくるわけではない
(生まれつきのプログラマなどは存在しない)

People



You



Myself

われわれの 原始コミュニティ

- Algorithm Page of CACM Magazine
- いまにして思えば Open Source/Free Software の始まりだったといえよう.
- 無名の People の中から, 名前のついた You を見つけることのできる場であった.

構造化の衝撃

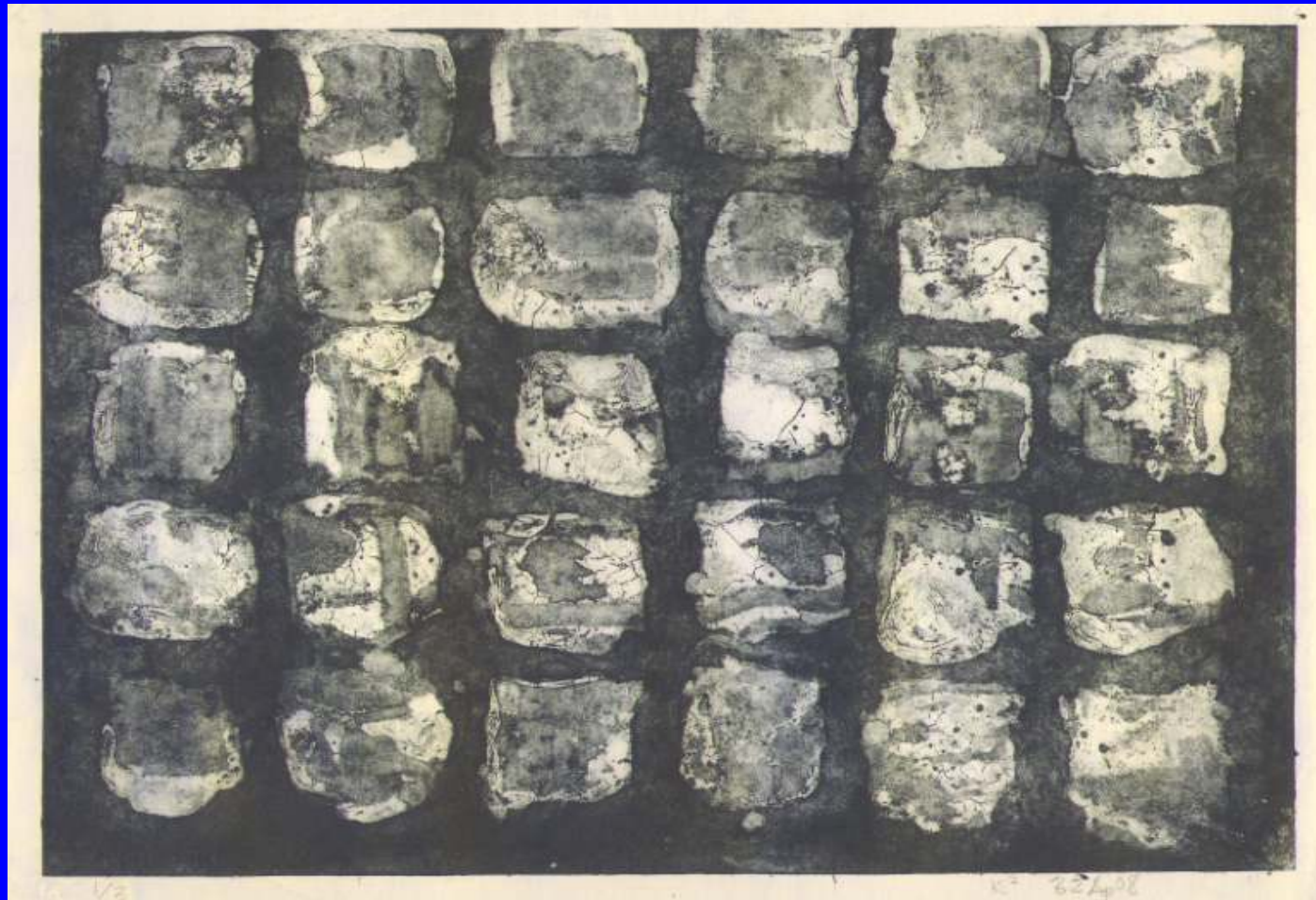
- Structured Theorem
 - 3人のイタリア人数学者による証明,
 - 詳細は CACM 1966
 - すべてのプログラムの構造は, 接続・分岐・反復の組み合わせに還元できる.

