

OSS 開発のための Web コミュニティ「C-DOS」における 自己組織化マップ (SOM) を用いたプロジェクトメンバの特徴分析

河村 進吾 田路 真也 森 秀樹 上原 稔

東洋大学大学院 工学研究科 情報システム専攻

オープンソース・ソフトウェア (OSS) の開発は、集合知 (Wisdom of crowds) の性質を持つ代表例である。より質の高い集合知を発揮するためには、開発経験のないライトユーザの意見や発想の取り込みが不可欠である。筆者らが提案するシステムは、集合知の持つ性質 (多様性, 独立性, 分散性, 集約性) を出発点にしたユーザ参加型開発支援 Web コミュニティである。ここでは、開発経験のないライトユーザを開発参加者として取り込み、今までのソフトウェア開発モデルでは得られなかった新しいアイデア, 人材を発掘することを目指す。本稿では自己組織化マップ (SOM) を用いて、開発グループ編成時における多様性を保障する新しいシステムを提案する。

A Method of Project Member Arrangement of Using Self-Organizing Maps for a Web Community to Support the Development Open Source Software

Shingo Kawamura Shinya Touji Hideki Mori Minoru Uehara

Department of Open Information Systems, Graduate School of Engineering, Toyo University

The process of open source software (OSS) development may be the same as the formation of wisdom in crowds. In this paper, we propose a web community to support the development OSS. They include the 4 factors in wisdom of crowds - diversity of opinion, independence, decentralization, and aggregation. Also, this system invites light users without software skills development to explore different approaches and opinions.

We propose a method by which keep the diverse of opinions the group is organized to include variety of different opinions using self-organizing maps (SOM). Further, while organizing then using SOM analyses, the performance of this method can be evaluated.

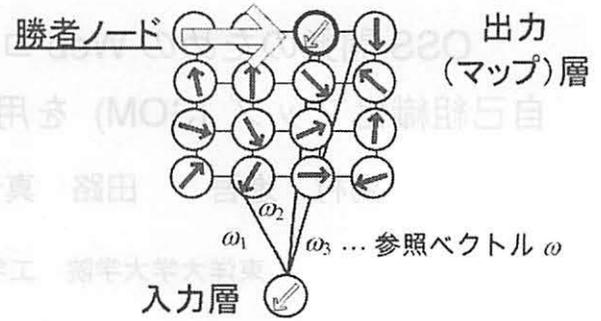
1 はじめに

オープンソース・ソフトウェア (OSS) 開発のためのユーザ参加型開発支援 Web コミュニティを提案する。特徴として、ソフトウェア開発経験のある技術者や各分野の専門家のみならず、開発経験のないライトユーザの参加を可能とする。また、開発グループの編成において、リーダー的役割を持つ個人を必要とせず、サーバ側に分類・評価の処理を実装し、完全にシステムによって人員の振り分け、管理をさせることで、その代役を務めさせる。実装は既存の SNS, Wiki のモデルを基盤とし、開発工程はスパイラルモデルを参考にしたモデルを考案する。筆者らが目指すものは、ユーザによる自発的なコンテン

ツの発信と、開放型のコミュニティによる新しい知の獲得を可能にする Web2.0 型開発支援環境の構築である。本稿では、集合知の発想を基点とした本システムの提案と、実現へのソリューションとして、強化学習アルゴリズムの1つである自己組織化マップ (Self-Organizing Maps; SOM) を用いて、登録された開発候補者 DB から、プロジェクトメンバを各工程へ分類する仕組みを考察する。2章では SOM について説明し、3章では、集合知について説明する。4章にて、システムの具体的な提案を行なう。5章で SOM を用いたユーザの特徴分析についての実験を行ない、6章でまとめと今後の課題を述べる。

2 SOM

SOMは、ヘルシンキ大学の Kohonen[1]によって考案された、教師無し学習型ニューラルネットワークの一種である。ヒトの脳における視覚野をモデル化した、入力層と出力（マップ）層の2層からなる多層式ネットワークを持つ。SOMのアーキテクチャを図1(a)に示す。



