RZ 3258 (# 93304) 7/10/2000
Computer Science 13 pages

Research Report

Design of Roles and Protocols for Electronic Negotiations

M. Ströbel

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

Design of Roles and Protocols for Electronic Negotiations

Michael Ströbel

IBM Research, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Abstract **-** Support for negotiations in electronic markets is one of the primary issues in today's e-commerce research. Whereas most activities are focused on automation aspects, only few efforts address the design of electronic negotiations. However, for the efficiency of electronic negotiation processes and the success of resulting settlements, it is essential to achieve an a-priori agreement among the negotiating parties about issues such as the syntax and semantics of offer specifications, the sequence of actions, possible responses, or time constraints, because these factors might influence, for instance, the fairness of the electronic negotiation.

This paper demonstrates how an explicit and specific design can capture the way electronic negotiations are organised. The organisation design meta-model presented is part of SILKROAD, a design and application framework for electronic negotiations. On the basis of this framework, organisations creating an electronic market or sellers intending to offer potential buyers the option to bargain, can generate, in a flexible and efficient way, customised electronic negotiation systems supporting the roles and protocols designed. Furthermore, the consequent application of this meta-model can lead to the discovery of common negotiation patterns, eventually resulting in a reference model for electronic negotiations.

Keywords: electronic negotiation, electronic commerce, market design, application framework

1 INTRODUCTION

Let us assume that a new electronic market is being created. Given the nature of the traded objects, price-focused discovery mechanisms such as auctions are not applicable. An agreement between a seller and a buyer has to consider both multiple attributes of the object (e.g. the quality), as well as terms and conditions of the transaction, such as the delivery time or the return policy. A critical factor for the success of the electronic market will be the market participants' common understanding of the protocol of such a multi-attribute negotiation, and the distinct roles of the involved parties. This a-priori agreement has to address a number of questions such as: Should all attributes be negotiated at the same time, or one after another? Can one party counter an offer by including or requesting additional offer attributes (e.g. a guarantee)? What happens if an offer is not signed within a certain timeframe? A representation for this settlement on roles and protocols in a multi-attribute negotiation is an agreement scenario.

Like any other information system, the creation of an electronic market can be structured along the typical system development phases of analysis, design and implementation. The design activity has to comprise the agreement scenario. Choice and further specification of this scenario will vary depending on the market and transaction requirements identified in the analysis phase. In the implementation phase, the agreement scenario is then mapped to a technical architecture and application system.

Nevertheless, there are no common means by which the market creator and participants can reason about the potential dimensions of agreement scenarios. In 1991, Holsapple et al. [1] already identified this need for general models of negotiations, which could be used to characterise the nature and process of the negotiation, formalise its aspects, and which have the flexibility to describe a wide range of possible structures and interactions. Modelling aspects have still been neglected in related research, with the undesirable consequence that it is difficult to discuss agreement scenarios on a conceptual level, and that design efforts cannot be reused and refined in the implementation phase in a formal way.

This lack of support for the design of agreement scenarios is the underlying motivation for SILKROAD – a design and implementation framework for electronic negotiations. One deliverable of this project, the organisation design meta-model for the specification of the roles and protocols in a negotiation, is presented in this paper. The aim of this meta-model is not to introduce a new modelling approach for a specific type of information systems, but to demonstrate how existing modelling concepts can be used and adapted to the domain of electronic negotiations.

Beyond this design dimension, SILKROAD also addresses the fact that today's agreement services, such as electronic auctions or agent systems, are not flexible or configurable enough to support the requirements of a larger range of existing or emerging agreement scenarios in business practice [2], [3]. Accordingly, unique and proprietary solutions are created again and again, and enormous efforts are spent on integrating isolated solutions. On the basis of the design approach presented in this paper and a framework of reusable negotiation service components, SILKROAD allows, for instance, electronic market organisations or negotiation service providers, to generate customised negotiation support mechanisms for a broad range of agreement scenarios.

After referring to the theoretical foundations in this introductory section, the approach chosen for SILKROAD will be illustrated in more detail in Section 2. The basic elements of the organisation design meta-model are presented in Section 3, and applied in Section 4, where based on sample cases the application of the meta-model is demonstrated. The further refinement of agreement scenarios and the generation of runtime specifications in the SILKROAD framework is then illustrated in Sections 5 and 6 before the current status of the approach is discussed in Section 7.

1.1 The concept of media

In SILKROAD, the notion of media and the media reference model [4] are used to conceptualise electronic negotiations. Media are platforms where the exchange (a transaction) of tangible or intangible objects is co-ordinated through agent interaction. These platforms can be described in terms of three main components:

- Channels:
  Agents access a medium via channels that can transport the objects to be exchanged.
- Logical space:
  The syntax and semantics defined for the objects, which the agents exchange.

