

対話的意図処理システム実現のための仕様記述

2C-3

増田 征貴 村尾 洋 榎本 肇

芝浦工業大学

1 はじめに

ユーザの要求は、環境に応じて動的に変化する総称的 (generic) な意図に基づいて行われ、その実現には意図をサービス提供可能な状態まで動的に具体化する事が必要である。我々が研究・開発を進める Extensible WELL (Window-based Elaboration Language)^[1] に基づく対話的システムでは、意図の構造^[2] を明確にし、様々な要求に対応する。その構造はネットワーク状に階層的に表現され、それを論理仕様として定義する事でシステム構造を可視化し、意図処理過程の構造をも明確にする。

2 Extensible WELL の言語体系

人間は、主に自然言語を用いて円滑な意思伝達を行ない、相互のコミュニケーションを実現する。従って、人間とコンピュータ間でも、自然言語を使える事が望ましい。WELL は、自然言語に極めて近い表現が成されている。これに基づいて4つの総称的モデル (Data-, Object-, Role-, Process Model)^[1] を実現し、協調的なサービスが提供可能なシステムを構築する。このシステムでは、Client 及び Server が Common Platform を介して多重対話^[3] を繰り返し、サービスの設計から実現までが階層的に実行される。

また、本システムは、Object Model 中の Object Network を基盤として構成され、モデルの仕様定義により、構造の明確化、実行状態の可視化を行なう事が出来る。それにより、複雑な実行処理過程やそれを実現する効率的なソフトウェア構成を明確にする事が可能となる。つまり、意図に基づく総称的かつ動的な要求も、仕様によってその構造及び実現過程を明確に表現する事が可能となり、それをシステムとして構築する事ができる。従って、WELL の言語体系は以下の3つ、

1. WELL は、準自然言語により記述される。
2. 各種サービスの実行過程は、グラフ構造を用いた Object Network により表現される。
3. 単文形式により論理仕様が記述される。

を基盤としてシステムを構成する。また、これらは各々一対一対応する (図 1)。1.と2.の関係は、名詞/動詞を用いて単文により表現される WELL の記述に対し、それらをサービス実行の流れと結びつけるために、グラフ構造を用いてネットワーク状に Object Network を表現する。2.と3.の関係は、Object Network が単文に対応して構成され、サービス実行過程を表現する事に対し、論理仕様も単文によってシステム構成及びサービス実行過程を可視的に表現する。3.と1.の関係は、論理仕様と WELL の記述を対応させるために、自然言語と対応性のある Montague 流の内包論理を用いて、限定された文法構造と対応する仕様を単文により箇条書形式で定義する。これら3つの対応付けは、図1で示すように、サービス設計プロセスで行なわれ、この過程で

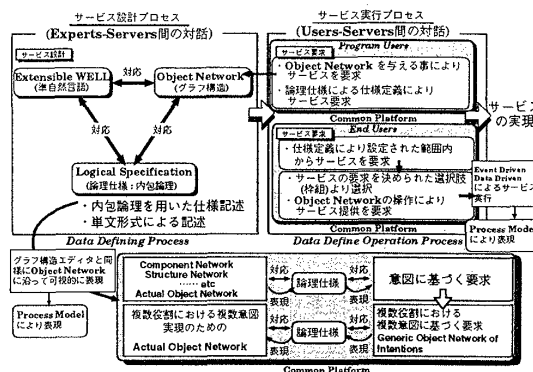


図 1: Extensible WELL の言語体系

は、仕様定義に基づいてテンプレートを設定し、サービスを実行可能な状態にするために、その設計を行なう。

3 意図構造

サービスは、Client 及び Server の意図に基づいて実現され、それらの意図をシステムに忠実に反映させる事は、ソフトウェア効率の有効性に継る。しかし、その意図が複数存在し、相互に影響を及ぼす場合、それらを解析し、システムに反映させる事は非常に困難である。ここでは、サービス実現までの意図の定義過程を以下のように階層的に捉え、その構造を明確にする。

