An Electronic Market Place:
Generic Agent Models, Ontologies and Knowledge

Marcel Albers\textsuperscript{1}, Catholijn M. Jonker\textsuperscript{2}, Mehrzad Karami\textsuperscript{1}, Jan Treur\textsuperscript{2}

\textsuperscript{1}CRISP BV, ‘s Hertogenbosch, The Netherlands
URL: http://www.crisp.nl. Email: {malbers, mkarami}@crisp.nl

\textsuperscript{2}Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
URL: http://www.cs.vu.nl/~{jonker,treur}. Email: {jonker,treur}@cs.vu.nl

Abstract
In this paper an agent-based electronic market architecture is described. The architecture has been designed in a compositional manner using elements of agent technology and knowledge technology. It is shown how a combination of generic agent models, ontologies and knowledge provides an adequate approach to address the dynamic, distributed and knowledge-intensive character of the application.

1. INTRODUCTION

In this paper an agent-based architecture for electronic markets is introduced. The application area of electronic markets has an interesting combination of characteristics:

- Processes in an electronic market take place in a distributed manner. A number of autonomous processes or actors are distinguished which operate in interaction with each other.
- The behaviour requirements for software systems supporting an electronic market are of a highly dynamic nature. Supporting systems should not only be able to behave autonomously in a reactive manner with respect to individual buyers or sellers, but also in a pro-active and adaptive manner, putting forward proposals that were not immediately asked for by users.
- The domains are highly knowledge-intensive. Buyers and sellers express themselves in different terms (ontologies), when they formulate properties (product models) of or demands for products. Besides these ontologies, also domain-specific knowledge relating the different ontologies plays an important role.
- The supporting software should be easy to maintain. For example, new buyers or sellers, or new product types will have to be included quite often. The system design will have to be transparently structured and flexible for such changing circumstances, making use of conceptually specified reusable components.

To address this combination of characteristics, agent technology and knowledge technology are used. Agent technology addresses the first two characteristics, whereas knowledge technology supports the last two characteristics. Within the compositional development method for multi-agent systems DESIRE (cf. [2]), both ingredients are combined in a transparent manner. After a brief introduction of the overall architecture in Section 2, a description is given in Section 3 of the product-related ontologies used. In Section 4 the domain-specific knowledge used to relate the different ontologies and product models is presented. Section 5 briefly describes some related work.

2. THE OVERALL MARKET PLACE ARCHITECTURE

The agents in the market place have been designed and developed using DESIRE, a compositional development method for multi-agent systems (Design and Specification of Interacting Reasoning components; cf. [2] for the underlying principles, and [1] for a case study) In Section 3.1 the generic broker agent model used is briefly described. In Section 3.2 the overall multi-agent architecture for the electronic market place is presented.
2.1 A Generic Broker Agent Model

The process of brokering as often occurs as a mediating process in Electronic Commerce involves a number of activities. For example, responding to buyer requests for products with certain properties, maintaining information on customers, building customer profiles on the basis of such customer information, maintaining information on products, maintaining provider profiles, matching buyer requests and product information (in a strict or soft manner), and responding to new offers of products by informing customers for whom these offers fit their profile. The generic broker agent architecture depicted in Figure 1 supports such activities (cf. [8]).

![Figure 1: Generic broker agent model](image)

Within the broker agent model, a number of processes are distinguished that support interaction with the other agents. First, a process that manages communication with other agents, modelled by the component agent interaction management in Figure 1. This component analyses incoming information and determines which other processes within the agent need the communicated information. Moreover, outgoing communication is prepared. Next, the agent needs to maintain information on the other agents (e.g., consumers and providers) with which it co-operates: maintenance of agent information. The component maintenance of world information is included to store the world information (e.g., information on attributes of products). The process own process control defines different characteristics of the agent and determines foci for behaviour. The component world interaction management is included to model interaction with the world (with a database or with the World Wide Web world, for example): initiating observations and receiving observation results.

