

# Provision of Market Services for eCo Compliant Electronic Marketplaces

Sena Arpinar

Asuman Dogac

Software Research and Development Center (SRDC),  
Middle East Technical University (METU), 06531 Ankara, Turkey.  
[asuman@srdc.metu.edu.tr](mailto:asuman@srdc.metu.edu.tr)

## Abstract

The progress and wider dissemination of electronic commerce will be facilitated through interoperability infrastructures. Commerce Net's eCo framework is a promising effort in this direction. In the eCo framework, businesses participate in a marketplace through standard interfaces for their services and by exchanging standardized documents. This framework does not specify any further interfaces for the services a marketplace itself may offer.

In this paper, we demonstrate that a rich set of marketplace-specific services such as automated discovery of the needed services, comparison shopping, and negotiation can be offered to market participants by introducing a marketplace as an eCo business. For this purpose, a previously developed marketplace, namely MOPPET, is made eCo-compliant. We demonstrate that introducing MOPPET as an eCo business increases the functionality of the eCo market in the sense that several market specific services become available to the market participants.

## 1 Introduction

In today's Internet businesses a level of interoperability is necessary which allows consumers and businesses to seamlessly and dynamically come together and do business without ad hoc and proprietary integrations. Such a level of interoperability involves being able to find potential business partners, discovering the goods or services they offer and investigating whether the involved systems can in fact make dynamic interoperation possible. The questions that need to be answered in this process are:

- What other businesses can I find?
- What services do they offer?
- What kind of interactions do they expect?
- What protocols do they follow?

- Can our systems communicate?
- What application interfaces do they provide?
- Are our interfaces compatible?
- What information must we exchange?

CommerceNet's eCo framework [eCo 99] provides an interoperability framework to address these issues. In the eCo framework, businesses agree on a common method of **describing** what they do rather than agreeing on standards of what they do and how they do it.

The eCo Framework consists of an architectural specification and a semantic specification. The Architectural Specification presents information about an e-commerce system in seven different categories (layers) where "networks" (layer 1) contain "markets" (layer 2) where "businesses" (layer 3) provide and use "services" (layer 4) which conduct "interactions" (layer 5) that exchange "documents" (layer 6) containing "information items" (layer 7). The eCo Semantic Specification, on the other hand, provides a sample set of business documents that can be used inside the eCo framework. These can be used as is, or extended and modified to meet specific needs [eCo 99].

The network layer contains various eCo compliant markets for providing or obtaining specific goods and services like computers, phones, or books. In the market layer, for a specific market like computers, their participating businesses are listed like Dell, or IBM. In the business layer, the services provided by a business are listed, for example catalog browsing, ordering products, making payment, or checking order status. At the service layer, the possible interactions are listed in terms of input and output documents and an optional execution URI. For example, the interaction for purchasing a computer can be an order document as input, and an invoice as output defined in XML using the corresponding Common Business Library (CBL) DTDs [CBL 00]. A service may invoke other services in order to

complete that service. These relationships among services are described in the interactions layer. In other words, this layer describes a “choreography” of interactions that may take place when a service is invoked and the types of messages which are exchanged during each interaction. The document types exchanged in an interaction are described by the document layer which lists its data elements, if any. At the data element layer, details of data elements are presented.

Each layer of an eCo-compliant e-commerce system presents information about itself. By examining this information, the users can:

- locate the system
- understand what it is for
- recognize what market(s) it participates in
- identify protocols the system uses to communicate
- discover what documents the system uses to conduct business
- learn how to interoperate with the system

This information is provided through published interfaces specific to each layer which involves specific sets of queries that needs to be implemented for a system to be eCo compliant. These queries return documents (i.e. property sheets) describing each layer.

This framework does not specify any further interfaces for the services a marketplace itself may offer. In this paper, we demonstrate that a rich set of marketplace-specific services like comparison shopping, negotiation, and locating and using the services automatically, can be offered to market participants by introducing a marketplace as an eCo business. For this purpose, we use the MOPPET electronic marketplace [ADT 00].

