

戦略的ソフトウェア研究フォーラムSSR  
「次世代ソフトウェアアーキテクチャ」  
ワークショップ

2000. 1. 25

## エージェント と コンポーネント

中所武司  
明治大学理工学部情報科学科

### 提案書：位置付け

#### ■社会的ニーズ

- ・ BtoB, BtoC 対応の新しいソフトウェアモデル
- ・ ユーザニーズの変化に応じた動的サービス提供

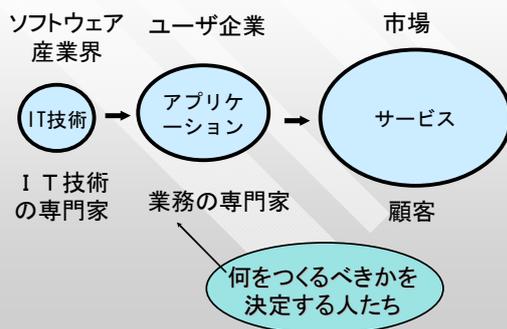
#### ■技術課題

- ・「ダイナミック ソフトウェア アーキテクチャ」

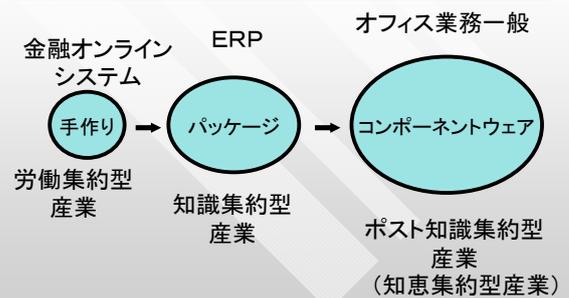
#### ■1つの研究アプローチ

- ・エンドユーザ主導のダイナミズムの実現

### 変化への対応とは



### 変化への対応：過去・現在・未来



### 絶えざる変化への対応

#### ■1つの理想形

- ・コンポーネントのダイナミックな差し替え

#### ■技術課題

- ・コンポーネントベースのアーキテクチャ
- ・コンポーネントの開発・流通機構

#### ■もう1つの理想形

- ・コンポーネントとしての適応エージェント

### エージェント技術の現状

#### ■エージェントの定義

- ・??

#### ■エージェント技術の標準化動向

- ・FIPA (Foundation for Intelligent Physical Agents)
- ・OMG Agent WG

#### ■エージェント応用

- ・EC + XML + Agent

## エージェントの定義(1/2)

### ■IEEE CS 解説論文

環境を知覚し、推論し、単独および他のエージェントと協調して行動する永続的計算 (by M. P. Singh)

### ■某AI研究者

observe-think-act cycle model  
(by R. Kowalski: observationの重要性和autonomyを強調)

### ■IPSJ 解説論文

自律性;協調性;持続性;学習・適応能力;  
代理性;移動性;(by 西田豊明)

## エージェントの定義(2/2)

### ■FIPA ACL (エージェントの定義はしない方針)

ソフトウェアエージェントの共通属性:

- ・ゴール指向 ・自律的決定 ・交渉・委任による交信
- ・擬人化された精神活動(信念, 意図, 願望)のモデリング
- ・立場とニーズに応じた柔軟性;

### ■OMG Agent WG

目的:

エージェント技術に共通の意味, メタモデル, 抽象構文を採用し, 再利用, 相互運用, 携帯可能なアプリケーション コンポーネントとして表現する.

## エージェントへの視点(1/2)

### ■ エージェント:

ドメインに近いコンポーネント(ビジネスオブジェクト)

### ■マルチエージェント:分散協調コンポーネント群

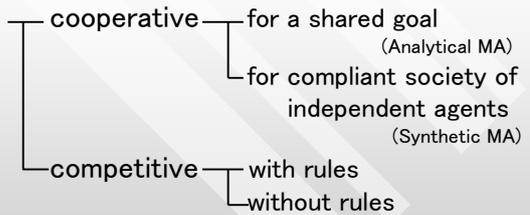
### ■特定ドメインのエージェント間対話インタフェース :デザインパターン

### ■エージェント間対話言語ACL:協調関係を規定 (Agent Communication Language)

### ■ JavaBeans, EJB, DCOM, CORBAなどの コンポーネントモデルはインフラ/実装技術

## エージェントへの視点(2/2)

### ■マルチエージェントの定義(分類)



### ■AI or SE (人工知能 or ソフトウェア工学)

AIの視点ではなく, SEの視点で, ACLに注目.

