情報システム 43－2
（1993. 3. 16）

利用者の参加を考慮した 視覚的な情報システム開発ツール

高 純
慶應義塾大学大学院理工学研究科管理工学専攻

従来より、利用者が中心となってシステム開発を行い、より使いやすく、対象業務要求を最も満足できる情報システムを作成できるようにするため、画面とその流れに注目した1つの開発方法とその開発をサポートする支援システムSOFT(Screen Oriented Flexible fabrication Tool) が提案されている．この論文では、SOFTの考えを基本におき、最近のGUIを導入して、視覚的かつ誘導的なシステム定義環境を備えた支援システムの構想を述べている．さらに、定義結果を解析しながら即時に試行することによって、設計段階でシステムの視覚的プロトタイピングを行なうことができる．

A Visual Tool for Information Systems Development which give Consideration to User Participation

Chun Gao
Faculty of Science and Technology, Keio University

The screen-oriented methodology for user-centered development of information system was proposed and a development support tool SOFT (Screen-Oriented Flexible fabrication Tool) based on the methodology has been developed. This paper describes a concept of enhancement of the SOFT which is incorporated with GUIs to navigate developers visually.The enhenced tool can execute visual specification immediately to support a prototyping approach for confirmation of user's reguirements in the design stage.

1. はじめに

コンピュータの利用分野は多様化されつつあり、その利用者も非常に拡大している．その一方で開発されるシステムには、対象業務に適し、より使いやすいことが望まれている．従ってシステムの開発及び保守に際して、コンピュータの専門家だけではなく、利用者の参加が不可欠となっている．それを容易にするため、扱いやすい支援ツールが求められている[15]．

本論文では、利用者が中心となって、画面とその流れに注目した情報システム開発を行う１つの開発形態を考察し、それを実現するのに必要な支援システムについて述べている．

ここで対象とするシステムは、利用者と対話しながら処理を進めていく小規模な対話型情報システムである．このようなシステムにおいて、利用者がそのシステムについて知ることができるのは、画面上で、操作によって、システムが流れていくことを通してのことに限られている．

従って、情報システムを設計するにあたり、まず画面構成とその流れを決定し、その後で、操作に対する必要なシステム内部の処理を決定するという手順が、利用者にとって、理解しやすく、システム設計に容易に参加できるのではないかと考えられる．

また、それらの設計に当たっては、紙面上のみでなく、定義した内容を試行、評価、再設計していくプロトタイピングを行って、次第に要求にふさわしいシステムを作成していけることが必要となってくる[1]．

本論文は、青木ら[11][12]が提案していた画面に基づいた情報システム開発支援ツールSOFTに、グラフィカル・ユーザインタフェース（GUI）を導入した．視覚的かつ誘導的な開発環境SOFT-2の構成について述べる．

2. SOFTの基本概念

本章では、SOFTにおけるシステム分析／設計作業[11][12]をGUIに適した形に整理し直したものについて、説明する．

2.1 画面に基づいた開発手順

利用者から見た対話型情報システムは、図１に示すように、ある画面上で必要な情報を入力し、システムからそれに対応する情報が表示される．また、ある項目を選択することにより、次の画面へ進む．利用者の操作に対して、システムは対応する内部的な処理を行う．

SOFTは、このことを考慮して、従来、システム開発のプログラミング段階で行なわれていた画面とその流れの設計に、システムの設計段階でもっと多く注目することによって、利用者を中心とした情報システム開発が行えることを念頭に置いている[16][17]．

画面に基づいたシステム開発を行う手順の方針はきわめて単純なもので、まず第１に画面とその流れを決定し、その後、画面の構成や画面上または画面間の遷移で行われる処理を決定していくというものである．その手順は次の通りである．

