

人の意図的行動の知の構造とそのはたらき

河野 善彌[†] 陳 慧[‡] ハッサン アボールハッサニ[#]

[†] Creation Project 〒251-0875 藤沢市本藤沢 2-13-5

[‡] 国士舘大学情報科学センター 〒154-8515 東京都世田谷区 4-28-1

[#] Sharif University of Technology Azadi, Ave., Teheran, Iran.

E-mail: [†] koono@vesta.ocn.ne.jp, [‡] chen@kokushikan.ac.jp, [#] abolhassani@sharif.edu

あらまし 目的達成の為の行動を意図的行動と名付ける。肉体的から知的な行動まで全ての意図的行動は脳内で階層展開を繰返し詳細化される。それは自然言語能力により可能になる。この知の構造である階層展開連鎖、設計の product と process の直交する階層展開網、知の構造を用いたソフトウェアの自動設計、これらの定量的特性等の知の世界の議論を行っている。

キーワード 人間知能, 知識, 設計論, ソフトウェア工学, IE, Software process, Automatic design of software

Knowledge Structure of Human Intentional Activity and Its Works

Zenya KOONO[†], Hui CHEN[‡] and Hassan ABOLHASSANI[#]

[†] Creation Project, Hon-fujisawa 2-13-5, Fujisawa, Kanagawa, 251-0875, Japan

[‡] Center for Information Science, Kokushikan University, 4-28-1, Setagaya, Tokyo, 154-8515, Japan

[#] Sharif University of Technology, Azadi, Ave., Teheran, Iran

E-mail: [†] koono@vesta.ocn.ne.jp, [‡] chen@kokushikan.ac.jp, [#] abolhassani@sharif.edu

Abstract A human conduct, toward the implementation of the person's intention, is named an "intentional activity". From that of human mental level to human physical level, it is detailed by repeated hierarchical decompositions in the person's brain. It bases on natural language capability. This discusses on the knowledge structure by the hierarchical decomposition network model, the networks of product and process, accumulation of knowledge, automatic design of software and their quantitative characteristics are discussed.

Keyword Human intelligence, Knowledge, Design, Software engineering, IE, Software process, Automatic design of software.

1. はじめに

目的の実現にむけたヒトの行動を「意図的行動」と名付ける。本技報はこの行動を知の面で解明した報告である。

近年、脳の内部活動の観測手段が発達し、脳科学が急進展している。未だ環境や入力に対する脳の活動を知る段階ではあるが、それらとヒトの実行動の間のギャップを埋める橋作りはできよう。ヒトを制御する脳の機能の働き[1]、またその情報が支配する人間科学[2]が説かれている。本技報はこれらに続く「知の働き」への挑戦である。

経営の初動期の意思決定、これと対照的に最も具体的な肉体的動作、中間にあるソフトウェアの開発、これら全ては「意図的行動」と一括できる。これらには共通した知的処理がある。最上位概念を出発点として、親概念から階層展開して子概念群にすることを繰返し、最終的な実現手段(例えばソースコードや筋肉的動作指令)に詳細化する[3, 4]。この展開は自然言語の学習から獲得したヒトの知の構造である

2章は知の構造である階層展開網モデルとその特性を報告

する。展開過程は線形系で、作業の各特性を外部的に計測した外部特性である生産性や誤り率は理論モデルも実測値も一定性を示す。網モデルのノードの工数消費や誤り率が、生産性や誤り率を支配している。ヒトの作業の定量化である[5]。

3章は、知の構造である階層展開網モデル[3]とヒトの関わりである。意図達成過程は、意図達成の様相である product とそれを達成する過程 process の、二つの相互に直交的な展開網状をした知である[5]。これらの上に管理技術がある。

4章はソフトウェアの自動設計の研究結果を使う。ここでは知の集積につき定量評価する[6, 7, 8]。それにより各外部特性が指数的に向上する。

5章は、以上を纏めて将来を見た課題の提起である。

上記のように、本報はソフトウェア工学以外の方々にもご参考になると考え、判り易くする為、全体が見えるように努力した。最後尾に記した各分野の方々のご高見とご叱正を望みます。

2. ヒトの意図的行動

ヒトは多様な行動を行う。認知科学や脳科学では、このヒトの行動は外界の情報各種感覚器官を経て、符号化されて脳内で処理され記憶され、また各種運動器官を駆動して外界に出ると云う。意図的行動は各種行動中の基本で、その働きの中核部分は共通構造と思われる。

殆ど全ての設計は、「疎から細」の流れであり、仔細に見ると階層展開である。ソフトウェア設計では階層構造が各所に現れる。数種の意図的行動を図1に示す。全て単位的な展開にレベルを描いた。図aは戦争の計画段階、図bはプログラムの設計、図cは「写真を撮る」肉体的動作である。

