2R−11　対話型アプリケーションにおけるコマンド仕様の設計法*

桂木真一郎†
NTTソフトウェア研究所‡

1　はじめに

本稿は対話型ソフトウェアアプリケーションのコマンド仕様の設計においてプロトタイピングを用いて行なう方式の提案ならびに実験結果である．HCPという木構造チャート設計用のシステム(図1)の設計時に行なった実験結果によれば,従来の状態遷移図などを用いる方式に比較して,設計上のケアレスミスの発見,コマンド体系の設計の細かい部分の改良などに効果が得られた．

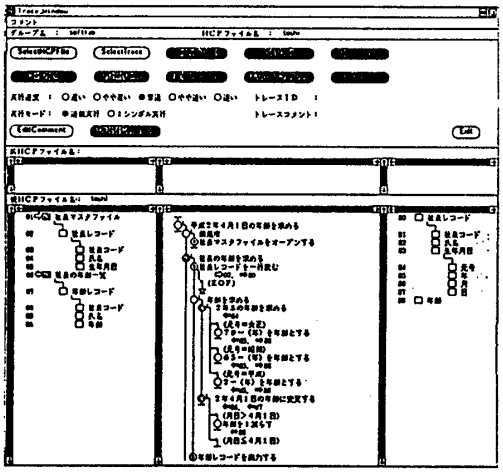

図1: 設計を行なったシステムの一画面

2　背景: コマンド仕様の重要性

図を用いた対話型アプリケーションではユーザインタフェース(ビジュアルインタフェース，コマンド体系)は重要な仕様である．

コマンド体系は,プログラムの構造と関係している．従って,もしコマンド体系が変更になると,プログラム本体の変更が生じる．

一方ユーザインタフェースのビジュアルな部分は,プログラムの本体と切り離すことが可能で,その場合ビジュアルインタフェースの変更はプログラム本体への影響が少ない．

表1: ユーザインタフェースの階層構造

ユーザインタフェース	ビジュアルインタフェース
	コマンド体系
処理部分	プログラム本体

*Designing User Interface for Some Application.
†Shin-Ichiroh KATSURAGI
‡NTT Software Engineering Laboratory

生産性良く対話型アプリケーションを開発するためには,その設計を行なうに当たり,コマンド仕様[1]の設計を重視し,これを早期に固定することが必要である．

3　仮説

コマンドを利用すればシステムの状態や,利用可能なコマンドや扱うデータが変化する．これらの変化をユーザにとって分かりやすく自然なものにすることは,良いシステムを作るための一つの目標である．

このようなコマンド体系の設計は,一般的にはコマンドの状態遷移図を作成し,その使用状況をイメージすることによりなされる(図2)．しかし状態遷移図からすべての場合をイメージすることは非効率である．

プロトタイピングを行なえば,状態遷移図のみを使う場合に比べて設計の結果を評価することが容易になると考えられる．

今回X上でアプリケーションを作成するにあたり，コマンド体系の設計をプロトタイピング手法を合わせて用いて行なうことにより,状態遷移図のみを用いる方法と比較して,

- アプリケーションの完成品を意識して設計を行なうことができる．
- ミスの少ない,ユーザにとってより好ましいコマンド体系の設計が行なえる．
- ユーザインタフェースの品質の向上とプログラミングにおける手戻りの解消につながる．

と考えた．

従来ユーザインタフェースのプロトタイピングについてはコマンド体系の定まった時点からのビジュアルな部分(メニューの提示方法，ボタンの位置など)の設計を専ら受け持っていた．

4　設計実験

X上の一アプリケーションとして,木構造チャート設計システムを開発した際のユーザインタフェース設計に関するプロトタイピングの方法について述べる．

利用したのはMacintoshのHyperCardである．これを用いてユーザインタフェースのプロトタイプを紙芝居的に作り,これを操作し使い勝手を調べることによりユーザにとっての利用のしやすさの観点から見たコマンドの論理設計の検討を行った.(図3)

