Part IV.

KnowMarket

7. IT Infrastructure for Electronic Knowledge Markets

Knowledge markets have to be well integrated into the daily work. To achieve this we propose an XML Web Service architecture, so that the knowledge market can be easily integrated into other programs like office applications or portals. We have developed *KnowMarket* as a proof-of–concept prototype for our knowledge trading model. *KnowMarket* is a prototype of an electronic knowledge market that has an XML Web Service centered architecture. The market trades online expert advice. The following is intended as an overview of the electronic knowledge market *KnowMarket*.

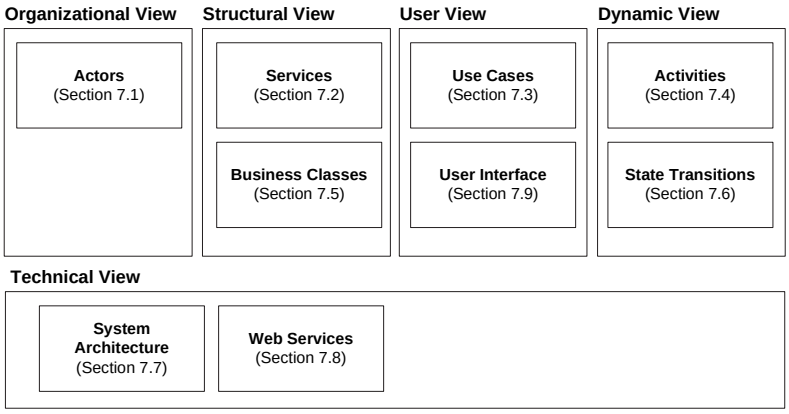

Figure 7.1.: Different Views of the IT Infrastructure and their corresponding UML Elements

We use UML for specifying the different aspects of *KnowMarket* (cf. Appendix A). In doing this we shall take a look at the main actors involved in the knowledge market, the services of the knowledge market, important use cases, the activities, the business classes, the state transitions, the system architecture and the user interface. In literature there are different proposed views to systematize the UML constructs (cf. [79,

p. 33],[146, p. 23]). However, there is no commonly accepted systematization of the different UML constructs. We propose the following views: the organizational view, the structural view, the user view, the dynamic view, and the technical view. Figure 7.1 shows the different views and their corresponding UML elements.

7.1. Actors

We shall look at two basic users in the electronic knowledge market: experts and advice seekers. Experts are users who possess specialized knowledge about one or more knowledge areas and make their knowledge available on the knowledge market. Advice seekers are users who request special knowledge from experts on the knowledge market in order to solve a problem. As a third actor the marketplace operator is also involved as an administrator and auctioneer. Figure 7.2 shows the different actors whereby the hierarchy reflects the inheritance structure of the actors.

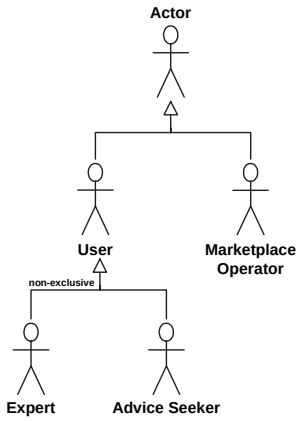

Figure 7.2.: Actors

7.2. Services

The most important services are displayed in groups according to the different transaction phases (see Figure 7.3). These phases include the registration of expert knowledge (hereafter known as expertise) and the request, the search for experts, carrying out the search and registration on the basis of a knowledge taxonomy, the management of time and contact profiles, provision of authentication methods on the knowl-

edge market, and the acceptance of certificates and reports as proof of expert knowledge.

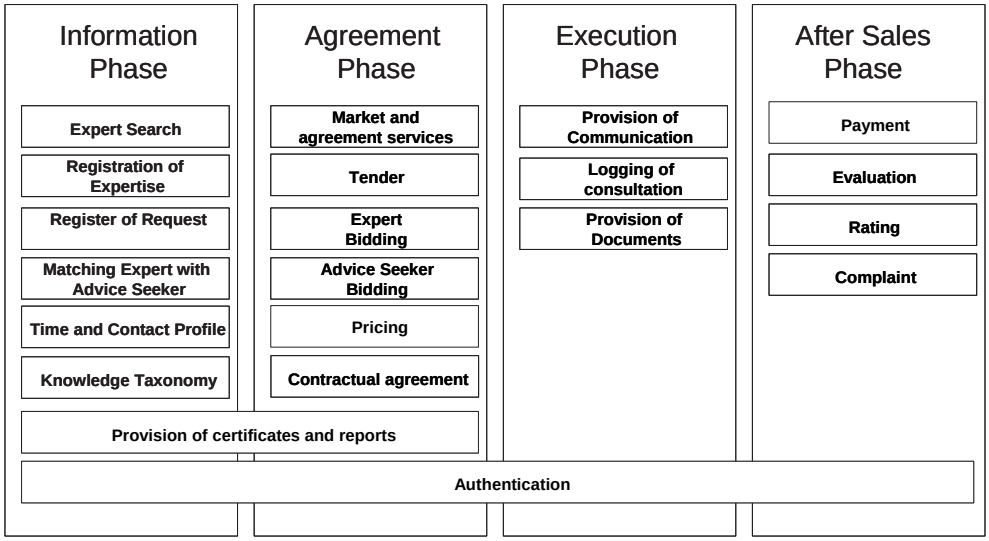

Figure 7.3.: Transaction Phases and Services in KnowMarket

Services that regulate the bargaining between the actors are grouped together in the agreement phase. This includes negotiation of transaction conditions, especially finding a price for the expert knowledge to be provided. The agreement phase ends with a legally binding contract. In the execution phase that follows, expert advice is communicated to the advice seeker. Online consultation between actors may be logged in order that this knowledge may be made available again at a later date. In the after sales phase, payment of the agreed fee is made and the actors on both sides are evaluated.

It needs to be possible to call up each of these services using Web Service interfaces in order to facilitate integration in heterogeneous IT environments and enable services of *KnowMarket* to be used independently of the platform.