(a) SOMのアーキテクチャ

$$d_i = \sqrt{\sum_{j=1}^n (x_j - \omega_{ij})^2} \quad (1)$$

x : 入力ベクトル, ω : 参照ベクトル

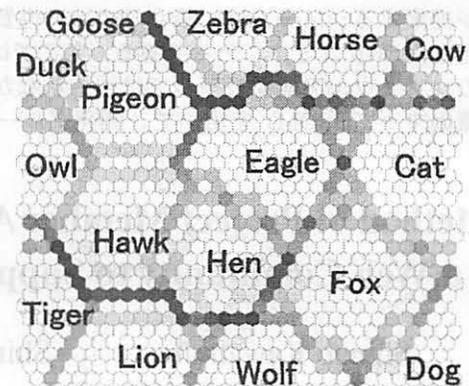
$$\text{if } i \notin N_c, \text{ then } h_{ci} = \alpha_0(1-t/T) \quad (2)$$

$$\text{else } h_{ci}(t) = 0$$

$$N_c(t) = N_c(0)(1-t/T) \quad (3)$$

SOMでは、多次元の入力を「入力ベクトル ($X = x_1, x_2, \dots, x_n$)」として与え、2次元格子状に配置されたニューロン（ノード）との間の距離（ユークリッド距離：式1）を計算する。ノードは初期値において重み値を持っており、重み値と入力値の間で、最も距離が近いノードを勝者ノードとして選択し、そのノードと近傍の重みを近傍関数と呼ばれる評価関数（式2, 3）を用いて繰り返し更新することによって、初期状態においてランダムな重み値を持っていた各ニューロンが、対象の特徴を学習する[2]。例として、動物の特徴データを入力として、SOM分析を行なうと図1(b)のようになる。このような特徴から、SOMはクラスタ分析やデータマイニングの手法として注目されており、応用研究の実績もある。[3]

SOMによって得られたマップから、マップ上で近い距離のユーザ同士ほど類似性をもったユーザであることがわかる。本研究では、提案するWebコミュニティにおいて、参加者の多様性を評価する手段としてSOMを選択した。

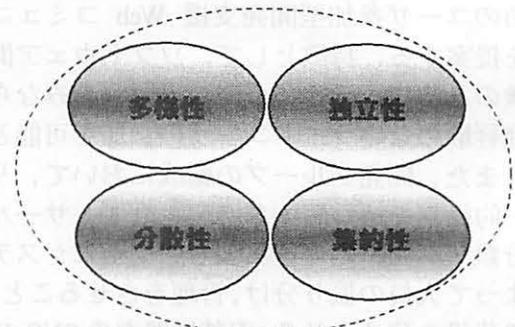


(b) 特徴解析の例

図1: SOM

3 集合知 (Wisdom of crowds)

集合知とは、「正しい状況下では、集団はきわめて優れた知力を発揮し、それは往々にして集団の中でいちばん優秀な個人の知力よりも優れている」[4]という考え方である。図2にその概念を示す。集合知を満たす条件として、意見の多様性、独立性、分散性、集約性の4つが挙げられる。昨今のWebサービスやオープンソースは、集合知の性質を持っている。



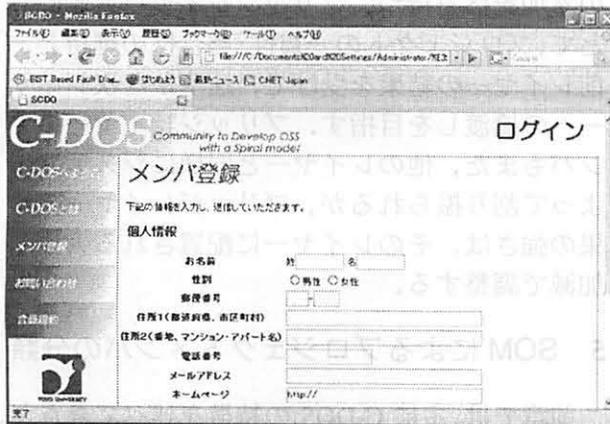
The wisdom of "crowds"