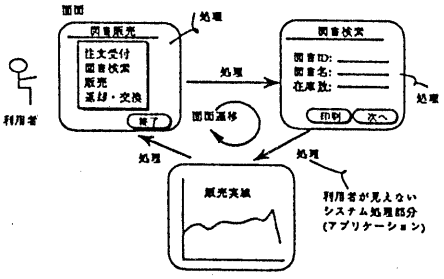

図１．　利用者から見たIS

[第1段階]　業務を小さな作業単位へ分割する．

最初から、システム全体の画面とその流れを決定していくのは考慮する範囲が広すぎて捕えにくい．そこでまず、対象となる情報システムの仕事を区切りのよいいくつかの処理単位に分割する．

[第２段階]　作業単位ごとに画面とその流れを決定する．

第１段階で分割した各処理単位で必要となる画面とその流れを決定することである．

[第３段階]　作業単位ごとにデータ項目とファイルを設定する．

作業単位毎に必要となるデータ項目を選びだし、それらを全体に眺め、必要とするファイルを決定する．

[第４段階]　各画面で行われる処理を決定する．

画面で行なわれるシステムと利用者との対話処理を記述する．

[第５段階]　画面間の遷移で行なわれる処理を記述する．

画面間の遷移過程でシステム内部で行なわれる処理を記述する．

SOFTは提案している画面に基づいたシステム開発手順において、画面とその流れという具体的イメージを紙面上の設計のみで止めるのでなく、コンピュータにより実際の稼働と似た環境で実現している．

2.2 システムの分析

2.2.1 State Transition Graph

対象システムの画面はそれぞれ利用者が操作したり処理を行なうある状態（State）を表している．システムは利用者の何らかの選択によって遷移（Transition）を起こし、新たなStateに遷移して処理が進むと考えられる．

システムの利用者から見たマクロな振る舞いは、画面をノードとする状態遷移グラフ(State Transition Graph、STGと略称する)で表現できる（図２）．

そこで、まず、そうしたシステムの利用者から見える側面を画面の定義と画面あいたの遷移の定義とに分けて

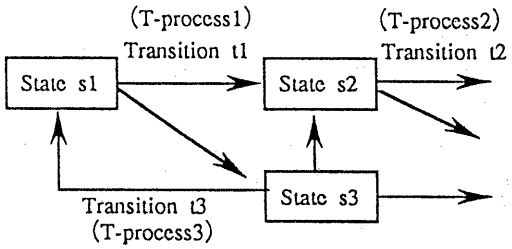

図２．　State Transition Graph

規定する．そして、遷移の際必要となるファイルの入出力などの利用者から見ることのできない内部的な処理（T-Process）を定義する．

(1) State

画面の構成を大きく２つの部分に分けてモデル化することができる．

画面(State)＝固定表示部分(Frame)＋対話要素(Sub State)

a) Frame

見出しなどの画面での固定されたものである．

b) Sub State

１つの画面はFrameとメニュー、入出力フィルドなどの対話要素（Sub State）から構成される．対話要素に対して、利用者により入力、コンピュータにより出力することで遷移（Sub Transition）を起こし、処理が進む（図３）．また、対話要素の間では入出力データに対する内部的な処理をSub T-Processと定義する．

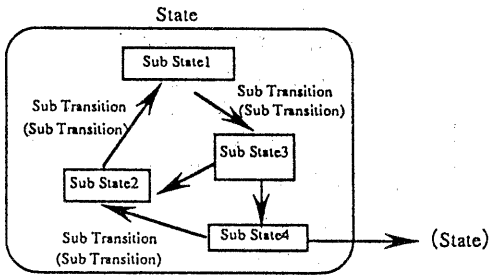

図３．　画面のモデル

(2) (Sub)Transition

一般に利用者が対話要素により入出力に対して、起こされる遷移は画面遷移（Sub Transition）と画面間（Transition）と２種類がある．

(3) (Sub)T-Process

(Sub)T-Processは利用者から見えないシステム内部の処理である．たとえば、対話により得た情報を対応するファイルに書き込んだり、必要な情報をファイルから検索し、編集することである．

2.2.2　Meta State

画面とその流れ（STG）はシステム設計の中心となる考えである．しかし、前も述べたように、最初からシステム全体のSTGを作成するのは、大変難しい．そこで、分割された作業単位に対応するものとしてMeta Stateの概念を導入する．