The agent processes discussed above are generic agent processes. Many agents perform these processes. In addition, often agent-specific processes are needed: to perform tasks specific to one agent, for example directly related to a specific domain of application. The broker agent may have to determine proposals for other agents. In this process, information on available products (communicated by provider agents and kept in the component maintenance of world information), and about the interests of agents (kept in the component maintenance of agent information), is combined to determine which agents might be interested in which products.
2.2 The Overall Architecture

The generic broker agent model has been used as a building block to design the electronic market place. At the process abstraction level of the market place, the internal structure and behaviour of the agents is hidden. The following types of agents have been specified for the virtual market place; each of these agents reuse (parts of) the generic broker agent model described in Section 2.1:

- **Interface agents**
  The communication between customers and the market place is maintained through interface agents. Users of the market place specify and forward their activities to the market place and receive information from the market through the interface agents.

- **Tent agents**
  They represent and control the products available in the market place. The available products are classified based on the category they belong to. For instance, the category used to represent vehicles is different from the category used to represent the insurance products. Each tent represents a distinct category of products. So far, only one tent, i.e. cars, has been designed and specified in order to test the prototype system.

- **Broker agents**
  Brokers act as intermediaries between the consumers and providers. It is through a broker agent that the buyers and sellers can communicate with each other without having to reveal their private information to each other.

3. THE MARKET PLACE ONTOLOGIES

In this section, we will focus on the ontology design of a market place. By this we mean the specification of the knowledge structures used to define concepts and the relationship among these. The primary focus when designing the ontology model of the market place is to satisfy and extend the share and reuse of ontologies within the market place infrastructure.

3.1 The Design of Ontologies

Ontology, as Gruber (cf. [6]) describes it “explicit specification of a conceptualization”, provides a vocabulary for talking about a domain. In his paper, Gruber writes:
An ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an Ontology is a systematic account of Existence. For AI systems, what ‘exists’ is that which can be represented. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. Thus, in the context of AI, we can describe the ontology of a program by defining a set of representational terms. In such an ontology, definitions associate the names of entities in the universe of discourse (e.g. classes, relations, functions, or other objects) with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms. Formally, an ontology is the statement of a logical theory. 

In [7] the following are listed as the most essential criteria during the design of ontology:

- **Clarity** The objectiveness of the definitions is crucial for a well-defined ontology. This means that no social or computational context dependency must exist when defining the ontology. In short, clarity refers to the effective definition of the terms (relations and concepts) in order to communicate the desired meaning. These definitions must be documented in natural language.

- **Coherence** An ontology should sanction inferences that are consistent with the definitions. At least the defining axioms should be logically consistent. This also includes the informal definitions of concepts, for example in a natural language document. Contradiction at any level violates the coherency principles of ontology.

- **Extendibility** A well structured ontology should provide the means to define new terms (extensions) without requirements for revision of the already existing ontology. Adding new terms, concepts, and relations to an existing vocabulary, for special uses, should not require revision of the existing ontology.

- **Minimal encoding bias** Encoding Bias, as Gruber defines it, happens when representation choices are made purely for the convenience of notation or implementation. This means that the conceptualization depends on the symbol-level encoding instead of being specified only at the knowledge level. The reason this bias must be minimized is because knowledge-sharing agents may be implemented in different representation systems and styles of representation.

- **Minimal ontological commitment** Ontological Commitment refers to the constraints defined on the concepts and relations which are sufficient to support the intended knowledge sharing activities. As [7] states, an ontology should make as few claims as possible about the world being modeled, allowing the parties committed to the ontology freedom to specialize and instantiate the ontology as needed. This can be done by defining the weakest theory and defining only those terms that are essential to the communication of knowledge consistent with that theory.

Within the domain of agent paradigm there is also a need to design models in which knowledge is context-independent, well defined, and structured with the purpose of having sharable and reusable components. Share and reuse of knowledge is one of the primary goals of agent-based models in order to connect and communicate with each other. Agents can communicate with each other without having to operate on a globally shared theory. In order to be able to communicate with each other, agents need to be coherent. This does not mean that the same identical ontology is used by all the agents, but it means that “Ontological Commitment”, as Gruber calls it, must exist among the community of agents. Each agent (using an ontology class) knows things that the other agents do not know, but each of them knows enough about the concepts and relations in order to maintain the communication with other agents in the environment. Agents sharing the same vocabulary can also contain different knowledge; cf. [6]:
“Pragmatically, a common ontology defines the vocabulary with which queries and assertions are exchanged among agents. Ontological commitments are agreements to use the shared vocabulary in a coherent and consistent manner. The agents sharing a vocabulary need not have a knowledge base; each knows things that the other does not, and an agent that commits to an ontology is not required to answer all queries that can be formulated in the shared vocabulary.” [6].

