

ソフトウェア再利用の管理的側面

磯田 定 宏†

多くのソフトウェア再利用の成功例が報告されている一方、期待されたほどには効果をあげていないという否定的な意見が少なくない。これは再利用の本質と、それからくる適用条件および期待される効果についてまだ共通認識がないことに起因している。本稿では、部品ライブラリ再利用運動の実践例を紹介し、この経験に基づき主として管理的側面からソフトウェア再利用の問題点を分析し、ソフトウェア再利用では技術的課題よりも管理的課題が占める比重の方が大きいことを示す。さらに、再利用を成功させるためには、上級管理者のコミットメント、良いドメインの選択、およびドメイン分析に基づく体系的な部品作成が必要であることを示す。

Managerial Aspect of Software Reuse

SADAHIRO ISODA†

While many successful software reuses have been reported, there are not a few negative critiques that reuse has not delivered yet as expected. This is due to the lack of consensus as to what software reuse actually is. This paper describes a four-year experimental software reuse project. The targets of reuse are program code modules stored in a common library. The project is first described in detail. The statistical results of the project are then shown. Some important aspects of software reuse are discussed mainly from the managerial aspects. Finally, it concludes that software reuse is inherently a managerial issue and that successful reuse requires the commitment of senior management, selection of good domains, and systematic development of reusable modules based on domain analysis.

1. はじめに

ソフトウェア再利用はソフトウェアエンジニアリングの基本とも言える重要な技術である。1980年代以降、主要な学会誌で頻繁にソフトウェア再利用に関する特集号が生まれ、ソフトウェア再利用に関する国際会議、ワークショップが開催されてきたのは当然と言えよう^{1),2)}。

ソフトウェア再利用は広義には、過去のソフトウェア開発で得た生産物および経験・知識を以降の開発で活用する技術である。コード、設計、要求仕様、開発プロセスなどが再利用の対象になりうる。このうち設計や要求仕様の再利用はCASEの普及に伴い現実となりつつある³⁾。開発プロセスの再利用はまだ研究の域を出ない⁴⁾。したがって、コードが依然として再利用の主たる対象である。

これまでに多くのコード再利用の成功例が報告されている。米Raytheon社ではCOBOLのビジネスアプリケーションについての標準プログラムパターン

と機能部品の組合せで15~85%の再利用率を達成した⁵⁾。国内でもほぼ同様な報告がなされている⁶⁾。しかし、報告されているほとんどのソフトウェア再利用の成功例がCOBOLのビジネスアプリケーション分野であり、他の分野への再利用技術の適用性についての議論はほとんどない。

少数の技術者が特定の分野を対象としてソフトウェア再利用を実践することはそう難しいことではない。しかし、多くの技術者が関与する環境では再利用の実践が難しくなる。つまり、組織レベルでソフトウェア再利用をいかに実施すべきか、そのときにどのような限界があるのかなどを明らかにすることこそソフトウェア再利用技術の課題である。しかし、これまでのソフトウェア再利用成功例の報告の中にはソフトウェア開発組織でどのように再利用を進めるべきかに関するものはほとんどない。

さらにより本質的な問題として、多くの再利用成功例が報告される一方で、「期待したほどには成果をあげていない」という否定的な意見が少なくないことがある。これは、再利用が成功するための条件や期待される効果などについて、十分な議論がなされていないため、いまだ共通認識を醸成するに至っていないこと

† NTT ソフトウェア研究所
Software Laboratories, Nippon Telegraph and
Telephone Corporation

を物語っている。

以上の3点について従来は検討が欠けていた。本稿ではこれらについて論ずる。まず NTT ソフトウェア研究所における4年間にわたる部品ライブラリ再利用の実践例を紹介する。これは従来から報告されているビジネスアプリケーション分野とは異なり、ソフトウェア開発支援ツール分野を対象とするものである。再利用運動を推進するために採用した体制とその運用方法、平均再利用率などの統計データ、およびその分析結果などについて述べる。また、この経験に基づき、主として管理的側面からソフトウェア再利用の問題点を分析し、ソフトウェア再利用では技術的課題よりも管理的課題が占める比重の方が大きいことを示す。さらに、再利用を成功させるためには、上級管理者のコミットメント、良いドメインの選択、およびドメイン分析に基づく体系的な部品作成が必要であることを示す。

