

グループによるソフトウェア開発演習における内省と協調による 問題解決支援システムの適用

島田和幸 小林祐介 樋山淳雄
東京学芸大学

ソフトウェア開発は知識集約的な協調作業である。ソフトウェア開発では問題解決活動が繰り返し行われる。著者らはグループによるソフトウェア開発演習において発生する問題の解決とその過程を通して獲得した知識を定着させることを目指した、内省と協調による問題解決プロセスモデルとその支援システムを構築した。そして、支援システムをグループによるソフトウェア開発演習に適用した。システムに蓄積されたログとアンケートの結果、1) 登録された問題解決情報の90%はプログラミングに関するものであった、2) 問題解決情報の95%は「開発技術」に関するものであり、「作業の進め方」に関するものは5%であった、3) 問題解決情報並びに質問・回答の共有については受講者全員が意義を認めた、4) 質問件数は少なかったが、それはグループメンバーやネット、問題解決情報で解決できなかったものであり、かなり高度な内容のものであることがわかった。

A Case Study of Applying a Collaborative Problem Solving Support System to a Software Engineering Project Course

KAZUYUKI SHIMADA YUSUKE KOBAYASHI ATSUO HAZEYAMA
Tokyo Gakugei University

Software development is a highly knowledge-intensive and collaborative activity. Problem resolution processes are performed iteratively during software development. The authors proposed a problem resolution process model that was based on reflection and collaboration for a software engineering project course. They also developed a support system and applied it to an actual university course. The results from the stored log data and questionnaire showed that 1) around 90% of the registered problem resolution information dealt with coding, 2) 95% of the registered problem resolution information dealt with technical information and only 5% dealt with process information, 3) all respondents agreed with sharing the problem resolution information and the question and answer information in the whole class, and 4) although the number of questions was small, the questions were not resolved by inquiring colleagues, investigating web pages and/or referring to the problem resolution information of the system, therefore the degree of difficulty was high.

1. はじめに

ソフトウェア開発は協調活動を伴う知識集約的な作業であり[8]、開発において問題解決活動が繰り返し行われる。ソフトウェア開発に従事する個々の開発者は開発に必要なすべての知識を有しているわけではなく、彼らは開発を行っているが、同時に必要な情報の収集を行っている。あるいは、知識を有している他者に問い合わせを行うこともある[8]。葉雲文はソフトウェア開発における知識コラボレーションの重要性を主張し、コーディング作業を対象としたサンプルプログラムの登録・検索、過去の議論の閲覧、専門家への問い合わせを可能とするソフトウェア開発環境を提案している[8]。また、文献[3]では、産業界のソフトウェア開発における情報共有やノウハウ共有の課題が指摘されている。

一方、問題解決活動を通じて獲得した知識を

定着し、学習効果を向上させるプロセスとして内省(reflection)がある。Reidは、内省を、「実践で学んだことを記述し、分析し、評価し、そしてそれを伝えるために実践経験を振り返るプロセス」と定義している[5]。また、失敗から学び、同じような失敗を繰り返さないことを目指した学問に失敗学がある[1]。失敗学では失敗について正しく記述し、その失敗を他者に伝達するために、6つの属性(事象、背景、経過、原因、対処、総括)を定義している。

我々はグループによるソフトウェア開発演習教育に取り組んでいる[4]。この演習では、学習者がグループにより協調してソフトウェア開発を行う経験を通じて、ソフトウェア開発に必要な知識やスキルを習得することを目指している。我々は、グループによるソフトウェア開発演習において発生する問題を解決し、問題解決により獲得した知識を定着させることを目指した、内省と協調による問題解決プロセスモデルを提

案した[2, 6]. 内省のための枠組として失敗学の考え方を導入している. 本論文では, 提案モデルに基づく支援システムの構築とその適用結果について報告する.

本論文の構成を以下に示す. 2 節で問題解決プロセスモデルを概説する. 3 節で支援システムについて述べる. 4 節でシステムの適用評価について述べ, 最後にまとめと今後の課題を述べる.

2. 問題解決プロセス

我々が提案したグループによるソフトウェア開発演習における問題解決プロセスモデルを図 1 に示す. 本プロセスモデルは, ソフトウェア開発において, 開発者が開発において直面する問題を解決する過程で, 失敗学で定義された各属性情報を記述させ, それらを開発者間で共有することによりソフトウェア開発における問題解決に必要な知識やノウハウを習得することを目的としている.

