

協調ソフトウェア開発における coordination communication 支援環境の要件の検討

丸山憂奈^{†1} 今野翔太郎^{†2} 樫山淳雄^{†1}

ソフトウェア開発におけるコミュニケーションは、2つの異なる目的 (1) 作業のコーディネーションに関するコミュニケーション (coordination communication), (2) 自分の問題解決に関するコミュニケーション (expertise communication) に分類でき、また、目的に応じてコミュニケーションの支援指針が異なることがすでに述べられている。本研究では、ソフトウェア開発におけるコミュニケーションの重要性、コミュニケーションタイプに応じた支援指針をふまえ、coordination communication 支援環境に求められる要件を検討する。そして、その要件を満たすと考えられる Google+ハングアウトに着目し、ソフトウェア開発演習の授業で適用実験を行う。適用実験の結果をふまえ、今後の coordination communication 支援環境の要件モデルを示す。

Investigation of Requirements for Supporting Environment of Coordination Communication in Collaborative Software Development

YUNA MARUYAMA^{†1} SHOTARO KONNO^{†2} ATSUO HAZEYAMA^{†1}

It is described that communication in software development can be classified into two different purposes, (1) communication for coordination of work (coordination communication), and (2) communication of problem solving of their own (expertise communication). Further, it is also described that guidelines for supporting communication are different depending on the purpose. This study, based on the importance of communication in software development and the guidelines depending on the communication type, examines the environmental requirements required for supporting coordination communication. Then, this study focused on "Google+ hangout" to fulfill the environmental requirements, and we performed an experiment that supported collaborative software development. Based on the results, evaluation and consideration of the demonstration experiment, this study shows the environmental requirements model for supporting coordination communication in the future.

1. はじめに

ソフトウェア開発の現場では、複数人のグループによる開発が主流であり、それゆえに開発者同士のコミュニケーションが必要不可欠である。

文献[1]では、ソフトウェア開発におけるコミュニケーションの目的を分析し、2つの異なる目的 (1) 作業のコーディネーションに関するコミュニケーション (coordination communication), (2) 自分の問題解決に関するコミュニケーション (expertise communication) を定義している。また、文献[2]では、expertise communication 支援に対するガイドラインについて述べられている。しかし、これらの文献では、コミュニケーションの分類や、おおまかな支援指針を述べることにとどまっている。

本研究では、文献[2]で述べられているソフトウェア開発におけるコミュニケーションの重要性、コミュニケーションタイプに応じた支援指針をふまえ、coordination communication 支援環境に求められる要件を検討する。そして、coordination communication 支援の要件を満たすと考えられる既存のコミュニケーションツール Google+ハングア

ウトに着目し、適用実験を行う。その適用実験の結果をふまえ、今後の coordination communication 支援環境に求められる要件について考察する。

2. 関連研究

2.1 ソフトウェア開発におけるコミュニケーション

中小路らは、ソフトウェア開発におけるコミュニケーションに至る情報ニーズの目的を分析し、coordination communication と expertise communication に分類した[1]。

coordination communication とは、作業の調整を目的とするコミュニケーションと定義されている。coordination communication は、開発者がリスクを回避するために依存関係にある仲間とタスクを調整することや、出現した衝突や潜在的な衝突を解決するために行われる。そのため、コミュニケーションの発起人と他のコミュニケーション参加者との間には対称な関係があり、それぞれに等しくコストとベネフィットがある。

expertise communication とは、自分の問題解決を目的とするコミュニケーションと定義されている。expertise communication は、開発者が差し迫ったタスクを解決するための情報を探し求めたり、仲間に助けを求めたりするために行われる。expertise communication で得た情報によって利益を得るのはコミュニケーションの発起人となる開発者の

^{†1} 東京学芸大学
Tokyo Gakugei University
^{†2} 東京学芸大学大学院
Tokyo Gakugei University

みである。そのため、コミュニケーションの発起人にベネフィットが偏ってしまい、他の参加者には大きなコストがかかってしまう。

2.2 インタラクシオンデザインに関する研究

中小路らは、coordination communication と expertise communication の違いをふまえて、コミュニケーションを支援するツールのインタラクシオンデザインの研究を行っている[2]。中小路らは、コミュニケーション支援ツールのインタラクシオンデザインを考える際に、コストとベネフィットの関係の違いは考慮しなければならないと考えている。コストとベネフィットの関係に関わる項目として、コミュニケーションを行う相手の選定の仕方、コミュニケーションのタイミング、コミュニケーションへの招待の仕方、用いるコミュニケーションメディア、コミュニケーション結果の利用の仕方、などが文献[3]で挙げられている。

