

## ユーザインタフェース

~ End User Programming ~  
~ Multimodal Interfaces ~

(第4回)

五十嵐 健夫

## Schedule

- 4/5 インTRODクシヨ
- 4/12 インタフェースデザイン・評価
- 4/19 Information Visualization
- 4/26 Programming by Example (課題出題)
- 5/10 Pen computing
- 5/17 3D User Interfaces
- 5/24 Real world Computing (課題切)
- 5/31 課題講評

## 前回の内容

情報視覚化 (Information Visualization)  
情報検索

- Information Visualizer (Xerox PARC)
- Focus + Context, FishEye
- Zooming UI
- HCIL (Shneiderman)
- Tool Glass and Magic Lenses
- Automatic Zooming

## 今回の内容

End-user Programming, Prediction  
(Programming by Example / Demonstration)

- MetaMouse, Eager, Chimera, repeat&predict, Pobox
- BitPict, Agentsheets, Viscuit

Multimodal User Interfaces

- Put-that-there, 音声補完, VoiceAsSound

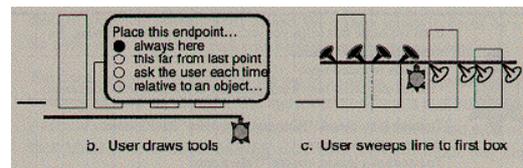
## End-user Programming

通常のプログラミング言語を用いず、  
エンドユーザが簡単にプログラムを書けるようにする。

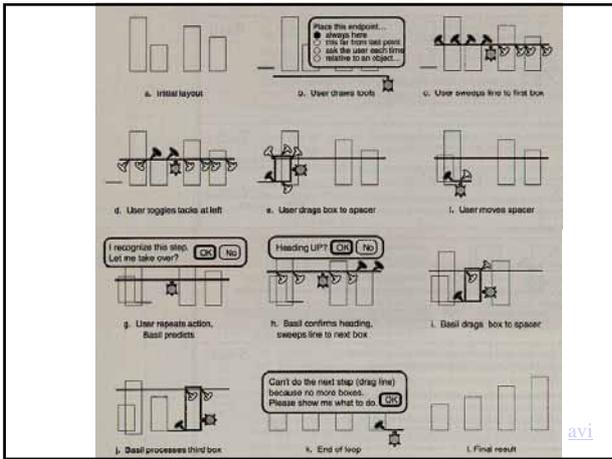
- 例示・予測インタフェース  
– Programming by example / demonstration
- ビジュアルプログラミング

## MetaMouse

[Maulsby 1989]

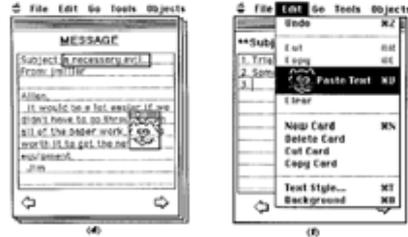


賢いマウス (Bazil) に操作を教え込む。  
「教師と生徒」メタファーで、マクロの作成。  
簡単な推論の実行。実行提示による確認。



## Eager

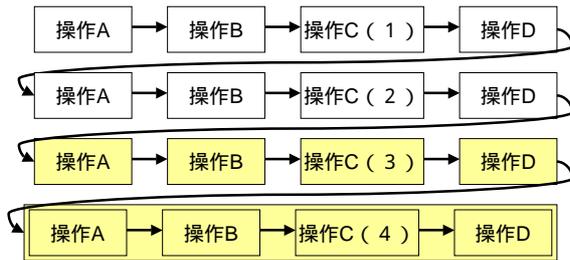
[Cypher 1991]



HyperCard上での操作を観察して、繰り返しを見つけると、次の操作を予測して提示。予測が確実になると、最後まで自動実行。

## Eager

[Cypher 1991]



HyperCard上での操作を観察して、繰り返しを見つけると、次の操作を予測して提示。予測が確実になると、最後まで自動実行。

[eager](#)

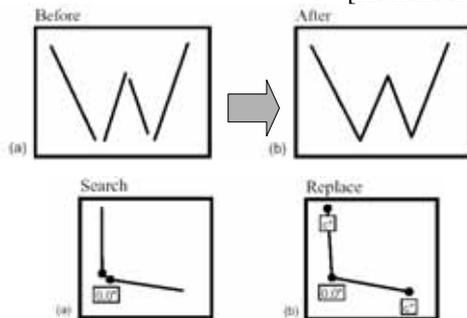
## Chimera

[Kurlander 1995]

- Graphical search and replace
- Constraint-based search and replace
- Constraints from multiple snapshots
- Editable graphics history

