新技術導入をトリガーとした
面接型教育方法の変更促進

海谷 治彦　　関本 理佳　　海尻 賢二

信州大学 工学部 情報工学科
〒 380-8553 長野市 若里 4-17-1

電話: 026-269-5469　　ファクシミリ: 026-269-5495
{kaiya,rika,kaijiri}@cs.shinshu-u.ac.jp

あらまし　既に遂行されている作業を支援するシステムの要求仕様は現状の業務に大きく依存している．その理由の一つは，現状で利用しているツールや技術に現状業務が制約されているからである．よって，本質的に利用する必然性のないツールや技術を根拠とする要求は，変更可能な部分とみなすことができる．このような判断を行うためには，利用されるツールや技術と要求仕様の部分の関係を明らかにする必要がある．本稿では，新技術の導入が既存業務に与える影響を推論することにより，現状業務の変更や改善を促進する手法を提案する．本方法では，現在利用されている技術と要求仕様の関係を明らかにし，交換可能な技術を模索することによって要求仕様の変更を行う．よって，本質的に変更不能な背景を持つ要求は変更されることがない．この手法を本学におけるプログラム作成の個別指導に適用し，指導方法，被指導者の学習方法の変更案を示す．

キーワード　要求工学，形式仕様

Changing Requirements by Introducing New Technologies: Case Study of a System for Supporting a Programming Coach

Haruhiko Kaiya　　Rika Sekimoto　　Kenji Kaijiri

Department of Information Engineering, Faculty of Engineering
Shinshu University

4-17-1 Wakasato, Nagano City, 380-8553, JAPAN
Phone: +81-26-269-5469　　Fax: +81-26-269-5495
Email: kaiya@acm.org　　{rika,kaijiri}@cs.shinshu-u.ac.jp
http://www.cs.shinshu-u.ac.jp/~kaiya/

Abstract　*When you define requirements for a system to support a task which is already carried out, the requirements would be heavily restricted to the current way of the task, and the way depends on the tools or technologies currently used. So you should carefully sort out what can be changed and what can not. In this paper, we presents a method to define the requirements for a system supporting such task, so as to facilitate the use of new technologies. In this method, first we deduce requirements of the task from the properties of technologies currently used, and second we change the requirements by comparing the properties of technologies currently used with the properties of new ones. We also presents a small but realistic example for applying this method.*

Key words　*Requirements Engineering, Formal Method*

1 はじめに

特定の業務を支援するシステムの要求仕様を定義する場合，要求分析者は，その業務に従事する作業者へのインタビューを持つことがある．また，これから開発されるであろうシステム無しで既にその業務が遂行されている場合，現状の作業方法や作業環境の観察することが可能かもしれない．しかし，インタビューや観察のみからでは，現状業務のどの部分が変更可能であり，どの部分が変更不能であるかを合理的に判断することは困難である．

一方，作業者は，新しいツールや技術の導入によって，現状の業務遂行方法とは異なる作業方法を強いられる場合がある．たとえば，ある会社で UNIX ベースの計算機システムがマイクロソフト Windows ベースの計算機システムに全て置き換えられた場合，いままでの業務の方法を大きく変更しなければならないだろう．そのような望ましくない状況下でも，システム開発者は，作業者の困難をとりのぞき，業務を改善するために，新システムの利点を見出し，提案していかなければならない．

どちらの場合でも，システムが導入される作業に対してどのような変更が可能で，どのような変更が不可能かを，システム開発者は予測しなければならない．予測材料の 1 つとして，現在の作業方法を支える技術やツールと，新システムと共に導入される技術やツールの関係を明らかにすることは有効だと思われる．

本稿では，著者らが提案した要求変更の枠組み [2] をもとに，システムの要求変更法を提案する．提案した枠組みでは，システム要求を UML のアクティビティ図 [1] によって表現する．アクティビティは作業を構成する機能とみなすことができ，個々のアクティビティは事前・事後条件で仕様化できる．そして，全てのアクティビティの事前・事後条件を充足するようにアクティビティ図のトポロジを構成する．よって，あるアクティビティが別のアクティビティに変更された場合，それにあわせて図全体のトポロジも変更しなければならない場合がある．要求仕様の部分的変更と要求仕様全体の変更の関係を我々のフレームワークではこのようにモデル化している．