In the MOPPET architecture, commerce processes are modeled as adaptable agent-based workflows. Electronic commerce, which is a complex business process itself, cannot be modeled effectively by current marketplaces, which support buyer/seller behavior in an overly simplistic manner. For this reason, MOPPET exploits workflow systems to model buying or selling processes. In addition, the higher level of the abstraction provided by workflow technology makes the customization of electronic commerce processes for different users possible. The agent-based implementation, on the other hand, provides for a highly reusable and flexible component-based system as well as a negotiation

capability and the ability to more easily adapt to dynamic changes in the environment.

MOPPET is made eCo compliant by implementing the necessary interfaces at the market layer. However by observing that it provides some market specific services, it is also introduced at the business layer so that other businesses can find the services offered and use them.

The following functionalities are supported by the eCo compliant version of MOPPET:

- 1 The eCo compliant intelligent interface agent of MOPPET can directly find the item a user is looking for, contact the related business and configure and use their services automatically.
- 2 It becomes possible to define a complex purchasing (or selling) scenario. The choreography of interactions that should take place when a service is invoked is defined through a workflow system. MOPPET provides workflow templates that are modified according to user input collected by the eCo compliant intelligent interface agent.
- 3 MOPPET also provides a comparison shopping facility that includes a negotiation capability through its coordinating agents.

This paper is organized as follows: Section 2 summarizes the MOPPET architecture. How to introduce a marketplace as an eCo business is presented in Section 3. Section 4 explains the way a service is executed in eCo-compliant MOPPET. Finally, Section 5 concludes the paper.

## 2 MOPPET Architecture

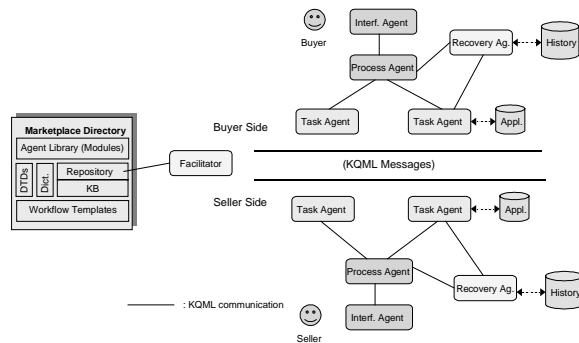
Businesses that participate in the MOPPET marketplace [ADT 00] can sell or buy products and/or services through comparison shopping, advertise their products and/or services to the interested parties, or simply search the catalogs of other businesses. The marketplace supports both business-to-business and business-to-consumer commerce.

MOPPET provides marketplace-specific services through predefined workflow templates that are stored in the marketplace directory. Each business can access and use these services through an appropriate Interface agent. Workflow process instances are generated from the templates by adapting them to the current needs and properties of

the user. Workflow processes are then executed by a number of coordinated autonomous agents.

There are four main types of agents for realizing an electronic commerce process: interface agents, process agents, task agents and recovery agents (Figure 1). Interface agents provide intelligent interfaces to users of a marketplace: individuals or businesses. Workflow processes are enacted by process agents who contact a number of task agents and other process agents to achieve individual activities in a process. These activities can be simple tasks such as catalog search or complex sub-processes such as a payment process. Task agents act as wrappers of simple tasks (applications).

Depending on the type of task they are in charge of, the task agents have different capabilities. For example, for user tasks, they have a worklist management capability. Recovery agents take the initiative when an unanticipated change or a failure occurs during process execution. They inform process agents about a new path for the execution. Agents communicate with each other by sending XML [XML 99] documents in KQML messages [LF 97]. There is also a Facilitator agent in the marketplace that helps agents to find each other. It collects advertisements in terms of the capabilities from all agents in the system and stores them in a repository to answer later queries about agents.



**Figure 1. The MOPPET Architecture**

There are four main types of roles in the marketplace, each modeled by a different workflow template: active suppliers, passive suppliers, active consumers and passive consumers. Active suppliers try to sell their products by initiating a selling process. Passive ones wait until a consumer makes a request to purchase its products. Similarly, active consumers make requests for products/services to buy and passive ones wait for an active supplier to propose a product. For each of the roles there are predefined workflow templates stored in the marketplace

directory. Below we provide pseudocode for two such templates. Other templates and further details of MOPPET can be found in [ADT 00].

