

ソフトウェア工学に関する研究を発展させるためには

落水浩一郎
金沢工業大学 客員教授

要旨

「ソフトウェア工学は経験科学である」とする立場から、ソフトウェア工学に関する研究を発展させるための手段を、カール・ポパーの反証主義に基づいて検討する。

1. はじめに

ソフトウェア工学は経験科学である。経験科学とは「純粹に理論を探求する科学に対して、経験的事実を対象として実証的に諸法則を探求する科学(大辞林)」である。カール・ポパーの哲学、反証主義「客観的真理にせまるための試行錯誤は、問題の発生、試験的理論、誤りの排除、新たな問題の発生、という推測と反駁の過程の連鎖である」に基づいて考察する。なお、ソフトウェア作りとソフトウェア工学をここでは以下のように区別し、②のみを考察の対象とする。

- ① ソフトウェア作りは創作活動であり芸術活動の一種であろう
- ② ソフトウェア工学は、ソフトウェア開発・保守にあたっての労力を緩和するものであり、作成されるソフトウェアの品質を高めるためのものである。

2. 経験論(経験主義)の歴史

経験科学の方法論は長い歴史をもつ[1]。

- ① フランシス・ベーコンによる帰納法：感覚的観察を無条件で信用せず、実験という方法を駆使して、少しずつ肯定的な法則命題へと上がっていく
- ② 論理実証主義：実験による調査研究、帰納的推論、演繹的検証
- ③ カール・ポパーによる反証主義

筆者は長い間、観察を基に、帰納法によって理論に到達するというベーコンの帰納法を漠然と意識してきたが、仮説を立てそれを実験に基づき反証することによって理論を進化させていくという立場のカール・ポ

パーの著書を読み、ごく自然な考え方であるということに彼の考え方にそって考察することとした。

3. 科学の課題

以下、3章および4章の内容はすべて、カール・ポパー著書の訳書[2, 3, 4, 5]を引用することにより記述する。

カール・ポパーによれば、科学の課題は、一部は理論的なもの—説明—であり、一部は実際的なもの—予測および技術的応用—である。

説明とは図1に示すように、被説明項Eが与えられており説明項を求めることである。ここで、説明項は、普遍法則U(または、仮説、または理論)と特殊な初期条件Iからなる。被説明項とは観測された事実であり、複数の被説明項(反復しておこる同じ事実)からそれを説明する一般性(普遍性)を導出してよいとする立場が帰納法であり、カール・ポパーはこの立場を否定する。

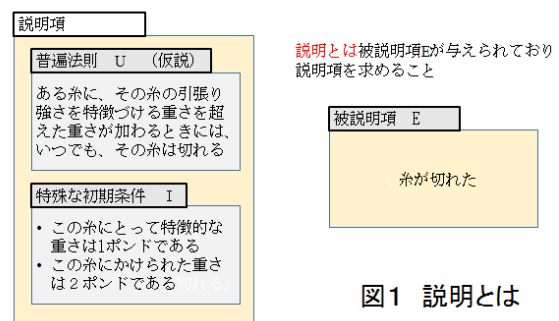


図1 説明とは

反証法においては、被説明項は仮説(または理論)のテストに使用される。すなわち、現実世界においておこる問題現象に対してそれを説明する仮説を導く、それを観測によって得られた事実(被説明項)とつぎあわせて確認する。仮説に反する事実が観測された場合は仮説(または理論)か特殊初期条件が間違っているので、双方またはどちらかを作りなおす。このように、初期の仮説を観測によって確認し、反駁によって仮説(理論)を進化させていく立場が反証主義である。以下のような手順を踏む。

P1(出発点となる問題)→
 TT(暫定的解決・暫定的理論)→
 EE(誤りの排除)→
 P2(より大きな深さを持ち、豊沃性に富む、新しい問題).