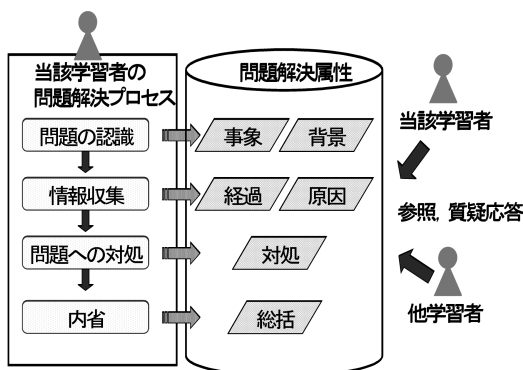


図1 問題解決プロセスモデル

本研究が想定する問題解決プロセスは, 問題の認識, 情報収集, 問題への対処, 内省のステップからなる.

(1) 問題の認識

開発者(あるいは開発者グループ)は問題に直面したとき, その事象, 背景を識別する. 問題として, インспекションやテストにより指摘された事項, プログラミングにおける不具合, 開発環境構築におけるトラブル等が考えられる.

(2) 情報収集

開発者(あるいは開発者グループ)は問題解決に必要な情報を収集し, 問題の原因を検討する. 情報源として, 本研究で構築する支援システムあるいは外部リソースに蓄積されている事例や議論が考えられる. また, 演習に参加している他開発者(グループメンバー, 他グループのメ

ンバー)や教授者(TAや教員)とのコミュニケーションにより情報を収集することも想定する.

(3) 問題への対処

(2)で収集した情報に基づき検討した原因を解決するために対処を行う.

(4) 内省

問題が解決できた場合には, 問題解決プロセスを振り返り, 学んだこと(教訓)を記録として残す.

以上の問題解決プロセスを経て蓄積された問題解決情報は他の開発者にも参照可能となるよう共有される. 他開発者が情報を参照した場合にフィードバック情報(評価やコメント)を提供することが期待される.

表1 失敗学で定義された属性の本研究における定義

属性	失敗学における定義	本研究による定義
事象	どのようなことがあったかに関する記述	発生した問題の内容に関する記述(エラーメッセージ等)と問題を含む成果物
背景	失敗の事象が生じた背景に関する記述	問題発生時の(開発・実行)環境・開発工程・ユースケース
経過	失敗の進行に関する記述	要求事項と, 問題解決のために参照した情報
原因	失敗の原因に関する記述	問題の原因に関する記述
対処	失敗に対してどのようなことを行ったのかに関する記述	問題解決のために行ったことに関する記述と問題解決後の成果物
総括	失敗から学んだことに関する記述	問題解決から学んだ教訓に関する記述

3. 支援システム

2 節で述べた問題解決プロセスを支援するシステムを構築した. 本システムは分散環境においても利用できるよう Web アプリケーションとして, またグループによるソフトウェア開発演習支援システム“心技体”[4]のサブシステムとして構築した.”心技体”は, グループを単位とした成果物管理, コミュニケーション支援, レビュー等のプロセス支援を提供している. 本サブシステムはグループの枠を越えて, 問題解決情報の共有や問題解決に関する協調を可能にしている. 以下に主要な機能を説明する.

■ 問題解決情報登録機能

問題解決情報の登録を行う. 登録者が表 1 の属性に従って入力することにより, 問題に関するコンテキストを記述することができ, また内省

を喚起することが期待される。

■ 問題解決情報参照機能

登録されている問題解決情報を参照する。参照画面からは、当該問題解決情報に対する評価やコメントを登録することができる。図2に問題解決情報参照画面を示す。

【ノウハウ閲覧画面】		
タイトル	Eclipseのビルド・バリエーション??	環境設定 tomcat: 5 mysql: 5 java: 1.4 eclipse: 3.2
工程	コーディング・単体テスト	担当コース ケース
背景・問題	データベースとの接続ができるようになったら、Eclipseが自動的にビルドして行けなくなった。	原因 いらないmysqlコネクタが入ってしまっていて、ビルド・バリエーションが型指定していた。
対処	・いろいろな人に相談 ・ネットで見つけた有用な情報(なかった) ・コマンドを再利用エラーの所在を突きとめる	考察 どこにエラーがでているのかを知ること、エラーの解決につながる。 コマンドを再利用したことで早い解決につながりました! ほんとうにありがたございます!
参照Webサイト	なし	キーワード Eclipse ビルドされない
添付ファイル	なし	議論内容へのリンク リンク