2. 部品再利用運動の具体例

NTT ソフトウェア研究所では4年間にわたり組織的な部品ライブラリ再利用運動を実施した。4年間継続したことにより、ソフトウェア再利用の多くの側面を経験できた²⁾。

2.1 外部条件

対象分野は NTT ソフトウェア研究所がそのミッションとしているソフトウェア開発支援ツールであった。再利用対象はライブラリに登録されたコード部品であり、設計やテストケースなどの再利用は対象外とした。再利用運動の目的は、部品ライブラリ再利用のフィジビリティと限界を明らかにすることと、ソフトウェア再利用運動の方法論を確立することであった。

2.2 活動内容

(1) 枠組み

再利用運動の枠組みは次のとおりであった。

- ①再利用運動に関わった NTT のソフトウェア技術者は約100人。このほかに外注先の技術者約500人が関与した。
- ②毎年のソフトウェア開発量は約1メガ行。
- ③開発対象ソフトウェアは主として、コンパイラ、設計支援ツール、プロジェクト管理ツールなどのソフトウェア開発支援ツール。
- ④開発対象マシンはワークステーション、パソコン、および NTT のメインフレーム (DIPS)。
- ⑤プログラミング言語は主として C と SYSL (PL/I の

方言)。

(2) 体制

再利用運動の体制は、再利用委員会、再利用支援グループ、支援ツールグループ、およびソフトウェア開発グループから構成された (図1)。再利用委員会は各ソフトウェア開発グループの代表各1名から構成された。再利用委員会の責務は、再利用活動マニュアルやモジュール検出チェックリストの作成、ソフトウェア開発に携わる技術者にインセンティブを与える方法などのさまざまな課題を検討することであった。

再利用支援グループの責務は再利用運動全体を指導することであった。再利用委員会に対する助言、上級管理者への再利用運動の状況の報告、および、部品ライブラリの構築がその内容であった。したがって、再利用支援グループは再利用運動の中心的な役割を果たした。

(3) 活動内容

再利用運動は、部品の登録・再利用などの草の根運動を組織的に推進する一種の TQC 運動として実施した。主要活動は次のとおり。

- ①ソフトウェア開発過程での部品の登録・再利用。
- ②部品ライブラリの構築。
- ③再利用活動の作業標準および実施ガイドラインの作成。
- ④部品の管理、検索、利用を効率的に支援するツールの開発。

活動①は個々のソフトウェア開発グループが実施した。活動②と③は再利用委員会と再利用支援グループ

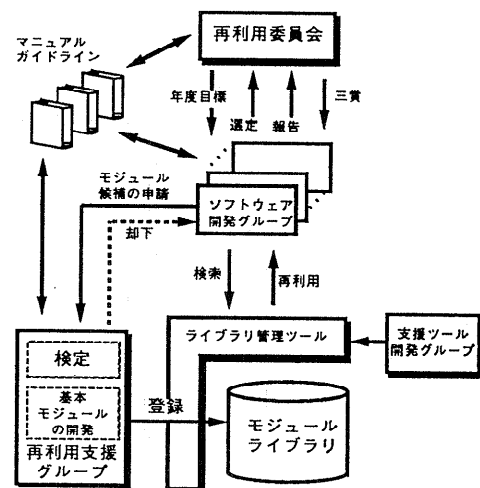


図1 ソフトウェア再利用の体制と情報の流れ

Fig. 1 Software reuse organization and activity flow.

