A Project Report for SENG 609.22

Agent Based Software Engineering
Course Instructor: Dr. Behrouz H. Far

Agent-Based Electronic MarketPlace System Design Document

Table of Contents

Table of Figures

1. System Specification

1.1 Business Case

Due to the increase in the popularity of the Internet, e-Commerce is expected to expand rapidly. With the introduction of system that can automate logistic planning among multiple entities such as suppliers, manufacturer, customers, and delivery agency, firms can cut costs and improve efficiency. By moving online, business becomes independent of time and location. Thus, improve sales and increase profit.

When customers want to purchase some goods in a traditional market place, they need to find out a list of potential suppliers and negotiate with each one of these potential suppliers. Then the customers determine which supplier offer the best deal or best bang for the buck. Negotiation can be a time consuming process. Negotiation can range from a couple of minutes to days or even months. It is possible that the best deal is not valid anymore due to time elapsed to find out the best deal.

The existing e-commerce systems present customers with a list of suppliers. The customers then choose the best deal based on factors such as price, quality of products, availability, reputation of the e-Commerce site and etc. Negotiation process is not allowed in the current e-Commerce implementation. The customers are presented with a fixed number of suppliers and their prices. There is a lack of flexibility of negotiating the best deal, which the suppliers are willing to sell.

An e-Marketplace system (eMS) aims to replace the process of human-to-human negotiation with the interaction of software agents. eMS has a great potential to help customers to compare and negotiate the best deals specified by pre-defined qualifications and helps consumers to make the most informative decision.

1.2 System Description

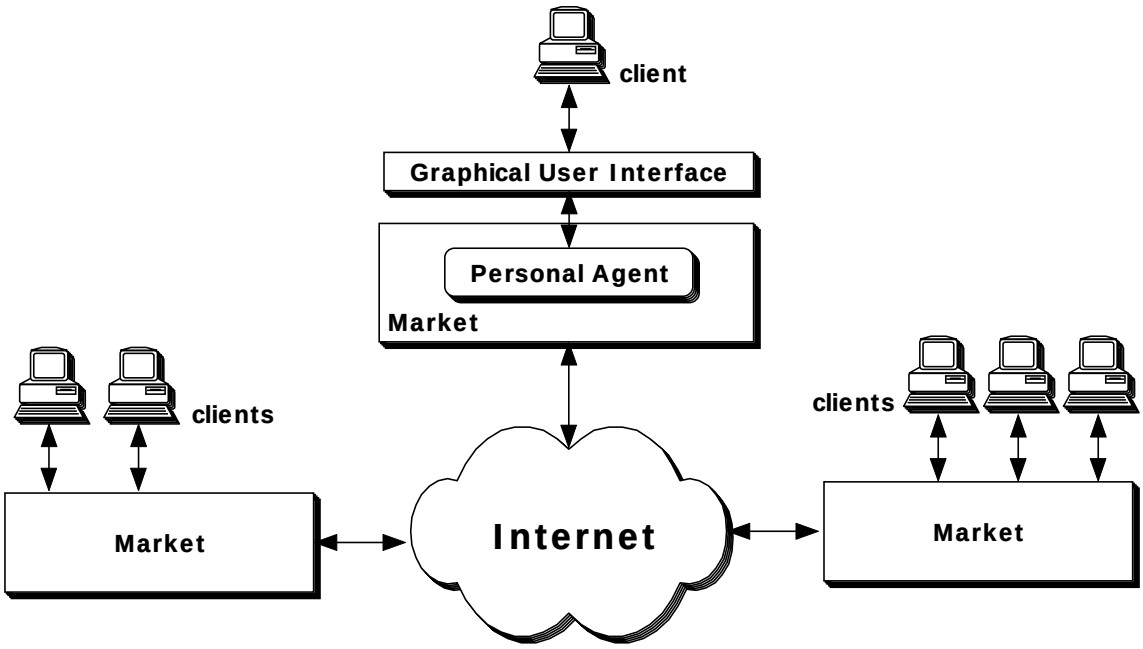

Figure 1 eMS System Description

Due to the large number of alternatives in the supplier-chains, the amount of time spent in evaluating and negotiating among a list of potential suppliers can vary from a couple of days to a couple of months. We believe that a multi-agent based system will result in a faster evaluation and negotiation process among suppliers and customers, prompter deliveries, and thus, in turn, reduce costs for both suppliers and customers.

The proposed e-Marketplace system (eMS) is a multi-agent system designed to support various types of electronic marketing: from the simple buying and selling goods to situations that require negotiation and contracting. In addition, the system also supports a wide range of agent types such as customer, supplier, deliverer and banker.

When selling something, the supplier first creates a software agent by specifying the type of goods/services he or she is selling, as well as specifying a strategy for negotiation such as the initial selling price, how low it can go, a plan to lower the price during the course of negotiation, and the availability of the goods/services. Similarly, customers create software agents by specifying a strategy for finding the desirable goods/services as well as a strategy for negotiation.

These agents then entered the market, find a list of potential customers or suppliers, and start the negotiation process by acting on behalf of their human counterparts autonomously to obtain a best deal.

1.3 Assumptions

- All the buys, sellers and Bankers are trustworthy, because the lack of mechanism to verify credibility of users.
- We will assume there is User Interface agent built in, and we will not consider User Interface in our design
- Customer agents always initiate request for quote and negotiation
- We will not consider any problems arose from competitions and uncertainties

1.4 Requirements

- The e-Commerce Market System (eMS) shall serve as a meeting place where suppliers, customers can meet and trade
- eMS shall allow a customer or supplier to specify the goods that he/she ants to buy or sell, desired price range, negotiation strategies and etc.
- eMS shall automatically query agent or service on behalf of the customer and/or supplier.
- eMS shall allow customer and suppliers to negotiate
- eMS shall monitor and record all the activities during the course of negotiation
- eMS can further be used to initiate, monitor and notify the customer and/ or supplier for the delivery and/or payment of the goods/services.
- eMS shall support different types of agents such as customers, suppliers, bankers and etc
- eMS shall matches what sellers wants to sell with what buys want to buy

1.5 Not implemented

- The eMS shall allow user to choose a user profile, such as user's preference and negotiation strategies from the database.
- The eMS shall allow user to cancel, modify, and suspend any task.
- New agents will be added to enhance eMS, such as Banker Agent, Delivery Agent, and Tax Agent.
- The eMS shall consider the issue of uncertainty and competition.
  The eMS shall support auction

2. System Design

2.1 System Architecture

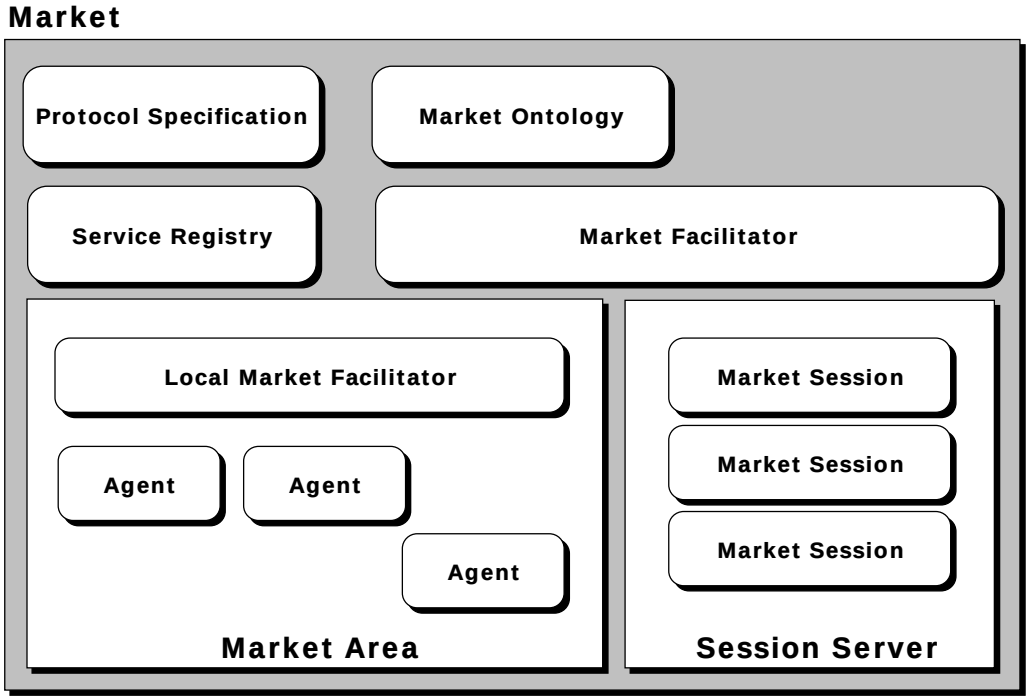

Figure 2 System Structure of Electronic MarketPlace

The fundamental elements of eMS are the market and the market area. The basic agents in eMS are: local market coordinator (LMF), market facilitator (MF), market session server (MSS) and participant agents (PA).

