

インターネット指向KJ法支援システム

杉浦 茂樹*

An Internet-Oriented KJ Method Support System

Shigeki SUGIURA

In this paper, we studied on the internet-oriented KJ method support system. The Internet-oriented KJ method support system is a Java-based groupware for creation of new-idea. As the result of practical use of Java technology's advantages of platform independence, portability, and rich Internet-support, our system can run widely platforms than former studies including GUNGEN, D-ABUDUCTOR, and KJ editor.

Keywords: CSCW, KJ method, Java, Internet

* 東北学院大学教養学部助教授 Touhoku Gakuin University

1. はじめに

近年のコンピュータとネットワークの急激な発展による高度情報化社会の到来は、社会や生産組織の大規模化と複雑化に拍車をかけている。このような変化に対応するために、創意工夫をこらして、新しいことを考え出したり、何かを改善したりすることを支援する、創造性開発技法に関する研究が盛んに行われている[1][2]。

国内では、創造性開発技法として、川喜田二郎氏が開発したKJ法[3][4]が企業などで広く用いられている。このため、KJ法を対象とした研究が多い。

従来のKJ法では、参加者が一ヶ所に集まって作業を行う必要があり、KJ法の実施の上での障害となっていた。コンピュータとネットワークを活用することにより、参加者は職場や自宅など身近な場所でKJ法に参加することが可能となり、移動のための負担の軽減に加えて、実施機会の増加が期待される。また、遠く離れた参加者と共同作業を行うことで、知的触発が誘発され、発想が広がるなどの効果も期待される。

分散環境を活用したKJ法支援システムとしては、郡元[5]、D-ABDUCTOR[6]、KJエディタ[7]などがあるが、いずれも、実行環境がワークステーションやMacintoshなど、あまり一般的ではないプラットフォーム上に構築されており、だれもがどこでも気軽に実施するには程遠い状況となっていた。

本研究では、広く普及したインターネットとPCを効果的に活用して、KJ法の実施を支援するシステムの構築を目指す。具体的には、インターネットとの親和性が高く、ハードウェアやOSなどのプラットフォーム依存性が少なく、

広く普及しているJava言語を用いてシステムを構築した。

本稿では、まず、第2章で、研究の対象であるKJ法について説明する。次に、第3章で、本システムの設計・実装を通して判明した問題点とその解決法について述べる。第4章では、作成したKJ法支援システムの動作の概要について説明する。第5章はまとめである。

2. KJ法

2.1 創造性開発技法

職場や家庭などの日常生活の中で、いろいろと頭を働かせ、知恵をしぼり、創意工夫をこらして、新しいことを考え出したり、何かを改善したりする能力を創造性や独創性などと呼ぶ。

これらの創造性や独創性を支援する技術を「創造性開発技法」と呼ぶ。創造性開発技法は、大まかに以下の4種類に分けることができる。

- 発散技法…アイデアを出すための思考法
ブレインストーミング、ブレインライティング、チェックリスト法、NM法など
- 収束技法…アイデアをまとめあげる技法
KJ法、PERT法など
- 統合技法…発散と収束を繰り返していく技法
ワークデザイン法、ハイブリッジ法など
- 態度技法…創造力の養成法
キャストイング・ロールプレイ法、カウンセリングなど

本研究では、これらの創造性開発技法の中で、企業の研究・開発活動などで広く使用されているKJ法を対象とする。

2.2 KJ法

日本の文化人類学者川喜田二郎氏が考案した収束的技法の一種で、川喜田二郎氏の頭文字をとってKJ法と名付けられている。

KJ法は、以下の4つの手順に従って、実施される(図1)。



図1 KJ法の手順

[手順1] ラベル作成 (意見出し)

参加者の脳に蓄えられた知識や経験を吐き出す段階。

具体的には、ブレインストーミングを行い、出された情報を要約して、「ラベル」と呼ばれる小さなカードに書き込んでいく。1枚のカードには、1つの情報のみを記入する。

[手順2] 島作成 (グループ編成)