7.3. Use Cases

The most important interactions between actors and the knowledge market are depicted using UML use case diagrams. All shown use cases are displayed in groups in line with the transaction phases mentioned in Section 5.2.

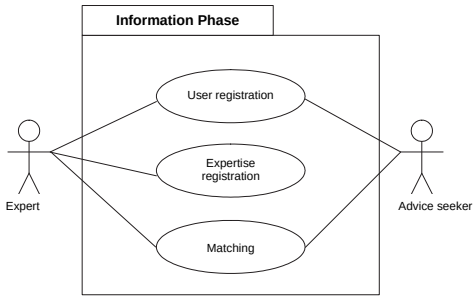

Figure 7.4.: Use Case Diagram for the Information Phase

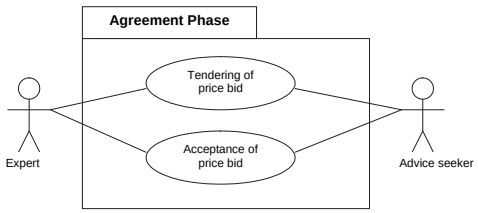

Figure 7.5.: Use Case Diagram for the Agreement Phase

Three important use cases concerning the information phase can be identified: user registration, expertise registration, and matching. The users first have to register the basic user information like name, email address, etc. If the user is an expert wanting to trade their expertise, further information is required about his expertise. In order for an advice seeker to make contact with appropriate experts, he needs to formulate his problem for the knowledge market. The electronic knowledge market will then match experts in relation to the request with whom the conditions of the knowledge transfer can be negotiated. This also signals the start of the agreement phase.

The agreement phase (see Figure 7.5) essentially consists of two important use cases: tendering of price bids and acceptance of price bids. Advice seeker and expert exchange price bids in a negotiation process in order to agree upon a price that is

satisfying for both parties. Once both actors accept the price bid of another, the agreement phase ends.

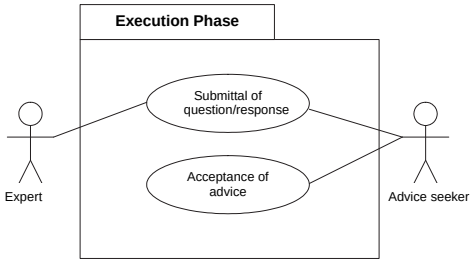

Figure 7.6.: Use Case Diagram for the Execution Phase

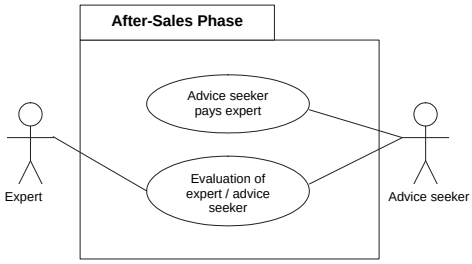

Figure 7.7.: Use Case Diagram for the After Sales Phase

The execution phase (see Figure 7.6) consists of two main use cases: the submittal of questions and responses and the acceptance of advice. In this phase, the expert advice is communicated to the advice seeker. This results in an interchange of questions and responses between the two actors. It is during this communication process between the advice seeker and the expert that an attempt is made to solve the question of the advice seeker. At some point, the problem of the advice seeker will be solved as a result of one or more answers from the expert and the advice seeker will hopefully consider his question as having been answered. The acceptance of the advice by the advice seeker signals the end of the execution phase.

In the final after-sales phase (see Figure 7.7), payment of the agreed fee is made to the expert and evaluation of the interacting parties is made by both sides based on predetermined rating criteria. This rating is stored further in a reputation system (see Section 5.10 for the analysis of different quality assurance mechanisms).

7.4. Activities

We further specify the information, agreement, execution, and after-sales phase with activity diagrams in Figures 7.8, 7.9, 7.10, and 7.11. See Section 5.2 for a description of the different transaction phases.

Figure 7.8.: Activity Diagram for the Information Phase

In the information phase the advice seeker begins by formulating a question (see Figure 7.8). The system generates a list of competencies that could be related to the question. This is done by decomposing the question text in separate words and querying the competency taxonomy with these keywords. The list of relevant competencies is then presented to the advice seeker who can add and delete competencies. Then weightings for each competency in the list are defined by the advice seeker. In the next step the system matches the right expert according to the competencies required. This is done using a Multi-Criteria Decision Making Method, namely TOPSIS, that is described in Section 5.8. If there is an appropriate expert, the advice seeker selects

one or more experts for further negotiation.

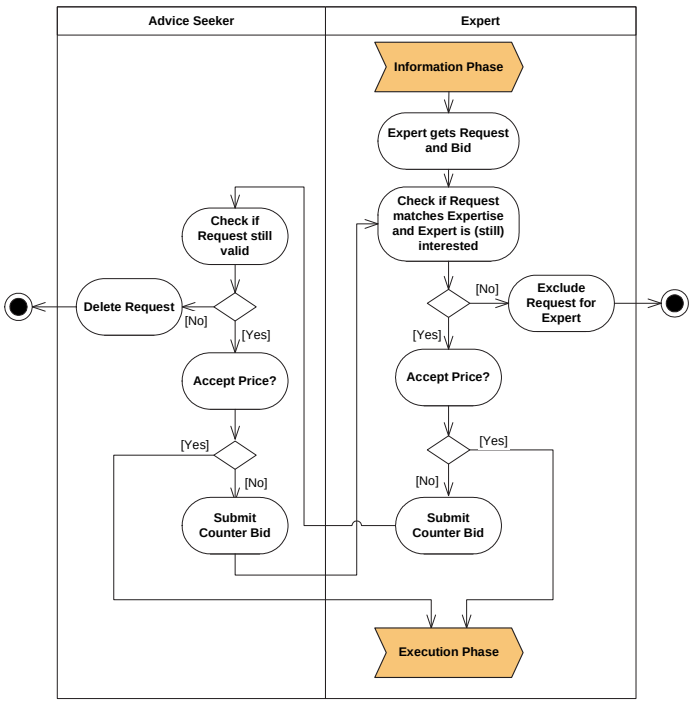

Figure 7.9.: Activity Diagram for the Agreement Phase

In the agreement phase (see Figure 7.9) the expert and the advice seeker negotiate the price for the online expert advice. This is done by mutually bidding for a request, respectively, an answer. This bidding and counter-bidding will continue as long as one will accept the other bid or drop out of the negotiation process. However, negotiation can take place with several possible experts at the same time, as is sketched in a state transition diagram in Section 7.6.