- Organisation:
  Roles describing the types of agents and protocols specifying their interactions.

An electronic medium, in particular, is a medium with electronic (digital) channels that transport data. The agents, however, still might be humans or organisational units and do not necessarily have to be software agents.

The media reference model (see Figure 1) identifies several phases of interaction. In the knowledge phase, agents gather information concerning the products offered or the profiles of other agents. The interface between the intention phase, where supply and demand is specified, and the agreement phase, where the terms and conditions of a market transaction are negotiated, is an offer [5].

If at least one agent submits an offer, the agreement phase is initiated. In the simplest case, another agent merely has to accept this offer in order to reach an agreement. Upon agreement, the transition to the settlement phase is marked by a signed contract. Negotiating takes place when, based on the offers made in the intention phase, an agreement cannot be reached, or the initial agreement has potential for optimisation and the agents want to discuss their offers. From the perspective of one agent, negotiating is characterised by the modification of its own offer or the efforts to change another agent's offer.

An electronic medium, supporting negotiation processes in the agreement phase, is denoted an electronic negotiation medium. The primary goal of SILKROAD is to facilitate the design and implementation of electronic negotiation media.

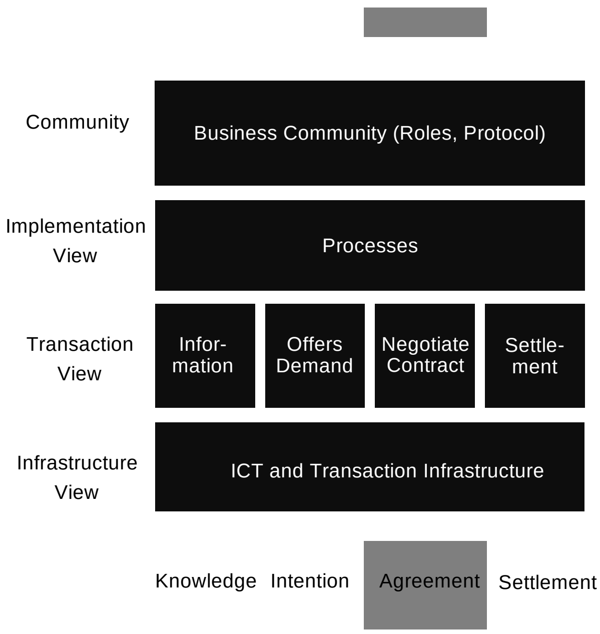


Figure 1: Agreement phase in the media reference model [4].

1.2 Media design

The design of an electronic negotiation medium has to consider three dimensions [6]:

- The organisational design describes the roles and protocols that will be supported.
- The communication design is necessary to structure the logical space for the agents.
- The IT design addresses the architecture of technical channels and interfaces.

SILKROAD supports all of the introduced design dimensions. The emphasis in this paper, however, is on the organisation design.

2 SILKROAD APPROACH

The two core elements of SILKROAD are the ROADMAP and the SKELETON. The SKELETON provides several modular and configurable negotiation service components that can be used for the implementation of electronic negotiation media. The SILKROAD SKELETON can be classified as an application framework [7] – the skeleton of an electronic negotiation medium, which can be customised to specific agreement scenarios. Following the reuse and 'inversion of control' paradigm of frameworks, SILKROAD developers can subclass framework components to implement specific application logic. But the most common usage of the framework will be a deployment of instances of the framework components, which are customised at runtime on the basis of specifications generated in the design phase.

The process of designing an electronic negotiation medium is structured according to the SILKROAD ROADMAP design action model, which provides two main constructs:

- Organisation design meta-model (ODMM).
  This meta-model supports, but also constrains, the process of designing concrete organisational models of negotiation media. It allows the specification of the structure (roles) and behaviour (protocols) of electronic negotiations with design building blocks that explicitly represent the functionality of the underlying negotiation service components on a conceptual level.

- Communication design meta-model (CDMM).
  The aim of this meta-model is to assist the design of the logical space of an electronic negotiation. It provides the means to express syntactical and semantical representations of the objects and the terms and conditions of market transactions.

Before the design action model can be applied, it is essential to perform an analysis of the preconditions of the agreement phase of the electronic transaction. For the organisation design, characteristics such as the transaction value (high, low, perishable etc.), the risk for the agents involved in this transaction, or the customisability of the object of the transaction have to be investigated in order to select an appropriate organisation design for the electronic negotiation (see for example [8]). In addition to the characteristics of the transaction, this analysis also has to cover aspects of the agents' roles (their beliefs, desires, intentions…) as well as the relationships between the agents (dependency, distribution of market power, level of confidentiality, intensity of information exchange, etc.).

