

Window の監視機能を利用した  
遠隔地型 CAI システム

茨城大学 工学部 システム工学科

市井 剛裕 91T6010T

新納 浩幸 (担当教官)

平成7年3月3日

# 目次

1	はじめに	1
2	CAIについて	3
2.1	CAI . . . . .	3
3	遠隔地型 CAI システム	4
4	Window の監視と talk	6
4.1	ウィンドウによる監視機能 . . . . .	6
4.2	talk . . . . .	7
5	X ウィンドウシステム	9
5.1	サーバクライアント . . . . .	9
5.2	ソフトウェア階層 . . . . .	10
6	作成したシステム	12
6.1	全体の構成 . . . . .	12
6.2	個々の Shell ウィジェットの機能 . . . . .	12
6.2.1	補助プログラムの説明 . . . . .	15
6.3	学習方法 . . . . .	16
6.3.1	生徒側の画面 . . . . .	16
6.3.2	教師側の画面 . . . . .	21

6.4	XREC	24
7	実験及び考察	26
7.1	XCAI の特徴	26
7.2	グループウェアとして見た遠隔地型 CAI	26
7.2.1	グループウェア	26
7.2.2	情報の流れの研究 (3 層構造)	27
7.3	欠点の解決策	28
8	まとめ	30

# 目次

3.1	遠隔地型の CAI . . . . .	5
4.1	ウィンドウによる監視機能 . . . . .	7
4.2	talk . . . . .	8
4.3	ウィンドウの監視による生徒一覧の獲得 . . . . .	8
5.1	Xt インタリシクスに基づくアプリケーションのソフトウェア階層 . . . . .	10
6.1	生徒側の画面 . . . . .	17
6.2	問題画面及び演習用画面 . . . . .	18
6.3	トーク用のウィンドウ . . . . .	19
6.4	呼びだしボタン . . . . .	20
6.5	教師側画面 . . . . .	22
6.6	XREC の実行結果 . . . . .	25
7.1	情報伝達の三段構造 . . . . .	28

# Chapter 1

## はじめに

従来、コンピュータを利用するといえば、ワープロとして使うとか複雑な計算を解くといった様な個人の手助けのためだけに使われる事が主であった。しかし現在までの技術の進歩によりコンピュータの性能は飛躍的に良くなってきている。また、高速回線などの通信技術も進歩してきている。それに伴い現在では個人の手助けとしての利用方法に加え、例えば通信や電子メールといったようなコンピュータを人と人とのコミュニケーションの橋わたしの道具として使う事が簡単に出来る様になってきた。また最近の都市事情によるオフィスの分散化などにより同じ会社の間でさえ、顔を知らない人がいるといった状況になってきている。こういった環境では昔に比べて人どうしのコミュニケーションが希薄になってしまうので、新しいコミュニケーションの取りかたを考えることが重要である。

そこで本研究では通信技術とコンピュータを用いて、遠隔地型 CAI システムを作成する。遠隔地型 CAI システムは従来の CAI システムと比べ、教室に集まるための時間や授業のための場所を用意する必要がないといった利点がある。しかし遠隔地型 CAI システムには、生徒の状況の把握が難しい、他者とのコミュニケーションがとりづらいなどの問題がある。これらの問題を解決するために、本研究ではウィンドウによる監視機能および、talk コマンドによる双方向の会話機能を取り入れた遠隔地型 CAI システムを作成する。ウィンドウによる監視機能とは他人の画面のウィンドウを自分の画面のウィンドウで監視する機能である。また talk とは双方向の文字による通信のやりとりができる、unix のコマンドであ

る。これらを用いて生徒の学習状況の把握や生徒との双方向のコミュニケーションがとれるようになる。

# Chapter 2

## CAI について

### 2.1 CAI

CAIとは(Computer Aided Instruction)の略であり本来ならば教師が教科書なり、黒板なりを用いて行なう教育を教科書や黒板の代わりにコンピュータを用いて行なうものである。CAIの長所としてはコンピュータを用いることにより問題の作成が容易になるという点や授業に音声や動画が使えるという点がある。具体的なCAIの例としては、計算問題をコンピュータがランダムに作成し、それを生徒に解かせて解答があっているかどうかを自動的にチェックする。そして問題の理解度やかかった時間などをコンピュータに記録しておいて、そのデータを生徒の学習状況の把握や教材改良の参考につかう、といったものがある。またコンピュータのもつ特性をいかして授業に音声や動画などを用い、生徒の学習の手助けとするCAIの利用方法もある。例えば理科のCAIでは銀河系の様子を色々な大きさで表示するなどしている。また音楽のCAIでは実際にコンピュータの演奏させて見本とすることもできる。このようにCAIは従来までにはない新しい授業方法を提供するものであり、昔から研究されてきている。

## Chapter 3

### 遠隔地型 CAI システム

遠隔地型 CAI システムは、通信技術とコンピュータを用いることにより教師と生徒がはなれた場所においても授業を行なえるシステムである。(図 3.1 参照) 遠隔地型 CAI システムの具体的な例としては大手の進学塾が行なう衛星通信を用いた人気講師による全国一斉授業などがある。これはある講師の講義をビデオや衛星通信を用いることにより生徒がどこにいても聞けるようにしたものである。

このシステムには、教師と生徒が同じ場所に集まる必要がなく、手元にシステムに参加するための設備だけがあればいいので移動に要する時間の節約になる、教室を用意する必要がない、などの利点がある。これにより従来までの CAI の問題であった授業のための場所と機材が必要という点が解決できる。また従来の授業を遠隔地で行なっているだけなので、従来までの CAI のように教材を特別に用意する必要がない [8][9]。

しかし遠隔地型 CAI システムには、教師からは生徒の様子がわからない、教師から生徒への片方向の通信にならち、生徒が同じ場所にいるとは限らないので生徒間での教え合いが難しくなる、などの問題点がある。例えば先ほどの進学塾が行なう授業などでは生徒は講師の話聞くだけになったり、講師はテレビカメラに向かって講義を行なうだけになる [10][7]。

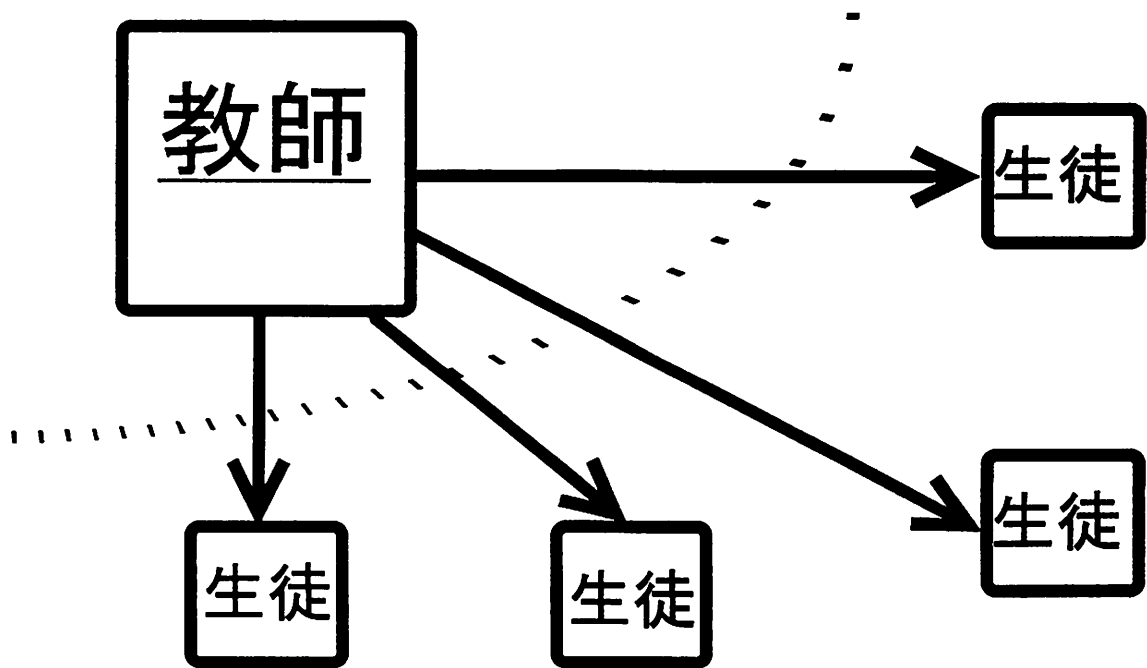


Figure 3.1: 遠隔地型の CAI

# Chapter 4

## Window の監視と talk

そこで前述の問題に対処するため、本研究で作成するシステムにウインドウによる監視機能と unix のコマンドである talk を用いる。

従来までの遠隔地型 CAI システムでは教師は生徒の演習画面を見ることができなかつたために生徒の状況を把握することが難しかった。そこでウインドウによる監視機能を用いることにより教師は生徒の画面を監視することができるようになる。これにより生徒の状況を把握することができるようになる。