軍の最終目的は、国の国益向上である。軍事科学では実験ができないから、過去事例から法則を導き出す。経験的法則「目的の階層性」は戦争計画の基本原則[9]とされ、図aはその1例である。最高司令官は、課された戦争の最終目的を実現するべく三軍を動かす方策（戦略）を決め、三軍の長に指示する。各軍司令官はその指示を更に具体化し部下に指令する。これを繰り返して得た計画で戦争に入る。

図1bはプログラム設計の1例[3]である。ここでは、データも機能同様に明記するデータフロー図を用いる。機能である「時計」の入出力のデータを規定し、単位データフロー(UDF)にする。UDF「時計」を引伸し、時刻と時刻表示で直列的に3分断すると(あるいは時計を階層展開すると)、詳細化データフローは3UDFの直列構成になる。プログラム実行の為、各機能の実行順序を示す小さなフローチャートが生じる。以降は図2に示した。次にUDF「時刻表示を求める」を階層展開するが、それには時刻と時刻表示の両データを並列的に3分断して、3並列のUDF群を得る。

プログラム言語表記が可能になると表記を乗換える。図の

最下段で「分針の角度を求める」は分の値の6倍であり次の展開ではFunsinKakudo = 6*Funの如くプログラム言語表記に移行する[4]。プログラムは実現手段で本質ではない。

図cは「写真を撮る」肉体的動作を3段階の動作に分けた。その一つ「カメラを構える」を更に展開すると、「狙い所を決める」「狙い所を画面中央に据える」「狙い所を中心に取り込む範囲を決める」等に詳細化される。

図1の3例のように、ヒトの意図的行動は、概念を階層展開による詳細化の繰返しで実現される。ヒトも動物も、幾つかの事象群を記憶し想起できる。ヒトはこの事象群に「言語による名称」を与えた。これで階層的な概念連鎖が可能になる。ヒトの記憶は高度化して正逆両方向に動作できた。ヒトの自然言語は次第にその句型/文法を高度化させた。これで階層展開を繰返せば意図的行動ができる。この知の構造は乳幼児の喃語から始まり、10才位迄の日常行動を通じて形成され、以後の教育に応じて更に高度な概念を備えて行くのであろう。

簡単な為、全て自然言語中心に考える。設計を扱う時の悩みは、その多様性にある。優れた高度能力設計者を「設計のモデル(規範的パターン)」にとる。彼等は、蛇にも似た巧みさで障壁を乗り越える。それは小進行単位毎に選択決定しながら前進できるから達成できる。以下を共通的条件とする。

- ・「ヒトの意図的行動」は概念展開を繰返す。
- ・親/上位概念から正確に下位概念に展開する
 - ・下位概念群は同レベルにして、均整を取る
- ・単位的な進行段階毎の決定を経て階層展開する
- ・詳細化し実現レベルに至ると手段に移行する

図2は階層展開連鎖の網状だから、図3[3]の網モデルで表せる。図の白丸は(概念である)情報、黒丸は階層展開する単位的処理である。設計は最上位から出発し一筆書きになぞりながら降る。展開を単位的毎に選んだから展開率は小さくなる。例えば図2の全ての展開率は3である。(3段階論法、ホップ・ステップ・ジャンプ、仏法僧と、辞書には3展開の例が多い。)図4は展開率の実績例で、プログラムと論理回路の