編集する 問い合わせ/コメント付加

2人中2人が
このノウハウは役に立つと登録しています

図2 問題解決情報参照画面

■ 質問登録機能

自身では解決できない問題について、質問を登録するための機能である。回答者が質問内容を的確に把握し、質問・回答を効率的に行えるよう、ここでも失敗学の属性の一部を活用した。事象については、KT法[7]の問題分析手法の属性も取り入れ、現在起こっている問題と実現させたいことに項目を分け、質問内容の構造化を図った。

【議論閲覧画面】		
登録者	akko	タイトル Eclipseのエラー??
背景・前面している問題 (実視したいこと)	Eclipseでファイルを作成⇒更新しています。更新した後、ファイルを実行してもその更新がなされていません。	背景・前面している問題 (実視の状況)
どんな対処を困ったか	ファイルを書き換えた後、 ・保存 ・Tomcatの再起動 ・Eclipse再起動 ・バッチコマンドの再起動 しましたが状況が変わりません。	考え得る原因 クラスが更新されていないので、コンパイルができていないのかも?
工程	コーディング・単体テスト	ユースケース 分科名登録など
開発環境	java: 1.4 mysql: 5 eclipse: 3.2.1 tomcat: 5	添付ファイル なし

1.<0>/	2007/12/21 08:37
ビルドされていないんじゃないですか?	
2.akko	2007/12/21 12:46
たぶんビルドされてないです...	
自動ビルドにチェックが入っている状態です。チェックをすべてビルドどか試してみましたが動いていないみたいです。どこを直したらいいかわかりません。	

図3 質問とそれに対する回答参照画面

■ 質問に対する回答登録機能

登録されている質問を解決するための議論を行うための機能である。議論の経過を閲覧することも可能である(図3)。議論により質問内容が解決した場合に、質問投稿者は問題解決情報を登録することが期待される。問題解決情報登録を効率化するために、質問時の属性の内容を引き継ぐことができる。また、質問・回答を経て登録された問題解決情報からは、問題解決に至った質問・回答も閲覧することができる。

4. 適用評価

本研究で提案した問題解決支援システムの有効性を検証するために、大学におけるソフトウェア開発演習に適用を行った。本節では適用対象、結果とその分析について述べる。

4.1 適用対象

開発したシステムを東京学芸大学教育学部情報教育専攻の3年次後期に開設されている「システム設計演習」の2007年度演習に適用を行った。本演習では4名から5名でグループを構成し、各グループが教授者から与えられた課題を実現するWEBアプリケーションシステムを開発するというものである。演習では要求分析からはじまり設計、実装、テストという工程を経てソフトウェアを完成することが求められる。上流工程の成果物に対してインスペクションが、開発されたシステムに対して受入テストが教師並びにTAによって行われている。2007年度の受講者は22名で、5つのグループが編成された。システムの適用期間は2007年10月31日から2008年1月22日までの84日間であった。開発スケジュール策定はグループに委ねられているが、クラス全体では表2に示すマイルストーンが設定された。従って、グループにより多少の違いはあるが、12月中旬まではシステム分析・設計作業を行い、12月中旬から1月初旬にかけてコーディング、単体テスト、システム結合を行い、1月中旬以降がシステムテストとバグ修正の時期と見なすことができる。受講者には演習開始時に、開発演習で得た問題解決情報を本システムに登録するよう指示を行った。なお、前年度(2006年度)の演習で収集した問題解決情報が13件あり、それらは事前にシステムに登録されている。

表2 適用対象の主要マイルストーン

主要マイルストーン	期日
設計インスペクション受付終了	2007年12月11日
システム仮提出	2008年1月10日
開発完了	2008年1月22日

4.2 結果と分析

本節では適用結果を報告する。以下のデータは、システムに蓄積されたログデータと、演習終了後に受講者に対して行ったアンケートによる。アンケート項目を表3に示す。アンケートの回答は20名からあった(回答率：91%)。