しかし，我々のフレームワーク [2] は，要求変更をモデル化しているのみであり，どのアクティビティを変更すべきかの決定や，アクティビティの変更とアクティビティ内で利用される技術やツールの変更との関係をどのように認識するかについては何も言及していない．本稿では，そのような決定や認識の方法を提案している．また，本稿では小規模ではあるが現実的な要求変更の事例を紹介する．

続く 2 節では，前述の方法の詳細を紹介する．3 節では，本方法の妥当性を示すための事例を紹介する．この事例は，著者らの大学におけるプログラムおよびレポート記述の個別指導 (「レポート面接」と呼称されている) を支援するシステムの要求変更および業務変更を提案している．3.1 節でレポート面接業務の詳細を紹介し，3.2 節で同業務に適用可能な業務を検討する．そして，3 節の残りで，現状の業務に新しい技術を導入した場合の要求変更の事例を紹介する．最後にまとめと今後の課題を述べる．

2 要求仕様および業務変更の促進法

2.1 要件と技術の関係

我々の枠組みは機能部品の比較を行う述語である *Plug-in match*[5] をもとにしている．

Plug-in match は以下のように定義される．

$$S \sqsupset R = (R_{pre} \Rightarrow S_{pre}) \wedge (S_{post} \Rightarrow R_{post})$$

ここで，R は検索関数 (要求仕様) であり，S は部品ライブラリ内で R に適合する関数を示す．原文献 [5] では，$\sqsupset$ の記号は用いられていないが，本稿ではこの記号を用いることとする．

同文献では *Guarded plug-in match* という述語も導入されており，それは Z 記法 [4] における詳細化の概念とほぼ一致する．しかし，我々の枠組みでは，Guarded plug-in match に比べ条件の緩やかな Plug-in match を利用する．

我々の枠組みおよび方法では，R を現状業務における活動とみなし，S を利用されている技術や道具とみなす．よって，$S \sqsupset R$ を「ある作業における活動 R は，技術もしくは道具 S に基づいているか，もしくは制限を受けている 」と解釈する．

本稿では関数仕様を Z 記法を用いて表現することにする．以下に $\sqsupset$ に関する簡単な例を示す．まず，関数 *Push* が以下のように仕様化されるとする．

$$Push_{pre} \mathrel{\widehat{=}} \text{true}$$
$$Push_{post} \mathrel{\widehat{=}} [s, s' : \operatorname{seq} X, a : X \mid s' = s \frown \langle a \rangle]$$

これを S，すなわち利用可能な機能ライブラリとする．また，*LimAdd* という関数は以下のように定義されているとし，これは R，すなわち要求仕様とする．

$$LimAdd_{pre} \mathrel{\widehat{=}} [s : \operatorname{seq} X \mid \#X < 50]$$
$$LimAdd_{post} \mathrel{\widehat{=}} [s, s' : \operatorname{seq} X, a : X \mid \#s' = \#s + 1]$$

この場合，$Push \sqsupset LimAdd$ より，*Push* は *LimAdd* の要件を満たすことを示すことができる．我々の枠組みでは $Push \sqsupset LimAdd$ を「作業者は *Push* という道具を有しているので，作業中に *LimAdd* という活動を遂行する 」と解釈する．一方，作業者が *Push* ではなく，以下のような別の道具 *Merge* を有していることを想定すると，

$$Merge_{pre} \mathrel{\widehat{=}} \text{true}$$
$$Merge_{post} \mathrel{\widehat{=}} [s, s' : \operatorname{seq} X, a : X \mid \{a\} \cup \operatorname{ran} s = \operatorname{ran} s']$$

LimAdd で示される活動を遂行することができない．これは，$Merge \sqsupset LimAdd$ が常に充足可能なわけではないという形で表現できる．

2.2 要求仕様内の関係

前述のように我々の方法では，仕様化対象の作業内の個々の活動を事前・事後条件で仕様化する．そして，作業全体の構造を UML によって定義されるアクティビティ図 [1] によって表現することにする．アクティビティ図は作業内におけるワークフローの一インスタンスを表現する図式である．我々はアクティビティ図の集合をその作業に対する要求仕様とみなす．

図中のアクティビティ間の関係に矛盾が発生しないために，本稿では以下に示す *CONS* というルールを導入する．

> *CONS*: それぞれのアクティビティの事前条件は，図中で先行するアクティビティ群の事後条件から推論できなければならない．

アクティビティ間の関係は，契約による設計 (DBC) [3] の原理に従うことが理想である，すなわち，アクティビティの事前条件は先行するアクティビティ群の事後条件の連言と等価であることが理想であるが，本稿では，DBC の原理より条件を緩めることにする．