画家としての連想



わたしの構造化

@ 毎日現代美術展 1962



Goto Letter

Edgar Dijkstra (CACM 1967)

- ハードウェアの内側で起こっているプログラムの実行「プロセス」
- 紙の上に書かれたプログラムの「構造」をそれと一致させたい。
- プログラムの実行プロセスが人間の目で読めるように！
- だから GOTO 文は Harmful
- しかし、機能的モジュール化誤解された。

余談

4th ICSE in Munich の Closing Session が
始まる前, 「タバコの火を貸してくれ」と頼まれた
ときに Dijkstra 先生と交わした会話

k2: “Professor, I think that cigarette
smoking is more harmful than
GOTO statement”

Dijkstra: “Yes, I know. But it has
some other good effect. If I smoke
in this conference room, all crazy
Americans will go out immediately”

わたしの構造化設計

- 「標準コンピュータ・プログラミング」シリーズ
(日本生産性本部刊, 未完)
 - プログラム構造パターンの統一
 - Jeenel もどきの構造化フローチャート
- 「システム・プログラム入門」(日本経営出版会)
 - 日本語で書かれた初めての類書として売れた!
 - しかし, 狙いは構造統一技法の単なる応用例だったのだが,

Memoir of 1970s

わたしにとっての指導原理 -1

by Paul Klee

- 分析とは？
 - 化学における分析: 毒物を構成要素に分解して調べ, どれが毒性を持つのかを突き止める.
 - 芸術における分析: 作品を構成要素に分解して特性を調べるのではなく, その作品が創造されたプロセスを探る.

Memoir of 1970s

わたしにとっての指導原理 - 2

Also by Paul Klee

美術は、目に見えるものをそのまま再現するのではなく、見えないものを見るようにするのだ。(その意味で)グラフィックアートは抽象に適している。

ソフトウェア工学の誕生

- NATO Workshop
 - 1968 (ガルミッシュ), 1969 (ローマ)
- 関心の移動
 - 機械内部のプログラム実行プロセスから
 - 機械の外側でのソフトウェア開発プロセスへ
- しかし、お手本にすべき既存の工学はすべて製造工学でしかなかった。

Waterfall Model

- Winston Royce のプレゼンテーション
 - 1970 夏 @ WESCON
 - たまたまわたしはそこにいた.
- この論文は, ハードウェア製造のMaterial Process とソフトウェア開発のImmaterial Processとの違いを論じていたのだが,
- しかし Waterfall Model を推奨したものと誤解されてしまった.

