SIG-KSN-006-02
人工知能学会
第六回知識流通ネットワーク研究会

Wiki を導入したソフトウェア開発コミュニケーションの分析

神戸雅一[†,††]　山本修一郎[†]　　太田敏澄[††]

[†]株式会社 NTT データ　技術開発本部　システム科学研究所
東京都江東区豊洲 3-3-9 豊洲センタービルアネックス

[††]電気通信大学大学院情報システム学研究科
東京都調布市調布ヶ丘 1-5-1

Analysis of Wiki Based Software Development Communication

Masakazu Kanbe[†,††], Shuichiro Yamamoto[†] and Toshizumi Ohta[††]

[†]Research Institute of System Science, Research and Development Headquarters, NTT DATA CORPORATION
Toyosu Center Building Annex 3-3-9 Toyosu Kotoku Tokyo Japan

[††]Graduate Department of Social Intelligence and Informatics, Graduate School of Information Systems,
The University of Electro-Communications,
1-5-1 Choufugaoka, Choufushi, Tokyo Japan

概要
ソフトウェア開発者間のコミュニケーションを分析するために TIE モデルを提案した．本稿では，TIE モデルを用い，Wiki を導入したソフトウェア開発のコミュニケーションを分析し，TIE モデルに基づき，ソフトウェア開発の効果と限界について考察する．

Abstract
We propose TIE model to analyze the communication among software developers. In this article, we used the TIE model to analyze real soft ware development communication case. And we also discuss effectiveness and limitation of the software development based on TIE model.

1. はじめに

ソフトウェア開発は，複数のソフトウェア開発者が関わる複雑なプロセスである．そのプロセスのコミュニケーションの形態にはさまざまなものがある．定期的な対面のミーティングやアドホックなディスカッション，電子メールによるドキュメントの交換は伝統的なコミュニケーション方法であると言える．近年はこれらに加え，Wiki や SNS，ブログ，統合開発環境(Integrated Development Environment: IDE)に組み込まれるコミュニケーションプラグインなどが，開発者間の知識流通をサポートしている．

本稿では，ソフトウェア開発コミュニケーションの知識ネットワークモデルとして TIE モデルを提案し，ソフトウェア開発者のコミュニケーションを分析する．TIE モデルは，暗黙知ネットワーク(Tacit knowledge network: TKN)，仲介知ネットワーク(Intermediary knowledge network: IKN)，形式知ネットワーク(Explicit knowledge network: EKN)の 3 階層からなる． 我々は TIE モデルを用いて，Wiki を導入したソフトウェア開発プロセスを分析し，その効果について考察した．本稿では 2 節で関連研究を紹介し，3 節で TIE モデルを説明し，4 節でケーススタディを行う．5 節で TIE モデルに基づいた Wiki を導入したソフトウェア開発の効果と限界について考察し，6 節でまとめる．

2. 関連研究

2.1 仲介知

著者らは，企業内の知識流通モデルとして，仲介知モデルを提案している[1][2][3]．仲介知は，CMC (Computer Mediated Communication)ツールを通じて行われる社員間の知識流通が行われる知識状態を指す．仲介知モデルは，SECI モデル[4]に代表される伝統的な暗黙知と形式知の概念に基づく知識創造モデルを拡張したものである．

伝統的な知識創造モデルでは，暗黙知と形式知のあいだの共同化，表出化，連結化，内面化の 4 つの知識変換モードを提案している．仲介知モデルは組織の知識変換プロセスを必要としない課題解決と業務遂行を提案したモデルである．社員は，知識を仲介知の状態で CMC ツールを用いて流通させる．仲介知モデルを用いることで，暗黙知として交換できない知識を形式知化するよりも低いコストで共有する知識流通を説明することができる．

仲介知モデルでは，社員は CMC ツール上で仲介知の知識変換モードを活用して，高速に知識を活用することが説明できる．仲介知の知識変換モードは，公開化，断片化，協同化，共鳴化，洗練化である．公開化は個人の経験やアイデアを伝えることである．断片化は形式知の一部を断片的に導入することである．協同化は社員の意見やアイデアに反応すること