#### Workflow template for an active consumer

```

process wf-temp1
  for each of the items do
    suppliers = findSuppliers (item[i]);
    /* find out who sells the specified item */
    for each of the suppliers do
      query (item[i], supplier[j]) ;
      /* query the catalogs of suppliers for details */
      if result is satisfactory do
        doNegotiate[j] = true;
        /* record whether supplier j should enter */
        /* into negotiation */
      endif
    endfor
  finalSupplier = negotiate (criteria, doNegotiate,
                             suppliers, result);
  /* find the supplier from whom the item is to */
  /* be purchased */
  payment (paymentInfo, finalSupplier);
  if payment is done do
    deliver (item[i], customerInfo);
  endif
endfor
endprocess

```

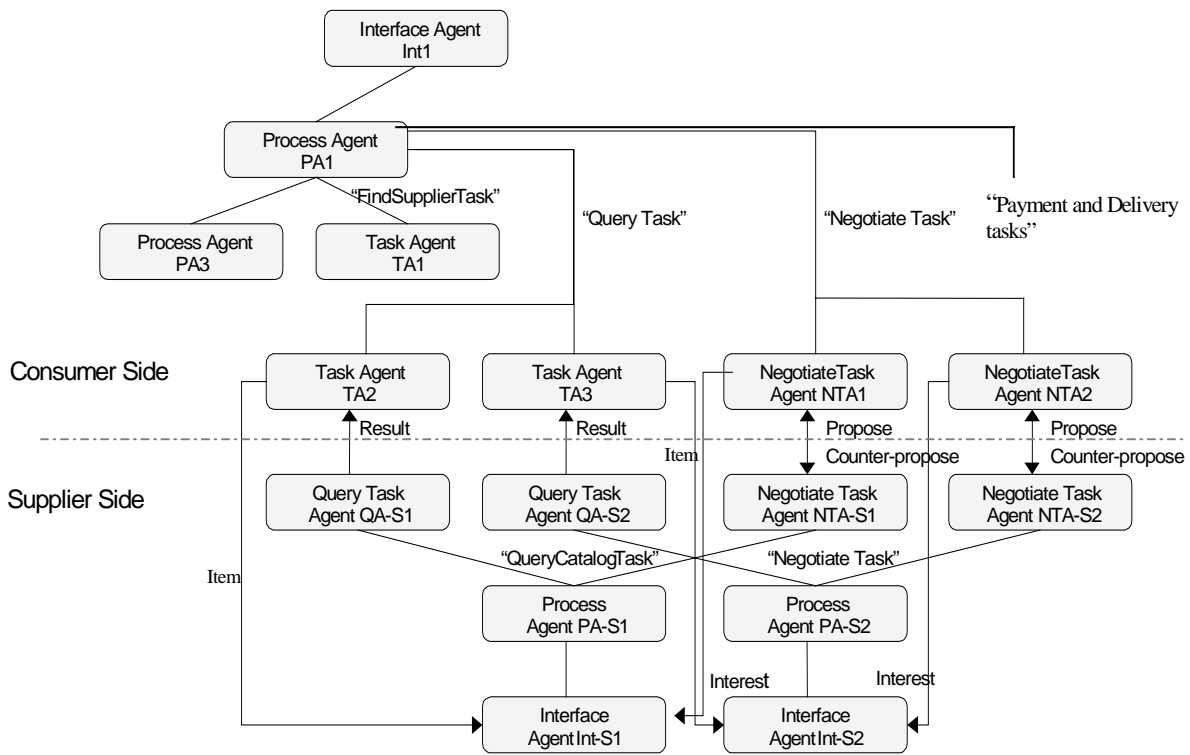
#### Workflow template for a passive supplier

```

Process wf-temp4
  QueryCatalog (item);
  /* result is sent to the agent who originally
  sends the query */
  if interest arrives do
    doSell = negotiate (item);
    /* if the consumer is interested, start */
    /* negotiation */
  endif
  if doSell is true do
    payment (item, paymentInfo, supplierInfo);
    deliver (item, customerInfo);
  endif
endprocess

```

Having defined these templates, we now describe briefly how a buying workflow is executed in MOPPET with coordinating agents for an active consumer and a passive supplier. The agents and the interactions among them are shown in Figure 2. Payment and Delivery tasks are omitted in this figure.