Market
Market represents a particular business area and serves as a meeting point where suppliers and customers meet and trade. It supports collaborative software agents by providing easy-to-use coordination services over the Internet. These services are:

Service registry
The Service registry stores services information of market participants, who have expressed interest in doing business in the market. An entry in this registry contains information such as the agent name (or agent ID), a catalog of services/goods that the agent provides, and performance statistic of the suppliers, which the customer could use in their decision processes.

Market Ontology
The market ontology specifies the resources that can be found in the market. The ontology contains not only a description of the resources, but also statistics such as the availability of a resource, the expected price of the resource, and the change in price over time.

Protocol Specification
The Protocol specification formalizes the types of negotiation supported within the market. For example, can a negotiation be extended after the negotiation deadline?

Market Area
Market Area is where participant agents reside and take advantages of the services provided by the Local Market Coordinator (LMF). It helps to form a logical grouping of the participant agents in a particular market. Statistics, such as the number of each agent type found in the market and, the majority agent types found in the market, can be collected and studied to help improve the market competitiveness and efficiency both in the real and hypothetical world.

Local Market Facilitator (LMF)
There is only one LMF per market. Each market has a unique LMF. LMF coordinates and represents agents in their area to the outside world, and helps them initiate agent-to-agent interactions. They manage the creation, registration, and deregistration of an agent in the market. In addition, it also serves as a proxy to the yellow pages and database of servers to the agents within the area.

Market Facilitator (MF)
MF is responsible for storing and making the information available forservices advertised by the agents in a local database. Other agents may retrieve the information by querying MF to determine the type of services offered by the other agents and the location of the other agents. All the agents are registered at a given MF form an agent domain. Thus, each MF manages a different agent domain.

Market Session Server (MSS)
MSS provides virtual environments supporting multi-agent communications, negotiations, and information sharing as well as the creation, the activation, and the deactivation of a market session. An agent initiates a session for a particular purpose (i.e. customer requesting proposal for bids); other agents can then join the session and participate as clients. The session enforces protocol rules, maintains internal state of the activity, maintains the continuity of partially completed transaction, and records all the transaction to allow history playback or undoing the

transaction, and time tracking. The session can be used for a wide range of tasks such as negotiation, payment of bills, delivery scheduling and etc.

Participant Agent (PA)
Each PA is an independent autonomous software entity, acting on behalf of their human counterparts who have different goals and different resources to satisfy. eMS current supports four agent types: supplier, customer, banker and delivery.

Supplier Agent
A supplier is an agent who has resources and responses to request for quotes and negotiation. Suppliers offer their goods/services for a specific price over a specific time periods.

Customer Agent
A customer is an agent who requires certain goods/services over a specific time periods in order to achieve its goal. The goal may have a value and it may vary over time.

Banker Agent
A banker is an agent who manages and records all the money transferred between the supplier agents and customer agents in the marketplace.

Deliverer Agent
A deliverer is an agent who managers the shipping and handling of goods in the marketplace.

Simply put, a market is an area or a meeting place where agents such as supplier and customer, reside. A market area has exactly one local market facilitator (LMF), which is an agent that coordinates and represents the agents in its area to the outside world. Agents can be identified being inside a market area if they have registered themselves with the LMF. Agents use the LMF services to access other agents in the market. Agents can advertised services and find out about other agents' services by going through the market facilitator (MF). Agent requires information sharing, participating in negotiations, scheduling for shipping of goods, requesting for quotes or proposal and etc., can join a market session. A market session is a virtual environment that provides services such as managing the passage of time, record all the transaction activities for history playback.

Agents
As mentioned previously, eMS is designed to support a various type of agent such as banker and deliver. However, we will focus the discussion on the supplier agent and customer agent in the following sections since they are the most prominent entity found in a marketplace, and during the

scope of negotiation, we could always generalize the participants as either a customer or the supplier.

2.2 Agent Architecture

2.2.1 Customer Agent

Architecture of the customer agent is shown below:

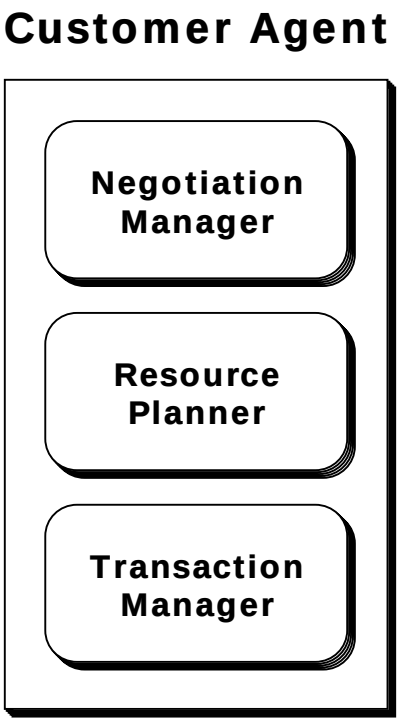

Figure 3 Customer Agent Architecture

A customer agent consists of three components. They are the resource planner, negotiation manager and the transaction manager.

Resource Planner
The customer specifies the task needed to be accomplished in the form of a Gant-Chart via the resource planner. Information such as a time window for each task, quality-level and the availability of the resources, and a strategy to lower the price during the course of negotiation for each task and, strategy for finding the desirable goods/services can be specified. The resource planner then transforms the task schedule to a data structure (i.e. adjacent link list) that will be used by the negotiation manager to carry out the tasks.

A high-level structure of the resource planner is shown below:

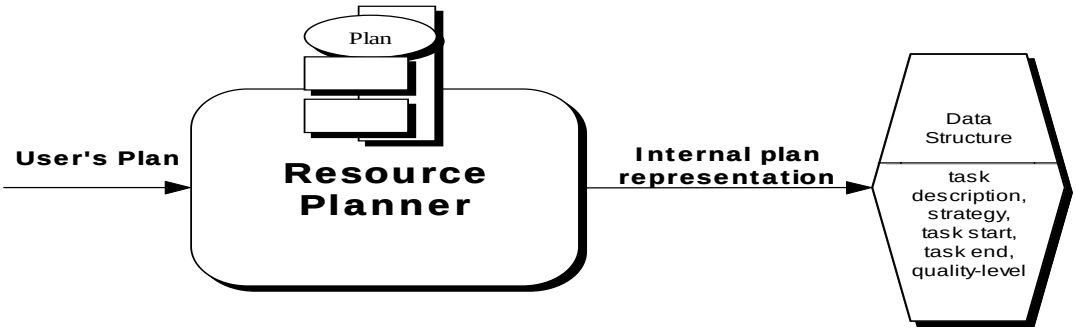

Figure 4 High-level structure of resource planner

Negotiation Manager

The negotiation manager makes use of the plan presented to it by the resource manager and devises a schedule for the negotiation process. The negotiation manager decides whether to issue a single request for negotiation for all the tasks or subdivide the tasks up among separate requests for negotiation. In addition, the negotiation manager monitors all the incoming bids from all the participated suppliers and carries out the negotiation strategy defined by the user to obtain the best deal. Once a consensus is made between the two parties, the transaction manager then kicks to handle the execution of the remaining plan.

A high-level structure of the negotiation manager is shown below:

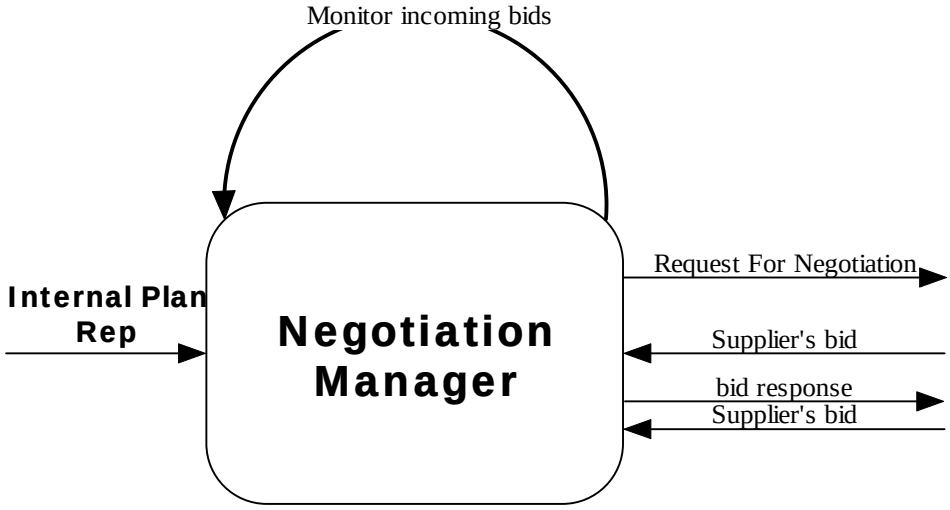

Figure 5 High-level structure of negotiation manager

Transaction Manager
Transaction manager monitors the money transfer as well as the ongoing process of the tasks such as delivery of the goods. Once the task is completed, the transaction manager notifies the customer and deactivates the market session.

