

知識共創のためのダイナミックコミュニティ: 理論・アーキテクチャ・応用

葉 雲文^{1,2} 山本 恭裕³ 岸田 孝一¹

¹(株)SRA先端技術研究所

² Department of Computer Science, University of Colorado, Boulder

³ 東京大学先端科学技術研究センター

yunwen@cs.colorado.edu yxy@computer.org k2@sra.co.jp

要旨

知識共創を促進するためのダイナミックコミュニティの理論およびアーキテクチャを提案する。従来の知識共創アプローチには、形式知に着目した知識管理/ナレッジマネジメントによる知識共創と、暗黙知に注目したコミュニティを通じた知識共創という二種類がある。ダイナミックコミュニティは、両方を統合した新しいアプローチに基づくもので、個々の情報ワーカーの時々刻々と変化する情報ワークタスクを中心としてアドホックかつオンデマンドに構成される知識ネットワークである。本論文では、ダイナミックコミュニティの理論とその形成を支援するソシオテクニカル基盤環境のアーキテクチャを解説し、ダイナミックコミュニティ理論のソフトウェア開発への応用について論じる。

1. はじめに

知識創造支援の必要性は高まる一方であるが、そのためには、パーソナルな知的活動としての知識創出という側面だけでなく、人々の知的活動の社会的な性質に目を向けた知識共創という側面が考慮されなければならない。知識創造のプロセスとしてソフトウェア開発を捉えるならば、ソフトウェア開発企業は、ソフトウェア開発者の個々人が持つ知識の共有や結合による知識共創を通して新たな価値を創造しているといえる。

知識共創を支援するメカニズムに関する既存の研究は、大きく二つのアプローチに分類することができる。一つは、個々の知識ワーカーが持っている知識を外在化し、リポジトリに蓄積した後に活用するアプローチ[2]である。もう一方は、同じドメインの知識ワーカーを一つのコミュニティオブプラクティス(Community of Practice)[10]と見なして、たとえばエキスパートリストを作成するなどすることにより、コミュニティメンバーのコミュニケーションを支援し知識交換の円滑化を図るアプローチである。

前者の知識リポジリアプローチは、初期の人工知能研

究の影響を受けたものである。人間の知識を論理やルールとして抽出することによって、専門家から聞き出した知識を知識ベースやルールベースなどに蓄積し、他のユーザに利用してもらうという仕組みである。しかし、様々な試行錯誤の結果として分かったのは、人間の知識のうち表出可能なものは極一部であり、表出できないコンテキストなどの暗黙知[9]が存在するということである。たとえ知識ベースから関連知識を見つけることができたとしても、元々のコンテキストが失われているために、うまく活用できない場合が多く見受けられた[8]。

この問題を解決するものとして、1990年代に入ってから、ウェンガー等がコミュニティによる知識共有と知識共創とを提唱し始めた[10]。知識リポジトリでの情報中心のアプローチとは異なり、コミュニティによる人中心のアプローチである。知識を直接的にデータベースに格納し管理するのではなく、人と人との自然な交流を通して知識を移転、創造、共有することを狙いとしている。換言すれば、80年代の know-how の管理から know-who の管理への移行である。この流れの研究においては、人間の社会的な関係を知識と同様に重要な資源であると考え、情報管理のためのデータベースシステムやファイルシステムなどを参考に、人間の社会的なネットワークを管理するツール ContactMap [7]などが開発されている。

上に述べた二つのアプローチは、知識提供とコミュニティ支援のいずれか一方に注目し、知識というものあるいはコミュニティというものを固定的かつ一元的な関係として扱っている。そこでは、知識を活用する作業現場におけるユーザの個別の情報ニーズや、知識活用のコンテキストといった側面は重視されていない。本稿で提案するアプローチは、知識を活用する作業現場に動的に生まれる個別の情報ニーズとそれを活用するユーザとを中心として、知識/情報とコミュニティの支援を同時に提供するダイナミックコミュニティという新たなコンセプトに基づくものである。

ダイナミックコミュニティとは、個々の知的作業者の時々刻々と変化する情報ワークタスクを中心にアドホックかつオンデマンドに構成される知識ネットワークである。以下では、ダイナミックコミュニティの理論とその特徴と形成プロセスの点から解説し、その形成を支援するためのソシオテクニカル基盤環境のアーキテクチャについて述べ、ダイナミックコミュニティ理論のソフトウェア開発への応用について論じる。