In the execution phase (see Figure 7.10) the advice seeker formulates a question and the expert answers the question. The advice seeker has the possibility to further ask clarifying questions until the answer solves the problem or the number of questions exceeds the contract.

In the after sales phase (see Figure 7.11) the advice seeker pays the expert, and both rate each other.

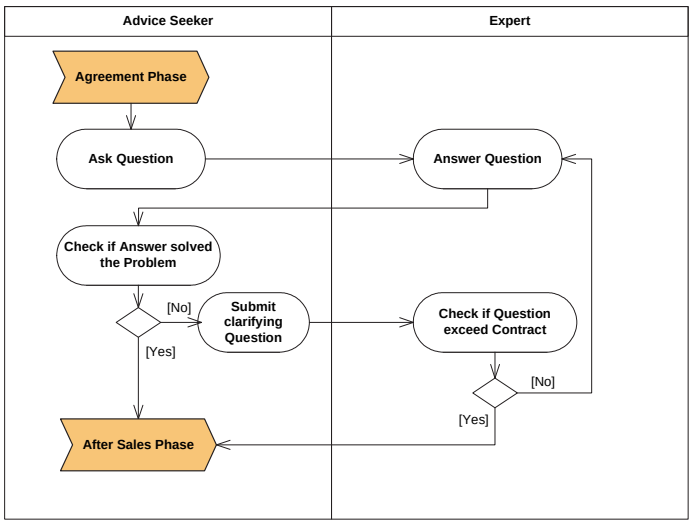

Figure 7.10.: Activity Diagram for the Execution Phase

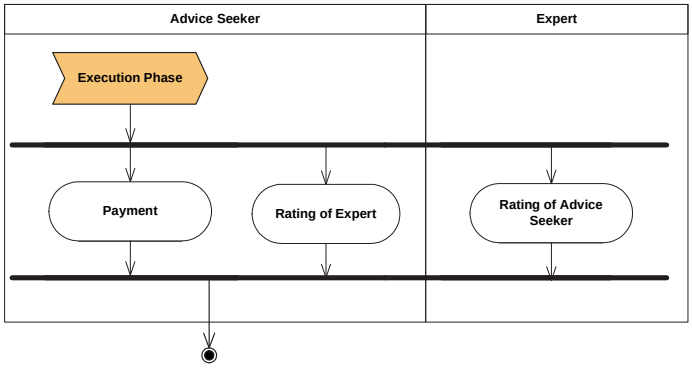

Figure 7.11.: Activity Diagram for the After Sales Phase

7.5. Business Classes

The business classes of the electronic knowledge market are depicted in Figure 7.12.

The class *Expert* and the class *Advice Seeker* are inherited from the class *Users* (see also Section 7.1).

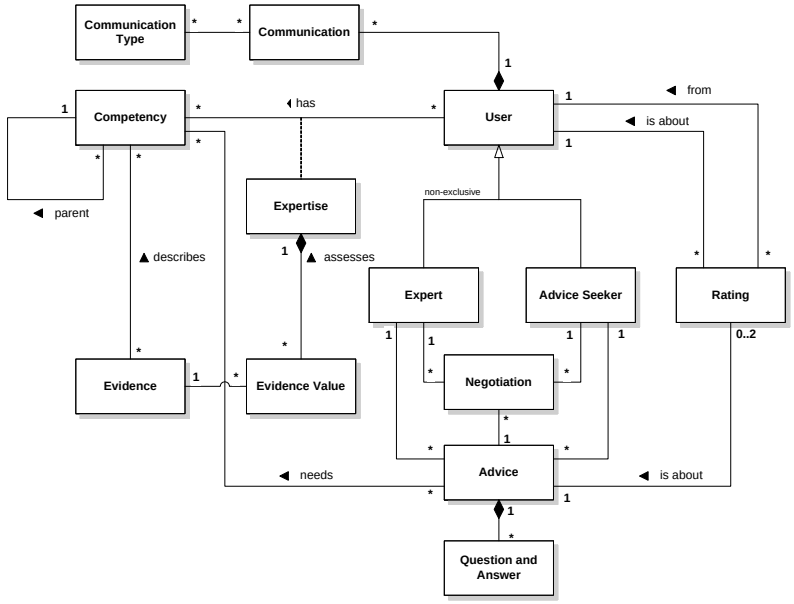

Figure 7.12.: Business Classes of an electronic knowledge market

Each user must be able to communicate by means of several communication channels which are modelled in the class *Communication Type*. This ensures that all possible forms of communication may be supported in the future. Communication in *KnowMarket* uses a web front-end. It would also be possible to transmit questions and responses using a chat system such as MSN Instant Messenger. The individual properties of the communication type for a particular user—like the e-mail address—are represented in the class *Communication*. The relationship between the classes *User* and *Communication* is represented as a composition, since existence of the class *Communication* is dependent on the class of *User*.

The knowledge domains are stored in the *Competency* class. This competency is organized in a hierarchy which forms a taxonomy. The hierarchy is represented as a parent-child relationship. An example of the hierarchical representation for the knowledge domain IT may appear as follows: *IT* > *Databases* > *Relational Data*

Bases > *Oracle*. The expertise of a user is a set of competencies with corresponding scores for their proficiency. One or more assessments or rating criteria are assigned to each competency using the *Evidence* class. This includes criteria such as personal valuation or proof of expertise in a particular field of knowledge. In the *Evidence Value* class, a value is assigned to an item of evidence and to a particular competency. Evidences might be "Personal Valuation" and "Certificate" for the competency "Oracle." Possible evidence values for these evidences are shown in Table 7.1.