## ACLの経緯

### ■ KQML(DoD ARPA, 1980年代後半-)

- \* Knowledge Query and Manipulation Language
- \* 開発G: ARPA Knowledge Sharing Effort
- \* 目的: 共有・再利用可能な大規模知識ベース構築
- \* 機能: 事実通知, 問い合わせ, サービス予約など.  
例:(tell :sender A :receiver B :content "raining")
- \* 内容表現: KIF(Knowledge Interchange Format)
- \* 欠点: ヘテロでない. 同質のエージェントとのみ交信.

### ■FIPA ACL(FIPA, 1997-)

### ■OMG Agent Working Group(OMG, 1998-)

## FIPA ACLの概要

### ■ 文献

- \* Agent Communication Language:  
FIPA Spec 2-1999, DRAFT ver.0.1, p.97, 16<sup>th</sup> Apr. 1999.  
(現在 DRAFT ver.0.2, p.112 ?)

### ■目的

- \* 仕様: 抽象構文の規定(具体化は自由)
- \* 範囲: 実装独立のメッセージ通信  
(CA[Communicative Acts]で表現)

## FIPA ACLのエージェント間通信の概要

### ■仕様の目標

- \* できるだけ広範囲の応用実現のため、できるだけ厳密にコンポーネントを定義。

### ■仕様の範囲

- \* 精神活動も対象: 信念(真偽の命題)、不確実性、意図(命題を真にしたい)。

### ■メッセージ転送機構

- \* エラー処理, 非同期, 内部形式と転送形式(文字列)の相互変換, など。

## FIPA ACL メッセージの概要

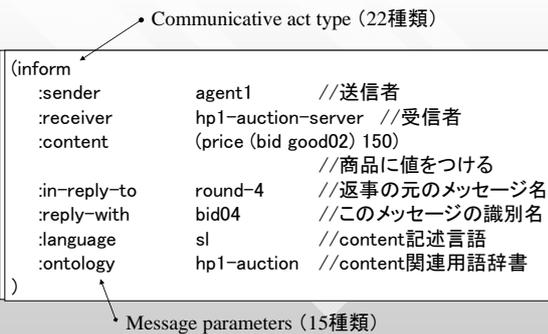
### ■設計目標

- \* 完全性(広範囲応用性)
- \* 単純性(特殊機能排除)
- \* 簡潔性(曖昧性・冗長性排除)

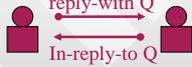
### ■要件

- \* 理解できないメッセージへのリプライ
- \* 正しい実装
- \* 標準機能の別名禁止
- \* プロトコル厳守

## FIPA ACL メッセージの構造



## 15種類のメッセージパラメータ(1/2)

- :sender :メッセージ送信元のエージェント名
- :receiver :メッセージ送信先のエージェント名(タプル可)
- :content :内容(本文)
- :reply-with :このメッセージの識別名 
- :in-reply-to :返事の元のメッセージ名 
- :envelope :メッセージ転送サービスで利用する情報
- :language :内容(本文)記述に用いた言語
- :ontology :内容(本文)関連用語辞書

## 15種類のメッセージパラメータ(2/2)

- :reply-by :返答期限
- :protocol :利用するプロトコル
- :originator :要求/返答メッセージの元々の送信者
- :reply-to :返答先
- :protocol (プロトコル名のネスト) :最左端が最深のもの
- :conversation-id (会話識別子のネスト) :上記に対応
- :conversation-id :一連の会話の識別名

## 22種類 (5グループ) のCAs(1/3)

### ■<情報伝達(5)>

- confirm :情報(事実)の確認
- disconfirm :情報(事実)の否定
- inform :情報(事実)の通知
- inform-if :情報(事実)の通知要求
- inform-ref :同上(特定項目に関して)