[手順1] で作成されたラベルを、直感により分類していく段階。

具体的には、以下の3つの手順により構成される。

①カード拵げ

ラベルを机の上などに重ならないように広げ、1枚1枚のラベルに書かれた内容を丹念に読み取っていく。

②カード集め

直感的に「近い」と感じたラベルを集めて、「島」と呼ばれるグループを作成する。このとき、どの島にも属さないカードは、一匹狼として無理にどこかに入れなくてもよい。

③表札作り

島の内容を表す、「表札」を付ける。

島作成では、まずカードから小グループを作成する。次に、小グループ同士で中グループを作成する。さらに、中グループ同士で大グループを作成する。

[手順3] A型図解化

[手順2] で作成された島を、意味の近さにより空間的に配置する。さらに、輪取りや線などを記入して、グループ同士の関係を示す。

[手順4] B型文章化

[手順3] で作成された図解をもとにして、まとめの文章を作成する。

3. システム設計・実装で判明した問題点とその解決法

本研究で開発する創造性開発支援環境は、以下の機能を提供する。

- ・創造性開発の代表的な技法であるKJ法の4つ手順をPC上で実施する
- ・ネットワーク上での共同作業の支援
- ・作業履歴の保存と再生

本システムの設計・実装を通して、以下のような問題点が判明した。

- (1) ラベルと島の移動機能の実現法
- (2) A型図解化の図形要素の実現法
- (3) ネットワークへの対応

以下、それぞれの問題点と解決法について述べる。

3.1 ラベルと島の移動機能の実現法

本システムの開発でまず問題となったのは、ラベルや島をどのようにして表示し、マウスでの操作に対応して移動させるかである。

この解決方法としては、大まかに分けて以下の2つの方法がある。

- ①**集中管理法**：ラベルや島などの部品は、主プログラムであるAppletクラスの単なるデータとして扱い、各部品の描画および動きの管理をすべて主プログラムであるAppletクラスで行う方法。
- ②**分散管理法**：ラベルや島などの部品を、JavaのGUIコンポーネントのクラス（Labelクラスや

Panelクラスなど）を拡張することにより作成し、各部品の描画および動きの管理を各オブジェクトに任せる方法。

この2つの方法の比較を表1に示す。

表1 ラベルと島の移動機能の実現法の比較

	集中管理法	分散管理法
プログラム作成の容易性	○	△
機能の拡張性	×	○

本研究では、機能の拡張性を重視して、基本的に②分散管理法を採用するが、部品間の協調が複雑となる処理では①集中管理法も積極的に利用する、ハイブリッドな方法でシステムを構築した。

具体的には、ラベルおよび島は、JavaのGUIコンポーネントであるLabelクラスおよびPanelクラスを拡張して、KJLabelクラスおよびKJIslandクラスとして作成した。

ラベルおよび島の移動は、KJLabelクラスおよびKJIslandクラスのそれぞれに実装された、マウス移動のイベントハンドラで検出されるが、実際の移動の処理はすべて主プログラムであるAppletクラスに委託する。このことにより、ネットワークへの対応はAppletクラスのみの変更で済み、また、機能の分離により保守性が向上した。

なお、島の移動では、集中管理方式が取り分け有効であった。

当初、島は他のGUIコンポーネントを中に含むことができるContainerクラスのサブクラス（実際にはPanelクラス）を拡張して作成し、島の大きさが決定した段階で、その島にどのラベルが含まれているかを判定し、該当するラベルの島への貼り付けを行い、島の移動時には、島

が含むラベルの移動を島自身が行う予定であった。この方法を用いることで、島の座標を変更することにより、Containerクラスの提供する機能により、島に含まれるラベルも自動的に移動される。

しかし、この方法には、いくつかの不具合があった。まず、単純に島を削除すると、島に含まれるラベルも一緒に削除されてしまう。また、ラベルを島の境界線を越えて移動可能にするのは、プログラミング的に非常に困難である。

これらの不具合を解決するために、最終的に採用した方法は、ラベルを島に貼り付けることをやめ、島の移動時には、主プログラムであるAppletクラスに実装された（指定された領域に含まれる部品を全部移動させる）メソッドを用いて処理を行うという方法である。

3.2 A型図解化の図形要素の実現法

本システムの開発で次に問題となったのは、A型図解化の図形要素（以下、単に「図形要素」とする）の実現方法である。以下の第1案、第2案を試行したが失敗し、最終的に第3案により実現できた。

(1) 第1案

図形要素は、主プログラムであるAppletクラスの単なるデータとして扱い、主プログラムであるAppletクラスの描画メソッド（paintメソッド）を変更して、図形要素の描画を行う。

第1案の問題点

この方法では、図形要素の上に、重量UIコンポーネントであるラベルや島が描画されてしまい、図形要素が隠れてしまう。

(2) 第2案

第1案の改良に加えて、ラベルおよび島の各クラスの描画メソッド（paintメソッド）の修正を行う。

第2案の問題点

ラベルおよび島のもとになった、LabelクラスおよびPanelクラスは、重量UIコンポーネントであるため、プログラムの改良が非常に困難であり、もし改良できたとしても実行のパフォーマンスが非常に悪くなる可能性がある。このため、第2案では改造の途中で断念せざるを得なかった。