図2: 集合知の概念

4 C-DOS (Community to Develop OSS with a Spiral model)

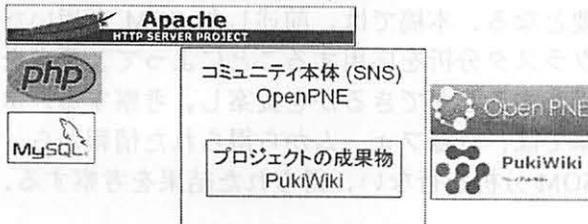
4.1 C-DOS の概要

C-DOS は、OSS 開発のためのユーザ参加型開発支援 Web コミュニティである。画面構成案を図 3(a)に、実装にあたってのシステムの構成を図 3(b)にそれぞれ示す。C-DOS では、プロジェクト提案コミュニティと、プロジェクト開発/進行コミュニティの 2つの性格を有している。コアとなるコミュニティの実装には、オープンソースの SNS 環境である OpenPNE [5]を用いることとする。



(a)

オンライン (Apache + PHP + MySQL)



オフライン



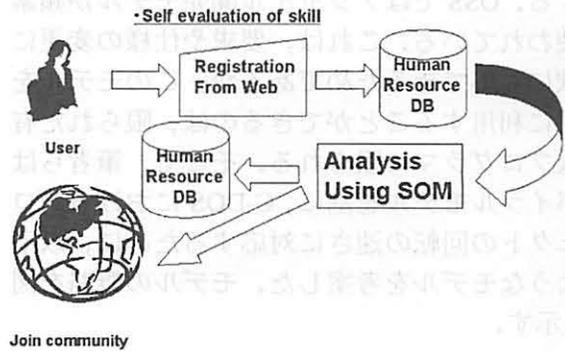
(b)

図 3: C-DOS の画面とシステム構成

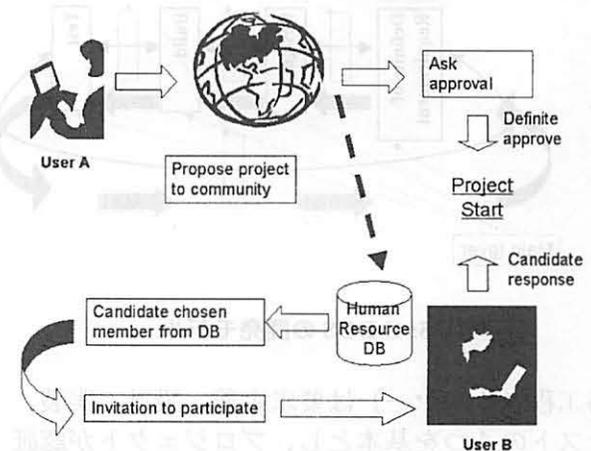
4.2 ユーザ参加

C-DOS では、希望者は条件を問わずにコミュニティに参加登録をすることができる。ユーザ登録の際に、SWEBOK (Software Engineering Body Of Knowledge) [6]を用いて、ユーザは自分のソフトウェア開発スキルについての自己評価を行ない、アンケートから嗜好や関心のある話題に関する情報を提供する。C-DOS でのユーザ