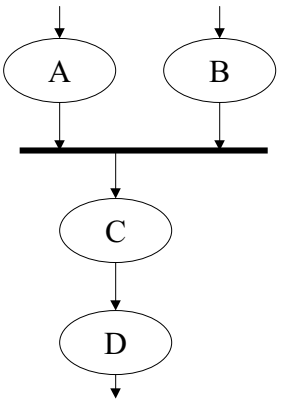

図 1: *CONS* 条件の例

図 1 に，*CONS* 条件の簡単な例を示す．アクティビティ *A* と *B* は *C* に先行しているため，我々は以下の推論が可能である必要がある．

$$A_{post}, B_{post} \vdash C_{pre}.$$

また，*C* は *D* に先行しているため，

$$C_{post} \vdash D_{pre}$$

も同様に推論可能である必要がある．

2.3 技術変更に伴う要求変更

図 2 は，本稿における要求仕様の全体構造を示している．この仕様は，要求仕様が利用されるであろうツールや技術と独立して存在するのではなく，それらと関連しあって存在していることを明示的に示している．要求仕様と技術の関係は図 3 に示すように常に一対一というわけではない．

これらの枠組みをもとに，技術やツールの変更による要求仕様の変更を行う手順を以下に示す．

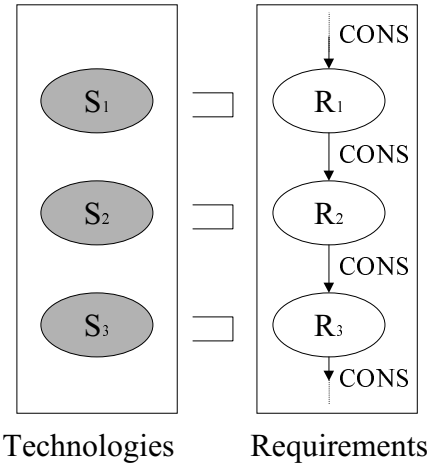

図 2: 技術と対応を持つ要求仕様

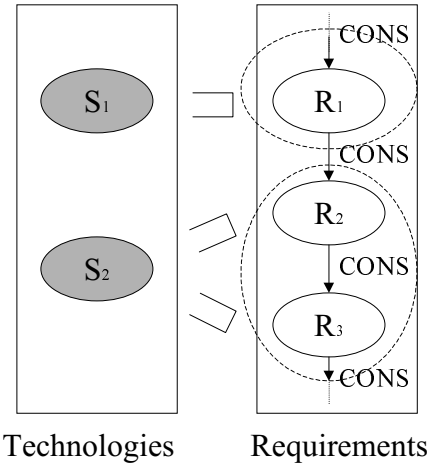

図 3: 要求仕様と技術の関係が一対一でない場合

1. 作業者へのインタビューや作業の観察をもとに，現状作業のアクティビティ図を作成する．
2. それぞれのアクティビティの事前・事後条件を記述する．尚，図間の関係は上記の *CONS* 条件を満たさなければならない．
3. 同時に，個々のアクティビティで利用される技術を認識する．個々の技術の仕様は完全には記述できないため，必要な部分を段階的に仕様化してよい．尚，技術仕様と要求仕様 (アクティビティ図) の要素間には，⊐ 関係が成立しなければならない．
4. ある技術が他の技術に置き換えられることを想定し，技術仕様を書き換える．それに伴い，⊐ 関係が維持されるように，要求仕様 (アクティビティ図の仕様) を変更する．
5. 前段階で要求仕様 (アクティビティ図の仕様) が変更された場合，*CONS* 条件を維持されるようにアクティビティ図のトポロジを変更する．

上記の手順を繰り返すことで，要求変更を確定してゆく．

3 適用事例

本節では「レポート面接」を支援する計算機システムの提案に関する適用事例を紹介する．本稿は本研究の単なる適用事例ではなく，実際の支援システム開発における分析段階に相当する．このシステム開発は，科学研究費 奨励研究 (A)#12780210 によって支援されている．

3.1 業務分析: プログラミングと技術文書作成の指導

本節で，我々が要件を収集し，その要件の変更を提案する対象である作業を非形式的に紹介する．著者らの所属する学科における C 言語の演習コースでは，プログラミング技術と技術文書作成の指導のために，全ての受講者に対して対面型の面接指導を行う．これを本学では「レポート面接」と呼んでいる．図 4 に，この指導業務の概要を示す．現状では，レポート面接自体は計算機システム等の支援なしに行われている．尚，便宜のため，図 4 中のアクティビティに A, B, C ... 等のラベルをつけており，それぞれのアクティビティの概要は以下の通りである．