2.2.2 Supplier Agent

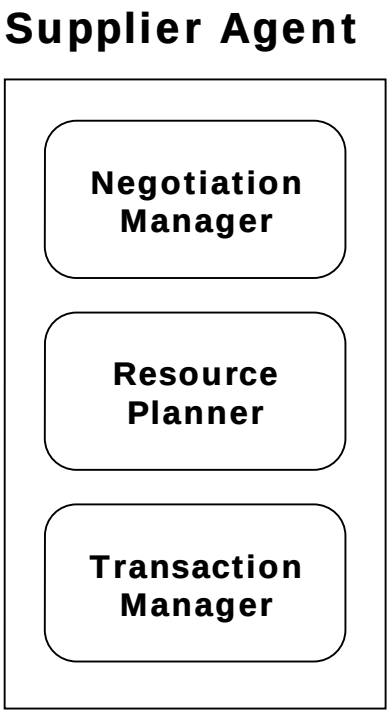

Figure 6 High-level structure of supplier agent

The supplier agent models the customer agent closely except for the functionality resource planner. When selling something, the supplier first specifies the type of goods/services he or she is selling as well as specifies a strategy for negotiation such as the initial selling price, how low it can go, a plan to lower the price during the course of negotiation, and the availability of the goods/services. The resource planner then transforms the negotiation strategy to a data structure that will be used by the negotiation manager to carry out the negotiation.

As an example, the following sequence diagram illustrates the interaction between a single supplier agent and a single customer agent through the use of a market section:

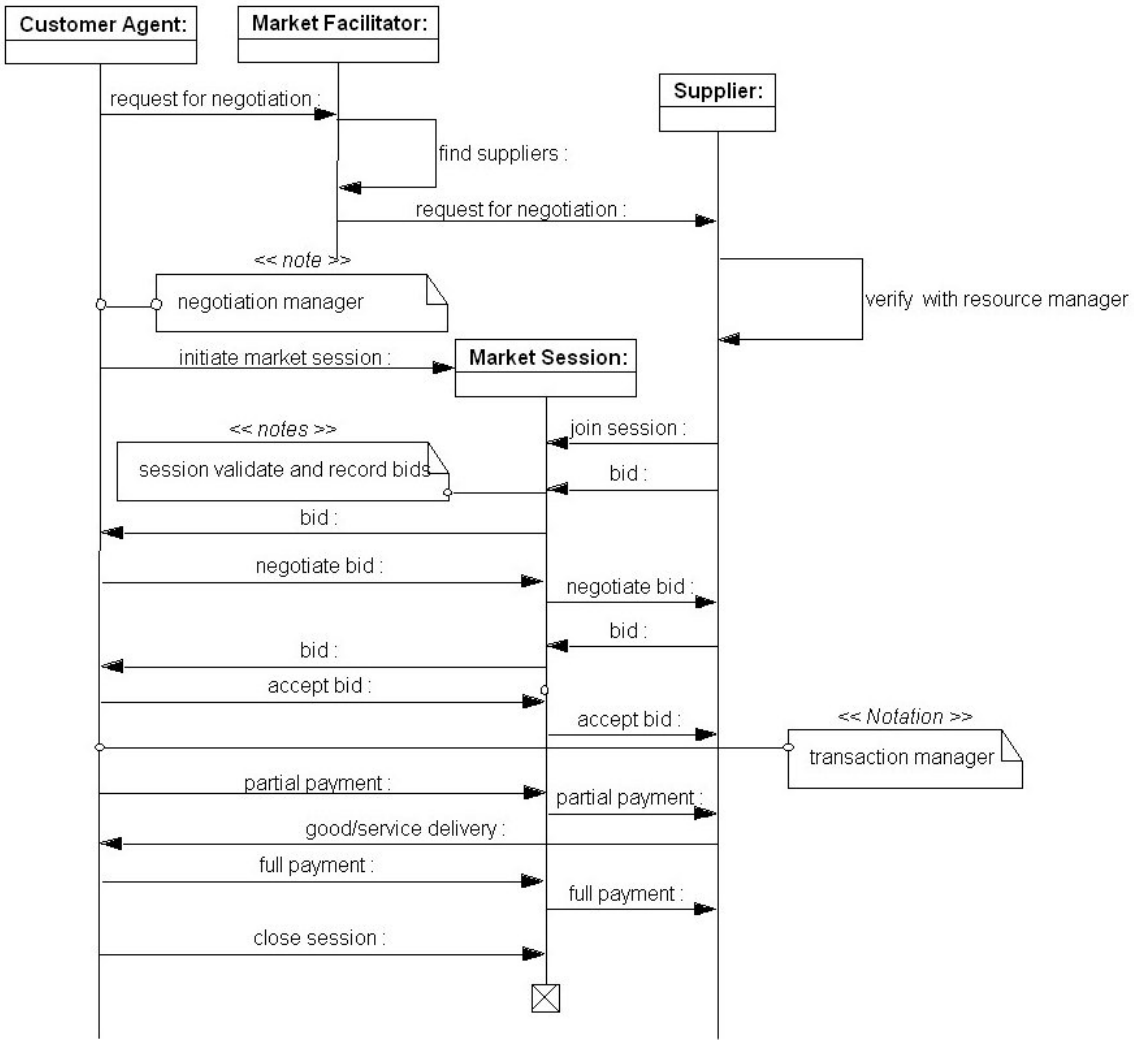

Figure 7 Example of supplier and customer negotiation

The customer initiates a request for negotiation to the market facilitator (MF) and initiates a market session through the market session server (MSS). MF then identifies a list of potential suppliers based on the customer selection criteria. MF then forwards the request for negotiation to the list of potential suppliers (in this example, just a single supplier). Interested party can then join the market session and participate in the negotiation process. Market session will keep track of the passage of time, record all the activities during the course of negotiation and etc. Once a mutual agreement is made between the two parties (i.e. the customer accepted the supplier's bid), the customer may decide to pay a deposit for the goods or services, the supplier then deliver the good/services to the customer. The customer then pay the remaining amount for the good/services delivered by the supplier. The deal is considered as close and the market session terminates.

2.3 Detailed Design

The following session contains use cases, class diagrams, sequence diagrams ,E-R diagrams and data dictionary of the eMS.

2.3.1 Use Cases

2.3.1.1 Use case definition: Customer and Customer Agent

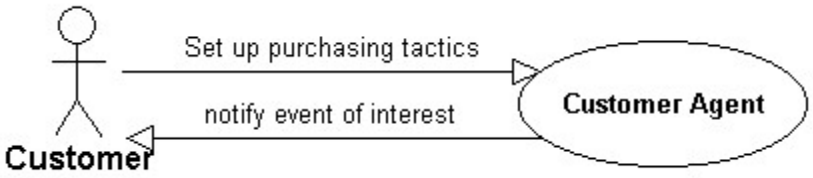

Figure 8 Use case for customer and customer agent

Brief description: Customer set up a list of purchasing tactics to Customer Agent through GUI.
Process steps:
1. Customer initiates a Customer Agent through GUI.
2. Customer inputs a list of desired goods information, for example what kind of the goods he is going to purchase, the maximum and minimum price he could pay, time window of the goods or service, and where the markets are etc
3. Customer sends out the above information to Customer Agent.
4. Customer Agent processes the information and enters the market.
Relationships: Initiating: Customer Collaborating: Customer Agent
Data requirements: Goods or service that the customer wants Price range Start date and end date Search criteria, such as market location, resource availability. Negotiation strategies, such as how to drop the price during the course of negotiation

Table 1 Brief description: Customer set up a list of purchasing tactics to Customer Agent through GUI.