登録の流れを図 4(a)に示す。



(a) ユーザ参加



(b) プロジェクトの提案

図 4: C-DOS における参加と提案の流れ

4.3 プロジェクト提案

登録が完了したユーザは、開発候補者 DB に登録されるが、同時にコミュニティへの参加者としての権利も与えられる。ソフトウェア開発のアイデアがある、もしくは既存のソフトウェアに対する改良点、要望などがあれば、すぐにプロジェクトとして話題を提供することができる。ユーザ発のプロジェクト提案は、SOM を通して開発者候補として認められたコミュニティ参加者に招待メッセージが送られ、参加者を募る。参加希望者が一定数を超え、この提案に対して、コミュニティ内で一定の賛同を得られた場合、プロジェクトは正式に発足する。だが、発案者は「発案するだけ」でプロジェクトのリーダーにはならない。プロジェクトの提案から発足までの流れを図 4(b)に示す。

4.4 開発モデル

C-DOS の実装にあたって、我々はスパイラルモデル[7]を参考にした新しい開発モデルを提案する。OSS ではアジャイル開発モデルが頻繁に使われている。これは、要求や仕様の変更に柔軟に対応できるためであるが、このモデルを有効に利用することができるのは、限られた有能なプログラマに限られる。そこで、筆者らはスパイラルモデルを基に、C-DOS におけるプロジェクトの回転の速さに対応するために、以下のようなモデルを考案した。モデルの概略を図5に示す。

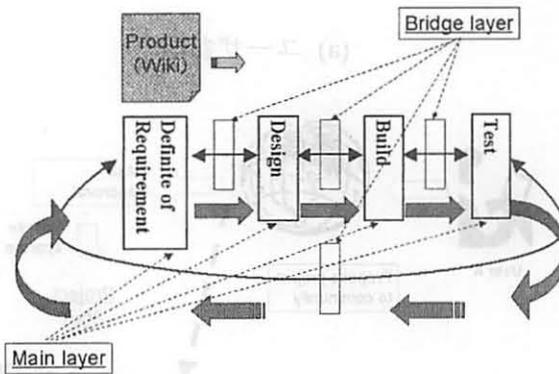


図5: C-DOS の開発モデル

各工程（レイヤー）は要求定義、設計、実装、テストの4つを基本とし、プロジェクトが認証された後、プロジェクト参加者らは開発メンバとして、SOM 分析の結果、各レイヤーへ配属される。各レイヤーに配属されたメンバは他のレイヤーの作業を閲覧することはできるが、参加することはできない。これは、レイヤー別に人を配置することで、複数のプロジェクトへの参加が可能になるためである。各レイヤーでの仕事が完成し次第、隣のレイヤーにその仕事を託すことになる。各レイヤーでの開発支援コミュニティは、Wiki と掲示板からなる。参加者間での議論は掲示板で行い、その結果をまとめたものをレイヤーにおける成果物として Wiki へアップロードし、順次更新してゆく。Wiki の更新はレイヤー上のすべてメンバが権利を持つが、更新の確定をする際にはレイヤー上のメンバから、一定以上の同意を得ることが必要になる。その際の同意を計るための評価基準は、「今回の更新で、成果物の完成度は以前よりも良くなっているか？悪くなっているか？」である。良いと思うメンバが過半数を超えれば Wiki の更新は確定される。また、その際には更新された

Wiki の満足度を、開発メンバ以外のコミュニティ参加者に評価してもらうことにする。満足度が一定数を超えた場合、そのレイヤーでの作業は終了となり、次のレイヤーにバトンが渡される。

以上が、筆者らが集約性をモデル化したシステムである。このシステムにおいて、特定のリーダーに決断を求めることを必要とせず、集団の意思によって開発を進めることができる。また、各レイヤー間にサブ工程として、ブリッジレイヤー (Bridge layer) を設ける。ブリッジレイヤーに配属されたメンバは両隣のレイヤーへの参加資格を持つ。ブリッジレイヤーを置くことで、プロジェクトの一貫性・整合性の維持と、前レイヤーの結果を受けて、速やかな次レイヤーへの橋渡しを目指す。ブリッジレイヤーのメンバもまた、他のレイヤーと同様にシステムによって割り振られるが、ブリッジレイヤーの効果の強さは、そのレイヤーに配置される人数の加減で調整する。