### ■<情報確認要求(3)>

- query-if :情報(事実)の真偽の問い合わせ
- query-ref :同上(特定項目に関して)
- subscribe :情報(事実)の真偽の変化の通知依頼

## 22種類（5グループ）のCAs(2/3)

### ■ <実行行為(8)>

- agree : 実行依頼への承諾
- cancel : 実行依頼の取り消し
- propagate : 転送依頼
- proxy : 同上
- refuse : 実行依頼の拒絶
- request : 実行依頼
- request-when : 条件付実行依頼(条件成立時に1回)
- request-whenever : 同上(条件成立毎に繰り返し実行)

## 22種類（5グループ）のCAs(3/3)

### ■ <交渉(4)>

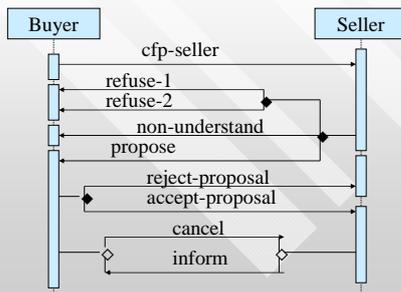
- accept-proposal : 提案受理
- cfp : 要求実行可能なエージェント募集
- propose : 提案(応募)
- reject-proposal : 提案不採用

### ■ <例外処理(2)>

- failure : 実行結果(失敗)の通知
- not-understood : 受信メッセージ理解不能の通知

## 交信プロトコルの仕様記述

### ■ UMLを拡張して利用



## FIPA ACLの考察(1/2)

### ■ 実装独立?

- \* メッセージの中でcontent記述言語を指定する方式で, 実装独立は可能?
- \* 例: SQLを理解するエージェントとPrologを理解するエージェントが会話できる?
- \* CORBA IDLのような機能が必要.  
(FIPAでは, content記述言語ライブラリを制定)

## FIPA ACLの考察

### ■ 完全性(広範囲応用性)?

- \* 実際のCA設定では, 電子商取引向きに見える.
- \* 種々のドメイン向きの機能を取り込むと他の設計目標である単純性(特殊機能排除)/簡潔性(曖昧性・冗長性排除)とコンフリクトを生じる. 22種類のCA設定は適切か?
- \* ちょっと複雑な「e-commerce」デザインパターン?  
(Buyerクラスとsellerクラス, さらにbrokerクラスも)

## OMG Agent Working Groupの概要(1/2)

### ■ ミッション

- \* エージェント技術支援のため, OMAの拡張推奨.

### ■ 目標

- \* エージェント技術に共通の意味, メタモデル, 抽象構文を採用. 再利用, 相互運用, 携帯可能なアプリケーションコンポーネントとして表現する.
- \* エージェント技術を用いたアプリケーション開発の方法を開発者に理解させ, 市場を育成する.

## OMG Agent Working Groupの概要(2/2)

### ■組織

\* the Electronic Commerce Task Force (ECTF) のWG

### ■紹介文献

\* Agent Technology: Green Paper  
Agent Working Group  
OMG Document ec/99-12-02  
Version 0.9  
24 December 1999

## <OMG AWG> エージェントの定義

### ■必須特性

- \* Autonomous:  
外部コントロールなしで経験に基づく行動
- \* Interactive:  
環境および他のエージェントとの通信
- \* Adaptive:  
他のエージェントや環境に対応し、  
経験に基づいて行動変化

(注)産業界の標準的な定義はまだない。

## <OMG AWG> エージェントの形態

### ●ソフトウェアエージェント:

- ・上記3種類の必須特性を備えた一般的なエージェント
- ・エージェントはソフトウェアデザインパターン
- ・オブジェクト指向技術での実装には、3種類の特性支援要

◎自律エージェント: 非決定的なふるまいなど。

◎対話エージェント: 異質のエージェントとの対話など。

◎適応エージェント: 学習, 進化機能

\* 移動エージェント: 通信問題(トラフィック, 帯域幅)の解決

\* 協調エージェント: マルチエージェントシステム

\* 知的エージェント: 知識に基づく状態(信念, 願望, 意図他)