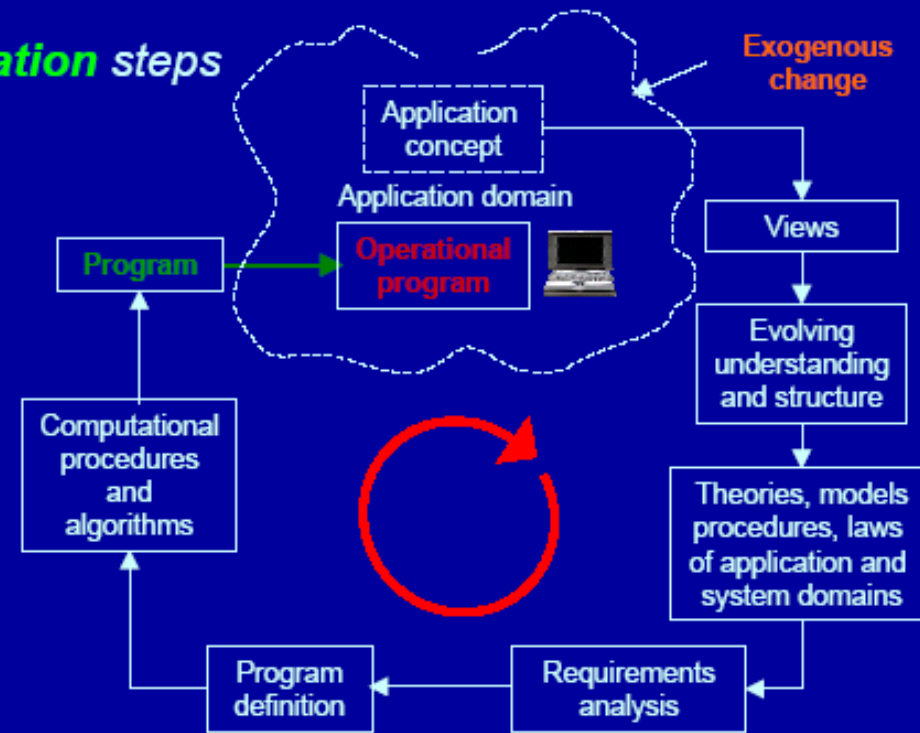
パラダイム・シフト

- プロダクト指向からプロセス指向へ
 - C.Floyd 女史の論文 (SEN, 1988)
- Waterfall の呪縛からの脱出
- ソフトウェアの開発および進化のプロセスは Multi-level, Multi-loop. Multi-agent の Feedback System である
 - M.M.Lehman のソフトウェア進化論
- だから Requirement は 永久に固まらない。

High Level View of E-type Process

Series of **development** and **implementation** steps

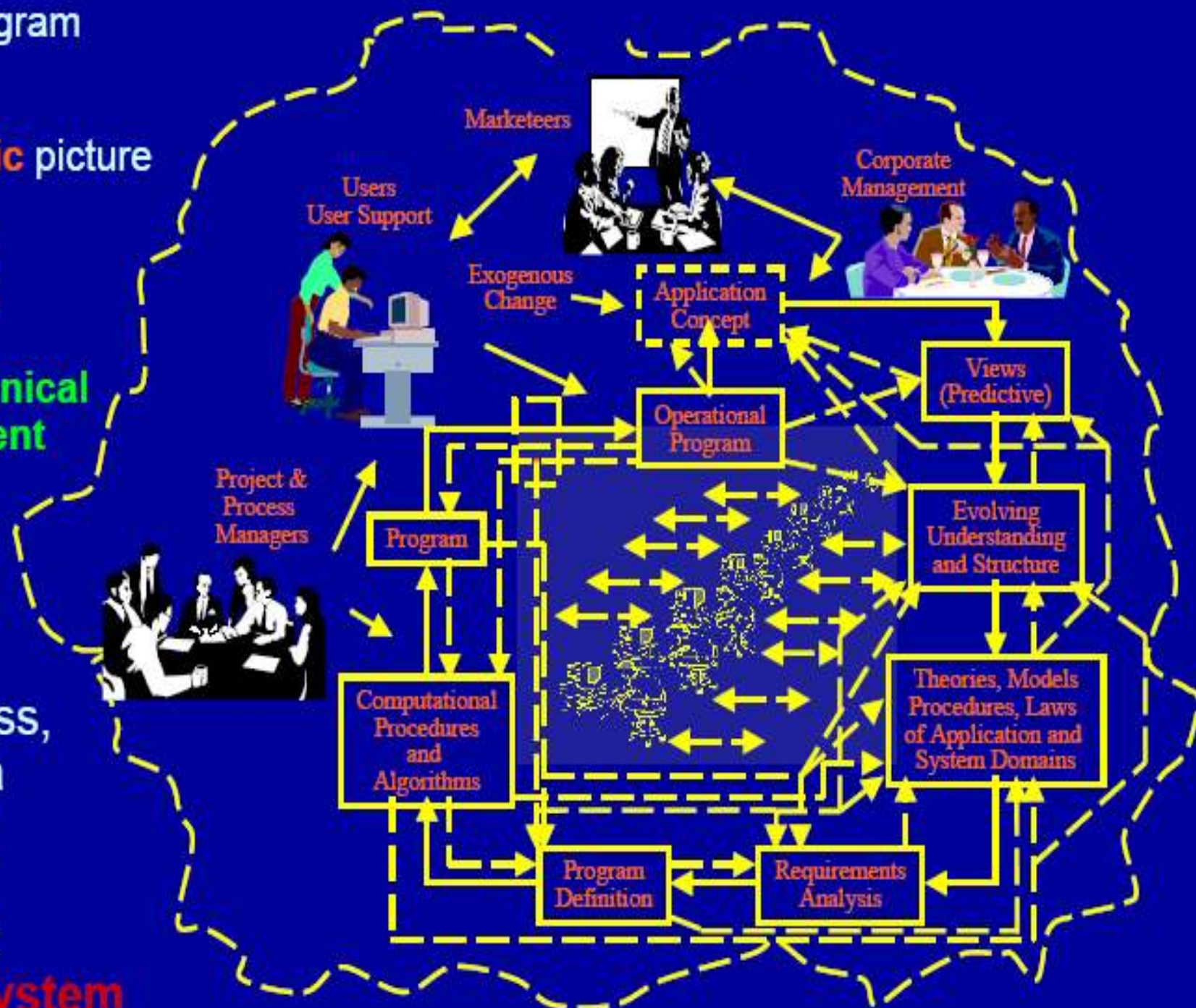
- Culmination: **validated program**
- Final step: **installation** which changes **domain** and **operation** which changes **application**
- All in **changing real world** domain
- **Installation** and introduction into **usage** closes major **feedback loop** that drives **iterative process** to **release** sequence of **upgrades** to users