が実施した。活動④は支援ツールグループの分担であった。

各ソフトウェア開発グループは高い再利用率が期待できそうなプログラムを選定し（重点プログラムと呼ぶ）、その開発状況を開発工程ごとに再利用委員会に報告した。再利用運動の2年目からは、再利用率と登録率（登録した部品の合計サイズを開発したプログラムの合計サイズで割った値）の目標値を設定した。各ソフトウェア開発グループはその目標の達成へ向け活動①を実施した。

(4) 部品の登録申請と検定

ソフトウェア開発グループは既存プログラムあるいは開発中のプログラムを分析し、部品候補を抽出した。部品検定チェックリストに従い必要な手直しを加えたのち再利用支援グループに提出した。再利用支援グループは、提出された部品候補に対して形式検定と機能検定とを実施した。形式検定では、モジュール登録票、プログラムコードおよび設計書が形式チェックリストの各項目に合致しているかを調べた。

形式検定チェックリストは22の必須項目と25の推奨項目から構成された。個々の項目は、モジュール設計、コーディング、ドキュメンテーション、およびモジュール信頼性の4種類に分類された。以下に例を示す。

①モジュール設計

- データとそれに関連するモジュールはパッケージ化されているか。
- モジュールは1入口と1出口を持つか。

②コーディング

- 系統的にインデントされているか。
- 個々の要素モジュールは100行以下か。
- モジュールおよび変数にはその意味を表す名前が付けられているか。

③ドキュメンテーション

- 機能の記述は既定義の用語を用いているか。

④モジュール信頼性

- モジュールは1キロ行当たり80件以上のテストケースでテストされているか。
- すべてのエッジがテストされているか。

機能検定では主観的な評価を行った。機能検定チェックリストは、非冗重性、一般性、オブジェクトの統一性（パッケージに対して）、オペレーションの完全性（同じくパッケージに対して）から構成された。検定に合格した部品候補は部品ライブラリに登録され、

また不合格だった部品候補はソフトウェア開発グループに返却された。

2.3 実施結果とその分析

4年間のデータを分析した結果、①平均再利用率、②再利用部品の平均サイズ、③再利用した部品の出所、④部品ライブラリの平均再利用頻度というソフトウェア再利用の4つの重要な側面について一応の成果があったことを確認した。

(1) 再利用率

平均再利用率は毎年上昇した。1年目は3%であったが、2年目には12%に跳ね上がり、4年目には17%に達した。しかし、20%前後で飽和する傾向を示した(図2)。目標値は2、3、4年目それぞれ10、15、20%であった。したがって、2、3年目には目標を達成したが、4年目には達成できなかった。

(2) 再利用部品の平均サイズ

再利用された部品の平均サイズは毎年上昇し、4年目には約600行に達した(図3)。大きい部品を利用することは以下に述べるように効率的な再利用にとって好ましいので、これは再利用運動のひとつの成果と言える。

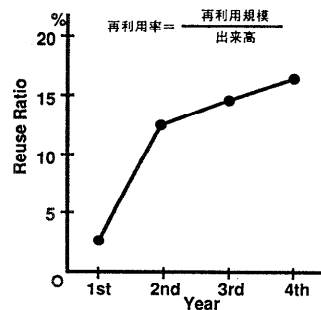


図2 平均再利用率
Fig. 2 Reuse ratio.

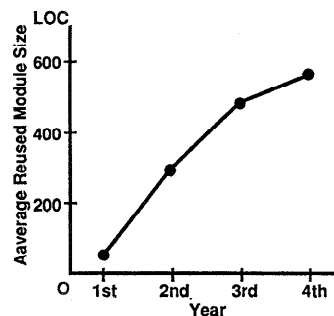


図3 再利用部品の平均サイズ
Fig. 3 Reused module size.

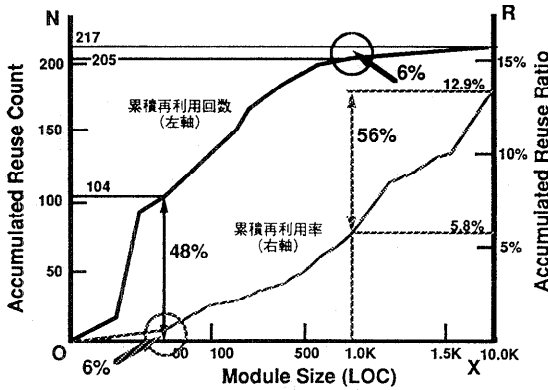


図 4 累積の再利用回数/再利用率
Fig. 4 Accumulated reuse count and ratio.