During the ontology design of the marketplace, we have based our design process on the principles discussed above. A hierarchy relationship between the relations and concepts exist, starting from the generic, context-independent structure towards more specific knowledge types. However, applicability in completely other areas has not been taken into account. The applicability of the ontology may be limited to the context of virtual market places and commerce.

3.2 Product Models

Due to the amount of information, and the limited size of this paper, we limit ourselves to describing only the product-related ontology models, namely

- the basic product ontology,
- the provider perspective ontology, and
- the consumer perspective ontology.

The aim is to create a structure that can be used in order to define and declare any kind of product. A product is defined using four other models. Figure 3 illustrates an example of a basic product model and the sub-models of which it is composed.

![Figure 3: A basic product model](image)

The following four models are distinguished as the underlying building blocks of the basic product model:

- **Ordination model**
  
  The ordinal model is the product specific identifier. Ordination characteristics and entities are used in order to identify and recognize a product when referring to it. These entities are used in combination with other aspects in order to make referencing possible. For example, Manufacturer, Model, Type, and Year identify the ordination model for category cars (BMW, 316I, 1997…).
Physical model
The physical model describes a product based on its physical existence. A product is considered to be an entity existing of physical parts assembled together in order to define the physical existence of it. A car as a physical entity, consists of components such as motor, wheels, or brakes. Each of these components can also exist of other, smaller components. As it appears, the physical model of a product can be defined as a tree structure where the components and entities are related to each other based on this hierarchical structure. The physical model defines a product from the material, existing-of point of view.

Functional model
A functional model defines a product from the property-of point of view; it provides a description of what a product can do or is used for. Many concepts and properties in the functional model are usually derived from the physical model of the product. For example, safety is a functional characteristic of a car, but it can be derived from the physical aspects such as existence of safety belts, bumpers, and airbags in a car.

Presentational model
In a real, non-virtual market place, representation of a product is done through seeing and feeling. When buying a car, you can see it, make a test drive, or even check the components of the car if necessary. The physical presence of car makes it possible to represent the car to the customer. In the virtual world however, this is different. The physical presence of many product types is not realized. This means that some other mediating factors must be available to present a product. Which entities can be used to define and specify the presentational model depends from one side on the type of product and on the available virtual facilities on the other side. Images, animations, audio and video clips are examples of entities by which a product can be presented. A car for instance, can be presented by a video clip or a text format specification.

For some product types, such as computer games or other software applications a presentation model consists of a part of the game or a demo. It depends therefore to the product type how the presentational model is defined and represented.

3.3 Ontologies for Product Models
The market place ontologies have been designed and formally specified within DESIRE. A distinction is made between product-related ontology models of the following three categories:

- **Generic information types**
  General, domain-independent concepts and their relations are identified here. For instance, specifying that a physical, functional, ordination, and presentational model exist. No detail is given about the domain-specific content of any of these building blocks.

- **Domain specific information types**
  In addition to the generic aspects, domain-specific, detailed concepts and terms are defined. Specifying that a car has a motor, chassis, or wheels are examples of the type of knowledge specified and represented by the domain specific information types.

- **Reference information types**
  The generic and domain-specific information types are merged together by reference information types. Taking the car example used above, the specific information types of the car are taken, together with the generic information type of a product model; a reference information type is used to merge these two together, in order to define the complete car product model.
In Figure 4 this distinction is illustrated. The reference information type product model info is used to import the generic information type product model and the domain specific information types ordination product info, physical product info, functional product info, and presentational product info.

![Figure 4: Generic, domain-specific and reference information types](image)

Within DESIRE, this form of knowledge composition is specified as shown in Figure 5. The reference information type on the left hand side is composed of (a) the generic information type depicted at the upper right hand side, and (b) the domain-specific information types depicted at the lower right hand side.