5 SOM によるプロジェクトメンバの分類

前章では、主に C-DOS の特徴を述べてきたが、C-DOS に求められる能力の初期段階として、メンバを各レイヤーに振り分けるための処理が必要となる。本稿では、前述した SOM を用いたクラスタ分析を応用することによって、要求を満たすことができるかを提案し、考察する。本章では、登録フォームから得られた情報から、SOM 分析を行ない、得られた結果を考察する。

5.1 スキル評価の指標

スキル評価の基準は大きく2つからなる。1つは、ソフトウェア開発に関係するスキルや経験からなり、もう1つは、情報分野以外での自分の興味や学習経験についてアンケートからデータを収集する。これは、ジョージア工科大学の教育方針「Creating Symphonic-Thinking Computer Science Graduates for an Increasingly Competitive Global Environment」[8] を参考にした。この方針では、情報技術者育成のために、情報分野と、1つの副専攻を学ぶことを重視している。本システムにおいても、ユーザを「ITスキル+IT以外のスキルや興味」から評価する。だが、これはあくまでユーザの特徴を分析するための指標であり、学力や能力の優劣は重視しない。

どちらの評価においても、Bloom の分類法[9]

に従い、ユーザは自分のスキルを6段階で自己評価する。6段階の詳細は、K: Knowledge (知っている), C: Comprehension (理解している), AP: Application (適用できる), A: Analysis (分析できる), S: Synthesis (統合できる), E: Evaluation (評価できる)からなり、このうち、Kが最も低次のレベルで、Eが最高レベルである。

5.2 実験

コミュニティ参加希望者の興味・趣味と、ソフトウェア開発に関する知識について SOM 分析を行なう。ソフトウェア開発に関する知識については、SWEBOK[5]を用いて評価する。これは、大学卒業後に少なくとも4年間の実務経験を経たプロフェッショナルが身に付けるべき知識の体系として、IEEE を中心に2001年に制定されたものである。なお、実験環境としてオープンソース数理解析環境の R[10]を用いた。詳細な実験状況を表1および表2に示す。

表1: SOM 分析の概要

入力データ	マップの形状	学習回数
26次元×20件	3×3 四角形	10000

表2: SOM への入力データ

(興味・趣味)

- I. 休日の過ごし方
- II. 読書
- III. 音楽鑑賞
- IV. 映画鑑賞
- V. スポーツ
- VI. 好きな色

(ソフトウェア開発に関する知識とスキル)

- I. 要求定義
 - (ア) ソフトウェア要求の基礎
 - (イ) 要求プロセス要求抽出要求分析
 - (ウ) 要求の仕様化
 - (エ) 要求の妥当性確認
 - (オ) 実践上の考慮事項
- II. 設計
 - (ア) ソフトウェア設計の基礎
 - (イ) ソフトウェア設計の主要な問題
 - (ウ) ソフトウェア構造とアーキテクチャ
 - (エ) ソフトウェア設計品質の分析と評価
 - (オ) ソフトウェア設計のための表記法
 - (カ) ソフトウェア設計戦略と手法
- III. コーディング
 - (ア) ソフトウェア構築の基礎
 - (イ) ソフトウェア構築のマネジメント
 - (ウ) 実践上の考慮事項

IV. テスト

- (ア) ソフトウェアテストの基礎
- (イ) テストレベル
- (ウ) テスト技法
- (エ) テストに関連した軽量尺度
- (オ) テストプロセス

5.3 結果

上記に挙げた入力フォーマットに沿った20人分のダミーデータを用いて SOM 分析を行った。ID: #1~#10 は開発経験がある、#11~#20 までは開発経験がないユーザのデータを設定した。図6に出力結果を示す。ここでは SOM の各ユニット(3×3の9ユニット、それぞれ左上から m_1 ~ m_9 とする)に、分類されたソフトウェア開発経験者と未経験者の ID をプロットした。各ユニットの形状は、最終的なノードの重み値をそれぞれ示している。