また talk を用いることにより教師と生徒または生徒同士の双方向の通信のやりとりができるようになり、コミュニケーションが取りやすくなる。以下にそれぞれについて説明する。

### 4.1 ウインドウによる監視機能

ウインドウによる監視機能はある画面のディスプレイ・サーバへの表示要求を `xwatchwin` というプログラムをもちいて監視し、監視したい画面のディスプレイ・サーバへ同じ要求を出すことで実現されている。(図 4.1 参照) この機能を用いることによって、ある画面に写っている画面のウインドウをどの画面のウインドウでも見るようになるので教師が生徒の画面を監視することができたり、通常は特定の人にしか分からない情報を

全員に知らせることができたりする。

本研究ではウィンドウによる監視機能は教師が生徒の演習画面を監視するために用いている。また教師画面に表示されている、システムに参加している生徒一覧を、生徒が生徒側の画面で監視するためにも用いている。

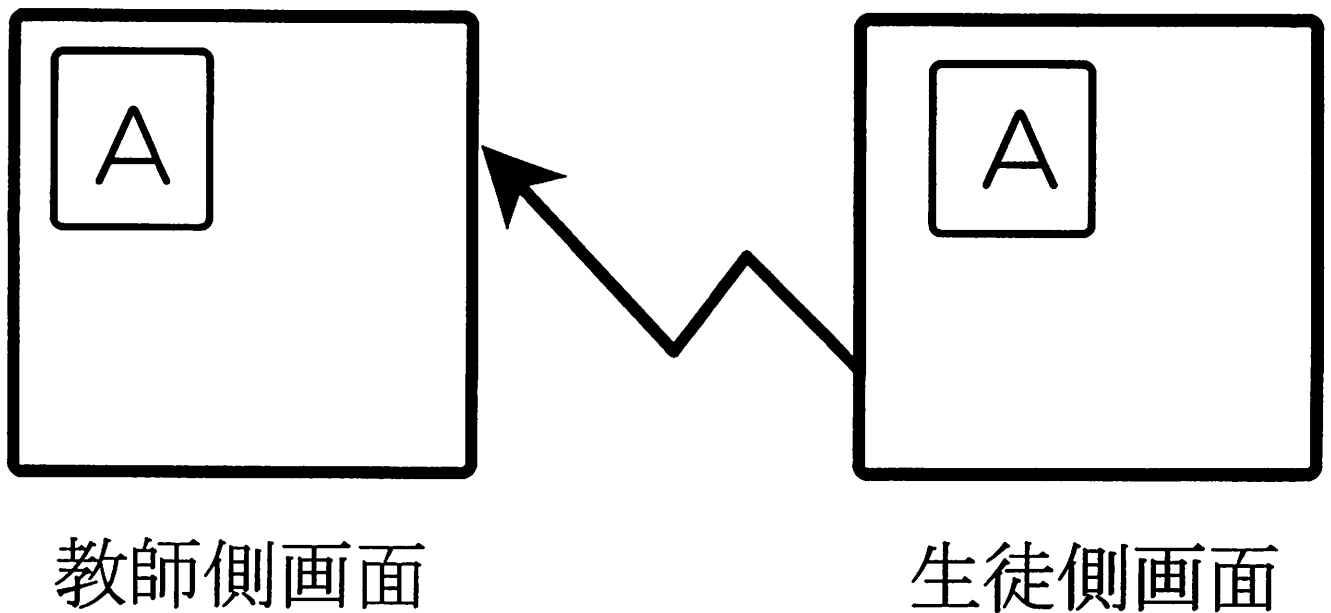


Figure 4.1: ウィンドウによる監視機能

## 4.2 talk

talk(以下トーク)は文字による双方向の通信を行なうためのunixのコマンドである。ここでトークの実行方法について説明する。まずトークを行なうにはトークを行なう相手のログイン名と使用しているマシン名を指定しなければならない。これは電話をかけるのに相手の電話番号を知っていなければならないのと同じである。

実際にトークを行なうには例えばログイン名がichii、使用しているマシンがlilyの人とトークを行なうには、コマンドライン上でtalk ichii@lily.dse.ibaraki.ac.jp と入力する。す

るとトークをかけられた人の画面には、トークがかかってきたという合図と相手のログイン名と使用しているマシン名が表示されるので、トークをかける時と同じようにこちらからも talk [ログイン名]@[マシンアドレス] と入力する。するとお互いの画面が図 4.2 に示すような画面に切り替わりトークを行なうことができる。

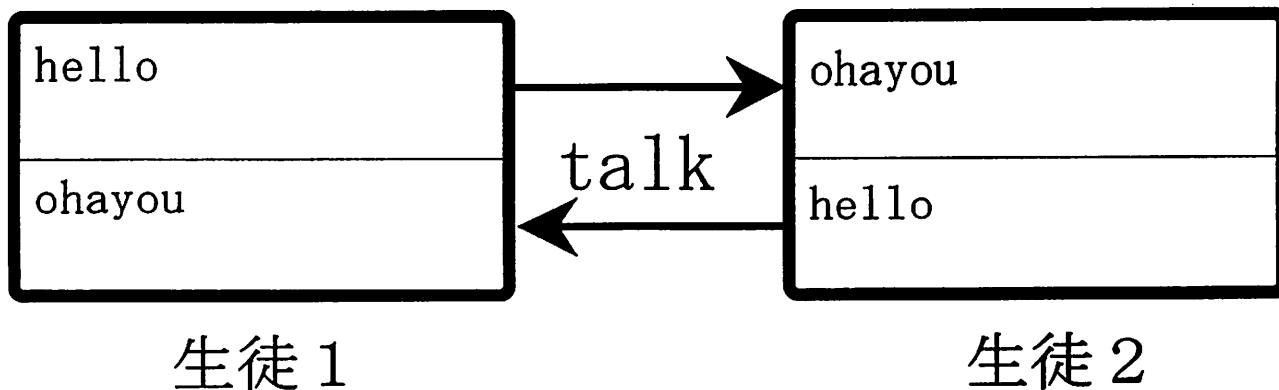


Figure 4.2: talk

本研究ではトークによる会話を教師と生徒の間だけではなく生徒間でも行なえるようにする。ただ生徒は現在他の誰が XCAI の参加中かが分からないのでこのままではトークの相手が指定できない。そこでこの問題を解決するのにもウィンドウによる監視機能を利用する。教師は現在誰が XCAI に参加しているのかを知ることができるので、生徒一覧を表示したウィンドウを教師側のディスプレイに表示する。生徒は生徒側の画面のウィンドウでこの一覧を監視することによりトークの相手が指定できる。(図 4.3 参照)