The organisational design is then performed on the basis of this analysis of preconditions and the ODMM. The result of this process is a negotiation media organisation design. The SILKROAD ROADMAP (see Figure 2) distinguishes between a conceptual design and an integrated design. The integrated design does not add more detail to the conceptual design in the sense that models are further refined, but links the organisation and communication design, thus resulting in a unified, consistent design model for electronic negotiation media. By emphasising these two distinct design activities, the complexity of the final media IT design and implementation is mainly reduced to a generation of executable design representations, which specify the interaction of the SILKROAD SKELETON negotiation service components at runtime.

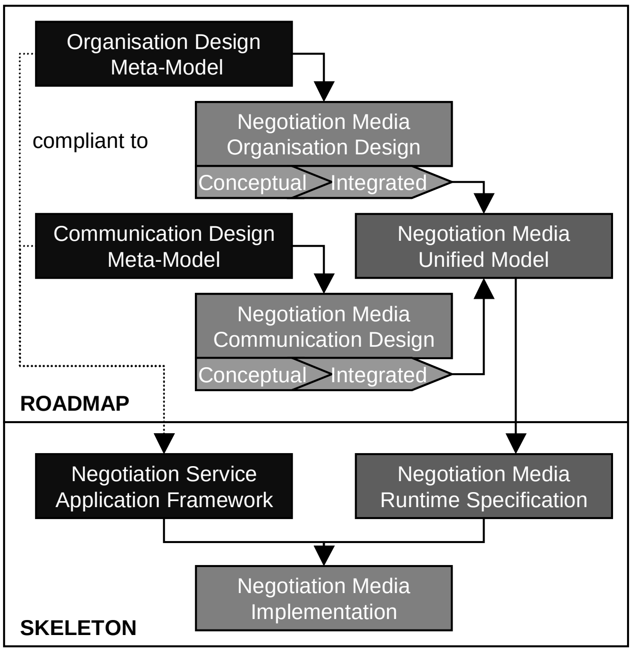


Figure 2: SILKROAD overview.

Referring to the views of the media reference model (see Figure 1), SILKROAD specifically addresses the community, implementation and transaction view. The roles and protocols of the community view are modelled within the SILKROAD design phase. Actual processes on the implementation view are then executed on the basis of the generated runtime specifications, which invoke the negotiation service components in the transaction view.

3 ELEMENTS OF THE ORGANISATION DESIGN

This section outlines the conceptual design elements of the SILKROAD ODMM in more detail. Related integrated design aspects will be discussed in Section 5.

3.1 Overview

The basis for all design activities in SILKROAD the common meta-model in Figure 3, which introduces the principal entity types, and the relations between these types, for the organisation design meta-model as well as the communication design meta-model. The entity types marked with a black background constitute the ODMM; those reserved for the CDMM are indicated with lighter grey.

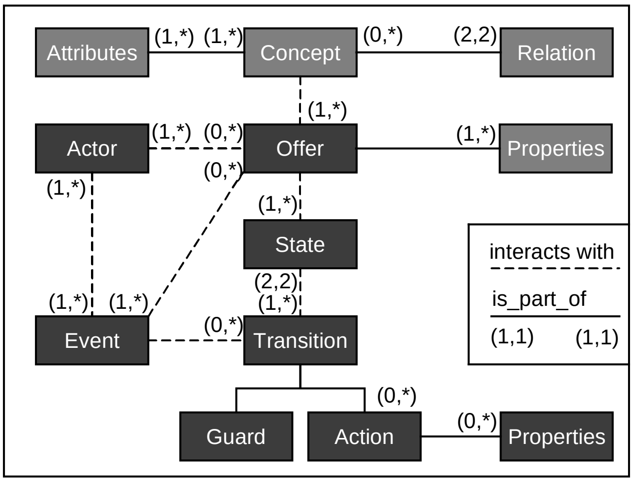


Figure 3: SILKROAD design meta-model.

The semantics of the ODMM can be summarised as follows: An *offer* has one or more associated *states*. *Actors* create, delete or modify offers and cause *events,* which can stimulate *transitions* between the states of an offer. One *event* can be caused by multiple actors and might be associated to a set of offers. A transition always transfers an offer from one state to another state and will only occur if the *guard* condition is true. The 'firing' of a transition might also invoke an *action* with certain *properties*.

The concept of state charts is the underlying modelling paradigm (for both the organisation and communication design). The advantage of state charts is that they are commonly used in information systems design and also part of UML [9]. Therefore it can be assumed that most designers are familiar with this approach.