1. 対象領域の決定。対象としての環境及び当事者 (サービス中の対象) やその構造、役割機能を規定する。
2. 意図の性質を分類。「独立的意図」「共通の意図」「相反的意図」の3つの性質に分類する。
3. 支援役割機能の決定。先に決定された役割機能を支援する機能を決定する。(e.g. 環境認識機能)
4. 戦略の決定。1,2で決定される制約に基づいて決定される。制約集合として表現される。
5. 戦術の決定。戦略として設定された制約集合に基づき、ユーザの具体的操作を決定する。

この過程は、総称的意図を明確にし、サービス実行可能な状態まで具体化を行う。本システムでは、サービス実現のために、この構造に基づいて意図の総称的 Object Network (図 2中) を生成し、その具体化を実現する。

4 意図の実現手法

意図に基づく要求は、意図の Object Network に従い、Common Platform 上で Client-Server 間の多重対話を繰り返し、意図を具体化する事により実現される。

4.1 対話プロセス

対話プロセスは、サービスの設計から実行までが以下のように分類され、各 Client (Expert, User) の意図を実現する。

- サービス設計プロセス (Expert-Server 間の対話):
 - サービス準備プロセス:仕様定義及び解析。
 - サービス定義プロセス:テンプレート設定。
- サービス実行プロセス (User-Server 間の対話):
 - サービス項目準備プロセス:テンプレート準備。
 - サービス項目定義プロセス:データ定義。

これらの対話プロセスは、サービスの設計から実現までが階層的に Common Platform 上で実行され、その実行過程は、各々の過程を全て 2 つの駆動形式 (イベント、データ駆動) で同様に実行する事ができる。この対話は、Client と Server の相互のコミュニケーションを実現し、総称的要求の具体化を可能とする。この対話は、意図に基づくサービス要求も、設計から実現まで階層的に実行し、多様な総称的意図を具体化する。そして、要求に応じたサービスを動的に実現する。

4.2 意図の具体化プロセス

意図に基づくサービスは、その意図の具体化によって実現される。本システムでは、その具体化の過程は、総称的なものから具体的なものへの Object Network の変換過程として表現される。図 2 に示されるように意図

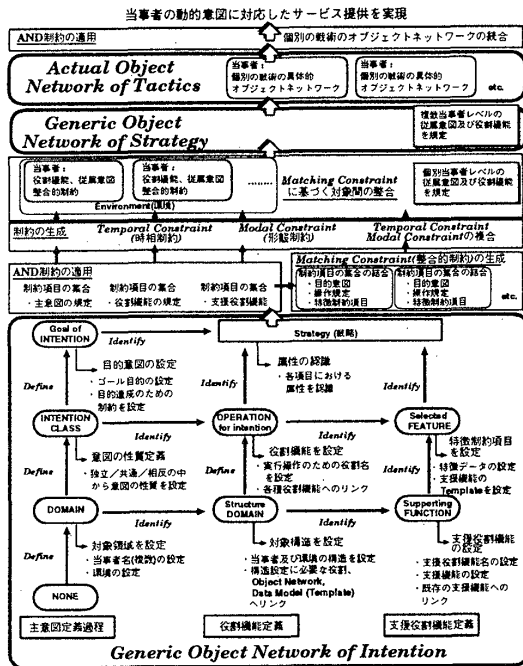


図 2: 戦略・戦術を用いたサービス設計及び実現過程

→戦略→戦術の順に総称的/具体的 Object Network に変換し、意図の具体化を実現する。そして、総称的かつ動的な意図に対応可能なサービスを Object Network として設計し、それをユーザが操作する事により、意図に基づく要求を実現する。これらの過程は、最終的に対象領域全体の意図を整合し、それを実現できる形で個々の当事者ごとに設計される。

図 2 の過程は、制約定義により様々なサービスの方向性を設定する。それにより、システムは様々な環境に応じて意図の具体化、サービス提供を動的に行なう事が可能となる。この制約定義は図 2 のような各オブジェクトに対して定義される仕様に基づいて行なわれる。

5 仕様記述

総称的意図や総称的モデルは仕様記述によって定義を行なう事が出来る。Extensible WELL では、テンプレート、関数、Object Network、役割等システムの構成要素全てを総称的に表現し、様々な要求に対して全て同等の概念で対応する。サービスは、これら総称的要素の具体化により実現し、その過程も仕様によって表現する事が可能である。また、制約条件^[4]の整合的パラメータ値を与える事により、この具体化を実現する。