**Figure 2. Interactions of Agents**

A workflow instance for a consumer is started by a process agent after getting the modified workflow template from the interface agent. The items to be purchased, the negotiation criteria, the customer payment information etc. are obtained and added to the template to create a workflow instance by the interface agent. Then, the process agent activates task agents for the subsequent activities in the workflow template. The supplier workflow instance is started when the "query" is sent by the consumer process agent to the interface agent of the supplier. The query task agent of the supplier is activated by the arrival of the "item" parameter. Similarly, the negotiate task agent is activated by the "interest" parameter. Negotiation is performed by negotiate task agents on both sides via propose and counter-propose messages. According to the result of negotiation, the item is purchased from one of the suppliers via the activation of task agents responsible for the payment task on both sides.

### 3 eCo Interfaces of MOPPET

In this section, we show how to make MOPPET an eCo compliant marketplace. In the following, sample

eCo interfaces for different layers are provided. Then, a scenario is described where complex marketplace specific services are automatically configured from the existing services of eCo compliant businesses.

MOPPET implements an eCo market layer since it is itself a marketplace. In addition, it acts as an eCo business since it provides services to the businesses in the marketplace. Thus, there is also an implementation of eCo business layer for MOPPET. These implementations involve providing MarketPropertySheet and BusinessPropertySheet for MOPPET in XML as a first step.

MarketPropertySheet gives information about market maker, its operator and other market specific information. BusinessPropertySheet describes the type of business, its location, web page and other business related information. Note that these documents are returned as answers to the queries <http://www.srdc.metu.edu.tr/moppet/MarketGetProperties> and <http://www.srdc.metu.edu.tr/moppet/BusinessGetProperties> respectively.

MOPPET provides many services to its participant businesses and also to individual users as described

in Section 2. These services are listed in response to the “BusinessGetServices” query of the MOPPET business interface as given in Table 1. Each service is realized through a workflow process enacted by eCo compliant agents. Agents are eCo compliant in the sense that they access and understand the meta-data provided in different layers of eCo interfaces of participant businesses described through various “Type Registries” associated with each layer. These type registries are intended to provide a common ontology of information to enable an interested party to use offered services. Each participant business should provide them through a pre-defined ontology so that MOPPET agents can access and use their services automatically.

```
<?xml version="1.0" >
<EcoInterfaceList xmlns="http://www.commerce.net/eco">
  <Head>...</Head>
  <Interface type="Service">
    <Head>
      <Identifier>http://www.srdc.metu.edu.tr/moppet/
        services/CatalogSearch</Identifier>
      <TimeToLive>86400</TimeToLive>
      <Label>Catalog Search</Label>
      <Description>facilitates search of products or
        services from supplier catalogs registered
    </Description>
    </Head>
  </Interface>
  <Interface type="Service">
    <Head>
      <Identifier>http://www.srdc.metu.edu.tr/moppet/
        services/InterestSearch</Identifier>
      <Label>Interest Search</Label>
      <Description>facilitates search of products or
        services consumers currently interested
    </Description>
    </Head>
  </Interface>
  <Interface type="Service">
    <Head>
      <Identifier>http://www.srdc.metu.edu.tr/moppet/
        services/Buy</Identifier>
      <Label>Buy Product</Label>
      <Description>implements buying process
    </Description>
    </Head>
  </Interface>
  <Interface type="Service">
    <Head>
      <Identifier>http://www.srdc.metu.edu.tr/moppet/
        services/Sell</Identifier>
      <Label>Sell Product</Label>
      <Description>implements selling process
    </Description>
    </Head>
  </Interface>
  <Interface type="Service">
    <Head>
      <Identifier>http://www.srdc.metu.edu.tr/moppet/
        services/advertise</Identifier>
      <Label>Advertisement</Label>
      <Description>facilitates advertisements of products
        (services) of suppliers and interests
        (products or services) of consumers
    </Description>
  </Head>
</EcoInterfaceList>
```

```
</Description>
</Head>
</Interface>
</EcoInterfaceList>
```