Driver of **continuing** software **evolution**

More Realistic Picture

- Previous diagram a **fiction**
- More **realistic** picture
- Process **not sequential**
- Not just **technical development**



Global process,
in general, a
multi-level
multi-loop
multi-agent
feedback system

構造化の発展

- 構造化設計からオブジェクト指向へ
- 対象システムの構造をどうとらえるか？
- 機能のモジュール化？
- データのモジュール化？
- オブジェクトのモジュール化？

- 素朴客観主義手法の限界が露呈された！

パトナムの定理

(詳細は: ジョージ・レイコフ 「認知意味論」)

- 次の2つの命題は相互に矛盾する
 - 意味論は, 記号と世界の中の「モノ」との結びつき方を規定する
 - 意味論は, 記号の意味を規定する
- たとえば, ある命題に含まれる単語を取り替えても, その命題の真偽が変わらないという場合がありうる!

この定理が意味するもの

- 素朴な客観主義の認識論には矛盾のない形式化は存在しない.
- 世界の事物はわれわれとは独立に存在してはいない.
- 神の視点に立った現実の客観的に正しい記述などありえない.
- つまり, 自分が現実の外に立って, 外部の視点から現実をとらえることはできない.
- ソフトウェア開発者自身もまた, 現実(対象システム)の一部なのである.

時の流れ

- ゆくものはかくのごときか，昼夜を措かず。
— 黄河のほとりでの孔子の発言
- しかし，時の流れに河岸は存在しない。
- われわれは時の流れの中にいる。
- 流れの方向は？ そして速度は？
- 「時は流れない！」（大森荘蔵）

コトバというもの

- コミュニケーションの道具

しかし、それ以前に

- われわれが世界を認識する道具

コトバと意識

井筒俊彦「意味と本質」

意識はつねに「何かへの意識」である。
つまり、意識はそれが成立する以前に、本能的な対象の本質把握を必要としている。
たとえば、ここに一輪の「花」があるとして、われわれがそれを「花」として意識するとき、われわれの心の中には、すでに「花とは何か (花の本質)」についての理解が形成されているのだ。

詩人たちの格闘 -1

蝶鳥の 知らぬ花あり 秋の空

(松尾芭蕉)

詩人たちの格闘 -2

さえぎるな

言葉！

わたしと海との間を

(谷川俊太郎)

古代の知恵

- 一切の言説は仮名にして、実なく、ただ妄念に随えるのみ....
- (すべての言語表現はただ仮に立てられた名前にすぎない。別にそれぞれの名前に対応する「実」すなわち本質があるわけではない。ただ、われわれの意識または無意識の働きによって、いろいろなコトバが浮かんで消えしているだけなのだ....)