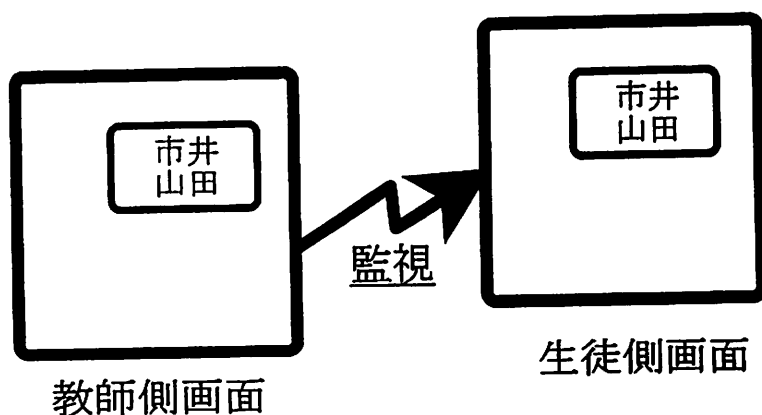


Figure 4.3: ウィンドウの監視による生徒一覧の獲得

## Chapter 5

# X ウィンドウシステム

本研究の遠隔地型 CAI システムは X ウィンドウシステムを用いて作成した [1]。X ウィンドウシステムはサーバクライアント方式を採用しており X プロトコルに基づいて作られているので、どのようなハードウェアや OS に対しても対応できるシステムになっている。これにより異なった環境の人同士が通信を行なうさいにもわざわざ同じ環境にする必要がなくなる。これは X のアプリケーションの唯一の必要条件は X のプロトコルの仕様に従った X プロトコルメッセージの生成と受信であるからである。このような理由で X ウィンドウシステムを用いて遠隔地型 CAI システムを作成することにした。

### 5.1 サーバクライアント

X ウィンドウシステムはあるマシンでプログラムを実行し別のマシンで表示できるようにするために、2つのプロセス間のネットワークプロトコルとして設計されている。これらのプロセスの一方がクライアントと呼ばれるアプリケーションプログラムで、他方がディスプレイハードウェア、キーボード、ポインタなどを制御するサーバである。

実際にはユーザはサーバを実行するマシンの前で作業を行ない、何かを実行する時にはクライアントにイベントと呼ばれるメッセージを送る。するとクライアントはサーバから送られてきたイベントを解釈し、対応する処理を実行し、リクエストと呼ばれるメッセー

ジをサーバに送る。このような方式をサーバクライアント方式といい、この方式はいかに上げる理由により通信を行なうのに適した方式である。

- ・組み込みグラフィックス機能を持たない強力なシステムにグラフィックス出力を提供できる。

- ・サーバクライアント方式による分散処理はシステム負荷の不均衡という問題の解決に役立つ。

## 5.2 ソフトウェア階層

本研究では X<sub>t</sub> イントリンシクスライブラリと一連のウィジェットを使って、遠隔地型 CAI システムを作成する。X はひとつの言語やオペレーティングシステムなどに制限されていないため、この方法は X のアプリケーションを作成する方法のひとつにすぎない。しかしこの方法が最も一般的であるので、X<sub>t</sub> イントリンシクスライブラリとウィジェットを用いることにする。図 5.1 に、X<sub>t</sub> イントリンシクスとウィジェットセットを使用するアプリケーションのソフトウェア階層を示す。

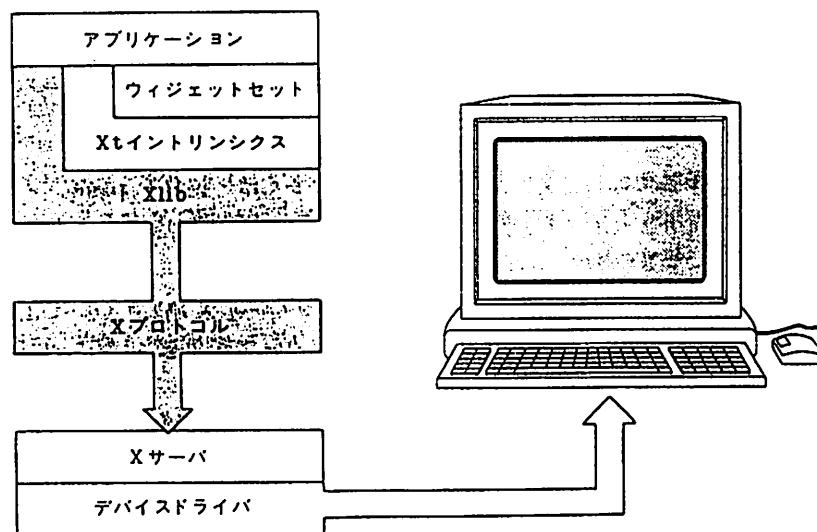


Figure 5.1: X<sub>t</sub> イントリンシクスに基づくアプリケーションのソフトウェア階層

Xlib を使えば、X プロトコルの機能すべてを利用できるが、プログラミングは X ツールキットに比べ簡単ではない。また Xt は Xlib を基礎としており、その目的はウィジェットと呼ばれるユーザインターフェースを使用可能にするためにオブジェクト指向の層を提供することである。ウィジェットセットは、一般に使用されるユーザインターフェース構成要素を提供するいろいろなウィジェットの集まりで、一貫した概観とユーザインターフェースを持つ。

# Chapter 6

## 作成したシステム

本研究で作成したシステムをここでは XCAI と呼んでいる。XCAI は X ウィンドウシステムを用いて作成されている。

### 6.1 全体の構成

XCAI は `text`、`next`、`question`、`ans`、`number`、`quit`、`end`、`check`、の 8 つの Shell ウィジェットで構成されている。XCAI の補助的なプログラムとして `xlist`、`xnlist`、`xrec`、`fxcai.sh` の 4 つのプログラムがある。また演習画面とトーク用にあらかじめ `kterm` を 3 つ用意している。

### 6.2 個々の Shell ウィジェットの機能

- `text`

生徒側画面に問題文を表示するウィンドウを作成する。

このウィンドウをクリックしても何もおこらない。また `label` を用いて問題文を表示している。`label` とはウィンドウに表示される文字列のことである。

- `next`

生徒側画面にネクストボタンを表示するウィンドウを作成する。

このウィンドウをクリックすると Next と Number1 と Number2 の 3 つの関数をコールバックする。

- Next

次に表示される問題の管理を行なっている。XCAI がスタートすると変数  $i$  を 1 に設定する。そして Next が呼ばれるたびに  $i$  を +1 していき  $i$  の値に対応した問題文を text ウィジェットに表示させている。またネクストボタンが押された時間を XREC というプログラムに知らせている。

- Number1

呼ばれるたびに  $k$  を +1 していき  $k$  の値に対応した番号を number ウィジェットに表示させている。

- Number2

number ウィジェットに表示させる番号が通常のものが特殊なものかをチェックしている。具体的には  $k$  の値が 5 と 12 の時は number ウィジェットに Now Reading の表示をさしており、 $k$  の値が 17 の時は Send 5mail、2 の時は Send 1-4mail の表示をさせる。

● question

生徒側の呼びだしボタンを表示するウィンドウを作成する。

このウィンドウをクリックすると Call1 と Call2 の関数をコールバックする。

- Call1

呼ばれるたびに ans ウィジェットのウィンドウの色を反転させている。

- Call2

呼ばれるたびに question ウィジェットのウィンドウの色を反転させ、label が ques1 なら Wait!、Call! なら ques1 に変更する。

