

6K-6

システム間インタラクションの仕様記述と それによる画像ソフトウェアへの応用

橋本 雄治、村尾 洋、榎本 肇

芝浦工業大学

1. はじめに

ソフトウェアプロセスをプログラミングとして考えると、定義と流れ制御の混在がある。これを除き、ソフトウェアプロセスとして明確にするためには、仕様定義部と手続き部分とを明確に分けて記述する必要がある。さらに並行処理や協調処理を考えると、意図を含めたインタラクション機能を自然言語風に記述する標準形式を定義し、仕様部と制御定義部の統合に使用できることが望ましい。

ここでは、ソフトウェアの仕様記述において、ソフトウェアの並行処理におけるシステム間の仕様参照動作関係をインタラクションとして表現できるような仕様記述方法について述べ、それを画像描画処理に応用した例について述べる。

2. サービスプロセスとインタラクション

ソフトウェアのサービスプロセスを考えると、それはクライアントとしてのユーザーのある要求意図がサーバーとしてのシステムに送られ、システムは要求意図を解釈し、その後クライアントとサーバーの協調的なやりとりによって意図が段階的に決定されていくプロセスと考えられる。意図はシステムの用意する種々のサービスから選択する形で詳細化し、詳細化された意図は段階的に達成されるものとしてユーザーにオブジェクトネットワークが呈示される。段階化された意図の一つ一つは単位意図として、ユーザーがそれらを一一つ達成することで意図が達成される。(図1)

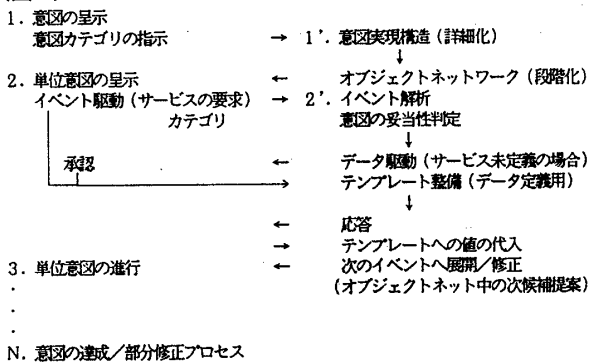


図1 意図の段階的決定プロセス

この関係は並行処理を行うシステム間のリクエスト-レスポンドの関係についてもそのまま同様のことがいえる。

このようなサービスプロセスにおけるシステムとユーザー間、あるいはシステム間の関係をインタラクションとして定義し、協調的かつ並行的なシステムのシステム表記を行う。

3. インタラクティブ仕様による

ソフトウェアシステムの仕様表現ソフトウェアシステムのインタラクティブ仕様は、前節のシステム間、あるいはシステム-ユーザー間のインタラクションを仕様として自然言語風に記述するものである。以下インタラクションの表現の特徴を挙げる。

a) ドキュメントの独立/自己完結性

インタラクティブ仕様はシステム、ユーザー各々に対して仕様ドキュメントとして作成される。システムが複数で並行処

理を行っている場合にはドキュメントもそのシステムごとに作成される。

その各々のドキュメントはそれを読むことのみによってそのシステム間、あるいはシステム-ユーザー間の振る舞いが分かるように記述されており、自己完結する。

b) 状態によるシステム間の仲介

図1のような意図の段階的決定プロセスを考えると、まずサービスはユーザーのある意図により行われる動作によってシステムに送られる。その動作は相手のシステム側のレスポンドとしての動作を引き起こすが、この間のインタラクションを仲介するものとして状態を考え、その状態を満たせば動作が起こるものとする。ユーザー側でもシステムからのレスポンドを状態として受けとめ再び動作を行う。

システム間の協調は、下図のように意図→動作→状態→動作→状態...と各々のシステムの動作が状態を仲介してなされるものとする。(図2)

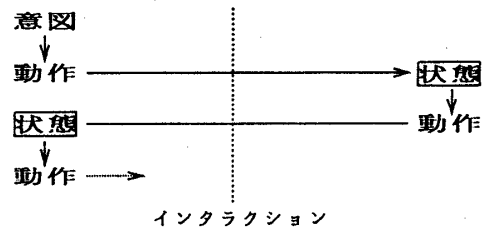


図2 状態によるシステム間の仲介

c) 意図、状態、動作の表現

前節の意図、状態、動作は、具体的には接続詞Whenを用いて、意図-動作、状態-動作を従節-主節として以下のように表記する。

- ・意図→動作
When the user intends to do ~, (意図)
the user does ~. (動作)
- ・状態→動作
When the user is doing ~ (状態)
受動態
the user does ~. (動作)