![Figure 5: Composition of information types defined for the basic product model](image)

The generic information type product model is specified as shown in Figure 6. Based on this ontology, a product is represented in the following (textual) form:

```plaintext
physical_model_includes(ordination(manufacturer("Renault")),
    physical(brake_system("ABS", standard)));
physical_model_includes(ordination(manufacturer("Renault"), model("Espace")),
    physical(bumpers(side, standard)));
functional_model_includes(ordination(manufacturer("Renault"), model("Espace"), type("RT2-0")),
    functional(active_safety(8.5)));
```
As noted before, the ordination model is used to identify a product. During the definition and specification of a product, a combination of facts and rules are applied to complete the specification and derive more information about a given product.

![Diagram of the generic information type product model]

Figure 6: The generic information type product model

The main advantage of the product model originates from the distinction that has been made between various aspects of the product model. This classification helps the experts in each category to be able to describe and specify their knowledge of a product without being overwhelmed by information from other categories. For instance, an expert in the presentational model declares and specifies the video or audio clip of a specific product without having to specify or know the physical model specification completely.

Rules and facts are used in order to describe the relationship among different aspects of a product. Rules are applied on the product ontology model in order to evaluate a product and derive new facts about a specific product. For example, physical information about the presence of airbags, bumpers, and safety belts can be used to derive the safety quality of the car (functional model). This will be addressed in more detail in Section 5.

3.4 Consumer and Provider Perspective Ontologies

After specifying the basic product model, we are now ready to define specific product ontologies from the the consumer and provider perspective. By consumers and providers we do not only mean human users, but any process or action that triggers an offer or demand placement. For example a personal agent, acting on behalf of a human is also considered to be a consumer or provider. In specifying a consumer or provider perspective model, the following criteria are taken into account:

- How does a consumer/provider look at (understand) a product?
- Which aspects of a product can be used in the design of the consumer/provider model?
- Which new elements have to be added to consumer/provider model?

The typical characteristic with many interactive shopping systems and virtual markets is that most of the time the consumer is regarded to know the very specific and physical properties of a product, despite the fact that an average consumer is unfamiliar to these. Questions asked from the consumer mostly refer to the physical properties of the product, rather than being more consumer-friendly, such as quality preferences or aspects of the consumer’s lifestyle or status.

For an average consumer, it would be easier to determine how important the safety and modulation of a car for her/him is, rather than having to specify the physical aspects such as safety belts, bumpers type, lights type, and brakes type. The same principles hold for a provider, offering a product on the market place. In this case, provider is regarded to know more about the physical model of the product, but even in this case many aspects can be derived by the market place itself. For example, specifying the model, type, year, and manufacturer of a car by the provider would be enough for the market place to derive many of the characteristics of the car concerning the physical or functional properties.
In Figure 7 an example is given of a provider perspective model, in Figure 8 a consumer perspective model. According to this representation, a consumer or provider perspective model is defined as a set of attributes originated from 5 classes (groups) of attributes. The user model is used to identify the consumer or provider. The consumer or provider specific attributes are used to represent attributes such as desired price range and negotiation strategy parameters. The other 3 groups specify the preferences or specifications of the product requested or offered by the consumer or provider. Looking at the consumer and provider perspective models illustrated in Figures 7 and 8, the following sub-models are identified:

- **Product related model**
  A consumer uses three aspects of the product model, e.g. the Functional, Physical, and Ordination model in order to specify his/her preferences. From these three sub-models, the functional model is the preliminary one used to specify most of the preferences. Consumers who want to specify the more specific characteristics of the product use the physical and ordination model. These consumers are either more familiar with the product domain or are looking for products with very specific characteristics. Adding these two models into the consumer model provides the opportunity to specify product preferences on different levels, depending on how good the product domain is known by a consumer. In this way, both novice and advanced users are taken into consideration. A provider, on the other hand, does not have to specify the functional model. This model is derived from the other models (Physical, Ordination) by the market place. The presentational model is specified by the provider, or by a visual or animation expert.

- **User model**
  A user model is used in order to identify the consumer or provider interacting with the market place. Name, Password, and ID-number are some examples of concepts defined in this model.