再利用運動 2 年目の全再利用ケースを部品サイズの昇順にソートし、累積部品利用回数および累積再利用率を部品サイズに対してプロットした (図 4)。この 2 つのグラフから次の事実が読み取れる。

50 行以下の小規模部品は利用回数では約半数 (48%) を占めているが、再利用率ではほんの 6% に過ぎない。一方、1 キロ行以上の大規模部品は利用回数では 6% に過ぎないが、再利用率では半分以上 (56%) を占めている。これより、大きい部品を利用することが高い再利用率を達成するために効率的な方法であることがわかる。

(3) 再利用部品の出所

一般にある開発プロジェクトが再利用した部品の出所は次の 3 種類に分類できる。

- ①他の開発プロジェクトが登録した部品、
- ②自プロジェクトが登録した部品、
- ③ライブラリ外部から手に入れた部品。

この中で①と②がライブラリからの再利用である。

他人の成果を効率的に活用するという意味で②よりも①の方が推奨される。また、①+②の割合が大きいことはライブラリが役に立っていることを表す。

再利用運動が進展するにつれ、各ソフトウェア開発プロジェクトは次第に他の開発プロジェクトが登録した部品を利用するようになった。2 年目にはその率は約 10% であったが、4 年目には 70% に達した。また、①と②を合わせたライブラリからの利用は 2 年目には約 30% だったが、4 年目には 100% に達した。すなわち、すべての再利用された部品がライブラリからまかなわれた。このように 4 年間の運動で、再利用された部品の出所は改善された (図 5)。

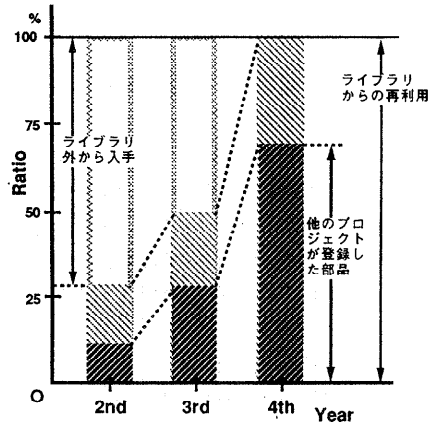


図 5 再利用部品の出所
Fig. 5 Origin of reused modules.

(4) 部品ライブラリの利用度

部品ライブラリの利用度を議論するために 2 つの指標を定義する。

①平均再利用頻度：ライブラリ内のすべての部品が 1 年間に再利用された回数の合計値をライブラリ内の部品数で割った値。

②活性部品率：1 年間に 1 回でも利用された部品の個数をライブラリ内の部品数で割った値。

平均再利用頻度は 2 年目には 0.18 であったが、毎年上昇し、4 年目には 0.28 になった (図 6)。すなわち、部品ライブラリの利用度は約 50% 改善された。

活性部品率は 2 年目から 4 年目までほとんど変わらずほぼ 0.2 という値で一定していた。このように低い値は、ライブラリの少なからぬ部品がいわばゴミであり、何らかの選別が必要なることを示している。再利用運動を開始するときこのような事態を予想し、「3 年間 1 回も使われなかった部品はライブラリから削除す

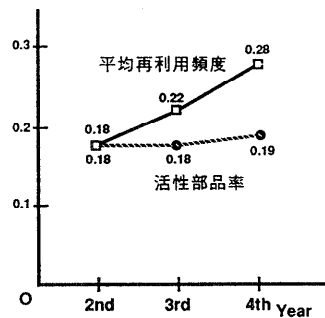


図 6 部品ライブラリの利用度
Fig. 6 Usage of module library.