Brief description: Customer Agent notifies the customer.
Process steps:
1. Customer Agent monitors the market and notifies the customer of what he desired, such as bought goods or service, money transfer, goods delivery and close deal.
Relationships: Initiating: Customer Agent Collaborating: Supplier Agent, Local Market Facilitator, Market Facilitator, Market Session, Bank Agent and Delivery Agent
Data requirements: N/A

Table 2 Customer Agent notifies the customer

2.3.1.2 Use case definition: Supplier and Supplier Agent

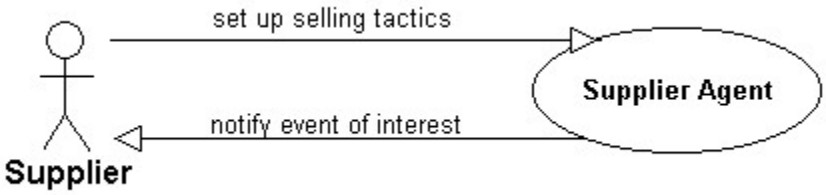

Figure 9 Use case for supplier and supplier agent

Brief description: Supplier sets up a list of selling tactics to Supplier Agent through GUI.
Process steps:
1. Supplier initiated a Supplier Agent through GUI.
2. Supplier inputs a list of selling tactics information, included what kind of goods he is going to sell, the range of price he required etc.
3. Supplier sends out the above information to Supplier Agent.
4. Supplier Agent processes the information and enters the market.
Relationships: Initiating: Supplier Collaborating: Supplier Agent
Data requirements: Goods or service that the supplier selling Price range Start date and end date Negotiation strategies, such as how to keep the price within the desired price range during the course of negotiation

Table 3 Supplier sets up a list of selling tactics to Supplier Agent through GUI.

Brief description: Supplier Agent noticed the supplier.
Process steps:
1. Supplier Agent monitors the market and notifies the supplier of what have happened, such as sold goods or service, money transfer, goods delivery and close deal.
Relationships: Initiating: Supplier Agent Collaborating: Customer Agent, Local Market Facilitator, Market Facilitator, Market Session, Bank Agent and Delivery Agent.
Data requirements: N/A

Table 4 Supplier Agent noticed the supplier.

2.3.1.3 Use case definition: Agent and Local Market Facilitator (LMF)

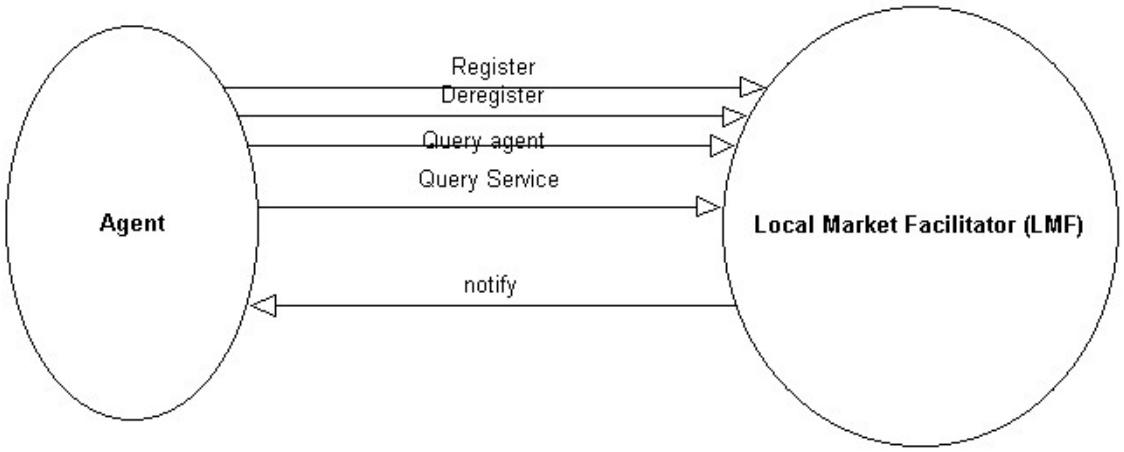

Figure 10 Use Case for Agent and Local Market Facilitator

Brief description: Agent registered with the LMF.
Process steps:
1. Agent requests for registration with LMF.
2. LMF verifies if the Agent exist.
3. LMF notifies the Agent if it has registered successfully
Relationships: Initiating: Agent Collaborating: Local Market Facilitator
Data requirements: Agent's ID, name and IP address.

Table 5 Agent registered with the LMF

Brief description: Agent deregisters with the LMF.
Process steps:
1. Agent requests deregistration from LMF.
2. LMF verifies if the agent exist.
3a. If yes, removes the agent from the LMF, and returns a successful deregistration message to the Agent
3b. If not, returns unsuccessful deregistration message to the Agent.
Relationships: Initiating: Agent Collaborating: Local Market Facilitator
Data requirements: N/A

Table 6 Agent deregisters with the LMF.

Brief description: Agent queries other agent with LMF.
Process steps:
1. Agent queries an agent from LMF.
2. LMF search for the requested agent.
3a. If found, returns the appropriate agent's address to the requested Agent
3b. If not, LMF forwards the query to MF.
4. MF search for the requested agent.
5a. If found, returns the appropriate agent's address to the requested Agent
5b. If not, notifies the Agent that the search failed.
Relationships: Initiating: Agent Collaborating: Local Market Facilitator and Market Facilitator
Data requirements: N/A

Table 7 Agent queries other agent with LMF

Brief description: Agent queries service with LMF.
Process steps:
1. Agent requests a service query to LMF.
2. LMF forwards the query to MF.
3. MF search for the service.
4a. If found, notifies the Agent
4b. If not, notifies the Agent that the search failed.
Relationships: Initiating: Agent Collaborating: Local Market Facilitator and Market Facilitator
Data requirements: N/A

Table 8 Agent queries service with LMF

2.3.1.4 Use case definition: Agent and Market Facilitator (MF)

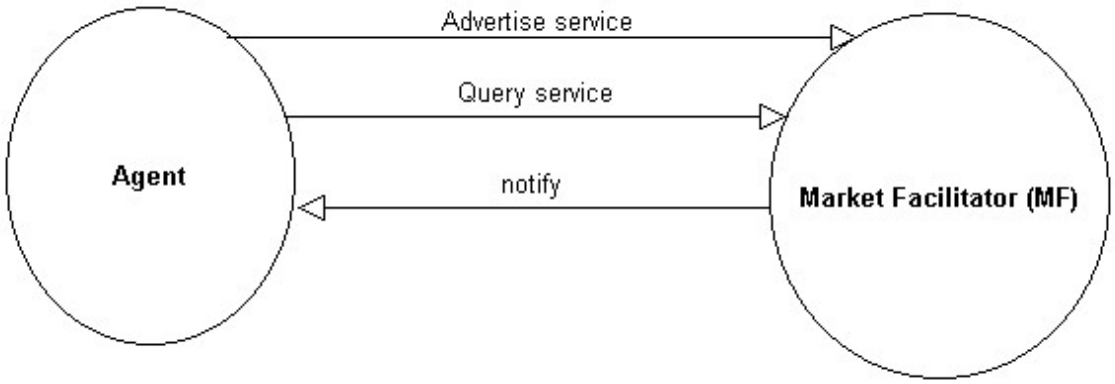

Table 9 Use case for Agent and Market Facilitator

Brief description: Agent advertises its service to MF
Process steps:
1. Agent advertises its service to MF, such as what kind of service or goods he is providing, the range of the price, and the quantity, etc.
2. MF verifies if the Agent has already advertised its service.

3. MF notifies the Agent if the advertisement is successful.
Relationships: Initiating: Agent Collaborating: Market Facilitator
Data requirements: Goods or service that the supplier selling Price range Start date and end date

Table 10 Agent advertises its service to MF

Brief description: Agent queries service with MF.
Process steps:
1. Agent queries a service from MF.
2. MF search for the service.
3a. If found, notifies the Agent
3b. If not, notifies the Agent that the search failed.
Relationships: Initiating: Agent Collaborating: Market Facilitator
Data requirements: N/A

Table 11 Agent queries service with MF.

2.3.1.5 Use case definition: LMF and Market Facilitator (MF)

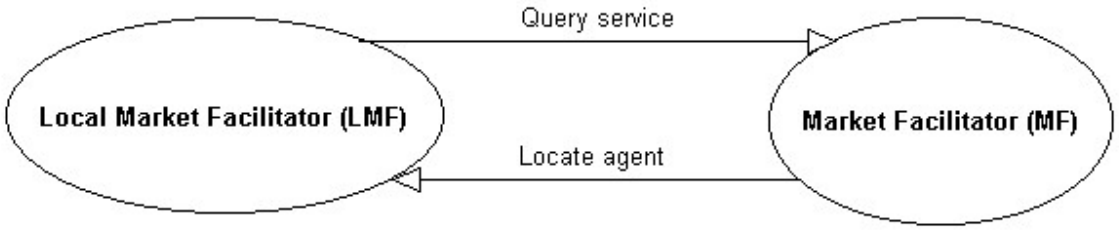

Figure 11 Use case LMF and MF

Brief description: LMF queries the service from MF
Process steps:
1. LMF queries a specific service from MF
2. Refers to the “Agent queries service with MF” use case process.
Relationships: Initiating: LMF Collaborating: Market Facilitator
Data requirements: N/A

Table 12 LMF queries the service from MF

Brief description: MF locates the Agent from LMF.
Process steps:
1. MF locates the agent from LMF.

