Too radical — or not radical enough?

- Progress report on AUML -

James Odell
Ann Arbor, Michigan  USA
www.jamesodell.com
Traditional Business Process Handling

Process request

Business Process Handler

Process

Process
Service-Oriented Handling - 2

Adaptive Process Manager (Agent Server)

Current State

Plan Library

Condition

Process

Events

Process request
Service-Oriented Handling - 3

Current State

Plan Library

Adaptive Process Manager (Agent Server)

Condition

Process

Condition

Process

Condition

Process
How “BDI” Plan-Execution Agents Work in Agentis

Adaptive Process Manager (Agent Server)

Goal Hierarchy

Current State

Plan Library

Agent

Goal

Sub-goal

Condition

Process

Condition

Process
How “BDI” Plan-Execution Agents Work in Agentis

- **Goal Hierarchy**
  - Goal
  - Sub-goal

- **Current State**

- **Plan Library**

Adaptive Process Manager (Agent Server)
Methodology Overview

Core Development Processes

- Requirements
- Area Analysis
- Area Design
- Area Implementation
- Assembly and Deployment

Development Spanning Processes

- Project Management
- Testing
- Maintenance
- Technology Architecture
- Strategic IT Planning
- Change Management
- Configuration Management
- Reuse Management
- Method Engineering

Agentis
Powering the Adaptive Enterprise
Requirements - Perform High-Level Analysis

1.0 Requirements

1.1 Prepare for Requirements

1.2 Perform High-level Analysis

1.3 Specify Initial Application Architecture

1.4 Determine Analysis Areas and Interactions

1.5 Validate Requirements and Plan Application

1.6 Review High-level Scenario Models

1.2 Perform High-level Analysis

Specify Actors, Goals, and Services

Refine Goals and Use Cases

Define Use Case Scenarios

Specify High-level Process

Specify High-level Class Model

Specify High-level Scenario Models
Specify Actors, Goals, Services: Describe the context
Identify actor roles (stakeholders)

Specify Actors, Goals, Services:

- Shipment Inquirer
- DHL
- Shipment Tracking System
- Check Point System
- Tracking Inquiry
Specify Actors, Goals, Services: Identify actor roles (stakeholders)

Or, you can use a shorthand notation.

DHL

Shipment Inquirer

Shipment Tracking System

Tracking Inquiry {general : Check Point System}
Specify Actors, Goals, Services:  Identify goals and subgoals

**GOALS:**
To identify “lost” shipment
  To obtain customer info
  To determine shipment status
  To trace shipment
  To sort out the problem
To identify “switched” shipment
  To obtain customer info
  To determine shipment status
  To trace shipment
  To sort out the problem
To compute ETA
  To obtain customer info
  To determine shipment status
  To calculate ETA
To locate a shipment
  To determine if a valid shipment
  To determine current status

**GOALS:**
To resolve shipment problems quickly (NF?)
  To appropriately allocate resolving customer service agents (CSA)

**GOAL:**
(None for this application.)

(NF = Non-functional goal)
GOALS:
To identify “lost” shipment
  To obtain customer info
  To determine shipment status
  To trace shipment
  To sort out the problem
To identify “switched” shipment
  To obtain customer info
  To determine shipment status
  To trace shipment
  To sort out the problem
To compute ETA
  To obtain customer info
  To determine shipment status
  To calculate ETA
To locate a shipment
  To determine if a valid shipment
  To determine current status

GOALS:
To resolve shipment problems quickly (NF?)
  To appropriately allocate resolving customer service agents (CSA)

Find Lost Shipment
  «include»
Determine ETA
  «include»
Determine Shipment Status
  «include»
Resolve Shipment Problem Quickly
  «include»
Allocate Appropriate CSA
  «include»

invocable service?
(If not, remove.)
Specify Actors, Goals, Services: Identify use cases (results)

- Shipment Inquirer
- DHL

Diagram:
- Find Lost Shipment
- Determine ETA
- Determine Shipment Status
- Allocate Appropriate CSA

Tracking Inquiry {general: Check Point System}
Specify Actors, Goals, Services:

