## 平成22年度

## 筑波大学情報学群情報科学類 卒業研究論文

## 題目

# データの自動周回を利用したディスカッション におけるさりげない情報共有手法

主専攻 ソフトウェアサイエンス主専攻

著者 藤田 訓義

指導教員 田中二郎 三末和男 高橋伸 志築文太郎

### 要旨

多人数がグループを作成し1つの作業に取り組むことは重要で一般的なことである。そして、コンピュータ技術の普及につれてそのような作業をシステムによって支援するグループウェアの研究が多数なされている。グループで作業を行う際に、ディスカッションは必要不可欠なことである。なぜならばそれはグループを形成するメンバ間での相互理解を深め、1人で考えているときよりも素早く、より発展したアイデアを生み出すことを容易にするためである。しかしながら、ディスカッションを円滑に進める上でディスカッションに参加しているメンバが互いの考えすべてを共有することは難しく、個人の中で失われてしまうインスピレーションやそれに基づく発言が発生してしまう。本研究では、そのような発言を失われる発言と定義し、その基となるインスピレーションをディスカッションの論点をずらすことなくさりげなく共有するための手法としてデータサイクルというシステムを提案し、実装を行った。

# 目次

第1章	はじめに	1
1.1	グループ作業においてのディスカッション	1
1.2	ディスカッションにおける意見の共有	1
1.3	失われる発言	1
1.4	失われてしまうであろうインスピレーションの共有	2
1.5	本研究の目的	2
1.6	本論文の構成	3
第2章	要件定義とアプローチ	4
2.1	システムが対象とするディスカッション環境	4
2.2	システムが満たすべき要件	5
2.3	さりげない共有手法	6
2.4	実装したシステム	6
第3章	データサイクル	7
3.1	データサイクルの特徴	7
	3.1.1 自動周回するデータ	8
	3.1.2 データサイクルにおけるデータの提示	8
	3.1.3 失われる発言への気づき	8
3.2	データサイクルによるインタラクション	9
	3.2.1 出現させるデータの用意	9
	3.2.2 データを出現させる	10
	3.2.3 データサイクル上に出現しているデータの操作	11
3.3	想定される利用シナリオ	12
第4章	実装	13
4.1	ハードウェアの実装	13
	4.1.1 FTIR タッチパネル	13
	4.1.2 テーブルトップデバイスの構成	14
4.2	ソフトウェアの実装	16
	4.2.1 文字画像データの作成	16
	4.2.2 起動時の楕円軌道と「LOAD IMAGE」アイコンの描画	17

	4.2.3 データの読込み	18
	4.2.4 データの向きの決定と描画	20
	4.2.5 データの操作	21
	4.2.6 データの自動周回	21
第5章	関連研究	22
5.1	ディスカッション支援	22
5.2	テーブル型ディスプレイで行う協調作業支援	22
第6章	議論	24
第7章	結論	25
	謝辞	26
	参考文献	27

# 図目次

2.1	システムが支援するディスカッション環境	5
3.1	テーブル型タッチパネルディスプレイ上に実現されるデータサイクル	7
3.2	テーブルトップデバイスを囲むユーザに対するデータの向き	8
3.3	手元のデバイスにおいてコメントを文字画像データとし、保存するアプリケー	
	ション	10
3.4	画像データの一覧を表示するイメージパネル	10
3.5	タッチによるデータの選択	11
4.1	FTIR タッチパネルの仕組み	13
4.2	作成したテーブルトップデバイス	14
4.3	作成した赤外線 LED アレイの全体写真	15
4.4	作成した赤外線 LED アレイの拡大写真	15
4.5	赤外線 LED アレイの回路図	16
4.6	文字画像データを作成するアプリケーションと実際に作成される画像	17
4.7	楕円軌道と「LOAD IMAGE」アイコンの描画	18
4.8	データ読込み時のフロー	19
4.9	データの向きの例	20
4.10	楕円軌道上にあるデータ	21
4 11	<b>通堂</b> 状能と選択状能	21

## 第1章 はじめに

多人数がグループを作成し1つの作業に取り組むことは非常に重要であり、また一般的なことである。グループで作業を行うことによって、1人で行う場合に比べ効率よく、質の高い作業の実現が見込まれるようになる。そのため、コンピュータ技術の発展とともにそのような作業を支援するグループウェアの研究が多数なされている[1][2]。

### 1.1 グループ作業においてのディスカッション

グループで作業を行う際には、グループを形成しているメンバが互いの意識を共有し、次に何を行うか決定するためにディスカッションを行う機会が必要不可欠となる。多人数でディスカッションを行い、グループのメンバ間での相互理解を深めることで1人で考えているときには思いつかないようなアイデアを得ることができ、またそれらを基に、さらに発展したアイデアを生み出すことも容易になる。このように最終的な目的達成のための手法を容易に導くために多人数でのディスカッションは非常に重要で有効な手段である。

### 1.2 ディスカッションにおける意見の共有

ディスカッションに参加しているメンバが互いの意見を共有する方法として、一般的な方法は自分の考えを言葉にし、発言として他のメンバへ伝えることである。しかし、参加しているメンバそれぞれが自由に発言を繰り返した場合、ディスカッションが円滑に進まなくなってしまったり、論点がずれたりする恐れがある。そのため、たとえある瞬間に作業を進める上で有効となるかもしれないインスピレーションを受けたとしても、ディスカッションの進行を妨げてしまう恐れから各自でそれについての発言を制限してしまう場合がある。このようにディスカッションの場には現れず、ある個人の中にとどまったままとなってしまう発言は少なからず生まれてしまう。