である．共鳴化は他者の意見を理解し，同意することである．洗練化は仲介知の内容をまとめて形式知を作り出すことである．

伝統的な知識モデルの知識スパイラル条件に従うと，社員が公式に組織間で知識を活用する場合に，それぞれの組織ごとに，組織内の知識スパイラルを経て形式知を作成する必要がある．組織内の知識スパイラルを経て公的な形式知を作るにはコストと労力がかかる．仲介知の知識変換モードでは，知識スパイラル条件を必要としないので，伝統的な知識変換モードに比べ，低いコストと労力で知識交換ができることを説明できる．また，仲介知モデルではコミュニケーションの記録の有効性も説明することができる．CMC ツールの利用で，社員は多くのコミュニケーションの機会を獲得し仲介知を交換する．CMC ツールで行われたコミュニケーションは仲介知として記録され，仲介知は，社員に効果的に再利用される．

2.2 IBIS

IBIS(Issue-Based Information System)[5] や gIBIS (graphical IBIS)[6]は, 伝統的なソフトウェア開発手法である．IBIS の目的のひとつは，ソフトウェア開発の議論のプロセスや進捗状況を完全に記録し，構造化することである．すべての議論のプロセスと進捗状況を記録し構造化することで，ソフトウェア開発者は開発に必要な情報を見つけることが IBIS モデルの効果とされている．しかし，IBIS の効果が有効であったとしても，完全に文書化された記録を作成したり再利用したりするコストと労力は大きい.

2.3 近年のソフトウェア開発研究

近年のソフトウェア開発研究のなかには，ソフトウェア開発者のコミュニケーションをサポートすることに焦点をおくものがある．ソフトウェア開発者は開発の中で，他の開発者とコミュニケーションする. Ko ら[7]は複数のソフトウェア開発者の活動を分析し，ソフトウェア開発者が他の協業者を情報ソースとして利用していることを明らかにした．近年のソフトウェア開発においては，開発者間のコミュニケーションが重要であることが示されている.

また Ye ら[8][9]は，ソフトウェア開発者が，ソフトウェアライブラリや開発チームのメンバーから開発に必要な知識を探索することをサポートするための手法を研究している．彼らは，API ドキュメント用のパーソナライズされた検索エンジンとソフトウェア開発チームのエキスパートを検索するためのコミュニケーションチャネルを提案している．効果的なソフトウェア開発のための開発者間でのコミュニケーション方法を支援するものである.

Marczak ら[10]は，要求変更管理のプロセスにおける情報ブローカーの重要性を示している．彼らの研究では，情報ブローカーがソフトウェア開発チームの社会ネットワークにおいて重要な役割であることが明らかになっている．情報ブローカーは，要求仕様の誤解が生じないよう情報の流れを円滑化していた．これらの研究はいずれも，ソフトウェア開発における開発者間のコミュニケーションの重要性を示唆している.

3. TIE モデル

3.1 TIE モデルの概要

我々は，ソフトウェア開発者の動的なコミュニケーションを分析するモデルとして TIE モデルを提案する．TIE モデルは，TKN，IKN，EKN の 3 階層からなるモデルである．表 1 は TIE モデルの各階層の特徴を示す.

表 1：TIE モデルの階層の特徴

知識ネットワーク	ネットワークノード	メディア	Documentation	成果物の例
暗黙知ネットワーク (TKN)	開発者	対面ミーティング、電話、テレビ会議	No Documentation	議論、会議
仲介知ネットワーク (IKN)	CMC コンテンツ	Wiki、SNS、ブログ、E-mail	Just in time Documentation	CMCツールのログ
形式知ネットワーク (EKN)	ドキュメント	文書管理サービス	Full Documentation	要件定義書、仕様書、ソースコード、マニュアル、ガイドライン

TKN は暗黙知を交換するネットワークである．TKN のネットワークノードは開発者であり，組織構造やメンバーの役割，意志決定プロセスなどから多くの影響を受ける．TKN は，対面ミーティング，電話, テレビ会議などのメディアを通じて実現される．TKN では文書化作業は行われない場合も多い．よって TKN では公式なドキュメントの作成は行われないと仮定する．TKN の成果物は，議論やミーティングといった行為であり，観察可能な有形の成果物は作成されない．TKN のネットワークエッジは，開発者間の口頭での会話を意味する.

