

創造的ソフトウェア構築を目指して

An Approach to Creative Software Design Frames

— デザイン論とイノベーション研究からのアプローチ —

本小論は、電子情報技術産業協会/ソフトウェアエンジニアリング技術分科会の 2010 年度の活動の中で行った「デザイン論」、および、「イノベーション研究」のヒアリングを契機として、次世代の創造的ソフトウェア構築手法に関する基本的な考え方や方向性について、まとめたものです。

ヒアリング報告というよりは、筆者のソフトウェア構築に関する想いや主張が色濃くなっていますが、ご容赦ください。忌憚の無いご意見などいただけると幸甚です。

大槻 繁 (おおつきしげる)



株式会社一 (いち) 副社長
アジャイルプロセス協議会 フェロー
知働化研究会 運営リーダー

2011 年元旦

(2011.1.28 修正)

1. 序.....	4
2. デザイン論	5
2. 1 デザイン論の系譜.....	5
〔デザインという用語〕	5
〔ASE モデル〕	6
〔英米アプローチの差異〕	6
〔デザイン思考と認知活動〕	7
〔SIT アプローチ〕	7
〔状況アプローチ〕	7
〔デザインと創造性〕	8
2. 2 デザイン活動の種類	9
〔実体概念と抽象概念〕	9
〔デザイン活動の種類〕	9
〔プッシュとプル〕	10
〔ソフトウェアデザインの特徴〕	10
2. 3 概念生成のモデル.....	10
〔概念生成のモデル〕	10
〔メタファ〕	11
〔アブダクション〕	11
〔抽象概念演算〕	11
2. 4 創造的デザイン	12
〔デザインの再定義〕	12
〔意味論的転回〕	12
〔イメージ・スキーマ〕	13
3. イノベーション研究.....	14
3. 1 開発思想史記述の3段階モデル	14
〔3段階モデル〕	14
〔ARPA ネットの技術史記述〕	15
3. 2 技術移転型のイノベーション.....	17
〔技術移転の記述〕	17
〔MARS-1 開発 (ハードウェア)〕	17
〔MARS-1 開発 (座席予約システム)〕	18
3. 3 イニシアティブの導入	19
〔3段階モデルの一般化〕	19

4.	デザインとイノベーションの実践.....	20
	〔IDEO〕	20
	〔d.school,i.school〕	20
	〔U理論〕	20
	〔Managing Flow〕	21
5.	創造的ソフトウェア構築へ向けて.....	23
5. 1	ソフトウェア領域の潮流.....	23
	〔ソフトウェアの本質〕	23
	〔知働化〕	24
	〔研究・探求の方向性〕	24
	〔新ソフトウェア宣言〕	25
5. 2	ソフトウェアデザイン論.....	26
	〔ブルックス,Jr.の主張〕	26
	〔概念構造体〕	27
5. 3	ソフトウェア哲学.....	28
	〔理論や技術を基礎付けるもの〕	28
	〔合理論と経験論〕	28
	〔言語ゲーム的転回〕	30
6.	おわりに.....	32
	〔結言〕	32
	〔参考資料〕	33
	〔謝辞〕	36
	〔本当のあとがき〕	36

1. 序

決まりきった問題を解き続け、組織的に役割分担を行い、手順や段取りを決めてソフトウェアを開発する世界は、産業としてこれからも形を変えて存続していくことでしょう。これに対し、今までに無い問題、新しい領域に取り組んでいく創造的ソフトウェアが、ますます重要になってきていると思われまます。

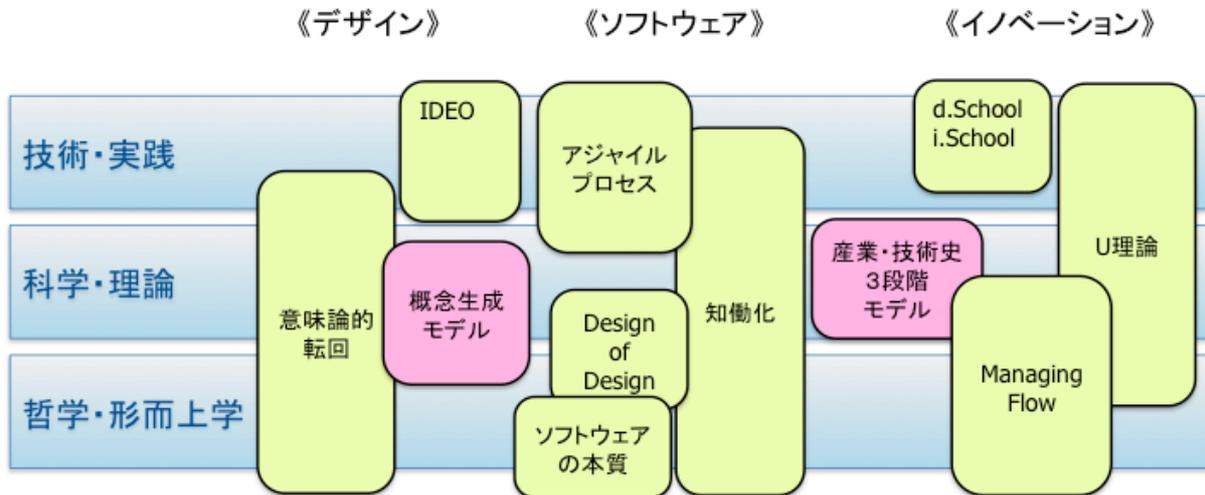
実世界におけるソフトウェアのもたらす意味は、人、あるいは、組織にとって主観的であり、多様です。また、ソフトウェアは、人間が創造する人工物（アーティファクト）です。本小論では、ソフトウェアより広い一般の人工物に対する「デザイン論」と、新しいものが生まれる「イノベーションプロセス」という二つの観点から、研究動向を把握し、それ等を踏まえてソフトウェア開発の将来像について考えてみようと思います。

ソフトウェア開発プロセスについては、従来のウォーターフォール型のプロセスが適用できない分野が多くなってきており、システムの置かれている状況や要求の不確実性に対応することを目指したアジャイル型の開発プロセスが台頭してきています。筆者は、近年、ソフトウェアに関する産業構造や、開発手法などが著しく変化してきていると実感しており、次世代のソフトウェア開発方法を模索するために知識主導型社会でのソフトウェアのあり方を探求してきました。

2010年4月1日に、『人月の神話』[Brooks1975,1995]で著名なブルックス,Jr.が、“The Design of Design”[Brooks2010]を出版しました。この書籍は、ソフトウェアが、開発からチームデザインの時代になってきていることを秀逸な随想というスタイルで語っています。ソフトウェアを「デザイン」の対象として見ることは、大きな視点の転換を意味しています。一般的な人工物とソフトウェアとの差異はありますが、まず、デザインとはどういった活動なのかを、押さえておく必要があります。また、社会的インパクトの大きいイノベーションが、どのようにして起こるのかという状況も歴史的な観点で見ておこうと思います。

本小論では、第2章で電子協のヒアリングで行った北陸先端科学技術大学院大学/知能科学研究科の永井由佳里先生の「デザイン論について」の講演、第3章で関西学院大学/総合情報学部の喜多千草先生の「イノベーションプロセス」の講演について解説します。第4章でデザインとイノベーションの実践状況についていくつかのトピックスを採上げます。引き続き、第5章でソフトウェアにおけるデザイン論とイノベーションプロセスについての考察を述べ、筆者のこれからのソフトウェアづくりについての見解をまとめています。

本小論の中で扱うトピックスの全体像を俯瞰したものを以下に示します。



2. デザイン論

本章は、2010年7月30日開催の電子協での永井由佳里先生の講演内容 [Nagai2010]、および、講演時に触れた論文『デザインの創造性と概念生成』 [TauraNagai2010]とに基づいて、筆者がまとめたものです。

2. 1 デザイン論の系譜

〔デザインという用語〕

英語の“design”には、動詞としての「意図する、立案する」「指摘する、指示する」「描く、デザインする、作図する」といった意味と、名詞としての「計画、意図」「目標」「設計」といった意味があります。レオナルド・ダ・ヴィンチ (1452-1519) の時代からデザインという行為は認識されていましたが、造形領域において、他の芸術分野から分離して、“design”という語が、“composition”といった特定の領域を示すようになったのは19世紀後半からとされています。

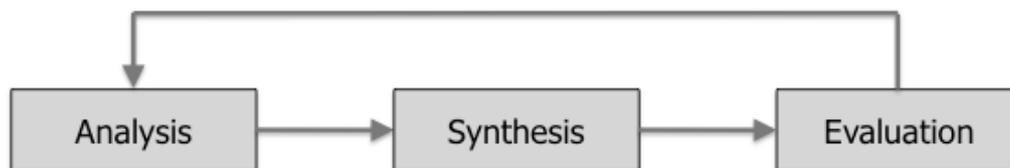
日本語の「設計」という言葉には、「建造物の工事、機械の製造などに際し、対象物の構造・材料・政策法などの計画を図面に表わすこと」「一般に計画をたてること」という意味に使われています。一方、カタカナの「デザイン」という言葉では、「建築、工業製品、服飾・商業美術などの分野で、実用面などを考慮して造形

作品を意匠すること」「図案や模様を考案すること」「目的をもって具体的に立案・設計すること」という意味に使われています。

英語、日本語のいずれにおいても、デザイン論における意味は、上記よりも広いものとなりつつあります。“** design”あるいは「〇〇のデザイン」といった表現に見られるように、細分化しやすく、これが学問領域でも縦割りになりがちです。実際、「ソフトウェアのデザイン」の領域は、ソフトウェア独特のものになっています。「デザイン」そのものは超領域的な概念です。

〔ASE モデル〕

語感として重要なことは、「未来、新規性」を内包していることです。過去や既存のものに対しては適用されません。また、過去のデータや現象の分析を中心とする科学 (science) とは異なり、「統合 (synthesis)」が含まれています。



永井先生 JEITA 講演資料より (2010.7.30)

上図 (ASE モデル) は、一般的なデザインプロセスですが、Analysis と Synthesis とが対峙しているかどうかというのは、学者によって見解が異なるようです。機械やソフトウェアでは、稼働や実行を伴うために、このような考え方が適合しています。要求定義、実現、テストの関係も、基本的に同じです。

