5R-7

マルチビューモデルによるＵＩ記述

日本電気㈱　Ｃ＆Ｃシステム研究所

神場　知成，橋本　治

1．はじめに

筆者らは，ユーザインタフェース（以下UI）シミュレーションの研究を進めているが[1]，これを発展させてUIの統合的な設計／評価システムを構築するためには，その基礎となるUI記述方式の確立が大きな問題となる．本稿では，従来の主なUI記述方式を検討／分類するとともに，UIの本質的特徴である多面性に着目したUI記述方式「マルチビューモデル」を提案する．

2．従来のUI記述方式[2]

従来UI記述には，主に次の４つが用いられていた．

(1)状態遷移ダイアグラム

システムの各状態に対応するノードと，状態間の遷移に対応するアークから成るダイアグラム．

(2)BNF記法

文脈自由文法など形式言語のシンタクスの記述に利用されていた記法を，UIの記述に転用．

［例］ <CMD>::=<H:OPEN> <C:OPEN-ACK>[2]

(3)対話記述用プログラミング言語

通常のプログラミング言語を，画面構成要素の視覚的属性，デバイス操作など，対話記述に必要な要素を高レベルで記述できるように拡張したもの．

(4)状態遷移表[1]

状態間の遷移規則を表形式にしたもので，表操作の機能を持つことによりUI全体の管理がしやすい．

しかし，最近ではダイレクトマニピュレーションの発展などに伴い，UI記述において並行／非同期な事象（システムの動作，ユーザの行動）が重要な課題であることが認識されてきた．これに伴い，上記のような記述法では記述力が不足していることが明らかになり，次のような記述法が検討されている．

(5)ペトリネット

プレイス(P)とトランジション(T)という２つのタイプのノードが連結された有向グラフである．Pはトークンを持つことができ，各Tはそこに連結されたPのトークンが条件を満たすと発火することができる．

(6)並行プログラミング言語

並行／非同期な事象の記述に対する従来の逐次型言語の欠点を補う形で発展してきたもの．

User-interface description with Multi-View-Model

Tomonari KANBA, Osamu HASHIMOTO

NEC Corp.

3．各UI記述法の特徴

上で述べた６つの記述法の特徴を検討する．UIは一般に，入出力と状態遷移という２つの項目に分けて記述することができる．それぞれの項目に対する，従来の記述方法の特徴を表１に示す．

表１．UI記述法の特徴

UI項目	入出力	状態遷移
状態遷移ﾀﾞｲｱｸﾞﾗﾑ	× ｱｰｸのﾗﾍﾞﾙ記述手段が別に必要	○ 人間が見てわかりやすい．並行/非同期には適さない．
BNF記法	△ 拡張が必要．	△ 並行/非同期には適さない．
対話記述用ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ言語	○ 適切に設計されていれば記述しやすい．	× 通常のﾌﾟﾛｸﾞﾗﾐﾝｸﾞ言語と同じ．
状態遷移表	△ 対話記述用ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ言語との相性が良い．	○ UI全体の管理が容易．並行/非同期には適さない．
ﾍﾟﾄﾘﾈｯﾄ	△ ﾌﾟﾚｲｽまたはﾄﾗﾝｼﾞｼｮﾝに割り付けることは可能	◎ 並行/非同期に適する．
並行ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ言語	× 通常のﾌﾟﾛｸﾞﾗﾐﾝｸﾞ言語と同じ	○ 並行／非同期に対する記述力は高いが，学習が困難．

UIの記述法は２つの項目をともに記述することができなければならないが，従来の各種記述法はそれぞれ，表１に示すように，記述に適した項目が偏っている．例えば，状態遷移ダイアグラム，状態遷移表，ペトリネットなどは状態遷移を記述することはできるが，それだけでは入出力に対する特別な記述

力はないので，対話記述言語などとの併用が必要である．また，対話記述言語は，画面デザインは高レベルに記述できるが，状態遷移に関しては通常のプログラミング言語である．

4．semanticsの記述

３ではUIを入出力／状態遷移という側面から見たが，別の重要な側面に，syntax / semantics がある．この例を図１に示す．

ex.1
syntax　　ｱｲｺﾝをｸﾘｯｸ → ｱｲｺﾝが反転
semantics　ﾌｧｲﾙを指示 → ﾌｧｲﾙが選択された
ex.2
syntax　　copy <fileA> <fileB>
semantics　<fileA> を <fileB> にｺﾋﾟｰ

