FIPA-OS Tutorial Step 5

eMarkets

Reference FIPA-OS Tutorial Step 5

Contributor(s):
Madhawa Shirantha Mutukumarana - Informatics Institute of Computer Studies (Sri-Lanka)

Publication History

29 June 2001
Initial version produced

18 January 2002
Added ‘Background’ section, amended minor formatting errors

Table of Contents

About this document

What is this document?

This document accompanies the eMarkets files included with the FIPA-OS Tutorial Distribution v2.1.0 and above. It explains how to use the eMarkets agents as part of the FIPA-OS tutorial.

Intended Audience

Developers using FIPA-OS to develop agent applications

Reading Guide

It is strongly recommended that the reader should look at the FIPA-OS web site at http://fipa-os.sourceforge.net/ to understand the rationale behind this platform and for information on future updates. The installation, configuration, start-up and test instructions are written assuming that the developer will be using a Windows95/NT/2000 or Unix based system to run FIPA-OS.

Developers using FIPA-OS are encouraged to provide extensions, bug fixes and feedback to help improve the planned future releases. All such input should be contributed to the Open Source project via the SourceForge site at http://sourceforge.net/projects/fipa-os/. You are required to register as a developer to access some of the services at the SourceForge site. General issues and thoughts can be discussed via the FIPA-OS mailing list on fipa-os-developers@lists.sourceforge.net although you must register at http://lists.sourceforge.net/mailman/listinfo/fipa-os-developers on this list before you can send and receive messages. An archive of the messages sent to this list can also be viewed from http://www.geocrawler.com/redir-sf.php3?list=fipa-os-developers. Should you experience difficulties using this list, then please contact the FIPA-OS co-ordinators at fipaos@emorphia.com. Please consult the *FIPA_OS_Public_Licence.txt* file for further details on the requirements for using, extending and evolving FIPA-OS.

Conventions used

Within the text filenames appear in *italics*. In examples where users should enter data, the suggested data appears in **bold**. For examples of entering data at the command prompt, variables are encapsulated in < and > and optional data is encapsulated in **[** and **]**, e.g. [<comms-transport>] is an optional parameter which can be specified at the command prompt.

Terminology

CS	Component Seller
MB	Manufacturer Buyer
MS	Manufacturer Seller
RB	Retail Buyer

Chapter 1
Tutorial Step 5 - The FIPA-OS eMarkets Tutorial

This is a new tutorial, introduced with the 2.1.0 release of the FIPA-OS agent development kit.

The eMarkets tutorial is an agent-based simulation of an electronic marketplace where agents represent buyers or sellers that wish to make purchases or sales.

Background

Electronic Markets (eMarkets) are one of the benefits that a global network provides. Currently, many electronic commerce systems exist, allowing end-users to buy and sell products through the Internet. However, the shortfall of these commerce systems is that the decision maker (normally the end-user) has a constant requirement to involve themselves with the marketing applications provided by each vendor.

The Internet's growth as a medium for conducting commerce presents a new set of problems, involving how to conduct electronic commerce and its automation. Incorporating autonomous agents into an electronic commerce environment provides a method to introduce dynamic pricing and negotiation schemes, which allows the vendor or service provider the opportunity to match consumer profiles to products and services. Creating specialised, niche markets that would otherwise be unavailable.

Furthermore, eMarkets are able to make items available to the entire global community 24 hours a day, eliminating the temporal and geographic limitations placed on traditional commerce systems. Additionally, electronic communication mechanisms make it is possible to improve the efficiency of the support processes of a commerce system such as: information retrieval, tendering for the supply of products or services and maintaining appropriate correspondence.

Therefore, this tutorial examines how to implement an eMarket using FIPA-OS. Beginning with a brief description of the scenario and main stakeholders in the marketplace. Following in from this the Tutorial examines the specifics of FIPA-OS market model and details how to run the eMarket system. It concludes by outlining the supplied code and strategies of each stakeholder.

Scenario

The eMarkets scenario is a 3-tiered marketplace (see figure 1) that contains the following agent types:

The **Component Seller Agent** constructs finished components from raw materials and sells these components to a **Manufacturer Buyer Agent**.

The Manufacturer Buyer Agent purchases finished components from a Component Seller Agent.

The **Manufacturer Seller Agent** uses finished components to create finished products, which it sells to a **Retail Buyer Agent**.

The **Retail Buyer Agent** purchases the finished products from a **Manufacturer Seller Agent** in order to sell them to consumers.