IKN は仲介知を交換するネットワークである. IKN のネットワークノードは, CMC のコンテンツであり, CMC ツール上でのソフトウェア開発者のコミュニケーションネットワークが実現される. IKN は, Wiki やブログ，SNS という CMC ツール内でのコミュニケーションを通じ成長していく. 電子メールも CMC ツールであるが，Wiki とは異なり宛先を明確にする点と返信を含むメールの送信ごとに新たなコンテンツが生成されるという特徴がある.

IKN を通じてソフトウェア開発者は，Just in Time Documentation のメリットを享受することができる．ある開発者が他者との調整が必要であった場合，その開発者は CMC ツールを使って他者と調整を行うことができる．その調整のコミュニケーション内容は記録され，他のメンバーに向けて公開される．この公開された調整の履歴はソフトウェア開発における有効なドキュメントなる．また調整以外の伝達事項も，CMC ツールを利用して伝達し，その履歴を残すこともできる．このプロセスを，Just in Time Documentation と呼ぶ．Just in Time Documentation では, CMC ツールを用い開発者同士で議論した内容が必要な知識として記録される．IKN の成果物はこの Just in Time Documentation により記録された CMC のログである．IKN のネットワークエッジは CMC コンテンツの繋がりを意味する.

EKN は形式知を交換するネットワークである．EKN はソフトウェア開発におけるドキュメントのネットワークを表す．EKN のノードはドキュメントであり，文書管理システム内で成長する．文書管理システムには EKN を成長させるための，文書更新履

歴管理，版管理，全文検索，文書共有などの機能がある．EKN は開発者に文書化機能を提供する．EKN の成果物は，用件定義書，仕様書，ソースコード，マニュアル，ガイドラインなどの文書である．EKN のネットワークエッジは，文書間の相互関係性を意味する．TKN と IKN のネットワークエッジは，CMC ツールを用いた個人による仲介知の提供と獲得プロセスを意味する．IKN と TKN のネットワークエッジは，CMC ツールを用いた形式知の文書化と引用を意味する．

3.2 TIE モデルとソフトウェア開発

TKN では公式な文書は作成されない．この TKN の特徴を No Documentation と呼ぶ．一方の EKN は緻密に文書を作成することを主な目的とする．この EKN の特徴を Full Documentation と呼ぶ．これまでのソフトウェア開発では，TKN と EKN でのプロセスを主な知識プロセスとして利用していた．

しかし，こうしたソフトウェア開発の知識プロセスでは 2 つの問題がある．ひとつ目の問題は，ソフトウェア開発における重要な情報の喪失である．TKN の知識プロセスはミーティングや開発者の机の周辺のコミュニケーションで行われる．TKN でのコミュニケーションの記録の多くは，ミーティングや議論が終わると消えてしまう．TKN のコミュニケーションによるコンテンツは，必ずしも文書化はされないし，ソフトウェア開発チームの全メンバーで完全に共有されることはない．

ふたつ目の問題は，ソフトウェア開発におけるすべての重要な情報を記録することが困難なことである．ソフトウェア開発のすべてのイベントが完全に即時に文書化されることができるならば，個々のソフトウェア開発者が要求，仕様，ソースコードなどを完全に理解することができる状態にあると言える．しかし，完全な文書化を実現することは，コストとその範囲が非常に大きいことからも難しい．また，完全に文書化されたものを開発者が十分に参照できる環境の構築も課題である．