〔英米アプローチの差異〕

デザイン研究の潮流は、英国では“Design Studies”と呼ばれ、ナイジェル・クロスが立ち上げました。米国では“Design Issues”と呼ばれ、CMU や MIT の研究者が中心となっています。英国と米国との差異が顕著なところは、人間の創造的性質 (creativity) についてです。英国は、誰でも教育やトレーニングによってデザインができるようになるという立場です。これに対し、米国では、デザインは天才しかできないと考えられています。日本は、あいまいさを許容できる文化があるため、独自のアプローチが採られています。

〔デザイン思考と認知活動〕

デザイン思考 (Design Thinking) [Rowe1991]は、ロウ (Peter G. Rowe) が広めたもので、もともとは建築デザイナーの思考プロセス研究です。図面やスケッチなどの視覚的情報による Visual Thinking が特徴です。基本的には ASE モデルをベースにしていますが、問題解決プロセスの枠組みを導入し、「目標 (ゴール)」を説明することができます。

デザインを「認知活動 (Cognitive Activity)」とみなして、デザイナーの認知プロセスを観察するデザイン思考研究があります。これにはデザインを問題解決とみなすサイモン (Herbert A. Simon) の提唱した「SIT アプローチ (The Stativity Approach)」と、状況との対話という実践的な立場をとるショーン (Donald A. Schon) の提唱した「状況アプローチ (Situativity Approach)」との二つの枠組みがあります。

〔SIT アプローチ〕

SIT (The Stativity Approach: general symbolic information processing approach) アプローチ[Simon1996]は、人工的なものを創造することが、個人が何かを行おうとする時に課題を表象する「問題 (problem)」から「解 (solution)」を求めるプロセスであるとするものです。「問題」は、複雑であるため、枠組みを変えたり、見方を変えたり、分割・詳細化する段階的な手続きが必要です。ソフトウェアの世界では、Michael Jackson の「問題フレーム (Problem Frames)」[Jackson2001]に近いアプローチと言えます。ソフトウェアエンジニアリングの立場から言うと、「デザイン (設計)」という用語は、「解」であるプログラムによる実現 (implementation) のことを示すので、問題とその解決の行為全体を「デザイン」と呼ぶことには、多少の違和感があります。

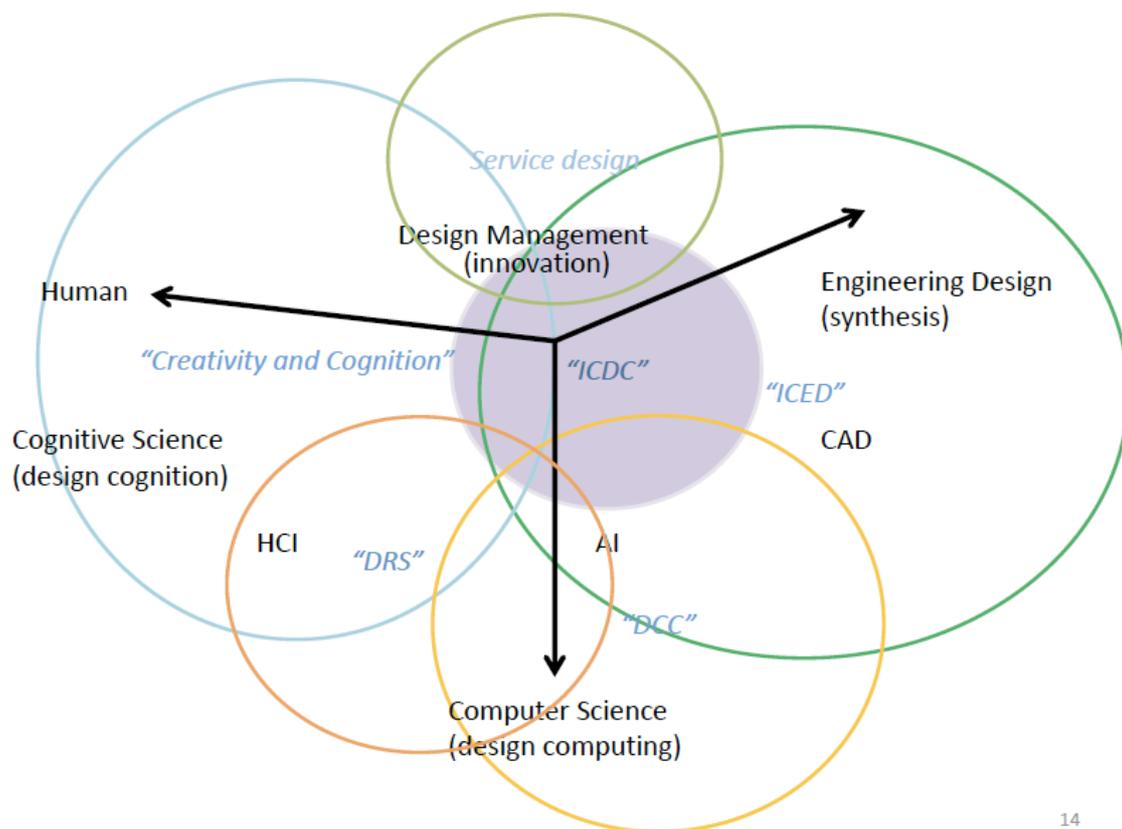
〔状況アプローチ〕

状況アプローチ (Situativity Approach) [Schon1983]は、Reflection-in-Action や、Seeing-Moving-Seeing といった見方に現れているように、デザイン活動を社会的なプロセスとみなす立場です。人工物のユーザの視点を理解したり、観点が異なる解釈の違いが、新たな表象を生むこともあります。表象と心的イメージとの関係性が研究対象となっています。

〔デザインと創造性〕

SIT アプローチと状況アプローチとを統合する試みもあります。ヴィサール (Willemien Visser) は、人工物 (Artificial Products) を心的表象と実体表象の二つを含むとし、総称して認知的人工物 (Cognitive Artifact) としてとらえることによって統合を試みています[Visser2006]。

認知と創造とは緊密な関係にあります。人間は本質的に創造的な生産活動を行いたいという欲求を持っており、その生産された人工物を認知・認識し、そして、次の生産を行っていきます。



14

ICDC: International Conference on Design Creativity
<http://www.jaist.ac.jp/ks/labs/nagai/DesignCreativityW/>

DRS: Design Research Society (プロダクトデザインや建築関係が中心)
<http://www.designresearchsociety.org/joomla/>

ICED: International Conference on Engineering Design (主催: the Design Society、工学系のデザインが中心) <http://www.designsociety.org/>

DCC: Design Computing and Cognition (International Conference)
 AI: Artificial Intelligence; CAD: Computer Aided Design
 HCI: Human Computer Interface

永井先生 JEITA 講演資料より (2010.7.30)

2. 2 デザイン活動の種類

〔実体概念と抽象概念〕

デザインにおける概念は、「人間が心のなかに抱く、既存、あるいは将来存在可能な実体、あるいはその類や属性に関する表象」と定義されています。一般に、概念には、人間が実体に関して形成する「実体概念」と、人間が意味ないし価値に導かれて実体概念を類に分類したとき、その各類に関する概念である「抽象概念」とがあります。

〔デザイン活動の種類〕

次図に示すようにデザインは、概念生成に基づいて図案表現型、問題解決型、理想追求型の三つに分類することができます。



『デザインの創造性と概念生成』（田浦, 永井）より

図案表現型のデザインとは、内部に記憶されているイメージに具体的な形や色を与え、外的に表出する行為です。これは時間的な方向性としては、過去を指向しており、過去の記憶に駆動されるプロセスです。デザインの能力としては、実世界に未だ存在していない新しいかたちをつくり出すことであり、「表現力」に類する創造性になりますが、真に新しい概念を生成するという本質的創造性の観点からは、内的なイメージそのものを生成する能力が求められます。

問題解決型のデザインとは、現状の問題を解決するために目標（ゴール）を設定して、それを解決するデザインプロセスに従っています。現在の時点で現存している問題を分析的に現状認識することによって駆動される現在指向のプロセスです。デザインの能力としては、問題の目標に照らして所与の対象である現状を分析する

こと、すなわち、それを構成する部分や要素などに分け入って解明する能力に深く関係しています。

理想追求型のデザインとは、未来の理想を描くことに主眼をおく、未来指向のプロセスです。ここで言う「理想」とは、現状から容易に類推することができないため、全く新しいものを構成する能力が求められます。前図のようにデザインを分類することによって、目的指向やニーズのような外部主導のものより、内的動機によって駆動されるものを扱うことができるようになると考えられます。

〔プッシュとプル〕

デザインされたアイデアの源を成している内的なデザインイメージがどのように生成されるかによって、目標をたよりに概念が引っ張られるように生成されるプル型 (pull) と、概念がデザイナーの内的な感性から生み出されるプッシュ型 (push) に分類することができます。問題解決型ではデザインイメージはゴールから分析的にプルされ、理想追求型ではデザインイメージは内的な感性から構成的にプッシュされ、さらに、図案表現型ではデザインイメージから具体像が表現されます。

〔ソフトウェアデザインの特徴〕

図案表現型、問題解決型、理想追求型の三種のデザイン活動は複合的であり、組み合わさったり、並行的に進められたりするものと思われます。ソフトウェアの世界では、ほとんど全て問題解決型が中心になってきたと言っても過言ではないでしょう。実装アーキテクチャの観点から可能であれば、ユーザインタフェース部分が切り出され図案表現型の活動が行われています。また、今までにないソフトウェアを開発する場合にも、未来の問題設定を行い、未来のゴールから牽引されるという想定のもとに問題解決型のパターンをあてはめて開発プロジェクトが推進されてきているとみなすことができます。

2. 3 概念生成のモデル

〔概念生成のモデル〕

デザインにおける「概念生成」のモデルとしては、メタファ、アブダクション、抽象概念演算等があります。これ等を明確にすることによって、先の「デザインイメージ」の生成過程の類型化を行うことができます。

〔メタファ〕

メタファ（隠喩）とは、あるものをその対象と似ている別のものに例える表現方法です。メタファによるデザインでは、デザインイメージが、結果として、例えられるものに類似するようにするのです。