インタフェースの規模は,プロトタイプでの画面の状態数としては約50枚分(評価の前後で改良のために枚数が変化した.)であり,全体を見通した状態遷移図2枚分である．

[1]メニューボタンの機能要素やメニューの階層構造などもコマンド仕様としてとらえる．

インタフェース設計の評価方法は，チェック項目を設けた上で，主観により5段階で評価した．

ユーザインタフェースの論理構造を決定するための手法として，状態遷移図を用いる方法の後にプロトタイピングを行なう方法を行なった．

5 結果と考察

状態遷移図のレビューに対して，プロトタイピングを利用した場合，以下の項目について新たな改良を行なうことができた．

- コマンドの階層を少なくできた．

  これは，実際にコマンドの階層を経験して，深過ぎる階層に気づいたことによる．この適正な階層数は一律に決まるものではないので，プロトタイピングは個々の場面の詳細な状況を調べるのに適しているといえる．

- コマンドの階層からのquit先の誤りを発見できた．

  これは，状態遷移図でも発見可能であるが，ケアレスミスによるものである．このことにより，プロトタイピングは状態遷移図を用いた方法のケアレスミスを補う性質を持つ．

- コマンドの用途がはっきりしたため，コマンドのカテゴライズを変更した．

  これは，プロトタイピングを行なったことにより，そのコマンドの利用状態が実感できたことによる．

- あるコマンドの実行の際に確認を行なうことが望ましいことを発見した．

  これは，消去などのコマンドを安易に利用することの危険性を実際に把握できたことによる．

二つの設計手法の比較（表2）によれば，

表 2: コマンド使用設計法の比較

	状態遷移図	プロトタイピング
全体の見通し（網羅性）	○	×
コマンド利用場面の詳細把握	△	○
ケアレスミスの発見	×	○
コマンド利用の実感	×	○
準備所要期間	○	△
設計所要期間	△	△
得られる精度	△	○

プロトタイピングによる設計は，コマンド利用場面の詳細の把握などには優れるが，全体の見通しが悪く，準備に時間を要する．従って，まず状態遷移図で全体を見通し，ついで主要部についてプロトタイピングを行なうことにより設計を行なうのが効果的である．

6 おわりに

対話型プログラムのコマンド仕様の設計において，プロトタイピングを用いることにより，仮説通り，ミスの発見が容易に行なえるなどの効果が得られた．この理由はプロトタイピングの持つ性質である，ユーザの身にたったコマンド体系の設計が行なえることによるものである．準備にかかる工数などの評価については今後の課題である．

参考文献

[1] D.A.Norman: 誰のためのデザイン？（認知科学者のデザイン原論），新曜社 (1990)

[2] Reinhard Budde, Heinz Zuellighoven:*Prototyping Revisited*,CompouEuro'90, Proceedings of the 1990 IEEE Interntional Conference on Computer Systems and Software Engineering, IEEE Computer Society Press, Los Alamitos California.

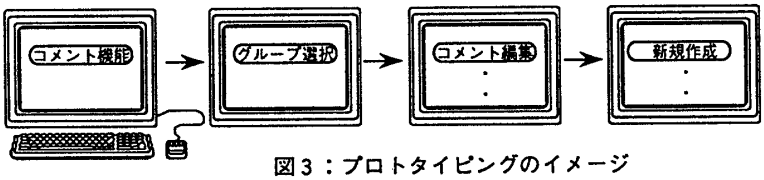

図3：プロトタイピングのイメージ

状態を追いかけて頭の中でイメージすることに比べ、実際に画面の前にすわってコマンドの結果や遷移の様子を知ることができるため、自然にユーザになり代り、コマンド使用の評価が行える。

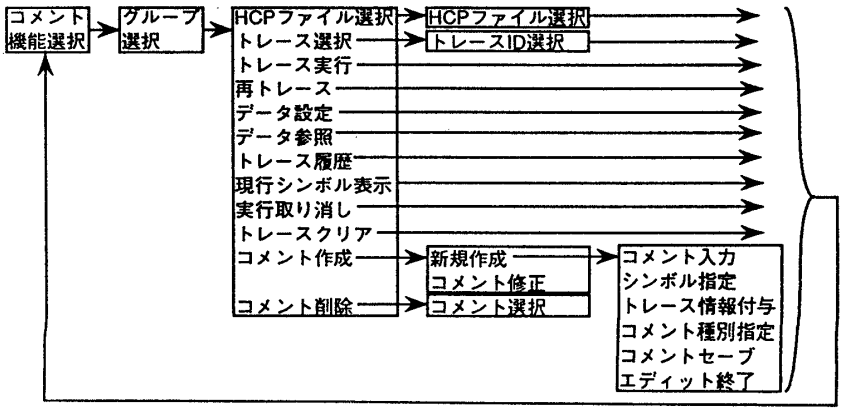

図2：状態遷移図のイメージ