中小路らは、コストとベネフィットの関係を考慮して、expertise communication を支援するインタラクシオンデザインの 9 つのガイドライン[2]および coordination communication と expertise communication の比較を行いながら coordination communication を支援するための要件を考察している[3] (表 1)。

表 1 2 種類のコミュニケーションのタイプとそのためのデザインガイドラインの比較 (文献[3]引用)

expertise communication 支援の デザインガイドライン	coordination communication 支援適用の可否
(1) コミュニケーションが他の開発作業から乖離しないこと	Yes
(2) どの開発者が何についてコミュニケーションをしたいのか、を考慮すること	Yes
(3) 同じ情報を得るためにドキュメントやコードがあればそちらを優先し、コミュニケーションがなるべく起こらないようにすること	No リスクに応じて促進
(4) 情報を欲している開発者のメリットと、プロジェクト全体の、グループとしてのメリットを考慮すること	No リスクに応じて促進
(5) コミュニケーション相手の選定にあたっては、聞き手との組織的、社会的関係を考慮すること	No 技術的依存性を考慮
(6) コミュニケーション相手として選定するにあたっては、その人の作業の中断を最小限に抑えること	Yes
(7) 質問し易い仕組みをつくること	No コミュニケーションを開始し易い仕組み
(8) 回答するかしないかを簡単に選べる仕組みをつくること	No 緊急性、重要性を判断する仕組み
(9) コミュニケーションチャンネルが、社会性を考慮したもの (socially-aware) であること	No impact-aware な仕組み

3. 開発工程とコミュニケーション

ソフトウェア開発のプロセスモデルとして、ウォーターフォールモデルがある。ウォーターフォールモデルによる開発は、おおきく上流工程 (要件定義/外部設計/内部設計) と下流工程 (コーディング/テスト) に分けられる。本節で

は、2 つの工程でみられるコミュニケーションの目的の違いを検討する。

上流工程では、開発者間および開発者と利用者間でのイメージの共有や理解の相違の解消をし、その後の工程をスムーズに進めるためのコミュニケーションが多くみられる。これは coordination communication に分類される。coordination communication は、コミュニケーションの発起人と参加者それぞれに等しくコストとベネフィットがあり、開発に良い影響を与えると考えられるため、積極的にコミュニケーションを行うことが望まれる。

一方、下流工程では、技術面の疑問を解決するコミュニケーションが多くみられる。これは expertise communication に分類される。expertise communication は、コミュニケーションに参加する際の立場によってコストとベネフィットが偏ってしまうため、なるべく避けることが望まれる。

ソフトウェア開発における関係者同士のコミュニケーションの支援は、その目的に応じて行われるべきである。特に、作業の調整を目的としたコミュニケーションは、ソフトウェア開発に良い影響を与えると考えられる。そのため、本研究では上流工程での coordination communication に分類できる目的をもったコミュニケーションの支援に着目し、coordination communication 支援環境の要件の検討を行っていく。

4. coordination communication 支援環境の要件

4.1 coordination communication 支援環境の要件の検討

2 節で紹介したデザインガイドライン (表 1) をもとに、coordination communication を支援する環境に求められる要件を検討する。

- (1) コミュニケーションが他の開発作業から乖離しないために、他の開発作業と同じ場でコミュニケーションができる環境を整えることが必要となる。
- (2) 誰が何についてのコミュニケーションをしたいのかを考慮するために、グループ内で誰が何の作業を担当しているのか、作業がどのような状況か、どの作業に依存関係があるのかを分かりやすく把握できる環境が必要となる。
- (3) ドキュメントやコードがあったとしても、リスクに応じてコミュニケーションを促進するためには、作業の中断によるコストを最小限に抑えコミュニケーションを行いやすい環境が必要となる。
- (4) (3) と同様に、リスクに応じてコミュニケーションを促進するためには、作業の中断によるコストを最小限に抑え、コミュニケーションを行いやすい環境が必要となる。
- (5) 技術的依存性を考慮してコミュニケーション相手を選定するためには、開発者間の作業の依存関係が分かりやすく把握できる環境が必要となる。