2. ダイナミックコミュニティ

ダイナミックコミュニティ(Dynamic Community; DynC)は、あるナレッジワークスペース(Knowledge Work Space; KWS)の中に動的に構成され存在するサブグループである。ナレッジワークスペースは、複数の知識ワーカーと彼らが有する知識から成る。そこでは、情報と情報の関係、人と情報の関係、人と人の関係、という三種類の関係が存在する。なお、これらの関係の定義は、具体的な応用ドメインに応じて、実装レベルでは異なるものとなる。

図1にナレッジワークスペース(KWS)の例を示す。図中では、人の集まりが $\Psi = \{A, B, C, D, E, M, N, O, P, Q\}$ 、各情報が $\Phi = \{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta\}$ となっている。三種類の関係が示されており、情報と情報の関係(点線で表す) $II = \{(\alpha, \beta), (\alpha, \gamma), (\beta, \epsilon), (\beta, \delta), (\gamma, \theta), (\zeta, \eta)\}$ 、情報と人との関係(破線で表す) $PI = \{(B, \alpha), (C, \alpha), (M, \alpha), (B, \beta), (D, \beta), (E, \gamma), (N, \gamma), (D, \delta), (P, \epsilon), (O, \zeta), (Q, \eta), (Q, \theta)\}$ 、人と人の関係(実線で表す) $PP = \{(A, B), (A, C), (A, D), (A, E), (A, O), (A, P), (D, O), (E, N), (E, Q), (M, P), (M, N), (O, Q)\}$ が存在している。ここで

のナレッジワークスペースは、

$$KWS = (\{ (\alpha, \beta), (\alpha, \gamma), (\beta, \epsilon), (\beta, \delta), (\gamma, \theta), (\zeta, \eta) \}, \{ (B, \alpha), (C, \alpha), (M, \alpha), (B, \beta), (D, \beta), (E, \gamma), (N, \gamma), (D, \delta), (P, \epsilon), (O, \zeta), (Q, \eta), (Q, \theta) \}, \{ (A, B), (A, C), (A, D), (A, E), (A, O), (A, P), (D, O), (E, N), (E, Q), (M, P), (M, N), (O, Q) \})$$

と表現される。

その名称が示すように、ダイナミックコミュニティは、固定的なものではなく、あるイベントが発生したときに、そのイベントに関連する知識および関連知識を有した人々をナレッジワークスペースから動的に同定することによって構成されるグループである。グループとして抽出された人々は、当該イベントを解決するために、互いの知識を共有し交換することになる。そして、イベントが解決されると当該ダイナミックコミュニティは消滅する。

以下では、ダイナミックコミュニティの必要性を従来のコミュニティ研究と対比しながら示し、続いて、ダイナミックコミュニティの形成するプロセスおよび特徴的な性質について述べる。

2.1. ダイナミックコミュニティの必要性

コミュニティによる知識共有と知識共創の既存の研究では、新たにコミュニティに参加した初心者／ノービスが、正当的周辺参加(LPP; Legitimate Peripheral Participation) [5]のプロセスを通して、エキスパートとともに仕事をしながらエキスパートの知識や技能を身に付け、自らも徐々にエキスパートになっていくという点に、着目している。その背景となっているのは、エキスパートは常にエキスパートであり、知識移転の方向はエキスパートからノービスへの一方通行であるという仮定である。

しかしながら、知識の深化と細分化とが同時に進行しつつある現在、このような想定では対処できない状況が多くみられるようになってきている。あるドメインではエキスパートであるような人が他のドメインではまったくのノービスとなる場合もあるし、同じようなドメインであっても自らの専門から少し外れた事柄についてそれほど熟知していない専門家も多い。特に情報技術産業においては、必要となる知識や技能が変化する速度が大きいため、エキスパートがエキスパートであり続けられる保証はない。つまり、ある人があるドメインを専門としていると

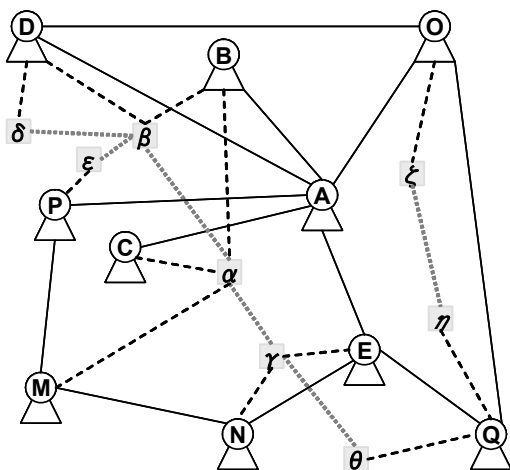


図1: ナレッジワークスペースの例

いう事実だけでは、ある瞬間に発生したタスクについてもエキスパートたりえるかは定かではなくなりつつある。

エキスパートやノービスといった呼称は、個々の人間の知識レベルを表現するものとして、もはや人の属性ではありえない。むしろ、タスクに大きく依存するものであることを前提に、深化し細分化した知識とそれを有する人との間の関係についての相対的な属性として捉えられるべきものである。