- ans

教師側の呼びだしボタンを表示するウィンドウを作成する。

このウィンドウをクリックすると Call3 と Call4 の関数をコールバックする。

- Call3

呼ばれるたびに ans ウィジェットウィンドウの色を反転される。

- Call4

呼ばれるたびに quistion ウィジェットウィンドウの色を反転させ、label が ques1 なら Call!、Wait! なら ques1 に変更する。

- number

教師側画面に問題番号を表示するウィンドウ作成する。

このウィンドウをクリックしても何もおこらない。

- quit

生徒側画面に終了ボタンを表示するウィンドウを作成する。

このウィンドウをクリックすると Quit2 の関数をコールバックする。

- Quit2

end ウィジェットの label を Click for quit に変更し、end ウィジェットを sensitive にする。

sensitive にするとはそのウィンドウをクリックできる状態にすることである。

- end

教師側画面に今生徒が学習中がどうかを表示するウィンドウを作成する。

このウィンドウをクリックすると sensistive の時に限り Quit3 の関数をコールバックする。

- Quit3

elist に生徒のログイン名を登録し、XCAI を終了する。elsit とは終了を選択した生徒一覧を記録したファイルであり xnlist の作成に用いられる。

● check

教師側画面に生徒の使用するディスプレイ番号を表示する。

このウィンドウをクリックすると Touroku1 と Touroku2 の関数をコールバックする。

- Touroku1

check をウィジェットを破壊する。

- Touroku2

nlist に生徒のログイン名を登録する。nlist とは XCAI を開始した生徒一覧を記録したファイルであり xnlist の作成に用いられる。

### 6.2.1 補助プログラムの説明

● xlist

nlist に記録されているのは各生徒が XCAI を実行した回数であり、elist に記録されているのは各生徒が XCAI を終了した回数であるので、nlist と elist をくらべることによってその生徒が現在 XCAI を実行中かどうか知ることができる。xlist では nlist と elist から 現在 XCAI を実行中の生徒一覧を登録しておく n-elist を作成している。

● xnlist

xlist により作成された n-elist をもとにして教師側画面に現在 XCAI に参加している生徒一覧を表示するプログラムである。

● xrec

各生徒がどの問題にどのくらい時間をかけたかを自動的に記録している。

- fxcai.sh

XCAI スタート用プログラムである。生徒は XCAI を実行するときにはまずこのプログラムを実行する必要がある。

## 6.3 学習方法

授業内容はコンピュータのアプリケーションを学習するものにする。これはコンピュータのアプリケーションを学習しようと思えばコンピュータ上で演習を行なう必要があるの  
で算数や国語などを学べるものよりもこのシステムに適切だからである。また本研究では Emacs 上のメイラー (Rmail) によるメールの利用方法を学習対象とした [2]。Rmail というのは Emacs 上で送られてきたメールを管理したり、メールに返信をだしたりすることができるメイラーである。生徒は XCAI 用に作成されたテキストを併用して学習を行なう。(テキストは付録参照) XCAI では教師専用の画面と生徒専用の画面をそれぞれ用意している。以下に教師側と生徒側のそれぞれについて説明する。

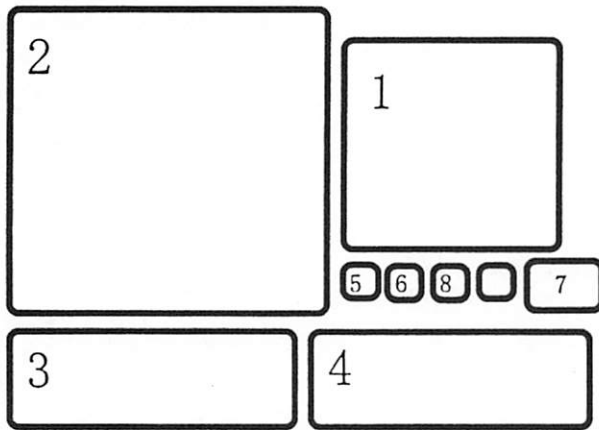
### 6.3.1 生徒側の画面

XCAI の使用方法を説明する。まず生徒は、XCAI スタート用のプログラムを実行する。そして画面の指示に従い、

- 自分の名前
- 自分の使用するディスプレイの番号
- 自分の使用するマシン名
- 教師の使用するディスプレイの番号

を入力する。

そしてしばらく待つと、教師側の画面に今このシステムに参加している生徒のリストが表示されるので生徒はそのリストを監視することにより、今誰がこのシステムに参加しているかが分かる。生徒側の画面を図 6.1 に示す。



1. 問題画面
2. 演習用画面
3. 教師とのトーク用  
ウィンドウ
4. 他の生徒とのト  
ーク用ウィンドウ
5. ネクストボタン
6. 呼び出しボタン
7. 生徒一覧
8. 終了ボタン

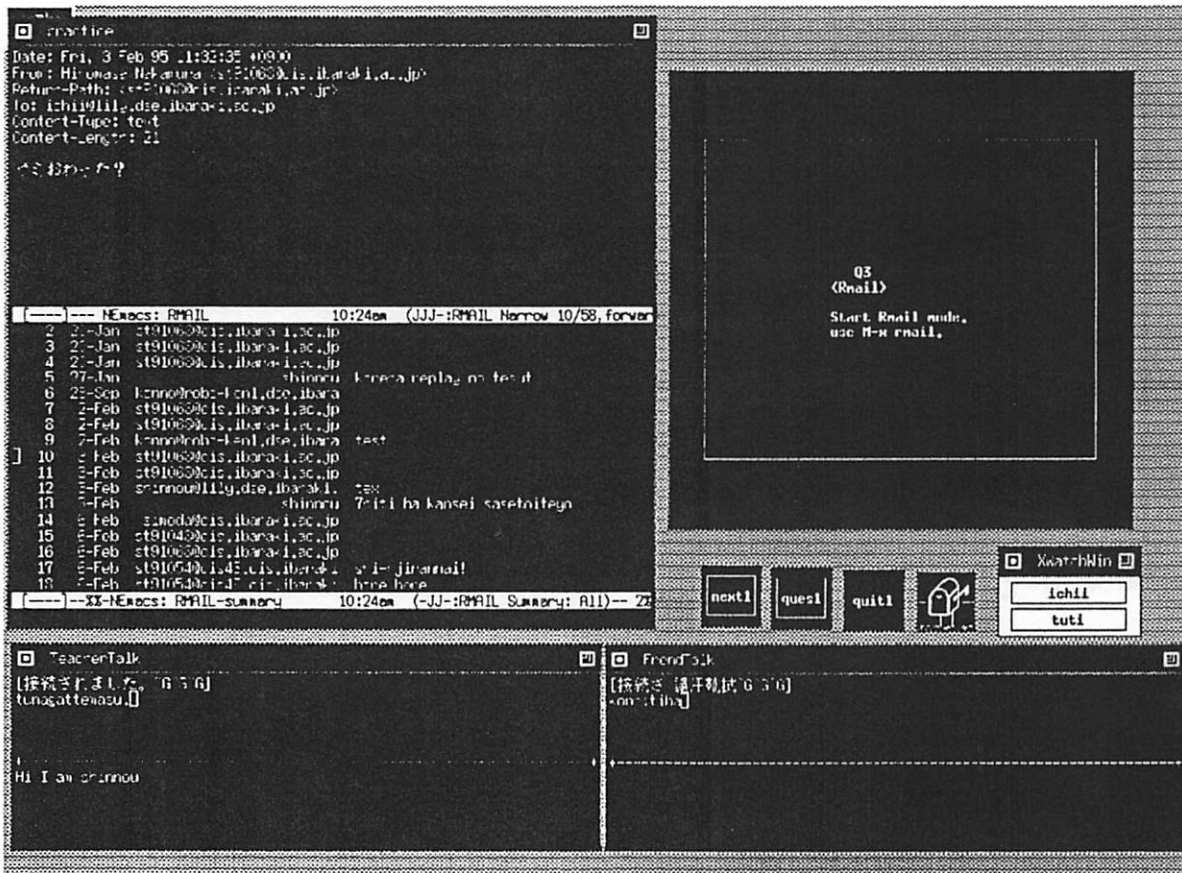


Figure 6.1: 生徒側の画面

## 1. 問題画面

このウィンドウには、問題が表示される。(図 6.2 参照) また「テキストを読みなさい」や、「教師からのメールを待ちなさい」などの指示も表示される。この問題や指示は実際の画面では Now Reaing などのように英語で表示される。これは X ウィンドウシステムにおいて日本語を画面に表示させる方法が分からなかったためである。生徒は基本的にはこのウィンドウの指示に従いながら演習用画面 2. を持ちいて学習を進める。またネクストボタン 5. を押すことにより、次の問題に進むことができる。