- (6) コミュニケーション相手の作業の中断を最小限に抑えるためには、コミュニケーションに伴うコストを最小限に抑えるよう、空間的制約や時間的制約に縛られることなくコミュニケーションが行いやすい環境が必要となる。
- (7) コミュニケーションを開始し易い仕組みとして、コミュニケーションに伴うコストを最小限に抑えられるよう、空間的制約や時間的制約に縛られることなく、コミュニケーションが行いやすい環境が必要となる。
- (8) 緊急性や重要性を判断する仕組みとして、コミュニケーションの手段を選択できる環境が必要となる。
- (9) impact-aware な仕組みとして、コミュニケーションを行う/行わないことによる影響に気付くコミュニケーションチャンネルが確立している環境が必要となる。

これらをふまえ、coordination communication 支援環境には以下の項目が必要だと本研究では考える。各項目の末尾に対応するガイドラインの番号を示す。

- i :他の開発作業と同じ場でコミュニケーションができる環境（作業と同じ場と記す）：(1)
- ii :グループ内で誰が何の作業を担当しているのか、作業がどのような状況か、どの作業に依存関係があるのかを分かりやすく把握できる環境（作業の担当・依存関係と記す）：(2), (5)
- iii :作業の中断を最小限に抑え、コミュニケーションを行いやすい環境（作業の中断と記す）：(3), (4), (6)
- iv :空間的制約や時間的制約に縛られることなく、コミュニケーションを行いやすい環境(コミュニケーションの時空間制約と記す)：(6), (7)
- v :緊急性や重要性を判断する仕組みとして、コミュニケーションの手段を選択できる環境(コミュニケーションの手段と記す)：(8)
- vi :コミュニケーションを行う/行わないことによる影響に気付くコミュニケーションチャンネルが確立している環境（コミュニケーションへの気付きと記す）：(9)

4.2 支援環境要件を満たした開発支援環境の検討

4.1 項で検討した coordination communication 支援環境の要件を満たす、開発支援環境に求められる要件を検討する。

- i（作業と同じ場）:他の開発作業と同じ場でコミュニケーションができる環境を満たす機能として、成果物、開発計画、進捗などの管理や、グループ内でのコミュニケーションが1つのツール上で行えることが求められる。
- ii（作業の担当・依存関係）:グループ内で誰が何の作業を担当しているのか、作業がどのような状況か、どの作業に依存関係があるのかを分かりやすく把握できる環境を満たす機能として、開発計画や進捗管理でメンバーがどのような作業をしたのか、どのような状況なのか把握できることが求められる。

- iii（作業の中断）:作業の中断を最小限に抑え、コミュニケーションを行いやすい環境を満たす機能として、コミュニケーションを行う際の物理的制約が少ないことが求められる。
- iv（コミュニケーションの時空間制約）:空間的制約や時間的制約に縛られることなく、コミュニケーションを行いやすい環境を満たす機能として、場所や時間に縛られずに行える文字によるコミュニケーション、音声によるコミュニケーションができることが求められる。
- v（コミュニケーションの手段）:緊急性や重要性を判断する仕組みとして、コミュニケーションの手段を選択できる環境を満たす機能として、複数のコミュニケーション手段を選択できることが求められる。
- vi（コミュニケーションへの気付き）:コミュニケーションチャンネルが確立している環境を満たす機能として、プロジェクトメンバーとのコミュニケーション手段が確立していることが求められる。

以上の要件をすべて満たした開発支援環境を検討したところ、既存のコミュニケーションツール Google+ハンガアウト[4]がこれらの要件を満たす機能を備えている可能性があることが分かった。そこで本研究では、Google+ハンガアウトに着目し、適用実験を行う。