**Table 1. Service List of MOPPET**

```
<EcoInterfaceList xmlns="http://www.commerce.net/eco">
  <Head> ... </Head>
  <Interface type="Interaction">
    <Head>
      <Identifier>http://www.srdc.metu.edu.tr/moppet/
        interaction/GetProductInfo</Identifier>
      <Label>Get Product Information</Label>
      <Description>obtains product specification from
        the user.
    </Description>
    </Head>
  </Interface>
  <Interface type="Interaction">
    <Head>
      <Identifier>http://www.srdc.metu.edu.tr/moppet/
        interactions/FindSuppliers</Identifier>
      <Label>Find Suppliers</Label>
      <Description> It locates possible suppliers of the
        product (or service) in question.
    </Description>
    </Head>
  </Interface>
  <Interface type="Service">
    <Head>
      <Identifier>http://www.srdc.metu.edu.tr/moppet/
        services/CatalogSearch</Identifier>
      <Label>Catalog Search</Label>
    </Head>
  </Interface>
  <Interface type="Interaction">
    <Head>
      <Identifier>http://www.srdc.metu.edu.tr/moppet/
        interactions/Negotiate</Identifier>
      <Label>Negotiate</Label>
    </Head>
  </Interface>
  <Interface type="Interaction">
    <Head>
      <Identifier>http://www.srdc.metu.edu.tr/moppet/
        interactions/Payment</Identifier>
      <Label>Payment</Label>
    </Head>
  </Interface>
  <Interface type="Interaction">
    <Head>
      <Identifier>http://www.srdc.metu.edu.tr/moppet/
        interactions/Delivery</Identifier>
      <Label>Delivery</Label>
    </Head>
  </Interface>
</EcoInterfaceList>
```

**Table 2. Interactions of "Buy"**

Only the service labeled as "Buy" is detailed in the paper to serve as an example and others can be found by querying interfaces at <http://www.srdc.metu.edu.tr/moppet>. The interactions that make up the "Buy" service are described in Section 2, where a workflow template for an active consumer is provided. This

template also gives the "choreography" of this service.

In Table 2, the list of interactions of "Buy" service as a response to <http://www.srdc.metu.edu.tr/moppet/services/Buy/ServiceGetInteractions> query is given. Interactions in the list need not be in the order in which they will be used when the service is executed. In addition, some interactions may be repeated (depending on the choreography) or may not be used when the service is being enacted since agents in the system adapt services according to specific instances of users and/or businesses.

## 4 Example

In this section, we describe how the eCo compliant businesses can be exploited by MOPPET agents. When a business publishes its eCo interfaces for the services it offers, MOPPET agents can use this information to interoperate with that business in realizing its own services. This in return provides more complex market specific services to be offered to an eCo compliant market.

Throughout this section, we assume that two eCo compliant computer companies, "CompWorld" and "E-comp" have been registered to MOPPET. Each company provides services to be used by other businesses in the marketplace. The list of services given by the Service Layer implementation of CompWorld is provided below (eCo interfaces are omitted due to space limitations).

1. Browse Catalog - gives the catalog of the company to the interested parties as an XML-CBL document. It has only one interaction, namely BrowseCatalog having an input document "request" and an output document "catalog".
2. Sell - this service involves selling the requested item to the consumer at a fixed price. It obtains the product information and the customer information and enacts the payment and shipping processes.

E-comp, on the other hand, provides the following services:

1. Catalog Search - this service is made up of two interactions namely, BrowseCatalog and SearchItem. BrowseCatalog returns the "catalog" of the company as an output document and SearchItem, gets this "catalog" and the "item" as

input documents and outputs the search result "product info".

2. Get Negotiation Criteria - this service returns the negotiation criteria of the company for the specific product in question. Ranges for the price, deadline, duration, quality etc. for the product, the weights of each given criterion and the negotiation functions to be used (linear, polynomial, or exponential) constitutes the negotiation criteria of the company. The existence of this service means that this company accepts negotiation while selling products.
3. Sell - This service is similar to the "Sell" service provided by the CompWorld. But, E-comp does not directly support the delivery of the product rather it outsources it to the third parties whose delivery service needs to be discovered.