システム全体の各作業単位はStateの集まりで構成される．それらは１つの仕事を行なうためにお互いに関係しあっているものと考えられる．このStateの集まりをStateより１つ上位の概念としてMeta Stateと定義する．Meta State間の遷移はMeta Transitionと呼ばれる（図４）．

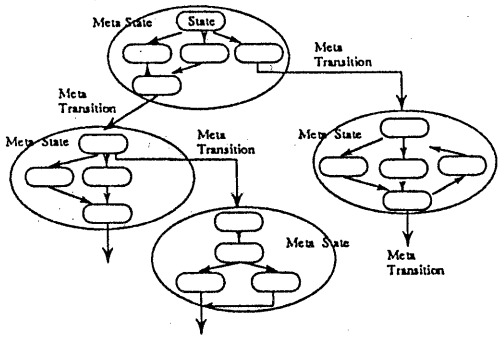

図４．　Meta Stateの概念

3.　SOFT-2による設計の概要

２節で述べたSOFTの基本概念にしたがって、画面に注目し、その構成要素を基本単位としてシステムを階層化することに基づいて情報システムを作成するために、大別して次の機能が必要となる．

(1) システムを定義するツール

- Meta State定義
- State定義
- Transition定義
- ユーザデータファイル定義
- T-Process定義

(2) システムを実行する制御モニタ

SOFTはStateなどの定義ツールを提供している．しかし、最近発展してきたGUIを導入することにより、さらに利用者にとっても、理解しやすく、容易にシステム設計に参加できるようにするため、視覚的かつ誘導的な設計環境を構築することが考えられる．このように改良したSOFTをSOFT-2と呼ぶ．

SOFT-2は、画面（State）を定義するのに必要な基本的な構成要素をボタンやメニューなどの部品として提供する．それらを組みあわせることにより、画面を簡単に構成し、その定義データを解析しながら即時に具体的なイメージと動作を確認できる．

さらに、画面内と画面間の遷移、また遷移に対応する内部的な処理に視覚的設計環境を提供する．

設計した画面、遷移、処理を連結し、システム全体の流れの視覚的プロトタイピングを行なう機能も持っている．

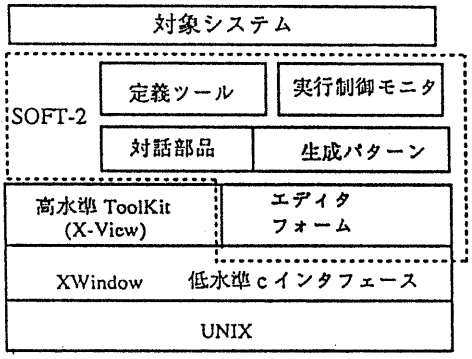

図５．　SOFT-2の位置付け

ユーザデータファイル定義に関して、今回の SOFT-2 は支援せず、画面とその流れの定義支援に注力する．

3.2　SOFT-2の位置つけ

SOFT-2 は、SUN ワークステーション上で汎用のウィンドウシステムであるX Windowシステム[9]をベースにして、開発中である（図５）．対話型システムを開発するためのXツールキットであるX View を採用し、画面を構成する対話要素を部品化する．また、対話部品のイベントの制御に関する部分は X View [13][14]のプロシージャとして実装している．従って、SOFT-2 はGUI[10]を持った支援ツールであり、開発対象システムもGUIを備えている．

3.3　対話部品

SOFT-2 では、システムと利用者との間の基本的な対話を実現する対話要素を部品化し、ボタンやメニューなどを対話部品ライブラリとして提供している．

用意している部品は、以下の７種類がある．

(1)フレーム（Frame)

画面に対応する．システムへのデータ入力（文字、数字など）や、システムの処理結果（文字、数字、およびイメージなど）の出力を行う長方形領域である．１つのフレームにはサブフレームを含むことが可能である．

(2)ボタン

コマンドを実行するために使用する，すなわち、マウスを押すことにより、対応する処理が行われる．

(3)選択的な部品

この種類の部品は、各種のメニューから成る．メニュー項目の配列は縦方向と横方向のどちらかを選択できる．この部品には、排他（Exclusive）（一回に１項目だけ選択できる）と非排他（Nonexclusive）（同時に複数の項目を選択できる）の２つタイプのメニューがある．