〔アブダクション〕

推論の形式では、アブダクション（仮説形成）とメタファの枠組みとは、ほぼ同じ構造です。メタファの概念生成の基本はグルーピングとして説明することができます。グルーピングによるグループ（抽象概念）のことを「一次の抽象概念」と呼びます。メタファやアブダクションを用いてデザインイメージを生成することは、結果的にグルーピングできるように「転写」を行うことです。従って、グルーピングを背景とする「転写」は、デザインの求める創造性が、既存の範疇の亜種を求める程度のものであれば有効な方法と言えます。

〔抽象概念演算〕

抽象概念演算は、より高度な概念生成の方法です。これは以下に述べるような数学的な定式化が可能です。

公理：操作公理

抽象概念は実体概念集合の位相。

定理 1：理想的知識（スーパーマンの知識）は、ハウスドルフ空間。

（スーパーマンの知識とは、現実の世界を完全に知っており、一対一対応です。どんな異なる二つの実体も、位相（近傍）として分離可能です。）

定理 2：要求は適当に選んだ抽象概念（一般には機能概念）の積として表現。

（これに、コンパクト性を導入することによって、位相空間に「距離」が定義できるようになると、「解」の探索に見通しが得られます。）

抽象概念を総合することにより、新たな概念を生成するデザインプロセスとして「範例モデル」（知っている実体から、構成要素を分解して、別の要素と交換する）、「近傍探索モデル」（ある領域を出発点として、隣接する領域に移動していく）とがあります。抽象概念の操作によって高度な概念を生成していくためには、「類似性」のみならず「差異性」の認識も必要です。

抽象概念操作によって生成される抽象概念を「高次の抽象概念」と呼びます。こ

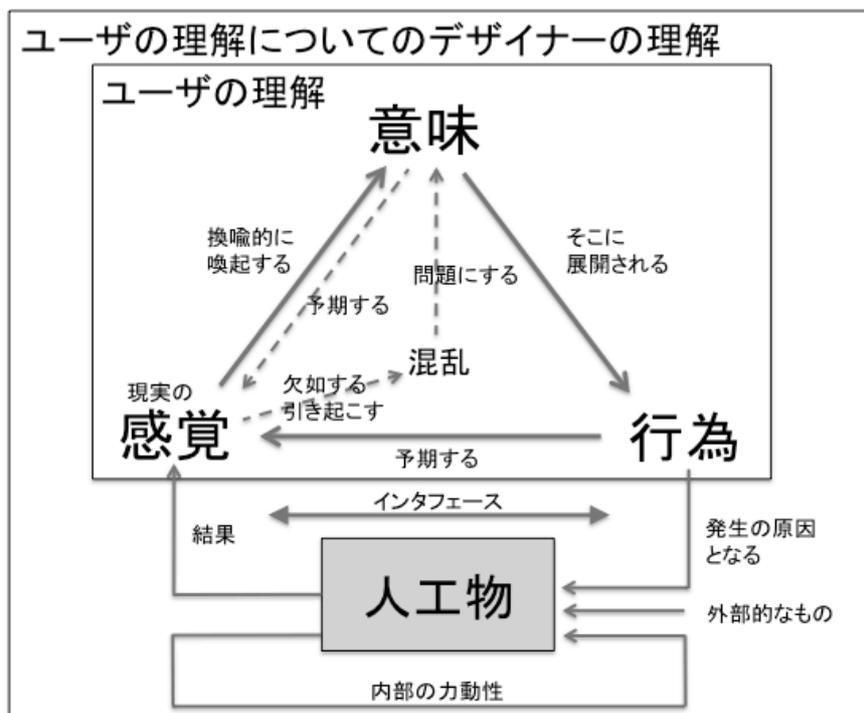
の高次の抽象概念は、「分類学的関連 (taxonomical relation)」と「主題的関連 (thematic relation)」として解釈することができます。分類学的関連に基づく生成プロセスのことを「(主題的関連に基づく) 概念合成 (concept blending)」と呼びます。これは、例えば「雪」と「トマト」から「パウダタイプのケチャップ」という概念が合成されることに相当します。一方、主題的関連に基づく生成プロセスのことを「概念統合 (concept integrating)」と呼びます。これは、例えば「雪」と「トマト」から「湿度を保つ冷蔵庫」という概念が生成されることに相当します。これは、二つの規定概念から構成される状況やシーンから生成されるものです。

2. 4 創造的デザイン [デザインの再定義]

デザインとは、未来に向かって、あるべき姿を構成すること

新規性 (独創性) や実用性がデザインにとって重要であることは間違いないのですが、それだけでは創造的で優れたデザインにはなりません。創造性の観点からは、「内的動機」が求められます。自己の感性、ユーザの感性や社会的に共鳴されうるものであることが重要です。

[意味論的転回]



『意味論的転回』(クリッペンドルフ) より

デザインされる人工物は、ユーザに感覚・共感 (impressions) を与えるものです。そして、それは人間の認知プロセスと関わりがあります。クラウス・クリッペンドルフは、著書『意味論的転回 (The Semantic Turn)』[Krippendorff2006]の中で、「科学」というのは、過去に起こったことを分析して何か法則を見いだしたり証明したりするのに対して、「デザイン」というのは将来について意思決定していくので、ぜんぜん違う、「人間中心」の「意味」を扱う体系でなくてはならない。」と述べています。

ここで言う「意味」が、言説 (ディスコース) や人間の社会的な言語行為 (言語ゲーム) によって規定されていくというのが、クリッペンドルフの主張ですが、これを感覚の背後にある「深層イメージ (depth image)」としてとらえる認知言語的なアプローチが提唱されています。

〔イメージ・スキーマ〕

工業デザインの世界では、感覚を評価する方法として、SD法 (意味差判別法: Semantic Differential Method) [Osgood1952] という言語概念の定量的数量化モデルが利用されていますが、この方法では深層イメージをとらえることはできません。ジョージ・ラコフの提唱している『認知意味論』[Lakoff1987]では、イメージ・スキーマと呼ばれる人間の意識領域での構造を分析しています。イメージ・スキーマについては、ラコフ以降洗練化されてきており、筆者が調べた範囲では、以下のようものがクロフト等[Croft2004]によって掲げられていました。

- ・ 空間 (Space) : 上昇/下降 (Up-Down)、前後 (Front-Back)、左右 (Left-Right)、遠近 (Far-Near)、中心/周辺 (Center-Periphery)、接合 (Contact)
- ・ 大きさ (Scale) : 経路 (Path)
- ・ 入れ物 (Container) : 収納 (Containment)、入出力 (In-Out)、表面 (Surface)、満空 (Full-Empty)、中身 (Content)
- ・ 力 (Force) : 平衡 (Balance)、反作用 (Counterforce)、衝動 (Compulsion)、制約 (Restraint)、能力 (Enablement)、妨害 (Blockage)、傾き (Diversion)、牽引 (Attraction)
- ・ 単一/多様性 (Unity/Multiplicity) : 統合 (Merging)、集合 (Collection)、分離 (Splitting)、繰返し (Iteration)、全体/部分 (Whole-Part)、分量/数量 (Mass-Count)、結合 (Link)
- ・ 同一性 (Identity) : 適合条件 (Matching Super-Imposition)
- ・ 存在 (Existence) : 除去可能 (Removal)、有界 (Bounded Space)、周期 (Cycle)、対象 (Object)、プロセス (Process)

3. イノベーション研究

本章は、2010年11月5日開催の電子協での喜多千草先生の講演内容[Kita2010]、および、講演時に触れた『イノベーションプロセスの思想的記述モデルについて』[Kita2008]とに基づいて、筆者がまとめたものです。

3. 1 開発思想史記述の3段階モデル

〔3段階モデル〕

産業や技術史の分野で「イノベーション」を的確にとらえるためには、これを「プロセス」として見なして、従来、重要な担い手たちのアイデア段階や計画段階での貢献を明確にしておくことによって、歴史から学べることも多いと考えられます。

ソフトウェア開発における要求、分析、設計、プログラミング、テストといったウォーターフォール型のプロセスや、着想 (inception)、起動 (elaboration)、構築 (construction)、変換 (transition) からなるプロセスを参考に、歴史上のプロセスを3段階で記述していくものが、「技術史記述の3段階モデル」です。

Phase	Leading Idea
Preliminary (アイデア段階)	Vision (開発思想)
Planning (計画段階)	Agenda (アジェンダ)
Implementing (実装段階)	Design (デザイン)



図 1 技術史記述の3段階モデル

喜多先生 JEITA 講演資料より (2010.11.5)

技術史記述の方法には、例えば、インターネット史の場合では、偉人伝型 (biographic)、組織重視型 (bureaucratic)、思想型 (ideological)、社会構成主義型 (social) の4つの傾向がありますが、こういったスタイルが統合されていくことによって、より良い歴史記述になっていくことが望まれています。3段階モデルは、思想の流れに着目しているため偉人伝型をベースにしていますが、その背景

となる社会的要因や組織的な観点も加味しているため、4つの傾向の統合化の方向になっています。

〔ARPA ネットの技術史記述〕

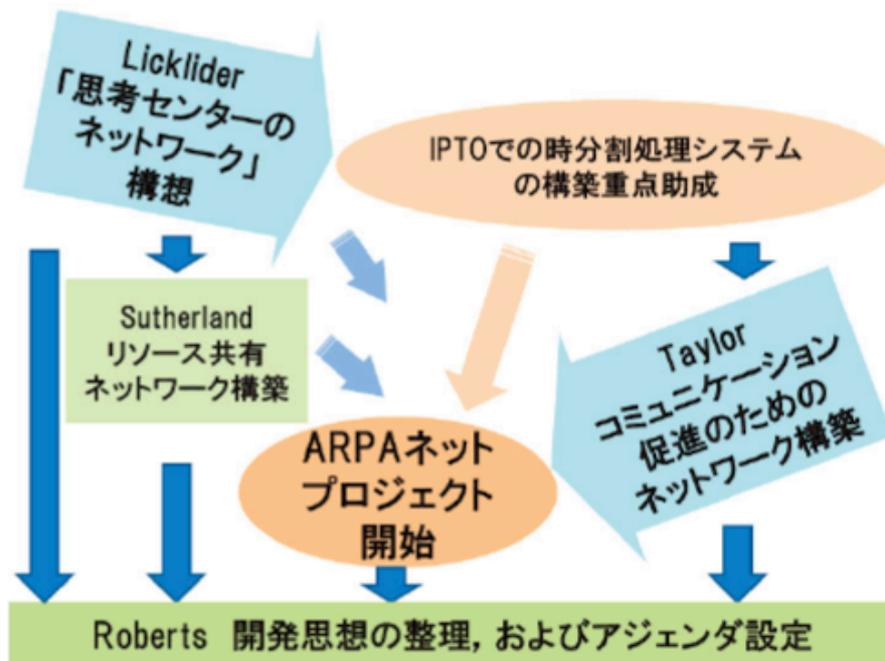


図 2 ARPA ネットのアジェンダ設定過程概略

喜多先生 JEITA 講演資料より (2010.11.5)