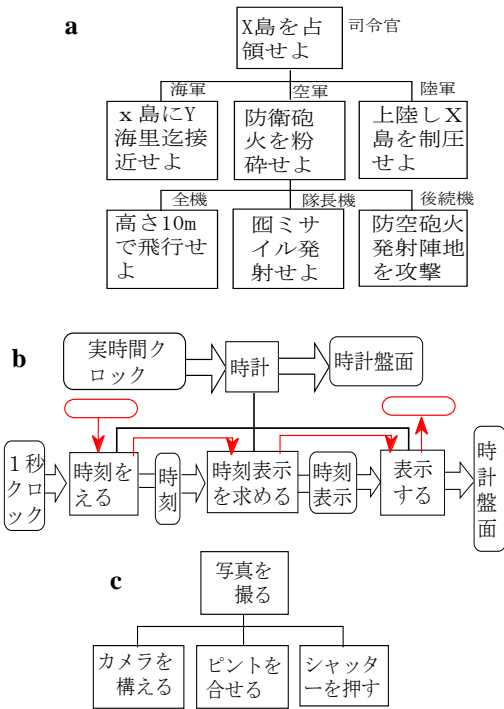


図1 各種の階層展開

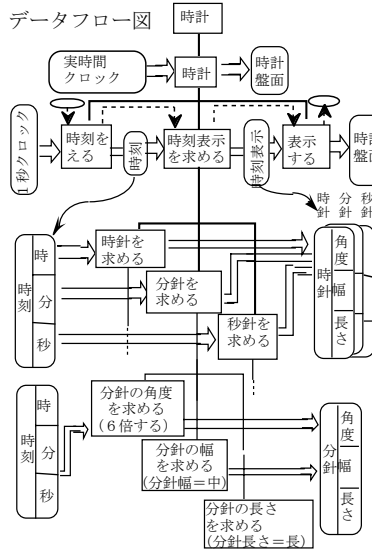


図2 時計プログラムの設計

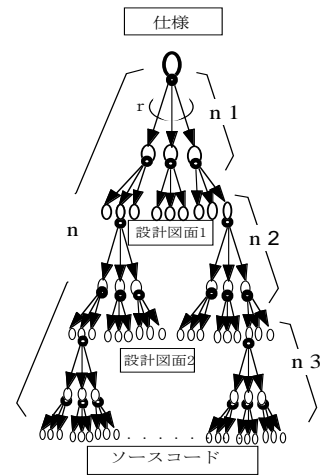


図3 階層展開連鎖網モデル

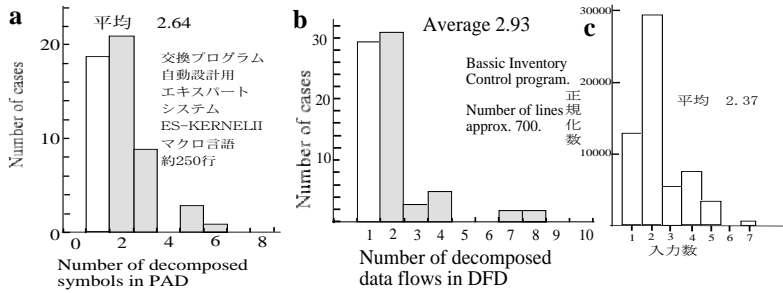


図4 各種の展開率

展開率の分布を示す。図aとbは、単位的な正しい展開に留意したプログラム設計の例[5]、図cは論理ゲートのFan-in数毎の使用数の統計である。平均値は3をやや下回る。

「何故3弱なのか?」を考える。纏めて記憶する「チャンク」が想起される。しかし、3弱はこの通常値より小さい。記憶の諸研究を調べると、時間を十分に与えると、より多数の記憶が再生できる実験があった。一方、高速記憶では想起される情報数は2程度で、更に小さい。ヒトの記憶のアクセスには時間が掛かるから、速度ネックで前記の中間の値になったと思われる。(ある条件では平均 = e ($e = 2.7183...$) が最適と証明できるが、全体は未だ証明できていない。)

平均展開率が一定なら、等比級数の計算で階層展開網を定量評価できる。図3の最上部の仕様は展開され具体化されて、最後には最下段の(ソースコードに対応する)出力になる。単位的知的処理に時間 τ が消費されると仮定すると、(消費時間の総和)/(最下段出力数) は生産性に対応する。単位的知的処理がある確率 ϵ で誤り、それは網を降って最下段の出力が誤ると考えると、(誤り総数)/(最下段出力数)は誤り(作り込み)率に対応する。n段の網の特性を等比級数の公式で計算して、 $n \rightarrow \infty$ とすれば、下記が得られる[3, 5]。

生産性の一定性 生産性 = 一定, 総工数 \propto 出力数
 誤り率の一定性 誤り率 = 一定, 誤総数 \propto 出力数
 厳密には何れも作込み数
 系は線形形で展開や統合ができる

ハードウェア産業で生産の管理を司る技術体系をIndustrial Engineering, IE, 経営工学と云う。ここでは、生産性の一定性を基本として、定量的な作業計画と従事者の能力格付け等を行う。誤り率の一定性は品質管理の基本である。3番目の展開統合は工程を最適化したり計測する為の基本である。これらは今迄経験則として運用してきた。前記は、この100年来運用してきたIEの基本経験則を理論証明したことになる。

ソフトウェアの実績で検証する。1970~80年代には広範な実績例が収集され公開されている。工数と出力数の関係はBoehmのCOCOMO資料[10]、また誤り数と出力数の関係はThayersの資料[11]を使い、両対数尺度でプロットして夫々図4aとbに示した[3, 5]。両図の打点群は、工数 \propto 出力数あるいは誤り数 \propto 出力数の傾向線であり、帯状に伸びている。図式的に勾配(即ち両者のべき乗関係の指数)はほぼ1だから、工数 \propto (出力数)¹と(誤り数) \propto (出力数)¹としてその中央傾向線を引いた。打点はこの中央傾向線の上下に帯状に分布する。その両限界線も中央傾向線に並行である。そこで、殆どの打点を