\* ラッパーエージェント: 既存のソフトウェアへの接続

## <OMG AWG> エージェント技術の現状

### ■位置付け

- ・エージェント技術は, アプリケーションの付加機能
- ・将来は, OSの支援するプラットフォーム機能
- ・エージェント技術は, 複合技術

### ■現在の適用分野

- ・ネットワーク管理 ・意思決定 ・興味マッチング
- ・ユーザ操作補助 ・組織構造模倣(例: EC分野)

### ■エージェントシステムの製品

- ・20種類のリストアップ

## <OMG AWG> エージェント技術の主要課題

### ■エージェント通信

- ・ACL ・メッセージ通信機構 ・オントロジー
- ・対話プロトコル

### ■内部構造

- ・ゴール表現 ・適応性 ・手続き的 vs 宣言的

### ■ライフサイクル管理

- ・永続性 ・行動履歴 ・動特性

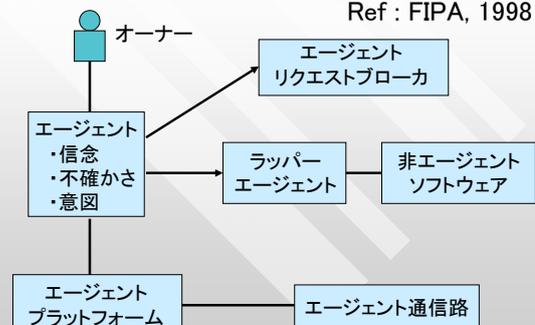
### ■移動性

### ■<代理人の行為=本人の行為>の保証

### ■セキュリティ

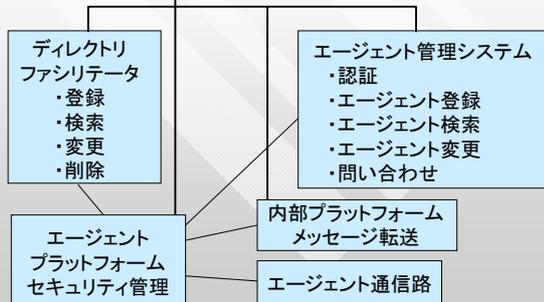
## <OMG AWG> エージェントのアーキテクチャ

Ref : FIPA, 1998



## <OMG AWG> エージェントプラットフォーム

エージェントプラットフォーム Ref : FIPA, 1998



## <OMG AWG> オブジェクト技術との関連(1/2)

### ■ 異質な面

- \* 異なるメッセージ毎にメソッドを対応させた場合:
  - ・メッセージの表現に限界
  - ・メソッド呼出しは、自律性と異なる面
  - ・メソッド呼出しは、長時間の会話・連携に不向き  
(注)代替案: "AcceptCommunicativeString" メソッドのみ.
- \* 能動的実体として、メッセージ駆動のほかに、イベント駆動も必要.
- \* オントロジー指定のメッセージ解釈による表現力強化.

## <OMG AWG> オブジェクト技術との関連(2/2)

### ■ エージェントの構成にオブジェクトの利用

- \* エージェントコンポーネント: オブジェクト候補
  - ・エージェント名
  - ・エージェント通信処理部
  - ・ACLコンポーネント(コード化; オントロジー; 語彙)
  - ・会話ポリシー(プロトコル?)
- \* マルチエージェントシステムのソフトウェアコンポーネント層 = エージェントインフラストラクチャ: オブジェクト候補
  - ・通信ファクトリ
  - ・転送ポリシー
  - ・ディレクトリ要素
  - ・エージェントファクトリ
- \* エージェントフレームワークにJava言語の利用.

## エージェント応用

### ■ 文献

CACM, Vol. 42, No. 3 March, 1999  
 特集: Multiagents Systems on the Net  
 特集: Agents in E-commerce

### ■ 初期のエージェントの応用分野

- ・情報フィルタリング
- ・類似の趣味の人々のマッチ
- ・反復作業の自動化.