表3 アンケート項目

項番	質問内容
PS-1	問題解決情報を登録することで、自身の問題解決プロセスを振り返ることはできましたか(4「できた」、…、1「できなかった」)
PS-2	問題解決情報をシステムに登録する際の記述量はいかがでしたか(4が「多かった」、…、1「少かった」)
PS-3	自身の登録した問題解決情報に評価が加えられることは、モチベーションの向上につながると感じますか(4「そう思う」、…、1「そう思わない」)
PS-4	参照した問題解決情報を、実際の問題解決に役立てることはできましたか(4「できた」、…、1「できなかった」)
PS-5	問題解決情報を参照することで、自身の知識を増やすことはできましたか(4「できた」、…、1「できなかった」)
PS-6	問題解決情報をクラス全体で共有することは有益だと思いますか(4「そう思う」、…、1「そう思わない」)
QA-1	質問を規定された形式で記述することで、自身の直面した問題を正確に伝えることはできましたか(4「できた」、…、1「できなかった」)
QA-2	質問・回答機能で議論(問題解決)を行うことで、自身の持つ知識の整理や補完に役立ったと思いますか(4「そう思う」、…、1「そう思わない」)
QA-3	質問をシステムに登録する際の記述量はいかがでしたか(4「多かった」、…、1「少かった」)
QA-4	学生が直面した問題をクラス全体で共有し、議論・問題解決を行うことは有益だと思いますか(4「そう思う」、…、1「そう思わない」)

図4は2週間ごとの問題解決情報の登録件数とその累積件数を示したものである。最終的に40件の問題解決情報が登録された(受講者全員が最低1件は問題解決情報を登録していた)。図4から期間別に見ると、1/16-1/22の期間が最も多く15件登録されている。これは「演習開始時に、開発演習で得た問題解決情報を本システムに登録するように、という指示」に対して、開発完了直前に駆け込んで登録したものと考えられる。これらの情報のほとんどは当該年度の開発中には有効活用はされなかったと考えられる。

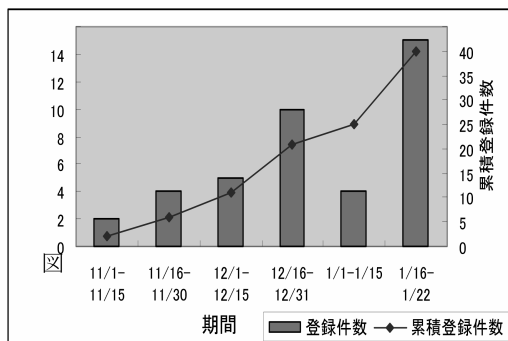


図4 登録された問題解決情報の推移

図5は2週間ごとの問題解決情報参照回数とその累積回数を示したものである。最終的に、登録数に比べて参照数が35倍であったことが分かる。図6は2週間ごとの質問登録件数とその累積件数を示したものである。質問は合計で8件にとどまった。そのすべてが解決を見た(7件は質問・回答から問題解決登録に移行していた。1件は質問・回答から問題解決登録に移行してはなかったが、質問者により質問内容に関する問題解決情報が登録されていた)。図7は2週間ごとの質問参照回数とその累積回数を示したものである。質問登録数に対して、参照数は60倍であったことがわかる。図8は2週間ごとの質問に対する回答投稿数とその累積を示したものである。