The task of the organisation design is to model the various states of offer types within an electronic negotiation and thereby to discover the associated actor roles, events, transitions, guards, and actions. This results in the design of the roles and the protocol of the electronic negotiation medium. *Roles* are defined as the total of all possible events an actor can raise. One graph of entities and relations constitutes an *agreement scenario*. One agreement scenario represents all necessary roles and the protocol for the complete agreement phase specification of a transaction in an electronic negotiation medium. The *protocol* constitutes all the rules in one scenario, represented by valid states and transitions, which define how agents come to an agreement. An electronic negotiation medium can feature several agreement scenarios, which, at runtime, can be instantiated in parallel.

For the specific usage in SILKROAD the semantics of the generic entity types introduced in the meta-model are further specified.

3.2 Actors

An actor is either of an agent type or of a service type. An agent type can assume three roles in a negotiation media organisation model (see Figure 4):

- Buyer (B)
- Seller (S)
- Intermediary (I)

Within SILKROAD, autonomous proxy agents can represent all of these roles in the agreement scenario and automate certain aspects of agent interaction. For the design phase, the distinction in human or proxy agent is not important – it is up to the organisations represented in a negotiation to decide whether to use proxy agents or not. The SKELETON provides both, interfaces to autonomous agents, which can be used for real-time synchronous interaction, as well as interfaces for human agents with asynchronous interaction.

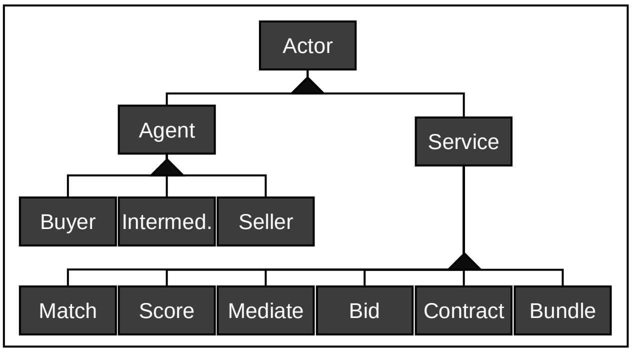


Figure 4: Actor types.

A service is defined as an input-output system that receives inputs (e.g. an offer), processes these inputs internally, and then produces outputs (e.g. an aggregated utility value for this offer). Actor service types in the ODMM represent the following negotiation service components in the SILKROAD application framework:

- Match
  The match service will test a set of offers-to-buy and offers-to-sell for compatibility. This test is based on an evaluation of the set of constraints in one offer (e.g. ‘to buy steel’) towards the property values specified in a corresponding offer (e.g. ‘to sell steel’). The result of the matching operation is a set of $n \geqq 0$ compatible offer pairs.
- Score
  The score service receives a set of candidate offers and an offer specifying evaluation criteria (combinative functions, weights and utility functions) to calculate a ranking for the set of candidate offers, depending on their evaluation scores, thus determining the ‘best’ offers from the perspective of one agent.
- Mediate
  For two non-matching offers, this service suggests an agreement depending on mutual negotiable offer elements (e.g. the constraint ‘price < $200’ might be negotiable for the buyer and the same is true, from the seller’s perspective, regarding the property value ‘price = $210’) and the situation-specific preferences (assignments of importance) of the agents towards these negotiable issues.
- Bid
  This service persists and advertises offers, if they comply with certain bidding rules (e.g. ‘ascending prices’), in a negotiation session until a clearing rule applies (expiration of the bidding period, time of inactivity etc.) and the winning offer can be determined.
- Contract
  Any offer can be transformed into a legally binding offer with a signature of the issuing party. If another agent also signs this offer or a matching offer, a contract is established. This service requests agents to sign offers, checks signatures and authorisation, and seals the offer(s).
- Bundle
  Based on this service, offers can be collected or aggregated to cumulative offers. A typical application is demand aggregation, where, for instance, buyer agents group together in order to increase their

negotiation power and consequently, to achieve better prices. Another type of bundling takes place if one offer-to-sell consists of a bundle of complementary services (e.g. transportation from A to B, B to C, and C to D) provided by different seller agents.

Two services in the framework are not situated in the core agreement phase: *bid* and *contract*. However, from a horizontal integration perspective, these services are necessary for the integration with the service layer in the neighbouring intention and settlement phases of an electronic transaction (see Figure 1).

In an agreement scenario, these negotiation service components can be combined to provide a scenario-specific level of negotiation support. For example, a scenario might define that the *bundle* service first be invoked to accumulate demand from several agents. Then the *bid* service collects offers from suppliers, which the *match* service tests for compliance with the demand. If no matches can be found, the *mediate* service can still suggest an agreement, and the *score* service may compare several potential agreements on the basis of the agent's preferences, before the *contract* service finalises the negotiation process (see the examples in Section 4).