説明には**理論的説明**と**歴史的説明**がある。理論家は普遍法則を見出し、テストすることに関心を持つ。歴史家は、特殊初期条件の適切さまたは正確さをテストする(図2)。

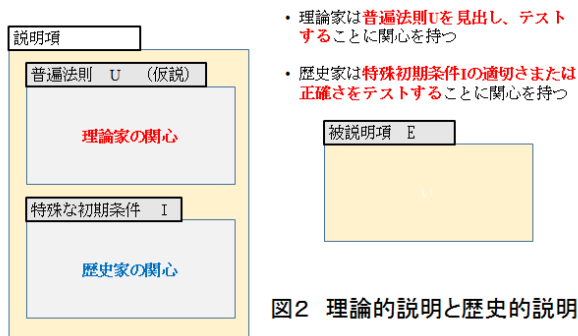


図2 理論的説明と歴史的説明

予測(図3)とは、理論Uが所与または既知であると仮定されており、また、特殊初期条件も観察によって知られているあるいは、知られ得ると仮定されている。見出されるべきものは論理的帰結、すなわち予測Eである。

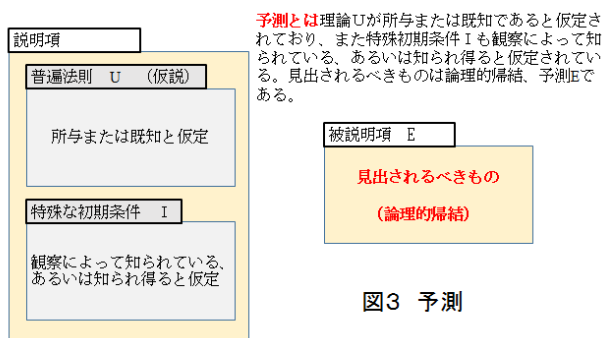


図3 予測

技術的応用(図4)とは、Eに相当する仕様明細書(顧客の仕様書)があり、これには一定の要求された事態が記述されている。ある種の実用的な目分量のやり方も含めて、関連のある物理学的諸理論Uも与えられている。われわれが見出さなければならないものは、技術的に実現でき、また理論と一緒にになって仕様明細書Sが導出できるような初期条件Iである。

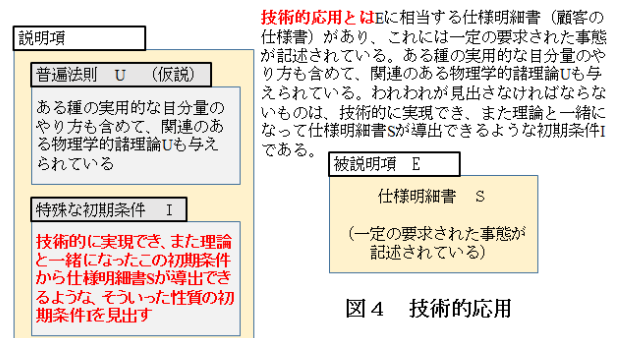


図4 技術的応用

4. テストの手続き

反証法においては、テストは重要な役割をもつ。テストの手続きは以下の通りである。

- ① テストとは、説明項からの予測Pの導出と、その予測を現実に観察可能な事態と比較することである。
- ② もし予測が観察された事実と一致しなければ、説明項は偽なることが立証される。つまり反証される。
- ③ この場合、反証されたのが普遍理論であるのか、それとも初期条件が偽であるのかはわからない。
- ④ 予測の反証は説明項が偽であることを立証するが、しかしその逆は成り立たない。つまり予測の実証をもって説明項を実証するものと解釈できると考えるのは間違いである。

5. ソフトウェア工学への適用

3章で紹介した考え方をソフトウェア開発方法論に適用してみる。説明、予測、技術的応用のどれを基本的な考え方として採用すべきかがまず問題である。ここでは、予測をもとに考察する。技術的応用の適用については今後の課題とする。また、ソフトウェア工学の種々の分野のうち、ソフトウェア開発方法論を例としてとりあげる。

図5に示す図式に従って、仮説や特殊な初期条件の妥当性を検証する手段を検討する。仮説そのものの妥当性を議論したいのではないことに注意して欲しい。仮説には、方法論の効用としてわれわれが期待している事柄を記述し、初期条件には、