インターネットの前身の米国国防総省の ARPA (Advanced Research Projects Agency) ネットの開発過程を 3 段階モデルのアイデア段階と計画段階とを記述してみると以下ようになります。

- ・ アイデア段階 (開発思想)

- ① Robert W. Taylor : 予算を獲得した IPTO (Information Processing Techniques Office) 副部長

時分割処理システムごとにメールシステムがありコンピュータがコミュニケーション装置として使えることを認識しており、しかし、システムごとに異なる端末を使う不便さを体験。

②Charles M. Herzfeld : 予算を許可した ARPA 長官

時分割処理システムを接続するネットワークを構築するプロジェクト提案が、機が熟していると判断した。

③L.C.R. Licklider : IPTO の初代部長

重点助成領域を時分割処理システムの構築と定めた。デジタルコンピュータとデータ通信とを結びつけたネットワーク利用の将来性をいち早く看破。「思考センターのネットワーク」構想を持つ。

④Ivan Sutherland : 第2代の IPTO 部長

コンピュータグラフィックス分野のパイオニア。失敗はしたものの、ネットワークシステムの構築を試みた。ネットワーク構築の意義をコンピューティング資源の負荷分散としての「リソース共有」に意義を見いだした。

こういった人々が開発思想として機能しうる「コミュニケーション増進」「知的リソース共有」「ハードウェアリソース共有」というアイデアの複数併存性を示しています。

- ・ 計画段階 (アジェンダ)

Laurence Roberts (MIT Lincoln 研究所から ARPA に引き抜き) が、ネットワーク構築責任者となり、開発思想をまとめ直し、アジェンダをたてていきました。

- ・ IMP (Interface Message Processor) と名付けられたネットワーク用小型コンピュータをつなげて、サブネットワークを構築する方法を採用
- ・ 時分割処理システムに好適と言われるパケット通信方式を採用
- ・ 要求仕様をまとめ実装のための入札を行う (BBN 社が受注)
- ・ ネットワークに参加するコンピュータどうしを対等につなぐ通信方式を要求

3. 2 技術移転型のイノベーション

〔技術移転の記述〕

海外の技術を参考にして日本国内でのシステム構築を行うような技術移転型のイノベーションでは、移転元の実装段階の成果からアジェンダや開発思想を、移転先が読み解いていくプロセスになっています。このプロセスは、欧米技術のローカライゼーションの説明モデルとしても有用なものと思われます。

3段階モデルによって技術史を記述することによって、従来の歴史記述では単なる模倣とひとくくりにされていたものでも、

- ・ 先行システムに基づいて、同じ開発思想のもとに新しいアジェンダをたてたもの、
- ・ システムを実際に使って、ユーザビリティや社会的側面を考慮し設計の細部を変更したもの、
- ・ システム成果あるいは設計から、開発思想を類推し、それに従って新しいアジェンダをたてたもの

といった、イノベーションの段階ごとの解像度を持った記述をすることができるようになります。

〔MARS-1 開発（ハードウェア）〕

国鉄座席予約システム MARS-1 開発では、以下のようなイノベーションプロセスを辿っています。

- ・ アイデア段階：鉄道技術研究所の穂坂衛が、米国留学時のサイバネティクスの概念習得、国鉄の業務合理化改革の気運、国鉄独自の通信網の存在といった背景のもとでの、穂坂自身のブール代数を使った予約システムのひらめきがあった。
- ・ 計画段階：論理設計と仕様の作成は、大野豊が行う
- ・ 実装段階：実装は日立製作所が行う

ハードウェアの論理設計に関する知識の流れは、BendixG15D のアーキテクチャからさかのぼって得たリアルタイムシステム設計のアイデアが構築されています。

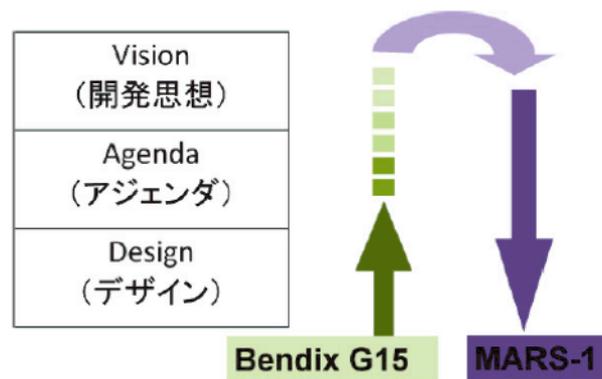


図 3 BendixG15D に関する知識が MARS-1 の設計に反映した構造

喜多先生 JEITA 講演資料より (2010.11.5)

[MARS-1 開発 (座席予約システム)]

座席予約システム全体については、米国の航空機予約システムを参考にしています。この場合には、外国文献の輪読会や技術報告書を通じて、独自の問題設定を行います。最終的には、アメリカン航空の Magnetronic Reservisor の資料を入手し、知識獲得を図っています。これに基づいて用語や概念を整備し、概念枠を構築しています。

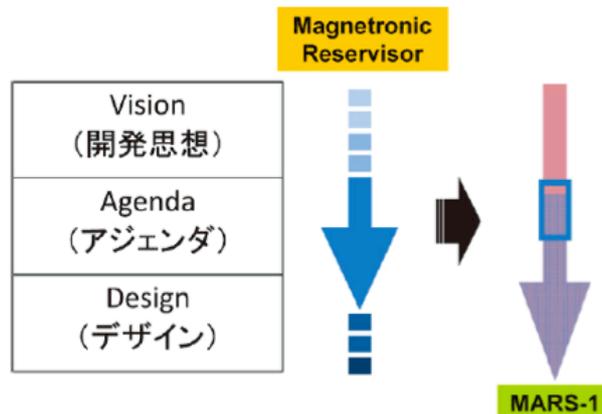


図 4 Magnetronic Reservisor に関する知識が MARS-1 の設計に反映した構造

喜多先生 JEITA 講演資料より (2010.11.5)

3. 3 イニシアティブの導入

〔3段階モデルの一般化〕

Phase	Leading Idea
Primary (初期段階)	Initiative (イニシアティブ)
Planning (計画段階)	Agenda (アジェンダ)
Implementing (実装段階)	Design (デザイン)



図5 イノベーション研究のための3段階モデル

喜多先生 JEITA 講演資料より (2010.11.5)

イノベーションの初期段階での「ユーザ」の役割を積極的にとらえることによって、「もの」や「こと」が社会に投入され、そのことが新しい開発のアイデアへ繋がるという連鎖を記述することができるようになります。

初期段階をイニシアティブ (initiative) としたものを「一般的3段階モデル」と呼びます。イニシアティブは、アイデア段階を先導するアイデアを、開発思想も含む、より広い概念です。これによって、ユーザの声がイノベーションを引き起こしたり、マーケティング上の戦略が初期段階の引き金になる場合などが記述できることとなります。無論、これはあくまでも「記述」のモデルであって、イノベーションを生むための(戦略)モデルではありません。ただし、こういった記述を積み重ねることによって、将来をデザインすることができるようになると期待できます。

4. デザインとイノベーションの実践

本章では、第2章のデザイン論、第3章のイノベーション研究というテーマに関連した実践的な取組みについて、トピックス的にいくつか採上げることにします。

[IDEO]

IDEO[Brown2008,2009][Kelly2001, 2005] は、米国のデザイン・コンサルティングファームで、多くの書籍で紹介されているように、「デザイン思考」と題するイノベーションを達成する実践的手法を提供しています。プロトタイピングによってユーザの共感形成や検証を行い、フィードバックサイクルを行いながら理解・観察・視覚化・評価・実現を繰り返して完成させていく方法を採用しています。商品開発のみならず、市場創造、戦略策定、組織改革に至るビジネス全体の改革支援を行っています。教育・啓蒙普及にも積極的に取り組んでおり、MITでの講義[Brown2006]もあるようですし、下記のスタンフォード大学のd.schoolの設立にも関わっています。

[d.school,i.school]

デザインやイノベーションに関する大学の取組みも盛んです。スタンフォード大学のinstitute of designは、“d.school (<http://dschool.stanford.edu/>)”とよばれています。デザインというよりイノベーションの教育を行う広義のデザイン学科です。AI（人工知能：Artificial Intelligence）やIA（Intelligence Amplifier）で有名なTerry Winogradが創立に関わっています。地理的關係からIDEOとの交流も盛んなようです。ブレインストーミングやプロトタイピングを中心にした実践的なコースになっています。