4.3 Google+ハンガアウト

本項では、4.2 項で議論した開発支援環境に求められる要件と、それに対応する Google+ハンガアウト[4]の機能を紹介する。Google+ハンガアウトでは、チャットやビデオ会議を1つのツール上で行うこと、アプリで機能を拡張することができる。アプリ機能のうち、Symphonical は付箋とマトリクスを使って議論を行うツール、Cacoo は図の作成・公開を行うツールである。開発支援環境に求められる要件に対応する Google+ハンガアウトの機能を表2に示す。

表2 支援環境を満たす要件とそれに対応する Google+ハンガアウトの機能

開発支援環境要件	対応する機能
i（作業と同じ場）	アプリ（Cacoo, GoogleDrive）
ii（作業の担当・依存関係）	アプリ（Symphonical）
iii（作業の中断）	携帯端末での利用
iv（コミュニケーションの時空間制約）	ビデオ通話、チャット
v（コミュニケーションの手段）	ビデオ通話、チャット
vi（コミュニケーションへの気付き）	グループトーク

5. 開発支援環境としての Google+ハンガアウトの適用実験

5.1 適用実験の概要

(1) 適用実験の目的

本実験の目的は、(a) 開発支援環境としての Google+ハンガアウトの有効性を調べること、(b) 今後の coordination communication 支援環境に求められる要件の検討をするためのデータを収集することの2点である。

(2) 適用実験の対象

適用実験は、東京学芸大学教育学部初等教育教員養成課程情報教育選修および情報教育課程情報教育専攻3年生を対象に、秋学期に開講されている「システムプログラミング」の2013年度受講者17名(4グループ)を対象に実施した。受講者の大半はこれまでにグループでのソフトウェア開発経験はない。

この科目では、グループによるソフトウェア開発演習を行っている。開発プロセスは、ウォーターフォールモデルに準拠しており、学生である開発者が、教授者から与えられた課題に対して、要求分析に始まり、設計、実装、テストというソフトウェア開発ライフサイクルを経て、ソフトウェアシステムを完成させるものである。また、この演習での学習目標は実社会で行われているソフトウェア開発の多くの活動を体験することである[5]。そのため、本研究では、ソフトウェア開発演習と、実社会で行われているソフトウェア開発の活動には大差はないと考えている。実社会で行われているソフトウェア開発と本研究でのソフトウェア開発演習の主な違いとしては、対象となる開発者の開発経験の差、コスト管理の有無、守秘義務の有無が挙げられる。

(3) 適用実験の内容

適用実験を行うソフトウェア開発演習の内容を述べる。

●課題について

本ソフトウェア開発演習の課題は、Webドリルシステム、論文投稿・審査管理システムである。

●開発支援環境について

本ソフトウェア開発演習では、授業時間以外での活動支援のために、グループにGoogle+ハングアウトもしくは心技体[5]を開発支援環境として提供した。開発支援環境は、コミュニケーションやレビュー、成果物の共有などに用いた。

●グループについて

受講者のグループ分けは、初回授業のアンケートおよび、春学期に開講されている「情報システム設計」の授業での成績をもとに、教授者によって行われた。取り組む課題と用いる開発支援環境は、各グループの話し合いによって決められた。本ソフトウェア開発演習のグループ、選択した課題、利用環境を表3に示す。

表3 本ソフトウェア開発演習のグループとその利用環境

グループ	課題	開発支援環境
G1 (5名)	Webドリル	ハングアウト
G2 (4名)	論文投稿・審査管理	ハングアウト
G3 (4名)	Webドリル	心技体
G4 (4名)	Webドリル	心技体

●レビューについて

本ソフトウェア開発演習では、上流工程で作成される成果物に対して教授者とティーチングアシスタントによるレビューが行われる。

●収集したデータについて

適用実験では、Google+ハングアウトおよび心技体(5.2項参照)の履歴、上流工程終了時に行ったアンケートを収集し、結果・評価・考察に用いた。

アンケートの項目は、各機能の利用状況(1:使わなかった~4:よく使った)、各機能を利用した目的(選択式、複数選択可)、各機能の評価(1:不満足~4:満足)、各機能に対するコメントとした。