Competency	Evidence	Evidence Value
Oracle	Personal valuation	Advanced
Oracle	Certificate	Grade: Good

Table 7.1.: Example Competency with possible Evidences

Evidence values may then be converted internally to cardinal values for matching purposes. A value *Advanced* might, on a scale of 0 to 1, be assigned to 0.5. A beginner might be 0.1 and an expert 0.9. The allocation of grades is even more intuitive and might be based on a grading system. Other evidences might be e.g. the total training sessions for one competency.

The *Evidence Value* is an individual skill level and must therefore refer to a person. The reference is made with the aid of the relation class *Expertise*. All competencies about which a user possesses knowledge are modelled using the *Expertise* class. A user may possess knowledge about several competencies. Conversely, several users may possess one competency of course.

An advice seeker's question is stored in the *Advice* class. An instance of this class relates to a particular question and answer sequence. It also enables a single advice to be linked to various competency fields. The class *Advice* is also the reference point for price negotiation between experts and the advice seeker (class *Negotiation*). The entire communication process—the actual question-answer sequence between experts and advice seekers is stored in instances of the *Question and Answer* class. This communication process needs not necessarily take place in a strict interchange of question and answer. One of the market partners could also detail their questions/responses in greater detail in an additional text.

Once a question, and therefore an *Advice* has been answered satisfactorily, payment to the expert and evaluation of the performance follow. No separate class is provided for this purpose as payment is not examined in any detail. An instance of the class *Rating* refers to a particular *Advice*. Both transaction parties—the advice seeker and the expert—may rate each other. The advice seeker evaluates the expert according to how good the given answer was, for example, or how friendly the ad-

vice was. The expert might also have the option of evaluating the advice seeker. A criterion such as punctuality of payment could be used.

7.6. State Transitions in the Agreement Phase

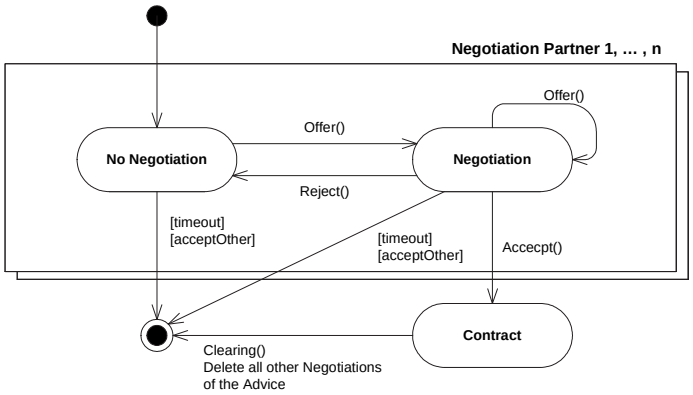

Figure 7.13.: State Transition Diagram for Negotiation

KnowMarket uses an indirect electronic negotiation between advice seekers and experts for the agreement phase. Figure 7.13 shows the negotiation process using a state transition diagram for a negotiation object which is an instance of the *Negotiation* class (compare Section 7.5). Once the advice seeker has chosen particular experts according to the matches, the advice seeker can make an offer and enter the negotiation state. The advice seeker can also wait for the first offer from the expert however. The price can then be determined more precisely by counter-offers or restricted by an accompanying text. This interchange of offer and counter-offer continues until one negotiating party has accepted the offer of the opposite party and thus laid the basis for the contract. If an offer is accepted, all other negotiation objets for this advice have to be deleted.

7.7. System Architecture

The *KnowMarket* prototype is developed with the Microsoft .NET Framework [187, 13]. We shall now present the system architecture of *KnowMarket*. Central to this are the Web Service interfaces. With their help, communication between services

of the knowledge market and client applications of heterogeneous systems is made easier and integration of outside services into heterogeneous system environments becomes possible. Communication with Web Services takes place using standard protocols such as HTTP and SOAP. Mobile client end-devices such as cell phones or PDAs could also access the knowledge market using mobile ASP.NET web sites or rich clients such as Microsoft Word or Microsoft Excel could use Web Services of the knowledge market, thus making use of its functions. By using Web Services of the electronic knowledge market it would thus be possible to locate and communicate with experts about a particular question using Microsoft Word. Using Web Services would also mean other portals or electronic markets could have access to and integrate the functions *KnowMarket* provides.

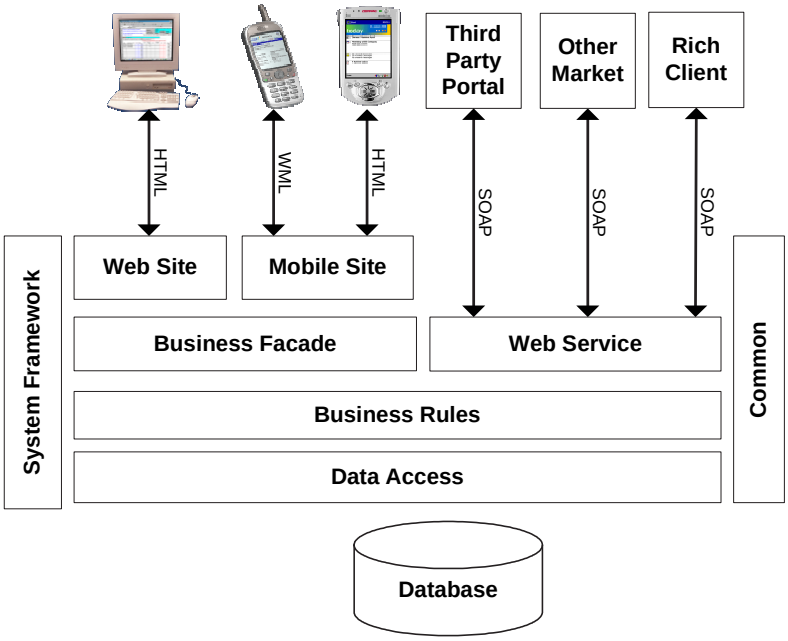

Figure 7.14.: System Architecture of KnowMarket

The *KnowMarket* is realized as a multi-layer architecture with the following layers: *Web*, *Web Service*, *Business Facade*, *Business Rules* and *Data Access* layers. The components *System Framework* and *Common Data* provide general help functions. The *Common Data* layer provides typed DataSets (see [13, p.360]) for the relevant knowledge market functions, while the *System Framework* layer provides functions

for configuration settings, logging and error response.