#### 1.3 失われる発言

多人数でのディスカッションではしばしば誰かが自分の意見を他のメンバに伝えようと意見を述べ、他のメンバはその意見を理解しようと耳を傾けている状況が生まれる。ある人が発言を行っている際には、その他の人はたいていそれを遮ってまで発言をしようとはせず、次に自分が発言してもよい適切なタイミングをうかがう。また、発言を行う際も、その時に思っ

ていることすべてを発言するとは限らず、些細な事柄であったり、現在の論点とは異なるなど、わざわざ発言しなくてもよいと発言者自身が判断した事柄については発言に含めないことがある。

これらは、ディスカッションの論点をずらすことなく円滑に進行させるためには非常に重要なことである。しかし、あるインスピレーションを感じた瞬間から発言できる適切なタイミングが訪れるのを待つまでの間に、別の事柄に対する新たなインスピレーションが生まれてくることは十分にありえることであり、発言までの時間が長くなればなるほど受けたインスピレーションは薄れていき、発言しようとしていたことの一部、あるいはその多くを忘れてしまう恐れがある。または、発言を終えた後しばらくしてから発言し忘れてしまった事柄があったことに気づくことや、非常に些細な事柄であるために発言者があえて発言に含めなかった事柄はディスカッションの場には現れない。このように、まさにその瞬間に感じたインスピレーションについての発言など、ディスカッションの場に現れることなく失われていってしまう発言は数多く存在するはずである。そして、これらの発言に他のメンバが気づくことは決してないため、他のメンバがそこから受けることができたであろうインスピレーションも当然失われてしまうことになる。

本研究では以上のようなディスカッションの場においてメンバ間で共有されずに個人の中でとどまり、失われてしまうであろう発言を失われる発言と定義する。

#### 1.4 失われてしまうであろうインスピレーションの共有

本研究では前節で定義した失われる発言を他のメンバと共有することにより、従来の多人数でのディスカッションをよりよいものへと発展させることを目指す。例えば、ある人にとっては現在の論点とはまったく異なり、発言にするのがためらわれるようなインスピレーションであっても、それを他の人が見ることでそこから現在の論点に合った的確なインスピレーションが生まれる可能性がある。このように失われる発言となってしまうであろうインスピレーションを少しでも他者と共有することにより、そこから本来生まれることのなかった新たなインスピレーションを生み出すことをシステムが支援する。

#### **1.5** 本研究の目的

ある瞬間に受けたインスピレーションが失われる発言となってしまう前にそれを具現化し、他のメンバと共有することで新たなインスピレーションを生み出す支援を行う。また、共有する際にディスカッションの論点をずらすことのないさりげない共有手法を目指す。これらを満足するために本研究では、テーブル型タッチパネルディスプレイを用いて、任意のデータを自動周回させることのできるデータサイクルというシステムを提案し、実装を行った。

## 1.6 本論文の構成

本論文の構成は以下の通りである。2章では本研究で支援する対象とするディスカッション 形態と提案するシステムが満たすべき要件、またそこから導き出されるシステムデザインに ついて述べる。3章では本研究で提案するシステムであるデータサイクルの概要と、それを用 いたインタラクション方法、想定される具体的な利用シナリオについて述べる。4章では実際 に作成したテーブル型タッチパネルディスプレイとデータサイクルの実装について述べる。5 章では本研究に関連する研究について述べ、6章でシステムを実装していく中で明らかになっ た問題点について議論する。7章で本論文をまとめる。

## 第2章 要件定義とアプローチ

単にディスカッションと言っても、参加する人数に注目すれば数人で行うものから、数十人規模のプロジェクト単位で行うものもあり、またさらに大きな組織単位やコミュニティ単位で行うものもある。それに加えて参加するメンバが同じ場所に集う対面形式のものであるのか、ビデオ映像などを用いた分散形式のものなのかという違いがある。またその目的も様々であり、伝達、創造、調整、決定の4種類に分類される[3]。

本章では、その中で本研究がどのような形態のディスカッションを対象とするのか、また、 提案するシステムがどのような要件を満たすべきかを定め、それを満たすためのアプローチ と実装を行うシステムの概要について述べる。

### 2.1 システムが対象とするディスカッション環境

システムが支援するディスカッションの環境として次の4点の事を想定する(図2.1)。

- 創造を目的とするディスカッション
- ディスカッションに参加するメンバは6~8人
- 全メンバが同空間に在席し、テーブルを囲む対面形式でディスカッションを行う
- 全メンバは何らかの情報端末(ノートパソコン、タブレット型のコンピュータ、携帯情報端末など)を所持している

1 つ目としてディスカッションを行う目的は創造であるとする。この目的を達成するためには同期・同室型のグループウェアが望ましいとされる [4]。

2 つ目として、ディスカッションに参加するメンバが 6~8 人のものを想定する。小規模なプロジェクトなど、6~8 人程度で行うディスカッションが行われることは多い。ディスカッションに参加するメンバの人数が少なすぎてしまうと、具現化すべき失われる発言が発生する可能性は低くなり、人数が多すぎてしまうと失われる発言が発生しすぎてしまう。そのためこの程度の人数で行うディスカッションは失われる発言が発生した際にそれを具現化し、他者に提示することでディスカッションを発展させようとする本研究にとって適切な規模であると考える。

3つ目として、全メンバが同空間に在籍し、テーブルを囲んで対面形式でディスカッションを行うことを想定する。これはディスカッションを行う際に参加するメンバがテーブルを囲み、対面形式で互いの顔を見合いながら話を進めていくことが一般的であるためである。

4つ目として、ディスカッションに参加しているメンバ全員が何らかの情報端末(ノートパソコン、タブレット型のコンピュータ、携帯情報端末など)を所持しているものとする。近年、情報端末は近年ますます発展し、高性能化および小型化が進んでいる。そのため情報端末を個人が普段から所持することの負担は軽減し、むしろ所持していることですぐに情報の検索を行うことができるなどの利点が得られる。このようにディスカッションに参加するメンバがその場に情報端末を持ち込み、それを必要に応じて利用しながらディスカッションを行うことが特別なことではなくなってきている。