### ■ ネット上のマルチエージェントシステム

- ・電子商取引
- ・バーチャル企業
- ・科学計算
- ・知的生産
- ・HA
- ・配電管理
- ・コンポーネントベースのソフトウェア構成

## 電子商取引(1/2)

- ・EC支援: 分散オブジェクト+Java+XML+Agent
- ・形態: 買い手と売り手がデジタル市場にエージェントを派遣し、彼らの最善の利益を自律的に代行させる.
- ・処理: 購入エージェントは、ベンダーと製品の情報を収集し、評価し、対象を決定し、取引条件を交渉し、注文し、自動決済する。(時間のかかる処理を自動化)
- ・仲介者(例:T@T): 商品の推奨、売り手の紹介、交渉支援。  
<購入条件: 両者が協調して設定>
- ・仲介者(例:BargainFinder): 9以上の売り手の価格調査。  
<CDの売り手の1/3は価格提供拒否>

## 電子商取引(2/2)

- ・仲介者(例:Jango): 買い手のブラウザから購入条件提示。  
<本物の買い手からの要求で、拒否反応減少>
- ・比較購入エージェント:  
XMLとモバイルエージェント技術で発展.
- ・競売(例:AuctionBot): ユーザにエージェント作成API提供.
- ・競売(例:Kasbah): 両者の条件提示により、交渉を自動化。  
\* 売り手の例: 期限、希望価格と下限、時間・価格特性  
\* 交渉方法: 買い手の指値に売り手は可否回答のみ  
\* CtoCの例: MITの学生が本と音楽媒体の売買に試用.
- ・コンピュータが理解可能なXML文書の活用:  
例: カタログ、フライトスケジュール、株情報など.

## 提言：絶えざる変化への対応

- 開発者の視点：  
BtoB, BtoC 対応の新ソフトウェアモデルとしての  
ダイナミック ソフトウェア アーキテクチャの実現
- ユーザの視点：  
変化に応じた動的サービス提供のために  
業務の専門家主導のダイナミズムの実現
- コンポーネントとしての適応エージェントの可能性

## 文献

- FIPA <http://www.fipa.org/>
  - ・FIPA Draft 2-1999 Agent Communication Language
- OMG Agent Working Group  
<http://www.objs.com/isig/agents.html>
  - ・Agent Technology: Green Paper, OMG Document ec/99-12-02, Version 0.9, Dec. 1999.
- 学会誌
  - ・CACM, Vol. 42, No. 3 March, 1999  
特集: Multiagents Systems on the Net  
特集: Agents in E-commerce
    - \* Pattie Maes, et al.: Agents That Buy and Sell.
  - ・Munindar P. Singh: Agent Communication Languages: Rethinking the Principles, IEEE Computer, 31,12, 1998.
  - ・西田豊明: ネットワークエージェント, 情報処理, 39, 3, 1998.

## 付録

## アプリケーション構築技法

- 自律的個体の再帰的構成

$\langle A \rangle = \langle a \rangle \mid \langle A \rangle$  の集合

$\langle A \rangle$ : アプリケーション (エージェント)

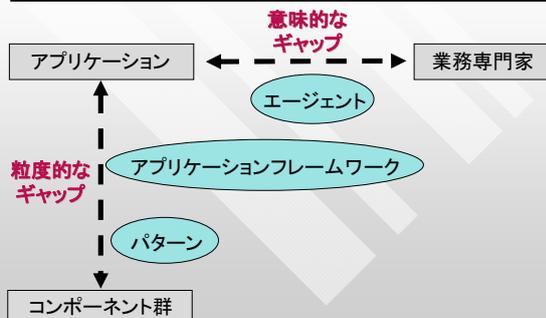
$\langle a \rangle$ : 最小の自律的個体 (エージェント)

(例)

- ・オブジェクト → 複合オブジェクト → ...
- ・クラス → パターン → フレームワーク → ...
- ・コンポーネント → ...
- ・エージェント → マルチエージェント → ...

- コンポーネントの組み合わせ技術(モデリング)

## コンポーネントの組み合わせ技術



## メモ