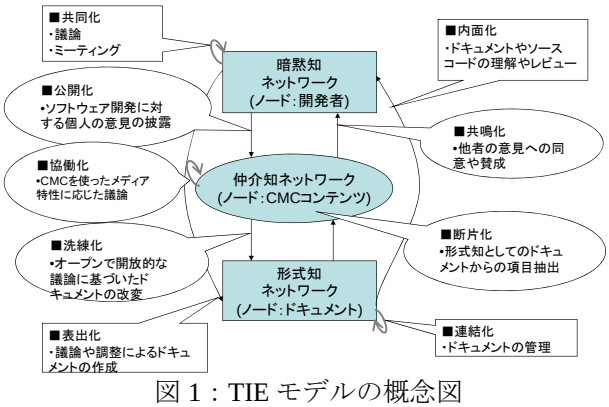

図 1：TIE モデルの概念図

こうした問題を解決するために，我々は TIE モデルに基づいたソフトウェア開発プロセスを提案する．図 1 に TIE モデルの概念を示す．TIE モデルは，これまでで伝統的なソフトウェア開発コミュニケーションに IKN を追加したものである．CMC ツールが IKN を実現する．図 1 中のバルーンは，TIE モデルにおける代表的な知識プロセスを示す．四角のバルーンは伝統的なソフトウェア開発の知識プロセスを示す．一方の丸いバルーンは TIE モデルに特化した知識プロセスを示す．

IKN の知識プロセスはオープンで迅速な CMC ツール上のコミュニケーションである．代表的な CMC ツール，表 1 に示したように，電子メール，Wiki，SNS などがあり，それぞれ固有の特性を持っている．電子メールは非同期型のコミュニケーションを実現している．Wiki のような CMC ツールは，ソフトウェア開発チームのオープンなコミュニケーションを促進する．IKN はこのような CMC ツールのログとして記録され，CMC の履歴は公的な文書ではないがソフトウェア開発にとっては非常に有益な情報である．ソフトウェア開発チームのメンバーはお互いの知識プロセスを，CMC ツールを通じて理解できる．TIE モデルの知識プロセスは，仲介知の知識変換モードと密接に関係する．暗黙知，仲介知，形式知はそれぞれ，TKN，IKN，EKN と密接に関係している．

4. ソフトウェア開発ケーススタディ

TIE モデルの有効性を評価するために，Wiki を導入したソフトウェア開発ケースの分析を行った．このケースにおいて，Wiki はソフトウェア開発のコミュニケーションを促進するために導入された．

4.1 ケーススタディの目的

ケーススタディは以下の仮説を検証するために実施した．

仮説 1：CMC ツールはソフトウェア開発のために重要な情報を記録することができる
仮説 2：CMC ツールは従来のソフトウェア開発スタイルでは効果的に共有できなかった知識を共有することができる
仮説 3：CMC ツールは適切なコミュニケーションの機会をソフトウェア開発者に提供することができる

これらの仮説を，ソフトウェア開発における CMC ツールの効果を確認するために設定した．CMC ツールが IKN をサポートするため，この CMC ツールについての仮説を検証することで TIE モデルの効果を検証することとした．

4.2 ケースの概要

ケースには Wiki を導入したソフトウェア開発を選択した．このケースのソフトウェア開発者数は 9 名であり，2 つの異なる企業に所属していた．彼らは協働し，特殊なセンサーを持つハードウェアデバイスを用いたソフトウェアシステム開発を行った．このソフトウェア開発は，ドキュメントの作成，プログラムのコーディング，プログラムの試験というプロセスを含んでいた．メンバーが所属する 2 つの企業は，時差はないが異なるロケーションにあった．原則として全メンバーが参加する週に一度の定例ミーティングを持っていた．定例ミーティング以外のコミュニケーション方法は，開発者間のアドホック・コミュニケーション，電話，電子メール，文書管理システムによるファイル共有があった．

電話のコミュニケーションは電話をする開発者しか議論に参加できなかった．また電子メールも利用されていたが，メーリングリストに送信されたメー

ルの読み落としや共有すべきメンバーに対しメールが送信先になっていないなどのコミュニケーションミスも発生していた．こうしたコミュニケーションミスが，ソフトウェア開発に悪影響を与えることがあったため，チーム内のコミュニケーションを補完する目的で Wiki を導入した．