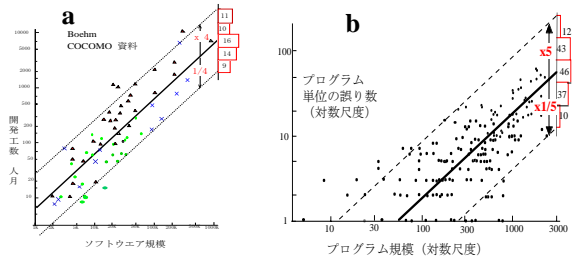


図5 作業過程の実績

帯状内に留めるように上下等間隔の限界線を引いた。次にこの帯領域を並行な5領域に等分割し、各領域毎の打点数を数えた。右上の水平棒グラフは5領域毎の度数分布である。

生産性(図a)も誤り率(図b)も、棒グラフ群はほぼ釣鐘状だから、実績は対数正規分布状である。完全な対数正規分布なら平均値 = 分散 σ で、平均値を中心にして 3σ の範囲に99.74%が含まれる。図5の限界線は、平均値を基準として、工数の場合には $\times 4 \sim 1/4$, 誤りの場合にはより大きく、 $\times 5 \sim \times 1/5$ の範囲に大部分が入る。(誤り数で大きくなるのは、設計後の机上チェックでの誤り摘出を管理しないから、作り込みと摘出の両者のランダム性が加算される為である。[5])

- 統計学では、多くの変数が相乗的に影響する場合の特性値は対数正規分布になると云う。(疫学や社会の統計等)
- 人間信頼性工学では人の単位的操作の誤り率HEPを計測し基準値を決める。多くのHEPは $\times 3 \sim 1/3$ のバラつきである。
- この分野の塩見は、人のKB操作について克明な実験を行ない、打鍵時間や誤り率が対数正規分布状と確認した[12]。
- ハードウェア製造作業では、上記と同様に、外部特性値は対数正規分布をなす事は常識である。

ハードウェア産業はこのバラつきを克服して高度製品を安定に製造している。ソフトウェア人には学ぶべき事が多々ある。

3. Product と process

多量にモノを作るには作業の管理を要する。「工程」はその為の手段である。工程の端面には設計文書や中間成果物を置き、管理、作業、責任を明確にする。早くから階層性を組合せて運用されprocessの技術として成長した。図6[5]の階層図は生産工程を示す。最上位の「生産」は開発と製造に大別される。開発は一過的な面が強く、製造はN回の繰返しが基本である。ソフトウェアは開発のみと誤解されるが、最終出力のN回

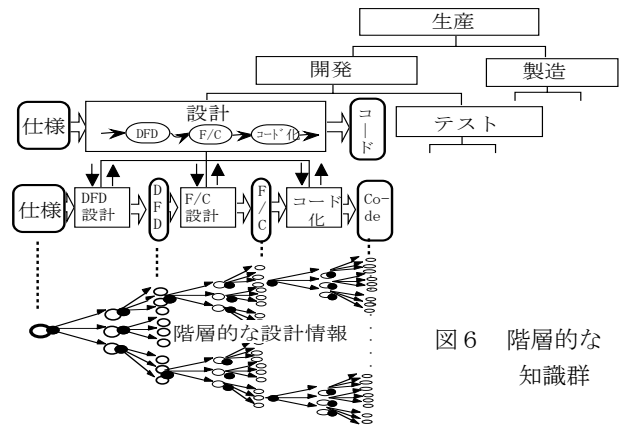


図6 階層的な知識群

繰返し、ファイルのコピー等の製造が存在する。

工程管理を知る為に図6の「設計」を見る。これは下位の状態遷移状に下位のDFD設計、FC設計と制御を移しす。各工程は動作をし、終了すれば上位の「設計」に報告し、「設計」は次の動作に入る。この指令/作業/報告のパターンは階層組織の全てにあり、確実に作業する仕掛けである。

図で更に降ると、作業は次第に具体化し、明確化し詳細化し、最後は人の単位的作業に帰着する。この様相は設計の場合と同じで、processは「作る」ことを最終目的とする概念の階層展開の連鎖になる。図6に示すように、最下底でprocessの階層展開網はproductの網を制御する。