3.3 Offers

Offers are the primary means of communication in the agreement phase (see for example [10]) – and in SILKROAD, the only supported type of agent interaction.

As will be shown in the integrated design (Section 5), the offer type links to the communication design as it addresses not only 'how to negotiate', but also 'what to negotiate'. In the communication design, offers are specified with a set of structural properties and associated to concepts in an ontology (see [11]). For the purpose of the conceptual organisation design, it is sufficient to derive states for the types of offers used in the negotiation protocol. In the communication design, these states are further specified from a syntactical and semantical perspective.

The modelling of conditions in the organisation design might require an evaluation of offer properties (see the section on guards below). To access the properties of an offer the notation *offer.property* is used. A set of offers may also have additional set-properties, such as the number of offers in this set.

The ODMM distinguishes between two types of offers which can be issued by agents: offers-to-buy (O2B) and offers-to-sell (O2S). Depending on the agreement scenario chosen, a final contract might require that two compatible offers be found, which are both signed by the originator with respect to the corresponding offer (one-sided contracting), or that one offer is signed by both agents (double-sided contracting).

Referring back to the design meta-model in Figure 3, an offer is associated to one or more states, to one or more actors, and might be related to certain events. To relate a state to an offer the notation *offer.state* is used.

3.4 States and events

Electronic negotiation media generated with SILKROAD can be characterised as hybrid negotiation systems. Kersten and Noronha express the need for hybrid (partly automated) negotiation systems, due to the fact that despite all advances in automation, humans still need to be involved in the agreement phase of electronic transactions [10]. In hybrid negotiation systems, structured or formalised tasks are automated, and decision support mechanisms are used to assess unstructured tasks, whereas humans interactively control the execution of the negotiation and perform the exception handling.

This hybrid system character is reflected in the ODMM in the two primary types of actors – agents and services (and again on a technical level in the distinction between human and autonomous proxy agents). The ODMM specifies that events are associated to actors. Hence, an event can be further characterised by the actor causing the event to be either *agent.event* or *service.event*. This is comparable to the notion of internal and external events in UML. If an offer is in a state where an activity is performed by a service and only an event caused by the service can stimulate a transition, the state is denoted a *service-state*. As the set of services is predefined, the number of corresponding service-states is limited to the six types of services available (e.g. 'offer.matching', 'offer.bidding'). Other offer states typically require an agent.event (i.e. the offer is 'waiting' for the agent's decision). Hence, transitions from service-states to non-service-states model the switch from automated framework processing to agent interaction.

An event can be associated to one or more offers. If, for instance, an agent submits a counteroffer, this event (e.g. 'buyer.submitted') will be related to the offer, which was actually submitted. If, in another example, the *score* service raises a ‘score.completed’ event, this event typically points to a set of evaluated offers.

3.5 Guards, actions and properties

The ODMM specifies that a transition fires if an event occurs that causes the guard condition of the transition to evaluate to ‘true’. Guard conditions can evaluate events and the offer(s) associated to an event. If, for instance, the *match* service causes the event *match.completed* with an associated result set ‘o’ of ‘n’ compatible offers, the offer-set property ‘o.n’ can be evaluated by the guard conditions of transitions connected to the service-state *offer.matching*. For ‘o.n = 1’ a guard condition of a transition to the state *offer.contracting* could evaluate to ‘true’, for ‘o.n > 1’ a transition to the state *offer.bidding* may fire.

Optionally, a transition causes an action with certain properties. Actions invoke services such as *match* or *score* and lead to service-states. If no action is specified in the design for a transition, the consecutive state is not a service-state. Thus, the execution within the framework stops in the consecutive state, and listens for an agent.event. An action can be customised with a set of *action.properties*. These properties determine the operation of the service, e.g. customise its behaviour. The *mediate* service, for example, has a *mediate.equality* property, which specifies the fairness of the suggested agreement (e.g. ‘equal distribution of the transaction value’ or ‘mediation in favour of the buyer’, see [12]). To give another example, the *match* service has a *match.fuzzy* property, which is used to turn on fuzzy matching – an operation mode returning ‘close’ matches that did not fulfil all constraints in the offer initiating the match operation. Not all properties have to be specified in an agreement scenario (default values are provided).

Property values are either specified during the design phase or requested from agents at runtime. An example for this second case of a dynamic property is the *match.fuzzy* property. Depending on the scenario requirements, a value for this property can be set either already at design-time, or by an agent during the electronic negotiation. In this case, a service posts a request for a property to the agent and pauses the service execution until the property value is received.

4 EXAMPLES

This section demonstrates how the elements of the ODMM can be used to model negotiation scenarios. Regarding the broad variety of supported scenarios, the entire expressive scope of the ODMM cannot be demonstrated in this overview. Furthermore, to simplify the presentation, the sample models are not complete agreement scenarios in the sense defined in Section 3.1, but selected clippings.