表 2 にこのケースのコミュニケーションメディアとその特徴を示す．

表 2：利用されたコミュニケーションメディア

コミュニケーションメディア	TIEモデルのネットワーク	特徴
定例ミーティング	TKN	週に一度の公式対面ミーティング
アドホック・コミュニケーション	TKN	開発者間が自席付近で行うローカルなコミュニケーション
電話	TKN	二拠点間での1対1の通話
電子メール	IKN	メーリングリスト、担当者同士でのコミュニケーション
Wiki	IKN	新たに導入されたオープンなCMCツール
文書管理システム	EKN	作成したドキュメントを管理するファイル共有機能

また付録に Wiki コミュニケーションの概要を示す．この Wiki は最終的には 13 ページ，21 項目のコンテンツが記録された．

この Wiki で観察された TIE モデルの知識プロセスは，18 個の公開化，7 個の断片化，2 個の協働化であった．公開化の例は，付録の W3 の項目である．W3 では, あるメンバーがこのソフトウェア開発に必要な課題を想像し，課題を公開化した例である．断片化の例は付録の W12 の項目である．W12 では，メンバーが組織内のソフトウェア開発会議規定から，会議に必要な文書のリストを抽出し Wiki に記述したものであった．協働化の例は W8 の項目である．W8 では, メンバーがテストのポリシーについて Wiki を使って議論している．一方で，共鳴化と洗練化についての知識プロセスは Wiki 上では確認することができなかった．これら 2 種類のプロセスは，電子メール，対面のミーティング，文書管理システムの操作などの Wiki 以外の活動で実行されていたであろうことが推測される．

4.3 仮説の検証

仮説を検証するために，Wiki のなかの CMC から論拠となる事例を抽出した．

仮説 1 の検証

この仮説は CMC の記録性に関するものである．Wiki はソフトウェア開発の重要な情報を記録した．9 人のメンバーが Wiki を利用し，ソフトウェア開発に必要な 21 の重要な項目を記録した．付録の W1 と W2 にある設計ドキュメント一覧という項目がある．これらのリストはドキュメント一覧という形式知から断片化された仲介知である．一人のメンバーがソフトウェア開発チームには，このドキュメントリストを共有することが重要であると考え，ドキュメント一覧から必要なドキュメントの名称をリストとして抽出した．

このメンバーはドキュメントリストの共有に電子メールを使用せずに Wiki を利用した．そしてほかのメンバーは，Wiki 上のドキュメントリストにそのドキュメントの作成作業の進捗状況を記載していった．この進捗状況の記載は，メンバーによる仲介知の公開化である．すべてのメンバーで Wiki を使い，日々のドキュメントの進捗状況を共有していた．Wiki はすべての開発メンバーに対してオープンであり，ひとつのオブジェクトをすべてのメンバーで共有することができた．仮にこの開発チームが進捗状況を電子メールで共有していたとすると，電子メールによる行き違いなどが理由で，進捗状況の継続的な更新と共有は行われなかった可能性がある．

仮説 2 の検証

この仮説は効率的な知識の共有に関するものである．このケースでは，Wiki の導入により従来のソフトウェア開発スタイルでは効果的に共有できなかった知識の共有を促進したことを確認した．付録の W15 のノウハウの項目は，この仮説に対する論拠であるといえる．W15 のノウハウは，かつて類似したハードウェアデバイスに関する開発経験のあるメンバーにより書き込まれたものである．このメンバーは丁寧に特殊なハードウェアを制御するための方法についての知識を Wiki に公開した．この知識はこのメンバーの個人的な経験に基づいたものであり，この時点では形式知化されたものではなかった．従来のソフトウェア開発では，こうしたノウハウは，口頭で会話に参加しているメンバーに限り，TKN のコミュニケーションで伝えられる知識であった．このケースでは，Wiki での知識共有によって，特殊なハードウェアを扱う知識をメンバーが実証により再発見するコストを抑えたことが観察できた．

