

Spiral-upプロセスにおけるクラス・ライブラリ設計の簡潔化
-ソフトウェアの信頼性に影響を及ぼす人的要素の向上についての実験-

江谷典子 山沢由美
富士ゼロックス情報システム株式会社

ソフトウェアの信頼性を向上させるために、変化に対して頑強なソフトウェアを作ることを目指してオブジェクト指向などの技術的要素やSpiral-up型あるいはプロセス指向アプローチといった人的要素の両者の相互作用が大切である。特に、技術の使い方を調整して問題を解決していく主体となる人的要素の向上は今後ますます重要な課題となるであろう。そこで、ソフトウェア開発実験を行い、人的要因についての原因を追究するのではなくて目標や問題の設定を変化させながら、設計者がspiral-upしていく行動について考察を行った。本稿では、人的要素の考察方法について提案を行うとともに、行動の変化や他者との相互作用およびクラス・ライブラリ設計の変化についての考察結果を示す。

CONCISE CHANGE OF CLASS LIBRARY DESIGN UNDER
THE SPIRAL-UP PROCESS
- AN EXPERIMENT ON ENHANCEMENT OF HUMAN ELEMENTS
WHICH AFFECT SOFTWARE RELIABILITY-

Noriko Etani 1) Yumi Yamasawa2)

1)Yokobori CO-OFFICE 2F,4-6-20, Honmachi,Chuo-ku,Osaka-city,Osaka, 541, JAPAN

e-mail:kerotan@fxis.fujixerox.co.jp, nifty-serve:HHG01256

2)16-6,Nishishinjuku 3-chome, Shinjuku-ku, Tokyo,160, JAPAN

Software reliability is enhanced by the mutual actions between technology element and human element to develop tough software toward a change of environment. Today, object-oriented technologies as technological element and process-oriented approach as human elements are researched. Especially, human elements will be controversial soon because a designer coordinates the technology to solve a problem and the technology is under the control of human beings. Therefore, a software designer's behavior under the spiral-up process was observed through a software development experiment by coordinating a purpose of designer's approach, not by analyzing causes of human error. In this paper, a way of observing human element is proposed. And a process on a change of designer's behavior, interaction between a designer and other group member and class library design will be shown based on the observation of the experiment.

1.はじめに

ソフトウェアの信頼性を向上させるために、変化に対して頑強なソフトウェアを作ることを目指してオブジェクト指向のソフトウェア再利用技術が着目されている。これは、クラスライブラリのクラスをできる限り利用するように設計することで新たに作るクラスを少なくして、効率化やバグの減少を図るものである。このような設計を行う上では、OMT(Object Modeling Technique)などの分析から設計や実装までをカバーする技法やCASEツールといった技術的要素とSpiral-up型あるいはプロセス指向アプローチといった人的要素の両者の相互作用が大切である。特に、技術の使い方を調整して問題を解決していく主体となる人的要素の向上は今後ますます重要な課題となるであろう。しかし、例えばオブジェクト指向に関する現在の学習の多くは、

- ・画一的な教育により答えは一つであるように考えさせてしまうために多様な解決方法などを考える独創性を妨げている
- ・技術の概念や定義から説明が始まり、その意図や背景については言及していないために応用がきかない

といった状況であり、技術を活用しながらその技術の背景や意図を理解していくようなフィードバック学習(図1)が効果的に行われていない。また、人的要素に関する試みはいくつも行われているものの、

- ・個々人の人的要因に着目したもので、人的要因相互の関係はほとんど議論されていない
- ・提案は仮説としてであり、実際のソフトウェア開発データによって検証されたものが少ないことが指摘されている[1]。

よって、技術自体の応用力を高め、対象のよりよい解決につながるアプローチが明確になればソフトウェアの信頼性を向上させるためのプロセス指向を目指した教育やマネジメントの改善を行うことができる。そこで、既にプロセス指向の開

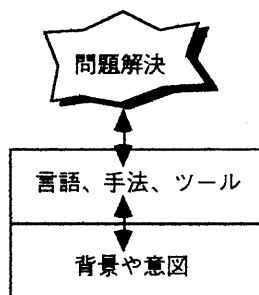


図1.技術の活用と理解

発が行われている環境に新人プログラマを配置してソフトウェア開発実験を行った。本稿では、この実験結果より、

- ・被験者の設計行動の変化
- ・グループメンバ(被験者への教育担当)と被験者の相互作用による影響
- ・クラスライブラリ設計(OMTを利用)の変化について報告を行う。