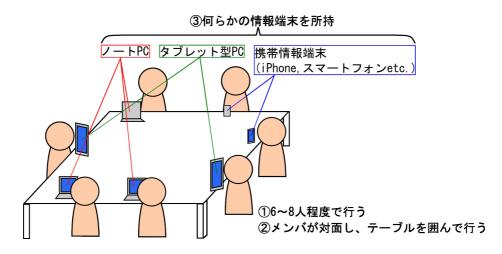


図 2.1: システムが支援するディスカッション環境

#### 2.2 システムが満たすべき要件

システムが満たすべき要件は次の2点であると考える。

- ◆ 失われる発言となってしまいそうなインスピレーションを具現化することで他者と共有 可能にする
- 具現化したインスピレーションを共有する際にディスカッションの進行を大きく妨げる ことがないようにする

1つ目は、失われる発言となってしまうインスピレーションを何らかの形で具現化し、他者と共有できるようにすることである。失われる発言となってしまうインスピレーションの中には、それを他のメンバが共有することでそこから新たなインスピレーションが生まれ、ディスカッションをより発展させる可能性を持っている。

2つ目は、失われる発言を共有する際に現在の論点をずらしてしまう等、ディスカッションの進行を大きく妨げることがないようにすることである。失われる発言とは 1.3 節で定義したように、ディスカッションの進行を円滑にするために失われてしまった発言のことである。

そのためそのような発言を具現化し他者と共有する際に、それを具現化したこと自体にディスカッションのメンバ全員が気づき、注目してしまうのでは結局のところ普通に発言することとなんら変わりはなくなってしまう。それゆえ、本研究では失われてしまうであろうインスピレーションの共有手法として、さりげなさをもたせた手法が適切であると考える。つまり、失われる発言を具現化する瞬間を他者に対して明示的にするのではなく、具現化の際は他者に気づかれずとも、それをなんらかの方法で他者に気づかせるというものである。

#### 2.3 さりげない共有手法

具現化したデータを他者に気づかせるための手法として、本研究では自動周回するデータ の流れを作成する。この流れに乗せたデータは乗せた地点から徐々に自動周回していき、い ずれ他のメンバの目にとまる。

ユーザが視認できる程度の速さでデータを周回させることにより、ユーザは目をデータに向ける動作を行うだけでそれを共有できる。もしデータを手元でよく見てみたいという場合のみ、それを自分の手元に持ってくるための操作を行えばよい。以上のような提示手法、操作手法をシステムが提供することにより、ユーザはディスカッションに向けられる注意をそらすことなくデータの閲覧が可能になると考える。

### 2.4 実装したシステム

以上より、本研究では失われる発言となってしまうであろうインスピレーションを画像データとしてテーブル型タッチパネルディスプレイ上に具現化し、それを自動で周回させることで他者のアウェアネスによるさりげない共有を可能とするシステムを実装する。

## 第3章 データサイクル

本章では提案するシステムであるデータサイクルの特徴、それを用いた具体的なインタラクション、及び想定される利用シナリオについて述べる。

## 3.1 データサイクルの特徴

本研究で提案するデータサイクルの特徴について述べる。データサイクルとは、自動周回するデータの流れをテーブル型タッチパネルディスプレイ上に投影することで実現されるシステムである。テーブル型タッチパネルディスプレイ上にデータサイクルが実現されている様子を図 3.1 に示す。

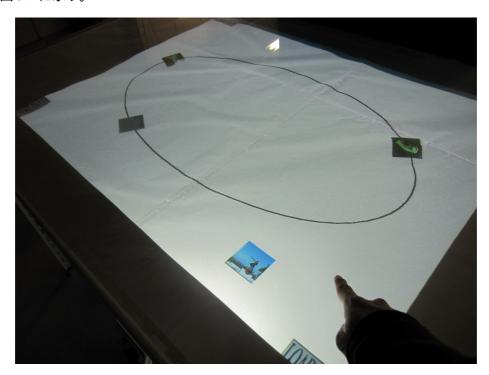


図 3.1: テーブル型タッチパネルディスプレイ上に実現されるデータサイクル

#### 3.1.1 自動周回するデータ

データサイクルは、テーブル中央にデータが自動周回するための楕円軌道が存在し、ユーザは具現化したインスピレーションをその軌道に乗せて周回させたり、周回しているデータを軌道上から自分の手元へと持ってきたりすることができる。

#### 3.1.2 データサイクルにおけるデータの提示

データサイクルはユーザがテーブルを囲んでディスカッションを行っていることを想定し、 またその支援としてさりげない共有を目的としているため、テーブル上のデータの閲覧がユー ザにとって容易でなければならない。

楕円軌道上で自動周回するデータはディスプレイの中心点から見て外側の方向を向く。また閲覧するときなど、楕円軌道上ではないところにあるデータは、画面を四分割した上でディスプレイの中心点から見て外側の方向を向く(図3.2)。このようにテーブルトップデバイスを囲んでいるユーザにとってデータが視覚情報として受け取りやすいようにデータの向きがその位置によって変わる。

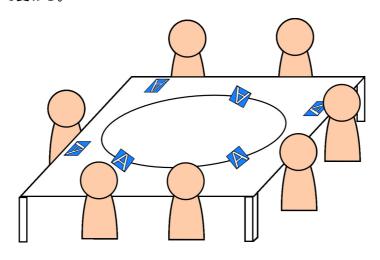


図 3.2: テーブルトップデバイスを囲むユーザに対するデータの向き

#### 3.1.3 失われる発言への気づき

楕円軌道上のデータは自動で周回し、やがてあるユーザの視界内に現れる。そしてユーザはその視界に入ったデータを見ることでその情報を共有する。つまり、ユーザはデータの閲覧を行うためにジェスチャなどの特別な操作をする必要がなく、データを閲覧する行為によってユーザの注意がディスカッションから大きくそれることがなくなる。これによってデータのさりげない共有が可能になると考える。