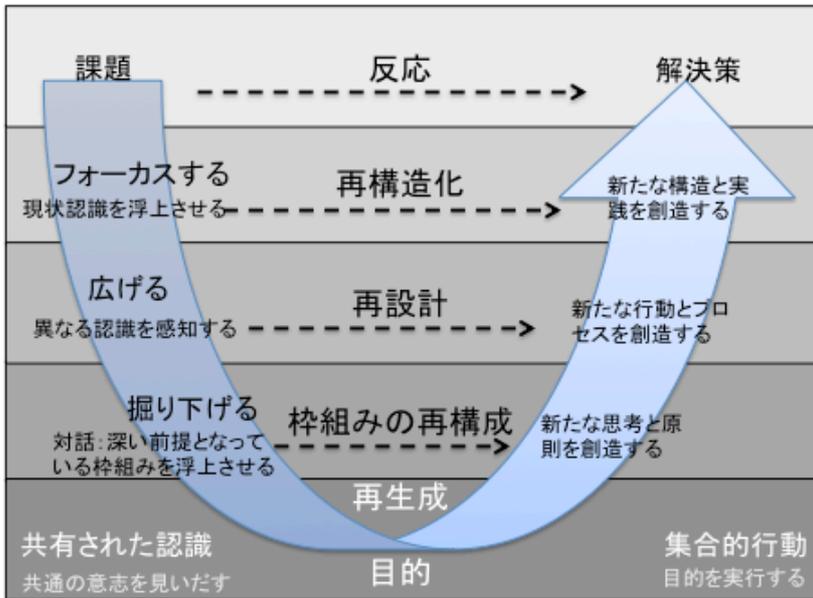
東京大学では、i.school (<http://ischool.t.u-tokyo.ac.jp/>) というイノベーションをうたったコースが開始されています[Todai2010]。「人間中心ワークショップ」といった実践を指向したリーダー育成のプログラムが組まれているようです。

[U理論]

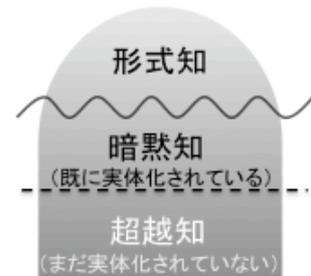
イノベーションのプロセスや思考方法を手法としてまとめる実践的な活動もあります。「U理論」[Senge2004][Scharmer2007]は、組織の変化を「思考」のレベルまで深めたものを提唱しています。啓蒙活動を推進しているシャーマーはMIT

の講師です。Apple のジョブスも好んでいるといわれている禅の修行や瞑想に似た考え方を取入れています。こういった表現や記述ができない領域を扱うことが、最近の一つの潮流と言えるでしょう。

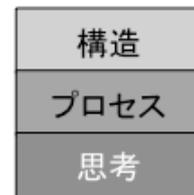
変化の5つのレベル



知識の3つの形態

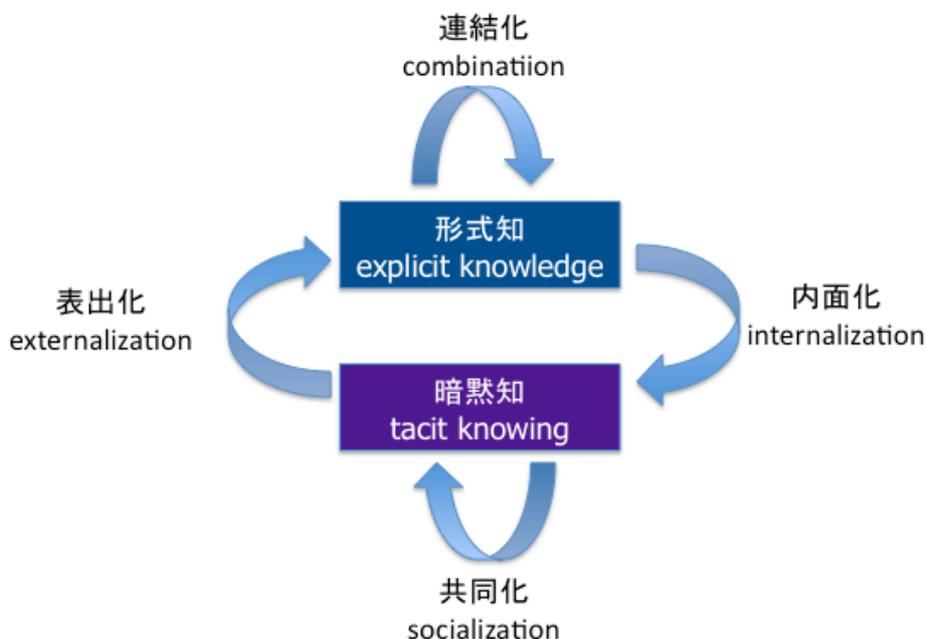


組織変化の階層



[Managing Flow]

経営論の分野では、知識やイノベーションが古くから主題となっています。SECIモデル[Nonaka1995]は、暗黙知と形式知の相互作用が、知識創造の源泉であると考え、継続的な知識の変化プロセスとして提唱されています。



これをさらに進化させたものが、“Managing Flow”[Nonaka2008]です。「場 (Ba)」は、知識を継続的に創造していくための相互作用が起こる心理的・物理的・仮想的空間です。企業活動の場では、実践的知恵（フロネシス）が重要であり、個別具体の場面の中で、全体の善のために最善の振る舞いを見出す能力のことです。

なお、ソフトウェア開発の分野で近年注目を浴びているサザーランド（Jeff Sutherland）等が提唱・普及活動をしている SCRUM は、源流は日本の製造現場の手法として行われていたものを野中郁次郎氏等が分析して論文発表したものです[Nonaka1986]。

5. 創造的ソフトウェア構築へ向けて

本章では、第2章のデザイン論、第3章のイノベーション研究、第4章のそれぞれの実践トレンド等を受けて、ソフトウェア構築の新しい取組みについて断片的に考えてみることにします。デザイン論は一般的な人工物を対象としています。ソフトウェアも人工物の一種ですから、多くの見方はソフトウェアにも当てはまると考えられます。また、ソフトウェア特有のデザイン論上の課題もあるでしょう。イノベーション研究は、一般的なITを対象としています。おそらくソフトウェアに限定した領域でも、技術思想史による記録的記述は有効でしょうし、ソフトウェア技術の戦略のヒントも数多く得られるようになるのではないかと思います。

5. 1 ソフトウェア領域の潮流

〔ソフトウェアの本質〕

一般的な人工物、工業製品とソフトウェアとが異なる特性は、ブルックス,Jr.の名著『人月の神話』[Brooks1975,1995]で的確に示されています。以下の「本質的困難」と呼ばれている4つの特性です。

複雑性

complexity

ソフトウェアは大きくて複雑なことそれ自身が問題なのだということです。レゴブロックを集めて組み合わせるようなわけにはいかず、他のどのような構造物よりも複雑です。ソフトウェアの構成要素間の依存関係も規模が大きくなるほど非線形に増大します。何千万行という規模のシステムがごろごろある現状からみても、この問題は深刻です。

同調性

conformity

ソフトウェアはソフトウェア単独で存在しているわけではなく、ハードウェアやネットワーク、他のシステム、人間の行動や習慣にいたるさまざまなものと関係を持ち続けるものであるということです。ソフトウェアの宿命は、こういった外部に常に同調(順応)し続けなくてはならないところにあります。

可変性

changeability

ソフトウェアは常に変化し続けなくてはならないということです。たとえ、当初の計画通りシステムが出来上がったとしても、それを使っているユーザはさらなる要求を思いつくでしょうし、そもそも出来上がったシステムが世界を変えてしまうため、人間の認識にも影響を及ぼし、新たな要求がでてくることになります。

不可視性

invisibility

ソフトウェアが複雑な概念の集積であることに起因して、それが目に見えないということです。開発プロセスや意思決定の経緯も見ることができませんし、単純な図面で全体が理解できるということもありません。

ソフトウェアをつくるということは、それがビジネスであろうと、エンジニアリングであろうと、芸術であろうとも、「実行」されるもの抽象的な機械であるといふのは、重要な特性です[Otsuki2009]。

実行可能性

executable

最終的にはプログラムコード、人による実践といった形で実現されるものと関係しています。抽象機械、計算、変換といったオートマタ(機械)やメカニズム(機構)が必要となります。

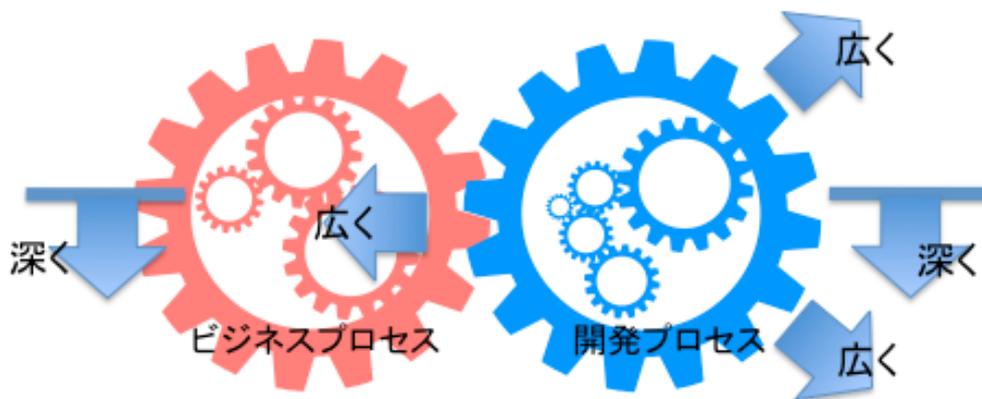
ソフトウェアを対象としたデザイン論を展開していく場合には、上記の5つの性質を考慮していく必要があると考えられます。

〔知働化〕

ソフトウェアの開発方法に関する学問領域は「ソフトウェアエンジニアリング」として1970年代から発展してきたわけですが、仕様を予め決めてフェーズ分けを行い、組織的にソフトウェアを開発していくウォーターフォール型の開発プロセスが主流になっています。近年、ビジネスや社会的環境の変化が激しくなってきたために、より俊敏で柔軟なアジャイル型開発プロセスが台頭して来ています。2009年夏に立ち上げた「知働化研究会」は、こういった流れを受けて活動しています。「知働化」というのは造語ですが、ソフトウェアづくりでは「知が働く」ことによって実行可能な知識（Executable Knowledge）が構築されていくことを表わしています。本小論で紹介したデザイン論やイノベーション研究の射程とも重なっています【Chidouka2010a】。

〔研究・探求の方向性〕

こういった研究・探求活動の方向性は、簡潔にまとめるならば、以下のようになるでしょう。



- ・ 従来のソフトウェア開発に閉じた世界から脱して、ビジネス、利用者、社会といった広い観点に移行していくこと。
- ・ ソフトウェアづくりや利用に関する人間の知的活動、実践知、認識、主観、共感の世界に深く入り込んでいくこと。
- ・ 組織、チーム、社会、制度のあり方を見直すこと。

〔新ソフトウェア宣言〕

ソフトウェアに関わる取り組み方やアプローチが変化している状況を、簡潔にまとめたものが、2010年のソフトウェアシンポジウム 2010 のソフトウェアエンジニアリングの呪縛 WG の『新ソフトウェア宣言』です[SS2010][Chidouka2010b]。

新ソフトウェア宣言

Declaration of New Software

従来のソフトウェアに関わる諸活動の呪縛から解放され、新たな世界を築いていくためには、新たな呪縛に捕われ、新たな中心を設定していかなくてはなりません。我々は、以下の関心の候補が有望であると確信しています。

In order to build a new world, freed from the curse of activities related to traditional software field, we shall be caught in a new curse, it does go to a new paradigm will set. We are strongly confident that the following are promising candidates for the matter of concern.