5.2 心技体

心技体[5]は、著者らの研究室で開発してきた協調ソフトウェア開発支援システムである。このシステムは、協調ソフトウェア開発を行う際の成果物やコミュニケーションの記録を一元管理することを目的としたものであり、Webブラウザ上で利用することができる。心技体には、成果物との関連付けを行うことのできる電子掲示板(BBS)機能、成果物管理機能、議事録機能、計画立案機能、進捗報告機能、障害管理機能が備わっている。心技体は、4.2項で議論した開発支援環境に求められる要件のうち、「作業の中断」、「コミュニケーションの時空間制約」、「コミュニケーションの手段」を満たしていない。

5.3 Google+ハングアウトと心技体の機能の対応

本研究でのソフトウェア開発演習で用いる心技体の機能に対応するGoogle+ハングアウトの機能を表4に示す。

表4 心技体とGoogle+ハングアウトの機能対応表

分類	心技体の機能	Google+ハングアウトの機能
コミュニケーション	BBS機能	チャット機能、ビデオ通話機能
成果物共有	成果物管理機能	GoogleDrive
議事録作成	議事録機能	該当なし (GoogleDriveで管理)
開発計画作成	計画立案機能	Symphonical
進捗報告	進捗報告機能	Symphonical
その他(図の作成等)	該当なし	画面共有機能、Cacoo

6. 適用実験の結果・評価・考察

6.1 適用実験の結果

(1) ツールの記録から

Google+ハングアウトの各機能の利用回数は(G1/G2)の順に、ビデオ通話機能(4回/2回)、チャット機能(432回/215回)、GoogleDrive(98回/62回)であった。Cacoo、画面共有機能、SymphonicalはG1、G2ともに利用がなかった。

心技体の各機能の利用回数は(G3/G4)の順に、BBS機能(81回/71回)、成果物管理機能(48回/28回)、議事録機能(14回/13回)であった。計画立案機能、進捗報告機能はG3、G4ともに利用がなかった。

(2) アンケートから

Google+ハングアウトの各機能の評価(グループメンバーによる平均)は(G1/G2)の順に、ビデオ通話機能(3.2/3.4)、チャット機能(3.1/3.0)、GoogleDrive(3.5/3.8)であった。コメントからは、「チャットは携帯でも手軽に連絡ができてよかった」、「ハングアウトは通知が来るのがよかった」、

「GoogleDrive では成果物の共有は簡単だが変更点などの比較が不便」, 「Cacoo・画面共有機能・Symphonical は使い方が煩雑だった」などの意見が得られた。

心技体の各機能の評価(グループメンバーによる平均)は(G3/G4)の順に, BBS 機能(4.0/3.5), 成果物管理機能(3.5/4.0), 議事録機能(3.3/3.7)であった。コメントからは, 「BBS は教授者からのレビューのみに用いてグループ内のコミュニケーションは別で行った」, 「計画や進捗は別の連絡手段(LINE, facebook)などで共有をしていた」などの意見が得られた。

6.2 適用実験における Google+ハンガアウトの評価

コミュニケーション支援ツールとしての Google+ハンガアウトの評価ならびに心技体との比較について述べる。

i (作業と同じ場):Cacoo の利用によって, 成果物の作成とグループの議論が 1 つのツール上でできることを満たすと考えていたが, Cacoo の使い方が煩雑で利用が避けられたため, 有効性は確認できなかった。また, GoogleDrive の利用によって, 成果物の管理とグループの議論が 1 つのツール上でできることが確認できた。しかし, 版管理が機能として提供されていないという問題が顕著であった。

ii (作業の担当・依存関係):Symphonical の利用によって, 開発計画や進捗管理においてメンバーの状況を把握できることを満たすと考えていたが, Symphonical の使い方が煩雑で利用が避けられたため, 有効性は確認できなかった。

iii (作業の中断):携帯端末で利用できることによって, 心技体に比べコミュニケーションを行う際の物理的な制約が少ないことが確認できた。

iv (コミュニケーションの時空間制約):ビデオ通話, チャットの利用により, 対面よりも場所や時間に縛られずに文字・音声によるコミュニケーションができることが確認できた。

v (コミュニケーションの手段):心技体では BBS 機能しかコミュニケーション手段がないが, Google+ハンガアウトでは, ビデオ通話, チャットの利用により, 複数のコミュニケーション手段を選択できることが確認できた。しかし, チャット機能ではコミュニケーションの記録を振り返ることが困難という問題が顕著であった。