得られたマップから、開発経験のないユーザが同じユニットに属し、逆に開発経験のあるユーザは離れたユニットに属していることがわかる。SOM 分析を行なった場合、ノードの重み値はランダムに初期化されるため、分析ごとに異なったマップ配置を得ることがある。だが、いずれの場合でも開発経験を用いてユーザの特徴を分類させることができた。

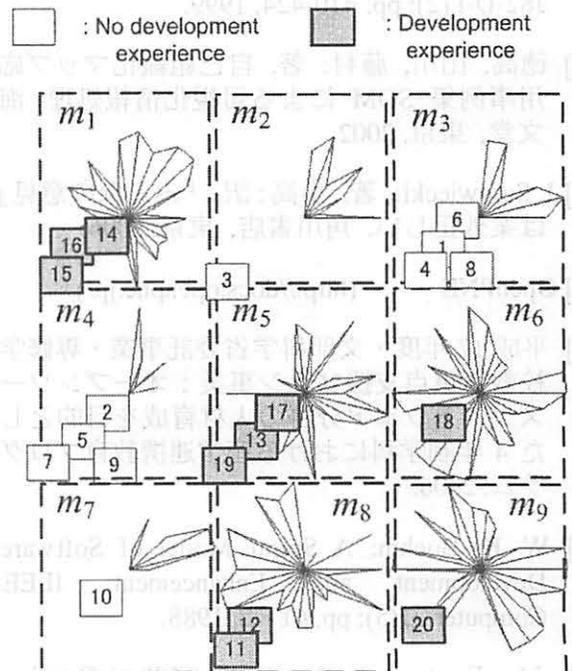


図6: SOM 出力結果

6 むすび

本稿では、OSS 開発支援 Web コミュニティ「C-DOS」における、SOM を用いたプロジェクトメンバの分類法について考察した。スキルに関するマップのみでなく、個人の性格や、心理テストのアンケートや調査の結果からもマップを生成することで、さらに詳しい分類のための手がかりになることが期待できる。

今後は、システムの実装に向けたスケーラビリティ、実現可能性の検証と、SOM 分析のための詳細なパラメータの検討、およびシステム全体の評価方法の策定が当面の課題である。

謝辞

本稿の執筆にあたり、東洋大学大学院・工学研究科・計算機アーキテクチャ研究室・博士前期課程修了生の大口尚紀氏に重要なアドバイスを賜りました。ここに感謝の意を表します。

参考文献

- [1] T. Kohonen: 著, Self-Organizing Maps. Springer, New York, 1995.
- [2] 安永, 八谷: 自己組織化マップハードウェアのフォールトトレランス評価. 信学論 J82-D-I (2): pp. 410-424, 1999.
- [3] 徳高, 山川, 藤村: 著, 自己組織化マップ応用事例集-SOM による可視化情報処理. 海文堂, 東京, 2002.
- [4] J. Surowiecki: 著, 小高: 訳, 「みんなの意見」は案外正しい. 角川書店, 東京, 2006.
- [5] OpenPNE (<http://docs.openpne.jp/>)
- [6] 平成 17 年度・文部科学省委託事業・専修学校教育重点支援プラン事業: オープンソースソフトウェア分野の人材育成を目的とした 4 年制学科における産学連携教育プログラム. 2006.
- [7] W. B. Boehm: A Spiral Model of Software Development and Enhancement. IEEE Computer 21(5): pp. 61-72, 1988.
- [8] M. Furst and R. A. DeMillo: Creating Symphonic-Thinking Computer Science Graduates for an Increasingly Competitive Global Environment. 2006. (http://www.cc.gatech.edu/images/pdfs/threads_whitepaper.pdf)
- [9] S. Bloom (Ed.): Taxonomy of Educational Objectives: The Classification of Educational Goals; pp. 201-207; B. Susan Fauer Company, Inc. 1956.
- [10] The R Project (<http://www.r-project.org/>)