- **Consumer or Provider specific attributes**
  So far we have seen how a product model is used to specify product-related concepts and a user model to specify the users interacting with the virtual market place. However, some attributes exists which, can not be classified within the user or product model and still must be defined and
specified. An attribute such as price is one. For the consumer this is the price that he/she is prepared to pay for a product. For a provider, price indicates the amount that the provider is asking for his/her product offered. A new model has been proposed where consumer/provider related attributes are defined. For now, e.g., the current phase of the project, the price is the only one specified within this model. Other concepts such as negotiation strategy parameter, or payment method can be added and specified in this model, in future.

Figure 8: A consumer perspective product model

4. EVALUATION AND MATCHING KNOWLEDGE

Together with the product related models, the corresponding evaluation and matching knowledge is described in this section. The product category Cars will be used as an illustration. Examples are given of how a product model for a car is specified within the market place and how the corresponding evaluation and matching knowledge is applied in order to derive (1) more information about the product and (2) matching of demands and offers. Due to the limited space of this article only a part of the related models is specified.

Evaluation of product related models (basic product model, provider perspective product model, consumer perspective product model).

A product or provider perspective model is evaluated based on product specific dependencies and a specific ontology of evaluation criteria that is defined.

Matching Demands and Offers.

Demands and Offers about a product, originated from customers, are compared with each other in order to find suitable matches, which satisfy both the consumers and the providers within the virtual market. The degree of mismatch is evaluated in order to select the best set of results.

In Section 4.1 the necessary parts of the models on which the evaluation is based are presented: the product model for cars, the model that represents the provider perspective, and the model that represents the perspective of the consumer. In Section 4.2 it is first explained how cars in general can be evaluated. Based upon the general evaluation method for the product, a specific evaluation method for the product as offered by a provider is explained. Before a match can be found between offers and demands, also the demand needs to be evaluated. The evaluation of a demand is based
upon the consumer model. The matching of the evaluations of offers and demands is explained in Section 4.3.

4.1 Example Models for the Car Market

Evaluation of offers and demands is based on a number of models: the product model for cars, the model that represents the provider perspective, and the model that represents the perspective of the consumer. To illustrate the evaluation and matching knowledge, the following example models are used.

4.1.1 Example product model: Car

Within the market place, the product model of a car is specified as illustrated in Figure 9. Note that this specification is based on the product model described in Section 3.

![Figure 9: Car product model (partially represented)](image)

The model in Figure 9 only contains a generic ontology for cars. For specific cars instantiated facts are necessary. For example, for the models Espace, type RT2.0 a (partial) list of such an instantiation is given by the following:

```plaintext
physical_model_includes(ordination(manufacturer("Renault"), model("Espace"), type("RT2.0")),
  physical(brakes_system("ABS")))
physical_model_includes(ordination(manufacturer("Renault"), model("Espace"), type("RT2.0")),
  physical(brakes_detail(front, Disk, standard)))
physical_model_includes(ordination(manufacturer("Renault"), model("Espace"), type("RT2.0")),
  physical(brakes_detail(back, Disk, standard)))
physical_model_includes(ordination(manufacturer("Renault"), model("Espace"), type("RT2.0")),
  physical(bumpers(front, standard)))
physical_model_includes(ordination(manufacturer("Renault"), model("Espace"), type("RT2.0")),
  physical(bumpers(back, standard)))
physical_model_includes(ordination(manufacturer("Renault"), model("Espace"), type("RT2.0")),
  physical(airbags(driver, standard)))
physical_model_includes(ordination(manufacturer("Renault"), model("Espace"), type("RT2.0")),
  physical(airbags(passenger, standard)))
physical_model_includes(ordination(manufacturer("Renault"), model("Espace"), type("RT2.0")),
  physical(airbags(sides, optional)))
```

A set of rules is defined on the car product model that uses the available set of facts in order to derive more information about this category of products. These rules are left out in this presentation.