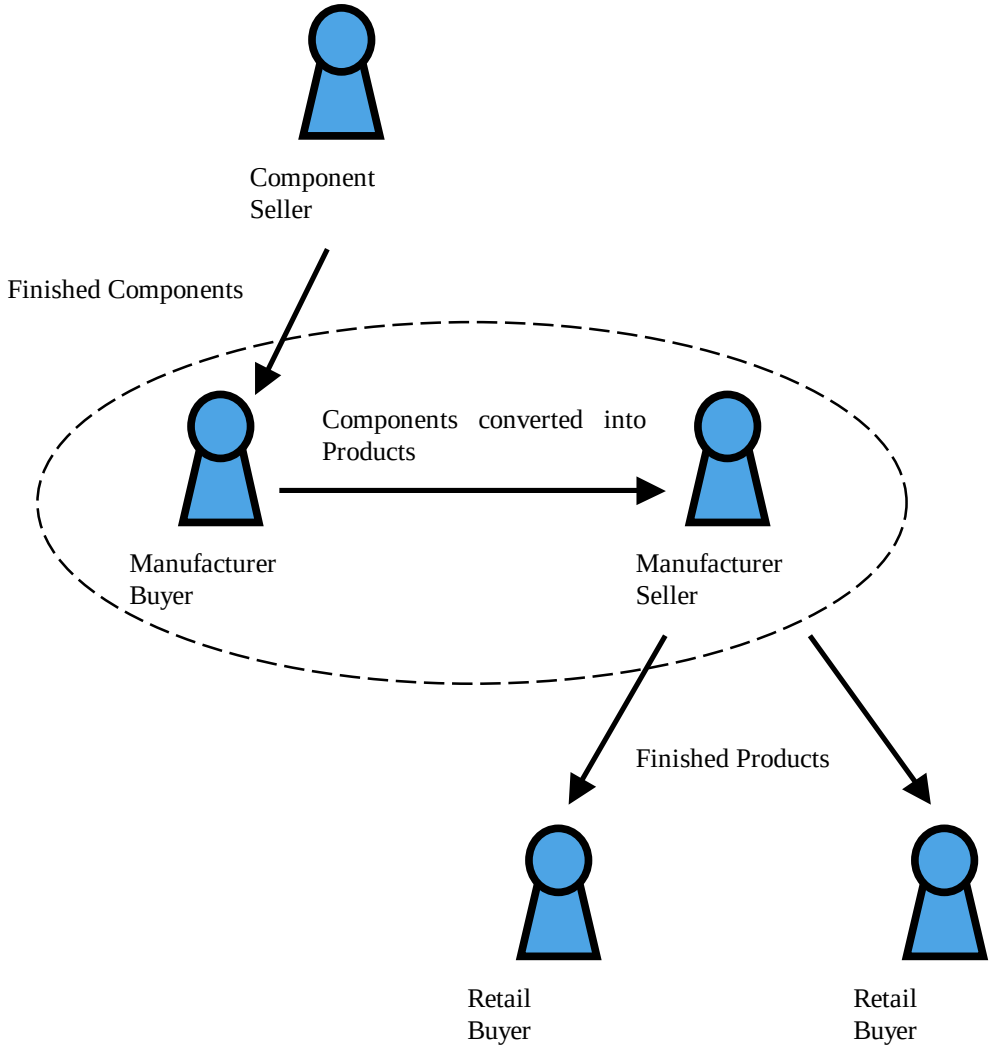

Figure 1. Three-Tiered marketplace

The Manufacturer Buyer and Manufacturer Seller are two agents that represent different parts of the same entity (company). The finished components that are purchased by the Manufacturer Buyer are processed by the manufacturing entity into finished products that can then be sold on by the Manufacturer Seller.

Part of the tutorial task is to implement an ACL messaging mechanism that allows the Manufacturer Buyer and Manufacturer Seller to communicate about demand from the Retail Buyer and purchases made from the Component Seller. This part of the tutorial is best left until you have successfully implemented new buying and selling mechanisms for each agent.

The FIPA-OS Marketplace Model

The eMarkets tutorial introduces a subscription-based framework for creating communities of agents. The Market Manager Agent is an agent that manages subscription requests from agents who wish to join the marketplace. See figure 2.

When an agent joins the marketplace, it is initially informed about all of the agents that are already registered in the marketplace.

When another agent joins or leaves the marketplace, the Market Manager Agent informs all subscribed agents of this event so that subscribers always have an up-to-date view of the marketplace.