vi (コミュニケーションへの気付き):グループトークより, プロジェクトに関わる人とのコミュニケーション手段が確立していることが確認できた。

6.3 適用実験の考察

6.1 項, 6.2 項をふまえ, 4.1 項で検討した coordination communication 支援環境の要件について考察する。

i (作業と同じ場):他の開発作業と同じ場でコミュニケーションを行う環境は必要であるが, 機能そのものや, 機能を使う際の負担が少ない環境が必要だと考えられる。

ii (作業の担当・依存関係):グループ内で誰が何の作業を担当しているのか, 作業がどのような状況か, どの作業に依存関係があるのかが分かりやすい環境は必要であるが, 機能そのものや, 機能を使う際の負担が少ない環境が必要だと考えられる。

iii (作業の中断):作業の中断を最小限に抑え, 物理的制約が少ないコミュニケーションを行いやすい環境は, 有効であると考えられる。

iv (コミュニケーションの時空間制約):空間的制約や時間的制約に縛られずにコミュニケーションを行いやすい環境は有効であると考えられる。また, コミュニケーションを行いやすい環境に加え, 発起人がコミュニケーションを開始した際に, 参加者がコミュニケーションに気付く環境が必要だと考えられる。

v (コミュニケーションの手段):緊急性や重要性を判断する仕組みとして, コミュニケーションの手段を選択できる環境は有効であると考えられる。また, 緊急性や重要性の判断だけでなく, 内容に応じてコミュニケーションの手段を選択することも有効であると考えられる。

vi (コミュニケーションへの気付き):プロジェクトメンバーとのコミュニケーションチャンネルが確立されている環境は有効であると考えられる。

7. 今後の coordination communication 支援環境

本節では, 4.1 項の coordination communication 支援環境の要件, 6 節での適用実験の評価, 考察をふまえ, 今後の coordination communication 支援環境の要件を検討する。

6.3 項の i, ii, iii より, coordination communication 支援ではベネフィットを得るための環境も必要であるが, その環境の有効性を大きくするために, 機能そのものや, 機能を使う際の負担が少ない環境, コミュニケーションを行いやすい環境が必要だと考えられる。つまり, 手軽にコミュニケーションが行える環境が必要だと考えられる。また, iv より, コミュニケーションが開始された際に, そのコミュニケーションに気付く環境が必要だと考えられる。さらに, v より, 緊急性や重要性, コミュニケーションの内容に応じてコミュニケーションの手段や対応を検討できる環境が必要だと考えられる。最後に, vi より, プロジェクトメンバーのような, 自分の開発作業と依存関係のある人とのコミュニケーションチャンネルが確立している環境が必要だと考えられる。また, 自分がどの開発者やどの作業と依存関係があるかが分かる環境も必要だと考えられる。

以上より, 本研究では今後の coordination communication 支援環境の要件は次の 5 項目と考える。以下, この 5 項目について説明する。各項目の末尾に対応する 6.3 項の考察の番号を示す。

- (1) 手軽にコミュニケーションが行える環境:i, ii, iii
- (2) 作業や開発者の依存関係が分かる環境:ii, vi

- (3) コミュニケーションに気付く環境:iv
- (4) コミュニケーションへの対応を検討できる環境:v
- (5) ベネフィットを大きくするための環境:i, ii

(1) 手軽にコミュニケーションが行える環境

コミュニケーションを促進するためには、コミュニケーションを行う際のコストや負担を軽減することが必要だと考えられる。そのため、手軽にコミュニケーションが行える環境が要件として考えられる。

(2) 作業や開発者の依存関係が分かる環境

スムーズなコミュニケーションを行うために、誰と何の内容のコミュニケーションをするかという情報が重要となる。この情報が分かれば、発起人がコミュニケーション相手を選定するまでのコストや負担は軽減できる。そのため、作業や開発者の依存関係が分かる環境が要件として考えられる。

(3) コミュニケーションに気付く環境

コミュニケーションの発起人によってコミュニケーションが行われた際に、コミュニケーションの参加者がそれに気付くことが重要となる。そのため、コミュニケーションに気付く環境が要件として考えられる。