4.1.2 Example provider perspective model

Aside from the generic information about specific manufacturers, information is also necessary that discloses which provider can deliver which specific car. The provider perspective model is used to represent this information. For example, a provider within the Car product model has given the following specifications of a car offered by her:
provider_ordination_model_includes(Anna, car, ordination(manufacturer("Renault")));  
provider_ordination_model_includes(Anna, car, ordination(model("Espace")));  
provider_ordination_model_includes(Anna, car, ordination(type("RT2.0")));  
provider_ordination_model_includes(Anna, car, ordination(year("1997")));  

A provider specifying his/her offered product is allowed to provide information about the ordination, physical, and presentation aspect of the offered product. In Figure 10 this hierarchy is depicted.

![Figure 10: Car provider perspective model (partially represented)](image)

**4.1.3 Example consumer perspective model**

A consumer looking for a suitable car can, e.g., specify the following as his preferences of a car:

consumer_functional_model_includes(John, car, functional(active_safety(9.0)), 1);  
consumer_functional_model_includes(John, car, functional(passive_safety(8.0)), 1);  
consumer_functional_model_includes(John, car, functional(theft_safety(8.0)), 4);  
consumer_functional_model_includes(John, car, functional(comfort(9.0)), 2);  
consumer_functional_model_includes(John, car, functional(modulation(8.5)), 3);  

The consumer is allowed to specify the product attributes within the categories of Functional, Physical, and Ordination model. The last two categories are meant for the more experienced consumers who have more specific preferences about their desired car. Figure 11 gives an illustration of the consumer model within the product category Cars.

![Figure 11: Car consumer perspective model (partially represented)](image)

**4.2 Evaluation Knowledge**

In this section, it is first explained how cars in general can be evaluated. Based upon the general evaluation method for the product, a specific evaluation method for the product as offered by a provider is explained. Before a match can be found between offers and demands, also the demand needs to be evaluated. The evaluation of a demand is based upon the consumer model. By way of an example it is shown which evaluation processes are performed and what type of knowledge is specified for these processes.
4.2.1 Evaluation: product model, generic perspective

A weight-value structure is built on top of the product model: nodes represent product attributes and are assigned values, arcs between the nodes represent the dependency between these attributes (nodes) and are assigned a weight-value. Using arcs, physical elements of the model are related to functional elements of the model. Two types of evaluations are applied, one evaluating the leaf nodes, and another evaluating the adjacent higher nodes. Evaluation makes use of knowledge expressed in weight-values specifying the degree of importance between different attributes within the product model.

For the car market example Figure 12 presents an illustration of how attributes of the product model depend on each other. The values assigned to nodes and arcs are used during the evaluation and matching process in order to calculate new values.

Note that in Figure 12, the left hand side of the structure concerns only functional elements of the product model, whereas the right hand side concerns only physical aspects of the product model. From left to right three levels can be discerned in which the arcs are assigned a weight value according to:

**Level 1: Functional -> Functional**

has_weight_value(safety, active_safety, 0.4);  
has_weight_value(safety, passive_safety, 0.5);  
has_weight_value(safety, theft_safety, 0.1);

**Level 2: Functional -> Physical**

has_weight_value(active_safety, brakes, 0.3);  
has_weight_value(active_safety, bumpers, 0.2);  
has_weight_value(active_safety, chassis, 0.2);  
has_weight_value(active_safety, airbags, 0.3);  
has_weight_value(passive_safety, safetybelts, 0.5);  
has_weight_value(passive_safety, headrests, 0.5);

**Level 3: Physical -> Physical**

has_weight_value(brake, brakes_system, 0.4);  
has_weight_value(brake, brakes_front, 0.3);  
has_weight_value(brake, brakes_back, 0.3);  

Figure 12: Car evaluation hierarchy
For our example the evaluation knowledge as presented by the level 3 arcs is shown in Figure 13.

**Product Evaluation: Car**  
*Renault, Espace, RT2.0, 1997*

Evaluation [brakes] = 10.0 * 0.6 + 9.0 * 0.2 + 9.0 * 0.2 = 6 + 1.8 + 1.8 = **9.6**
Evaluation [bumpers] = 10.0 * 0.4 + 10.0 * 0.3 + 6.0 * 0.3 = 4 + 3 + 1.8 = **8.8**
Evaluation [airbags] = 10.0 * 0.4 + 10.0 * 0.4 + 7.0 * 0.2 = 4 + 4 + 1.4 = **9.4**

**Figure 13: Car product evaluation results**

The complete evaluation method is then used to evaluate the offers of the providers.