When an agent leaves a marketplace, it forgets all knowledge of agents in the marketplace (It will of course be informed of the existing subscribers if it decides to rejoin a marketplace).

An agent can be subscribed to more than one marketplace and may have different roles in different marketplaces.

Each marketplace is managed by a different Market Manager Agent.

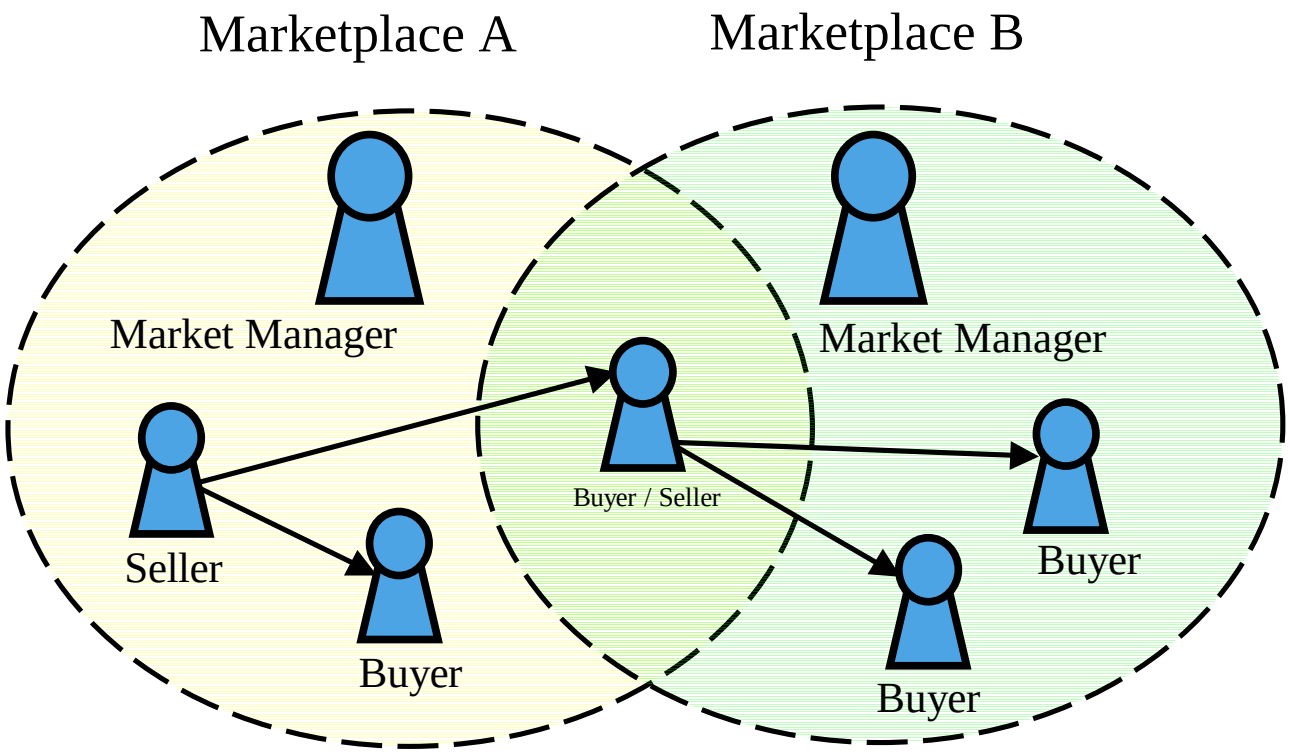

Figure 2. Marketplace model

How to run the eMarkets system:

The files in the tutorial are enough to demonstrate the running of the eMarkets system as they contain simple buying and selling algorithms. You should ensure that you could run the original system before making changes to the source code.

Please carry out the following steps to run the eMarkets system:

1. Start the FIPA-OS platform and Agent Loader by calling the startFIPAOS script as described in the FIPA-OS Distribution Notes [1].
2. Start the market manager agent using the Agent Loader
3. Choose the "Load Marketplace" option on the market manager GUI and select the *marketplace.map* file from your FIPA-OS *bat* directory. You should see the starting values load into the market manager GUI.

Now start the following agents from the AgentLoader GUI: cs1, mb1, ms1, rb1, rb2 (The *loader.profile* is updated to add these agents when you install the FIPA-OS tutorials.

For each of these agents, press the "Manage Subscriptions" button and subscribe to the market manager agent. You can close the subscription GUI once you have subscribed an agent. You should now see all of the eMarket agents subscribed to the market manager in the market manager subscribers GUI.