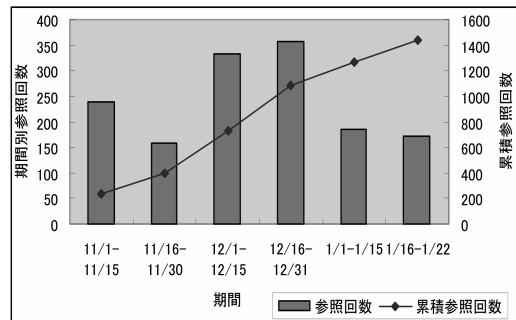


図5 登録された問題解決情報への参照数の推移

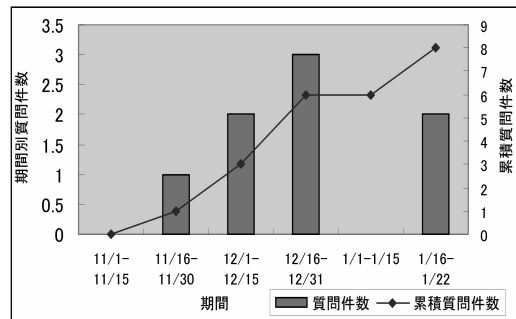


図6 質問登録件数の推移

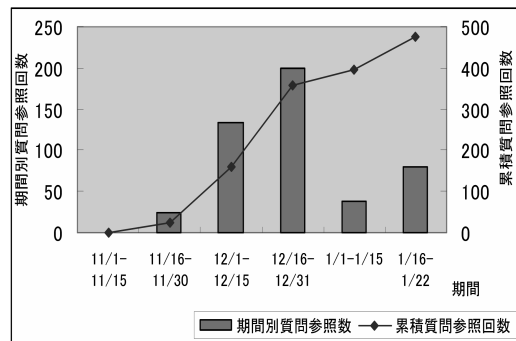


図7 登録された質問への参照数の推移

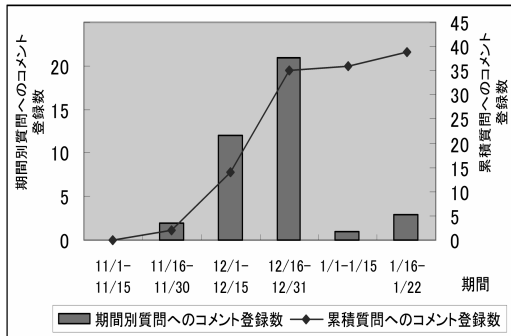


図8 質問に対する回答投稿数の推移

図9は登録された問題解決情報の内容を著者が読み、どの開発工程に関する情報かを分類したものである。開発環境に関するものも含めると、全体の90%以上がプログラム作成に関する内容であった。本演習ではプログラム作成は、個人単位の作業になり、それぞれに必要な知識や技術が異なっている。そのため、グループ内で知識共有を行いにくく、自分が実現したいことに関する知識がない、コンパイルが通らない、正しく実行されないというように、問題が顕在化しやすい。さらに、システム実現に必要なすべての技術や知識を講義でカバーできないということも割合の大きさの原因になっていると考えられる。一方、分析や設計等の上流工程ではグループでの活動が中心でありグループとして問題解決ができる、インスペクションでの指摘に対応すれば問題が解決できるということが登録件数の少なさに関係していることが考えられる。ただ、現時点で分析・設計作業に関する問題解決情報が少なかったことに関する明確な原因を把握できていない。このことについては今後調査して行く必要がある。

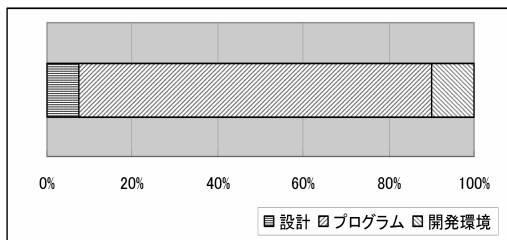


図9 登録された問題解決情報の工程による分類

文献[3]では、ノウハウを「進め方」と「作り方」に大別して考察を加えている。そこで、本システムに蓄積されている問題解決情報をこの2つに分類することを試みた。その結果38件(95%)は「作り方」(ツールの使い方も含む)に関する

ものであり、「進め方」に関するものは2件(5%)であった。「進め方」に記述された2件は技術的問題に遭遇した時の解決方法(分かる人に聞くことと自分で調べることを繰り返す、対面での打ち合わせに可能な限り出席して分かっている人に聞く)であった。

図10は、表3で示したアンケートの結果である。アンケート項目PS-6とQA-4に対する回答結果から、全回答者が問題解決情報ならびに質問・回答をクラス全体で共有することの意義を認めている。PS-1に対する回答結果から、約80%が問題解決情報の登録を通じて自身の問題解決プロセスを振り返ることができたと回答しており、提案手法の有効性を確認できた。PS-4に対する回答結果から、約80%が登録された問題解決情報が自身の問題解決に役立ったと回答している。これにより、登録された問題解決情報が再利用されていることが明らかになった。

また、QA-2に対する回答結果から、質問者全員が質問したことの有用性を評価している。評価が割れたのは質問時の記述量である(QA-3に対する回答結果)。質問者のうち40%はシステムで記述項目が多いと回答した。一方60%は必ずしも多くないと回答している。議論の経過を追ってみると、2つの質問に対して回答者から何に困っているのか詳細な説明を要求する書き込みがあった。

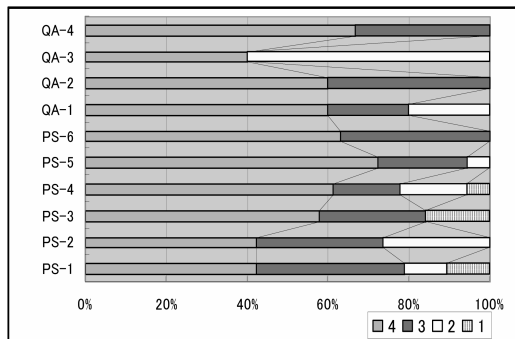


図10 アンケート結果

アンケート項目QA-4に対する回答結果から、質問・回答を共有する意義について回答者全員が認めたにも関わらず、今回の適用では問題解決情報に対して、質問登録数が極めて少なかった。その理由として以下の回答があった。