## 2. 演習用画面

生徒は、このウィンドウ上で、実際の演習を行なう。(図 6.2 参照) 具体的には、本研究で学習に選んだ題材が Rmail の利用方法であるのでこのウィンドウで Rmail を実行する。

またこのウィンドウが教師によって監視されているウィンドウであり、教師はこのウィンドウによって、生徒の学習状況を知ることが出来る。

```
From: [redacted]
From: Hironaka Hironaka <h1062@ic.ibaraki.ac.jp>
Return-Path: <h1062@ic.ibaraki.ac.jp>
To: ichi@illy.dco.ibaraki.ac.jp
Content-Type: text
Content-Length: 21

ピミおわたた?

----- NEAcas: RMAIL 10:25am (JJJ-RMAIL Narrow 10/58, forward)
 2 21-Jan <h1062@ic.ibaraki.ac.jp>
 3 21-Jan <h1062@ic.ibaraki.ac.jp>
 4 21-Jan <h1062@ic.ibaraki.ac.jp>
 5 27-Jan shinnou <crena repia no tesut>
 6 28-Sep <konno@ico-ken1.dco.ibaraki.ac.jp>
 7 28-Feb <h1062@ic.ibaraki.ac.jp>
 8 22-Feb <h1062@ic.ibaraki.ac.jp>
 9 22-Feb <konno@ico-ken1.dco.ibaraki.ac.jp> test
10 22-Feb <h1062@ic.ibaraki.ac.jp>
11 22-Feb <h1062@ic.ibaraki.ac.jp>
12 22-Feb <shinnou@illy.dco.ibaraki.ac.jp> test
13 22-Feb shinnou <Truti ha kansai sasetsuiteyo>
14 6-Feb <shinnou@ic.ibaraki.ac.jp>
15 6-Feb <h1062@ic.ibaraki.ac.jp>
16 6-Feb <h1062@ic.ibaraki.ac.jp>
17 6-Feb <h1062@ic.ibaraki.ac.jp>
18 6-Feb <h1062@ic.ibaraki.ac.jp>
----- NEAcas: RMAIL-summary 10:25am (JJJ-RMAIL Summary: All) 22
```

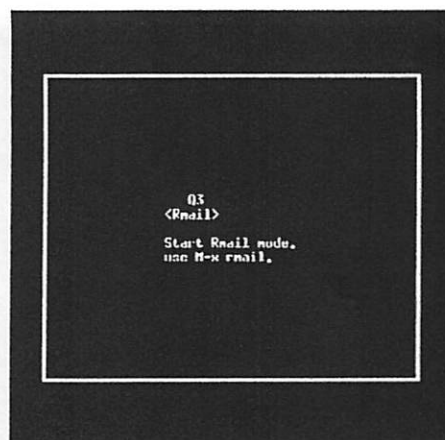


Figure 6.2: 問題画面及び演習用画面

### 3. 教師とトークするためのウィンドウ

生徒は、このウィンドウを使うことによって、教師に直接質問をしたり、教師の指導を受けたりすることができる。(図 6.3 参照) また授業開始後、一度でもトークコマンドを実行しておけば、最後までトークがつながったままになる。これはトークを行なうたびにコマンドを入力しなおすのは手間がかかるので、トーク用のウィンドウをあらかじめ開いておくことで再入力の手間を省けるようにした。

しかし、教師は複数の生徒を相手にしている可能性があるので生徒の呼びかけに対して気づいてくれない可能性がある。そこで生徒は呼びだしボタン 6. を用いることによって教師を呼び出す必要がある。

### 4. 他の生徒とトークするためのウィンドウ

このウィンドウをつかうことによって生徒は、現在 XCAI を利用中の他の生徒と、トークを行なうことができる。(図 6.3 参照) これは教師には聞きづらいことや、他の生徒に相談したいと思った時に利用することができる。

現在誰がこのシステムに参加しているかというのは教師側に表示されている、現在参加している生徒のリスト 7. を監視することにより知ることができる。

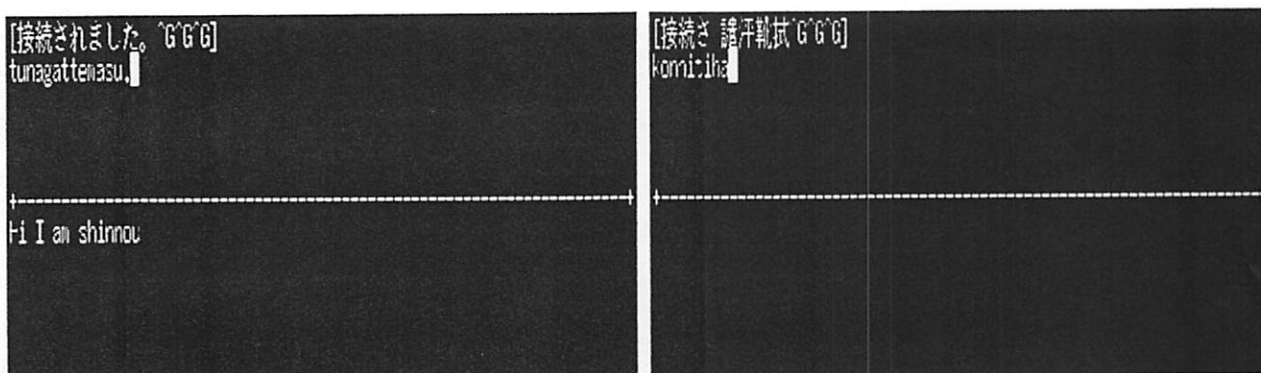


Figure 6.3: トーク用のウィンドウ

## 5. ネクストボタン

生徒は問題画面に表示されている指示を終えた時にこのボタンを押すと、次の問題に進むことができる。このボタンは教師側画面の問題番号表示ウィンドウ10.と連動しており、番号をひとつ進める。

また一回ネクストボタンを押してから次にネクストボタンを押すまでの時間を計っておき、XREC という学習時間管理プログラムに利用している。

## 6. 呼びだしボタン

生徒は呼びだしボタンを押すことによって教師に合図を送ることができる。このボタンは通常は ques と表示されているが、こちらからボタンを押すと wait! と表示されウィンドウの色が変わり、教師からの合図を待つ状態になる。そして wait! のメッセージが ques に戻りウィンドウの色がもとに戻ったら教師がこちらの合図を確認したということが分かり、トークを開始することができる。(図 6.4 参照)

また教師からこちらに合図が送られてくることがある。この場合はウィンドウには call! というメッセージが表示される。その時には呼びだしボタンをこちらから押し教師に合図を確認したことを知らせる。