2.機会把握型の問題

著者らが直面している問題の種類によって実験におけるアプローチが異なる。一般に問題には2つのタイプがある。過去の実績水準と現在の実績とのギャップを排除する乖離型問題と現在の業績とより高い、より望ましい業績水準との間のギャップを縮める機会把握型問題がある。前者は主に原因の追究と問題や目標の再規定を行い、後者は問題や目標の再規定を行うだけである[2]。

今回の人的要素の考察は、spiral-upといった向上していくモデルの追究であり、機会把握型の問題と捉え、人的要素を規定する人的要因の追究は行わなかった。被験者自身で問題に気づき、自分で分析し、対処できるようなアドバイスの工夫を行った。

3.開発実験

3.1.目的

プロセス指向の開発が行われている環境の中で新人プログラマ(被験者)が従来の開発手順(water-fall-down model)との違いに気がつき設計行動を自ら変化させていくプロセスとその成果としてのクラスライブラリの設計の変化を考察する。ここで述べるプロセス指向とは、

- ・結果そのものではなく、プロセスの向上を図ることが主目的である
- ・プロセスを向上させるメタ・プロセスを指向する
- ・結果を生成するプロセスのやり方(少なくとも一つ以上)にフィードバックを行い変更するといったことを目指して行動が行われている環境である[3]。

3.2.方法

(1) 被験者

1名 女性。勤続 2年。

分析設計 経験なし。

実装 C言語 (約3年)。

WindowsおよびVisual C++(MFC)は今回初めて利用。

(2) 相互作用

プロジェクトリーダーは助言者、メンバ1名は開発実験用仕様提供者、メンバ1名は被験者の教育担当といった関わり環境。直接的な対話のほとんどが教育担当者によって行われた。(図2)

(3) 方針

- ・被験者にはオブジェクト指向の背景と目的や意図から説明を行い、必要最低限の基礎知識を学習させる。
- ・次に、実際に具体的な仕様を与え、被験者自身のやり方で行わせる。
- ・障害となるであろう部分に対して仕様変更を加

え、障害を被験者自身で気づかせ、障害を分析し、対処させる。

- ・そこで、もう一度文献を読ませたりアドバイスをを行い、理解を促進させる。

(4) 実験計画

・期間 1994年1月～3月 (3か月)

・工程 設計から実装 (図3)

・仕様 CardEditorの制作 (図4)

情報整理ツール

・開発環境

機器 PC(DELL)

OS 日本語Windows

言語 VisualC++(MFC 2.0)

分析/設計 OMT

(5) 手続き

被験者自らが記述したノートでの発話行動とインタビュー結果の分析から、

- ・被験者の設計行動の変化
- ・グループメンバ (被験者への教育担当) と被験

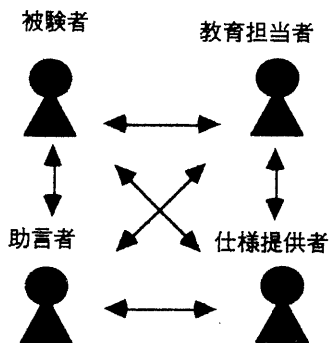


図2.相互作用環境

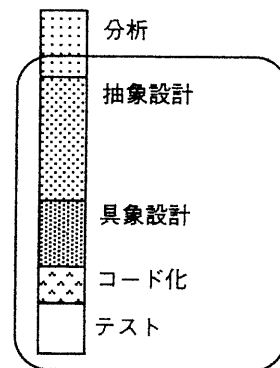


図3.今回のソフトウェア開発実験の工程

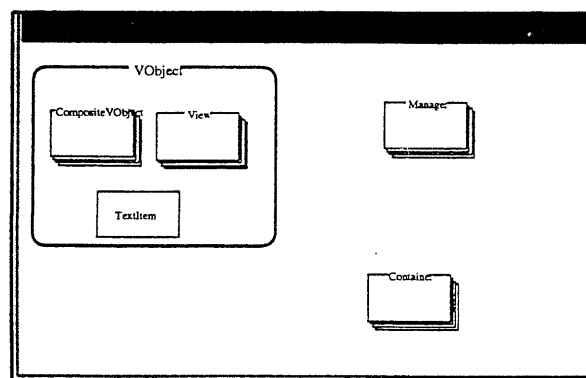


図4. Card Editorの完成イメージ図

- 者の相互作用による影響
- ・クラスライブラリ設計（OMTを利用）の変化の抽出を行う。
ノートには、
- ・クラスライブラリ設計過程
- ・グループメンバ（主に教育担当者）からのアドバイス
- ・被験者自らの設計行動に対する自省が詳細に記述されている。