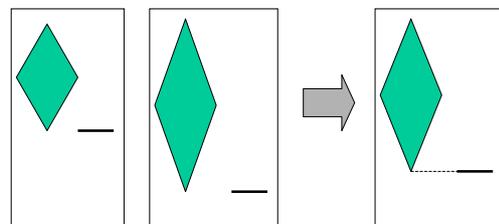
## Constraint-based search and replace

[Kurlander 1995]



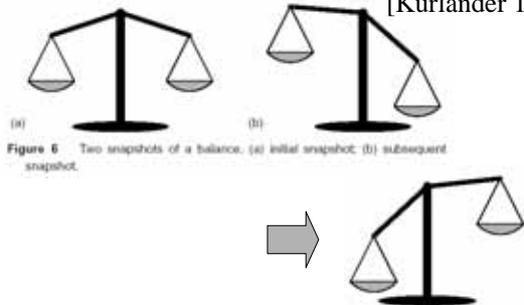
## Constraints from multiple snapshots

[Kurlander 1995]



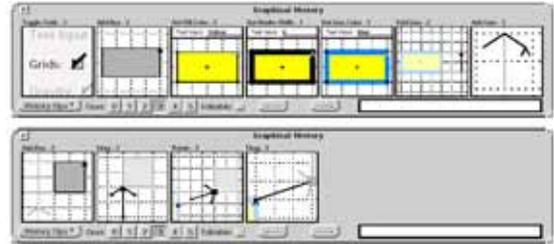
## Constraints from multiple snapshots

[Kurlander 1995]



## Editable graphics history

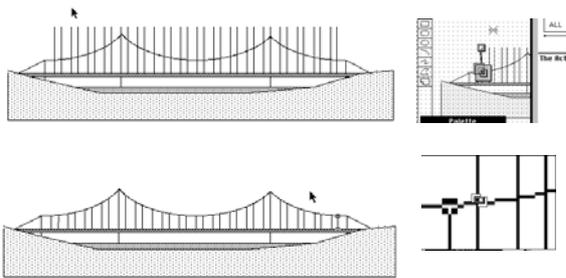
[Kurlander 1995]



[\\_chimera.avi](#)

## Triggers

[Potter 1993]



ビットマップパターンを探して操作を行う [triggers](#)

## Repeat and Predict

[Masui 1994]

テキスト入力(emacs)の補助

Repeatボタン：直前の操作の繰り返し

`[%][space][^N][^A][%][space][^N][^A]`

→ `[repeat]=[%][space][^N][^A]`

Predictボタン：複数の候補を順に提示

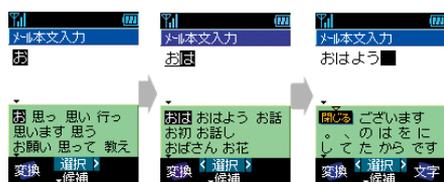
6789 → 6789: → 678910 → 6789A

`[predict]`      `[predict]`      `[predict]`

<http://pitecan.com/DynamicMacro/>

## PoBox (Sonyの携帯電話)

[Masui 1998]



予測と曖昧検索による文字入力方式  
辞書と履歴の利用

プログラミング支援  
スクリプティング



## Visulan

[Yamamoto 1995]

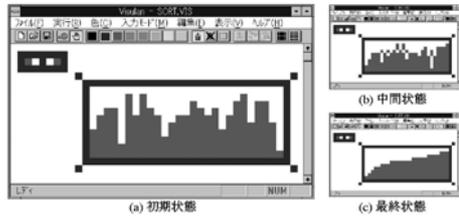


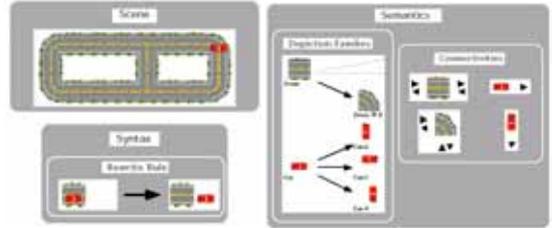
図 8. アプリケーション例 1: ソーティング

3D版もあり。



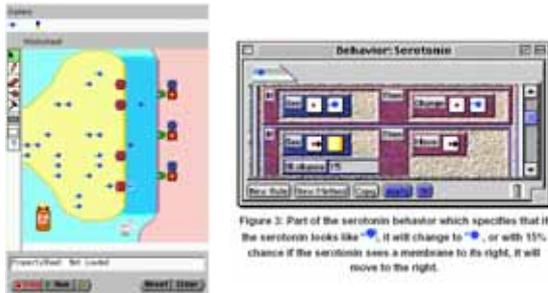
## AgentSheets

[Repenning 1995]



2次元配置されたセル世界とエージェント。  
簡単なシミュレーション。教育用。

## AgentSheets [Repenning 1995]



センサー情報に基づいて行動を選択する。

## AgentSheets [Repenning 1995]



その他の例。

## Viscuit

[Harada 2003]