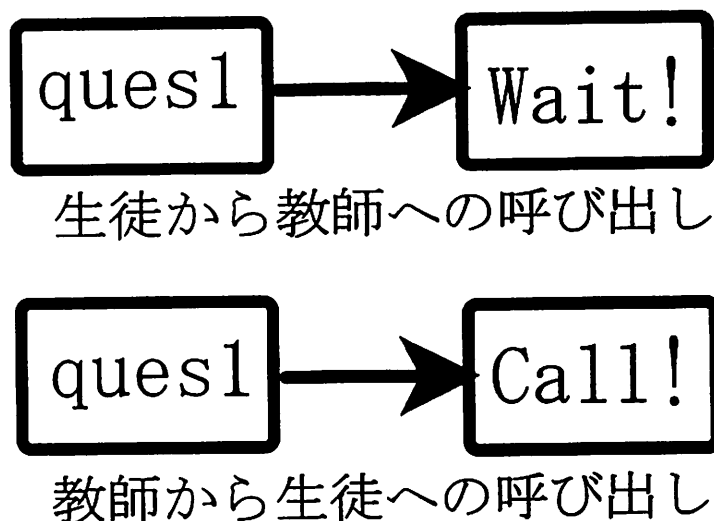


Figure 6.4: 呼びだしボタン

## 7. 現在参加している生徒のリスト

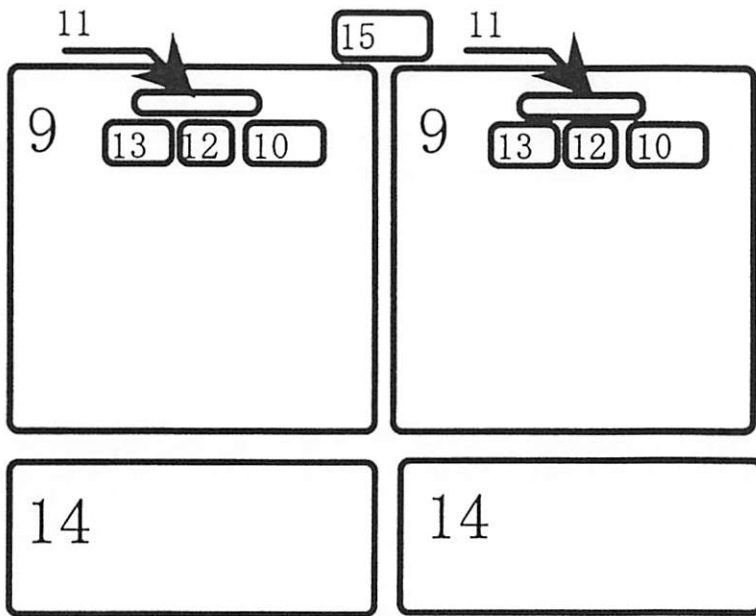
生徒は他のこのシステムに参加している生徒とトークを行なおうとする時には相手のログイン名と使用しているマシン名を知る必要がある。しかし通常はその情報を教師は得ることができるが生徒の方は得ることができない。そこで生徒は教師側の画面に表示されている、生徒一覧を生徒側の画面で監視することによってその情報を得ることができるようになる。

## 8. 終了ボタン

生徒はいつでもこのボタンを押すことによって XCAI を終了することができる。

### 6.3.2 教師側の画面

生徒が XCAI を実行すると教師の方では生徒の名前、使用するマシン名、ディスプレイ番号の3つの情報が得ることができるので、教師は XCAI スタート用のプログラムを実行し、画面を整理する。そして xnlist というプログラムを用いることにより教師側の画面に最新の生徒一覧を表示する。XCAI では生徒は二人まで同時に参加できるので一人目の生徒を画面の左、二人目を右に決めておく。教師側の画面を図 6.5 に示す。



- 9. 生徒の演習用画面
- 10. 問題番号表示  
ウィンドウ
- 11. 生徒のディスプレイ  
番号
- 12. 生徒のマシン名
- 13. 呼び出しボタン
- 14. トーク用ウィンド  
ウ
- 15. 生徒一覧

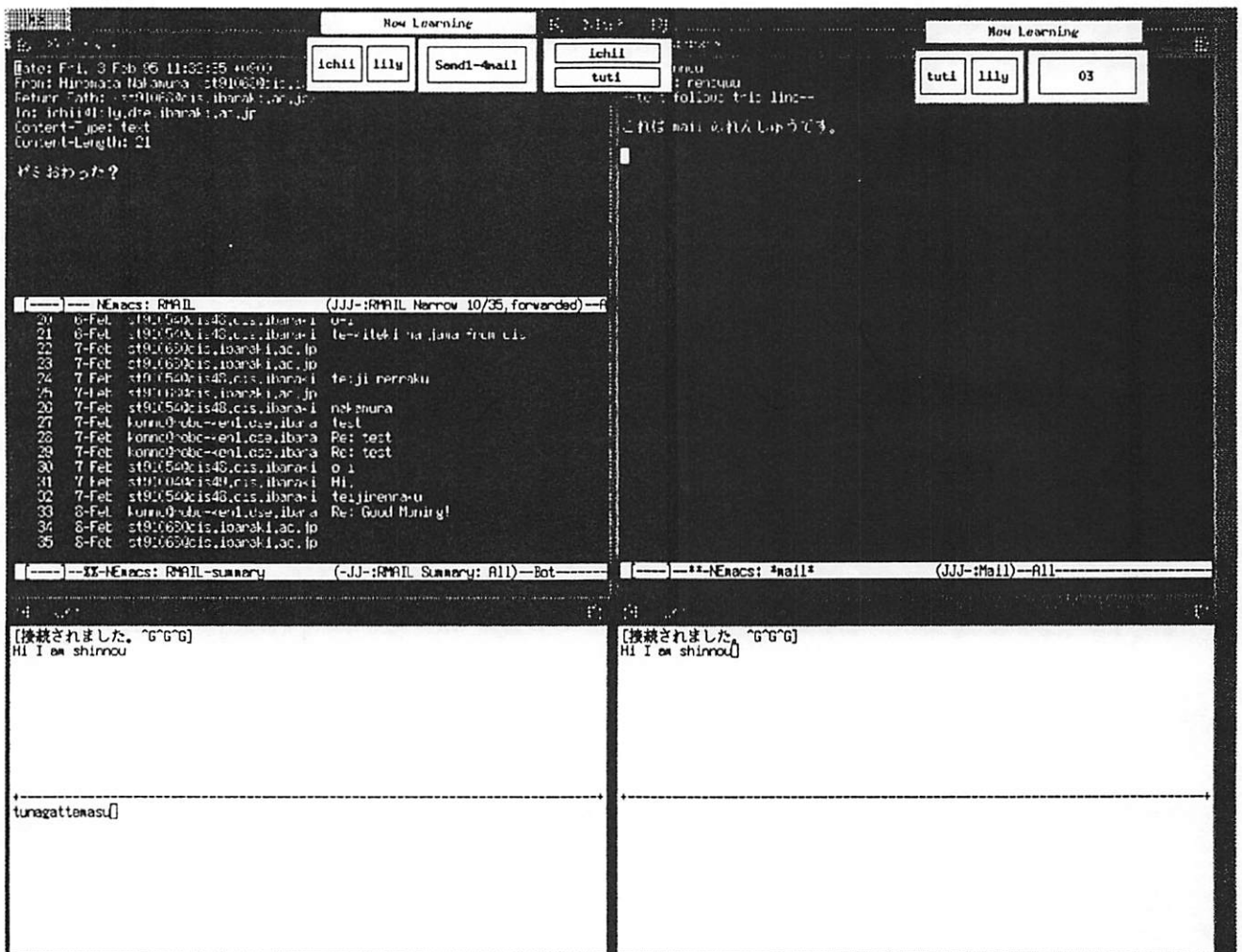


Figure 6.5: 教師側の画面

## 9. 生徒の演習用画面

教師はこのウィンドウを用いて生徒の演習用画面 2. を監視しており、これにより生徒の学習状況を把握できる。そして指導の必要などを感じた時は、トークを用いて生徒と会話をすることができる。

## 10. 問題番号表示ウィンドウ

このウィンドウには通常、生徒が今何番の問題を学習しているかが表示されている。このウィンドウは生徒側画面のネクストボタン 5. と連動しており、生徒がネクストボタンを押すたびに次の表示に切り替わる。

また生徒がテキストを読んでいるあいだは Now Reading と表示され、生徒がテキストを読んでいるということが分かる。この間は生徒は演習用画面を用いないので他の生徒に集中することができる。そして教師が生徒に対して何かしなければならない時には Send 1 mail などのメッセージが表示される。教師は指示が表示されるとその指示を実行する。

## 11. 生徒の使用しているディスプレイの番号

現在生徒が使用しているディスプレイの番号を表示している。この番号はウィンドウによる監視を行なうときに必要になる。

## 12. 生徒のマシン名

現在生徒が使用しているマシン名を表示している。教師と生徒で使用しているマシンが違う時にはトークを行なう時にマシン名が必要になる。

## 13. 呼びだしボタン

教師は通常は生徒の演習用画面 2. を監視しているだけだが、指導の必要を感じた時にこの呼びだしボタンを押して、生徒に合図を送ることができる。呼び出しボタンには生徒のログイン名が表示されている。このボタンを押すと教師側の呼びだしボタン

