NORTEL NETWORKS

FIPA-OS Agent Tasks

Ian Bourke

Overview

- What is a task?
- Why use tasks?
- What type of task are there?
- How do I use tasks?

What is a task?

- A task is an agent ‘behaviour’
- It encapsulates the functionality needed to perform one distinct task
- It provides a convenient way of programming an agent and promotes object re-use

Why use tasks?

- **More logical agent code structure**
  - —Component based
  - —Easy to add new functionality
  - —Removes the need for complicated state management
  - —Easier to develop agents
- **Prevents code clashes in a complicated agent**
- **Allows agent developers to share code**
- **Follows a recognised design pattern**

What types of task are there?

- **Application tasks**
  - —'Normal' task – used in an agent for application specific tasks
- **Hidden tasks**
  - —Form part of the agent API – programmer is not aware of them
- **Library tasks**
  - —Code for common tasks that can be shared

What types of task are there?

- **Each task type can be further classified into:**
  - Singleton
    - Only one instance of the task per agent
  - Concurrent
    - Any number of instances of the task per agent
  - Listener
    - Singleton task that listens for ‘unknown’ incoming messages

How do I use tasks?

- **Decompose agent application functionality into distinct tasks**
  - E.g. participation in Contract-Net
- **Write a task object to handle each task**
- **Write a listener task**
- **Chain the tasks together using the ‘main’ agent class**

How do I use tasks?

- **Writing a task**
  - Extends fipaos.agent.task.Task
  - Has handle*X* methods for each performative that will be received, e.g. handleAcceptProposal
  - Super constructor takes a FIPAOSAgent and a Task Manager
- **Task API provides**
  - forward
  - getNewConversation
  - …

How do I use tasks?

- **Example task:**

```
class MyTask extends Task {
        public MyTask( FIPAOSAgent owner, TaskManager tm ) {
                super( owner, tm );
        }
        public void handleRequest( Conversation conv ) {
                // whatever
        }
}
```

How do I use tasks?

- **What about the notify() method?**
  - You don’t need one – just register conversation IDs with the Task Manager and messages will be delivered to the correct handle method automatically

```
registerTask( acl.getConversationID() );
```

How do I use tasks?

- **What about messages where I don’t know the conversation ID?**
  - These will be delivered to the listener task
  - This task should determine what the message is and pass it to an application task
  - Register with the Task Manager when the agent starts up

```
setListenerTask( new MyTask( this, _tm ) );
```

How do I use tasks?

- **Instantiate tasks as you would any other object**
  - — From the agent's 'main' class
  - — From other tasks
- **Inter-task method calls are fine**
  - — Directly using task defined methods
  - — Through the owning agent if necessary

NORTEL NETWORKS

How do I use tasks?

- **Spin off as many instances of a task as you need**
  - E.g. an agent wants to participate in 100 Contract-Nets at the same time – spin off 100 Contract-Net tasks
- **The state information for the task should be entirely encapsulated**
  - No clash of state variables

More information

- **Basic introduction to tasks**
  - — See FIPA-OS tutorial Step 2 – Ping Agent
- **Guidelines for writing ‘well-formed’ task based agents**
  - — See FIPA-OS tutorial Step 4 – Tag Agent