本研究では、このような前提に基づいて、ある人をエキスパートとして同定する作業は、ある特定の情報ワークタスクが明らかになって初めて可能となるものであり、そして、即時的かつ動的におこなわれるべきである、と考えている。このように、ある特定の情報ワーカーの特定の情報ワークタスクの発生に対して、エキスパートたる人々を集め構成したものがダイナミックコミュニティとなる。

2.2. ダイナミックコミュニティの形成プロセス

ダイナミックコミュニティは、次のようなプロセスで形成される(図2)。

いま、図1のナレッジワークスペースで示した関係の中で、A という人が情報 α を必要とするタスクを抱えているとし、これを $\langle A, \alpha \rangle$ と表記する。情報 α に関連する他の情報は情報 β と情報 γ となっている。また、同じナレッジワークスペース内で情報 α との関係(具体的な応用分野により、「知っている」「興味がある」「使いたい」「作成した」などのように定義される関係)を持つ人は B、

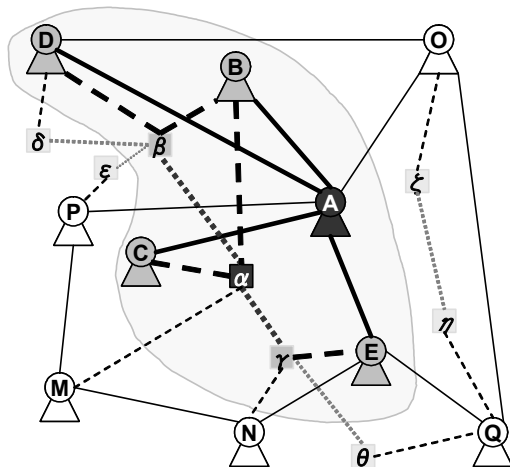


図 2: ダイナミックコミュニティ $\text{DynC}(A, \alpha)$

C, M, である。また情報 β あるいは情報 γ と関係を持つ人として、B, D, E, N, がいる。

$\langle A, \alpha \rangle$ のために形成される「ダイナミックコミュニティ $\text{DynC}(A, \alpha)$ 」に入るのは、以上の B, C, D, E, M, N の内、これまでに A と何らかのインタラクション履歴のあった B, C, D, E, であるので、 $\text{DynC}(A, \alpha) = \{A, B, C, D, E\}$ となる。

M と N については、それぞれ情報 α と情報 γ とに関係しているが、A とのインタラクション履歴を有しないため、ダイナミックコミュニティ $\text{DynC}(A, \alpha)$ には入らない。また、A とのインタラクション履歴は有しているものの情報 α, β, γ との関係を持たない O もダイナミックコミュニティ $\text{DynC}(A, \alpha)$ には入らない。

2.3. ダイナミックコミュニティの特徴

ダイナミックコミュニティ $\text{DynC}(A, \alpha)$ は、情報 α を必要とするようなタスクを抱えている A が存在するという状況に対して動的に形成されるもので、当該タスクに関して有益な知識を提供するポテンシャルを有した人々から成るコミュニティである。

このコミュニティは固定的なものではなく、タスクに応じてその都度形成される「恣意的」なコミュニティである。つまり、同じ A のタスクに関してのダイナミックコミュニティであっても、情報 α ではなく別の情報 β を必要とするタスクをおこなうときには、情報 α の場合とは異なったコミュニティが形成される。したがって、A は、あるコミュニティに固定的に属しているのではなく、A のおこなうタスクによって様々なコミュニティに属することになる。

また、同じ情報 α を必要とするタスクであっても、そのタスクをおこなう人が異なれば、たとえば A ではなく B であれば、形成されるダイナミックコミュニティは異なるものとなる。つまり、ダイナミックコミュニティはタスクおよびタスク実行に必要な情報に依存するものであるが、タスクをおこなう人それぞれに「相対的」なコミュニティとなる。これは、A と B とで彼らを取り巻く人間関係が異なることに起因している。

恣意性と相対性というダイナミックコミュニティのこれら二つの特性は、従来の知識共創支援における問題点に対処しうるものであると考えている。

3. 支援基盤環境のアーキテクチャ

ダイナミックコミュニティの形成を支援し知識共創のプラ

ットフォームとして機能する基盤環境を構築するためには、情報検索やエキスパート発見などの技術的課題に関して対処するだけでなく、ダイナミックコミュニティに積極的に参加する動機や共有された情報に対する信頼などといった社会的側面をも統合的に考慮した、ソーシャル・テクニカルな観点が必要である。