#### 3.2 データサイクルによるインタラクション

データサイクルでは、特定のディレクトリにある画像データをテーブルトップデバイス上から読み込むことでテーブルトップデバイス上に任意のデータを出現させる。読み込まれるディレクトリはユーザ間でファイル共有される。また、出現させたデータをドラッグ操作を行うことによって自由に移動させることができる。本研究におけるドラッグ操作とは、タッチパネルディスプレイ上で指を接触したまま滑らせる操作を指す。

#### 3.2.1 出現させるデータの用意

データサイクルでは、ユーザはある瞬間に受けたインスピレーションが失われてしまう前に、それに関連する画像データもしくはコメントを他者と共有する。それらのデータは手元のコンピュータを用いて用意する。

#### 画像情報

ある瞬間に受けたインスピレーションを具現化する際に、それに関連した写真などの画像情報を提示したい場合、ユーザはそのデータを共有している特定のディレクトリに移動させればよい。

#### 文字情報

画像情報だけでなく、インスピレーションをコメントのような文字情報として具現化させたい場合は少なからず存在すると考えられる。そのような文字情報をデータサイクル上で具現化できるようにするため、本研究ではデータサイクルを補助するための簡単なアプリケーションも実装した。このアプリケーションはユーザの手元のコンピュータ上で起動させ使用し、入力した文字列を画像データとして特定のディレクトリへ保存することができる(図3.3)。これを使用することで、ユーザはコメントなどの画像では表しづらいインスピレーションもデータサイクル上へ具現化することができる。

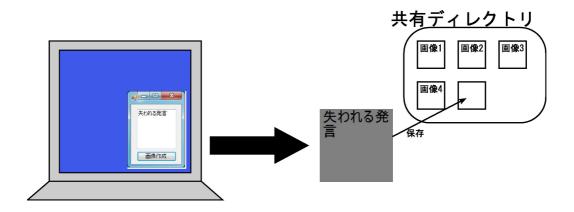


図 3.3: 手元のデバイスにおいてコメントを文字画像データとし、保存するアプリケーション

#### 3.2.2 データを出現させる

テーブルトップデバイス上に投影されたディスプレイの四隅にある「LOAD IMAGE」アイコンのいずれかをタッチすると、共有ディレクトリにある画像データの一覧が表示されたイメージパネル(図 3.4) が出現する。イメージパネルに表示された画像データをタッチすることでユーザはデータサイクル上へ任意の画像データを出現させることができる。画像データをデータサイクル上に出現させるとイメージパネルは閉じられる。

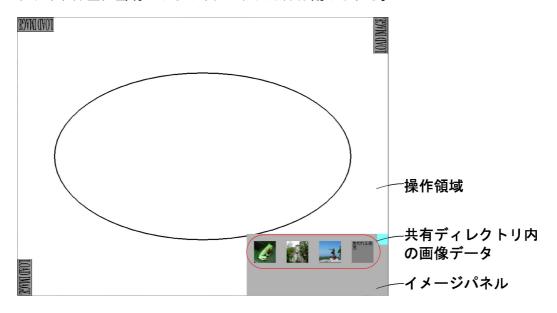


図 3.4: 画像データの一覧を表示するイメージパネル

#### **3.2.3** データサイクル上に出現しているデータの操作

表示されているデータをタッチするとデータの外枠が赤枠で囲まれ、選択状態であることを表す(図3.5)。この状態にあるとき、ドラッグ操作を行うことでユーザは自由にデータの移動を行うことができる。これによって、出現させたデータをディスプレイ中央にある楕円軌道に乗せ、自動周回させたり、自動周回しているデータを自分の手元へ持ってきたりすることができる。

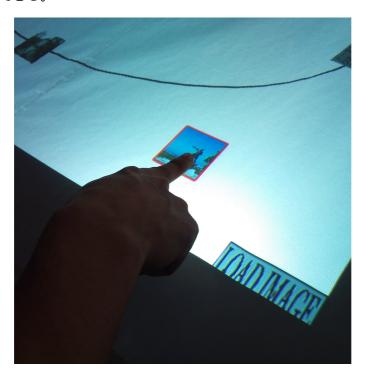


図 3.5: タッチによるデータの選択

#### データを楕円軌道に乗せ自動周回をさせる

データをドラッグ操作によって楕円軌道付近へ移動させていくと、楕円軌道から一定の距離 に達したときにデータは楕円軌道上に乗る。この状態で指を離し選択状態を解除するとデー タは楕円軌道に沿って自動周回を始める。

#### 自動周回しているデータを自分の手元へもってくる

楕円軌道上で自動周回を行っているデータをタッチして選択状態にすると、選択状態である間データは自動周回を止める。この状態でドラッグ操作を行うことによってユーザは楕円軌道上から自分の手元へデータを持ってくることができる。

### 3.3 想定される利用シナリオ

データサイクルを用いたインタラクション方法をより明確にするため、具体的な想定される利用シナリオを述べる。

#### 新商品の開発

新商品の開発における一場面を想定する。開発担当のAさんは新商品に使用する素材についての説明を行っている。その説明を聞いていたBさんはそこからふとデザインについてのインスピレーションが浮かんだ。しかし、現在ディスカッションの焦点となっている事柄はあくまでも素材として何を採用するかであり、デザインについては後ほど詳しく議論されるだろう。そのような思いからBさんはその瞬間に受けたインスピレーションを発言としてディスカッションの場に出すことをためらっている。そこで、そのデザインについてのインスピレーションを簡単に手元のノートパソコンのペイントツールを用いて作成し、データサイクル上に具現化し、自動周回させた。Bさんのデザインは軌道に乗って自動周回していき、やがて離れた場所にいるCさんの前に辿り着いた。Cさんはそれに気づき、興味を持ってその画像を手元へと持ってきた。そしてそのデザインを見て、現在Aさんが紹介している素材よりもさらによい素材が思い浮かんだ。Cさんはそれを発言として述べ、新商品の開発に貢献した。