大乘起信論

System ということば

- Sy: 「一緒に」
- Stem (Histanai): 「モノを置く」
- 一緒に置かれたモノたちのあいだに、われわれはどんな「関係」を認識するのか？
- 量子力学における「観測者」の問題と同じ状況がそこにある。

Virtual は「仮想」ではない

- 新英和中辞典 第6版（研究社）
 - **vir·tu·al**（ラテン語「力のある」の意）
 - 1 (表面または名目上はそうでないが)事実上の, 実質上の, 実際(上)の.
 - 2【光】虚像の
- 画家たちは, すでに, かれらの目に映るものがほんとうの世界ではないことを知っていた.

SIA/STC

ソフト協技術委員会

- 1974年に発足
- 1984年, SIA が JISA に合併され, 以降は JISA/STC に.
- 業界団体の委員会であると同時に, 技術者個人がボランティアで参加・活動するというユニークな運営.
- 国際化を意識し, 1975年に始まったICSEに積極的に参加し, 6th ICSE の東京への招致に裏方として働いた.

SS started in 1980

- 1～4回は SIA/STC 主催
- 5回目はJISA主催
- 1985年12月にSEA設立
- 6回目以降は SEA 主催
- 第10回(1990 in 京都)以降, 地方都市を巡歴

SS の招待講演者たち

- 国際的な技術潮流との同期をとる
T.Gilb, M.M.Lehman, T.Demarco,
N.Haberman, R.Balzer, L.Osterweil,
W.Scacchi, 唐稚松, G.Fischer, L.Belady,
P.Zimmerman,
- 産学連携を目指す
大野豊, 米沢明憲, 齊藤信男、石井宏, 徳田
英幸, 棟上昭男, 山崎利治, 村井純,

SIA/STC & SEA の活動

- 討論主体のWorkshopというコンセプトを日本に導入
 - ソフトウェア開発環境
 - ソフトウェア信頼性
 - ソフトウェア・プロセス
 - 若手技術者の会
- SSをWG討論中心にモデル・チェンジ

Other Happenings

- Unix 革命
- Junet, IJ, そしてインターネット
- SIGMA プロジェクトの空騒ぎ
- 事件としてのCMM

SEA そして SS が 目指してきたもの

- さまざまな思想や意見が自由に交換されるオープンな「場」あるいはイベントを創出すること.
- ミハイル・バフチンが「フランソワ・ラブレール論」で提起した「カーニヴァル」の概念.
- 「舞台」と「観衆」の差別を取り払うこと.

ミハイル・バフチンに学ぶ

- ミハイル・ミハイロビッチ・バフチン

ロシアの思想家, 文芸理論家 (1895-1975)

20世紀が生んだ最高の思想家の1人

- 「ドストエフスキーの詩学」 (ちくま学芸文庫)
- 「ミハイル・バフチンの時空」 (せりか書房)
- 「ミハイル・バフチンの世界」 (せりか書房)
- 英国シェフィールド大学にあるバフチン研究センターの Web Page:

<http://www.shef.ac.uk/bakhtin/>

バフチンは語る

問い： ドストエフスキーの創作の基本思想とは？

答え： 世界の究極の問題である真理は一個人の意識の中では解明できないと、かれは考えていました。

真理は常に部分的にですが、多くの対等な意識が対話的に交流するプロセスの中で解明されて行きます。

この対話は、真理について考え、真理を追究する人間が存在する限り、終わることも完結することはありません。

対話の終わりは人類の滅亡を意味します。

多声的文学

バフチンのドストエフスキー論

- ドストエフスキーの創作の作品は世界で最初の「ポリフォニー文学」であった。多彩な登場人物の間で行われる対話が作品の中核をなしているが、すべての登場人物は平等に扱われ、作者はだれにも肩入れしない。小説は終わるが、対話は読者の意識の中で、そのまま続けられる。
- この「対話性」および「非完結性」がかれの作品のユニークな特徴である。

多声的プロジェクト そして 多声的コミュニティ

- ソフトウェア・プロジェクトやソフトウェア・コミュニティも、同様に、「対話的」であり、また「非完結的」であらねばならない。
- プロジェクトはいつか終わるが、そこで行われた対話や議論は、そのままそれぞれのメンバーの意識の中で続けられて行く。

結びに代えて

バナッハ=タルスキーのパラドクス

- 小さな円を分解し、再構成してより大きな円を作ることができるか？

No!