Specify actor interfaces

- Find Lost Shipment
- Allocate Appropriate CSA
- Determine ETA
- Determine Shipment Status

Request to find lost shipment
Request for ETA
Request for tracking info
Request for shipping info

Lost shipment status
ETA
Tracking info
Specify actor interfaces (w/service)

Shipment Inquirer

Find Lost Shipment

Request to find lost shipment

Lost shipment status

Allocate Appropriate CSA

«include»

Determine ETA

Request for ETA

ETA

Determine Shipment Status

Request for tracking info

Tracking info

Shipment Search

Tracking Inquiry
{general: Check Point System}

DHL

Provider
General Architecture:
Service-Oriented Model

Web Services Architecture
W3C Working Group
Note 11 February 2004
Specify actor interfaces (w/service)

Specify Actors, Goals, Services:

- **Shipment Inquirer**
- **DHL**
- **Tracking Inquiry** {general: Check Point System}
- **Shipment Search**

- Find Lost Shipment
- Allocate Appropriate CSA
- Determine ETA
- Determine Shipment Status
- Allocate Appropriate CSA
- Request Shipment Status
- Request for ETA
- Request for tracking info
- Include

- Request to find lost shipment
- Lost shipment status
- ETA
- Tracking info
Specify Actors, Goals, Services: Identify providing actor

- **Shipment Inquirer**
  - Request to find lost shipment
  - Request for ETA
  - Request for tracking info

- **DHL**
  - Find Lost Shipment
  - Allocate Appropriate CSA
  - Determine ETA
  - Determine Shipment Status

- **Shipments Search**
  - Tracking Inquiry {general: Check Point System}

- **Allocation Mgr**
  - Shipments Inquiry Interface {general: CSA}

- **Request for tracking info**
  - Tracking info
Refine Goals and Use Cases:

**Identify sub-goals**

**GOALS:**
- To identify “lost” shipment
  - To obtain customer info
  - To determine shipment status
  - To trace shipment
  - To sort out the problem
- To identify “switched” shipment
  - To obtain customer info
  - To determine shipment status
  - To trace shipment
  - To sort out the problem

**Shipment Inquirer**

**Find Lost Shipment**

**Shipman Inquiry Interface**

{general: CSA}
Refine Goals and Use Cases: Identify sub-use cases

- Shipment Inquirer

- Request to find lost shipment

- Lost shipment status

- Find Lost Shipment

- Provider

- Retrieve Customer Info
- Determine Shipment Status
- Raise Shipment Trace Request
- Allocate Appropriate CSA

Shipment Inquiry Interface {general: CSA}
Refine Goals and Use Cases:

**Identify sub-use cases (improved)**

- Shipment Inquirer
- Request to find lost shipment
- Lost shipment status
- Find Lost Shipment
- Include
- Shipment Inquiry Interface
  - [general: CSA]
- Provider
- Retrieve Customer Info
- Include
- Determine Shipment Status
- Include
- Raise Shipment Trace Request
- Include
- Allocate Appropriate CSA

**Agentis - Powering the Adaptive Enterprise**
Refine Goals and Use Cases:

Identify provider actor

- Find Lost Shipment
  - Request to find lost shipment
  - Lost shipment status
  - provider

- Retrieve Customer Info
  - provider

- Determine Shipment Status
  - «include»

- Raise Shipment Trace Request
  - «include»

- Allocate Appropriate CSA
  - provider

- Shipment Inquiry Interface
  - general: CSA

- Allocation Mgr
  - general: CSA

Shipment Inquirer
Define Use Case Scenarios

For each use case, identify, and specify the various kinds of scenarios in the following steps:

1) Identify use case subtypes (specializing the use case into possible use case subtypes)

2) Specify basic scenario for each subtype (main successful scenarios, i.e., where nothing goes wrong.)