1. ソフトウェアは、数学的理論探求の上に成り立つ
Software is built upon exploration of mathematical theory.
2. ソフトウェアは、部分に還元することが不可能な全体である
Software can not be reduced to parts.
3. ソフトウェアは、実行可能な知識である
Software is executable knowledge.
4. ソフトウェアは、学びの副産物に過ぎない
Software is just a by-product of learning.
5. ソフトウェアは、制約条件下で創造される美しい人工物である
Software is a beautiful artifact, created under some constraints.
6. ソフトウェアは、富を生む経済活動の資源である
Software is a resource of wealth-creating economic activities.
7. ソフトウェアは、言語ゲームである
Software is a language game.

上記、基盤化、全体論、知働化、進化論、意匠論、経営論、遊戯論などの広範で、かつ、相互補完的アプローチが、新しい物語を生み出し、新たな地平を切り開いていくことでしょう。

Above, the mathematical infrastructure, Holism, executable knowledge / texture, evolution, design theory, management theory, and language game, those are broad and the complementary approaches, creating a new story, we will open up a new horizon.

大槻繁, 芝元俊久, 高野明彦, 竹内雅則, 時本永吉, 夏目和幸, 萩原正義,
羽生田栄一, 濱勝巳, 本橋正成, 山田正樹, 綿引琢磨

Shigeru Otsuki, Shibamoto Toshihisa, Akihiko Takano, Masanori Takeuchi, Eikichi Tokimoto, Kazuyuki Natsume, Masayoshi Hagiwara, Eiichi Hanyuda, Katsumi Hama, Masanari Motohashi, Masaki Yamada, Takuma Watabiki



知働化研究会

2010年6月11日 横浜にて
June 11, 2010 at Yokohama

<http://www.exekt-lab.org/Home/newssoftdecl> (2010.6.5)

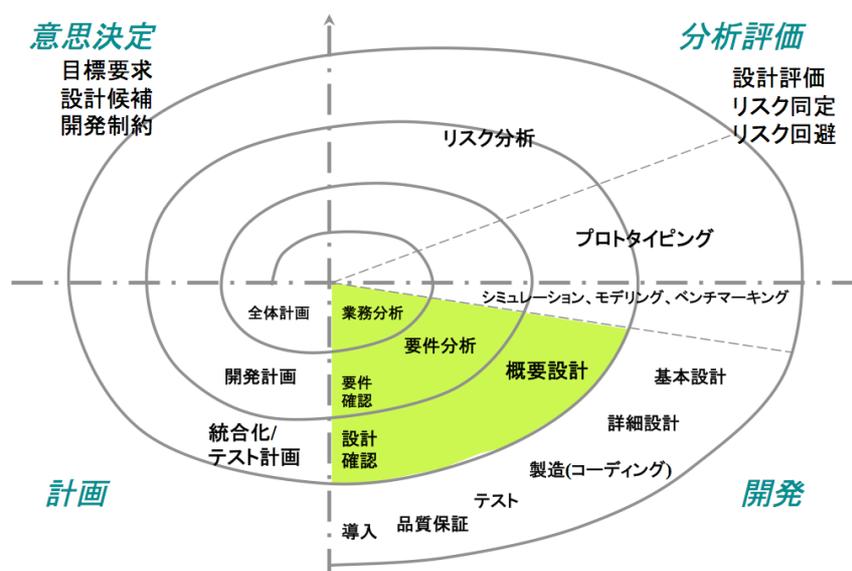
5. 2 ソフトウェアデザイン論

〔ブルックス,Jr.の主張〕

ブルックス,Jr.の『デザインのためのデザイン (The Design of Design)』[Brooks2010]では、「デザイン」を主題とした随想が掲載されています。1975年の著作『人月の神話』[Brooks1975,1995]では、工業的、組織的なソフトウェア開発が主題で、今で言う「プロジェクトマネジメント」の必要性を説いたものです。35年を経て2010年に出版したものは、「デザインプロセス」を主題としています。デザインの対象は、ソフトウェアに留まらずコンピュータハードウェアや建築も扱っていますが、中心にあるのは、『人月の神話』と同じソフトウェアです。

デザインプロセスについては、ブルックス,Jr.自身が著作の中で簡潔な要約をしています。

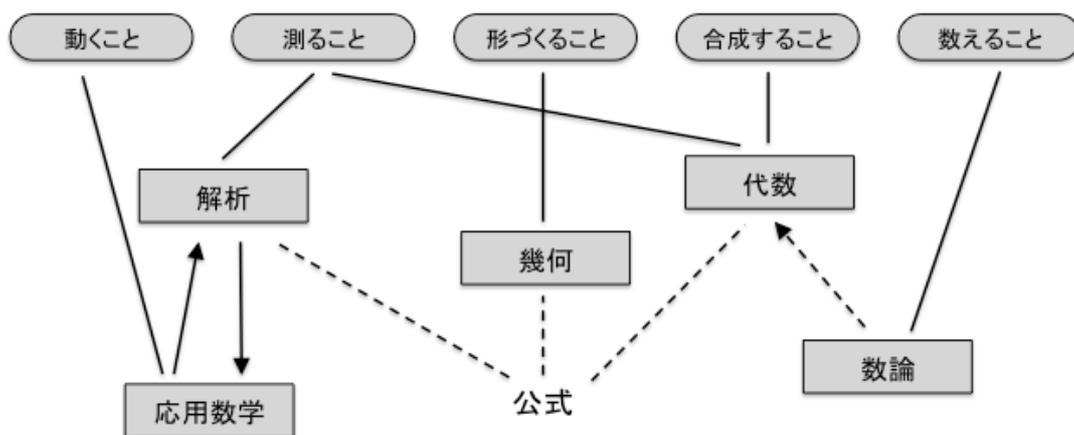
- ・ デザインプロセスモデルは必要で、仕事の体系化、プロジェクト内でのコミュニケーション、人材育成に役立ちます。
- ・ デザインプロセスモデルが幾何学的形状を持つことは、デザイナーが空間的思考を得意としていることから重要です。
- ・ デザインプロセスの合理的モデルは、エンジニアの頭の中で自然に生まれてくるものです。数多くのモデルが提唱されてきました。
- ・ 線形で、段階的なモデルは、実践者から見ると乖離しているが、契約上の制約から単純であるために使われてきました。
- ・ Barry Boehm のスパイラルモデルは有望と思われます。



〔概念構造体〕

さらに、ブルックス,Jr.の慧眼は、ソフトウェアというものが「概念構造体」であると主張しているところにあります。ソフトウェアづくりでは、抽象的な概念を扱い、これをプログラムコードに変換していくという知的な作業が中心になります。第2章で紹介した「デザイン論」では、メタファ、アブダクション、転写、抽象概念演算（概念生成）は、ソフトウェアづくりでも行われます。ソフトウェアの難しさは、これ等がさらに、コンピュータサイエンス、計算の理論、プログラミング言語の諸概念に変換・対応させていかななくてはならないところにあります。

おそらく、イメージ・スキーマのソフトウェアデザイン版を整備していくことが、有効な処方箋の一つになっていくのではないかと筆者は予想しています。第2章の末尾に掲げたクロフト等[Croft2004]のイメージ・スキーマを眺めてみると、ある程度の部分は数学の概念[Mclane1985]と対応させることができます。無論、全てが形式的な分析や記述が可能ではありませんし、実践知のように暗黙知を含む領域も存在します。



S.マクレーン『数学：その形式と機能』より

ブルックス,Jr.は、優れたデザインは、デザインプロセスから生まれるのではなく、優れたデザイナーから生まれると主張しています。いわゆる、デザインコンセプトのような一貫性や完全性が求められる部分は一人で行うべきだと言っています。しかし、大部分のプロセスはチームの共同作業が中心になります。この段階ではデザインプロセスが重要な役割を果たしますし、諸概念を正確に伝達するためにも形式的、あるいは、数学的な手段が必要になります。

5. 3 ソフトウェア哲学

〔理論や技術を基礎付けるもの〕

ソフトウェアというものが概念構造体であり、目に見えないということになると、それをどのように記述するか、記述されたもの（モデル）が正しいことをどのように確認するのかといった問題がでてきます。デザイン論での、抽象概念演算や、デザイナーの頭の中の思考プロセスの妥当性はどのようにして担保されるのでしょうか。形式化できない知識、暗黙知、思考等は、どのように伝達、伝承、了解されていくのでしょうか。物理学のように実験による検証が難しく、反駁不可能な命題は意味があるのでしょうか。こういったソフトウェアに関わるエンジニアリングやサイエンスの下支えをするために「ソフトウェア哲学」とでも言うべき学問領域が必要になってきているのではないかと筆者は考えています。

イノベーションとは何かという問いもあるかもしれませんが、ソフトウェアの領域では、「パラダイム」という用語が援用されています。ウォーターフォール型開発プロセス、アジャイルプロセス、オブジェクト指向技術、形式手法等、数多くの開発パラダイムが提唱されてきています。大同小異という声もありますが、それぞれ考え方、手順、文化が根付いており、一度染まってしまうと、他への移行が難しくなる傾向があります。