従来のコミュニティ理論の多くは、個々のメンバーがコミュニティに属しているか否かということだけに注目するものといえる。コミュニティ内メンバーであれば関係は均一であるとされ、コミュニティ内部でのメンバー同士の相互の近接度などはほとんど無視されていた。ダイナミックコミュニティでは、個々のメンバーが互いにどのような関係にあるのかに着目するとともに、人間関係、信頼、動機、といったコミュニティメンバーの社会的側面を考慮した基盤環境を構築しようとしている。

本節では、提案する基盤環境のコンセプトレベルでのアーキテクチャを示し、構成する五つのサブシステムについて説明する。

3.1. 知識リポジトリ

知識リポジトリは、前述したナレッジワークスペースをモデル化したものである。従来の知識リポジトリシステムと同様、ある特定のドメインの知識を情報として抽出、表現し、リポジトリに格納する。本研究の知識リポジトリが既存の知識リポジトリアプローチと異なるのは、情報と情報の間に存在する関係だけではなく、情報と人との関係および人々との関係も格納する点である(図3)。

また、本研究では、コミュニティを外部に対してオープンなものとして捉え、ある特定のドメインでのインタラクション履歴だけでなく、メンバー間のインタラクションをできるだけ多く取り入れることを考えている。たとえば、業務上では同じプロジェクトに従事したことはないが趣味の集まりで親しくしている関係であるとかメールのやりとりは頻繁におこなっている関係であるとかいったインタラクションである。当然のことながらプライバシーには留意する必要があるが、問題のない範囲で可能な限り多くのインタラクション履歴を知識リポジトリに格納する。

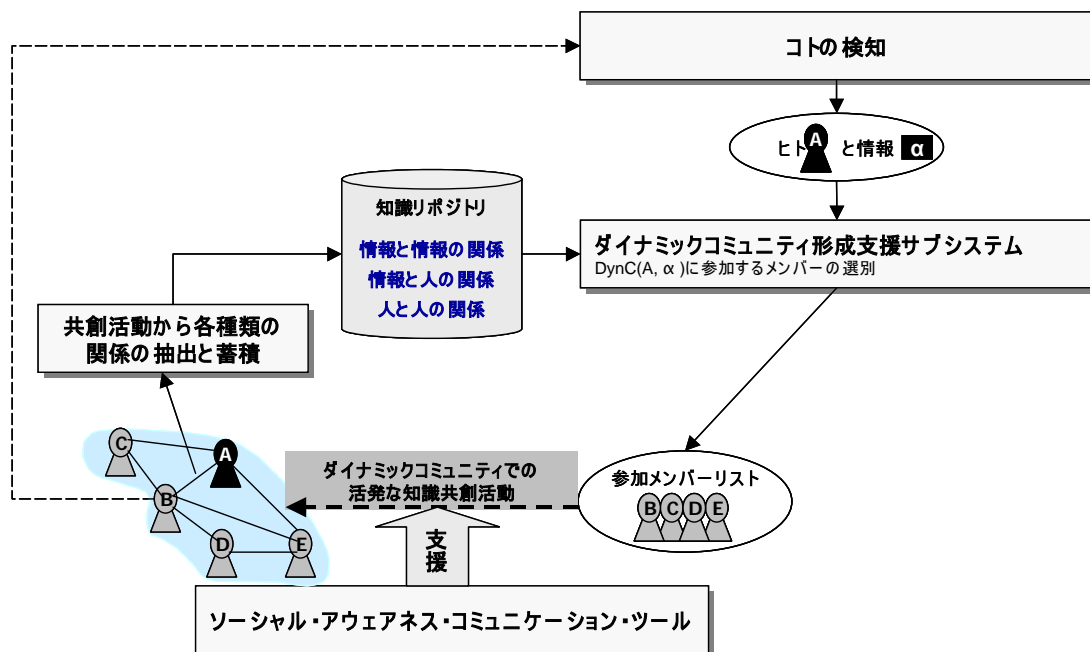


図3: ダイナミックコミュニティの形成を支援する基盤環境のアーキテクチャ

3.2. コトの検知

コトの検知のためのサブシステムは、支援対象となるナレッジワークスペースをモニターし、そこで発生するイベント(コト)を、ダイナミックコミュニティ形成のトリガーとなるものとして検知する。コトの検知は、コンピュータシステムによって自動的に起こることが可能な場合もあるし、人間が介在して能動的に生じさせる場合もある。