本言語の仕様は、Object Network に一対一対応し、内包論理を用いた単文形式で論理仕様として表現され、準自然言語で記述された WELL の書式に対応する。この仕様を以下の事に基づいて定義する。

1. 総称的モデルを階層的に定義する。
 - Data-, Object-, Role Model: 静的表現。
 - Process Model: 動的表現。
2. 総称的/具体的 Object Network 双方に対応可能な同等の表現。
3. 制約条件の整合的パラメータ定義により総称的表現の具体化過程を表現。
4. WELL が準自然言語で表現されるため、内包論理を用いて容易に論理仕様表現が可能である。
5. この仕様定義によりサービスの実行状態を明確に表現し、システム構造の可視化を行なう。

上述のように、仕様は具体的なサービスの実行過程のみではなく、Object Network の総称から具体化への変換過程をも可視的に表現する事ができる。つまり、総称的な意図も図 2 のように、意図の総称的 Object Network に沿った仕様定義を行う事で、具体化が実現される。また、この仕様定義は、制約定義及びテンプレート設定が中心に行なわれ、動的な意図に対応可能なシステムの構造を明確にする事ができる。

Object Model の仕様の一例を自然言語と論理仕様で表すと 'DOMAIN' の定義過程は以下のようになる。

```

DOMAIN = Define(NONE)
'DOMAIN' has value of I.D.,
domain_name, component_name,
environment_name in the
template.
'NONE' has empty value of cell
in every template.
'Define' is a generic_function.
'Define a domain' is a actual
function: defining a constraint.
- A domain_name is defined.
- A component_name is defined.

DOMAIN' =
[[APAx(P(x))](NONE)](Define')
['has('N('value('in(' APEx[Vx[
template'(x)→x=y]AP(y))]]))]]
(DOMAIN')
N = {I.D.,domain_name'
component_name',environment_name'}
[-has('cell('value('in(' APEx[Vx[
template'(x)→x=y]AP(y))]]))](NONE')
[AzEx(G_func'(x)Az=x)](Define')
[AzEx(A_func'(x)Az=x)](Define a
domain')
['define('AQEx[constraint'(x)AQ(x)]
(A_func')
[-Q(x)](A_func')
- 'define('AQEx[component_name'(x)
AQ(x)](A_func')

```

これら仕様の明確な定義により、サービスの実行状態を表現する事を可能とし、同時に、総称的モデルの定義により、システム構造を明確にする事を可能とする。それにより、総称的意図の動的処理過程を明確にする事ができる。これらの構造が準自然言語に対応性のある論理仕様により定義される事で、システムの論理構造をも明確にし、可視化を行なう事が可能となる。また、このような WELL システムの核部分の動作は、当事者概念によりこれらを定義する事で明確になる^[5]。

6 まとめ

今回の概念により、多種多様な意図に動的に対応し、サービスを提供するシステムが実現され、そのソフトウェア構成も明確にする事が出来た。これにより、動的な意図に応じた多様なメディアの統合や現実的なマルチメディアコンテンツの効率的生成が可能となる^[5]。また、意図、対話、モデル、サービスの実行処理が全て階層的に表現される事から、システム構成が明確かつ効率的なソフトウェアを体系化する事ができた。

文献

- [1] H.Enomoto, Y.Murao, "Interactive realization system of visual reality using hierarchical model driven concurrent processing", Proc. Electronic Imaging '98: Science & Technology, IS&T/SPIE, Vol.3295, 1998.1
- [2] M.Masuda, Y.Murao, H.Enomoto, "Dialogic generation system of realistic multimedia contents corresponding to dynamic intentions", Proc. Electronic Imaging '99: Science & Technology, IS&T/SPIE, Vol.3639, 1999.1
- [3] 増田, 村尾, 榎本, "Extensible WELL における対話プロセスの表現", 情報処理学会第 56 回全国大会, 1998.3
- [4] 榎本, 村尾, "モデル駆動とプロセス間結合", 情報処理学会第 52 回全国大会, 1996.3
- [5] 太田, 村尾, 榎本 他, "意図・戦略・戦術のオブジェクトネットワークによるコンテンツの動的生成システム", 情報処理学会第 58 回全国大会, 1999.3