Press the "Start eMarket" button on the market manager control GUI to start the eMarket. You should see CFPs and proposals moving between the sellers and buyer agents. The default map file runs for five negotiations (numbered 0-4). You should then see the buyers and sellers send escrow messages to the market manager (which also acts as an escrow service). You should see "escrow-succeeded" displayed in each agent GUI.

Press the "Save Reports" button on the market manager GUI. This will save diagnostic files from the eMarkets system into your FIPA-OS bat directory.

Once you have successfully run the system, you can try altering the values in the *marketplace.map* file to ensure that your negotiation mechanisms run correctly under different starting conditions.

Supplied Code

For details of the functionality of the supplied code see the comments within the code itself.

Classes to Modify

You will need to change the code in the following eMarkets classes in order to modify the buying and selling behaviour of agents in the eMarkets system.

Agent Classes

The Component Seller
fipaos.tutorial.emarkets.agent.seller.component.ComponentSellerAgent.java

The Manufacturer Buyer
fipaos.tutorial.emarkets.agent.buyer.manufacturer.ManufacturerBuyerAgent.java

The Manufacturer Seller
fipaos.tutorial.emarkets.agent.seller.manufacturer.ManufacturerSellerAgent.java

The Retail Buyer
fipaos.tutorial.emarkets.agent.buyer.retail.RetailBuyerAgent.java

Seller Engines

You will need to create intelligent “offer engines” for your seller agents that examine a purchase bid from a buyer agent and return an intelligent counter-bid.

These offer engines must implement the interface defined in
fipaos.tutorial.emarkets.agent.seller.offer.OfferEngine

This interface defines a single method
public abstract NegotiationMessage makeOffer(NegotiationMessage neg_msg);

This method takes a proposal from a buyer and creates a counter-proposal in the form of a new cfp (call for proposals)

Buyer Engines

You will need to create intelligent “bid engines” for your buyer agents that create intelligent counter-bids in response to offers from a seller agent.

These bid engines must implement the interface defined in
fipaos.tutorial.emarkets.agent.buyer.bid.BidEngine

This interface defines a single method
public abstract NegotiationMessage makeBid(NegotiationMessage neg_msg);

This method takes a cfp from a seller and creates a new proposal.

Strategies

The objective of a buyer agent is to develop a buying strategy to purchase enough items to achieve its target and still remain within its budget of credits. A successful buyer agent will contain strategies that minimise a buyer's average price paid per item.

The objective of a seller agent is to develop a selling strategy to sell enough items to achieve its target number of credits. A successful seller agent will contain strategies that maximise a seller's average price received per item.

You must not allow your buyer agents to spend more credits than they own.
You must not allow your seller agents to sell more items than they own.

New / Updated FIPA-OS Files

loader.profile
This contains a list of agents to load into the FIPA-OS Agent Loader. This profile loads a component seller agent (cs1), a manufacturer buyer agent (mb1), a manufacturer seller agent (ms1), and two retail buyer agents (rb1 and rb2). This file will update the existing copy in your FIPA-OS profiles directory.

default.profile
This contains the protocol mappings needed by the eMarket communications protocols. This file will update the existing copy in your FIPA-OS profiles directory.

emarkets.constants
This file allows you to change the timeouts associated with the eMarket messages. If you find that the seller timeout is occurring before the buyer proposals are received then you should increase the value of CFPTimeout. This file will be placed in your FIPA-OS bat directory.

marketplace.map
This file contains a marketplace map that sets up an eMarket between the agents specified in the loader.profile file. You should load this file using the "Load Marketplace" button on the MarketManagerAgent GUI. This file will be placed in your FIPA-OS bat directory.

Exercises

1. Update the buyer and seller engines for each of the supplied agents as described in the Classes to Modify section of this document to enable competitive bidding between each of the agents, such that each agent aims to maximise its profit.
2. Update the Manufacturer Seller and Manufacturer Buyer to use FIPA ACL messages to enable them to co-operate as a team in their respective buying and selling strategies to ensure that they maximise their profit.
3. Introduce additional competitive agents i.e. Component Sellers, Retail Buyers or Manufacturer Seller and Manufacturer Buyer.
4. Update the **marketplace.map** file to change the goals for your agents to test their performance in a variety of markets. It may be necessary to update the buyer and seller engines to deal with the new market conditions.

Bibliography

[1] FIPA-OS v2.1.0 Distribution Notes, July 2001
http://fipa-os.sourceforge.net/docs/FIPA_OSv2_1_0.pdf