図１．syntax と semantics

UI記述における semantics の問題は，特にUI評価の際に重要となる．semanticsの記述が行われていなければ，設計されたUIにおいて意味的に不完全な点がないか（例えば，ファイルをコピーする手段がない），というようなことは評価できない．

semantics を明示的に記述するという手法をとった代表的なものとして，Moran のＣＬＧがある[2]．ここでは対話を複数のレベルに分けて各レベル毎に形式的記述を行うことにより，syntaxとsemantics の記述を明確に分けている．

しかしＣＬＧを，２で述べた並列／非同期な事象の記述という観点で見直すと，やはり記述力が不足していることがわかる．これは並列／非同期な事象の意味論という問題となり，これに対処するには，例えば，AIの分野での Allen による時間論理[3]が有効であると考えられる．Allenは，" I stood on the corner for an hour. "のように時間的連続性を持つ動作や，互いに関連を持つ動作の記述を行うために，事象を時間的インターバルによって記述し，それら相互に成り立つ論理的関係を検討している．

5．マルチビューモデル

以上で述べたように，UIは，入出力と状態遷移，syntax と semantics など，本質的にいろいろな側面から見ることができるという特徴を持っているため，ある１つの記述法によってUIのすべての要素が網羅されることはない．これを「UIの多面性」と呼び，多面性を持つUIの記述手法としてマルチビューモデルを提案する．

マルチビューモデルは，ある１つの記述法によってUIをすべて記述しようとするものではなく，UIのいろいろな側面を，それぞれの見方に適した記述法によって記述しておき，モデルを利用するそれぞれの場面（UI仕様設計，プロトタイピング，評価など）で最適な記述法を利用しようという考え方である．

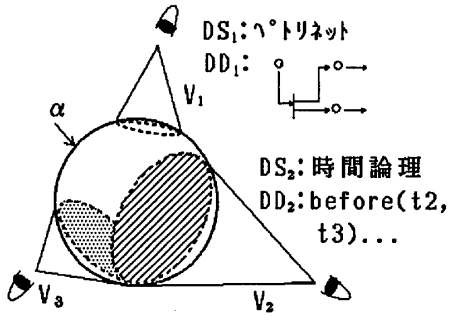

図２．マルチビューモデル

マルチビューモデルの概念図を図２に示す．マルチビューモデルは，形式的には次のように，ビュー，記述法，記述の組から成る集合として定義できる．

$M\alpha = \{ (V_i, DS_i, DD_i) \mid i=1,2,3,..\}$

Ｍα ：ﾕｰｻﾞｲﾝﾀﾌｪｰｽαのﾏﾙﾁﾋﾞｭｰﾓﾃﾞﾙ
V_i ：ﾋﾞｭｰ識別子，DS_i：V_iにおける記述法
DD_i：DS_iを用いた記述ﾃﾞｰﾀ

本報告では，ビューV_iとして(入出力,状態遷移)×(syntax,semantics) の４つを考慮する必要があることを示した．それぞれのビューに適切な記述法は次のようになると考えられる．

(1) 入出力のsyntax：
入出力の高レベル記述を可能にした対話記述用のプログラミング言語．
(2) 入出力のsemantics：
例としてＣＬＧがあるが，新たな記述言語が必要．
(3) 状態遷移のsyntax：
状態遷移表，ペトリネットなど．
(4) 状態遷移のsemantics：
時間論理の利用が考えられる．

6．おわりに

従来のUI記述法の特徴を整理し，マルチビューモデルを提案した．今後，UI記述に必要なビューの検討を進めるとともに，各記述相互間の記述力の関係，整合性，変換法等について検討を進める予定である．

＜参考文献＞

1)神場他：画面ｼﾐｭﾚｰﾀ U-faceの試作，情処39全大．
2)H.R.Hartson et al.:"Human-Computer Interface Development",ACM Comp. Surveys,26,1(1989).
3)J.Allen:"Towards a General Theory of Action and Time", Artificial Intelligence,23(1984).