The system architecture uses a multi-layer approach. The basic concept consists of dividing up the whole system according to functional tasks, data and operating load in order to make maintenance and scaling possible (see Section 7.10). The task of the functions in all layers basically consists of passing on data inputs, after these have been processed if necessary, to functions in the next layer, then returning values to the layer that made the original call. Brief descriptions of the tasks for each layer are as follows:

Web. The web layer's task is to dynamically create HTML web sites for the display on web browsers and to validate client-side data sent from the browser. Other tasks are passing on hints and error messages to the user, transferring data inputs to the *Business Facade* or *Web Service* layer, as well as displaying return values from the *Business Facade* or *Web Service* layer. The web layer is realized with ASP.NET pages [13].

Business Facade. The *Business Facade* layer makes it easier for the client to access the *Business Rules* by abstracting the entire sequence of interactions between the business objects. This layer forms an interface to the underlying business objects and isolates the presentation layer from the business logic. The *Business Facade* accepts input data from the web layer, then forwards this to the *Business Rules* layer. In some cases *Business Facade* will access the *Data Access* layer if a query only requires simple read access to data.

Web Service. The *Web Service* layer implements the public Web Service interface, which can be made accessible for Web Service clients. This layer is similar to the *Business Facade*, but is different in that methods implemented in this layer can be called up by Web Service clients, meaning it is not limited by only being able to be called by other layers of the system. Functions of the knowledge market are called using SOAP requests to Web Service, which then pass back SOAP return values depending on the request.

Business Rules. The *Business Rules* layer implements the application logic. It validates data, carries out calculations and data manipulation. All updates to underlying data storage are carried out by means of the *Data Access* layer. The task of *Business Rules* is to accept calls from the *Business Facade*, process these according to the *Business Rules* using the *Data Access* layer for the purpose of data manipulation, then return the results to the *Business Facade* or *Web Service* layer.

Data Access. The *Data Access* layer functions as a central interface for all data access to the underlying database. The Data Access layer contains functions for querying and manipulation of data. The data manipulation and querying is done by stored procedures. An *Microsoft SQL 2000 Server* is used as the database management system. However, through the layered system architecture a switch to another database can easily be done, because the particularities of the data source are only implemented in this layer.

7.8. The Role of XML Web Services

XML Web Services is central to the architecture of *KnowMarket*. Web Services are encapsulated, self-contained units that make well-defined interfaces available upon request. Kreger [114] defines a Web Service as "an interface that describes a collection of operations that are network accessible through standardized XML messaging."

Using XML Web Services enable the entire expert market or individual system components to be used by other applications. One possible scenario would be integration in a Human Resource (HR) application. It would be possible, for instance, to extend the expertise profile of an expert automatically when a training is booked in the HR application. Also, to run an expert search from within a word processing program—perhaps depending on the document that the advice seeker is currently working on—would be feasible with this architecture.

Five elements are essentially required for communication and use of Web Services: XML, XSD, SOAP, WSDL and UDDI.

XML (Extensible Markup Language). XML is a text-based mark-up language specification of the World Wide Web Consortium (W3C) [208]. In contrast to HTML, which uses tags for simultaneous description of presentation and data, XML is used to define portable, semi-structured data.

XSD (XML Schema Definition). XML Schema [188] can be used to define data exchange structures and message protocols.

SOAP (Simple Object Access Protocol). SOAP is "a standard, extensible, composable framework for packaging and exchanging XML messages" [29]. It enables the exchange of structured and typed data in a distributed environment.

WSDL (Web Service Definition Language). Web Services are described using WSDL [41]. This includes the message and methods offered by the Web Service with their arguments and their return values.

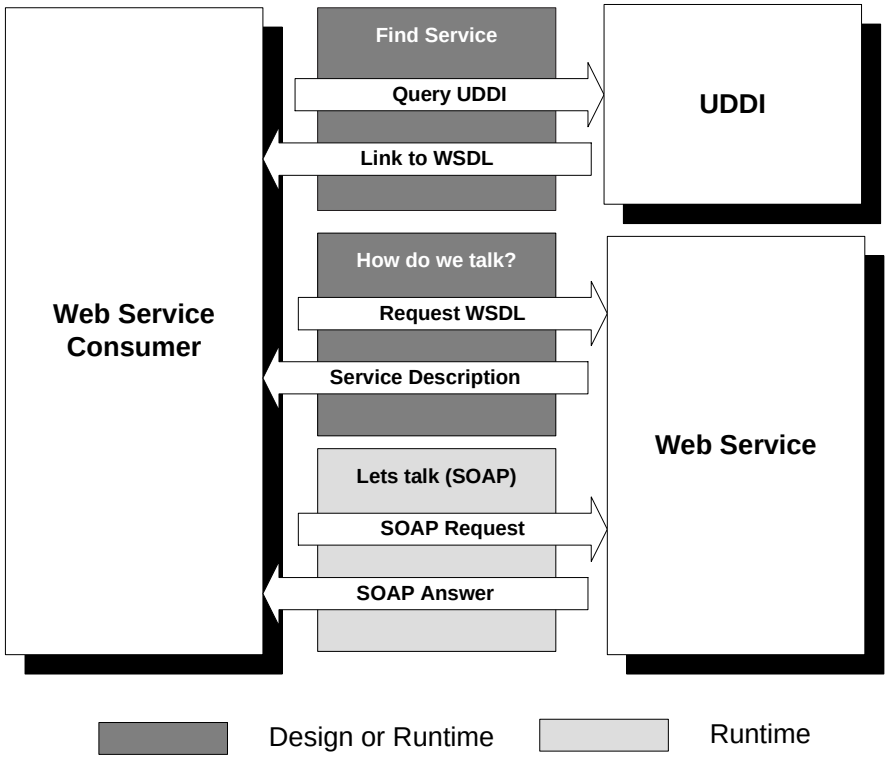

Figure 7.15.: Cooperation between UDDI, WSDL and SOAP (partially adapted from [173])

UDDI (Universal Description, Discovery and Integration). UDDI [23] is a standard for a directory service and is used to locate Web Services on request.