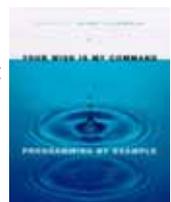
Fuzzy Rewriting: 曖昧なマッチングを許す。  
もっとも現在の状況に近いルールが発火する。

## 参考資料

Watch What I Do:  
Programming by Demonstration  
by Allen Cypher



Your Wish is My Command:  
Giving Users the Power to Instruct  
their Software  
by Henry Lieberman



## マルチモーダルUI

音声とジェスチャーなどを組み合わせる。

## Put that There

[Bolt 1980]



「あれをここに移動しろ」などと指示する。  
音声 + 手振り

## Mutual Dissambiguation for AR and VR

[Kaiser 2003]

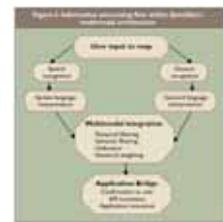


「あれをここに移動しろ」などと指示する。  
音声 + 手振り

[video](#)

## QuickSet

[Oviatt 2000]



音声とペンでの指示を組み合わせる。

[video](#) [video](#)

## 音声入力 of 拡張

## 音声補完

[後藤 2000]

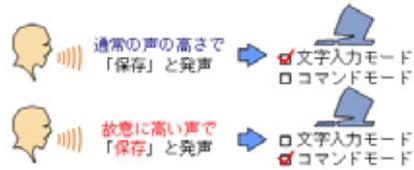


言いよどむと助けてくれる。

[video](#)

## 音声シフト

[後藤 2001]

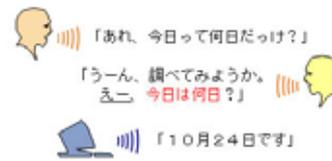


声の高さでモードを切り替える。

[video](#)

## 音声スポット

[後藤 2004]



言い淀んだ後に高い声で発声した箇所だけ音声認識する

[video](#)

## 音声による直接操作

UI ST 01

Voice as Sound:  
Using Non-verbal Voice Input  
for Interactive Control

Takeo Igarashi  
John F. Hughes  
(Brown University)



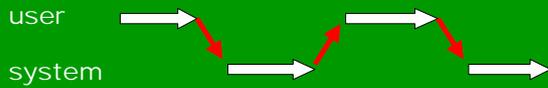
## Two Aspects of Voice

- Verbal information
  - ➡ Speech recognition
- Non-verbal information (pitch, volume, speed, etc)
  - ➡ Voice as Sound techniques

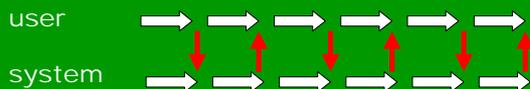
Interaction turn-around is long in voice recognition.



Interaction turn-around is long  
in voice recognition.



Voice as Sound achieves  
more immediate control.



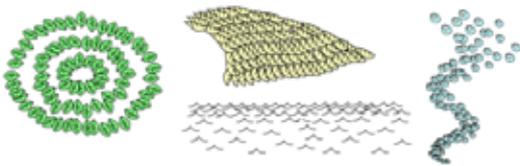
## まとめ

例示・予測インタフェース、  
音声インタフェースについて紹介した。

どちらも最も単純なものが実用化されているが、  
複雑なものは研究レベルにとどまっている。

## 課題

多量のオブジェクトを効率よくレイアウトする  
ためのインタフェースをデザインして評価する。



移動、回転や拡大縮小操作を効率よく行えるようにする。

## 課題

### 実装

基本機能のみのJavaのサンプルを配布する。他言語でもよい。  
ショートカット、ドラッグ操作、グループ操作などを実装してほしい。  
オリジナリティを重視する。

### ユーザテスト

自分+自分以外2人の合計3人で操作時間を計測する。  
改善すべき点などを明らかにしてレポートにまとめる。

### 提出物

プログラムのソースと実行ファイル、実験結果をまとめたものを  
メールで提出する。

### 詳細は

[www-ui.is.s.u-tokyo.ac.jp/~takeo/course/2007/ui/index.html](http://www-ui.is.s.u-tokyo.ac.jp/~takeo/course/2007/ui/index.html)

## プログラミングなしの課題

情報科学科以外の学生で、特にプログラミングが  
できないという学生は以下の課題とする。

1. 自分で実装したインタフェースの代わりに、既存のソフト  
ウェア（フリーでも製品でもよい）でもっとも今回のタス  
クに適していると思われるものを使って実験を行う。  
（パワーポイント、イラストレーター、など）
2. 実験結果に基づいて、理想的なインタフェースを自分でデ  
ザインして、それを説明した文書を作成する。

## スケジュール

- 4/26 課題発表
- 5/16 実装終了（目安）
- 5/23 課題締め切り
- 5/31 課題講評