前者の場合であれば、後述するソフトウェア再利用支援のためのCodeBrokerのように、エージェントを利用してプログラマのタスクをモニターし、プログラミング環境から再利用情報の要求を自動的に推測することによって、当該プログラマのためのダイナミックコミュニティ形成が可能になる。後者の場合であれば、コトの発生をシステムへと伝えるためのインターフェースが必要となる。

いずれの方法にしても、コトの検知のためのサブシステムは、誰がどのようなタスクを抱えているのかを分析し、「誰」と「タスク」とを出力として次のサブシステムへと提供することになる。

3.3. ダイナミックコミュニティ形成支援

本サブシステムは、ダイナミックコミュニティ形成のトリガーとなる検知されたコトの「誰」と「タスク」とを入力として受け付け、知識リポジトリに蓄積された各種の関係から、このコトに関わりのある人のリスト、つまりダイナミックコミュニティに参加する可能性のある人のリストを作成する。

図4で示すように、ダイナミックコミュニティ形成を支援するサブシステムは、関連情報検索、エキスパート同定、エキスパート選別、という三つのステップから構成される。

関連情報検索は、知識リポジトリに格納された情報と情報との関係をたどり、当該情報に関連する情報を検索する。情報間の「関係」は、応用分野と目的とに依存して様々である。たとえば、ソフトウェア再利用を支援するダイナミックコミュニティでは、プログラマが開発中のプログラムに類似したコンポーネント群が関連した情報となるだろう。それに対して、ソフトウェア保守を対象としたダイナミックコミュニティなのであれば、バグ報告の類似度ではなく同じプログラムについて報告された他のバグが関連情報となる。

二つ目のステップとなるエキスパート同定は、知識リポジトリに格納された情報と人との関係に基づき、前段階で検索された関連情報と関係を持つ人を検索するプロセスである。情報と情報との関係と同様、情報と人との

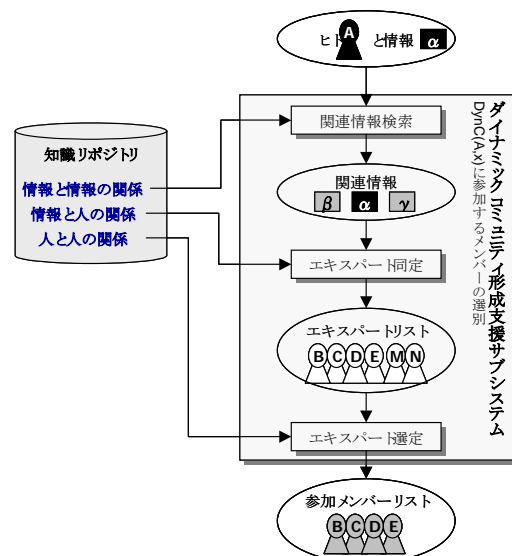


図4:ダイナミックコミュニティ形成支援サブシステムのアーキテクチャ

関係についても、その定義は、分野と目的とに応じて異なるものとなる。ソフトウェア再利用支援であれば、コンポーネントを作成した人や利用経験のある人がそのコンポーネントと関係を持つ人であると考えられる。ソフトウェア保守であれば、バグが影響するプログラムを作成した人、保守した人に加え、そのプログラムを利用したコードを作成したプログラマなどもエキスパートとして同定されることになる。

ここまでで同定されたエキスパートの集まりは、情報からのつながりで導出されたものとなっており、そのタスクを抱えた「人」のためのダイナミックコミュニティにはなっていない。同定されたエキスパート候補者の中から適切なエキスパートを選別するのがエキスパート選別の部分である。

選別にあたっては、知識リポジトリに格納された人との関係に基づいて、当該タスクを抱えた知識ワーカーと同定されたエキスパートの社会的な人間関係を参照し、インタラクション履歴のあるエキスパートを選択する。たとえば、Aがある情報αを必要とするタスクを抱えていて、B、C、Dが情報αに関するエキスパートであった場合に、Bは一週間前に別の問題でAに助けをもらった、Cは以前に同じプロジェクトで仕事をしたことがある、DはAと面識がない、という状況であったとしたら、Bお

よび C が選別される。

また、ナレッジワークスペースに属する知識ワーカーは、個人が知識ワークを有しているが、多くのダイナミックコミュニティに参加することによって本来の知識ワークを阻害してしまう可能性がある。したがって、エキスパート選別にあたっては各メンバーの負担を分散するように配慮することが必要となる。