3.3. 結果

(1) クラスライブラリ設計の簡潔化（図5）

- ・1994年1月10日現在
MVCモデルを手がかりにしている。
- ・1994年3月15日現在
クラスライブラリのクラスをできる限り利用するように設計し、新たに作るクラスが少なくなる。

(2) 設計行動と相互作用

[自省 (reflection) の過程]

被験者自らが自分のアプローチがうまくいっていない事に気がつきその時点までのプロセスを顧みてアプローチを変更させることである。図6では設計活動と自省の関係を示している。

- ・設計活動の時間の経過とともに自省が起きる間隔が短くなっている。

1回目は15日目、2回目は前回から10日目、3回目は前回から5日目に自省が行われている。

- ・自省する設計プロセスの範囲が広がっている。

1回目は前14日間、2回目は前9日間と1回目の自省の範囲、3回目は前4日間と1回目および2回目の自省の範囲となっている。

- ・自省の内容の変化

1回目や2回目では実験者自身の個々のアプローチについて自省を行っている。3回目にはその時点までに行ってきたアプローチ全体を眺めて実験者自身の開発の進め方や考え方について自省を行っている。この3回目の自省後にはクラスライブラリの設計が大きく変化した。

[相互作用]

被験者の設計行動や自省と教育担当者の行動の関係を表1に示した。

- ・3.2.方法（3）方針で述べたように被験者が設計行動を起こす前に説明した背景や意図が被験者が自省する時に必ず想起されている。
- ・被験者自らプロセスにフィードバックを行う際に再度受けたアドバイスや以前受けた説明などを取り込んで、新たな設計行動へと進んでいる。

3.4. 考察

今回の開発実験では、オブジェクト指向の開発環境を初めて体験した新人プログラマがわずか3か月で再利用性の高いクラスライブラリの設計を行うことができた。単にOMTの記述方法やクラスライブラリの仕様を学習しただけではクラスライブラリの設計をOMTに表現することが難しく、何度もアプローチを行い失敗する中で技術の背景や意図の理解を深めたことが、開発実験後期16日間で簡潔なクラスライブラリ設計OMT図を仕上げるに至ったことが分かった。そこで、被験者の設計行動からオブジェクト指向技術などの従来とは考え方の異なる技術を導入する時の課題を次に示す。

(1) なぜ、背景や意図までを理解して技術を活用できないのか

被験者は何度も不安であることを述べており、また不安を回避するために学習成果が明確に分かる方法を取っている。またその方法を向上させることに専念するために詳細なことにひきずられ、現在の目標とは異なる方向に進んでいることに気がつかず、背景や意図を理解できないままに開発が進められると考えられる。今回の実験では、

- ・不安とは

従来のように成果がすぐに出ない不安

従来とは異なるアプローチに対する不安

技術的要素について詳しく知らない不安

- ・不安を解消するための設計行動とは

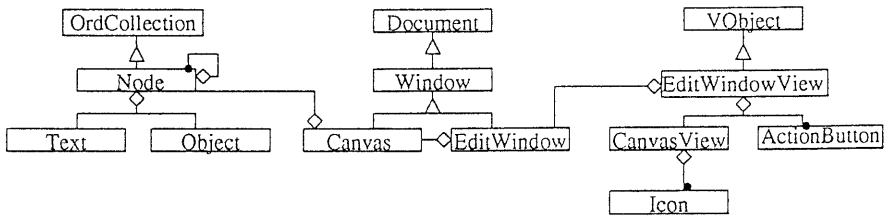
VC++のレファレンスを学習

プログラムソースを読む

見た目（ビュー）のデザインをすぐに行う

といったことが行われた。しかし、前述の方針通り、設計を行う前に背景や意図について説明したことや自省の時に再度説明を行ったために、時間の経過とともに被験者自らプロセスにフィードバックを行い理解を深めていったことが伺える。

