

ソフトウェア/システム開発でのドメイン分析・モデリングの使い方

ドメイン分析・モデリングの研究の進め方

ソフトウェア工学危機とドメイン分析

阿草 清滋 名古屋大学情報工学科

1 はじめに

ソフトウェア工学の歴史はすでに四半世紀になろうとしているが、その研究成果によりソフトウェアシステムの開発の諸問題が解決したとはだれもが考えていない。それどころかソフトウェア工学の研究そのものに疑問が投げかけられている [1]。

ソフトウェアシステムは汎用で単純な機能を持つハードウェアと、個々に異なる形態を持つ実社会の要求を橋渡しするものであり、その多様性は単純に一つの手法で解決できる問題ではない。優れた手法も一つのハードウェアとみなせ、一つの開発はまたそのソフトウェアとなっている。ソフトウェアの研究は対象を分析し、分析された論理構造を明らかにするものと考えられがちであるが、経験的な評価を求めるために多くのフィールドワークを必要とする。しかし、ソフトウェア工学者は実用規模のソフトウェア開発での研究の適用を行わず、その適用は現場の問題としている。このため研究者と実践家（ソフトウェア開発のリーダー）の間でのソフトウェアのとらえ方にずれがあり、互いの上に大きな溝ができていく。ソフトウェアの開発能力は個人差が大きいことはよく知られているが、生来の能力とともに経験の差が大きい。これはソフトウェア開発が大きく分けて2つのフェーズに分けられ、特に、その前者の問題解決に経験を必要とするからである。2つのフェーズとは応用ドメインから概念モデルを得る過程と、形式モデルからインプリメンテーションドメインに変換する過程である。前者は論理現行モデルの獲得過程と呼ばれることもある。後者は論理将来モデル

から実現システムの構築である。この前者の過程の重要性の主張はソフトウェア工学の発端の一つであり要求工学と呼ばれた。ドメイン分析は正確な要求の獲得に必要不可欠のものである。しかし、正確な要求が本当に獲得できるのか、またできたとして本当に求められるシステムが構築できるかは別の問題である。すなわち、概念モデルと形式モデル間のDijkstraも指摘している「知的ギャップ」が人間の極めて精緻な創造活動によって越えざるを得ないからである。ソフトウェアの問題点を謙虚にとらえ、できることは何かを明確にしていくことが大切である。

2 プロダクトからプロセス、そして

ソフトウェアの見方にはいくつかの軸がある。計算とは何かを明らかにし、その定義の抽象レベルを軸とするものがあり、ソフトウェア科学の目から見るものである。また、ソフトウェア開発過程の時間進行の軸があり、ソフトウェア工学の目から見るものである。ソフトウェア工学研究の流れはソフトウェア開発におけるプロダクトの管理からプロダクト間の変換を定義するものに変化してきている。また、流れは下流工程から上流工程に向かっている。すなわち、プログラムの構造はどうあるべきかという研究から、仕様書はどうあるべきか、要求定義はどうあるべきかに移ってきた。プロセスとしてもプログラミングのプロセスから要求分析・仕様化のプロセスへと進んできた。ソフトウェア工学の一つの側面はプロセスとプロダクトを統合的に管理することで、