2. LMF searches for the agent.
3a. If found, returns the appropriate agent's address to the MF.
3b. If not, notifies the MF that the search failed.
Relationships: Initiating: LMF Collaborating: Market Facilitator
Data requirements: N/A

Table 13 MF locates the Agent from LMF.

2.3.1.6 Use case definition: Supplier Agent and Customer Agent

Figure 12 Use case for customer agent and supplier agent

Brief description: Supplier Agents negotiates with Customer Agent.
Process steps:
1. Customer Agent initiates a request for negotiation and a Market Session (MS).
2. Supplier Agents joins the MS.
3. Starts the negotiation.
4. Customer award the bid to the Supplier Agent.
Relationships: Initiating: Customer Agent Collaborating: Supplier Agent, Local Market Facilitator, Market Facilitator, Market Session
Data requirements: N/A

Table 14 Supplier Agents negotiates with Customer Agent.

Brief description: Performs the transaction after negotiation.
Process steps:
1. Customer Agent engages Bank Agent to pay a deposit to Supplier.
2. Supplier Agent informs Delivery Agent to deliver goods to Customer.
3. Customer Agent pays the remaining amount to the Supplier.
4. The deal is closed and the Market Session is terminated.
Relationships: Initiating: Customer Agent Collaborating: Supplier Agent, Local Market Facilitator, Market Facilitator, Market Session, Bank agent, Delivery agent.
Data requirements: N/A

Table 15 Performs the transaction after negotiation.

2.3.2 Class Diagram

This section provides a series of UML class diagrams for the key elements of eMS.

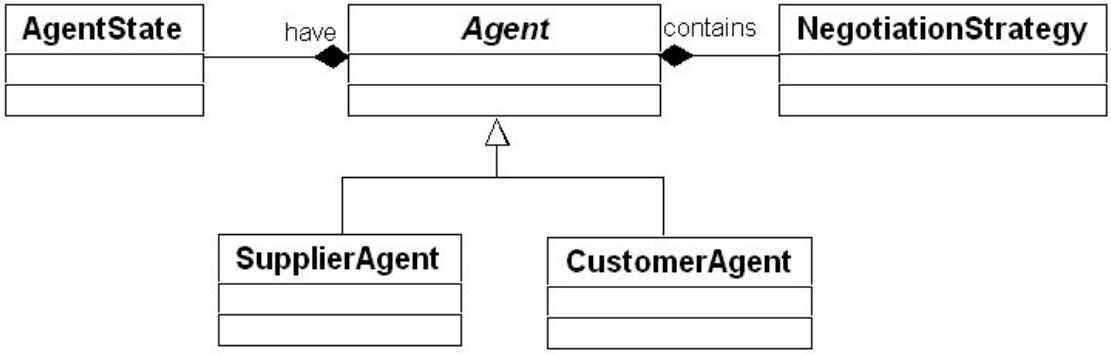

Figure 13 Agent class diagram

SupplierAgent, CustomerAgent and other agents are a specialized form of Agent. Agent is the abstract common denominator of all agents in the eMS. They implement the basic communication functionalities needed in agent-to-agent communication and interaction. An agent has states (i.e. AgentState) such as waiting for initialize state, initialized state and fail state. Depending on the agent's state, the agent might change its behavior at run-time. AgentState can be implemented following the State pattern introduced by the Gang of Four (GoF). In addition, agent also contains strategy for negotiation (NegotiationStrategy). However, the negotiation strategy may differ from agent to agent. Thus, the NegotiationStrategy serves as a common interface to allow new family of negotiation strategy to be defined. NegotiationStrategy can be implemented following the Strategy pattern introduced by the GoF.

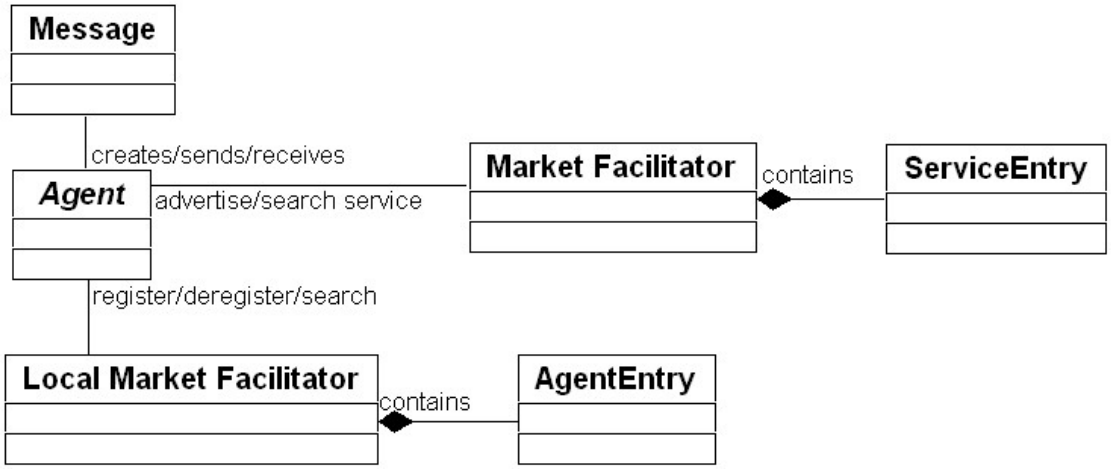

Figure 14 Agent relationship

Agent can create, send and receive message. Message contents are in XML format. Agent also register, deregister with the local market facilitator, query the local market facilitator to find out about other agents. Agent advertises their services to the Market Facilitator and can query the market facilitator to find out what services are available. Local Market Facilitator stores the registered agent's information by means of AgentEntry. AgentEntry is a key-value pair, where the key is the agent name (unique for each agent) and the value is the location of the agent. In a similar fashion, the Market Facilitator also stores the advertised agent in the ServiceEntry. Unlike the AgentEntry, the ServiceEntry is a key-n-tuple grouping. A service entry uses the agent-name as the key, and contains value such as description of the services the agent provides, quality-level and its performance, and availability.

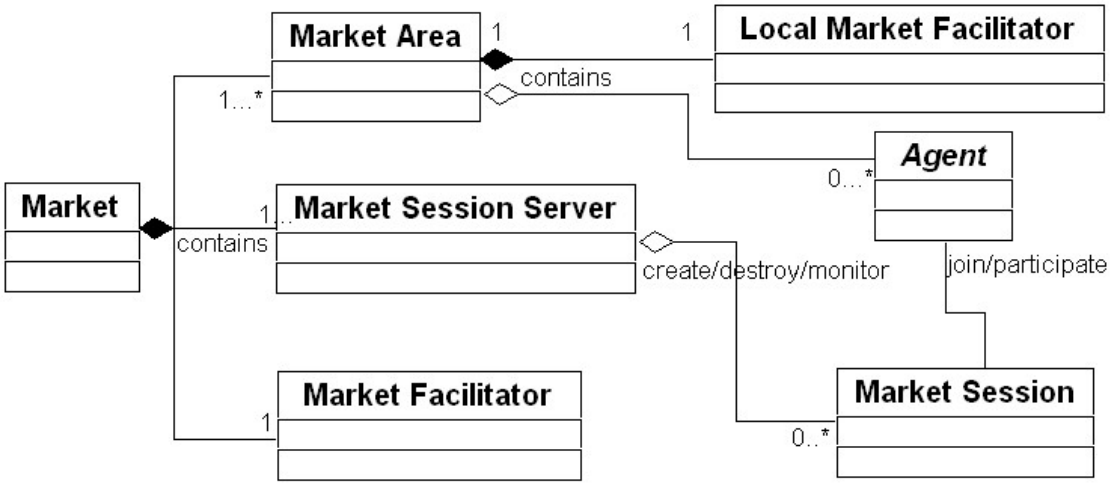

Figure 15 Market relationship

A market is an area or a meeting place where agents such as supplier and customer, resides. A market area has exactly one local market facilitator (LMF), which is an agent that coordinates and represents the agents in their area to the outside world. Agents can be identified being inside a market area if they have registered themselves with the LMF. Agents use the LMF services to access other agents in the market. Agents can advertised services and find out about other agents' services by going through the market facilitator (MF). Agent requires information sharing, participating in negotiations, scheduling for shipping of goods, requesting for quotes or proposal and etc., can join a market session. A market session is a virtual environment that provides services such as managing the passage of time, record all the transaction activities for history playback.