Figure 7.15 visualizes the interaction between UDDI, WSDL, and SOAP. With the help of these elements, Web Services enable applications to make certain functions available for other applications via the web. A particular feature of Web Services is that these functions can be addressed over the Internet by means of the HTTP protocol. Web Services can thus be easily incorporated into existing infrastructures.

Using existing, open Internet protocols and data formats such as HTTP and XML means Web Services are dependent neither on platform nor language, thus distinguishing it from previous component technologies that use component model-specific protocols such as Distributed Component Object Model (DCOM), Remote Method Invocation (RMI), or Internet Inter-ORB Protocol (IIOP) for communication [187].

The use of functions encapsulated by Web Services also requires no detailed knowledge about receiver infrastructure, as Web Services use message-based communication via SOAP, by means of which requests are made to and responses returned from the Web Service; the receiver (external application) needs only to be able to communicate using standard HTTP and SOAP. This response/request architecture means communication using Web Services is stateless. Another property of Web Services is the ubiquity which comes with using standard protocols like HTTP and XML for communication: each client that supports these technologies can both offer and use Web Services. Therefore, also communication via HTTP over WLAN (Wireless Local Area Network) is possible.

By this principle, services of the knowledge market can be made available using Web Service interfaces that enable knowledge trading. Web Services could be used for registering expertise, matching advice seeker requests with offers from experts, price negotiation, communication of questions and answers between actors and rating of actors. Furthermore, the Web Services would allow access to the competency taxonomy of the knowledge market, which would enable registration of expertise to be carried out. The Web Service responsible for accessing the competency taxonomy may return the complete taxonomy structure on request. The return of the taxonomy is packaged as a SOAP message, which can then be further processed as required by third-party applications.

Another example of an electronic market Web Service would be expertise registration. As we have outlined previously, experts wanting to trade their knowledge on the electronic knowledge market need to register exact details about their expertise. To do this they quantify evidence that provides proof of a particular competency (see Business Classses in Section 7.5).

The expertise registration Web Service also accepts messages (requests) from a client that are then processed by Web Services operations, returning a SOAP message (response) to the client. The request to the Web Service also contains, among other things, details of the expertise. As mentioned earlier, other client applications (e.g. for PDAs, cell phones) may also be used for this purpose. The Web Service returns a SOAP message as a response, which then confirms that an expertise registration has been carried out.

Web Services for the other knowledge market functions mentioned follow a similar concept, and enable functions of the knowledge market to be used with external applications regardless of implementation details.

The major Web Services can be grouped together and called *myExpert* and *myExpertise* (see Figure 7.16). *myExpertise* is a set of services for storing the individual expertise and defining the access rights to it. *myExpert* is a set of services for find-

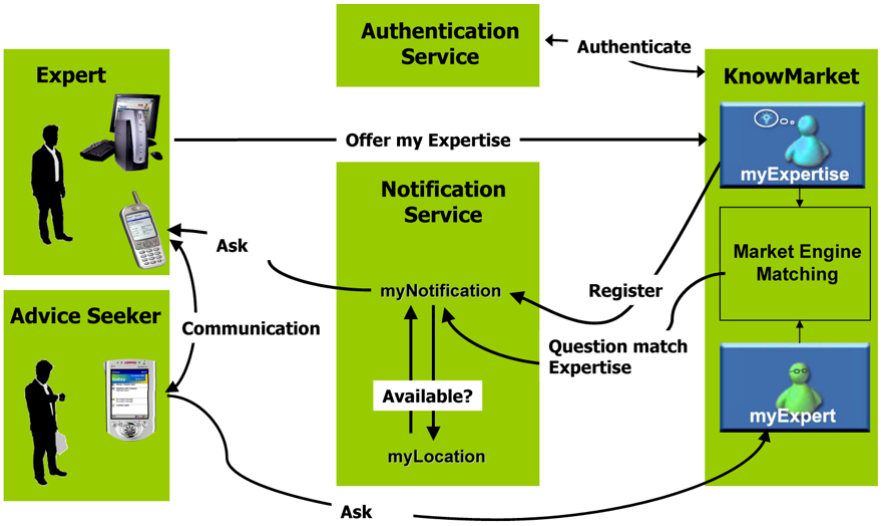

Figure 7.16.: Orchestration between different Web Services in KnowMarket

ing the right expert, asking questions and formulating bids for expert advice. These services can be integrated with other Web Services like Authentication—for example Microsoft Passport [3] or Liberty Alliance [2]—and Notification—for example MSN Messenger.

This Service-Oriented Architecture (SOA) [62] allows the loosely coupling of different knowledge markets inside a company and between companies that form an alliance (see Figure 7.17). Also, the integration of an internal knowledge market with a public knowledge market would be possible. In this federated structure a query for an expertise would first check if an expert is available inside the company. Otherwise the knowledge market of the alliance partner would be asked. Finally the public knowledge market would be queried.

The data model of *KnowMarket* is compatible with HR-XML and therefore HR-XML can be used for part of the Web Service interactions. The HR-XML Consortium [1] is an independent non-profit organization which is concerned with developing a standardized XML vocabulary for the area of Human Resources. It is intended that these standards support inter-organizational information exchange with regards to HR management functions such as recruiting, training etc. A working group from this consortium founded in October 2000, the Competencies Workgroup has developed an XML schema for the exchange of competency information.