(4)入力部品

利用者がシステムへの入力を行うため、一行の文字や数字を入力できるフィルドを提供している．

(5)出力部品

処理の結果を表示するための部品である．この部品には、少量のデータ表示を行うフィールドと連続量のデータ表示を行うスクロールバーを提供している．

(6)ノーティス部品

この部品は、利用者の操作に対して、対応する処理を本当に実行するかどうか、確認させる場合、注意させる場合、エラーメッセージを表示する場合に使用するポップアップ部品である．

(7)テキスト部品

一般文章を入力できるエディタ部品である．テキスト部品はテキストファイルをロード、保存する機能の他に、文章に対する挿入削除、検索、移動、置換などの様々なテキスト編集機能を持っている．

3.4　開発環境

経験のない利用者の開発への参加を容易にするため、開発過程を誘導する環境が必要となる．2.1 節で述べた開発手順は、システムの分析した結果によって、実際に、SOFT -2は図６のような開発手順で設計者を誘導し、設計を進めていく．

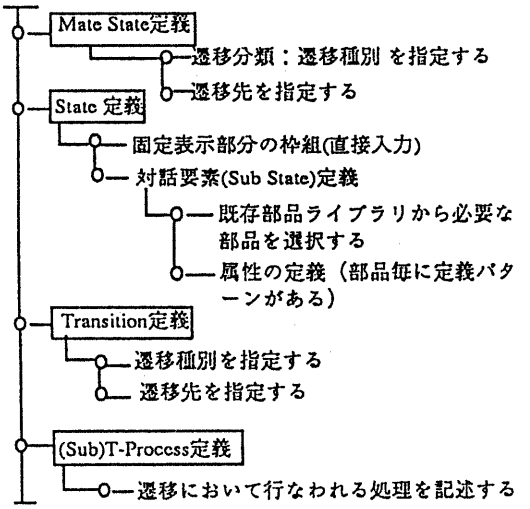

図６．　開発手順

さらに、システムの設計は紙面上の検討だけで完成するのではなく、開発現場で設計の変更、やり直すことが必ず発生する．設計対象を確認しながら開発を進めることにより、開発効率の向上を期待できる．このような観点から、SOFT-2 は視覚的な定義環境を提供している．設計者はそれを通して、コンピュータと対話しながら、システムの開発を行なう．

以下に、SOFT-2 における開発過程の例（図７）を示す．

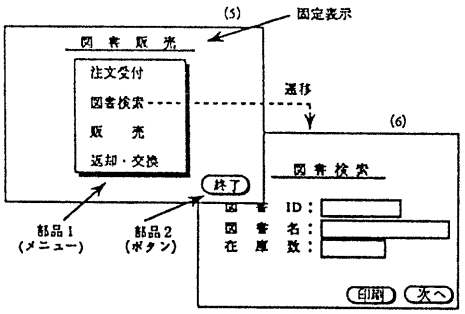

図7．　画面設計例

（1）State（画面）定義

画面をより簡単に定義できるよう、WYSIWYG(What You See Is What You Get)方式の定義環境を提供している（図8）．画面を対話部品の組み合わせにより構成し、対話部品のレイアウト、表示形式などの属性を対話的に行なう．

1. 画面番号

画面は新規の場合、番号を定義する．修正の場合、メニュー（Screen）から対象画面を選択する（図8）．

2. 固定表示部分

画面定義領域で、マウスで表示開始位置を指定し、表示内容を直接入力する（図8）．

3. 対話要素生成

まず、図9に示すようにメニュー（Comp）より、既存の対話部品の中から対象部品を選択する．その部品の大きさ、表示形式など属性を指定する．そして、画面定義領域で表示位置をマウスで指定し、その場で対象部品が即時に生成される（図10）．従って、設計者は確認しながら設計ができる．