図7[5]で工程により「分割し統治する」ことを説明する。[純化]統制する階層を下に降すと(工程は細分化され)、より多い中間点で拘束され、右図のようにバラつきが減る。[品質向上]上記で更に、工程の判定条件を厳しくする、環境/作業手順を厳しくする等を行う。計測すれば管理も向上。[トレーサビリティ]外部特性を工程毎に計測する。細分化工程程、悪い工程が精細に特定できる。[誤り率低下]各工程の作業は必ず誤りがあるから、必ずチェックする。工程をM分割すれば、誤り率は1/Mに低下する。(作込み誤り率×見逃しの第2種誤り率で評価する)ソフトウェアでの工程の細分化は、「標準作業を決める」「設計文書/図面の設定」「作業の細区分」等である。これで机上チェックを行えば必ず品質は向上する。

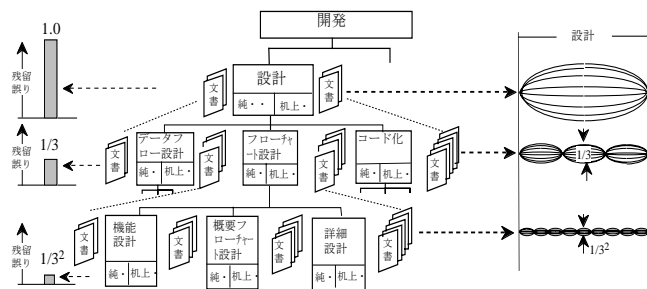


図7 工程を分割し統治する

工程の管理対象productには、肉体作業/機械作業/化学的処理作業など各種ある。しかし、「工程」processは外部的に評価/計測し管理するのみで、対象内部は各productの技術に委ねる。そこで図7等processの技術はproductに直交的で[5]、一般性があり広く利用される。「日本のソフトウェア工場」[21]もその一つである。

次にprocessやproductでのヒトの面を考える。図1aの「目的の階層性」の運用法を見る。最高司令官は国の勝敗の責任を負うから国の三軍の権限が与えられ、優れた資質の人を選んで必勝を期する。部下も階層の職に応じた技術一権限一責任一能力の人材を配置すると云う。開発のリーダーと周辺の人材も同様で、プログラマや論理設計者の上に立つ方式技術者/開発リーダーの育成が教育の要ではなからうか？

人の作業は定率の誤りを伴う。無意味な情報をコーディングすると1件/10行程度誤る。初心者は簡単なプログラムでも1件/10行程度誤る。プログラム作りの他に、誤り原因を分析し再発を防止する能力の育成・教育も必要ではないか？

良く機能する組織では、メンバを上下左右のネットワークで支えながら教育し作業させ伸ばして行く。全ての人々がネットワークでつながっているようだ。TVの戦争物では、各自が自らの範囲を護りながらも周辺に気を配り手を伸べあうチーム、全員を統率して極限状況を切抜ける優れたリーダー等が紹介される。これらのチームワーク、高いモラル、強いリーダーシップは高成熟度組織の特徴である。processの中核はヒトである。ヒトとその教育が重要課題ではないか？

4. ヒトの知の蓄積

経験の集積は知の第一歩である。新人チームがプログラム開発を繰返した時の生産性(作業工数/規模)の推移を説明する。図8の横軸は経験回数、縦軸は生産性、図a[13]は直線尺度で図b[5]は両対数尺度、2曲線は夫々の年の実績例である。図aのように、初めに急激に立上がるが次第に勾配が低下する傾向を示す。これはIEでは習熟効果(一部では学習効果)と呼ばれ、ヒトの知的作業でも肉体的作業でも現れる[14]。

両対数尺度の図bのプロットの傾向線は直線状であり、この種の習熟は対数習熟効果と呼ばれる。第X回目の作業時間Yは、Aを習熟指標値として次式で表され、

$$Y = (\text{初回作業時間}) \cdot X^{-A}$$

対数尺度用紙上では直線傾向になる。ヒトや組織の外部特性は殆ど対数習熟効果である。これは記憶で支えられると推定されてきた。ヒトでなくても組織の仕掛け/システムや方法は持続するから、組織や産業界全体でも対数習熟効果が現れる。経験則ではあるがよく使われる将来予測手法である。

筆者等は「人に倣ったソフトウェア自動設計」の研究の中で、この理論モデルを解明した[3]。以下迂遠ながらヒトの設計から説明する。人に倣うべく、図1, 2のように親子の概念展開を繰返して設計をする。更にZipfの「労力最小化の法則」[15]を基礎としてヒトの知を3種のエンジンで模擬した。

この法則は「ヒトは問題に直面すると、まず最も簡単な解を試み、不可ならより高度な解法を取ることを繰返す」と云い、ヒトの知は複数種のエンジンであることを意味する。Rasmussenは多様に現れる知の3典型[16]を指摘した。筆者らは最低から出発して3典型の場合を研究した。初めは反射的な「技能レベル」で、親概念であるUDFと詳細化データフロー(小さな制御の流れを含む)の対を単位知識とした。次は「ルールレベル」でフレーム形メモリを使い、ルールはデータである文型のスケルトンとメソッドを持つ。子の詳細データフローは、「親:(時を)知らせる=子:(時刻を)得る→(時刻表示を)得る→(時計盤面を)表示する」の下線部分のスケルトンに括弧内のデータや助詞をメソッドで完成させる[7, 8]。