仮説 3 の検証

この仮説はメンバー間の効果的なコミュニケーションに関するものである．Wiki の導入により，ソフトウェア開発チームが，適切なコミュニケーションの機会を得ることができたことを意味する．その根拠は付録の W7 と W8 にある．テスト項目の考え方という W7 の項目をあるメンバーが Wiki に公開化した．それに他のロケーションにいる別のメンバーが W7 に対するコメントとして W8 の項目を記述した．従来のソフトウェア開発では，このメンバー間のコミュニケーションは週に一回開催される定例ミーティングまで保留されていた可能性がある．このケースでは，Wiki の導入により，適切なコミュニケーションの機会が設定されソフトウェア開発の遅延要因である意見交換の留保が解消されたことが観察された．

5. 考察

本項では記録性，効率的な知識共有，メンバー間の効果的なコミュニケーションという観点で Wiki を導入したソフトウェア開発の有効性について考察する．さらにソフトウェア開発コミュニケーションのための TIE モデルに基づいた分析の限界についても議論する．

5.1 記録性

ソフトウェア開発者が重要な情報を CMC ツールに記録する条件について議論する．我々は彼らが重

要な情報を Wiki に記録した 2 つの理由を仮定した．そのひとつは，共有されるべき情報が広くメンバーに公開される必要があったという条件である．Wiki はソフトウェア開発者にとってオープンなコミュニケーション環境を常時提供していた．Wiki を利用しないソフトウェア開発では，彼らは対面のミーティング，アドホック・コミュニケーション，電話，電子メール，文書管理システムを用いてでコミュニケーションを実施していただろう．Wiki はこれらのコミュニケーション手段に比べオープンで動的なものであった．この Wiki のオープンで動的な特徴が，開発者に自らの知識を書き込ませることとなった．この特性により，気楽なコミュニケーションが行われ，開発者は自らの進捗状況を相互に公開しあうこととなったと考えられる．対面ミーティングでは，権力をもったメンバーが他の権力の弱いメンバーの発言を阻害することがあったかもしれない．Wiki は権力の弱いメンバーの発言を促進したことが考えられる．

ふたつ目の条件は，Wiki のコンテンツは開発者チームにとって単一のオブジェクトであったことである．開発者はそれぞれのドキュメントの進捗状況を Wiki 上のドキュメントリストに毎日書き込んでいった．Wiki を利用しなかった場合では，メンバーはドキュメントの進捗状況を毎日電子メールで交換したと推測される．電子メールですべてのメンバーの進捗状況を管理することは，Wiki の場合よりも労力がかかることが予測される．それは，電子メールはメールの送信または返信ごとに新たなオブジェクトが生成されるため，調整のために複数のオブジェクトを管理しなければならないからである．この電子メールコミュニケーションの特徴により，開発者間の知識は分散してしまう可能性がある．メンバーの中の一人が，こうした分散したオブジェクトに存在する知識を統合して管理することは，Wiki による単一のオブジェクト管理に比べ効率的ではない．またすべてのメンバーのメールを読まない限り，すべてのメンバーの進捗状況を把握することができない．共有される知識がオープンであり，単一のオブジェクトとして編集されたほうが，都合がよい場合には，ソフトウェア開発者は Wiki のような CMC ツールに重要な情報を記録すると考えられる．

5.2 効率的な知識共有

ソフトウェア開発者が効率的に知識共有をする条件について考察する．我々は，Wiki の活用は，開発者個人の経験の共有に適していると推測している．特に今回のケースのように，特殊なハードウェアの利用など試行錯誤を要する開発経験の共有に有効であると考える．こうした経験等の知識は体系化されていない場合も多く，開発者たちはこうした知識を TKN のコミュニケーションで共有することが多い．

Wiki のような CMC ツールは，こうした知識をすべてのメンバーに効果的に浸透させる可能性がある．もしあるメンバーが，新たなデバイスを利用するための知識を 30 分間かけて Wiki に公開し，他の 8 人のメンバーがそれぞれ 5 分間かけてその知識を読んだとすると，知識の獲得にかかった時間は 70 分である．一方，そのメンバーの知識を活用できずに，残り 8 人のメンバーが試行錯誤の末 120 分かけてこの知識を獲得した場合には，960 分の時間がかかる．これは極端な試算の例であるが，CMC ツールによる知識の共有は効率的である．我々は試行錯誤を通じて得られる知識の共有について Wiki のようなオープンな CMC ツールの活用は有効である．