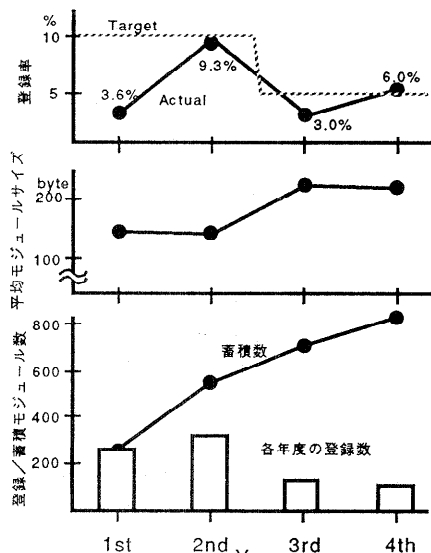


図 7 再利用モジュールの登録状況
Fig. 7 Deposition of reusable modules.

る」という規則（3年削除規則）を定めていたが、その必要性が実証された。

(5) 部品ライブラリの大きさ

部品ライブラリは順調に成長し4年目には800部品に達した。しかしこの過程で1つの教訓を学んだ。2年目には目標登録率として10%を設定した。各開発プロジェクトはこの目標を達成しようと努力し、目標に近い9.3%という登録率を達成した。しかし、登録された部品の品質は期待したほど良くなかった。このため3年目以降は登録目標値を5%に引き下げた。この結果開発グループに対する負担が軽減され、より品質の良い部品が登録されるようになった。3年目および4年目の実際の登録率はそれぞれ3%と6%であった(図7)。

3. 議 論

前項で述べた再利用運動での経験に基づき主としてソフトウェア再利用の管理的側面について議論する。

3.1 ソフトウェア再利用の阻害要因

部品ライブラリ再利用運動の過程でさまざまな問題点に遭遇した。その中から主要なものを10件あげる。

- ①十分な部品ライブラリがない。
- ②部品を作っても自分は得をしない。
- ③開発プロジェクトが協力しない。
- ④外注業者は得をしない。
- ⑤保守・改造要望に対する即応体制がない。

- ⑥信頼性が低い。
- ⑦ドキュメントが未整備。
- ⑧開発プロセスの標準化が不十分。
- ⑨性能が低下する。
- ⑩効果が不明。

問題点①は部品ライブラリの構築には多大の初期投資が必要だが、その効果ははっきりしていない状況で実施するのはきわめて困難なことによる。

問題点②と④の理由は次の3点に要約できる。

- 忙し過ぎ再利用を考えている余裕がない。
- 再利用の大切さに対する理解が不足している。
- 心理学的な拒絶（他人の作ったプログラムなど使いたくない）。

このような理由に起因する問題点②③を解決するには組織の意識改革を地道に進めるしかない。プロジェクト管理者および技術者に対して組織的に再利用の教育をすること、また、これが可能となるようにプロジェクト管理者に十分なリソースを与えることが必要である。

問題点④は同じく管理的課題である。これは、外注先への支払いが行数ベースあるいは工数ベースであるため、生産性を上げると収入が減ることによる。この問題を解決するには、再利用による生産性向上を考慮するように契約形態を変更するしかない。

問題点⑤も管理的課題である。部品開発には専任チームを当てるべきである。

問題点⑥から⑧は一見すると技術的課題のように見える。しかし、これらはプロジェクト管理者に十分なリソースを与えれば解決する問題であり、むしろ管理的課題ととらえるべきである。

問題点⑨と⑩のみが純粋に技術的な課題である。

したがって、再利用推進上の問題点の多くは管理面の理由によるわけで、再利用のためには技術面よりも管理面からの取り組みが重要なことがわかる。これは、ソフトウェア再利用が次の本質的な特徴をもつことに起因している。

①ソフトウェア再利用が具体的な成果を出すまでには約3年ないし4年という長期間の継続的な努力が必要⁸⁾。

②100人を越える規模の組織的な活動である。

③個人個人が自分のためではなく組織のために行動するという協調的な姿勢が必要。

このような性質をもつソフトウェア再利用が成功するためには、上級管理者の理解と支援がきわめて重要

である。

3.2 インセンティブ制度

ソフトウェア再利用運動を成功させるためには、組織内に再利用運動を奨励する雰囲気を作ることが必要であり、このためには報奨制度が有効である。米 Hartford Insurance 社では、再利用可能情報の提供を奨励するインセンティブ制度を導入し、効果を上げている。同様な試みを米 AT & T ベル研でも実施している。