The group defines competencies as follows: “A specific, identifiable, definable, and

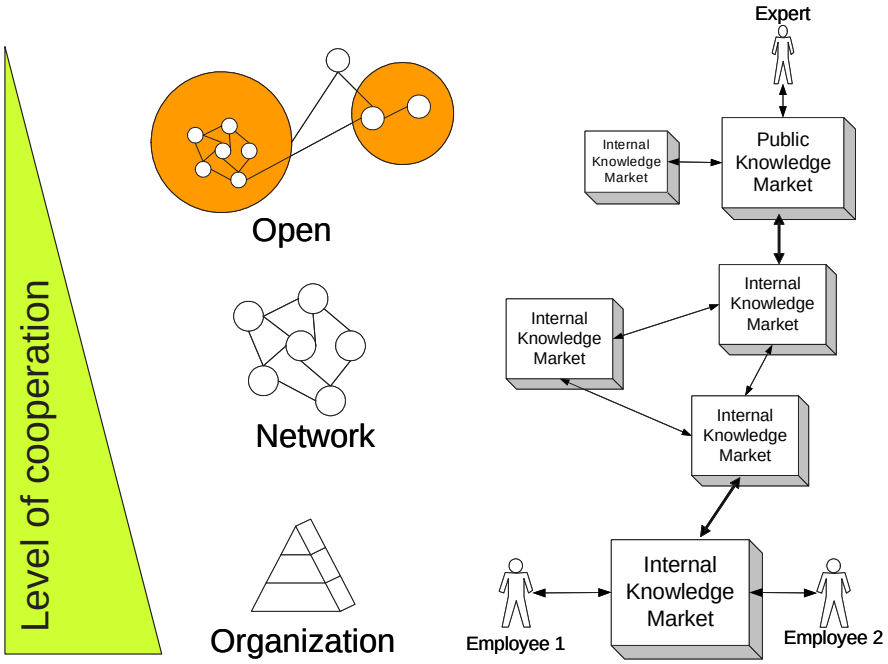

Figure 7.17.: Interaction between internal and public Knowledge Markets

measurable knowledge, skill, ability, and/or other deployment-related characteristic (e.g. attitude, behavior, physical ability) which a human resource may possess and which is necessary for, or material to, the performance of an activity within a specific business context" [12, p. 3].

General characteristics of a human resource are known as skills. Companies use a wide variety of skill taxonomies to categorize these. The HR-XML competencies XML schema is designed to reference these different taxonomies.

"The HR-XML Consortium's competencies schema is designed as a reusable schema fragment that might be applicable to a wide range of business processes. Generally speaking, the schema could potentially be useful in any process involving the comparison, matching, weighting, or rating of a competency demanded against an actual available competency" [12, p. 4]

The most important requirement of the HR-XML competencies schema is the ability to relate taxonomies with differing competency definitions to one another. There are until now no design standards for the development of a competency taxonomy. The generic idea also allows the use of different rating scales for gauging a competency. The following example is intended to clarify the data exchanged using this schema [12].

An employee at the ACME company has knowledge of the JAVA programming language (Competency = JAVA). In a test carried out by the company, the employee obtains a test value of 89. The company's own database shows that he has four years of experience with this programming language and that his level of interest, on a scale of 1-100 points, measures 90. XML representation of these facts is as follows [12, p.20]:

```
<Competency description="Java is an object oriented computer
language" name="Java">
    <CompetencyId description="Competency id is
    based on Acme internal taxonomy" id="574" idOwner="Acme Company"/>
    <TaxonomyId description="My ids are based on Acme Company
    Taxonomy" idOwner="Acme Company" id="1"></TaxonomyId>
    <CompetencyEvidence dateOfIncident="2001-08-23" name="Test Score"
        typeDescription="Test Score from internal test" typeId="54">
        <EvidenceId description="Java Test from internally
        administered test" id="547" idOwner="Acme Company"/>
        <NumericValue description="100 point scale" maxValue="100"
        minValue="0">
            89
        </NumericValue>
    </CompetencyEvidence>
    <CompetencyWeight type="levelOfInterest">
        <NumericValue description="Acme Company Scale 100 point"
        maxValue="100" minValue="0">
            90
        </NumericValue>
    </CompetencyWeight>
</Competency>
```

The use of standard human resource XML schemas makes the reusability of the components of *KnowMarket* much easier.

7.9. User Interface

Dynamic HTML pages are used for the user interfaces (cf. [104, 204] for implementation details). Some web pages from the *KnowMarket* website are shown in the following website structure.

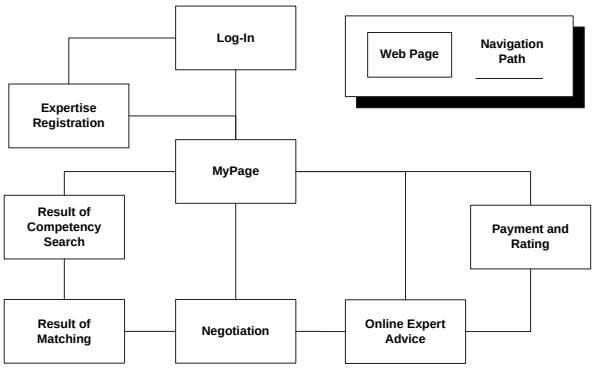

Figure 7.18.: Website Structure

We describe the web pages for expertise registration, matching and price negotiation.

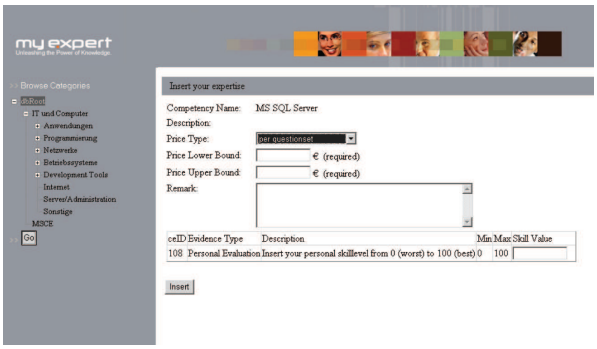

Figure 7.19.: Expertise Registration

The expertise registration allows the expert to insert his expertise according to a knowledge taxonomy (see Figure 7.19).