（2）(Sub)Transition（遷移）定義

ついで、画面を構成する複数の対話部品の間の遷移、また、次の遷移画面を定義する．

ここでは、遷移先の部品／画面を指定し、また遷移の間で行われる処理（(Sub)T-Process）と関連づける．図11は遷移設計視覚的エディタである．

この遷移設定は部品毎に行われている．指定項目は、部品がボタンや入力の場合、1つであるが、メニューやノーティスの場合、複数がある．

1. 遷移の種別

遷移には4種類がある．

(a) 無条件遷移

遷移先は1つしかない．

(b) 条件遷移

処理の結果によって、遷移先が違う．

(c) 臨時遷移

操作発生した場合、対象処理が行われている間に、遷移先の画面（部品）がポップアップされ、処理が終わった後、その画面（部品）が点滅する．

(d) 消去

ある対話部品を操作したら、対応処理が終わった後、

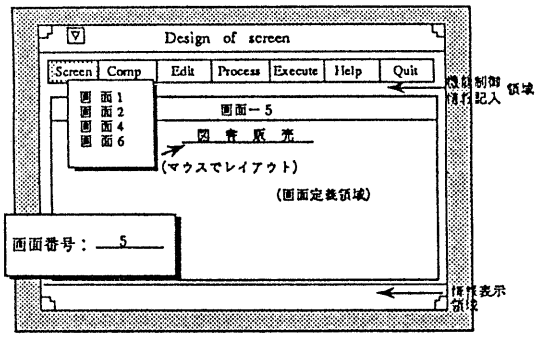

図8．　画面設計エディタ

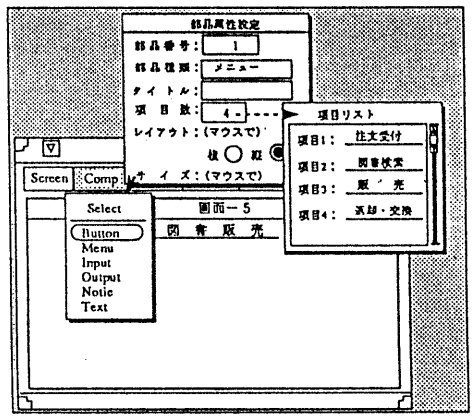

図9．　部品の定義例

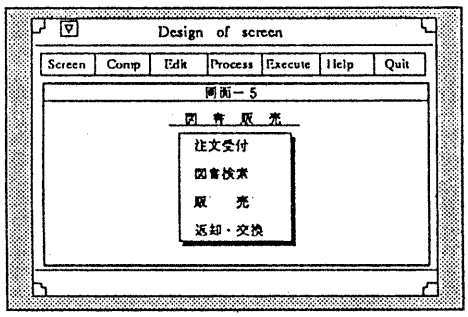

図10．　部品生成

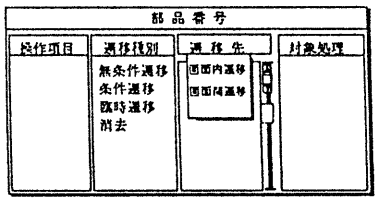

図11．　遷移設計エディタ

その部品を削除する．

遷移関係の指定はマウスで操作する．まだ定義されていない遷移先ならば、その画面（部品）の仮定番号を指定すればよい．

2. 処理と連結

遷移に対応する処理名を指定する．図１２は遷移設計のイメージである．

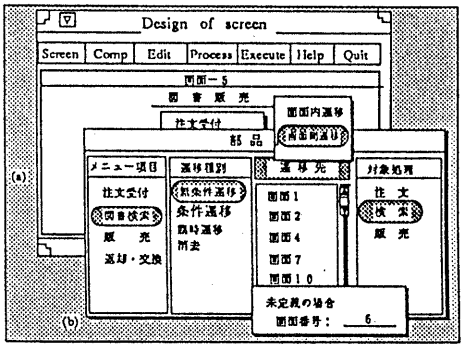

図12.　遷移定義例

（３）(Sub)T-Process 定義

SOFT -2 は (Sub)T-Process を記述するエディタを用意している．処理ファイルをロード、保存する機能の他に、検索、挿入、削除、カット／置換など多くの編集機能を持っている（図１３）．開発者はC言語によって定義する．