「知識レベル」は、行き詰まった時に基礎用語毎に各種概念を定義した辞書を併用して解決する。例は図2のジャクソン形の並列データフロー分割で、ここでは(時と時計)で同一性を知るから、長/短針で定義すると認識出来ない。しかし、時分秒と針属性等の定義した辞書を使えば克服できる。

設計を繰返す時、3レベルの知識は全て対数習熟状に蓄積された。ルールレベルの(追試確認資料も含む)2例を示す。図8c[6]は直線尺度で始め急激に立上がり次第に緩やかにな

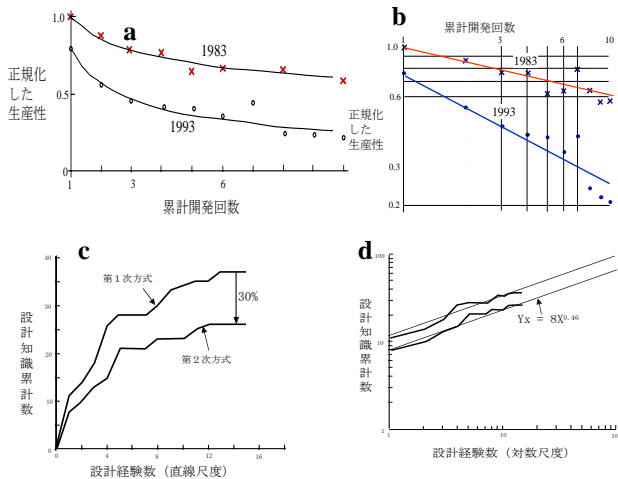


図8 経験回数と経験の蓄積

り、両対数尺度の図8d[6]で直線状で対数習熟状である。即ち、ヒトに似ている。

3レベル共に対数習熟効果状になる理由は二つある。第1は、(Zipfの法則の云うように)簡単で効果の高い知識を優先的に使い成功して偏りが生じた。第2に、知識を使用頻度順にソートすると負の指数分布状なので、新知識の蓄積は指数状に増加した。Zipfは元来言語学者で、頻度n番目の語の頻度はほぼ1/nに比例する(負の指数状でもある)ことで自分の法則を説明した。実験結果の3場合とも将に同じであった。もし、ランダムに選ばばこの偏りは生じない。高度能力者と低度能力者が作業すれば高度の方がより先鋭な分布になる。レベル数が増えるあるいは粒度がより小さいなら、より先鋭になる筈。(小さな単位での展開の選択は正解であった。)

以上により、対数習熟効果は、記憶とそのメカニズム由来することが明らかになった。これはBoeingの飛行機工場の適用例で初めて公式報告された1936年の70年後に当たる。

スキルレベルでの一連の設計状況は、経験例を基に作業する初心者に似ている。自動展開率は速やかに上り80%を超えるが、それ以上はなかなか向上しない。つまり浅慈慧の限界である。ルールレベルで経験を増すと、初めに親概念の使う動詞数に指数的に比例して直線的に急上昇し、次いで動詞の意味に指数的に比例して緩やかに直線的に増える。そこで、ある程度のルールを備えれば、100%近い自動展開率で作業でき、不可な場合には該動詞のその意味のルールを作り込めれば良いことが判った。

図8bの習熟曲線は、改善/向上の努力×成功率に応じて大きくなり、同一状況で同一の改善/向上努力を続ける時に同傾向で伸びることが経験的にも知られていた。これは全て知識の集積とすれば完全に経験に適合する。

図9はNEC佐々木会長が「NECに於ける品質革新の取組み」を講演された資料[17]からの引用である。1980年代初めから継続して品質向上の努力をされた。図の縦軸は正規化したフィールドバグである。1985年から初めて対数習熟曲線状に急降下し、2001年に至った。急降下した後、一時は平坦に見えるが、実体は各所で改善/向上が続いた。同図中のフィールドトラブル件数の第2の図でも、対数習熟状に急降下してい