3.4. ソーシャル・アウェアネス・コミュニケーション

選別されたエキスパートがダイナミックコミュニティのメンバーとして活発な知識提供をおこなえる状況を作るためには、ソーシャルアウェアネスを考慮したコミュニケーションツールが必要である。ダイナミックコミュニティの成否は各メンバーが動機を持って積極的に参加することに大きく依存する。したがって、各メンバーの動機を損なわないようにソーシャルアウェアネスが考慮されなければならない。

A が情報 α を利用する際のダイナミックコミュニティ $\text{DynC}(A, \alpha)$ が形成される際に、選別されたエキスパートにその旨を伝えるメッセージは十分に注意を引くものとして伝達されなければならないが、そのメッセージの伝え方は、エキスパート自身の本来の知識ワークを阻害しないものとなっているのが望ましい。本来の仕事を邪魔するような伝え方ではエキスパートの参加動機を損なうことになる。

また、 $\text{DynC}(A, \alpha)$ に選別されたエキスパートのリストは A に対しては提示しない方が望ましいのではないかと考えている。当該ダイナミックコミュニティへの参加要請が送信される先のエキスパートのリストが A に対して明示されると、最終的に参加したエキスパートと参加しなかったエキスパートとが誰であるのかが A に伝わることになる。参加要請を受けたエキスパートの側にとってみると、自身の参加／不参加が A に伝わるということは一種の社会的な圧力となる。短期的には、少々忙しい場合でもエキスパートが参加する駆動力となるため、プラスの面が大きいかもしれない。しかし、長期的な観点からは、望ましくないであろう。むしろ、A に対して参加／不参加の状況が伝わらない方が、エキスパートには言い逃れの余地 (excuse space) が残されることになり、余裕のあるときにだけ参加するという態度がとりやすい。知識を提供する側 (この場合、選別されたエキスパート) の意欲や時間を最優先とするのは、活発な知識共有と知識移転の鍵となる。

エキスパートに対して参加要請のメッセージを提示する

際に、適切な付加情報を伴わせることも、コミュニティに参加する動機に様々な影響を与えるだろう。 $\text{DynC}(A, \alpha)$ への参加要請が B へと伝達されるとする。たとえば、この要請メッセージとともに、前の週に A が B のためのダイナミックコミュニティに参加したという履歴を提示すれば、B の参加動機にプラスの影響を与えるだろう。ただ、このような社会的な力も、先に述べた言い逃れの余地をうまく残しておかないと、長期的にはネガティブな効果となってあらわれる可能性がある。

ダイナミックコミュニティへの参加動機を高めるための別の手がかりとしては、ソーシャルキャピタル[3]というコンセプトが利用できると考えている。ソーシャルキャピタルとは、他者の協力を引き出すことを可能にする個人々の資産であり、他者への協力をおこなうことによって蓄積される資産である。

知識ワーカー間の知識交換と相互支援の履歴をデータとして知識リポジトリに格納することによって、個々の知識ワーカーのソーシャルキャピタルに関する一つの指標として「エキスパートポイント」の導入が可能となる。エキスパートポイントの持ち点が高ければ、他の知識ワーカーから協力を得られる可能性を大きくすることができる。また、ナレッジワークスペース内において、ドメインエキスパートとしてのある種の地位を得て、他の知識ワーカーからの評価や尊敬といった社会的に認知されることになる。これはダイナミックコミュニティへの参加動機を高めるポジティブな圧力としての役割を果たすと考えられる。

3.5. 新たな知識の抽出と蓄積

ダイナミックコミュニティは、アドホックかつダイナミックに構成されるものとして、タスクの分析とエキスパートの選別とを経て新たなコミュニティとして「発生」し、当該タスクの終了を持って「消滅」する。ダイナミックコミュニティの発生と消滅の間に交わされるコミュニケーションや議論を通しておこなわれる知識活動は、新たな知識として知識リポジトリへと格納され蓄積される。

新たな知識として蓄積される際には、その知識とその知識の創出に関わった知識ワーカーとが関連するものであるという「情報と人との関係」も同時に知識リポジトリに格納される。

また、ダイナミックコミュニティでの知識交換や知識共創をおこなうプロセスの最中に、新たなダイナミックコミュニティを形成するトリガーも発生しうる。

4. ソフトウェア開発における応用

ダイナミックコミュニティの理論を、様々な具体的な分野に応用するべく、現在継続的に研究をおこなっている。本節では、ダイナミックコミュニティの理論に沿った拡張を進め実装しているソフトウェア再利用支援システムについて述べ、その他の応用分野における理論的な議論をおこなう。

4.1. ソフトウェア再利用支援システム

ソフトウェア再利用は、検索、理解、使用という三つのプロセスから構成される[1]。再利用支援に関する既存の研究の多くは検索に注目しており、理解や使用といった側面については重視していない。理解と使用が高度に知的な作業でありコンピュータによる支援が困難であることに起因しているのであろう。