本再利用プロジェクトでは次の3種類の賞を設け、優れた成績をあげた開発チームあるいは個人に与えた。

①登録賞：高い登録率を達成したプログラムの開発チームに与えられる。

②利用賞：高い再利用率を達成したプログラムの開発チームに与えられる。

③提供賞：利用回数×部品規模が大きい部品を開発した個人に与えられる。

賞の授与は頻度が高いほど効果がある。本再利用プロジェクトでは年に1回表彰したが、頻度が少なかったと反省している。小規模な表彰を毎月行い、年度末に年間での最優秀賞を与える形態が好ましい。

再利用運動の初期に、どんな賞を与えるべきかを議論した。多額の金品を与えることは特にモジュールの登録に効果があるが、意図的に冗長なコーディングをしたり、本来の担務をないがしろにして部品の開発に没頭するようなモラルの低下を招く危険性があることが指摘された。そこで、賞としては表彰状と少額の現金を与えることとした。つまり、賞には経済的な意味はもたせず、社会的な意味を与えた。

3.3 ドメイン分析

(1) 再利用運動での反省

運動の4年目で達成した再利用率は約17%であり、運動を継続した場合には20%程度に飽和する傾向を示していた。このような低い再利用率しか実現できなかった理由は、対象ドメイン(分野)が良くなかったこと、部品の収集方法が未成熟だったことによる。

(2) 対象ドメインの選択

多くのソフトウェア再利用の成功例での共通点は、対象分野が、

- ①適度に狭い、
- ②十分に理解されている、
- ③技術が静的で短期間で変化しない、
- ④多量の類似プログラムを開発する、

という性質を満足していることである⁹⁾。①②③によりドメイン分析および部品開発が比較的容易に実施でき、また④により再利用の初期コストを回収できる。

本再利用プロジェクトでの開発対象はソフトウェア開発支援ツールであり、条件①は満たしていた。しかし、多くのツールが新規機能を実現することが目標であったためかならずしも条件②③を満たしていなかった。また、個々のツールを1品ずつ開発するという状況だったため、条件④を満たしていなかった。

ソフトウェア再利用が成功するかどうかはドメインの性質とソフトウェア開発組織の人数に強く依存する(図8)。一般に良いドメインであれば、人数が少ないときは成功しやすいが、人数が多くなるにつれ成功しにくくなる。ソフトウェア再利用における管理の役割は図8の成功領域を押し広げることである。

(3) 再利用モジュールの収集

再利用部品の収集方法は場当りの(Ad hoc)手法、協調設計、ドメイン分析の3とおりがある。場当りの手法とは、個々の開発グループがそれぞれまちまちの規準に従い、再利用できそうなモジュールを抽出し、必要があれば手直しを施す方法である。したがって、抽出されたモジュールは他の開発グループにより再利用されるという保証がない。

協調設計とは、ほぼ同時期に開発を進める2つあるいはそれ以上の開発グループが協調して設計することにより、共通に使えるような機能を抽出し、これを1つのグループが開発し他のグループが再利用する方法である。本再利用プロジェクトでは協調設計を適用したケースが2例あるが、ライブラリへの貢献度は低かった。

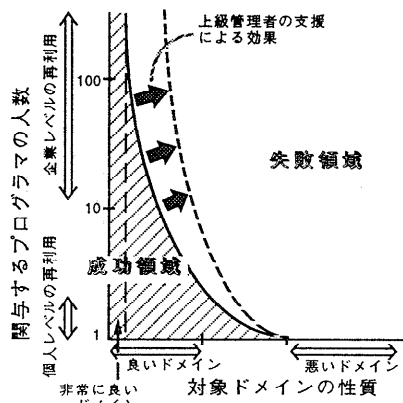


図8 再利用の成功・失敗領域

Fig. 8 Reuse success and failure regions.