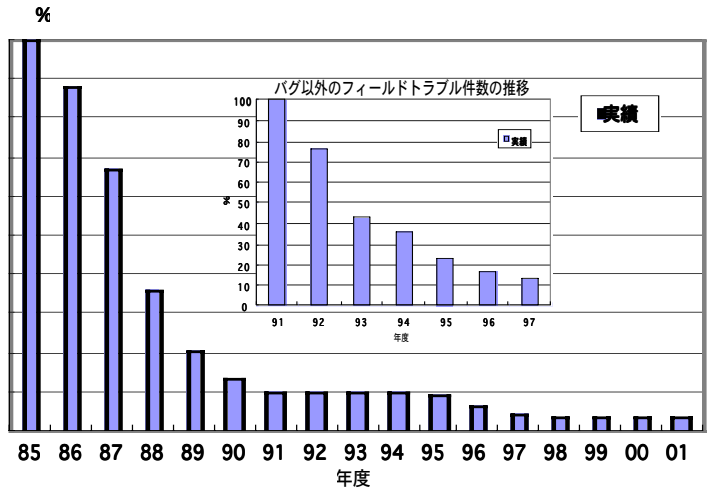


図9 NECにおけるフィールドバグ削減の状況

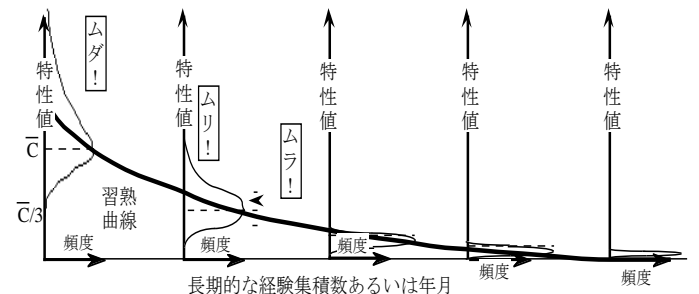


図10 超長期的な習熟曲線

る。NECではソフトもハードも等分に扱いつつながら改善を続けて居られる。この超長期の習熟の図は、「習熟効果により改善した力、それは知の集積総量の反映で、常識的用語では技術力」と云えよう。

5. むすびにかえて

ヒトの意図的行動は、ヒトの記憶している自然言語の概念展開の繰返しにより行われる。それはヒトの産れもった記憶を反映して、優れているとは言い難い。それ故、効果的に働かせるにはヒトの面に多大の配慮を必要とする。既に記した強調事項はここに起因する。しかし、これらは従来判ったつもりで、なおざりにされてきた感が強い。

ヒトは言語を操る能力で初期的な文明を築き、ネアンデルタールが死滅した最終氷河期を乗り切ったと云う[18, 19]。これを手始めにヒトは遺伝を離れ、文化を継承し発展させる動物とは異なる継続的な発展を始めた。概念の階層展開連鎖とその記憶機構はヒトの知の基本構造と云える。

本報に記した概念の階層展開方式は、個人としてのヒトの発達、ヒトの文化等の発達の定量評価手法になると考える。ヒトや社会の発達に関わる各位にご利用をお願いしたい。

ソフトウェア等の自動設計方式として、更に研究を積みあげICOTの果たせなかったヒトの知のsimulatorを1/10のコストで作れる可能性がある。これはヒューマノイドやintelligent controllerとして、従来ソフトウェアより有益な利用が見込める。後者を推進する中で前者を推進する力が得られると考え

ている。

ヒトの知の作業として見れば全てのプロセスは共通性を持つ。本報では定量特性や習熟効果の基礎を明らかにした。(これらは夫々の分野の方々にご吟味頂き、ご叱正を頂きたく思う。)この種の process の共通技術は日本製品の世界進出を支え、現在も KAIZEN は世界をリードしている。今横幹連合 [20] として再度の挑戦が計画されている。本研究は新しいプロセスのお役に立てると考える。

純粹に知を考えることで、ヒトの共通的な特性が明らかになった。図 9 に作った図 10 の超長期的な習熟曲線で考えると、現在のハードウェア製造は曲線の最右端に位する。ハードウェア開発や一領域のソフトウェアを継続開発する組織は、中央の左右の領域であろう。

最も問題が多いのは、依頼に応じて開発するソフトウェア組織で、十分な技術蓄積 / 仕掛け / プロセス / 設備に恵まれない場合である。毎回異種領域、異種のプロセスで、違う人々と作業するならば、十分な知の集積無しの開とに成りかねない。この時、習熟曲線の左端の如くに $\times 1/N \sim \times N$ 倍の大きなバラつきに悩まされる危険性がある。

この領域では定量化が遅れている。我国では鳥居教授、米国では Basil や Boehm 等の初期からのパイオニアの方々が Empirical software engineering の面から努力されるに留まる。それは「科学的であるには定量的であるべき」との信念と、繰返し開発中心に安定なプロセス特性が出ていた往時の経験が基礎にあると思われる。定量性の共通認識を確立しよう。

