

Extensible WELLのインタラクティブ仕様と それによる実行プロセスの導出

4G-10

橋本雄治、村野剛、古里剛一、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は、クライアントとサーバー間のインタラクションを繰り返して進行する。ソフトウェアプロセスの仕様表現において、インタラクションを明確に定義するインタラクティブ仕様がある。ここでは拡張機能言語Extensible WELL^{[1][2]}より構築されるシステムにおけるクライアントとサーバー間のインタラクションを例に仕様表現を行う。仕様は総称的表現がなされており、Denotational schemaに基づいた部分規則を段階的に適用することにより詳細化が進められる。このプロセスにより具体的な実行レベルでの仕様が関数の系列として導出されるものと考えられる。

2. インラクティブ仕様とDenotational schema

協調実行型システムのソフトウェアプロセスにおいては、クライアントとサーバーの共通認識の場、Common Platformを媒介としたインタラクションが存在する。

またここで、ソフトウェアプロセスは、サービスの要求意図を単位意図に分割しモジュール化したオブジェクトネットワークに沿って、逐次的にクライアントが満足することで進行する。

インタラクティブ仕様は、このインタラクションを仕様として明確に定義するものである^[3]。この仕様は、上述のプロセスを反映した表現を行うため、表現形式がDenotational schemaとして標準化されている（図1）。この中で、動詞は型が分類されており^[4]、スキーマもそれに従う。記述の型に対し、仕様中の名詞オブジェクトのデータ型（属性構造）はData schemaで定義される^[5]。実行時、サーバーはオブジェクトをData schemaの形で認識する。

インタラクティブ仕様は、総称的表現によって与えられる。そして、ある分野に対応したオブジェクトネットワークが与えられたときに、変換規則を適用し、Denotational schemaに沿って詳細化され、具体的表現が導出される。

3. インラクティブ仕様の構成

インタラクティブ仕様の構成、以下のように大きく二つに分けて考えられる（図2）。

3-1. Common platform上のオペレーション定義

仕様の基本フォームとして、クライアント側に、サーバーに対しあるサービスのあるウィンドウ(Common platform)を通じて要求する文、サーバー側に、指定されたオブジェクトに対しあるアクションを行うことでその要求に答える文を総称的に与える。これらの文に対して、ある特定分野のオブジェクトネットワークを具体的に与

```

<Interactive specification> ::= <Interactive sentence>
  | <Interactive sentence> <Interactive specification>
<Interactive sentence> ::=
  <When clause>, <Sentence that represents action>,
  *インタラクティブ文は、When節を持つ。
<When clause> ::=
  When <Subject> <Verb phrase that uses passive form>
  *受動態の動詞で状態を表す（満足すれば主節を実行）。
<Sentence that represents action> ::=
  <Subject> <Verb phrase that uses active form>
<Verb phrase that uses active form> ::=
  <Verb> <Object> | <Verb> <Object> <P.p phrase>
  ↑ 制約演算付加
  *主節の動詞は動詞の型の分類に従ってスキーマが定まる。
<P.p phrase> ::= <Preposition> <Noun phrase>
<Noun phrase> ::= <Article> <Noun>
  | <Article> <Noun> <P.p phrase>
  | <Article> <Passive verb> <Noun>
  | <Article> <Passive verb> <Noun> <P.p phrase>
  *制約演算は動詞の受動態（状態）を用いる。

```

図1. インラクティブ仕様のDenotational schema (一部)

えることで具体的な双方のCommon platform上でのオペレーションとオブジェクトのデータ定義の表現が導出される。

3-2. Extensible WELLシステム内部の定義

サーバーとしてのシステムは、Common platform上のオペレーションを解釈方法が定義されていなければならない。ここでは2章のソフトウェアプロセスを反映するExtensible WELLシステムを構成するモジュール間のインタラクションの定義である。

サーバーとしてのExtensible WELLシステムを記述することで、クライアントのCommon platform上でのオペレーションがどのように解釈されるかを定義する。具体的には、Extensible WELLシステムを構成するカーネル、各マネージャ間のインタラクションが定義される。

4. 仕様の詳細化による実行プロセスの導出

実行プロセスの導出とは、総称的関数（動詞）で表された表現が、仕様の詳細化を経て、最終的にプリミティブな関数の系列で表されることである。前節にある二つの仕様からインタラクティブ仕様全体がどのように詳細化されていくかを示す。

4-1. Common platform上のオペレーションの詳細化

まず、3-1.に対応する部分の詳細化過程を示す（図2のAの丸数字に対応）。仕様文のある語句に対し、変換のための部分規則が逐次適用することで詳細化される。

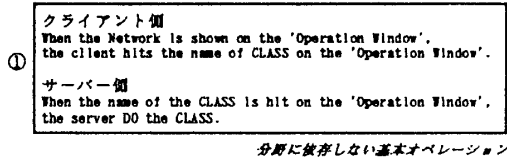
① 2章のスキーマにしたがって、クライアント側、サーバー側のインタラクションを表す文を与える。

どのネットワークにおいても共通するものと考えられるので、この部分を総称的に記述することにより図2のような基本フォームが与えられる。

Interactive specification for Extensible WELL and generation of execution process
Yuji HASHIMOTO, Takeshi MURANO, Koichi FURUSATO, Yo MURAO, Hajime ENOMOTO
Shibaura Institute of Technology