2.3.3 Sequence Diagram

Supplier registration and advertising service

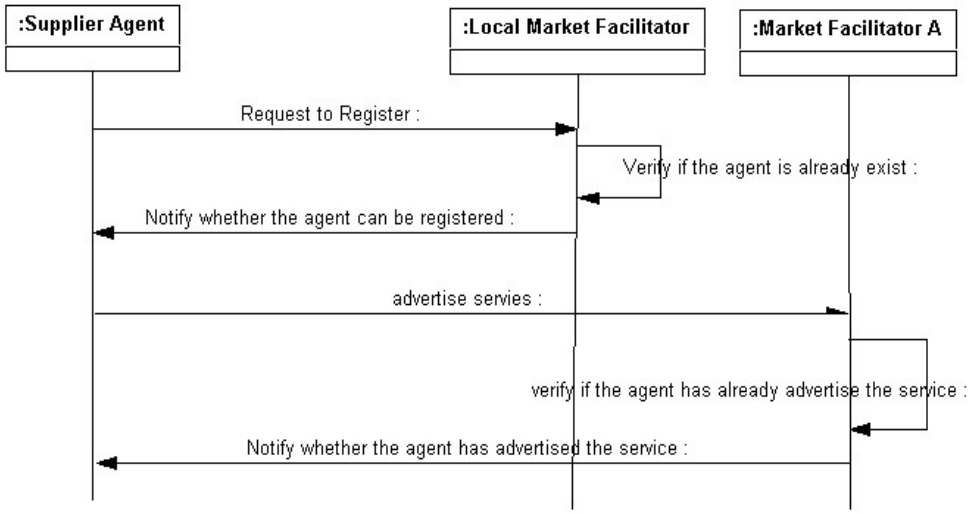

Figure 16 Supplier registration and advertising service

The supplier agent imitates a request for registration with the local market facilitator (LMF). LMF then checks its own registry, and identify whether this particular supplier agent has been registered from before. If it has, then LMF just send a message to notify the supplier that it has registered before. If it has not been registered before, LMF sends a message to notify supplier that it has registered successfully. The suppliers also advertise its service with Market Facilitator (MF). MF also checks its own registry, and identify whether this supplier has advertised the service before. If it has done so, then MF send a message to notify the supplier that it has advertised before. If not, then MF sends a message to notify supplier that it has advertised the service successfully.

Customer finds product in marketplace

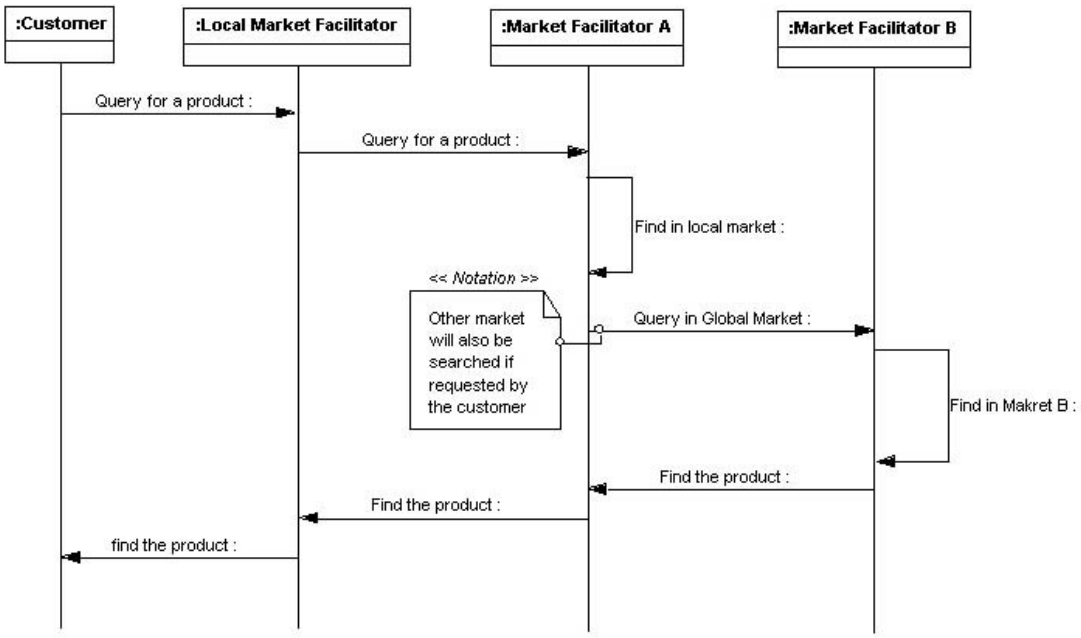

Figure 17Customer find the product in marketplace

The customer queries Local Market Facilitator (LMF) for the product; the message is then forwarded from LMF to Market Facilitator (MF). MF verifies whether this product is registered in the local market. If this product is found in local market, then a message will be sent back to customer to notify the product is found. If the customer specifies to find the product in other market, then MF will also query with other MF in different market. The customer will then be notified when the product is found in other markets.

Customer negotiates with one supplier

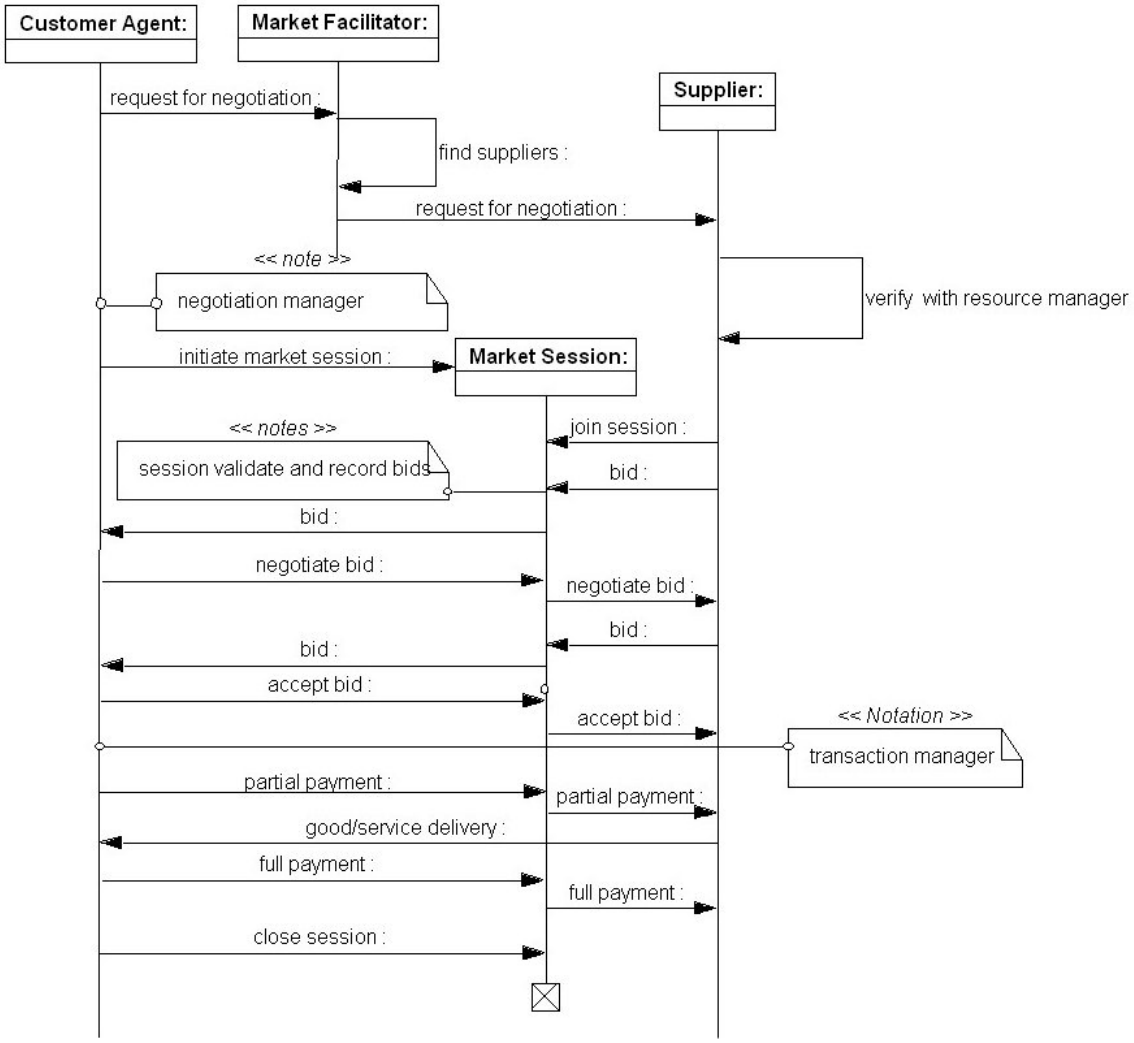

Figure 18 Customer negotiates with one suppl.ier

The customer initiates a request for negotiation to the market facilitator (MF) and initiates a market session through the market session server (MSS). MF then identifies a list of potential suppliers based on the customer selection criteria. MF then forwards the request for negotiation to the list of potential suppliers (in this example, just a single supplier). Interested party can then join the market session and participate in the negotiation process. Market session will keep track of the passage of time, record all the activities during the course of negotiation and etc. Once a mutual agreement is made between the two parties (i.e. the customer accepted the supplier's bid), the customer may decide to pay a deposit for the goods or services, the supplier then deliver the good/services to the customer. The customer than pay the remaining amount for the good/services delivered by the supplier. The deal is considered as close and the market session terminate.