(Sub)T-Processの内容をこの場で必ずしも記述する必要がない．

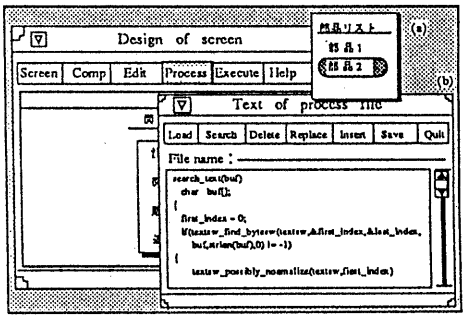

図13.　処理の記入エディタ

4.　SOFT-2の実現

4.1　SOFT-2の構成

SOFT-2のシステムは主に定義ツール、対話部品ライブラリ、定義データファイル、実行ファイル生成パターンおよび実行制御モニタから構成される（図１４）．

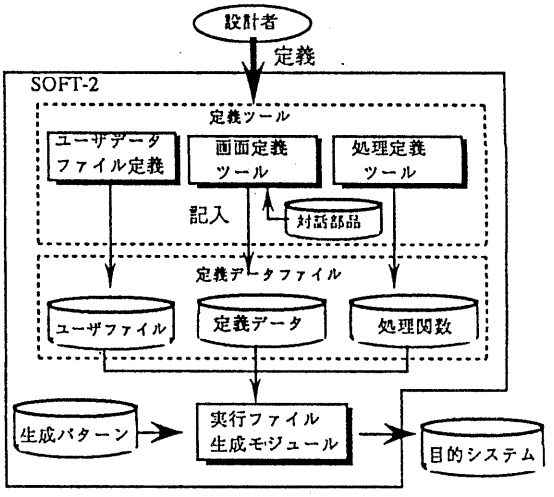

図14.　SOFT-2の構成

(１)定義ツール

開発システムを設計するために、視覚的な定義ツールである．その動きについては前節を参照すること．

(２)定義データファイル

定義ツールで設計した情報を各定義データファイルに記憶される．

(３)生成パターン

SOFT -2は画面を構成するには、定義データによって各対話部品を作成するために、生成パターンを次のように定義した．

部品名=（ オーナー、属する部品名、
属性リスト、
イベントの制御処理
）

・オーナー：その部品の親を指定する．
・属する部品名：部品の種別
・属性リスト：その部品の性質を指定する（サイズ、配置など）．
・イベントの制御処理：その部品に対するイベントが発生すると、事前に登録しおいた処理を呼び出す．

(４)実行ファイル生成モジュール

定義データを生成パターンによって、実行形式ファイルに変換するモジュールである．

4.2　定義データファイルの構造

設計者が定義した情報はユーザデータファイル、画面情報ファイル、処理ファイルの３種類のファイルに格納される．ここでは、画面情報ファイルについて説明する．

画面情報ファイルには、画面管理テーブル、対話部品データテーブル、処理名リストテーブルがある.

(1)画面管理テーブル

すべての画面の番号、またその画面を構成する部品群へのポイントを格納している.

(2)部品データ

このテーブルは2つの部分からなる.

(a)データ部

部品の番号、属する部品名、属性などを記入している.また、対象処理名を含まれている.それは処理ファイルと対応付けためのである.

(b)テキスト部

ここでは、画面の固定表示情報を記入している.表示位置と表示値がある.

(3)処理名テーブル

定義した全ての処理ファイル名を保有するテーブルである.

4.3 実行プログラムの生成

実行ファイル生成モジュールは上記の定義データを解釈し生成したプログラムは大きく初期化処理部、画面の枠組みの作成部、部品群定義部、処理定義部からなる.プログラムの構造は次に示す.

```
ヘッダファイル宣言
main()
{
 /* 初期化処理 */
        :
        :
 /* 画面の枠組み作成 */

 /* 対話部品の定義 */
    (コールバック関数を設定)

}

/* コールバック関数 */
{
        :
    (処理ファイルを関数として呼び出す)
}

/* 処理関数群 */
```

5. 考察

これまで、対話型情報システムの開発手順とそれをサポートするシステムSOFT-2についてのべてきた.本節で、その有効性について考察する.