4. カラー画像描画システムのインタラクティブ仕様表現

協調型並行システムの例として、画像システム記述言語(WELL-PPP⁽¹⁾)上でインプリメントされたカラー画像描画システムを取り上げ、インタラクティブ仕様によるシステム表現を行う。

このシステムは大きく二つのシステム-Frame Section, Color Section-から成り立ち、そのセクション間で協調的に処理を進めていくことで一枚の絵(ピクチャ)の描画を行うものである。二つのセクションは異なるシステム(WSなど)の間で独立して稼働することも考慮されている。

ユーザーの描画するという意図は、クラスによって階層化されたオブジェクトネットワークとしての要素ネットワークのディスプレイ上への呈示によって段階的に達成される。ユーザーは単位意図の達成のためにクラス名をマウスでヒットし、そのクラスのオブジェクトの描画を行う。そして段階的に上位のクラスを生成していくことで最終的に一枚の絵を完成させる。

インタラクティブ仕様ではシステム間の協調関係、あるいはシステム-ユーザー間の協調関係をインタラクションとして仕様表現する。このシステムの場合、自己完結した3枚(ユーザー, Frame

A description of specification for an interaction between systems, and its application to picture painting system
Yuji HASHIMOTO, Yo MURAO, Hajime ENOMOTO
Sibauro Institute of Technology

Section, Color Section)のドキュメントによってインタラクシ
ョンが表現される。

5. インタラクティブ仕様による表現例

図3がインタラクティブ仕様による画像描画システムの表現例
である。以下、ドキュメント各構成部について説明する。

a) Configuration部

まず最初に3枚のドキュメントに対するConfigurationとそれ
に対する制約条件、及びこのドキュメントで使用される動詞が
定義される。

Configuration部ではオブジェクトネットを構成するクラスと
そのクラスに対するオペレーションが関数として記述される。

制約条件部においては状態としてのダイナミッククラスの宣
言や、クラス間の関係が記述される。

この二つの部分によって、オブジェクトネットワークがテキ
ストとして表現される。

動詞はここではほとんどの動詞がシステム内に定義されてい
るものとしているが、オブジェクトのidentifyに関してはユー
ザーのマウスのhittingによって定義されている。

b) ユーザー側の記述

ユーザーはPainting Serviceを受けたいという意図をシステ
ム側に伝えるところから始まる。その後、単位意図の達成のた
めにシステムからメッセージウインドウに表示されるメッセ
ージを認識して、データのput、オブジェクトのidentifyを行う。

c) Frame Section と Color Sectionの記述

各々のシステムのドキュメントには基本的にテンプレートの
defining processが記述される。ユーザーのクラス名のhitによ
りテンプレートがdefining processに入り、ユーザーのデー
タウインドウ上でのデータのputによりdefining operationが実行
される。

またそのクラスのテンプレートの準備ができない時などは、
前のクラスに戻るために、ダイナミッククラス名を用いてシス
テム内の状態の変化を表現している。

Color Sectionでは、Frame Sectionのオブジェクトデータに

色情報等の属性を付加するものゆえ、そのオブジェクトに対す
るidentify動作が記述され、システム間の協調作業が表現され
る。その動作条件としてFrame Sectionからのデータが送られて
きていることが記述される。

6. インタラクティブ仕様と並行手続き

インタラクティブ仕様のConfiguration部に記述された関数は総
称的動詞オブジェクトとして定義され^[2]、各々のシステムの仕
の内容がドキュメントとして表現される。

このドキュメントは、本研究室で開発されたソフトウェア環境
TELL-SCP^[3]におけるスペックに落とすことができる。例題のよ
うな並行処理システムはTELLスペックのドキュメント中ではミッ
ションとして表現され、ソフトウェアが実際に実行可能な形で並行
手続きを導出することができる。

またこれらWord Specのドキュメントに対し、ある一般的な解析
規則を適用することで、こうした並行システムの図式表現(拡張
ベトリネットによる表現^[4])を導くことができる。

7. まとめ

ここでは、協調型並行システムのシステム間の関係をインタラ
クションとして定義して、システム個々の動作表現と合わせるこ
とで、あるシステム全体のインタラクション統合が可能になるこ
とを述べた。この仕様化によって各々のシステムを独立に表現で
き、かつそのドキュメントを突き合わせればその中にシステムの
インタラクションを表現できる。