該当する方法論の適用条件を記述する。被説明項には、実際の開発活動の結果得られた事実を記述する（図5）。

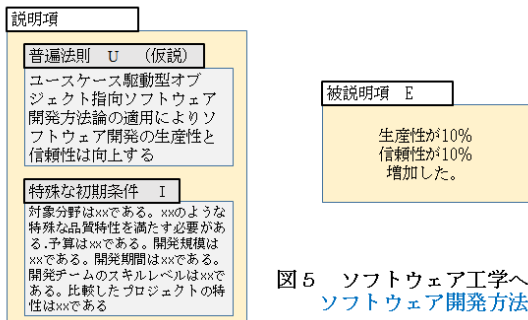


図5 ソフトウェア工学への適用ソフトウェア開発方法論

予測においては、被説明項は説明項の論理的帰結でなければならない。ソフトウェア工学への適用の図式においては、説明項から被説明項を論理的に帰結するのは困難であるので、論理的帰結（予測）は実際の開発活動の結果得られたものとする。

実際の活動は、仮説が成立する特殊な初期条件を精錬する活動になるものと思われる。

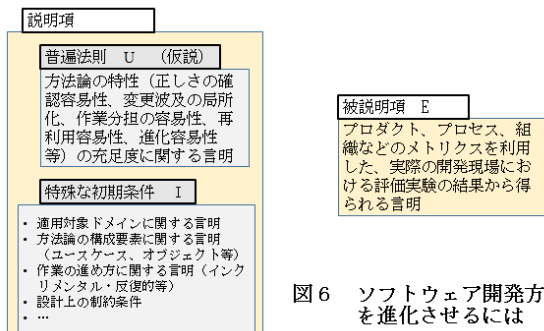


図6 ソフトウェア開発方法論を進化させるには

図5の記述はひとつの考え方であるが、成果が特定の**方法論・ドメイン**に依存しすぎるかもしれない。ソフトウェア開発方法論の発展のためには、「**ソフトウェア開発方法論の進化の手段を提供する**」という立場のモデル化がより重要であると考えられる。すなわち、現在利用中の方法論を改善する立場ではなく、よりよい方法論を求める立場である。図6に一つの考え方を示す。

もちろん図6の定義内容そのものは、いまだ十分なものとはいえず、今後の課題である。

6. まとめと課題

経験にもとづいて、皆にとって有益であるよう

な法則性を発見するにはどのように思考すべきであろうか。「ちょっと考えてうまくいった、みんなにひろめよう」、「ちょっとやってみたらうまくいった、みんなにまねさせよう」はレベルの低いアプローチである。これに対して、カール・ポパーの哲学をソフトウェア工学研究の背景哲学として採用することは有益であると考えられる。すなわち、「ちょっと考えてみてうまくいった、確かめてみよう」、「ちょっとやってみたらうまくいった、本当にそうか確認してみよう」のように考え方を換えれば、誰かが定義した方法論の良し悪しを論議するだけの立場から脱却できる。すなわち、

- ①. 問題状況を解決する仮説を立てる(初期の説明項)
- ②. 初期条件を明らかにすることにより、理論の適用範囲を明確にし
- ③. 予測と実験による反駁を繰り返すことにより、よりよい理論に仕上げていく

ことが可能になる。よく考えてみると以上の事柄は皆様が日常行われていることではないかと推察する。それぞれが関心のある分野(形式手法, 要求定義, 設計, プログラム言語, プロジェクト管理, メトリクス等)で考え方(攻め方)を明示的に整理し、皆で共有することは有用であると考えられる。

このとき、テストは、組織や会社の枠組みを超えて実施する必要があるであろう。テストを実施するための場の提供と共有は、ソフトウェア技術者協会の一つの役割ではなからうか。

参考文献

- [1] WikiPedi 「経験論」<http://ja.wikipedia.org/wiki/経験論>
- [2] カール・ポパー著, 大内 義一・森 博 訳, 「科学的発見の論理 (上)」恒星社厚生閣, 2010.
- [3] カール・ポパー著, 大内 義一・森 博 訳, 「科学的発見の論理 (下)」恒星社厚生閣, 2010.
- [4] カール・ポパー著, 森 博 訳, 「客観的知識—進化論的アプローチ—」, 木鐸社, 1999.
- [5] カール・ポパー著, 藤本 隆志, 石垣 壽郎, 森 博 訳, 「推論と反駁 科学的知識の発展」, 法政大学出版局, 2014.