(4) ドメイン分析

ドメイン分析は以上の2つの方法より優れた再利用モジュール収集法である。ドメイン分析とは、特定のアプリケーション分野についてその業務あるいは既存ソフトウェアを分析することにより、その分野のプログラムの構造、機能などを明らかにし、ドメインモデルを作ることである¹⁰⁾。分野の共通機能を実現した部品群を分析することもドメイン分析の一種である。通常、ドメイン分析には専担グループを当てる。

ドメインモデルの表現方法として以下のものが提案されている。

- ①ドメインの語彙、シソーラス、
- ②意味ネットや ER 図で表現したドメインの概念の関連、
- ③部品機能の多面的分類、
- ④部品機能のオブジェクトとオペレーションの組み合わせによる表現。

ドメイン分析は、対象分野の共通機能を過不足なく実現した標準的な部品セットを開発するための指針を与えてくれる。したがって、ドメイン分析は再利用性の高い部品を作成するための基本的ステップである。

ドメイン分析の問題点は多大の工数を要することである。このため、大量のプログラムを開発する分野でなければ元が取れない。

本再利用プロジェクトでは、文字列、スタック、キュー、2進木、画面などの基本的な機能に関するドメイン分析を実施し、約110個の関数を開発した。ドメインモデルとしてはオブジェクトとオペレーションの組み合わせによる表現を採用した。しかし、関数の機能が小さいため平均再利用率にはごくわずかしかなかった。

3.4 部品検索ツール

多くの人が再利用支援ツールの持つべき機能の筆頭として検索機能あげる。しかし、これは部品再利用の実態を良く理解していないことから来る誤った要求である。

本再利用プロジェクトでは部品ライブラリ再利用運動の一環として統合化 CASE システム SoftDA¹¹⁾の再利用支援サブシステムを開発した。これは部品をUNIXのファイル・ディレクトリに階層的に管理し、ユーザが指定したキーワードで部品を検索する機能を提供する。シソーラス機構を用いて同意語および類義語を適切に処理することにより、検索効率を向上する機能を備えている。しかし、この再利用支援機能を提

供してまもなく、高度な検索機能は不要ことがわかった。利用者は、ライブラリ登録部品を印刷した部品ハンドブックのページを繰る方が、検索機能を用いてライブラリを検索するよりも効率が良かったため、もっぱら部品ハンドブックを利用した。

大規模なライブラリの中から部品を検索し、その機能を理解した上で利用するような「ナイーブな」再利用形態は実際には成立しがたい。これは、決められた工期の中で、それまで見ず知らずの部品を理解し利用可否を判断するのは困難なことによる。特に、部品の規模が大きい場合にはほとんど不可能である。

したがって、実際の再利用では標準化された部品を用意しておき、これを熟知した技術者が利用する。特定の機能を実現するときに関係する部品数はせいぜい数10個と極めて少数であり、検索するまでもない(ライブラリに登録されている部品数が多かったとしても、特定の機能に関係する部品はライブラリ登録部品のごく一部である)。

したがって検索機能の用途は、初心者が再利用ライブラリを学習する場合と、きわめて特殊な機能を実現するような場合に限定される。

4. おわりに

ソフトウェア再利用の成否を強く支配するのは技術的要因ではなく管理的要因である。ソフトウェア再利用を成功させるためには、上級管理者のコミットメントのもとに、再利用推進組織に十分なリソースと権威を与え、再利用を奨励する雰囲気を作り上げることが肝要である。また、再利用運動を開始する前に対象が良いドメインかどうかを判断しなければならない。悪いドメインを対象として努力しても無駄である。良いドメインを対象として選択し、ドメイン分析に基づき

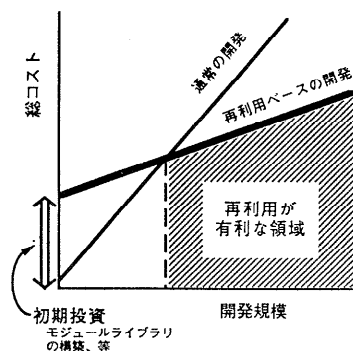


図9 ソフトウェア再利用コストモデル
Fig. 9 Software reuse cost model.