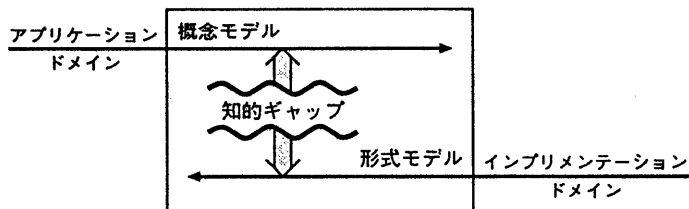


図 1: 知的ギャップ

属人性を排除し制御された品質のソフトウェアを実現可能とすることであった。このようなアプローチでソフトウェア工学の危機と呼ばれる状況が訪れたのは、もう一つの軸である情報化の成熟度の軸を無視してきたためである。計算機が科学計算の道具として使われてきたことから、現在のように我々の思考過程、創造過程を探る道具として利用されようとしている状況まで、ソフトウェアの対象が変化してきているので、その開発過程も変化せざるをえない。この状況に目をつむり、ある特定の領域のしかも固定的な問題に対するモデルの提示と、そのモデルの有効性を議論することが多い。ソフトウェア工学で「要求さえ正確に定義してもらえば」との前提が仮定し得ないことが分かった。人間はソフトウェアシステムが導入されることで自分自身を変化させるため、要求分析時と異なる利用者となるからである。ドメイン分析も対象領域に固有な概念構造を得ることを目的としているが、固有の、また安定している概念構造で、その領域のあらゆるシステム開発に利用できるものが存在しうるかは疑問である。ドメイン自身がまた変化するものであるからである。ソフトウェア開発の本質的な難しさが「知的ギャップ」をいかに飛び越えるかである以上、ドメイン分析はこれまでも解決可能であった問題をより容易に解決する手段であり、これまでのソフトウェア工学のかかえてきている問題解決の切り札ではないであろう。また、ソフトウェアは対象が数限りなく存在することにより解決が難しくなっていることを考えると、ドメイン分析をいつまで続けると問題の本質的

な解決になるかは疑問として残る。ドメイン分析もまたソフトウェア開発手法の一つであるとする、その適用を通しての実践的な研究が求められる。

3 ドメイン分析と再利用

ソフトウェアの開発能力には個人差が大きく、また同じ個人でも同種のシステムの開発経験の有無によって開発の期間や成否が大きく作用される。経験は誰にでも伝授できるものではないが、その努力は必要である。プロセスプログラムは経験の記述であり、一意的に解釈されるものとなる。ソフトウェア開発プロセスの再利用が可能となる。ソフトウェアの効率的な開発はこれまでに得られた知見をすべて再利用することで達成される。ドメイン分析は要求仕様の定義に重要であるのみならず、その結果得られたドメインモデルはソフトウェア再利用にも重要である。ソフトウェア工学が目指す属人性の排除は、手法の形式化・定式化であり、形式化されれば計算機による実現も可能となる。すなわち、ソフトウェア工学の目指すところは自動プログラミングにほかならない。しかし、どのようなシステムにすべきかという欲求や理想は自動プログラミングシステムでは導出できない。このため限られた範囲の自動化を行い、瑣末で手数のかかる仕事が計算機支援のCASEとして自動化される。仕事の定式化から自動化は当然の流れであり、その定式化には何らかの記述系が必要となる。同じようなソフトウェアシステムを何度も開発することは無駄であり、共通部分と各システムの固有部分に分離し、その組み合わせで実現す

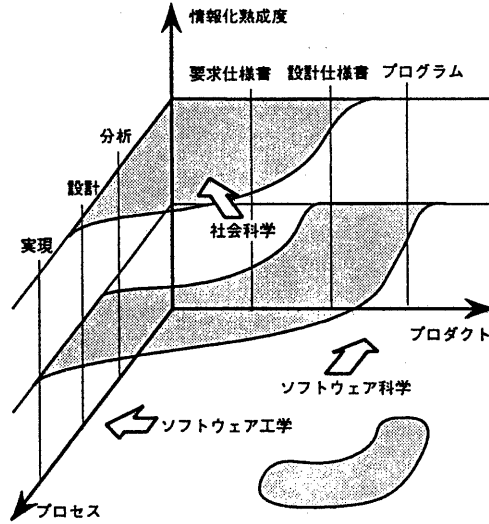


図 2: 情報化の成熟度

ることが考えられる。パッケージソフトを用いての開発はこれに当たる。共通部分には固有部分を取り込めるように、パラメータ化などを行う。このパラメータ定義としての固有部分定義を汎用の言語を用いて行うのか、あるいは専用の言語で行うのかの選択がある。また、パラメータ化の度合いが大きい時、POL(Problem Oriented Language) として言語の形で共通部が与えられる。ドメインモデルも言語の意味論に取り込むことによって、POL の中核になる。スプレッドシートは一種の POL であり、現在話題のエンドユーザコンピューティングの推進役となっている。これは帳票処理というドメインでの仕事の記述に向けた言語である。しかし、これで経営戦略を建てるためのシミュレーションを行うこともされる。人は保守的であり、一旦慣れた道具を器用に使いこなし、道具の提供者が考えもしなかった用法をするものである。この用法を許す汎用性が要求され、結局はドメインを限定できなくなる。ソフトウェア工学研究のアジェンダの中にいつも出てくるものに「開発過程のマニュアル化」がある。このマニュアルがドメイン毎に異なるものになるとする