The graphical notation in this section follows the UML conventions for state-chart diagrams [9]. States are represented by rounded rectangles. The pre-defined service-states are additionally marked with lighter grey. The offer type related to a state is indicated with capital letters preceding the state identifier. Transitions are arrows connecting states, and events (‘E:’), guards (‘G:’), actions (‘A:’), and properties (‘P:’) are specified as textual information complementing the transition arrows.

4.1 Matchmaking with conditional scoring

Figure 4 illustrates an example of an agreement scenario that provides basic matchmaking of offers from buyers/sellers and scores (ranks) offers in those cases where more than one offer matches a request.

In this example, an actor of the type buyer causes an event ‘buyer.match’ for an offer-to-buy, which activates a transition from the state ‘O2B.advertised’ to the service-state ‘O2B.matching’. Linked to this transition is the action *match*. If the set of matching offers associated to the ‘match.completed’ event contains only one offer, a transition to the state ‘O2B.matched’ takes place. In this state, only the buyer agent can cause events (‘buyer.accept’, ‘buyer.decline’) that activate transitions, hence the framework execution stops and waits for an agent.event. If the buyer agent chooses the option ‘buyer.accept’, the *contract* action is invoked and the offer transitions to the state ‘O2B.contracting’. Alternatively, while processing the ‘match.completed’ event, the guard condition ‘n > 1’ may evaluate to ‘true’. In this case the offer-to-buy transitions with the action *score* and the property ‘score.max = 3’ to the state ‘O2B.scoring’. The event ‘score.completed’ then results in the state ‘O2B.scored’, where again the buyer agent has the option to accept one of the scored offers.

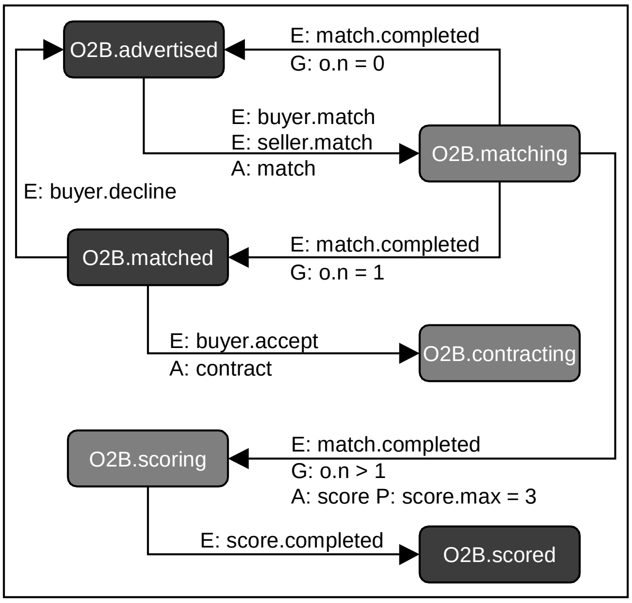


Figure 5: Matchmaking example.

For offers-to-sell, the scenario is identical (not shown in Figure 5) and specifies the states ‘O2S.advertised’, ‘O2S.matching’, ‘O2S.matched’, etc. This symmetry in the design of the negotiation scenario is reflected in Figure 5 in the fact that the transition to the state ‘O2B.matching’ can be caused by the event ‘buyer.match’ or ‘seller.match’. Both sellers and buyers can initiate the matchmaking process. Accordingly, one event can also cause multiple state transitions for different offer types. In the example, the event ‘match.completed’ may lead to the transition of offers-to-sell associated to this event from the state ‘O2S.matching’ to the state ‘O2S.matched’.

The protocol reflected in this agreement scenario, defines, for instance, a rule that in the state ‘O2B.matched’ the buyer has only two options: to either accept or reject a matched offer-to-sell. In addition to this, guard conditions can also evaluate the time an offer persists in a certain state. In the example, an automated transition back to the state ‘O2B.advertised’ could be triggered if a certain time limit for initiating the contracting process expires.

The roles of buyers and sellers are defined by the events these actor types can raise in the scenario. In the case where a buyer agent initiated the matchmaking process, the role of a seller agent could have, for example, the option to monitor the success of its offer by inspecting the evaluated offer score in the state ‘O2S.scored’ (per default, the framework notifies agents if their offers assume a new state). Another aspect of the seller role might be the responsibility to sign an offer-to-sell within a certain timeframe.

4.2 Binding offers with mediation

The second example illustrates a different usage of the *contract* service and also introduces the *mediate* service. It could be a requirement for the scenario illustrated in the previous section that the offers of the sellers be binding in the sense that a legal contract can be generated on the basis of a one-sided buyer acceptance. In this case, the *contract* service obliges the sellers to sign their offers before they are advertised and available to the *match* service.