〔合理論と経験論〕

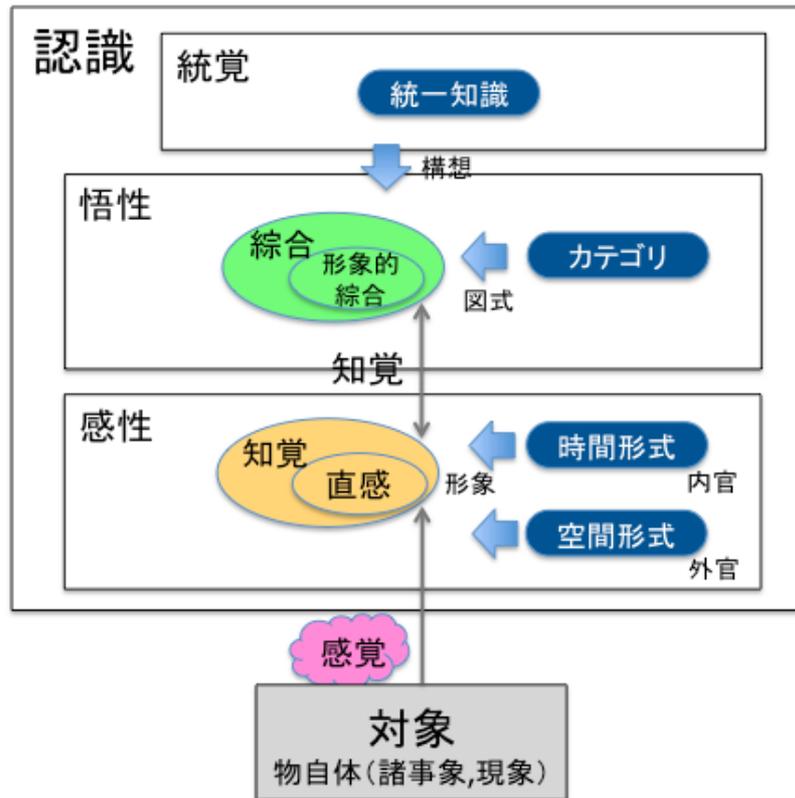
ソフトウェアエンジニアリングの歴史を紐解くと、正反合とか、合理的アプローチと経験的アプローチが対峙し、次第にそれ等が統合されていくという流れがあるように思えます[Watanabe2010]。

トップダウン設計法（段階的詳細化法や複合設計法）に対してボトムアップな方法が対峙し、次第に統合されていきました。ウォーターフォール型開発プロセスのように、予めプロセスを明確に定義して手順化していく方法に対し、アジャイルプロセスのように実践に重点を置く方法が対峙しています。

デザイン論でも、第2章で紹介したように「SIT アプローチ」が合理的アプローチで、それに対峙する「状況アプローチ」が経験的アプローチということになると思います。

次図は、カントの認識論モデルです。これは第2.4節で述べた〔意味論的転回〕の感覚・行為・意味のモデルと整合させることができると考えられます。デザイン

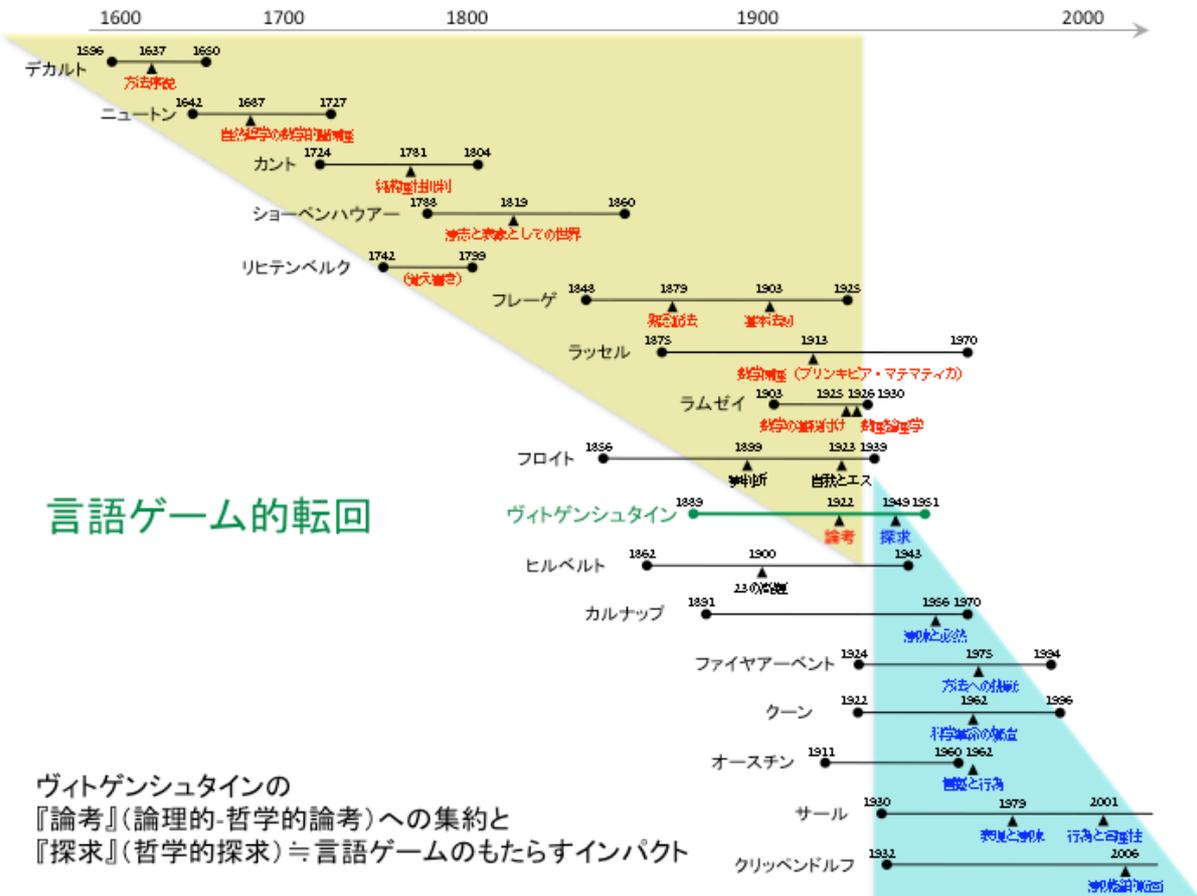
されて実世界に投入された人工物としてのソフトウェアは、その利用者に対して認識の対象として位置づけられます。それがもたらす知覚（直感）は、人間の本来備わった知的能力による総合（継承的総合）とが一致することによって知覚されます。また、その一致が起こるように意味の修正も行われることとなります。



さらに、知的能力の源泉としての統一知識やカテゴリがどのように規定されるのかという問題は、深淵な哲学的な問題です。筆者のその拠り所を次項目として掲げる「言語ゲーム」が向いていると考えています。

〔言語ゲーム的転回〕

クリッペンドルフの『意味論的転回』では、デザイン論に関する「哲学」として、ウィトゲンシュタインの「言語ゲーム」[Kurosaki1980]に基づいていると明言しています。これによって人間の主観的な感覚や認識を対象とすることができます。



大槻『 ΔV モデル：V字モデルからの意味論的転回』(2010.1.1) より

ソフトウェアエンジニアリングというのが、プログラムコードの記述に限らず、あらゆる活動を<言語活動>や<言語現象>であるとみなすという立場は、とても重要だと私は考えています。無論、無意識や心理的な事項もあることは認めますが、最終的には言語として現れることになります。こういった哲学的な基礎付けをしたのは、20世紀初頭に活躍した奇才、ウィトゲンシュタインです。彼の後期哲学を『探求』とか『言語ゲーム』の哲学と称しますが、筆者はこの思想は、ソフトウェアの世界にも当てはまると考えています。

- ・ ソフトウェアが「知」の織物であるとすれば、「知」とは何かを問い、「知」を実行可能な機械(マシン)として実現していく手法を探求していかなくてはなりません。

- ・ ソフトウェアのコンテキストは、多様な文化や理論体系であり、その中でのソフトウェアの意味をデザインしていく必要があります。
- ・ ソフトウェアに関わる活動は、言語活動であり、主知主義、言語ゲームとしてとらえるのが、自然な方向と考えられます。
- ・ 問題フレーム、ドメイン、ディスコースなどの知働化アプローチを探求していく際に、通奏低音として、探求を支えるコンセプトの一つが「言語ゲーム」という位置づけになるでしょう。

一言で筆者の主張を集約するならば、「ソフトウェアエンジニアリングの世界で言語ゲーム的転回を進めよう」ということです。新しい世界を描く努力なくして、未来は切り開けません[Otsuki2010]。

6. おわりに

〔結言〕

本小論では、実践的な手法から哲学的な事項まで採上げました。全体を俯瞰すると以下のようなになるでしょう。科学、技術、芸術、哲学等に渡り、横断的で学際・業際的色彩が濃く、多くの手法が提唱されています。大局観としての方向性は、次の3点に集約されているのではないかと考えています。

- (1) 理論や法則で定式化できない「主観」「思考」「暗黙知」「実践知」を扱う
- (2) チームや組織改革、言説（ディスコース）、コンテキスト、規範といった社会的事項に重点を置く
- (3) 産業構造の変革、組織改革、パラダイムシフトといった哲学や価値観の変化を伴う

上記をソフトウェアの領域で取り組んで行くべき事項を掲げるとすると、(1)については、ソフトウェアの特徴(本質的困難や実行可能性)を視野に入れた概念・認知・思考過程と、それを実践していく手法を確立していかなくてはなりません。

(2)については、開発チーム内のマネジメントのみならず、ソフトウェアが作用する実世界の振る舞いを分析・制御していく必要があります。(3)については人々の認識の変化、成長、進化といった事項を総合的に扱う理論や実践体系がのぞまれています。

ソフトウェアづくりの世界が、創造的な「デザイン」活動中心になっていくことは間違いありません。社会的インパクトのあるソフトウェアを構築・維持していくためには、手法(方法論)とマネジメントが必要です。ただし、従来の伝統的な工程による手順化、フェーズ分割、役割分担の方法ではなく、人間の創造的な能力を最大限にはっきできるような思考方法、認識・分析方法、モデル化(定式化)の方法、チームの協同・協調方法を整備していかなくてはなりません。筆者は、こういった創造的ソフトウェアづくりの知識体系を「超ソフトウェアデザインフレーム」と呼んで、少しずつ検討を進めています。