と、まずはドメインの分類から行わなければならない。今から新たにその要求が発生する可能性があるシステムまで含めての分類は可能であろうか。コンパイラや OS のようにドメインが明確になった分野は独立した学問分野としてソフトウェア工学の分野とは一線を画してきた。医学の分野では多くの独立した研究分野があり、また人工知能の研究分野も機械翻訳、推論機構、画像認識など多くの分野を生んできた。この意味でソフトウェア工学の発展とは明確に定義されたドメインを切り離すことであるかも知れない。ある特定の技術ですべての問題を解決するのではなく、特化された技術の集団としてソフトウェア開発せよとの主張もある。しかし、ソフトウェアの持つ汎用性は次々に新たな問題を見つけ出し、いつまでもソフトウェア工学は進歩がないと見られがちである。この意味でソフトウェア工学が得た成果自身の再利用がもっとも必要とされている。それぞれの研究成果をコード化して誰もが使えるようにすることとともに、過去に作られた優れた遺産的ソフトウェア (Legacy Software) を解析し、そこから学ぶべきものを明確化する必要がある。この解析の

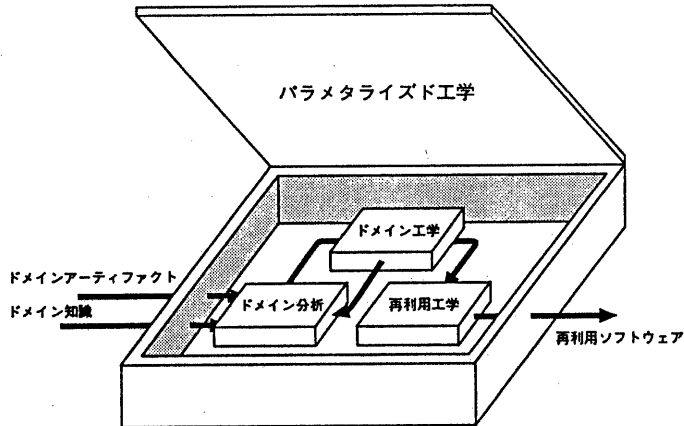


図 3: パラメタ化学

中でどのようなドメイン知識が使われているかに興味を持たれる。

4 これからのソフトウェア工学

ドメイン分析はソフトウェア開発のキーではあるが、万能薬ではない。ソフトウェア開発は多くの相対立する要求項目を調整する過程である。特定の評価項目の評価値を上げようとするとはそれはそれで可能である。この調整は開発者の価値観、利用者の価値観で異なるものである。この意味ではある対象領域の価値観を把握することは満足されるソフトウェア開発に重要であり、ドメイン分析の目的の一つに対象領域の価値観が含まれることを望みたい。ソフトウェア工学の軸に「情報化の成熟度」をあげたが、情報化の進展とは情報システムがより我々の身近なものになることである。これは要求される機能がより個人に近付き、個人の特性や環境に特化される機能が求められることになる。ドメイン分析の対象となる領域がますます細分化され、単に対象領域という言葉でくくれなくなる。その時、対象領域を定義できるのは、結局その利用者でしかあり得ない。これはソフトウェア工学の究極の目的である「自動プログラミング」そのものであり、その達成は限られたものとなろう。ドメイン分析技術にしてもその成

果は限定されたものであることを念頭に置いて、何をどこまで明確化するかははっきりしなければ、これまでの研究の焼き直しになってしまう。これまでのソフトウェア工学の研究の成果をより多くの技術者が利用できるようにする地道な研究と、これまでの限界を打破する新たな研究とのバランスのよい研究推進が求められる。過去の研究の遺産の上にこれからの研究があり、過去の研究を否定することで次の研究があるとすれば、ソフトウェア工学の研究成果を適用しようとしているソフトウェア開発現場の技術者の期待を裏切ることになる。

参考文献

- [1] R.L.Glass, "The Software-Research Crisis", *IEEE Software*, pp42-47, Nov. 1994
- [2] 田村他, "ドメイン分析・モデリング技術の現状と課題", *情報処理*, pp952-961, Oct 1994
- [3] B.I.Blum, "Software Engineering A Holistic View", *Oxford University Press*, 1992
- [4] A.S.Peterson, "Coming to Terms with Software Reuse Terminology: a Model-Based Approach", *ACM SIGSOFT*, Apr. 1991