A: 学生はそれぞれに課題のプログラムとそれについてのレポートを作成する．勿論，学生はプログラミングにはコンパイラやテキストエディタが利用可能であるし，レポート作成には通常，既存のワードプロセッサソフトが利用可能である．また，レポート自体は，ハードコピーとして提出することが規定されている．

B: 指導担当となった教官がレポートをチェックする．プログラムの妥当性を独自に確認するために，指導学生に対してプログラムのソースコードを提出 (電子メール等で) を要請する教官もいるが，大抵は印刷されたレポートのみでチェックを行う場合が多い．また，チェック内容は印刷されたレポートに直接記載する．

C: 教官は個々の学生と個別面接を小会議室で行う．通常，口頭で指導内容を指示する．レポート上のチェックは，コメントすべき個所と内容を見つけるために利用される．最後に，教官はレポートを学生に返却する．

面接時間は一人平均およそ 15 分と短いが，学生によっては指導するポイントが少なくない場合があるため，学生が指導された点をすべて把握するのは困難な場合がある．一教官は通常，15 名程度の学生への面接を行い，面接的は時間的に連続して順番に行われる場合が多い．

D と E: 個々の学生は返却されたレポートを見直し，自分達の技能向上のためにコメント点等を復習する．一部の学生はレポートの再提出および再面接を行う場合がある．

3.2 利用可能な技術の検討

前述のように，レポート面接での指導業務では，計算機システムによる直接的な支援はほとんど行われていない．このような業務に計算機システム支援を導入することが必ず効果的であるとは限らないが，その適用可能性を議論することは有益である．

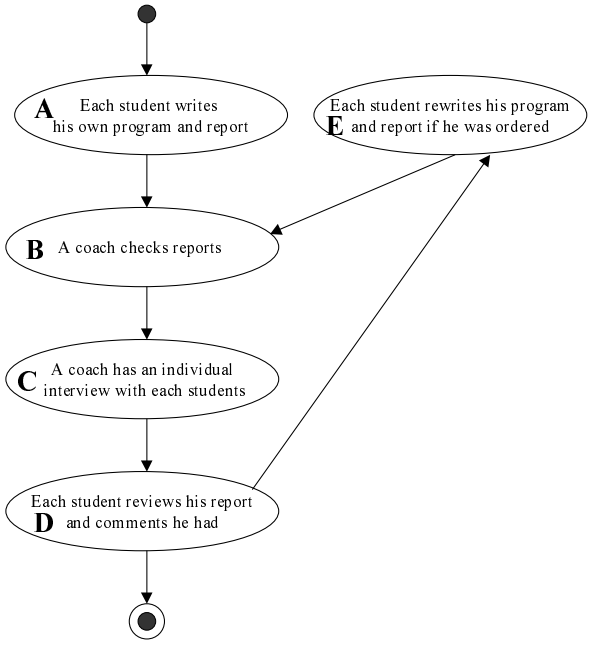

図 4: 「レポート面接」の概要

一方，我々の学科でも授業や演習において計算機を利用することがかなり容易となってきた．例えば，本学科では 1998 年以降，全ての新学部生に対してノート型パソコンの購入を義務付けるようになった．しかし，その利用は一部の授業や演習での直接的な利用 (例えばプログラミングなど) に限られており，教育ツールとして十分に利用されているとはいえない．加えて，現在では一般的なパソコン上において，マルティメデァ情報を用意に扱うことができるようになっている．

そこで，現状で利用されている道具 (適用および置換可能な技術と対応付けるため) と，適用可能性のある技術の検討を行う．一部の技術やツールは他の技術等に依存しているが，本稿では技術間の相互依存関係については議論しない．尚，現在利用されいる道具および技術は 下線 が引かれている．

人間の記憶 ：現状の業務では，教官も学生も自分自身の記憶に大きく頼っている．例えば，教官はアクティビティ*A* において学生へのコメントを準備するが，コメントの全てをレポート用紙等に書き込むのではなく，簡単なキーワードをメモしたり問題部分に下線をひいたりするのみである．学生も面接時のコメントを完全には記録することができないため，場合によってはコメントされた内容を忘れてしまう場合もある．