- 小さな球体を分解し、それを再構成してより大きな球体を作ることができるか？

数学的には Yes!

ひとつの夢想

- バナッハ=タルスキーの定理が数学の世界で成り立つひとつの理由は、3次元の形の「体積」が、数学的には厳密に定義されていない(定義できない)からである。
- SEA は4次元時空間の中に存在する組織であり、その「体積」は厳密には定義されていない(できない)。
- であれば、それをいくつかの小さな組織にいったん分解し、再構成してより大きな組織を創り出すことが可能かもしれない！？

Thank you!



Postscript-1

Slide14: オルテガといえば, まず思い出されるのは「大衆の叛逆」だろうが, わたしには「技術とは何か」のほうが印象的だった. この成長モデルは, かれの遺著「個人と社会」(未完に終わった最終講義のノート)に載っている.

Slide15: 60年代初期の Best-hit Algorithm は, Tony Hoare が発表したQuick Sort であった.

Slide17: 学生時代, 友人の卒業研究の手伝いで, Paul Klee のこの大著を日本語に訳した. Buhaus でのかれの講義には, 抽象絵画を構成するさまざまな「かたち」の構造に関する指摘がたくさん含まれており, 構造化定理はそれを思い出させた.

Slide18: この公募展でわたしはただの入選, 故・池田満寿夫が新人賞を獲得した.

Postscript-2

Slide19: 同じ「構造化」という名前で呼ばれていても, Dijkstra, Jackson, Warnier, そしてわたしが目指していたのは, それぞれ違う目標であった.

Slide22: 「構造」と「プロセス」, この2つの概念をどうとらえるかが, ソフトウェア工学の中心の課題であると思う.

Slide23: われわれが作るソフトウェアの本質は、ほんらい目に見えないものである.

Slide24: わたし自身が NATO Workshop のレポートを手にしたのは, 何年か後のことだった. 第2回のローマ会議では, Dijkstra が基調講演をしている. しかし, かれはその後ソフトウェア工学のコミュニティから離れてしまった. 1987年の第9回ICSE でHarlan Mills がそのことを嘆いていたのを思い出す.

Postscript-3

Slide25: M.Lazzarato の論文 “Immaterial Labour” のわれわれが注目し、それを主題に討論を始めたのはIWFST2007 in 杭州からだから、比較的最近のことである。この概念はもともと T.Negri の世界的ベストセラー「帝国」のなかで、政治的なニュアンスを含んで提起されたものだが、われわれはそれを技術・社会学的は観点から注目した。

Slide26: Floyd 女史の論文は 1988年春に読んだとき、かなり強い衝撃を受け、周囲の人たちに紹介したのだが、あまり共感は得られなかった。なぜだろうか。

Slide27-28: この2枚のスライドは M.M.Lehman 先生のホームページから借りてきたもの。先生の S-type/E-type というソフトウェアのカテゴリ分けを初めて聞いたのは1970年代末のことだった。

Postscript-4

Slide29: M.Jackson が、その構造化プログラミングのアイデアを一步先に進めるというアイデアを発表したのは 1978年春の第3回 ICSE だった。翌年秋の第4回 ICSE のパーティでかれと話をしたとき、「なかなか面白い」という感想を述べたのだが、内心では、ちょっと方向が間違っているのではないかと感じていた。

Slide30: Patnam 先生自身による解説は “Reason, Truth, and History”(邦訳「理性・真理・歴史 - 内的实在論の展開」, 法政大学出版社)。システム・エンジニア必読の書だと思う。