(3) 第3案（解決方法）

第1案、第2案の問題点より、以下の2つの修正を合わせて行う。

- ①現在のラベルと島を軽量UIコンポーネントのもとにして作成し直す。
- ②新たに図形要素を表示するための軽量UIコンポーネントを作成する。

JavaのGUIコンポーネントには、重量UIコンポーネントと軽量UIコンポーネントが存在する。重量UIコンポーネントと軽量UIコンポーネントの比較を表2に示す。

表2 重量UIコンポーネントと軽量UIコンポーネントの比較

	重量UI コンポーネント	軽量UI コンポーネント
100%Java言語	×	○
背景の有無	必ず背景色で塗られる	透明化可能

一般的に用いられているAWT（Abstract Widget Toolkit）で提供される、LabelクラスやPanelクラスなどはすべて重量UIコンポーネントである。

AWTの機能を継承して、機能の拡張と軽量UIコンポーネント化したものにSwingがある。AWTからSwingへの変更を行うことにより、軽量UIコンポーネント化が可能である（しかし、AWTとSwingの互換性は完全ではないため、使用メソッドの修正などの変更が必要となる）。

①の修正では、ラベルをAWTのLabelクラスからSwingのJLabelクラスを拡張するように変更し、鳥をAWTのPanelクラスからSwingのJPanelクラスを拡張するように変更した。

②の修正では、透明なGUI部品のもとになるJComponentクラスを拡張することで、各図形要素の作成を行った。このとき、各図形要素は直接JComponentクラスを拡張するのではなく、JComponentクラスを拡張して抽象クラスであるKJGraphクラスを作成して、そのKJGraphクラスを拡張することによって作成した。このように、抽象クラスであるKJGraphクラスを定義することにより、新しい図形要素を追加する場合でも、元のプログラムには変更が不要となった。

本システムでは、A型図解化の機能として、以下の3つの機能を作成した。

- ・ 図形の選択（線・四角・円）
- ・ 図形の色の選択（黒・赤・青・黄・緑）
- ・ 図形の太さの選択（1～30）

3.3 ネットワーク対応

本システムでは、クライアント・サーバ方式に基づきネットワーク対応を実現した。これを実現するために、クライアントとサーバ間のデータの送受信と処理の方法とプロトコルを以下のように設計・実装した。

(1) クライアントとサーバ間のデータの送受信と処理の方法

クライアントからサーバ、および、サーバからクライアントへのデータの送信はいつ発生するか予測できない。本システムでは、クライアントとサーバのいずれでも、主プログラムとは別の処理の流れを持つことのできるスレッドを作成して、受信と処理を行っている。

本システムの構成（クライアント2台での例）を図2に示す。

このときの、データの送受信の動作は以下の通り。

- ①サーバ側でメインプログラムを立ち上げ、そのときログ保存用の一時ファイルが生成される。
- ②クライアント側でメインプログラムを立ち上げる。
- ③サーバに接続をする。そのとき、サーバ側ではそのクライアントに対するデータ受信・処理用スレッドを作成し、クライアント側ではデータ更新・表示用スレッドを作成する。
- ④ユーザの操作に応じて、クライアントのメインプログラムからデータを送る。
- ⑤サーバのデータ受信・処理用スレッドがデータを受け取る。
- ⑥データ受信・処理用スレッドはサーバのメインプログラムにデータを送る。
- ⑦サーバのメインプログラムは一時ファイルにデータを保存し、同時にサーバの保持しているスレッド全てにデータを送る。
- ⑧サーバのそれぞれのスレッドは自分の担当しているクライアントにデータを送る。
- ⑨クライアントのスレッドは受信したデータに