Customer negotiates with multiple suppliers

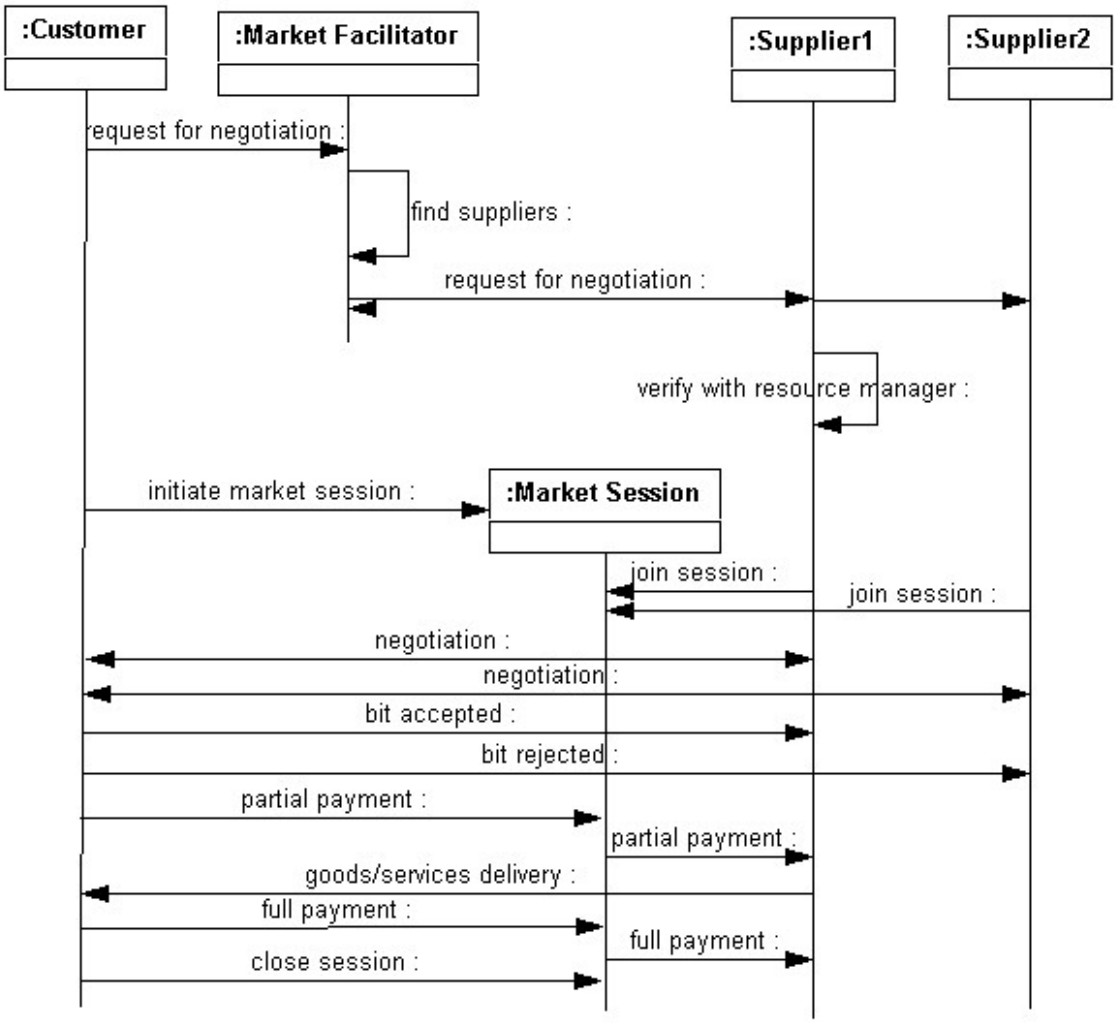

Figure 19 Customer negotiates with multiple suppliers

Market Facilitator (MF) forwards the negotiation request to a list of potential suppliers(In this example, there are two suppliers). Both suppliers joined the market and participate in the negotiation process. The negotiation process continues. After the negotiation process finishes with both suppliers, the supplier compares the negotiation result from both suppliers and makes the final decision. The customer sends bit-accepted messages to supplier 1, and sends bit-rejected message to supplier 2. The payment transaction process continues afterwards

2.3.4 Communication Protocol

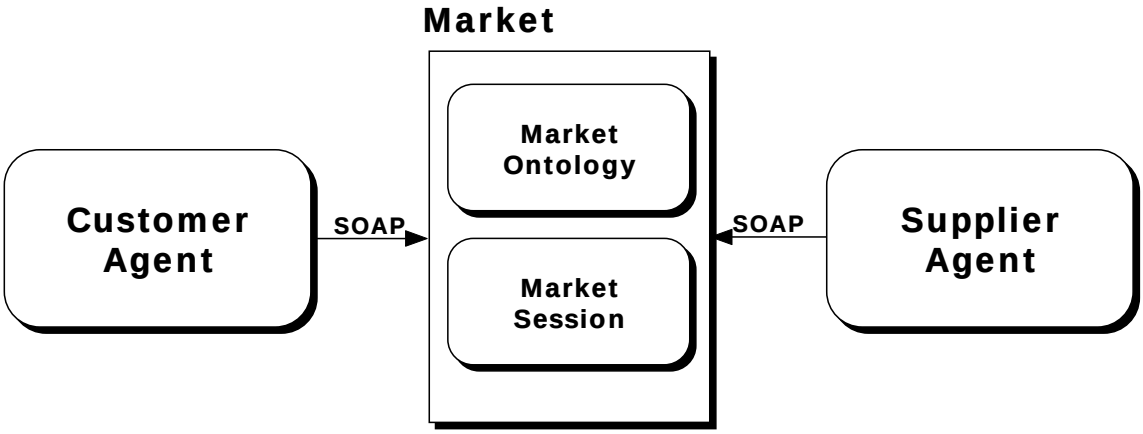

Figure 20 Agent communication

SOAP (Simple Object Access Protocol) is a lightweight message transfer protocol for exchange of structural information in a decentralized distributed environment. In addition, SOAP has been implemented on different hardware and software platform (C++, Java, Visual Basic). For example, agents written in C++ running under Microsoft Window can easily invoke code running on Unix boxes. Thus, using SOAP improves application interoperability.

Agents in eMS use SOAP as the communication protocol and use XML (Extensible-Markup Language) for defining the message data structure. Thus, message exchange is at the level of an HTTP protocol. This approach is convenient for sending data to other agents or end users in XML format and it allows easy interaction and integration with web service that uses SOAP technology.

From this point of view, a customer agent interacts with the market as a SOAP service requester (i.e. request for negotiation) and a supplier agent interacts with the market as a SOAP service provider.

2.3.5 E-R diagram

The following E-R diagram shows the high-level data model of the major entities and their relationships in the eMS system.