このシナリオにおける失われる発言とは、A さんの素材についての発言を受けてB さんが思い描いたデザインについてのインスピレーションに基づく発言である。もしB さんがこのデザインについてのインスピレーションを発言としてディスカッションの場に提示した場合、そこからそのデザインが本当に良いのか、という議論へと移ってしまい、論点がずれてしまう恐れがある。このような場合にデータサイクルはその有用性が発揮されると考える。

## 第4章 実装

本研究で実装を行ったハードウェアの実装部、およびソフトウェアの実装部について述べる。

### 4.1 ハードウェアの実装

データサイクルの実現にあたり、本研究ではテーブル型タッチパネルディスプレイとして Han の方式 [5] に基づく FTIR タッチパネルを作成、使用した。

#### **4.1.1 FTIR** タッチパネル

Han の方式による FTIR タッチパネルはアクリル素材の板と赤外線を用いる。アクリル内部に赤外線を照射すると、アクリルと空気の屈折率の関係により内部全反射を起こす。(図 4.1(a)) この状態においてアクリル表面に接触すると、接触点で赤外線の内部全反射が阻害され拡散反射が起こる。(図 4.1(b)) この拡散反射した赤外線を検出することによって接触点の認識を行う。(図 4.1(c)) Han の方式であれば、比較的安価で容易にマルチタッチインタフェースを作成できるため、多人数が同時にシステムを操作することを想定しているデータサイクルの実現に適していると考えられる。

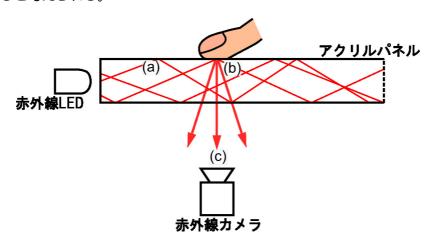


図 4.1: FTIR タッチパネルの仕組み

#### 4.1.2 テーブルトップデバイスの構成

本研究を進めるにあたり作成した FTIR タッチパネルのテーブルトップデバイスを図 4.2 に示す。

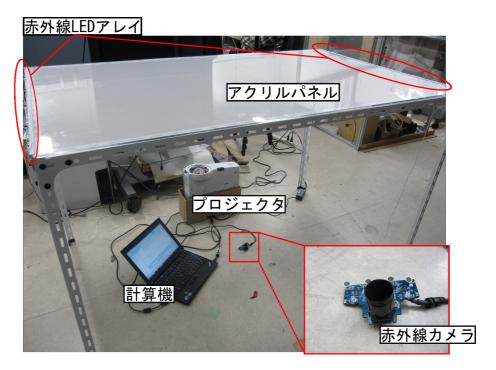


図 4.2: 作成したテーブルトップデバイス

デバイスの構成として、まずタッチパネルディスプレイとなるところには  $90 \, \mathrm{cm} \times 150 \, \mathrm{cm} \times 5 \, \mathrm{mm}$  のアクリルパネルを用意した。アクリルパネルの 2 つの短辺には赤外線 LED アレイが装着されており、ここからアクリル内部へ赤外線が照射される。赤外線 LED アレイとはコの字型塩ビ部品に赤外線 LED である東芝製  $\mathrm{TLN231}(F)$  を  $3 \, \mathrm{cm}$  間隔で均等に配置したものである。図 4.3 は実際に作成した  $2 \, \mathrm{cm}$  本の赤外線 LED アレイの全体図であり、図 4.4 はその拡大図である。赤外線 LED を発光させるための回路は内藤ら [6] のものを参考とし、電源には  $12 \, \mathrm{V}$ 、 $1.5 \, \mathrm{A}$  以上の出力が可能なものを使用し、抵抗には 47 のものを使用した。実際に作成した回路の図を図 4.5 に示す。一方、テーブルトップ中央下方には、赤外線除去フィルタを取り除き可視光除去フィルタを装着することで赤外線のみを検出できるようにした  $30 \, \mathrm{fps}$  のカメラPLAYSTATION Eye が設置してある。これにより、アクリルパネル下方へ拡散反射した赤外線を検出を行う。また、焦点距離  $f=6.48 \, \mathrm{mm}$ 、解像度が WXGA ( $1280 \times 800$ ) である EPSON 製単焦点プロジェクタ  $EB-410 \, \mathrm{W}$  も配置されており、インタフェース部の投影を行う。プロジェクタとアクリルパネル間の距離は  $590 \, \mathrm{mm}$ 、表示面の大きさは  $1135 \, \mathrm{mm} \times 690 \, \mathrm{mm}$  である。



図 4.3: 作成した赤外線 LED アレイの全体写真



図 4.4: 作成した赤外線 LED アレイの拡大写真

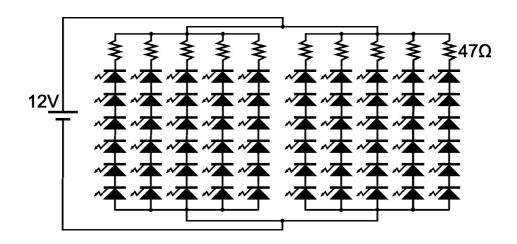


図 4.5: 赤外線 LED アレイの回路図

### 4.2 ソフトウェアの実装

開発環境として Visual Studio2008 を使用し、システム全体としては C++ によって実装を行った。また、実装を行う上で OpenCV2.1 $^1$ 、OpenGL $^2$ を用いた。OpenCV は、Intel が開発・公開したオープンソースのコンピュータビジョン向けライブラリであり、OpenGL は、SGI 社が中心となって開発した 3D グラフィックスのための API であり、2D グラフィックスも可能である。