### 4.2.2 Evaluation: provider perspective model

After evaluating the physical aspects of an offer using the *weight-values* at level 3, the weight-values at level 1 and level 2 are applied in order to derive more information about the offer. Especially, the functional aspects of the car are of interest for the matching with the demands. Figure 14 gives an overview of how the evaluations at this model take place.

**Figure 14: Car provider evaluations**
The evaluation results are as follows:

Evaluation [active_safety] = 9.6 * 0.3 + 8.8 * 0.2 + 9.1 * 0.2 + 9.4 * 0.3 = 9.28

In the same way, evaluation results for other aspects are derived.

Evaluation [passive_safety] = 8.0
Evaluation [theft_safety] = 7.5
Evaluation [safety] = 9.28 * 0.4 + 8.0 * 0.3 + 7.5 * 0.3 = 8.362

The knowledge for the two types of evaluations is specified as follows:

if physical_model_includes(O:ORDINATION_MODEL, physical(brake_system("ABS")))
then product_evaluation(car, O:ORDINATION, brake_system, 10.0);

For other types of brake systems similar rule is applied, and depending to the quality a value is assigned to the corresponding brake system type.

if physical_model_includes(O:ORDINATION_MODEL, physical(brake(front, "Disk", standard)))
then product_evaluation(car, O:ORDINATION, brake_front, 9.0);

if physical_model_includes(O:ORDINATION_MODEL, physical(brake(back, "Disk", standard)))
then product_evaluation(car, O:ORDINATION, brake_back, 9.0);

if product_evaluation(car, O:ORDINATION, brake_system, V1:VALUE)
and product_evaluation(car, O:ORDINATION, brake_front, V2:VALUE)
and product_evaluation(car, O:ORDINATION, brake_back, V3:VALUE)
and has_weight_value(brake, brake_system, W1:WEIGHT)
and has_weight_value(brake, brake_front, W2:WEIGHT)
and has_weight_value(brake, brake_back, W3:WEIGHT)
then product_evaluation(car, O:ORDINATION, brake, V1*W1 + V2*W2 + V3*W3);

4.2.3 Evaluation: consumer perspective model

Using the specified demand preferences and the priorities of these preferences by the consumer, an evaluation is performed on the consumer model in order to determine the degree of mismatch, e.g. the minimum and maximum values allowed by the consumer. The following specifications are obtained from our example:

Active_safety = 9.0 Priority = 1 -> mismatch value = 9.0 * 1 : 100 = 0.09
Passive_safety = 8.0 Priority = 1 -> mismatch value = 8.0 * 1 : 100 = 0.08
Theft_safety = 8.0 Priority = 4 -> mismatch value = 8.0 * 4 : 100 = 0.32

The minimum and maximum values are calculated as follows:

Active_safety min_value = 9.0 – 0.09 = 8.91 max_value = 9.0 + 0.09 = 9.09
Passive_safety min_value = 8.0 – 0.08 = 7.92 max_value = 8.0 + 0.08 = 8.08
Theft_safety min_value = 8.0 – 0.32 = 7.68 max_value = 8.0 + 0.32 = 8.32

Before storing these values a check is performed on the calculated values in order to prevent violating the allowed minimum and maximum, [0…10].

4.3 Matching Knowledge

Matching is performed during two phases. During the first phase, those offers are selected that fall within the min & max range of the corresponding demand specifications. Applied rules in our example look like the following:
if demand_range(ID_demand, car, PRODUCT_ATTRIBUTES, Min, Max) 
and offer_evaluation(car, ID_offer, PRODUCT_ATTRIBUTES, E_Value) 
and E_Value > Min 
and E_Value < Max 
then primary_match(ID_demand, ID_offer, car, PRODUCT_ATTRIBUTES);

Following our example specifications, the following results are obtained:

demand_range(ID976, car, passive_safety, 7.92, 8.08) 
offer_evaluation(car, ID965, passive_safety, 8.0)

This results in:

primary_match(ID976, ID965, car, passive_safety)

The other two preferences, active_safety and theft_safety are not within the range of corresponding min and max, and therefore are not derived.