(1)利用者参加の可能性

SOFT-2は具体的イメージで捕えることのできる画面をシステム設計の概念として与え、視覚的かつ誘導的な定義環境を提供している.また、画面構成要素を部品化し、対話的に部品を組み合わせて、画面を構成できることは、利用者にとっても、理解しやすく、容易に設計を行える.

(2)開発効率の向上

画面定義は処理定義から明確に分離され、処理定義に影響を与えなく変更できる.画面の定義データからプログラムを自動生成する.

(3)実行の即時性

システムの設計とその実行を即時に切り替えることにより、早期に完成品の姿を確認することができる.

今後の課題として、次のことを考える必要がある.

(1)対話部品の制限

基本的な部品に限られているため、より高度な部分を提供する必要がある.(イメージ、音声、動画)

(2)部品の動的定義ができるようにする.

6. おわりに

本論文では、画面とその流れを中心においたシステム開発方法において、視覚的開発環境を提案し、その構成について述べた.この方法によって、システム開発への利用者の積極的な参画を促し、また、開発したシステム自身がGUIを備えることによって,使い勝手のよい画面を設計できると思われる.

参考文献

[1] 西尾 高典：有向グラフの階層に基づく対話型応用ソフトウェア仕様記述の一方法ソフトウェア工学, 76-1, (1990.12.4)

[2] 大竹 勤, 中村 能章, 中川 透：画面ドライバと設計支援システムへの適用, ソフトウェア工学 56-4, (1987.9.9).

[3] 橋本 治, 宮非 均：ユーザインタフェースシミュレータINTERA, 情報処理 学会論文誌, Vol. 31 No. 10,Oct. 1990.

[4] 中村 能章、大竹 勤, 中川 透：ユーザインタフェースの設計モデル, ソフトウェア工学56-3, (1987.9.9).

[5] Shneiderman,B. : Designing the User Interface Addison-Wesley (1987).

[6] Myers, B. : User Interface Tools : Itroduction and Survey, IEEE Softw., pp. 15-23(Jan. 1989).

[7] 高資, 中谷等：プログラム群を統合するウィンドウ型ユーザーインタフェースシステム：UAI／X, 電子情報通信学会論文誌 '92/7 Vol. J 75-D-I No. 7.

[8] Danr. Olsen, Jr. : UserInterface Management Systems Models and Algorethms, Morgan Kaufmann Publishers San Mateo, Califoenia.

[9] Joel McCormack et al.：X Toolkit Intrinsics－C Language Interface、X Window System, Version 11, Release 3.
[10] Paul Barth：An Object-Oriented Approach to Graphical Interfaces, ACM Transactions on Graphics 5, 2(April 1986).
[11] 青木　隆：システム開発支援ツール、慶応義塾大学大学院　1979年度　修士論文
[12] 石川　由美子：情報処理システムの画面に基づいた設計とその実現、慶応義塾大学大学院　1982年度　修士論文
[13] 森下　茂：X-Viewプログラミング入門、工学図書株式会社、1990.11.25　第1版第1印刷発行
[14] Dan Heller：X-View Programing manual, 1990、O'Reilly & Associates, Inc
[15] Shoji Ura, Takashi Aoki, Takeki Ogawa, Akira Kawaguchi：Adenaced Patient Oriented Pharmacy System (APO-S)、MEDINFO 80.Lindberg/Kaihara, Editors c IFIP, North-Holland Publishing Company (1980)
[16] Shoko Kondow, Yumiko Ishikawa, Shoji Ura: Screen Oriented Designs and Implementation of Ifomation System、6 th International Conf on Software Eng.1982
[17] S.Kondow, Y.Ishikawa, S.Ura：Screen Oriented System Development Technology、MEDINFO-83,van Bemmel/Ball/Wigertz, Editiors c IFIP-IMIA;North-Holland(1983)