印刷物 ： 現状の業務では，レポートを印刷物で提出することになっている．印刷物は自由に注釈をつける観点からは有効なツールであるが，その内容を同時並行的に共有する機能を有していない．ま

た，検索機能や，記載されているプログラムの実行も行うことができない．

共有ファイルシステム：電子的ファイルによるレポート提出を許せば，それを共有ファイルシステムに蓄積することで，教官も学生もその内容を同時に参照することが可能である．そのため，例えば，レポート面接前に学生は自分の指摘されるであろう部分の復習をすることで，面接の効率・効果を向上することが可能である．

マークアップ可能ファイル：HTML 等の電子様式は他の情報資源との関連付けを行うことが可能であるため，レポートに対するコメントに充実した解説を添付したり，コメント自身を再利用したりすることが可能となる．

卓上型パソコン：現状の業務では，学生は自宅もしくは計算機室に備え付けられた卓上パソコンでレポートを作成し，その結果を印刷している．

ノート型 ***PC***：前述の通り，本学科でのレポート面接は時間的に連続して順番に行われる場合が多い．また，小規模な欠陥修正で済む場合でも，短いスパン，例えば次の面接者が終わった後に再度など，での再面接はあまり行われない．これは，教官および学生にとって，ある面接を中断し，別の面接を開始し，再度，中断していた面接を再開するような作業は負担が大きいからである．

ノート PC などを利用して個々の面接環境を中断・再現機能を実現すれば，短いスパンの再面接を容易に実現できると思われる．結果として，短時間で解決可能な欠陥や質問をある学生が解決している間，次の学生の面接を行うことができる．

我々は上記のような技術を要求変更のための材料として利用する．我々の方法では，技術そのものではなく，技術によって利用可能となった機能に注目する．上記技術による機能は続く部分で必要に応じて適宜言及する．

続く本節では，2 節の方法に従い，レポート面接業務自身，およびその支援システムの要求仕様の変更を模索する．方法の概要を再要約すると，初めに，現在利用されている技術によって現状業務が如何に制約されているかを認識し，次に，その制約を与えている技術を他に置き換えた場合，業務をどのように変更して良いかを模索する．

3.3 変更例: 効果的な面接準備を促進

図 4 に示すように，学生は自分の面接の順番になるまで自分のレポートがどのようなチェックを受けたかを知る術がない．よって，チェックされた個所を考慮して面接の準備を行うことはできない．もし，このような準備を行うことが可能であれば，面接時間を短縮したり，他の個所について十分な議論を面接時間に行ったりすることか可能になる．

上記の業務にかかわる図 4 内の部分を図 5 により詳細に記述する．現状で図 5 に示すような業務の流れと

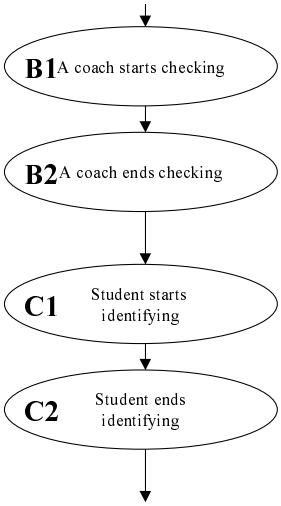

図 5: 現状業務の詳細

なる理由は，レポート自身への参照が排他的であるからである．これは，レポートが印刷物なため，当然である．

この事実は以下のように形式的に記述することができる．

$$B2_{pre} \mathrel{\widehat=} [s : \mathbb{P}\, person \mid \#s = 1]$$
$$B2_{post} \mathrel{\widehat=} [s' : \mathbb{P}\, person \mid \#s' = 0]$$
$$C1_{pre} \mathrel{\widehat=} [s : \mathbb{P}\, person \mid \#s = 0]$$
$$C1_{post} \mathrel{\widehat=} [s' : \mathbb{P}\, person \mid \#s' = 1]$$

について，

$$unref \sqsupset B2$$
$$ref \sqsupset C1$$

のような技術-要求仕様間の関係が成り立つ．ただし，*unref* と *ref* は印刷物のもつ機能であり，以下のように仕様化できる．

$$ref_{pre} \mathrel{\widehat=} [s : \mathbb{P}\, person \mid s = \{\}]$$
$$ref_{post} \mathrel{\widehat=} [s' : \mathbb{P}\, person;\ x? : person \mid s' = \{x\}]$$
$$unref_{pre} \mathrel{\widehat=} [s : \mathbb{P}\, person;\ x? : person \mid \{x?\} = s]$$
$$unref_{post} \mathrel{\widehat=} [s' : \mathbb{P}\, person;\ x? : person \mid s' = \{\}]$$