The *MyPage* site is the navigation center of *KnowMarket* (see Figure 7.20). It provides access to other personalized pages depending on the processing stage of a

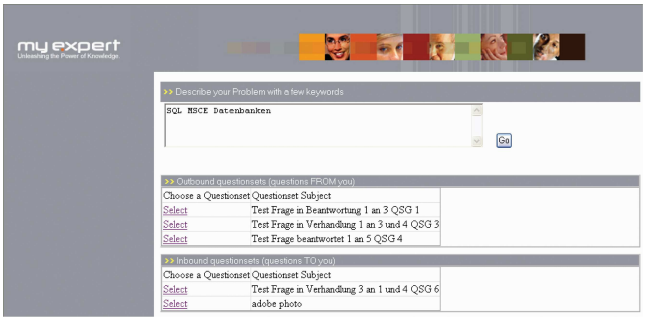

Figure 7.20.: MyPage: Personalized *KnowMarket* homepage

question or response. The *Outbound question sets* area contains questions that have been posed by the user currently logged-on. The *Inbound question sets* area contains a list of question that have been posed to the user.

A question can be entered in the upper text box. After clicking the *GO* button, the question is split up into component parts and then checked against the competency database. The *Result of Competency Search* page then appears.

Figure 7.21.: Result of Competency Search and Definition of Weightings

The *Result of Competency Search* page (see Figure 7.21) shows competencies that have been located for a question. To select or deselect competencies, the checkboxes in the tree view control element on the left page need to be activated or deactivated. The *Refresh* button updates the selection of competencies. For each competency a

weighting can be defined. The *Find Experts* button starts the matching algorithm and then takes the user to the *Result of Matching* page.

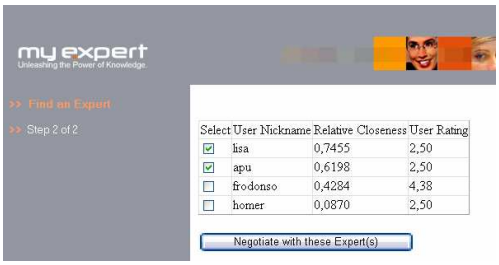

Figure 7.22.: Result of Matching

Experts suitable for answering the question are shown in the *Result of Matching* page (see Figure 7.22). Activating the checkboxes selects these for the negotiation phase.

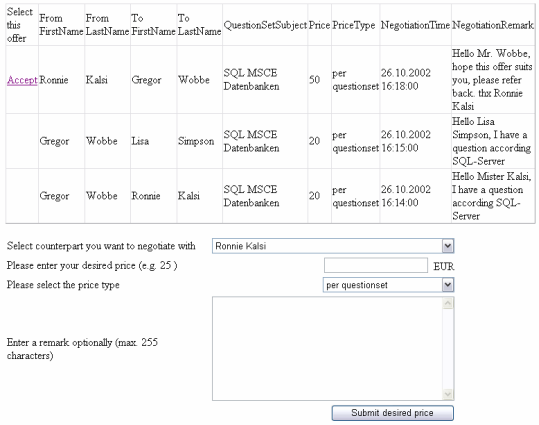

Figure 7.23.: Example of a Negotiation

Figure 7.23 shows the page for a negotiation between experts and advice seeker. Negotiation history regarding the question of an advice seeker can be seen here. At the start, the advice seeker addresses his price expectation towards the selected experts. Once the user that is being asked is logged on to the knowledge market, he will see the question in the *MyPage inbound box*. He can now either accept the offer using the *Accept* Button or make a counter-offer.

7.10. Evaluation of Software Engineering Aspects

There are different possible criteria to evaluate the software engineering aspects of the *KnowMarket* system.

Functionality. Functionality is the extent a system has implemented the planned features. Even though *KnowMarket* is only a proof-of–concept prototype, the main functions of an electronic market for online advice which have been envisioned in Chapter 5 are implemented. The only main function missing is payment. However, it is also possible to integrate the payment function by a specialized service provider like *PayPal* [8], which also offers Web Service interfaces for its functions (see e.g. [158] for payment infrastructures).

Scalability. Scalability describes how easy a system can handel an increased number of users or transactions. The multi-layer architecture of *KnowMarket* ensures a high scalability, because the layers can be installed on different machines and some layers can run on parallel machines. For example, the Web layer can run on several internet servers and an additional load-balancing component can distribute the HTTP calls between them.

Maintainability. Maintainability means how easy a system can be adapted, enhanced, or corrected to satisfy changing requirements [99, p. 46]. *KnowMarket* is developed according to an object-oriented approach. Also, the layered architecture assigns clear roles for each layer. The service are encapsulated in the form of XML Web Services. This should help to maintain the system because there is a clear modularization and limited interdependency of classes and layers.

Reusability. Reusability is the degree to which parts of the systems can be used again for other applications [99, p. 64]. The service oriented architecture (SOA) of *KnowMarket* makes it easy for components to be called and therefore reused from different applications. An important factor for the reusability is the compliance with open standards like XML, SOAP, UDDI and XML schemas like HR-XML. *KnowMarket* is developed according to these standards and ensures therefore high inter-operability of different components and consequently better reusability.

Reliability. Reliability is the ability of a system “to perform its required functions under stated conditions for a specific period of time” [99, p. 62]. Because *KnowMarket* is only a proof-of–concept prototype, no statements can be given for the reliability in operational usage.

Portability. Portability describes how easy a system can be transferred to another hardware and software platform than the planned one [99, p. 56]. Because the prototype is written for the .NET Framework, it can run on every platform that supports .NET. Beside the Windows version, the Mono project [4] also offers Linux, MacOS X, BSD, and Sun Solaris versions of the .NET framework.

7.11. Summary

With the change from a functional to a process-oriented view in companies, it is essential that company processes beyond company boundaries are also supported by IT systems. Particularly with smaller firms, a problem with cross-company integration is firstly that internal integration is still insufficient, and secondly that data is not available in a form that's suitable for exchange between companies.

XML Web Services can make a considerable contribution to meeting the increasing demand for integration of heterogeneous IT environments and to simplify business-to-business communication using standardized interfaces. Making company-internal and external communication simpler helps keep transaction costs to a minimum. The electronic knowledge market discussed here demonstrates one way in which, in view of the increasing relevance of knowledge resources, the need for knowledge management in companies may be linked with the benefits of Web Service technology for acquiring knowledge, at the same time keeping transactions low-cost and time-efficient, and enabling knowledge market functions to be used by various clients with the aid of XML Web Services.