再利用性の高い部品を作成すべきである。このようにして作成した部品の再利用を地道に進めることにより組織的なソフトウェア再利用を成功させることができる。

ソフトウェア再利用には当然限界がある。過度の期待を抱いてはならない。「ソフトウェア再利用が期待したほどには成果をあげていない」という批判は、再利用の成否が管理的課題と対象ドメインに強く依存することを認識していないことからくる。

本稿ではコード部品ライブラリの再利用を対象として議論した。しかし、導かれた結論は再利用対象が設計、要求仕様、テストケースなどであっても成立する。それはいずれの再利用であっても部品の開発にかかる大きな初期投資を以降の再利用で回収するという原理に基づいているためである (図9)。

今後の課題としてソフトウェア再利用のコストモデルの構築がある。これにより、対象ドメインがどのような条件を満たしているとき再利用を実施すべきか、またどの程度の効果が期待できるかなどに関する指針を得られる。そのためには、再利用の実践の中でデータを集める努力を継続することが必要である。

参 考 文 献

- 1) *IEEE Trans. Softw. Eng.*, Special Issue on Software Reusability, Vol. SE-10, No. 5 (1984).
- 2) *The First International Workshop on Software Reusability*, Dortmund, Germany (1991).
- 3) Maiden, N. A. and Sutcliffe, A. G.: Exploiting Reusable Specifications through Analogy, *Comm. ACM*, Vol. 35, No. 4, pp. 55-64 (1992).
- 4) 山口高平, 落水浩一郎: 事例ベース推論とモデル推論の相互作用に基づくソフトウェアプロセスモデル獲得支援環境, 知能ソフトウェア工学の動向と展望シンポジウム, pp. 75-81 (1992).
- 5) Lanergan, R. G. and Grasso, C. A.: Software Engineering with Reusable Design and Code, *IEEE Trans. Softw. Eng.*, Vol. SE-10, No. 5, pp. 498-501 (1984).
- 6) Matsumoto, M. et al.: Specifications Reuse Process Modeling and Case Study-Based Evaluations, *COMPSAC 91*, pp. 499-506 (1991).
- 7) Isoda, S.: Experience Report on Software Reuse Project: Its Structure, Activities, and Statistical Results, *14th ICSE*, pp. 320-326 (1992).
- 8) Prieto-Diaz, R.: Making Software Reuse Work: An Implementation Model, *ACM SIGSOFT Software Engineering Notes*, Vol. 16, No. 3, pp. 61-68 (1991).
- 9) Frakes, W. and Prieto-Diaz, R.: Tutorial on Software Reuse and Domain Analysis, *CASE 92* (1992).
- 10) Prieto-Diaz, R.: Domain Analysis for Reusability, *COMPSAC 87*, pp. 23-29 (1987).
- 11) Isoda, S. et al.: Integrated CASE System: SoftDA, *NTT Review*, Vol. 2, No. 2, pp. 52-61 (1990).

(平成4年10月7日受付)

(平成5年1月18日採録)



磯田 定宏 (正会員)

1946年生。1969年東京大学理学部物理学科卒, 1971年同大学理学大学院物理学専攻修士課程修了。1971年日本電信電話公社(現NTT)に入社。管理システム研究室長, ソフトウェア基礎技術研究部長などを経て, 現在, NTTソフトウェア研究所主席研究員。1978~79年米国イリノイ大学計算機科学科客員研究員。ソフトウェア工学全般, 特にCASE, 設計法, 再利用, メトリックス, プロジェクト管理, ヒューマンファクタ等に興味を持っている。1993年より情報処理学会ソフトウェア工学研究会主査。COMPSAC '91, ICSE '93, ICCI '93, ICRE '94等のプログラム委員。電子情報通信学会, ソフトウェア科学会各会員, IEEE シニアメンバー。