文字画像データを作成するアプリケーションのみ、C# によって実装を行った。

#### 4.2.1 文字画像データの作成

文字列を画像データに変換するアプリケーションはデータサイクルのシステムとは独立したアプリケーションである。ユーザは自分の手元のデバイス上でこのアプリケーションを実行する。テキストボックスに任意の文字列を入力した後、「画像作成」のボタンがクリックされると、アプリケーションはテキストボックス内に入力されている文字列を png 形式の画像データとして共有ディレクトリへ保存する。なお、作成する画像の大きさの関係上、表示できる文字数は現在 20~25 字程度となっている。文字画像データを作成するアプリケーションとそれによって実際に作成される画像を図 4.6 に示す。

<sup>&</sup>lt;sup>1</sup>OpenCV: http://opencv.jp/

<sup>&</sup>lt;sup>2</sup>OpenGL: http://www.opengl.org/

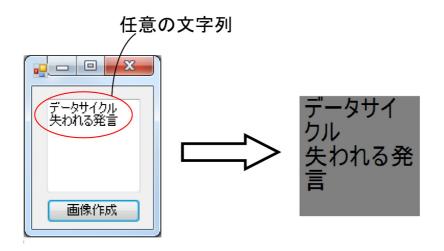


図 4.6: 文字画像データを作成するアプリケーションと実際に作成される画像

#### 4.2.2 起動時の楕円軌道と「LOAD IMAGE」アイコンの描画

データサイクルを起動するとまず、データを乗せ自動周回する楕円軌道が中央に、「LOAD IMAGE」アイコンがウィンドウの4つの頂点付近に描画される(図 4.7)。中央に描画される楕円軌道の大きさは作成されるウィンドウの大きさ(横: window\_width、縦: window\_hight)によって決まり、その長軸 2a と短軸 2b はそれぞれ、

$$2a = window\_width \times 0.6 \tag{4.1}$$

$$2b = window\_hight \times 0.8$$
 (4.2)

となっている。また、楕円は64角形で近似的に描画され、その各頂点は次の式を満たす。

$$\theta = 2\pi \times \frac{n}{64}$$

$$p_n = (a\cos\theta, b\sin\theta) \quad [n \in N, 0 \le n < 64]$$
(4.3)

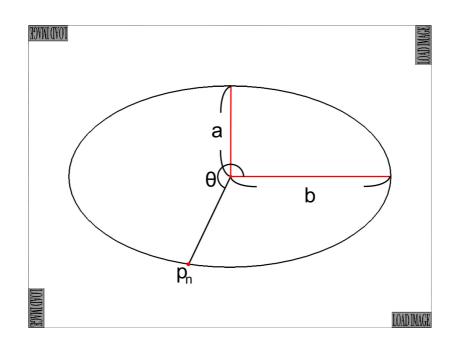


図 4.7: 楕円軌道と「LOAD IMAGE」アイコンの描画

#### 4.2.3 データの読込み

「LOAD IMAGE」アイコンがタッチされると、共有ディレクトリ内の特定の拡張子を持ったファイルを列挙する。まず共有ディレクトリ内において.jpg をもつファイルを検索し、ファイルが見つかるとそのファイル名を一度バッファにコピーする。そしてそのファイル名をもとに画像データをテクスチャオブジェクトとして格納する。このとき、テクスチャオブジェクトとして格納した画像データのファイル名を別に格納しておく。検索によって取得したファイル名は格納されているファイル名と文字列比較を行うことで、そのファイルがすでにテクスチャオブジェクトとして格納されているかを確認する。ファイル名が一致し、検索によって見つかったファイルがすでにテクスチャオブジェクトとして格納されている場合はそのファイルはテクスチャオブジェクトとして格納しない。このようにして拡張子.jpg について共有ディレクトリ内のファイル検索が終了すると、次に拡張子.png をもつファイルについても同様の操作を行う。

共有ディレクトリのファイル検索が終了すると、テクスチャオブジェクトとして格納されている画像データを描画したイメージパネルが開かれる。

イメージパネル上でタッチにより選択された画像は中心座標、中心点との角度、識別のためのラベル、移動用のフラグ、自動周回用パラメータなどの値が割り当てられ、操作領域へ描画される。このとき同時にイメージパネルは閉じられる。

「LOAD IMAGE」アイコンがタッチされてから、画像データが操作領域に描画されるまでのフローを図 4.8 に示す。

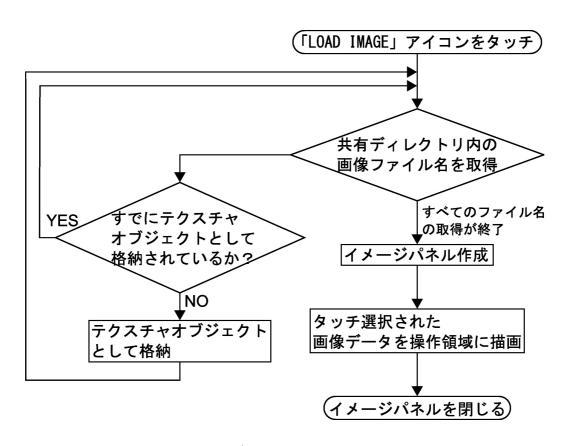


図 4.8: データ読込み時のフロー

#### 4.2.4 データの向きの決定と描画

操作領域に存在する画像データはそれぞれがもつ中心座標を用いてテクスチャマッピングにより描画される。また、描画されるデータの向きはそのデータの中心座標がどの領域に存在するかによって決定する。