In the following, we describe how the MOPPET service "Buy" exploits the services offered by these businesses to come up with a comprehensive service. The first interaction in "Buy" is to get product specifications from the consumer. A user may wish to buy several related products and there can be an order relationship among these products. For example, s/he may wish to buy a computer and a printer but the printer should be bought only if the purchase of the computer succeeds. These control and data flow dependencies are expressed as a workflow and it is the responsibility of the interface agent to obtain this information from the user.

Then, for each product requested, the "findSuppliers" interaction is executed to locate possible suppliers of the product. The "findSuppliers" interaction of MOPPET is realized by a specialized task agent that investigates businesses' published interfaces to find out whether they supply the required kind of product. Note that, if the businesses have already advertised themselves through the "advertisement" service in this marketplace, it is easier for the task agent to locate them. The output of "findSuppliers" interaction is thus a list of suppliers.

The next interaction (for each product) is to query the catalogs of these suppliers. Note that different businesses in the market may provide different capabilities when it comes to querying the catalogs. In the running example, CompWorld only publishes its full catalog. Therefore this catalog requires an agent with a querying capability which is readily provided by MOPPET. However, E-comp allows its catalog to be queried on a per item basis. The execution of the "Query" interaction is therefore performed as follows: The consumer-side query task agent forms the query and sends it to the supplier's

interface agent. The interface agent of the supplier forwards it to the process agent that in turn activates a query task agent on the supplier side.

As this example indicates, the behavior of query task agents differs for different businesses. In other words, businesses' meta-data of services may demand agents with different behaviors and capabilities. The query task agent responsible for querying *CompWorld*, needs to have full querying capability since this business only provides a service for supplying the catalog, namely "Browse Catalog". On the other hand, the query task agent associated with *E-comp* simply executes the service "Catalog Search" by sending the query as the input document and gathering the output document as the result.

The capabilities offered by the businesses' eCo interfaces also effect the agent behaviour at the negotiation phase. The negotiate task agent (supplier side) associated with *CompWorld* simply returns the fixed price for the product and does not accept further proposals. The negotiate task agent running for *E-comp*, however, enters into a negotiation by gathering the criteria of the company via the "Return Negotiation Criteria" service. It uses the ranges proposed by the user for attributes like price, delivery date etc., and the chosen negotiation function in determining the next proposal and finally determining the end of the negotiation.

In the example, both companies provide "Sell" services which includes the payment process. Thus the agents responsible for performing the "Payment" service simply execute these services. However, since *E-comp* does not have its own delivery service, the "deliver" service needs to be executed in this business's workflow process instance. Adapting the workflow template to specific cases is performed by the interface agent when the company advertises itself, by examining the eCo interfaces and type registries.

This example demonstrates that by introducing an electronic marketplace like MOPPET at the business layer of the eCo architecture, complex marketplace services like "Buy" that may involve finding the suppliers, negotiation during purchase, etc., can be configured dynamically and automatically by using the existing services of the eCo compliant businesses.

## 5 Conclusions

The eCo specification [eCo 99] provides a framework for electronic commerce interoperability that allows consumers and businesses seamlessly and dynamically come together and do business without ad hoc and proprietary integrations. This paper first describes how to make an existing electronic marketplace eCo compliant and then demonstrates how more complex marketplace specific services can be automatically configured from existing services of eCo compliant businesses by introducing a marketplace at the business layer.

## References

- [ADT 00] Arpinar S., Dogac A., Tatbul N., "An open electronic marketplace through agent-based workflows: MOPPET", Intl. Journal on Digital Libraries, Vol. 3, No. 1, July 2000, pp. 36-59.
- [CBL 00] Common Business Library <http://www.commerceone.com/xml/cbl/index.html>
- [eCo 99] "eCo architecture for electronic commerce interoperability", <http://eco.commerce.net>.
- [LF 97] Y. Labrou, T. Finin, "A Proposal for a new KQML Specification", Technical Report TR-CS-97-03, University of Maryland, Baltimore.
- [XML 99] Extensible Markup Language (XML) 1.0. W3C Recommendation.