と生徒側の呼びだしボタンのウィンドウの色が変わる。そしてウィンドウの色がもとに戻ると生徒が合図を確認したことになりトークを開始することができる。

また生徒の方からも教師を呼び出すことができ、その場合もウィンドウの色が変わるので、呼びだしボタンを押し生徒に合図を確認したことを伝え、トークを始めることができる。

#### 14. トーク用のウインドウ

教師はこのウインドウを用いて生徒とトークを行なうことができる。これにより通信が片方向になりがちという問題は解決できる。

#### 15. 現在参加している生徒のリスト

教師は生徒から送られてきた情報をもとに生徒一覧を作成し、教師側画面に表示する。この生徒一覧はまず生徒がログインした時点で xlist というプログラムを実行し、作成している。これを xnlist というプログラムを実行し表示している。生徒はこのウィンドウを生徒側画面で監視している。

## 6.4 XREC

XCAI では、XREC というプログラムにより、生徒の学習の記録を自動的に取っている。XREC はネクストボタン5. と連動しており生徒が各問題に対してどのくらい時間をかけているかを秒単位で記録している。XREC を実行すると生徒の名前とその生徒が各問題にたいして何秒かけているかを表示する。これは例えばほとんどの生徒がある問題のみ非常に長い時間がかかっていたとするとその問題は難し過ぎるなどの問題があるということになるので教師が今後の教材改良の参考にすることができる。図 6.6 に XREC の実行結果を示す。

Name ichii	
Q 1:	5 秒
Q 2:	1 秒
Q 3:	433 秒
Q 4:	17 秒
Q 5:	26 秒
Q 6:	25 秒
Q 7:	6 秒
Q 8:	9 秒
Q 9:	9 秒
Q10:	20 秒
Q11:	12 秒
Q12:	102 秒
Q13:	13 秒
Q14:	12 秒
Q15:	9 秒
Q16:	23 秒
Q17:	8 秒
Q18:	17 秒
Q19:	6 秒

Figure 6.6: XREC の実行結果

# Chapter 7

## 実験及び考察

### 7.1 XCAI の特徴

XCAI を用いて模擬的に授業を行なった。実験方法としては、教師1生徒1の形式で2ヶ所の部屋にわかれて、授業を行なった。

その結果このシステムの欠点として、教師が常に画面の前になければならないということや、同時に学習できる生徒を大人数にできないということが分かった。また、長所としては呼びかけボタンと、教師や他の生徒とのトーク用のウィンドウのおかげで、質問などが簡単にできるので生徒は教師に勉強をみてもらっているという感じをうけられることが確認できた。

### 7.2 グループウェアとして見た遠隔地型 CAI

#### 7.2.1 グループウェア

本システムの構成上のヒントはグループウェアの研究から得ている。グループウェアというのは、「協調して作業を進めるグループのために特別に設計されたシステム」のことである [5]。コンピュータ上に仮想の部屋を用意して参加者全員がそこにアクセスすることに

より話し合いなどを行なうことを可能にする電子会議室などが例としてあげられる [4]。そして遠隔地型 CAI はグループを教師と生徒、作業を生徒の知識の向上と考えればグループウェアのひとつの形である。

グループウェアでは遠隔地型 CAI の片方向の通信になりがちなどの問題点は全てコミュニケーションの問題に帰着される。その理由はグループウェアによる共同作業というのは通常のみんなが同じ場所に集まって行なう共同作業に比べるとコミュニケーションが希薄になるからである。このコミュニケーションの問題に対してグループウェアでは共用ウィンドウやマルチメディアを利用した双方向の通信が考えられており XCAI でのウィンドウの監視機能やトークはこれらからヒントを得たものである [3]。

### 7.2.2 情報の流れの研究 (3層構造)

さらにグループウェアでは情報の流れの研究も行なわれている。特に人間社会の情報伝達には brain、gatekeeper、end user の3段の階層構造が認められており、(図 7.1 参照) これによると end user は brain の情報を直接 brain から得るのではなく、gatekeeper を介して得ている [6]。

これを授業に置き換えて考えれば brain は教師、gatekeeper は優秀な生徒、end user は一般の生徒に対応する。つまり教師と生徒間よりも、生徒どうしのコミュニケーションを向上させる方が生徒の知識の向上には有効だと推測できる。これは通常の授業でも見られる生徒どうしの教え合いや、何か分からないことがあれば教師には質問せずに、まず身近にいる優秀な生徒に質問するという動向からも観察できる。この理由としては end user の数に対して brain の数が少ないという点やあまりにも自分と持っている情報の量が離れている人とは話しづらいという心理が働くからだと思われる。

本システムのトークによる生徒間でのやりとりはこの点の実現を狙っている。

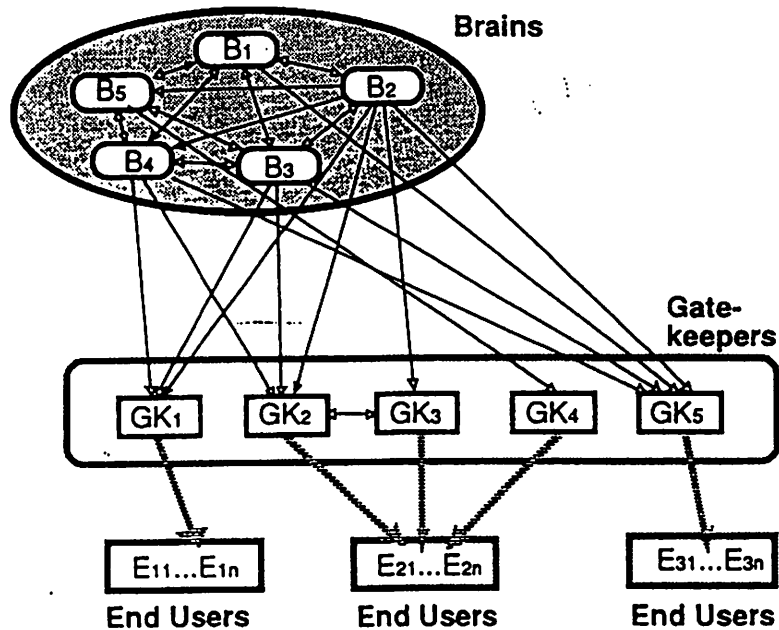


Figure 7.1: 情報伝達の三段構造

### 7.3 欠点の解決策

ここで上記の欠点にたいして今後どのようにしたらよいかを考える。まず教師は同時に大人数を相手にできないという点について考える。本システムでは生徒の上限は二人になっているが、実際は仮想画面を使うなどしてもう少したくさんの生徒画面を表示をすることができる。しかし生徒の人数が増えると教師は生徒一人一人に対してきめ細かな対応ができなくなる。本システムでは教師と生徒で用意しなければならない機材などは同じであり、学習教材にたいして知識があるかどうか重要な点である。よってある学習教材にたいして始めは生徒であった人も、学習を終えてその知識を得ることができれば次からは教師になることが可能であるので、そういった環境を整えることができれば教師が大人数を相手にできないという点はあるていど解決できる。

次に教師が常に画面の前にいる必要があるという点について考える。まず生徒がいつログインしてくるかわからないという点についてはログインしてもいい時間をあらかじめ決めておくなどの方法で解決できる。またいつ生徒からの呼びだしがあるかわからないとい

うことに対してはこれは本研究が教師と生徒とのコミュニケーションを第一に考えている  
以上しかたのない問題である。

# Chapter 8

## まとめ

本研究では遠隔地型 CAI システムを作成した。これは、通信技術とコンピュータを用いた CAI のシステムである。ただ従来までの遠隔地型 CAI システムでは通信が片方向になりがちといった問題があるので本システムではグループウェアの考えかたをヒントにウィンドウの監視機能とトークを用いた双方向の文字による通信のやりとりを取り入れた。本システムの作成には X ウィンドウシステムを用いた。

これにより教師は生徒のあるウィンドウを監視することができ、トークを用いて生徒を指導することができる。また生徒は画面に表示された問題を解きながら、分からないことがあればすぐにトークを用いて教師や他の生徒に聞くことができる。