Slide31: 世の多くのシステム・エンジニアは「神の視点」に自らを擬することを好んでいるように見える。

Slide32: 大森荘蔵先生の「時は流れず」(青土社)は、わたしにとってまさに「目から鱗」の衝撃的な本であった。

Postscript-5

Slide33: コミュニケーション・ツールとしてのコトバの扱い方については、S.I.ハヤカワが書いた一般意味論の入門テキスト「思考と行動における言語」(岩波書店)が、ソフトウェア技術者必読の書だと考えていて、以前から勧めているのだが、あちこちのプロジェクトでコミュニケーションのトラブルが頻発しているのを見ると、あまり読まれていないようだ。

Slide34: 世界認識のツールとしてのコトバについて、もっともその本質をついているのは、井筒先生のこの本(岩波文庫)であろう。かつて、この本に刺激されて“ZOO – Zen and Object Orientation”と題するエッセイを書きかけたのだが、未完。

Slide 35 & 36: この2つの例は、野口武彦および三浦雅士両氏の指摘によるもの。

Postscript-6

Slide37: 井筒俊彦先生の遺著「意識の形而上学 『大乘起信論』の哲学」がお勧め.

Slide38: システム分析・設計における「観測者問題」は興味深い. 現代物理学とのアナロジーはどこまで通用するだろうか.

Slide39: 近代諸流派の画家たちは, 自らの眼を信じてそれぞれの Virtual World を描いてきた. わたしを含む現代の画家たちはそれに一步先へ歩みを進めている. しかし, それはソフトウェアの分野とは関係がない.

Slide40: SIA/STC 設立のきっかけは 1972年にアメリカ・ヨーロッパへの視察団で, あちらの業界団体の活動スタイルに刺激を受けたことであった. 当初は, 技術報告会やセミナーを実施. そのあと自主的な調査や技術討論の活動を開始した.

Postscript-7

Slide41: 最初の2回のSSは, STC における技術調査活動の剰余金を活用するかたちで運営された. プログラム構成は, 1970年代に始められた業界協同の技術開発の成果発表が中心であった. それらの共同開発は, 国(当時の通産省)からの補助で行われた. 補助金行政については何かと批判が多いが, 使いようによっては有用であることの1つの証明.

SEA 設立に際しては, 協会の理事さんたち(各社の社長さんたち)からかなり批判の声が聞かれた. 技術者たちが組織の壁を越えて恐竜することへの抵抗があったのだろう. そうした状況はいまでもあまり変わっていない.

Slide42: このリストを見ただけで, SS が日本のソフトウェア・コミュニティにおいて, きわめて先駆的な役割を果たしていたことがわかるであろう.

Postscript-8

Slide43: われわれが始めるまで、日本には Workshop スタイルの会合は存在しなかった。SS のような産学協同の技術シンポジウムもやはり存在しなかったのである。

Slide44: ここにリストされたさまざまな出来事については、すでにほかの場で語ってきたので、いまさら付け加えることはない。

1つの重要なプロジェクトをリストし忘れた。それは1980年代後半から90年代初めにかけて行われた国際・産学協同プロジェクト SDA である。わたしが SIGMA から脱退したことをきっかけに、いくつかのソフトウェア会社から集めた資金を利用して、ソフトウェア設計支援環境についての連続ワークショップ討論を続けるというかたちで行われたこのプロジェクトは、日本の学界の研究者たちそれまで国際ソフトウェア工学のコミュニティに引き入れ、後年の京都 ICSE 開催の基礎を築いた。

Postscript-9

Slide45: 以前, アメリカMCCの研究者が書いた「パフォーマンス・アートとしてのソフトウェア開発」というエッセイを翻訳して bit誌に載せたことがある. そこにあった舞台と観客を隔てる壁を取り払った「カーニヴァル」の概念は, たとえばオープンソース開発を議論するさいのフレームワークとして有用だと考えられる.

Slide46-49: Lazzarato の “Immaterial Labour” の最後の節に G.Simmel と並んで M.Bakhtin の業績についての紹介があった. 現在の社会状況の分析的記述としては, 1世紀前の Simmel の予見が当たっている. R.Florida の「Creative 階級」論はその延長線上にあるようだ. しかし, ソフトウェア開発という仕事の本質を考えるための概念装置としては, Bakhtin の「対話性」および「非完結性」のほうが適切だと感じる.