(4) コミュニケーションへの対応を検討できる環境

参加者がコミュニケーションへの対応を考える際に、自身の作業を中断してコミュニケーションに参加するコストと、そのコミュニケーションの重要性や緊急性を考慮し、コミュニケーションへの対応の仕方を判断する必要がある。そのため、コミュニケーションへの対応を検討できる環境が要件として考えられる。

(5) ベネフィットを大きくする環境

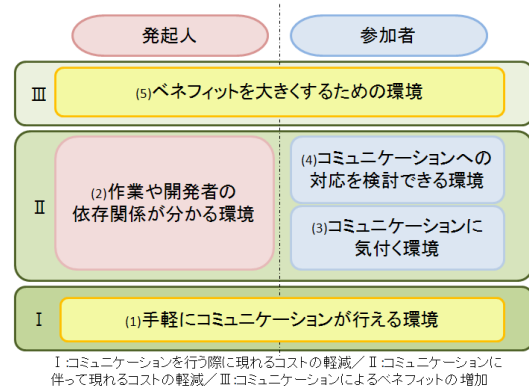
これらの環境をふまえ、コミュニケーションに参加するすべての人にベネフィットをもたらすことが、coordination communication 支援として必要である。そのため、メンバーの状況を把握しやすい環境や議論の場で図や成果物を作成・共有できる環境など、ベネフィットを大きくする環境が要件として考えられる。

以上の環境要件を、コミュニケーションに参加する際の立場も考慮して図示したものを図1に示す。

環境要件の5項目は3つの段階（Ⅰ）コミュニケーションを行う際に現れるコストの軽減、（Ⅱ）コミュニケーションに伴って現れるコストの軽減、（Ⅲ）コミュニケーションによるベネフィットの増加、に分類することができる。

また、この3つの段階には依存関係がある。（Ⅰ）は、コミュニケーションを行うか否かを決定するための最も重要な段階であり、依存関係の基盤となる。この段階には、環境要件（1）が含まれる。（Ⅱ）は、コミュニケーションに伴うコストを軽減するための段階であり、（Ⅰ）の上に成り立つ。この段階には、環境要件（2）、（3）、（4）が含まれる。（Ⅲ）は、コミュニケーションによるベネフィットを増

加するための段階であり、（Ⅰ）、（Ⅱ）の上に成り立つ。この段階には、環境要件（5）が含まれる。さらに、この3つの段階は、（Ⅰ）、（Ⅱ）がコスト、（Ⅲ）がベネフィットに関わるため、coordination communication の発起人と参加者のコストとベネフィットが対称となることも示している。



Ⅰ コミュニケーションを行う際に現れるコストの軽減 / Ⅱ コミュニケーションに伴って現れるコストの軽減 / Ⅲ コミュニケーションによるベネフィットの増加

図1 今後の coordination communication 支援環境の要件モデル

8. おわりに

本研究では、協調ソフトウェア開発での coordination communication を支援することを目的として、Google+ハンガアウトを用いた適用実験、今後の coordination communication 支援環境の要件モデルの検討を行った。

しかし、今回検討した coordination communication 支援環境の要件モデルの5項目はまだ検討の余地がある。3つの段階をふまえた上で、環境要件の項目の検討を続けていくことを今後の課題とする。

参考文献

- 1) Nakakoji K., Ye Y. and Yamamoto Y.: “Comparison of Coordination Communication and Expertise Communication in Software Development: Motives, Characteristics, and Needs,” New Frontiers in Artificial Intelligence, Lecture Notes in Computer Science, Volume 6284, pp.147-155, 2010.
- 2) Nakakoji K., Ye Y., and Yamamoto Y.: “Supporting Expertise Communication in Developer-Centered Collaborative Software Development Environments,” Collaborative Software Engineering, Springer-Verlag, 2010.
- 3) 中小路久美代, イェユンウエン, 山本恭裕: “ソフトウェア開発における知識コミュニケーションのためのインタラクションデザイン,” The 23rd Annual Conference of the Japanese Society for Artificial Intelligence, pp.1-4, 2009.
- 4) Google, Google+ハンガアウト, <http://www.google.com/+learnmore/hangouts/>, accessed 2013/7/20.
- 5) 三浦真人, 小林祐介, 島田和幸, 高橋晃一, 樫山淳雄: “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,” 電子情報通信学会, OIS2007-4, pp. 19-24, 2007.