また問題点として教師が常に画面の前にいる必要がある、小人数しか参加できない、などがある。これらの対応を今後の課題としたい。

# Bibliography

- [1] Adrain Nye and Tim O'Reilly(著) 今泉貴史(監訳)：『X ツールキット・イントロ  
シクプログラミング・マニュアル』，ソフトバンク，東京(1992)
- [2] Richard Stallman(著) 竹内郁雄・天海良治(監訳)：『GNU Emacs マニュアル』，bit  
別冊，pp.150-166(1988).
- [3] 猪井一代 他：“グループウェアによる授業支援システム”，情処全 47 回大会，3U-3，  
1-17(1992).
- [4] 岡田謙一：“情報を通して触れ合う”，『知的触発に向かう情報社会』，bit 別冊，  
pp.77-87 (1994).
- [5] 岡田謙一 他：“グループウェアの実現に向けて”，情報処理，Vol.34 No.8，pp.983-  
1062(1993).
- [6] 後藤滋樹 他：“人間社会の情報流通における三段構造の分析”，人工知能学会誌，Vol.8  
No.3，pp.348-356(1993).
- [7] 鈴木栄幸 他：“オンライン受講状態把握システムの試作と実践利用”，信学技報，  
ET90-60，pp.13-18(1990).
- [8] 田村武志・佐藤文博・上西慶明：“遠隔教育教授法に関する一考察”，信学技報，ET92-60，  
pp.49-54(1992).

- [9] 中山実・清水康敬：“通信衛星を利用した遠隔教育システム (PINE-NET) の学習成績による評価”，信学技報, **ET92-61**, pp.55-62(1992).
- [10] 吉田功 他：“テレビ電話を利用した個別教育システムの開発”，信学技報, **ET90-65**, pp.47-52(1990).

# 付録：EMACSによるメールの利用方法

市井 剛裕

Ver 1.0

このテキストは XCAI を利用する際に併読するテキストです。XCAI の指示に従って、このテキストを参照しながら演習を行なって下さい。

## 1 Rmail を使ったメールの読み方

Rmail は、受けとったメールを読んだり処理するための Emacs のサブシステムです。Rmail を呼び出すには、M-x rmail と入力します。Rmail ファイル内では、メッセージは受けとった順番に並べられます。各メッセージには、順にメッセージ番号が割り当てられます。q(rmail-quit) で、Rmail が終了します。

### 1.1 メッセージ内スクロール

Rmail で表示するメッセージが画面に入りきらないときは、画面をスクロールさせる必要があります。

SPC	順方向へスクロールします。
DEL	逆方向へスクロールします。
.(ピリオド)	メッセージの先頭へもどります。

### 1.2 メッセージ間での移動

メッセージに関して行なう最も基本的な操作はそれを読むことです。

Rmail では目的のメッセージをカレントメッセージにすれば読むことができます。Rmail にはいったときには新しいメッセージの最初のもの位置にいます。もし新しいメッセージがなければ、最後のメッセージの位置にいます。次の新しいメッセージを見るためには順方向に移動します。昔のメッセージを確認するためには逆方向に移動します。

n	次の消去されていないメッセージへ移動します。 消去の印の付いたメッセージは飛び越します。
---	---

- p 前の消去されていないメッセージへ移動します。
- M-n 次のメッセージへ移動します。消去の印の付いたメッセージも含みます。
- M-p 前のメッセージへ移動します。消去の印の付いたメッセージも含みます。
- j 最初のメッセージへ移動します。

### 1.3 メッセージの消去

あるメッセージを保存しておく必要がなくなったときには、それを消去できます。消去や、抹消はメイルリスト上で行います。

これを行なうとメッセージに消去の印がつけられ、いくつかの Rmail コマンドはそのメッセージは存在しないのとして扱います。

しかし、そのメッセージはまだ Rmail ファイルのなかにあつて、メッセージ番号を持っています。

Rmail ファイルで抹消をおこなうと、消去の印の付いたメッセージが本当に消されます。残ったメッセージには番号を順に付け直します。

- h メイルリスト上に、移動します。
- x メイルリストから出て、カレントメッセージを表示します。
- d カレントメッセージに消去の印を付けて  
次の消去されていないメッセージへ移動します。
- C-d カレントメッセージに消去の印を付けて  
前の消去されていないメッセージへ移動します。
- u カレントメッセージ、または消去の印の付いたメッセージへもどって、  
その消去の印を消します。
- e Rmail ファイルから消去の印の付いたメッセージを抹消します。

## 2 メールの送信

Emacs 内でメッセージの送信をおこなうには、まずコマンド (C-x m) を入力して、'\*mail\*' バッファを選択し初期化します。次に、このバッファでメッセージのテキストとヘッダを編集してから、別のコマンド (C-c C-c) を入力してメッセージを送ります。

- C-x m 送信するメッセージを作成します。
- C-x 4 m 上と同様ですが、メッセージを別のウィンドウに表示します。
- C-c C-c Mail モードでは、メッセージを送信して別のバッファを選択します。

メール作成バッファは普通の Emacs バッファなので、メール作成中に別のバッファに切替えたり、そのあとでもどったり（もどらなったり）することができます。

## 2.1 メールバッファのフォーマット

メッセージには、テキスト（送る内容）に加えてヘッダフィールドがあり、そこには、だれが、いつ、だれに、なぜ、送るのかなどがかかれています。

日付や送信者などのいくつかのヘッダフィールドは、メッセージが送られると自動的に作成されます。

受信者など他のフィールドは、メッセージが適切に送られるように指定する必要があります。バッファに表示される次の行

-text follows this line-

は、指定したヘッダとテキストを分ける特別な区切りです。

この行に続くものはすべてメッセージテキストです。ヘッダはこの行の前に置かれます。区切り行自体は、実際に送られるメッセージには現れません。

## 2.2 メールのヘッダフィールド

'\*mail\*' バッファではいくつかのヘッダフィールドが使えます。

それぞれのヘッダフィールドは、コロンで終る行頭のフィールド名で始まります。フィールド名は大文字でも小文字でも構いません。コロンといくつかの空白の後ろにフィールドの内容をかきます。：

'To'                   メッセージを送るメール宛先をかきます。

'Subject'              メッセージの内容を示す短いテキストを書くべきです。

'Subject' フィールドが便利なのは、メールを読むプログラムのほとんどは、メッセージのサマリとして各メッセージのテキストではなくこの内容のリストを表示するからです。

'From'                 メールを送るために使用したアカウントが自分のものでない時、自分がだれであるかを示すために 'From' フィールドを使います。通常、返事はここにくるので、'From' フィールドには有効なメールの宛先をかくべきです。

'Reply-To'            返事を自分ではなく、別の宛先に送ってもらいたい場合に 'Reply-To' フィールドを、使います。

## 2.3 Mail モード

'\*mail\*' バッファで使用される主モードは Mail モードです。

これは、C-c 接頭辞の付いたいろいろな特別コマンドが用意されているという点を除けば、Text モードとよく似ています。

これらのコマンドはすべて、メッセージの編集や送信用です。

C-c C-s      メッセージを送ります。'\*mail\*' バッファは選択されたままに残します。

C-c C-c      メッセージを送り、別のバッファを選択します。

C-c C-f C-t   'To' ヘッダフィールドへ移動します。なければ作成します。

C-c C-f C-s   'Subject' ヘッダフィールドへ移動します。なければ作成します。

C-c C-w      メッセージテキストの最後にファイル'/.signature'の内容を挿入します。

C-c C-y      Rmail で選択されたメッセージをヤंकします。

## 2.4 Rmail からの返事の送信

Rmail には Mail モードを使ってメールを送信するためのいくつかのコマンドがあります。

m            メッセージを送信します。

c            書き始めたメッセージの編集を続けます。

r            カレントメッセージに対する返事を送信します。

f            カレントメッセージを他の利用者に転送します。

以上で Emacs 上でのメールの基本的な使い方の説明を終わります。