In Figure 6, this is reflected in the ‘O2S.signature*’ property which is associated with the ‘contract’ action. The ‘*’ indicates that this is a dynamic property, meaning that the value for this property is requested at runtime from the agent initiating the action. If the signature of the agent is valid, the offer-to-sell transitions to the state ‘O2S.signed’. In this modified example, the subsequent *match* action is customised at design-time with a *match.fuzzy* property value of ‘true’ in order to always identify close offers-to-buy.

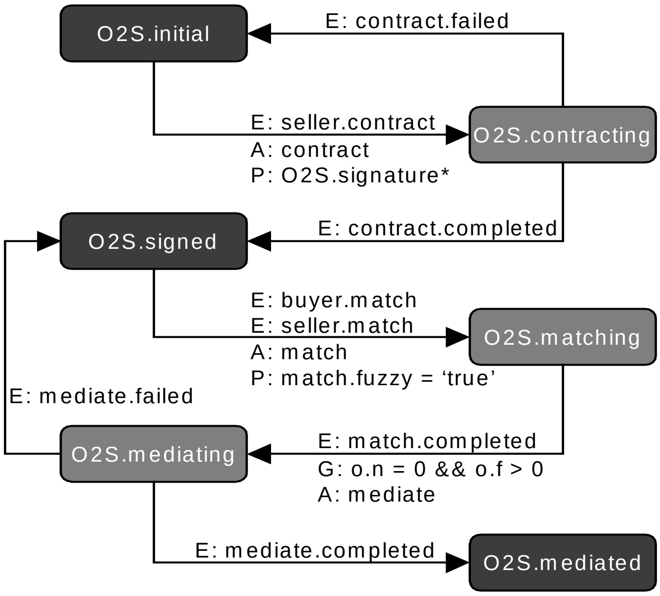


Figure 6: Binding offers example.

Once the matchmaking process is completed, a guard condition evaluates to ‘true’ if the set of identified offers contains no complete matches, but at least one close match (‘o.f’ in the guard refers to the number of fuzzy matches in the set of offers). The action associated with this transition is *mediate*. The *mediate* service will suggest a fair agreement based on the elicited preferences of the buyer and seller agent towards negotiable issues in their respective offers. This mediation process results in pairs of modified offers in the ‘O2S.mediated’ and ‘O2B.mediated’ state with suggested property values for the negotiable issues.

In a likely continuation of this scenario, the agents could be requested to sign these modified offers in order to reach an agreement. Another alternative arises if the number of mediated offers is greater than one. In this case, the subsequent action could be to either *score* the mediated offers or to enter a phase of competitive bidding using the *bid* service.

5 INTEGRATED DESIGN

In the integrated design, the results from the conceptual organisation and communication design are combined to one unified negotiation media model.

The goal of the communication design in SILKROAD is to structure the logical space of an electronic negotiation medium. The objects of the communication design are the offers exchanged in a negotiation. In the CDMM, the design of offers is separated into a definition of domain ontologies for the semantical aspects of offer communication, and the specification of offer states from a syntactical perspective.

Whereas the conceptual design mainly deals with the offer semantics, the communication design in the integrated design phase complements the offer states identified in the organisation design with offer properties defining the structure of offer types in these states and the corresponding syntax representation.

In the CDMM the following offer structure properties with associated sets of property values are available to represent an offer state:

- Signatures (none, buyer, seller, both)
- Timestamps (none, start, end, both)
- Domains (attributes, values, ranges, dynamic)
- Constraints (basic, negotiable, weighted)
- Counters (none, one, many)
- Criteria (none, importance, utility, functions)

Details regarding the semantics of these properties can be found in [13]. To give an example, the value *dynamic* for the property *Domains* explicitly allows an agent to define the range of values for an attribute domain in an offer-to-sell, only if the agent knows more about the agent interested to buy. A typical example can be found in the insurance industry, where quotes are usually dependent on age, medical record, driving experience, etc. Another example is the *negotiable* value for the *Constraints* property. It allows an agent to express the intention to concede on this offer attribute if he/she is compensated on another attribute, thus enabling tradeoffs between buyer and seller.

To continue with the example introduced in the previous sections, an offer in the 'O2S.signed' state requires a signature of the seller. This change in structure has to be specified in the integrated design by setting the *Signature* property for offers in the state 'O2S.signed' to the value *seller*. In addition to this, the subsequent *mediate* service requires negotiable constraints. Therefore, the *Constraint* property value for offers in the state 'O2S.signed' also has to be set to *negotiable*.