以下では、開発者同士の相互支援を通してソフトウェア再利用を促進することを狙って、ダイナミックコミュニティの理論に沿って機能拡張している再利用支援システム CodeBroker [11]について述べる。システムの挙動に関する説明は、図3および図4において用いたプロセスの順におこなう。

コトの検知 CodeBroker は、Java プログラム A のエディタ (Emacs) をモニターし、ドックコメント (Doc Comment) の入力を検知するとドックコメントの内容を抽出する。図5に示した例では、A がある整数区間での乱数を生成しようとしていることを検知している。

関連情報検索 CodeBroker は、ドックコメントから抽出した内容を検索要求とし、情報検索の手法を用いて再利用リポジトリに格納されているコンポーネントとの類

似性を計算する。類似しているとされたコンポーネントはエディタ下部のバッファ内に自動的にデリバリされる。図5では「getInt」が最上段にあらわれ再利用可能となっている。デリバリされたコンポーネントをクリックすることによりドキュメントが提示される。そして、ナレッジワークスペース内のプログラム中に当該コンポーネントを使用したプログラムが存在すれば、CodeBroker は最も簡単なものを自動的に例示する。

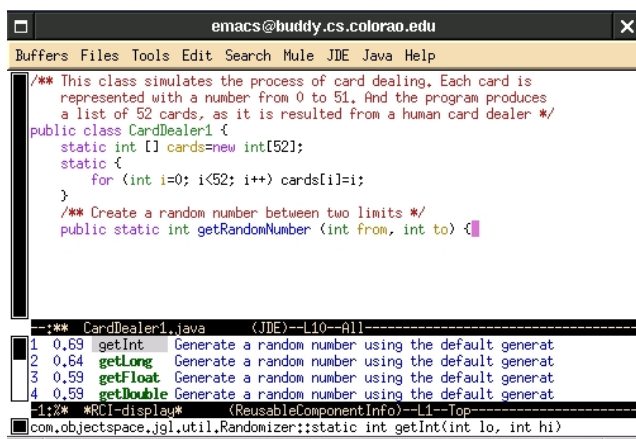
エキスパートの同定と選別 上の使用例だけではコンポーネントの利用方法が分からない場合、この getInt を使用したことのあるプログラマに助けを求める必要が生じる。そこで、CodeBroker は、ナレッジワークスペース内のプログラマのディレクトリから getInt を使用した Java プログラムを探し、その作成者をエキスパートとして同定する。プログラマ B, C, D, E の四人であるでしょう。この中でプログラマ E 以外はプログラマ A とのインタラクション履歴があったとした場合、プログラマ B, C, D がエキスパートとして選別される。

参加要請メッセージの送信 エキスパートの同定と選別はバックグラウンドで実行され A さんの知るところとはならない。生成されたエキスパートの内容についても A は知ることがない。A はデリバリバッファの中から getInt に対して「Get peer help」というメニュー項目を選択するだけである。これを受けて CodeBroker は、A に知らせることなく B, C, D の三人に参加要請メッセージを送信する。

知識交換と知識共創 たとえば、B が A を手助けしようとして A からのメッセージに返信したとする。この返信メッセージは、同じように参加要請を受けた C および D に対しても送られる。そのため、B が A を助けようとしていることは C と D の知るところとなり、B のソーシャルキャピタルの増加をもたらす。参加要請のメッセージが C と D に送られていることは本人以外が知るところではないので、C と D はネガティブな圧力がかからない。もちろん、その後、A のためのダイナミックコミュニティに C や D が加わることは可能である。このプロセスにおいて交換されるメールはすべて知識リポジトリに格納されるので、同じコンポーネント getInt に関して知りたいプログラマがあらわれた際には参照可能となる。

4.2. その他の応用分野

ダイナミックコミュニティの応用分野として、



```
emacs@buddy.cs.colorao.edu
Buffers Files Tools Edit Search Mule JDE Java Help

/** This class simulates the process of card dealing. Each card is
represented with a number from 0 to 51. And the program produces
a list of 52 cards, as it is resulted from a human card dealer */
public class CardDealer1 {
    static int [] cards=new int[52];
    static {
        for (int i=0; i<52; i++) cards[i]=i;
    }
    /** Create a random number between two limits */
    public static int getRandomNumber (int from, int to) {

--:** CardDealer1.java (JDE)--L10--A11-----
1 0.69 getInt Generate a random number using the default generat
2 0.64 getLong Generate a random number using the default generat
3 0.59 getFloat Generate a random number using the default generat
4 0.59 getDouble Generate a random number using the default generat
-1:** *RCI-display* (ReusableComponentInfo)--L1--Top-----
com.objectspace.jgl.util.Randomizer::static int getInt(int lo, int hi)
```

図5: CodeBroker の使用例

アプリケーションソフトウェアの利用支援や E-ラーニング支援, そして知識マネジメント支援や地域コミュニティの活性化など, 様々なドメインでの適用を考えているが, 現時点では理論的な議論に留まっている. 以下では, それらの中でもスムーズに適用可能だと思われるソフトウェア保守の支援への応用をダイナミックコミュニティの文脈で述べる.