・ 1994年1月10日現在



・ 1994年3月15日現在

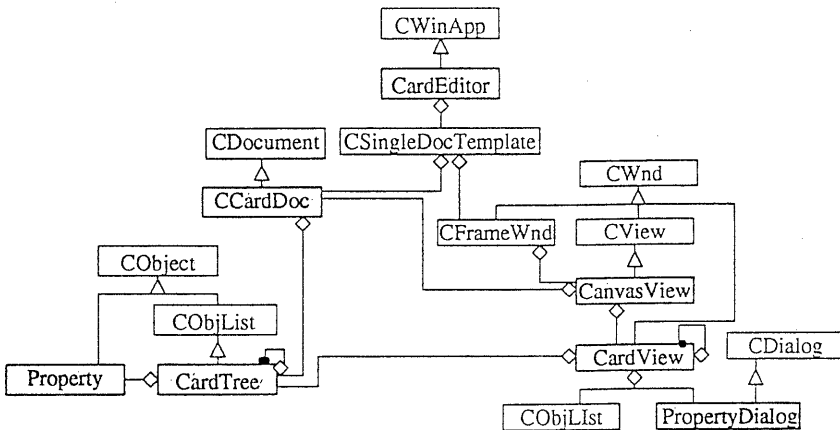


図5.クラスライブラリ設計OMT図の変化

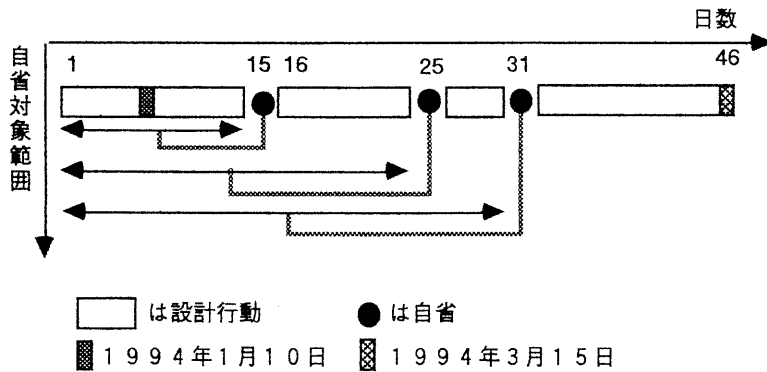


図6.設計活動と自省の関係

経過 日数	ノート記 述年月日	被験者の設計行動	アドバイス	被験者の自省		
				1回目	2回目	3回目
1	1994.1.5	Viewデザイナーの検討	MVCモデルのMが重要、設計のシナリオを提示 本質を見極めることが重要	C++を理解していないの にクラスの宣言や動作 の定義を無理にC++的に 考えようとした		
2	6	MVCモデルを手がかりにOMT図を作成				
3	7	VisualC++講習会参加レビュー	ユースウェアの主旨 設計条件の提示	C++の文法を知らない 不安からレファレンス を学習してしまった		
4	10	VisualC++レファレンスマニュアル を読んで学習				
7	13					
9	17					
13	21					
14	24					
15	25	自省 dead-endは認識...	プロトタイプに必要なクラスを抽出して みれば	単にきちんとしたアル リズムを考えればいい のだ		
16	26	VisualC++ソースプログラムを分析		OMTに戻ってプログラムの 抽出を行おう...	プログラムソースを踏 んでもしっくりこない 書いてやっただけになっ た	
18	28	OMT図を作成				
19	3					
20	2.1					
21	2					
22	3					
23	4	新規クラスを設計する				
24	6					
25	8	自省 細かいたと狂戻り回数を減らしたことの 認識...	抽象OMT、具象OMT、実装テストのプロセス を説明	MFCを利用したプログ ラミングへの不安		
26	9					
27	10	シナリオの検討				
28	14					
29	15					
31	17	自省 従来の開発手順がら脱北していない... 認識...				
32	18	1月6日に提示されたシナリオをも とに設計をやり直す	物事は柔軟と収束が必要			1月6日のシナリオを 検討すべまでであった 予定の流れがも必要 なプログラムは...
33	21					
34	22					
35	23					
37	25					
38	28					
39	3.1					
40	2					
41	3					
42	4					
43	7					
44	8					
45	9					
46	10	OMT図を作成				
47	11	OMT図を作成				

表1. 被験者の設計行動および自省の過程とアドバイスによる相互作用

(2) 背景や意図の理解を促進させ、被験者自らがspiral-upしていくためのアプローチ
前述したプロセス指向アプローチは今回の開発実験では次ぎのように行われた。