5.3 メンバー間の効果的なコミュニケーション

メンバー間の効果的なコミュニケーションが起こる条件について考察する．このケースにおけるメンバー間でのコミュニケーションメディアと，その特徴は表 2 に示した．定例ミーティング，アドホック・コミュニケーション，電話，電子メール，文書管理システムに基づいたコミュニケーションでは不十分であったため，このソフトウェア開発チームは Wiki の導入を決定した．仲介知の協働化や断片化に続く公開化が観察されたことからも，Wiki の導入により新たなコミュニケーション機会が発生したと言える．Wiki で行われるコミュニケーションは，オープンであり，定例ミーティングで行われるコミュニケーションの一部を代替していた．これにより定例ミーティングまでの意見交換の留保が解消されたと言える．しかし，Wiki の導入により生じる意見交換の留保解消にはある条件が必要である．それは Wiki で獲得される知識が，対面のミーティングにより獲得される知識と同種のものであるという条件である．我々は実際のソフトウェア開発の現場では，Wiki のような CMC ツールで共有できる知識と対面ミーティングで共有できるタイプの知識があることを理解している．我々は，こうした知識を分類し，知識コミュニケーションの条件について検証を行い Wiki のような CMC ツールを十分に活用する方法を検討する必要がある．表 2 に示したソフトウェア開発者のコミュニケーションメディアを効果的に組み合わせる方法について検討する必要がある．

5.4 限界

ケースの分析により，CMC ツールがソフトウェア開発における重要な情報を記録すること，ある一定の条件においてソフトウェア開発プロセスを効率的にし，効果的なコミュニケーションを促進することを議論した．CMC ツールは，ソフトウェア開発チームが適切に使いこなせばソフトウェア開発を促進することが明らかになった．しかし本稿では，CMC ツールがソフトウェア開発に有効に作用した少数の事例を抽出しただけである．CMC ツールがソフトウェア開発を促進する条件についてより詳細に調査する必要がある．また，Wiki に記載された誤った情報やソフトウェア開発者たちの情報過多など，Wiki がソフトウェア開発に与えるネガティヴな効果についても分析する必要がある．さらに，TIE モデルのノードである，人，CMC コンテンツ，ドキュメントの関係性についても明らかにする必要がある．

6.まとめ

ソフトウェア開発者間の知識コミュニケーションを促進することは重要である．我々は仲介知モデルとソフトウェア開発分析のために TIE モデルを提案した．既存のソフトウェア開発研究では，エキスパートとしての個人の特性やドキュメントやチームの

プロセスに焦点が置かれていた．TIE モデルはソフトウェア開発のコミュニケーションモデルであり，Just in Time Documentation を説明するモデルである．我々は，Wiki を導入したソフトウェア開発プロセスを分析し，ソフトウェア開発における Wiki 活用の効果を明らかにした．ケースを通じ，Wiki のようなオープンで動的な特性を持つ CMC ツールがソフトウェア開発プロセスを促進する条件について議論した．今後の研究では，TIE モデルを用いて CMC を活用したソフトウェア開発ケースを分析する．この際に，CMC ツール以外での開発者の活動や，電子メールや Wiki といった異なる CMC ツールの使い分けに着目する予定である．また，コミュニケーションとドキュメントの作成やコーディングなどのコミュニケーション以外のソフトウェア開発に重要な業務の関係も調査する予定である．