に基づき、各種データの更新や画面への表示を行う。

以下、④～⑨を繰り返す。

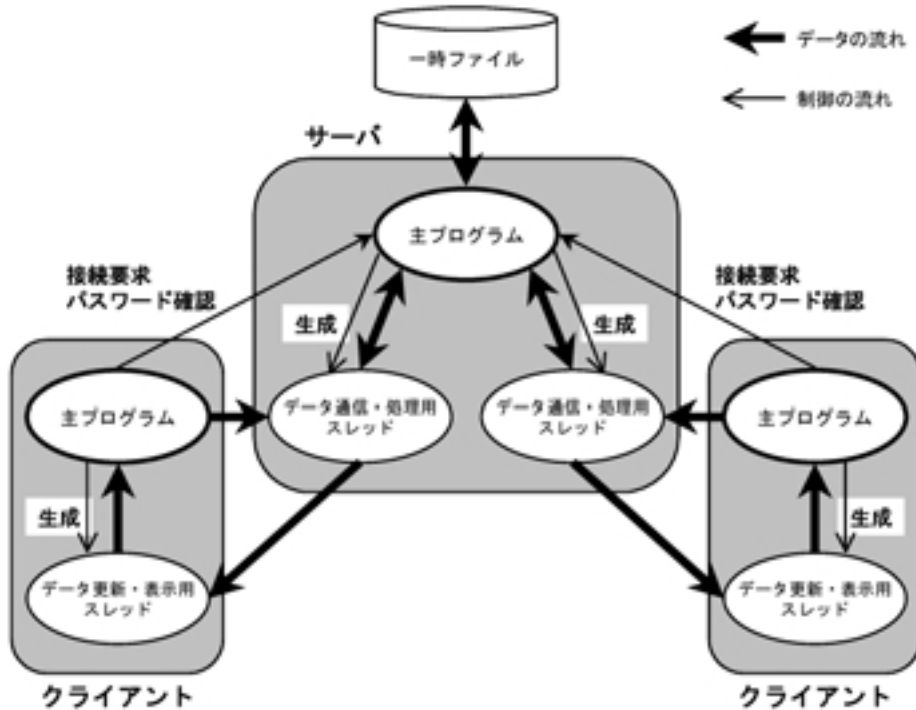


図2 本システムの構成

(2) プロトコル

本システムでは、クライアントとサーバのデータのやりとりを単純化するために、サーバは必要最小限の処理しか実装しない方針とした。

サーバは以下の3つの機能のみ有する。

- ・クライアント間のデータの中継。
- ・作業履歴（ログ）の保存と再生。
- ・接続の開始（パスワードによる認証）と切断。

以下に、サーバとクライアントが処理する命令を示す。

○サーバで処理される命令

EMIT : クライアントから送られてきた作業データをサーバの作業履歴用一時ファイルに書き込み、サーバに接続されている全てのクライアントにデータを送る。

CHAT-EMIT : クライアントから送られてきたチャットのデータをサーバのチャット用一時ファイルに書き込み、サーバに接続されている全てのクライアントにチャットのデータを送る。

HIST : 一時ファイルにある作業履歴とチャットのデータをHISTを送ってきたクライアントに送る。

PASS : パスワード認証を行い、成功したら接続を有効にする。

QUIT : QUITを送ってきたクライアントとの接続を切断する。

○クライアントで処理される命令

MAKE-LABEL : ラベルの作成

REMOVE-LABEL : ラベルの削除

MOVE-LABEL : ラベルの移動

MODIFY-LABEL : ラベルの内容の修正

SHUFFLE : ラベルのシャッフル

MAKE-ISLAND : 島の作成

REMOVE-ISLAND : 島の削除

MOVE-ISLAND : 島と中に含まれるラベルの移動

MODIFY-ISLAND : 島の表札の修正

DRAW-GRAPH : 線、四角形、円の作成

CLEAR-GRAPH : 線、四角形、円の削除

MAKE-COMMENT : コメント文の作成

REMOVE-COMMENT : コメント文の削除

MOVE-COMMENT : コメント文の移動

MODIFY-COMMENT : コメント文の文章の修正

START-SUMMARY : B型文章化の開始

MODIFY-SUMMARY : B型文章化の内容の修正

CHAT : チャット

START : クライアントでの操作を許可

STOP : クライアントでの操作を禁止

4. 本システムの動作

本システムの動作の概要を、以下に示す。

(1) ラベル作成

テキストフィールドにアイデアを書き込み、「発言」ボタンを押すとラベル作成され、発言順に配置される。



図3 ラベル作成

(2) 島作成 (①カード拡げ)

発言順に並んだラベルは「シャッフル」ボタンを押すとランダムに並び替えられる。

シャッフル機能は一見あまり意味のない機能



図4 島作成 (①カード拡げ)

に思える機能だが、カード並びの観点で非常に重要な機能となってくる。

そもそもKJ法は、バラバラとなったカードの情報群の中から人間の直感的な感覚をたよりに、問題を形成し、構造化されるように考えていくものです。

よってカード並びの段階でいかに関連性を失くし、バラバラにおいてあるかということが重要となる。発言順に並んでいるので関連性がないと思われるが、すでに発言順という時点で、前後の並び、発言した時間関係などによって人間は無意識にこれらが近いものと感じてしまうことが多々ある。全ての発言が終わった時点でこのシャッフル機能を使うことにより、ランダムにカードが並び替えられ、一切の関連性を失くしより人間の六感に頼る作業が可能となる。この機能は、従来のKJ法支援システムには実装されておらず、本システムの特徴的な機能の1つである。