#### データの中心座標が楕円軌道付近以外にあるとき

中心座標が楕円軌道付近以外にあるとき、操作領域を四分割し、その内のどの領域に存在しているかによってデータの向きを決定する。データの位置とその向きの関係を図 4.9 に示す。また、このとき同時に中心点との角度 を常に算出しておき、次に述べる楕円軌道付近にデータがあるときのデータの描画の処理に用いる。

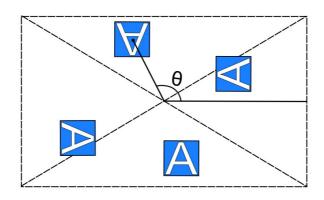


図 4.9: データの向きの例

#### データの中心座標が楕円軌道付近にあるとき

データが楕円軌道付近の領域内にある場合、データは楕円軌道上に乗り自動周回を始める。 描画されている楕円軌道に対してある値 の分だけ内側のところから だけ外側までの領域 内にデータの中心座標があるとき、データが楕円付近にあるとみなしている。すなわちデー タの中心座標(x,y)が、

$$\frac{x^2}{(a+\alpha)^2} + \frac{y^2}{(b+\alpha)^2} \le 1$$
 かつ  $\frac{x^2}{(a-\alpha)^2} + \frac{y^2}{(b-\alpha)^2} \ge 1$  (4.4)

を満たす場合に、そのデータが楕円軌道付近にあるとみなす。軌道上に乗る際には、常に算出している全体の中心点との角度 を基に描画位置を決定する。また、楕円軌道上にあるデータの向きは、全体の中心点からデータの中心座標へ向かう方向が下になるよう描画される。データの位置とその向きの関係を図 4.10 に示す。

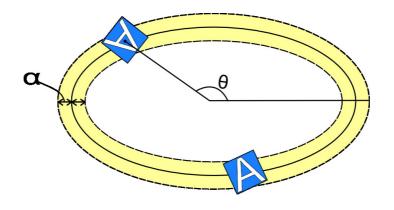


図 4.10: 楕円軌道上にあるデータ

#### 4.2.5 データの操作

描画されているテクスチャ内の特定の領域をタッチすることでそのテクスチャを選択した 状態になる。選択中のテクスチャの外枠を赤く描画することにより、ユーザはそのデータが 選択状態であることが容易に把握できる。この状態のままドラッグ操作を行うことで、ユー ザはテクスチャを自由に移動させることができる。また、選択状態にあるデータはたとえ精 円軌道上にあっても自動周回はせず、指を離して通常状態に戻ったとき、自動周回を始める。 テクスチャの通常状態、選択状態の遷移関係を図 4.11 に示す。

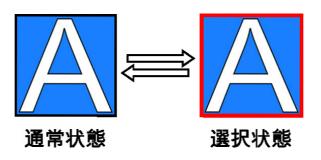


図 4.11: 通常状態と選択状態

#### 4.2.6 データの自動周回

精円軌道上に乗ったデータはそれぞれの自動周回用パラメータを用いて、中心点との角度 を右回りに徐々に増加させていく。

## 第5章 関連研究

本研究と関係する既存の研究について述べる。

### 5.1 ディスカッション支援

平島らはプレゼンテーション形式の会議において、聴衆が付箋をスライドに張るようにして即座にメモを行えるシステムを提案した[7]。このシステムではプレゼンテーションが行われているときに聴衆が抱いた疑問や質問事項、コメントを即座に残すことができ、それによってより多くの有益なコメントを収集することを可能とした。

この研究は、ある瞬間に抱いたコメントを即座に付箋を貼るようにして残すという点において本研究の失われる発言となってしまうようなインスピレーションをその瞬間に提示しようとするアプローチと類似している。これに対し、本研究では支援するディスカッションの形態としてスライドが聴衆に配布されるプレゼンテーション形式のディスカッションに限らないという点で異なる。また、平島らはコメントの収集は会議中に行い、そのコメントを会議後の非同期な場で効率的に解消することを目的としているが、本研究では、あくまでも会議中の同期的なコメントの収集とその解消を目的としている。

### 5.2 テーブル型ディスプレイで行う協調作業支援

複数人が1つのディスプレイを囲んで作業を行う際に問題となる、表示データの向きに着目した研究が多くなされている。

Chia らはマルチタッチを用いて複数ユーザが同時にドキュメントを任意に移動、または回転させる操作を可能とする DiamondSpin を提案した [8]。また、それを拡張したアプリケーションとして UbiTable [9] などがある。このアプリケーションは各ユーザが自分の情報端末を持ち寄り、その中に保存しているデータをテーブルトップ上に出現させ、注釈や印をつけたり、交換する操作を実現している。

吉原らはアプリケーションウインドウをユーザが必要に応じて容易に移動、回転、複製などを操作を行うことのできるトレイに乗せて操作することを可能にし、協調作業を行うユーザ間での情報共有支援を行った[10]。

北原らはテーブルトップの外側にレールを作成し、ユーザの操作によって動かすことでレール上に乗っているデータを動かすことができるようにし、ユーザ間でのデータ共有を支援した [11]。この手法では、移動させるときにレール上のデータがすべて移動するため、複数の

データを一括で渡すことができることや、あるデータを渡す際にそれが第三者の目にとまる ことでそのユーザにも有用な情報を与えることができる可能性が向上する利点が得られる。 これらの研究では、ユーザ間で効率よく情報共有を行うために、ユーザが主体的に表示デー

タの操作を行わなければならない。本研究ではユーザの注意が表示データの操作に向けられてしまうことでディスカッションからそれてしまうことを防ぐために、データの位置によってシステムがその向きを決定し、また情報共有のためのデータの周回を自動で行っている。これにより、ユーザが主体的に表示データに対して操作を行うことをシステムがあまり要求しない点で異なる。