- ・今回の実験では、被験者と関わったグループメンバーは被験者の設計プロセスのモニタリングを行い、結果そのものではなく、被験者のプロセスの向上を図ることを主目的としてきた。
- ・被験者へのアドバイスや学習の際には技術の背景や意図の説明を行い、その技術の目指していることについて説明を行い、被験者のプロセスを向上させるメタ・プロセスとなるように調整を行った。
- ・被験者には一切アプローチ方法を指導せず、被験者自らが決定したアプローチを行うように進めた。前述したように不安を回避するために従来のやり方を行うがその方法もうまくいかないために被験者自らが気がつく。

このような環境からの支援により、被験者はSpiral-upしていくためのアプローチを身を持って体験し認識できた。その成果として自省までの期間が短くなり自己改善が早く行われるようになったと考えられる。

以上より、Spiral-upを引き起こしながら、より良い設計に導くためにはプロセス指向開発が行える人物が今回の実験で取った方針を基盤として被験者のモニタリングとアドバイスをを行うことが重要であることが分かった。

4.まとめと今後の課題

ソフトウェア工学やヒューマンインタフェースにおけるヒューマンファクタの研究は、原因追究を行い因果関係を明確にする乖離型問題として扱われているものが多い。一方、CSCW(Computer Supported Cooperative Work)では集団活動や相互作用についての研究が進められており、人間の知的活動は本質的にspiral-upであることが指摘されている[4][5]。そこで、人的要素に関する研究においては対象としている問題自体が、期待された業績からの乖離として把握する乖離型問題か、むしろ新たな高い業績水準に移行するための機会把握型問題かといった分析を行うことを提案する。

今回のソフトウェア開発実験の目的は、ソフトウェアの信頼性に影響を及ぼす要素の一つである人的要素の向上について、クラスライブラリの設計を利用してそのアプローチを模索し、ソフトウェアプロセスの改善を図ることである。そこで、「ソフトウェアの信頼性の向上」「人的要素の向上」「spiral-up(プロセス指向)」といった問題を機会把握型問題と捉えて、人的要因追究は行わずプロセス指向が追究しているように

- ・プロセスを向上させるメタ・プロセスを目標として
- ・結果を生成するプロセスのやり方(少なくとも一つ以上)にフィードバックを行う

といった問題や目標の再規定を行い、その成果としてクラスライブラリの設計の変化を提示した。

今後さらに、実験あるいはソフトウェア開発を利用して、設計行動のモニタリングとアドバイス(あるいはコンサルテーション)を行い自省によるプロセスの変化に関するデータを収集し、学習効果を高め人的要素を向上させていくアプローチを蓄積していく予定である。そこで、CSCWで行われている問題発見・解決プロセスや知的生産性の向上など人間の知的創造に関する研究成果を活用しながら、

- (1) より多くの人のspiral-upを支援し、相手に変化を気づかせるために、変化の周辺を提示することで変化そのものを浮き彫りにできるようなアプローチを多く見つけ出すこと
- (2) 自省(reflection)により行動にフィードバックを行う際の評価基準の設定をどのように行うか

といった課題を追究し、教育や開発プロセスマネジメントなどに反映していきたいと考えている。

現在、工学のあらゆる領域で人間や社会との関係が問い直されている。特に、ソフトウェア工学では設計者の存在は必須であり、設計手法やツールの向上を追求するとともに人的要素の向上については今後ますます議論されていく課題となるであろう。

参考文献

- [1]高橋、古宮、三宅：ソフトウェアの信頼性に影響を及ぼす人的要因の一実験,情処研報 Vol.94,No.43,94-SE-98(1994).
- [2]ハーベイ・J・ブライトマン：グループ戦略思考,プレジデント社(1992).
- [3]江谷：協調活動のフレームワークとしてのプロセス指向アプローチ-問題解決型研究開発におけるspiral-upの考察-,信学技報 Vol.93,No.361,KBSE93-25~32(1993)
- [4]西田：ヒューマン・インタフェースから見た教育・訓練・システムの現状と今後の展望,ヒューマン・インタフェース講習会資料,計測自動制御学会ヒューマン・インタフェース部会(1993).
- [5]江谷：人間の行動分析によるヒューマン・インタフェースのモデル化と実現-人間中心のシステムデザインの提案-,ヒューマン・インタフェース研究論文集 Vo.:3 No.1,自動計測制御学会 ヒューマンインタフェース部会(1993).
- [6]Watts S.Humphrey:ソフトウェアプロセス成熟度の改善,日科技連(1992).
- [7]H.R.マトゥラーナ：オートポイエーシス,国文社(1991)