文 献

- [1] 丹羽、守屋、関、村尾、榎本：“並行実行型画像システム記述言語(Concurrent WEL-PPP)の
実現、情報処理学会第46回全国大会、1993.3
- [2] 川野辺、榎本、村尾、榎本：“総称的動詞オブジェクトによる画像システムの仕様記述”、
情報処理学会第46回全国大会、1993.3
- [3] 森谷、村尾、榎本：“内包的テンプレートによるソフトウェア記述プロセスの実現”、情報
処理学会第42回全国大会、1991.3
- [4] 守屋、村尾、榎本：“協調型画像描画システムの記述形式”、情報処理学会第46回全国大会、
1993.3
- [5] 榎本、朝志田：“分界型記述言語の調査”、情報処理学会第44回全国大会、1992.3

Document for the system for 'Painting Service':

```
Configuration:
Component Network for 'Painting Service':
For 'Frame Section'
Class : 'None', 'Colored Point', ..... 'Colored Region'
Function : Draw_up(None) = Colored Point, ..... Paint(Colored Region Seg.) = Colored Region

For 'Color Section'
Class : 'None', 'Point', 'Point Seq.', ..... 'Region'
Function : Draw_up(None) = Point, ..... Connect(Region, Seg.) = Region

CL1 is a generic name of all classes of 'Frame Section'.
CL2 is a generic name of all classes of 'Color Section'.
```

Constraints of configuration:

```
When CL1 is used as a state data, CL1 is recognized as
a dynamic class of CL1.
When CL2 is used as a state data, CL2 is recognized as
a dynamic class of CL2.
.....
```

Definition of verbs:

```
"prepare" is implemented in the system.
"assign" is implemented in the system.
"hit" is implemented in the system.
"show" is implemented in the system.
"put" is implemented in the system.
```

```
.....
"identify" means hitting the button of the mouse
on an object.
```

The user:he

```
When he intends to request 'Painting Service',
he hits 'button' on 'Operation Window'.

When 'C.N' is shown to him on 'Op.W',
he hits a class:CL1 in 'C.N' on 'Op.W'.

When the message: 'A definition of CL1 is possible.'
is shown to him on 'Message Window',
he puts a datum of the class:CL1 on 'Data Window'.

When the message: 'An identification of object of CL1
corresponding to CL2 is possible' is shown to
him on 'Ms.W',
he identifies the object of the Class CL1 on 'D.W'

When the message: 'A definition of CL2 is possible.
is shown to him on 'Message Window',
he puts a datum of the class:CL2 on 'Data Window'.
```

The system for 'Frame Section':it

```
When 'button' is hit on 'Op.W',
it shows 'Component Network' on 'Op.W'.

When the class:CL1 in 'C.N' is hit on 'Op.W',
it assigns CL1 as the dynamic class to the state in
itself.

and
if the datum of the class previous to CL1 is assigned
to the template for the class previous to CL1,
it prepares a template for CL1,
and
it shows a message: 'A definition of CL1 is
possible' on 'Ms.W'.

else
it shows a message: 'A datum of the class previous
to CL1 is not assigned to the template.' on 'Ms.W',
and
it assigns CL1 as a d.class to the previous state
in itself.

When the datum of CL1 is put on 'D.W',
it assigns the datum to the template for CL1,
and
it transmits the datum to CL2 corresponding to CL1.
and
it shows 'C.N' on 'Op.W'

When it is requested to assign the datum of CL1,
it assign CL1 as a d.class to the state in itself.
```

The system for 'Color Section':it

```
When the class:CL2 in 'C.N' is hit on 'Op.W',
it assigns CL2 as the d.class to the state in itself.
and
if the datum of the class previous to CL2 is assigned
to the template for the class previous to CL2,
and
if the datum of CL1 corresponding to CL2 is transmitted,
it prepares a template for CL2,
and
it shows a message: 'An identification of object of
CL1 corresponding to CL2 is possible' on 'Ms.W'.
else if the datum of the class previous to CL2 is not
assigned,
it shows a message: 'A datum of the class previous to
the class:CL2 is not assigned to the template.' on
'Ms.W',
and it assigns CL2 as a d.class to previous state
in itself.
else
it requests to assign the datum of CL1 to 'F.Sec'.

When the object of CL2 is identified on 'D.W',
it shows the message: 'A definition of CL2
is possible' on 'Ms.W'.

When the datum of CL2 is put on 'D.W',
it assigns the datum to the template for CL2.
and it shows 'C.N' on 'Op.W'.
```

図3 カラー画像描画処理システムのインタラクティブ仕様の例