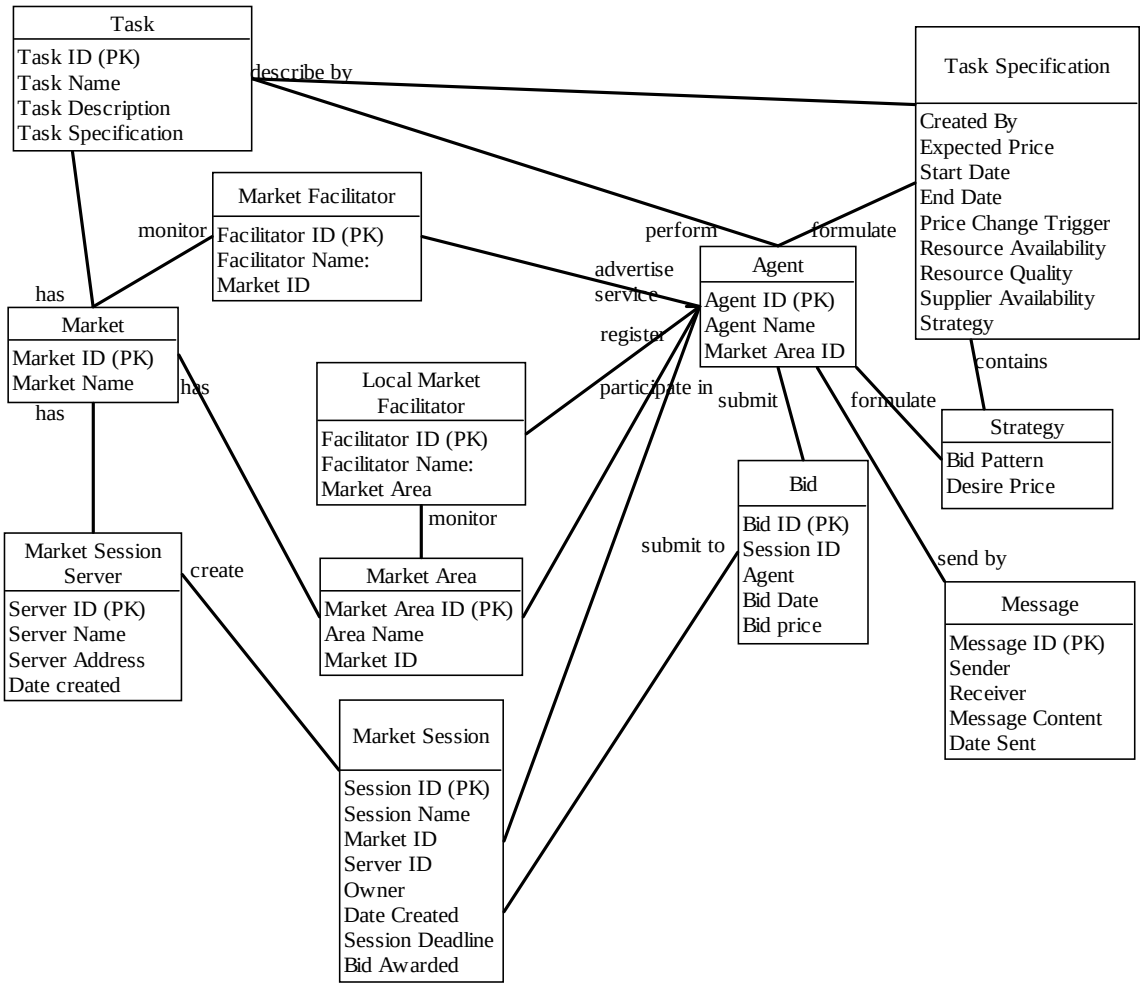

Figure 21 Relationship diagram

2.3.6 Data Dictionary

The following shows the data definitions used in the eMS.

Request for negotiation:

Field	Description	Type
Request ID	Uniquely identified the request	Long
Requester ID	Uniquely identified the requester	Long
Market Area ID	The location the requester resides	Long
Service Request	A description of the	String

	service requested	
Service Selection Criteria	Criteria that is used to select the service provider	String
Market Session ID	The market session that the negotiation is held	Long

Negotiation Request Response

Field	Description	Type
Response ID	Uniquely identify the response	Long
Request ID	The request ID that it responded	Long
Request Status	The status of the request. Accepted = the requested service is available Rejected = the requested service does not exist	String

Bid Request:

Field	Description	Type
Bid ID	Uniquely identify the bid	Long
Bidder ID	Uniquely identify the bidder	Long
Bid Time	The time when the bid is offer	Time
Bid Price	The price that the bidder offer	Double

Bid Request Response:

Field	Description	Type
Bid Response ID	Uniquely Identify the Bid response	Long
Bid Request ID	The Bid request ID it responded to	Long
Negotiated Price	The price that the responder is of interest	Double
Bid Time	The time this response is created	Time

Bid Status	The status of the Bid Open = bid is open for negotiation Closed = bid is closed; Awarded = bid awarded to the bidder	String

Request to open a market session:

Field	Description	Type
Requester ID	Uniquely Identify the requester	Long
Market Area ID	The location the agent resides	Long
Reason to open a session	The reason to open a session: Negotiation, Money Transaction, Service delivery	String

Response to an open market session request:

Field	Description	Type
Request status	The status of the request: Granted, Rejected	String
Session ID	Uniquely Identify the session	Long

Agent:

Field	Description	Data Type
Agent ID	Use to uniquely identify an agent	Long
Agent Name	The name of an agent; an alias	String
Market Area ID	The market area the agent resides	Long

Agent Service/Task:

Field	Description	Data Type
Service Name	Uniquely identify the	Long

	service	
Service Name	A description of the services	String
Service Specificication	The service specification	Service Specification

Service/Task Specification:

Field	Description	Data Type
Service Start Date	The start date of the service	Date
Service End Date	The last date of the service	Date
Service Availiability	The availability of the service (%)	Integer
Service Quality	The quality level of the service. An accumulated percentage based on the quality reported by the customer (%)	Integer
Service Fee	The expected service fee	Double
Service Fee Trigger	A trigger for monitoring a service change. Could be used to monitor price change, progress of the service and etc	Trigger

Request For registration

Field	Description	Data Type
Agent ID	Uniquely identify the agent	Long
Agent Address	The IP address of the agent	IP Address

Request For deregistration

Field	Description	Data Type
Agent ID	Uniquely identify the agent	Long

Simple Agent Query

Field	Description	Data Type
Agent ID	Uniquely identify the agent to be query	Long

Response to simple Agent query

Field	Description	Data Type
Query Status	The status of the query. Exist or Not Exist	String
Agent Address	Address of the queried agent	IP Address

Query Services

Field	Description	Data Type
Service	The service of interest	String
Selection Criteria	The selection criteria for the service	String

Response to Service Query

Field	Description	Data Type
Query Status	The status of the query. Exist or Not Exist	String
Total Found	The total number of query found	Long

2.3.7 Inter-Agent Messages

The following shows the format of some of the inter-agent message.

Description	Message format
Request for negotiation	<RequestForNegotiation> <Request Id>Long</RequestId> <RequesterId>Long</RequesterId> <MarketAreaId>Long<MarketAreaId> <ServiceRequest>String</ServiceRequest> <ServiceSelectionCriteria>String</ServiceSelectionCriteria> <MarketSessionId>Long</MarketSessionId> </RequestForNegotiation>

Description	Message Format
Request Negotiation Response	<RequestForNegotiationResponse> <ResponseId>Long</ResponseId> <RequestId>Long</RequestId> <RequestStatus>String </RequestStatus> </RequestForNegotiationResponse>

Description	Message Format
Bid Request	<BidRequest> <BidId>Long</BidId> <BidderId>Long</BidderID> <BidTime>Time</BidTime> <BidPrice>BidPrice</BidPrice> </BidRequest>

Description	Message Format
Bid Request Response	<BideRequestResponse> <BidResponseid>Long</BidResponseId> <BidRequestId>Long</BidRequestId> <NegotiatedPrice>Double></NegotiatedPrice> <BidTime>Time</BidTime> <BidStatus>String</Status> </BidRequestResponse>

Description	Message Format
Request to	

open a market session	<RequestOpenMarketSession> <RequesterId>Long</RequesterId> <RequesterMarketAreaId>Long</RequesterMarketAreaId> <RequestReason>String</RequestReason> </RequestOpenMarketSession>

Description	Message Format
Response to an open market session request	<ResponseOpenMarketSession> <RequestStatus>String</RequestStatus> <SessionId>Long</SesssionId> </RequestOpenMarketSession>

Description	Message Format
Request for Registration	<RequestAgentRegistration> <AgentId>Long</AgentId> <AgentAddress>IpAddress</AgentAddress> </RequestAgentRegistration

Description	Message Format
Request for Deregistration	<RequestAgentDeregistration> <AgentId>Long</AgentId> </RequestAgentDeregistration>

Description	Message Format
Simple Agent Query	<QueryAgent> <AgentId>Long</AgentId> </QueryAgent>

Description	Message Format
Agent Query Response	< ResponseQueryAgent> <QueryStatus>String</QueryStatus> <AgentAddress>IpAddress</AgentAddress>

	</ResponseQueryAgent>

Description	Message Format
Service Query	<QueryService> <Service>String</Service> <SelectionCriteria>String</SelectionCriteria> </QueryService>

Description	Message Format
Response to Service Query	<ResponseQueryService> <QueryStatus>String</QueryStatus> <TotalFound>Long</TotalFound> </ResponseQueryService>