参考文献

1. 山本修一郎, 神戸雅一(2008), 企業内 SNS による知識創造，人工知能学会第二回知識流通ネットワーク研究会, http://www4.atpages.jp/sigksn/conf02/SIG-KSN-002-03.pdf
2. Kanbe, M. and Yamamoto, S. : An Analysis of Computer Mediated Communication Patterns. The International Journal of Knowledge, Culture and Change Management. Vol.9 No.3 35--47 (2009)
3.神戸雅一, 山本修一郎(2009), 企業内デジタルコミュニケーションの分析，第 15 回社会情報システム学シンポジウム学術講演論文集，pp.25-30
4. 野中郁次郎，竹内弘高(著)，梅本勝博 (訳) (1996),知識創造企業，東洋経済新報社.
5. Rittel, H. and Kunz, W. : Issues as elements of information systems., Working paper# 1 31. Institute fur Grundlagen der Planung I.A.University of Stuttgart.
6. Conklin, J. and Begeman, M.L. : gIBIS: A Hypertext Tool for Exploratory Policy Discussion., ACM Transactions on Office Information Systems, 4, 6, pp. 303--331 (1988)
7. Ko, A.J. , DeLine, R. and Venoloa, G. : Information Needs in Collocated Software development teams., 29th International Conference on Software Engineering (ICSE'07) (2007).
8. Ye, Y., Yamamoto, Y. and Nakakoji, K. : Expanding the Knowing Capability of Software Developers through Knowledge Collaboration., International Journal of Technology, Policy and Management Vol. 8, No. 1, pp. 41--58 (2008)
9. Ye, Y., Yamamoto, Y., Nakakoji, K., Nishinaka, Y. and Asada, M. : Searching the Library and Asking the Peers: Learning to Use Java APIs on Demand., in V. Amaral, L. Veigaet al (eds.): Proceedings of 2007 International Conference on Principles and Practices of Programming in Java, ACM Press: Lisbon, Portugal, pp. 41--50 (2007).
10. Marczak, S., Damian, D., Stege, U. and Schroter, A. : Information Brokers in Requirement-Dependency Social Networks., 16th IEEE International Requirement Engineering Conference, pp. 53--62 (2008)

付録：ソフトウェア開発に用いられたWiki上のCMCコンテンツ

ページ名	項番	記載項目	内容	知識変換モード
基本設計	W1	・基本設計ドキュメントリスト	・完成したドキュメントに〇印がつく. 更新日、コメントを記載	断片化、公開化
詳細設計	W2	・詳細設計ドキュメントリスト	・完成したドキュメントに〇印がつく. 更新日、コメントを記載	断片化、公開化
	W3	・課題事項	・この工程での課題が記述される.	公開化
製造/単体試験	W4	・連絡事項	・議事録提示による決定事項の提示	公開化
	W5	・試験項目の考え方	・試験項目数と収束に関する基本姿勢の記述	公開化
	W6	・試験密度	・試験項目数と開発規模から算出する試験密度の計算	公開化
結合試験	W7	・試験項目の考え方	・試験項目の抽出条件と網羅性に対する基本姿勢の記述	公開化
	W8	・考え方へのコメント	・上記基本姿勢へのコメント	協働化
	W9	・試験密度	・試験項目数と開発規模から算出する試験密度の計算	公開化
	W10	・試験に必要な機材一覧表	・ソフト，ハードからなる機材一覧表	公開化
総合試験	W11	・総合試験確認のコメント	・総合試験実施要綱の必要性を喚起するメッセージ	公開化
開発会議1	W12	・開発会議1資料	・開発会議1のために作成する資料の一覧表	断片化
開発会議2	W13	・開発会議1資料	・開発会議1のために作成する資料の一覧表	断片化
グラフ	W14	・仕様策定	・開発システムが出力するグラフの仕様の説明	公開化
ノウハウ	W15	・ノウハウ	・類似したシステム開発経験者によるノウハウ	公開化
プロジェクト管理メモ	W16	・プロジェクト管理上の要改善項目	・プロジェクト実施のためのコミュニケーションの留意事項	公開化
デモンストレーション検討	W17	・目的	・デモンストレーションの実施目的を記述	公開化
	W18	・シナリオ	・オフィス，製造現場におけるユースケースの記述	公開化
	W19	・作業内容	・デモンストレーション開発のフェーズ分割と実施項目の記述	公開化，断片化，協働化
成果物名称	W20	・章立て	・システムの説明書の章立てと修正コメント	断片化，公開化
	W21	・成果物	•成果物及び成果物作成のための作業メモ	断片化，公開化