3) Specify alternative scenarios (business process faults; scenarios where the external actor(s) can exercise a choice or ability to make a corrective action)

4) Specify exception scenarios (technical faults; scenarios where the external actor(s) cannot make a corrective action)
Specify High-Level Process

Find Lost Shipment

{Plan rule = BasicScenario}

Shipment Inquiry Interface

- Retrieve Customer Info
- Determine Shipment Status
- Raise Shipment Trace Request

Allocation Mgr

Allocate Appropriate CSA
Specify High-Level Process (w/Actor role)

Or, as an alternative to partition “swim lanes.”

Find Lost Shipment

(Shipmen Interface) Retrieve Customer Info

{Plan rule = BasicScenario}

(Shipmen Interface) Determine Shipment Status

(Shipmen Interface) Raise Shipment Trace Request

(Allocation Mgr) Allocate Appropriate CSA
Specify High-Level Process (Sequence Diagram)

sd Find Lost Shipment

{Plan rule = BasicScenario}

Shipper

Locate Shipment

Retrieve Customer Info

Determine Shipment Status

Raise Shipment Trace Request

Allocate Appropriate CSA

Lost Shipment Status
Specify High-Level Class Model

For each scenario, this step identifies and defines business concepts. Business concepts are those entities that are necessary for shared understanding and communication within the enterprise.

For each scenario, the applicable concept classes are modeled. In the example below, the concepts for the Find Lost Shipment use case are represented as a class diagram.
Review High-Level Scenario Models

- **Review models** - developed in the two previous steps for completeness and consistency.

- ** Consolidate models** - bringing together each different type of model separately to reveal common patterns and structure without losing individual variation.
**Requirements - Perform High-Level Analysis**

1.0 Requirements

1.1 Prepare for Requirements

1.2 Perform High-level Analysis

1.3 Specify Initial Application Architecture

1.4 Determine Analysis Areas and Interactions

1.5 Validate Requirements and Plan Application

1.6 Review High-level Scenario Models

1.7 Specify High-level Process

1.8 Specify High-level Class Model

1.9 Refine Goals and Use Cases

1.10 Define Use Case Scenarios

**Requirements**

Area Analysis

Area Design

Area Implementation

Assembly and Deployment

1.2 Perform High-level Analysis

Specify Actors, Goals, and Services

Review High-level Scenario Models
Case Study — so far

- Many different notations can be used to represent high-level analysis notions.
- UML 2.0 is used here as a “first solution” because most of IT customers are familiar with it. It is not necessarily the only solution.
- UML 2.0 is also being extended to support agents.
- The approach is not solely agent focused, because applications are hybrid in nature. It supports many approaches, such as relational, OO, components, SOA, agents, etc.
- It is a
  - goal-directed,
  - service-oriented, and
  - role-based approach.
- It models:
  - business stakeholders,
  - business goals,
  - business processes, and
  - business concepts.

Is this: too radical—or not radical enough?
Dynamic Role Classification

UML 2.0 Interaction Diagram

Some next steps
Some next steps

Roles and Groups

UML 2.0 Composite Structure Diagram
Some next steps

Addressing all UML 2.0 diagrams—or not

Structure Diagrams
- Class diagrams
- Composite structure diagrams
- Component diagrams
- Packages
- Deployment diagrams

Behavior Diagrams
- Sequence diagrams
- Communication diagrams
- Activity diagrams
- State Machine
- Use Case diagrams
Some next steps

Other deficiencies in UML 2.0 diagrams

- goals
- agent
- group
- multicasting
- generative functions, such as cloning, birthing, reproduction
- parasitism and symbiosis
- emergent phenomena
- . . .
And what about: TROPOS

Goal dependency:
- depender
- dependum
- dependee

Contribution:

AND decomposition:

OR decomposition:

Diagram:
- Customer
  - Buy Media Items
  - Happy Customers
- Media Shop
  - Consult Catalogue
  - Increase Market Share
  - Media Items
  - Continuous Business
  - Continuous Supply
- Media Supplier
  - Quality Packages
- Media Producer

Diagram elements:
- Actor
- Hard goal
- Soft goal
- Plan
- Resource
And what about: PROMETHEUS
And what about: MaSE
And what about: MESSAGE
Too radical —
or not radical enough?

- Progress report on AUML -

James Odell
Ann Arbor, Michigan  USA
www.jamesodell.com