ソフトウェア開発期間の終了後に, 開発者が他のプロジェクトにアサインされ保守作業が他のソフトウェアエンジニアに任されるケースは非常に多い. また, 一つのソフトウェアの開発と保守が地理的に分散した状況でおこなわれることも増加している[4]. このような状況において, 保守対象となるプログラムを開発した人間が容易にアクセス可能な所において, 元々の意図や設計について訊ねる, という場面は非常に稀なものとなる. ここから生じる問題についてダイナミックコミュニティは一つの解を提供できるのではないかと考えている.

保守担当者 A の保守対象がプログラム α となっているとしよう. プログラム α には開発の終了までおそらく複数のプログラマーが関わってきたであろう. これらの人々のリスト (α に関するエキスパートリスト) は CVS などのデータを分析することにより可能である[6]. このデータに対して, A とエキスパートリスト内の人々との人間関係を重ね合わせることを可能にすれば, 手助けをしてくれる可能性という点でよりポテンシャルの高いプログラマーに対して参加要請をおこなうことによって, ダイナミックコミュニティ DynC(A, α) を形成できるだろう.

5. おわりに

本稿ではダイナミックコミュニティの理論的な枠組みの提案をおこなった. ダイナミックコミュニティが知識管理アプローチと知識リポジトリアプローチを統合したコンセプトに基づいたものであることについて述べ, ダイナミックコミュニティの形成を支援するソシオテクニカル基盤環境のアーキテクチャを示した. 本研究は, ダイナミックコミュニティ情報環境の構築に向け今後の発展が望まれる段階にあり, 様々なドメインの実務者からのフィードバックや議論を通して, 次のステップへとつなげていきたいと考えている.

6. 謝辞

本研究の一部は, 文部科学省独創的革新技術開発研究提案公募制度による研究助成を受けておこなわれました. ダイナミックコミュニティ理論に関して研究の初期

段階から御議論いただいた中小路東京大学特任教授に感謝します.

7. 参考文献

- [1] Fischer, G., S. Henninger, and D. Redmiles, Cognitive Tools for Locating and Comprehending Software Objects for Reuse, in *Proceedings of 13th International Conference on Software Engineering (ICSE'91)*, Austin, TX, 318-328, 1991.
- [2] Fischer, G. and J. Ostwald, Knowledge Management — Problems, Promises, Realities, and Challenges. *IEEE Intelligent Systems*. **2001**(January/February):60-72, 2001.
- [3] Fischer, G., E. Scharff, and Y. Ye, Fostering Social Creativity by Increasing Social Capital, in *Social Capital*, M. Huysman and V. Wulf, (eds.), (in press), 2004.
- [4] Herbsleb, J. and A. Mockus, An Empirical Study of Speed and Communication in Globally-Distributed Software Development. *IEEE Transactions on Software Engineering*. **29**(3):1-14, 2003.
- [5] Lave, J. and Wenger, E. *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press: Cambridge, UK, 1991.
- [6] Mockus, A. and J. Herbsleb, Expertise Browser: A Quantitative Approach to Identifying Expertise., in *Proceedings of 2002 International Conference on Software Engineering (ICSE'02)*, Orlando, FL., 503-512, 2002.
- [7] Nardi, B.A., S. Whittaker, and H. Schwarz, It's Not What You Know, It's Who You Know: Work in the Information Age. *First-Monday: Peer-Reviewed Journal on the Internet*. **5**(5), 2000.
- [8] 野中郁次郎, 竹内広高. 『知識創造企業』 東洋経済新報社. 1996 (梅本勝博訳)
- [9] Polanyi, M., *The Tacit Dimension*, Doubleday: Garden City, NY. 1966.
- [10] Wenger, E., *Communities of Practice — Learning, Meaning, and Identity*, Cambridge University Press: Cambridge, UK, 1998.
- [11] Ye, Y. and G. Fischer, Supporting Reuse by Delivering Task-Relevant and Personalized Information, in *Proceedings of 2002 International Conference on Software Engineering (ICSE'02)*, Orlando, FL., 513-523, 2002.