(3) 島作成 (②カード集め)

直感的に似た感じのするラベルを近くに配置する。



図5 島作成 (②カード集め)

(4) 島作成 (③表札作りとグループ化)

テキストフィールドに表札のタイトルを書き込み、プルダウンメニューより小、中、大を選び、「グループ化」ボタンを押し、さらに、マウスを使って、島としてグループ化する範囲を指定する。



図6 島作成 (③表札作り と グループ化)

(5) 再配置

グループ化した後、島やラベルを移動させることができる。島の移動時には、島の内に入っているラベルは一緒に移動する。



図7 再配置

(6) A型図解化 (線、四角形、円)

輪どりで棒線などでグループ同士の関係を表示し、全体が姿・形をもった図解となるようにまとめる。

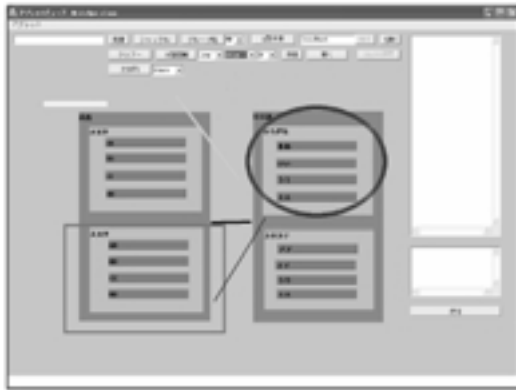


図8 A型図解化 (線・四角形・円)

(7) A型図解化 (コメント文)

(6)で作成した線、四角形、円などにコメント文を付加し、図を分かりやすくする。

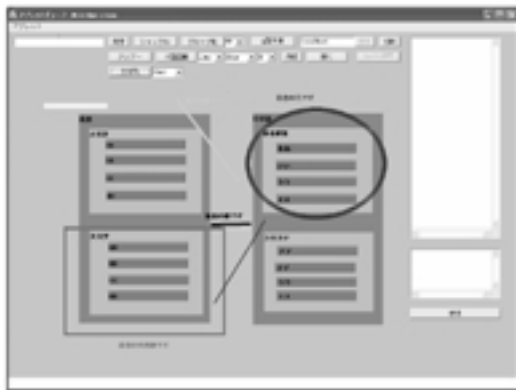


図9 A型図解化 (コメント文)

(8) B型文章化

(7)で作成した図をもとに、まとめの文章を作成する。



図10 B型文章化

(9) チャット機能

共同作業を円滑に進めるために、参加者同士で文字によるコミュニケーションを行うチャット機能を提供している。



図11 チャット機能

5. まとめ

本研究では、一般的なPCとインターネットを活用することにより、だれでもどこでもKJ法に参加することが可能となる、インターネット指向の創造性開発支援環境の設計と実装を行った。

本システムにより、遠隔の参加者と協調して

PC上でKJ法を実施することが可能となった。
今後の課題としては、より効果的、効率的に
KJ法を実施できるようにする、機能の設計と
実装を考えている。

<参考文献>

- [1] 特集：知の共有から知の協創へ、情報処理学会論文誌, Vol.46, No.1, pp.1-196 (2005).
- [2] 特集：発想支援システム, 人工知能学会誌, Vol.8, No.5, 1993年9月
- [3] 川喜田 二郎: 発想法—創造性開発のために, 中公新書136, 中央公論社 (1967).
- [4] 川喜田 二郎: 続・発想法—KJ法の展開と応用, 中公新書210, 中央公論社 (1970).
- [5] 重信 智宏, 吉野 孝, 宗森 純: 発想支援グループウェア郡元DXIIの開発, 情報処理学会研究報告, GN, Vol.2001, No.98, pp.49-54 (2001).
- [6] 三末 和男, 杉山 公造: 図的発想支援システムD-ABDUCTORの操作性の評価, 情報処理学会論文誌, Vol.36, No.1, pp.133-143 (1996).
- [7] 竹田 尚彦, 塩見 彰睦, 河合 和久, 大岩 元: カード操作ツールKJエディタを用いた協調作業実験, 情報処理学会研究報告, GW, Vol.93, No.56, pp.49-56 (1993).
- [8] 湯瀬 裕昭, 杉浦 茂樹, 乙藤 岳志, 相川 利樹: 視覚障害者と晴眼者のための分散環境を活用したKJ法支援システム, マルチメディア通信と分散処理ワークショップ論文集, IPSJ Symposium Series, Vol.2005, No.19, pp.339-34 (2005).

[2006年3月3日受付]