## 第6章 議論

本研究で提案したシステムであるデータサイクルを実装していく中で、いくつかの問題点が明らかになった。本章ではそれらについての議論を行う。

本研究では想定するディスカッションの形態として参加する全メンバが何らかの情報端末を所持してディスカッションを行うという点を挙げたが、メンバの内の誰かが情報端末を所持していない場合も十分に考えられる。そのため手元のデバイスを使わず、テーブルトップデバイス上でもインスピレーションの具現化が行うことが可能となればデータサイクルのさらなる発展性が見込まれる。

また、データサイクルによって生み出される新たなインスピレーションがそのときのディスカッションの論点に合わず、データサイクル上に具現化される場合も考えられる。そのため、データサイクルにおいて具現化されているデータ同士の関連付けを表現することはさらなる効率的なインスピレーションの共有を可能にすると考える。

また、本研究ではインスピレーションを具現化する際にそれに関連する画像か文字情報を用いた。しかし、具現化するインスピレーションによっては画像あるいは文字情報単独では伝わりきらないこともある。この問題はデータサイクル上に表示されているデータに対して付加情報を追加することで解決できると考える。例えば手元のデバイスで用意し、データサイクル上へ読み込んだ画像データに対して自由に書き込みを行えるようにすることで、よりそのインスピレーションがどういったものなのかを提示することができる。また、自分のもとへやってきたデータに対してそこから受けたインスピレーションを付加情報として書きこむことも可能となる。

最後に、ディスカッションが長く続いていくにつれ、データサイクル上に表示されるデータの数も多くなっていく。そのためディスカッションがある程度進んだところで、具現化されたデータで埋め尽くされてしまう。こうなってしまうとデータ共有の際に、ユーザは大量に表示されているデータの中からどのデータがすでに自分が閲覧済みのものか判断しなければならなくなってしまう。判断するべきデータの数が増加するほどユーザの注意はその判断に向けられてしまい、ディスカッションからは離れてしまう。そのため、時間経過によるデータの描画方法の変化が必要であると考える。例えば、選択されなくなってから一定時間経過したデータの表示サイズを小さくすることや、楕円軌道の内側に向かっていくようにすれば解決できると考える。

## 第7章 結論

本研究ではディスカッションを円滑に進める上で、個人の中で制限されてしまい、ディスカッションの場に現れることのない発言を失われる発言と定義し、それを減少させることでグループで行う作業において必要不可欠となるディスカッションの支援を目的とした。そのために、あるメンバが受けたインスピレーションを即座に具現化し、他のメンバへとの共有を行うためのシステムとしてデータサイクルを提案し、実装を行った。また情報共有を行う際も、その行為によってディスカッションから注意が大きくそれてしまうことのないようなさりげない提示方法を提案した。これについては、データを自動周回させること、及びデータをユーザにとって見やすい方向に向けるという手法を実現した。

今後は、本研究で想定したディスカッション形態以外のものも含め、様々なディスカッションにおいてデータサイクルを導入し、その客観的な有効性の評価を行いたい。

## 謝辞

本論文を執筆するにあたり、指導教員である田中二郎先生をはじめ、志築文太郎先生、三末和男先生、高橋伸先生には多くの有益な助言、適切なご指導をいただきました。特に、NERFチームの担当である田中二郎先生には日頃から細やかな助言をいただきました。心より感謝を申し上げます。

最後に、心身ともに多大な支援をいただきました田中研究室の皆様には深く感謝いたします。 また、ここまで私を支えてくださいました家族・友人にも心から感謝いたします。

## 参考文献

- [1] 松下温. グループウェアの実現のために. 情報処理学会論文誌, Vol. 34, No. 8, pp. 984–993, 1993-08-15.
- [2] 緒方広明. グループウェア・CSCW の研究動向. 教育システム情報学会誌, Vol. 15, No. 2, pp. 102–104, 1998-07-01.
- [3] 高橋誠. 会議の進め方. 日本経済新聞出版社, 第2版, 2008.
- [4] 杉山公造, 永田晃也, 下嶋篤. ナレッジサイエンス 知を再編する 64 のキーワード. 紀伊國屋書店, 2011.
- [5] Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, UIST '05, pp. 115–118, 2005.
- [6] 内藤真樹, 田中二郎, 志築文太郎, 三末和男, 高橋伸. 卒業論文, 筑波大学, 2007.
- [7] 平島大志郎, 土井健司, 勅使河原可海. プレゼンテーション形式の会議のための効率的なディスカッション支援システムの提案. 情報科学技術フォーラム一般講演論文集, Vol. 6, No. 4, pp. 171–172, 2007-08-22.
- [8] Chia Shen, Frédéric D. Vernier, Clifton Forlines, and Meredith Ringel. Diamondspin: an extensible toolkit for around-the-table interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pp. 167–174, 2004.
- [9] Chia. Shen, Katherine. Everitt, and Kathleen. Ryall. UbiTable: Impromptu face-to-face collaboration on horizontal interactive surfaces. In *UbiComp 2003: Ubiquitous Computing*, pp. 281–288. Springer, 2003.
- [10] 吉原正樹, 荒木博文, 中島誠, 伊藤哲郎. テーブル型ディスプレイ上での協調作業を支援するためのユーザの離合集散に対応できるトレイの実現. 情報科学技術フォーラム一般講演論文集, Vol. 6, No. 3, pp. 403–406, 2007.
- [11] 北原圭吾, 井上智雄, 重野寛, 岡田謙一. 協調学習支援を目的としたテーブルトップインタフェース. 情報処理学会論文誌, Vol. 47, No. 11, pp. 3054-3062, 2006-11-15.