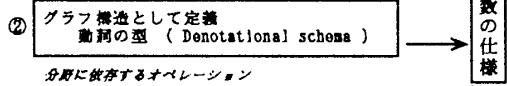
A. Common Platformを媒介とするオペレーションの定義

・基本フォーム (Common Platform上での基本オペレーション)



分野に依存しない基本オペレーション

・ネットワーク定義

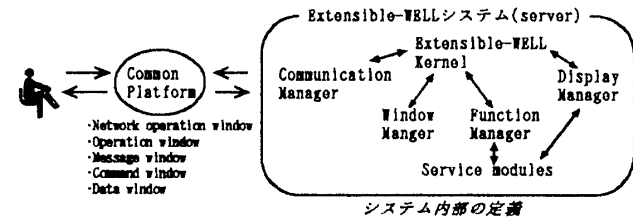


分野に依存するオペレーション

B. Common Platform上のオペレーションの解釈定義

・Agentとしての'server'の詳細化

Extensible WELLシステム内部のカーネル、及び各マネージャー間のインタラクション



システム内部の定義

図2. インタラクティブ仕様の構成

規則としては、スキーマのどの型を採用し、その中にどのような単語を割り当てるかが指定される(すなわち、文全体の単語をすべて与える。)ネットワーク名、クラス名は仕様の環境変数を用いて総称表現する。

② ①で与えられたインタラクティブ仕様の基本フォーム、とくにサーバー側に対して、変数部分に対する部分規則を与え、仕様が詳細化される。この詳細化過程では、変数名への値の代入と、動詞についてはその値による型の決定を行う。

動詞の型は分類されており、与えられたネットワークに依存するものである。またクラスと関数も、ネットワークに依存してグラフ構造として定義されているものである。これらのことは、ネットワークのconfigurationとして定義される。

③ ②で表現されたactionの詳細化を行う。関数の中には、動詞の型に基づき制約が付加されるものが存在する。この場合関数は、順序性を持つ関数の系列として詳細化される。例えば、点に色をのせるというactionの場合、オブジェクト指定→色選択→(Data schemaに基づく)色づけという動作の系列がこの順で存在しなければならない。

関数の仕様は、Extensible WELLシステムのDefining process, Define operationという実行過程が図1と同じスキーマで定められ、プリミティブな関数の系列として定義される。詳細化はこの関数の仕様の特定のスキーマに基づき記述された文をスキャンすることで行われる。

4-2. Extensible WELLシステムの仕様と詳細化

4-1. で仕様としてはクライアントが認識できる場、Common platform上のすべてのオペレーションが導出されるが、実際の実行プロセスを表現するためにはサーバーとしてのシステム内における要求の解釈の方法が具体的に導出されねばならない。以下の文において、

When a point is hit on the 'Data window',
the server identifies the point.

主節の主語(agent)である'server'によるオブジェクトのidentificationは、実際にはシステム内の各モジュール内で、座標値の認識、認識番号のscanning等を行うことである。この文は、モジュール名が部分規則により変換され、以下のような文に詳細化される。

When a point is identified on the 'Data window',
the 'Window manager' transmits the coordinate value of the point to the 'Kernel'.

When the coordinate value of the point is transmitted,
the 'Kernel' scans the I.D._number of all CLASS,
and transmits the I.d._number of the CLASS to the
'Communication manager'.

When the I.D._number of the CLASS is transmitted,
the 'Communication manager' transmits the I.D._number
of the CLASS to the service module for the point.

When the I.D._number of the point is transmitted,
the service module
(サービスモジュールによる処理。省略)

5. まとめ

あるソフトウェアプロセスの仕様記述において、クライアントとサーバー双方の共通認識の場を仕様の一つのモジュールとしてインタラクティブに表現するために記述の標準形式を定めた。これは、総称的に与え得るものであり、ある特定の分野のオブジェクトネットワークが与えられれば、部分規則の適用によって逐次詳細化を進め具体的実行手順が得られる。

またこのことは、ソフトウェアプロセスを形式的論理的に表現し得ることを示したもので、今後ペトリネットを使用した視覚的なプロセスの把握に応用できるものと思われる。

文献

[1]Y.Miyamoto, H.Ino, Y.Murao, H.Enomoto, Y.Moriya, M.Kamoshida: 'Integration of picture painting process by domain specific extensible system', Visual Communications and Image Processing '93, SPIE Vol.2094/1400-1411, Nov. 1993.
[2]宮本、平井、青木、村尾、榎本: "拡張機能言語Extensible WELLの体系化", 第48回情報処理学会全国大会, 4G-08, 1994.3
[3]Y.Hashimoto, T.Murano, Y.Murao, H.Enomoto: 'Interactive specification and data schema for picture painting process', Visual Communications and Image Processing '93, SPIE Vol.2094/1412-1423, Nov. 1993
[4]榎本、鴨志田: "分野記述言語の構造", 第44回情報処理学会全国大会, 6F-03, 1992.3
[5]村野、橋本、宮田、戸川、村尾、榎本: "画像生成におけるデータスキーマ", 第48回情報処理学会全国大会, 7U-05, 1994.3