〔参考資料〕

第1章

- [Brooks1975,1995] Frederick P. Brooks Jr., *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley, 1995.
人月の神話—狼人間を撃つ銀の弾はない, ピアソンエデュケーション, 2002
- [Brooks2010] Frederick P. Brooks Jr., *The Design of Design: Essays from a Computer Scientist*, Addison-Wesley, 2010.4.1
デザインのためのデザイン, ピアソン桐原, 2010.12.25

第2章

- [Nagai2010] 永井由佳里, デザイン論について, 電子情報技術産業協会/ソフトウェアエンジニアリング技術分科会講演資料, 2010.7.25
- [TauraNagai2010] 田浦俊春, 永井由佳里, デザインの創造性と概念生成, *Cognitive Studies*, 17(1),66-82, 2010.3
- [Rowe1991] Peter G. Rowe, *Design Thinking*, The MIT Press, 1991.2.14
- [Simon1996] Herbert A. Simon, *The Sciences of the Artificial, 3rd Edition*, The MIT Press, 1996.10.1
- [Jackson2001] Michael Jackson, *Problem Frames: Analyzing and structuring software development problems*, Addison-Wesley, 2001
プロブレムフレーム: ソフトウェア開発問題の分析と構造化, 翔泳社, 2006.5
- [Schon1983] Donald A. Schon, *Reflective Practitioner: How Professionals Think in Action*, Maurice Temple Smith Ltd., 1983.12.31
- [Visser2006] Willemien Visser, *The Cognitive Artifacts of Designing*, CRC Press, 2006.8.8
- [Krippendorff2006] Klaus Krippendorff, *The Semantic Turn: a new foundation of design*, Taylor&Francis Group, LLC, 2006
意味論的転回: デザインの新しい基礎理論, 星雲社, 2009.4.1
- [Osgood1952] Charls E. Osgood, *The Nature and Measurement of Meaning: Semantic Differential Technique*, Chicago: Aldine Publishing Company, 1952
- [Lakoff1987] George Lakoff, *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*, University of Chicago Press, 1987
認知意味論—言語から見た人間の心, 紀伊國屋書店, 1993
- [Croft2004] William Croft, D. Alan Cruse, *Cognitive Linguistics*, Cambridge University Press, 2004.1.17

第3章

- [Kita2010] 喜多千草, 「革命的」技術はいかに芽生え定着したのか, 電子情報技術産業協会/ソフトウェアエンジニアリング技術分科会講演資料, 2010.11.5
- [Kita2008] 喜多千草, イノベーションプロセスの思想的記述モデルについて, 情報処理学会論文誌, Vol.49 No.4, 1594-1601, 2008.4

第4章

- [Brown2006] Tim Brown, *Innovation Through Design Thinking*, MIT World Distributed Intelligence(Video), <http://mitworld.mit.edu/video/357>, 2006.3.16
- [Brown2008] Tim Brown, *Design Thinking*, Harvard Business Review, 2008.1, pp.85-92
<http://www.unusualeading.com/wp-content/uploads/2009/12/HBR-on-Design-Thinking.pdf>
- [Brown2009] Tim Brown, *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*, HarperBusiness, 2009.10.1
デザイン思考が世界を変える—イノベーションを導く新しい考え方, ハヤカワ新書 juice, 2010
- [Kelly2001] Tom Kelly with Jonathan Littman, *The Art of Innovation – Lessons in Creativity from IDEO, America’s Leading Design Firm*, Crown Business, 2001.1.16
発想する会社! 世界最高のデザイン・ファーム IDEO に学ぶイノベーションの技法, 早川書房, 2002
- [Kelly2005] Tom Kelly with Jonathan Littman: *The Ten Faces of Innovation: IDEO's Strategies for Defeating the Devil's Advocate and Driving Creativity Throughout Your Organization*, Doubleday, 2005.10.18
イノベーションの達人 発想する会社をつくる 10 の人材, 早川書房, 2006
- [Todai2010] 東京大学, 東大式 世界を変えるイノベーションのつくりかた, 早川書房, 2010.5.21
- [Senbe2004] Peter M. Senbe, et.al., *Presence: Human Purpose and the Field of the Future*, Society for Organizational Learning, 2004.3
出現する未来, 講談社, 2006.5.30
- [Scharmer2007] C. Otto Scharmer, *Theory U: Leading from the Future as Emerges*, Berrett-Koehler Pub, 2007.1
U理論: 過去や偏見にとらわれず、本当に必要な「変化」を生み出す技術, 英治出版, 2010.11.25

- [Nonaka1995] Ikujiro Nonaka, Hirotaka Takeuchi, *The Knowledge Creating Company*, Oxford Univ Pr., 1995.5.18
知識創造企業, 東洋経済新報社, 1996.3
- [Nonaka2008] Ikujiro Nonaka, Hirotaka Takeuchi, *Managing Flow - A Process Theory of the Knowledge-Based Firm*, Palgrave Macmillan, 2008.8.15
流れを経営する : 持続的イノベーション企業の動態理論, 東洋経済新報社, 2010.7.8
- [Nonaka1986] Ikujiro Nonaka, Hirotaka Takeuchi, *The New New Product Development Game*, Harvard Business Review, 1986

第5章

- [Otsuki2009] 大槻繁, ソフトウェア開発はなぜ難しいのか : 人月の神話を超えて, 技術評論社, 2009.10.21
- [Chidouka2010a] 知働化研究会, 知働化研究会誌 Volume 1, 2010.11.3
<http://www.exekt-lab.org/Home/exerev>
- [Chidouka2010b] 知働化研究会, 新ソフトウェア宣言, 2010.6.11
<http://www.exekt-lab.org/Home/newsoftdecl>
- [SS2010] ソフトウェア・シンポジウム 2010, ソフトウェアエンジニアリングの呪縛 WG, ソフトウェア技術者協会
<http://www.oide-osaka.org/ss2010/wg1.html>
- [MacLane1985] Saunders Mac Lane, *Mathematics, Form and Function*, Springer, 1985.12.19
数学 : その形式と機能, 森北出版, 1992.4
- [Watanabe2010] 渡辺嘉二郎, カントがつかんだ、落ちるリンゴ —観測と理解—, オーム社, 2010.3.15
- [Kurosaki1980] 黒崎宏, ウィトゲンシュタインの生涯と哲学, 勁草書房, 1980.10
- [Otsuki2010] 大槻繁, \wedge V モデル : V 字モデルからの意味論的転回, 知働化研究会誌 Volume 1, 2010.11.3
(<http://www.exekt-lab.org/Home/otsuki/philosophy/lambdav>, 2010.1.1)

〔謝辞〕

本小論は、2010年の元旦に『ΛVモデル：V字モデルからの意味論的転回』を公開して以来、ビジネス、エンジニア、研究領域の方々と議論をしてきた事項に、JEITAでのヒアリング内容、関連したテーマの調査、ブルックス,Jr.の著作などから得られた知見によって肉付けしたものです。

第2章は永井由佳里先生のご講演の内容をまとめたものです。第3章は喜多千草先生のご講演の内容をまとめたものです。両先生には、第2章、第3章について草稿段階でレビューをしていただきました。黒川利明氏より草稿段階で追加すべき事項をアドバイスいただき、それに従って第4章を追加いたしました。ご講演いただいた永井由佳里先生、喜多千草先生、そして、黒川利明氏をはじめとするソフトウェアエンジニアリング技術分科会のメンバの方々、さらに、このような執筆の機会を与えてくださった二木厚吉主査、片岡欣夫幹事、飯泉純子幹事にこの場をかりて感謝の意を表したいと思います。

書き下ろしの随想的なスタイルの小論ですが、さらなる、ご意見など賜うことができれば幸甚です。

〔本当のあとがき〕

2009年夏に知働化研究会を立ち上げた時に、アジャイルプロセスが実践段階に入り、世の中変わってきたなと肌身で感じ、その先はどうなるのかということに想いを馳せていました。当時、「価値駆動」「進化」といったキーワードでぼんやりと考えていたこと、「ソフトウェア経済学」という名でソフトウェアエンジニアリングと経営・経済を統合しようとい図っていたことなどが、少しずつ形になってきているように思えます。

今回のJEITAヒアリングまとめの機会が無ければ、ソフトウェア以外のデザイン論やイノベーション研究については勉強することは無かったと思います。こうして少し書き物にまとめることによって、自らの研究との関連も明確になりました。何よりも、分野は異なっても考えている人は考えているなど共感しましたし、とても刺激になりました。研究を前進させるモチベーションが得られましたし、本当の問題や課題に焦点が合っていく感覚を持つこともできました。

本章論で採上げたトピックスで言えば、デザイン論の体系の「理想追求型」やそ

れに伴う内的動機による抽象概念操作は、ソフトウェア開発では、今まであまり取り組まれてきていませんが、今後有望だと考えています。イノベーション研究については、「アジェンダ」という見方は産業発展の解像度を上げるために必要だということを理解できましたし、ソフトウェアの分野で特徴的な開発プロセスやオープンコミュニティを視野に入れたものを確立していく必要性を感じました。

本小論は、JEITA の年度末報告書というお決まりの成果であると同時に、今後発行予定の知働化研究会誌 Volume 2 の論文の原型にもなっています。さらには、[結言]の最後で述べた「超ソフトウェアデザインフレーム」をまとめる際の序論に相当するものになるかもしれません。その他いろいろ使うことを見込んで参考資料のリストは過大な編集まで施しています。

筆者は、新しい手法や考え方を世の中に仕掛けていく割には、相当保守的であることを自覚しています。価値駆動と言いつつ、実務のコンサルティングでは人月ベースの見積り評価をしていますし、アジャイルプロセスの普及・啓蒙をしていながら温故知新的なソフトウェアエンジニアリングの重要性を説いています。言語ゲームと言いつつ数学的な客観性を基軸にしています。ジャクソンの構造化プログラミング、システム開発法、問題フレーム等は普遍的な知見だと信じています。おそらく「超ソフトウェアデザインフレーム」も、基底部分は旧来の本質論をベースに、時代の要請による事項の指向性を伴った形でまとめていくことになるでしょう。

2011年1月吉日 横浜にて
大槻 繁