In the second phase of matching process the derived set of primary matches is filtered to a best (final) set of matches, based on the number of attributes of an offer that falls within the range of the given demand preferences. The offer with the most number of attribute matches is selected. In our example, there was only one match and this is selected in the final set as well. After this phase, the degree of mismatch between the found offers and the demand is computed and represented to the consumer.

if primary_match(ID_demand, ID_offer, car, passive_safety) 
and offer_evaluation(car, ID_offer, passive_safety, E_Value) 
and demand_range(ID_demand, car, passive_safety, Min, Max) 
then final_match(ID_demand, ID_offer, passive_safety, abs[((Min+Max):2):E_Value) – 100]);

We get the following results:

demand_range(ID976, car, passive_safety, 7.92, 8.08) 
offer_evaluation(car, ID965, passive_safety, 8.0); 
primary_match(ID976, ID965, car, passive_safety);

This results in:

final_match(ID976, ID965, car, passive_safety, %0.0);

The above statement means that this particular offer shows around 0 % mismatch with the current demand with respect to the passive_safety attribute. For other attributes active and theft safety the degree of mismatch is computed as well. The active_safety of this offer (9.28), for instance, shows 3.017 % mismatch with the current demand.

5. RELATED WORK

In this section, some of the recently developed and operational models of virtual market places and Web commerce based applications are briefly mentioned.

- **Kasbah**
  Kasbah (cf. [3], [4]) is a web-based multi-agent system using agents interacting with each other within the virtual market domain. The agents act on behalf of their users [3]. Price Negotiation is one of the interesting features applied within Kasbah [4].

- **Market Space**
  Market Space is an open agent-based market infrastructure. It is based on a decentralized infrastructure model in which both the humans and the machines can read information about the
products and services, and everyone is able to announce interests to one another [5]. The aim in
designing Market Space is to design a market place where searching, negotiation and deal
settlement, e.g. interaction with users is done using agents. The AMP (Agent Marketplace
Project) is a collaboration project between Uppsala University and Swedish telecom, Telia.
Market Space has been developed mainly in Prolog. For the communication with the Web, the
standard protocol (HTTP) has been used.

A difference with our approach is that these approaches have been implemented without using a
principled design method, and do not use components as building blocks that are (formally)
specified at a conceptual level. This is also a difference with the work described in [9].

6. DISCUSSION

The prototype system for an agent-based virtual market support system presented in this paper has
been designed and implemented using the compositional development method for multi-agent
systems DESIRE and its software environment. This development method combines agent
technology and knowledge technology. Both ingredients have been used to achieve the following
combination of characteristics:

- The system is a distributed system, hosting a number of autonomous actors which operate as
agents in interaction with each other.

- The agents within the system show different types of behaviour, not only reactive
manner with respect to both individual buyers and sellers, but it also shows pro-active and
adaptive behaviour.

- The system is highly knowledge-intensive. Buyers and sellers are supported by different
ontologies to express themselves, when they formulate properties (product models) of or
demands for products. Besides these ontologies, also domain-specific evaluation and matching
knowledge relating the different ontologies has been included in the system.

- The system is easy to maintain. The system has a transparent compositional structure based on
a generic broker agent model and is flexible for maintenance in changing circumstances.

Our focus on the use of generic agent models, knowledge representation and ontology design is the
most distinctive feature to be named comparing with other systems some of which were mentioned
in the previous section. The compositional structure applied within the product representation and
the architecture of the agents and agent components make it easier to understand, adjust, or expand
the market place.

Within the software company CRISP (CReative Internet Solution Partners), the prototype
system described here is being used as a basis for the further development of a distributed, agent-
based software environment supporting electronic markets. To enable interaction with other virtual
markets, a meta-mediating agent can be used to maintain the communication with the other virtual
markets, the partners with whom our virtual market has a commitment to cooperate with. Such
interaction can take place for example when no match is found within the virtual market, which
could satisfy the requirements specified by the user. In that case a new process can start where the
search is forwarded to virtual market partners, using the meta-mediating agent.

REFERENCES

International Conference on Multi-Agent Systems, ICMAS’95, MIT Press, Cambridge, MA,