- ①グループメンバーに聞くことで、問題が解決した
- ②問題解決情報共有機能を利用することで、事足りた
- ③インターネットや書籍を使って、自分自身で解決できた

逆に言うと、今回質問として登録されたものは、上記3つには当てはまらないものであったかを分析する。

ある質問では「とにかくネットで調べまくっています。グループの人に聞いてもみんなよくわからないみたいです」という記述があり、グループ内に当該の問題について解決策を有しているメンバーはおらず、インターネット等でも解決策を見出せていないものであった。また、別の質問として「DB に日付データを格納する際の問題」に関するものがあつた。この質問は開発経験者とのやり取りで解決に至っているが、開発経験者もインターネットを使ってかなりの調査をして解決策を見つけ出していることから、この質問も上記3つに当てはまらないものであつたと考えられる。他にも、例外状況下でのエラーや、サンプルプログラムはあるものの、自分の思い通りの挙動をさせようと改良を試みたがうまく動作しないというものもあつた。このように、今回の適用では、質問は上記3つに当てはまらない高度な内容や複雑な条件下での問題に対するものであることがわかつた。そのため、質問・回答の過程で交わされた40件のメッセージのうち、質問者による状況説明や問題解決後の感謝を表す14件を除くと、回答者はTA(のべ20件)と先行して開発を進めている他グループのメンバー(のべ6件)と限られた範囲にとどまつた。質問に対する議論に参加しなかつた理由として、「答えるだけの知識が無かつた」や「自分の持つ知識に自信が無かつた」は、このことを裏付けた発言である。

5. おわりに

本論文では、グループによるソフトウェア開発演習における内省と協調による問題解決支援システムについて述べた。そして、支援システムを大学の演習において適用した結果について述べた。適用の結果、以下のことがわかつた。

- ・登録された問題解決情報の90%はプログラミングに関するものであつた
- ・問題解決情報の95%は文献[3]の分類による「作り方」に関するものであり、「進め方」に関するものは5%であつた
- ・問題解決情報並びに質問・回答の共有については全員が意義を認めた
- ・質問件数は少なかつたが、それはグループメンバーやネット、問題解決情報で解決できなかったものであり、かなり高度な内容のものであることがわかつた

今後も継続的に運用を続け、データ収集と分析を行うとともに、現時点で明確にできていない分析・設計段階での問題解決情報の有無についても調査を進めて行きたい。

謝辞

本研究の一部は、科学研究費補助金(C)18500701の助成のもとに行われている。記して謝意を表す。

参考文献

- [1] 畑村洋太郎, 失敗学のすすめ, 講談社, 2000.
- [2] 樋山淳雄, 高山知佳, 小林祐介, 大瓶佳秀, グループによるソフトウェア開発演習における内省と協調による学習支援, 情報処理学会グループウェアとネットワークサービスワークショップ (GNワークショップ2006), pp. 1-6, 2006.
- [3] 石田厚子他, ソフトウェア開発における情報共有の課題と効果に関する研究, http://www.juse.or.jp/software/pdf/17_spc/17dep71.pdf
- [4] 三浦真人, 小林祐介, 島田和幸, 高橋晃一, 清木進, 樋山淳雄, 個人とコミュニティの支援を有するソフトウェア開発グループ演習環境の提案, 情報処理学会研究報告グループウェアとネットワークサービス 2007-GN-64(4), pp. 19-24, 2007.
- [5] B. Reid, "But we're doing it response to the concept of reflective practice in order to improve its facilitation. Nurse Ed Today, Vol. 13, pp. 305-309, 1993.
- [6] K. Shimada, Y. Kobayashi, and A. Hazeyama, A Collaborative Problem Solving Method for A Software Engineering Project Course and Its Environment, Proceedings of the First Workshop on Knowledge Reuse (KREUSE2008), Beijing, China, 2008.
- [7] Q. Spitzer, and R. Evans, 小林 薫 (訳), 問題解決と意思決定—「ケブナー・トリゴアの思考技術」, ダイアモンド社, 1998.
- [8] Y. Ye, Socio-Technical Support for Knowledge Collaboration in Software Development Tools, Proceedings of the Workshop on Integrating Software Engineering and Usability, pp. 39-51, 2005.