定量性無くしては評価や識別は困難であり、各種の因果関係が不明になる。各種の比較で技術の分化が行われ難くなるから、各種の考えが次々と重なりあふ。他分野では、新提案は既存要素と新規要素に分けられ、全て位置づけが確定する。この領域では、全く異なる様相を感じる。

出発点が「依頼に応じて作る」構想だから、技術の集積とは基本的に反する所が多い。これが重なると従事者は、毎回経験するプログラム技術しか集積されず、上流で死命を制する分野毎の技術、より共通度の高い方式技術等の意識や技術が育ち難くなる。本報等の共通性に基づく考え等とすり合わせを行って新たな路を拓く必要がある。

各種色々として記させて頂いた。次は「知の世代」との声がある。本報記載のように「それは本当だ」と信ずる。大事なことは、今も既に知の世界があることで、次世代は旗色がより鮮明になるのみと考える。

謝辞

この研究は、埼玉大学での「人に做ったソフトウェア自動設計」の研究の成果が使われている。B.H. Far 先生を始め、プロジェクトに携わったの当時の学生諸君に厚くお礼申し上げます。

参考文献

- [1] 養老孟司, 唯脳論, 筑摩書房, 1998.
- [2] 養老孟司, 人間科学, 筑摩書房, 2002.
- [3] Koono Z., Chen H. and Far B. H., Expert's Knowledge Structure Explains Software Engineering, Joint Conference on Knowledge

Based Software Engineering 1996, pp. 193-197, 1996.

- [4] Chen, H., Far, B. H. and Koono, Z., A systematic construction method of an expert system used for automatic software design - Acquisition and reproduction of design knowledge from design process-, Journal of Japan Society for Artificial Intelligence, vol. 12, no. 4, pp. 616-626, 1997.
- [5] 河野善彌, 陳慧, 人の設計知識と定量評価 (1/2-2/2), 信学技報, Vol. 103, KBSE 2003-57-58, pp. 67-78, (2004・3)
- [6] Chen H., Tsutsumi N., Takano H., Koono Z, Software Creation: An Intelligent CASE Tool Featuring Automatic Design for Structured Programming, The Journal of Institute of Electronics, Information and Communication Engineers -Special Issue on Knowledge-Based Software Engineering-, Vol. E81-D, No.12, pp. 1439-1449, Dec. 1998.
- [7] Abolhassani, H., 河野善彌, 陳慧, ソフトウェアクリエーション: ルールによる自動設計と知識による自動設計, 情処研報, ソフトウェア工学, 138-15, pp. 105-112, 2002. 7.
- [8] Koono, Z., Abolhassani, H. and Chen, H., A new way of automatic design of software (Simulating human intentional activity), New trends in software methodologies, tools and techniques, (Proc. of the fifth SOMET 06), p. 407-420, IOS Press, 2006.
- [9] Cluasewitz, Karl von, Vom Kriege, 1832. 淡徳三郎 (訳), 戦争論, 徳間書房, 1965.
- [10] Boehm, B. W., Software engineering economics, Prentice Hall, 1981.
- [11] Thayers, T. A., et al., Software reliability study, Final Technical Report, RADC-TR-76-238, Rome Air Development Center, 1976.
- [12] Shiomi, H., On analysis and summarization of human reliability data for simple VDT operation., BICRMS 92, pp. 372-377, 1992.
- [13] 森岡洋介, 長野文子, 大野治, EAGLE/P を用いたプログラム開発の習熟, 情処学会全国大会 (平成 3 年後期), 1K-3, 1991.
- [14] 諸岡孝次, 習熟性工学 (改定版), 建帛社, 1982.
- [15] G.K. Zipf, Human Behavior and the Principle of Least Effort," Hafner Publishing, 1972.
- [16] Rasmussen, J., The role of hierarchical knowledge representation in decision making and system management, IEEE Trans. on Software Engineering, 18, 6, pp. 523-533, 1985
- [17] 佐々木元, [基調講演] NEC における品質革新の取組み, 第 25 回ソフトウェア品質シンポジウム 発表報文集, pp. 13-39, 日科技連, 2006.
- [18] 赤澤威編, ネアンデルタール人の正体, 朝日選書, 2005. 第 10 章 言葉を話したか, 内田伸子, p. 257-282.
- [19] 内田伸子, 心理学一心の不思議を解き明かす一, 光生館, 2005.
- [20] 横幹連合, 第 1 回 横幹連合総合シンポジウム一統合知の創成と展開を目指して, 横幹連合, 2006, 12.
- [21] Cusumano, M. A., Japan's software factory: A challenge to U. S. management, Oxford University Press, 1991. (訳) 富沢宏之, 藤井留美, 日本のソフトウェア戦略ーアメリカ志気経営への挑戦, 三田出版会, 1993.