もし印刷物の代わりに以下で仕様化されるような共有ファイルシステムを利用することが可能であるならば，

$$ref_{pre} \mathrel{\widehat=} [s : \mathbb{P}\, person;\ x? : person \mid x? \notin s]$$
$$ref_{post} \mathrel{\widehat=} [s, s' : \mathbb{P}\, person;\ x? : person \mid s \cup \{x?\} = s']$$
$$unref_{pre} \mathrel{\widehat=} [s : \mathbb{P}\, person;\ x? : person \mid x? \in s]$$
$$unref_{post} \mathrel{\widehat=} [s, s' : \mathbb{P}\, person;\ x? : person \mid s \backslash \{x?\} = s']$$

*B*2 と *C*1 の仕様を以下のように変更しても良いことに

なる．

$$B2_{pre} \mathrel{\widehat=} [s : \mathbb{P}\, person;\ x? : person \mid x? \in s]$$
$$B2_{post} \mathrel{\widehat=} [s' : \mathbb{P}\, person;\ x? : person \mid x? \notin s']$$
$$C1_{pre}[s : \mathbb{P}\, person;\ x? : person \mid x? \notin s]$$
$$C1_{post} \mathrel{\widehat=} [s' : \mathbb{P}\, person;\ x? : person \mid x? \in s'].$$

よって，もし学生と教官が同一人物でなければ (これは自明である)，図 6 に示すようにアクティビティ図のトポロジを変更しても良いことになる．結果として，技術の変更をトリガーとして要求変更を提案することができた．

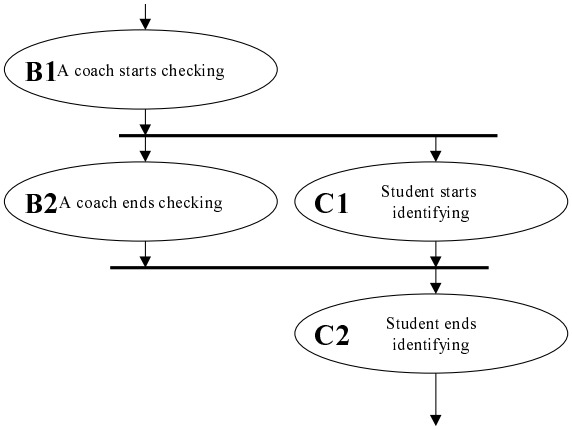

図 6: 変更後の要求仕様

4 おわりに

本稿では，システムの要求変更法を提案し，その例を示した．本方法の特徴は，

- システム要求の部分を，その要求を支援する技術やツールの性質と対にして扱う．
- 要求変更を要求自身ではなく，その背景となる技術やツールの変更をトリガーとして模索することが可能なこと．

これらの特徴から，要求仕様において，どの部分が本質的であり，どの部分がたまたま現在利用している技術等によって規定されているのみであるかも見分けることが可能である．

本稿では，技術やツールを単に関数として仕様化したが，関数の集合，もしくはオブジェクトとしてそれらを仕様化することがより望ましいと思われる．また，本稿では技術間の関係については言及しなかったが，それらも考慮にいれて，要求変更の模索を行うべきである．

謝辞

本研究を進めるにあたり，情報処理学会 ソフトウェア工学研究会 要求工学ワーキンググループでの議論が参考になった．ここに記して謝辞を示す．また，本研究の一部は，文部省科学研究費 奨励 (A) 課題番号 12780210 の援助の下に実施された．

参考文献

[1] M. Flowler and K. Scott. *UML Distilled – Applying the Standard Object Modeling Language*. Addison Wesley, Oct. 1997.

[2] H. Kaiya and K. Kaijiri. Conducting Requirements Evolution by Replacing Components in the Current System. In *Proceedings of APSEC'99*, pages 224–227. IEEE Computer Society Press, Dec. 1999.

[3] B. Meyer. *Object-oriented software construction*. Prentice Hall, second edition, 1997.

[4] B. Potter, J. Sinclair, and D. Till. *An Introduction to Formal Specification and Z*. Prentice Hall International, 1991.

[5] A. M. Zaremski and J. M. Wing. Specification Matching of Software Components. *ACM Trans. Software Eng. and Methodology*, 6(4):333–369, Oct. 1997.