The integrated design may result in additional offer states, in order to reflect necessary changes to the offer structure. These changes also might require additional agent interaction. In the example, the *score* service requires the initiating offer to feature evaluation criteria such as utility functions. Therefore, an additional state 'O2B.updated' is necessary if an offer in 'O2B.matched' does not necessarily contain evaluation criteria. The event activating a transition from 'O2B.matched' to 'O2B.updated' is 'buyer.submitted'. The guard for this transition specifies a successful validation of the modified offer according to the offer structure properties set for the state 'O2B.updated'.

At runtime, the structure of an offer instance can be validated for compliance with the offer type state structures defined in the integrated design (see below). Before the actual generation of runtime specifications takes place, a number of consistency checks regarding the unified negotiation media model can be performed. These checks test the model for completeness and accuracy towards the underlying negotiation service application framework – from a structural and behavioural point of view. The unified model has to comply, for example, with the following types of conditions:

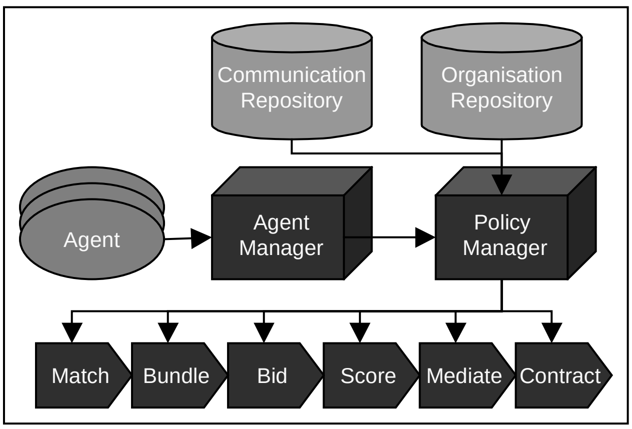
- Only service-states and actions available in the application framework are used, and events with actions activate only transitions to service-states.
- Guard conditions evaluate only offer properties, which are available at the preceding offer state.
- Offer structure properties required for subsequent service executions are specified.
- Some services require the usage of other services (e.g. *match* is a necessary predecessor of *mediate*).

If the unified negotiation media model passes the consistency check, the next step in SILKROAD is the generation of executable representations for this design[1].

6 RUNTIME REPRESENTATION

On the basis of the negotiation media design, runtime representations of the modelled agreement scenarios for the SILKROAD SKELETON are generated. These runtime representations are persisted in communication and organisation design repositories as *agreement scenario policies* (see Figure 7).

The facility in the SKELETON responsible for controlling the execution of actual agreement scenarios is the *policy manager*. It checks, depending on the current state of offers in the agreement scenario, offer instances for semantical and syntactical correctness as well as events and actions of agents for compatibility with the protocol and role specification in the organisation design.



[1] Once graphical tools are available to support the design process in SILKROAD, the consistency check can already be performed at design-time, when new states or transitions are added.

Figure 7: Runtime architecture overview.

To represent the communication design, XML schemata are generated. The runtime representation for the organisation design is an extended state machine table. The generation of this table is straightforward: for every transition in the organisation design state charts, the old and the new state, as well as the specifications for the events, guard, action and properties are converted to table values as shown in the following example for transitions from the state 'O2B.matching' in Figure 3:

State.ol	O2B.matching	O2B.matching	O2B.matching
Event	match.complet	match.complet	match.complet
Guard	o.n = 1	o.n > 1	o.n = 0
Action		score	
Props		score.max = 3	
State.ne	O2B.matched	O2B.scoring	O2B.advertise

Table 1: State machine table example.

This table is interpreted by the policy manager (PM) in the following way: for every event, the PM looks up the current state of the offer associated with the event. The table is then queried for an entry with this combination of offer.state and actor.event. If an entry (a transition) is found and the guard condition evaluates to true, two flows of execution are possible. First, there is an action associated with this transition, then it is invoked with the given action.properties (if applicable) and the offer state is changed accordingly. Second, for transitions with no actions, the PM simply changes the state of the offer and listens for further events.

In this case, the state machine table can also be used to determine the next possible actions available to an agent. Depending on the current state, the PM looks up all transitions, which can be activated through events caused by the specific agent type. This set of events represents the next possible actions an agent belonging to this agent type can invoke. The agent manager component in the skeleton provides an interface to the agents for retrieval of this set of next possible actions and for activation of the corresponding events.

7 DISCUSSION

This paper presented an approach for the design of roles and protocols in an electronic negotiation. On the basis of such a design and the SILKROAD negotiation service application framework, an electronic negotiation medium implementation can be generated. This combination of design approach, runtime generation process and generic architecture framework, allows, for instance, negotiation service providers to support, with low setup costs, a broad variety of agreement scenarios for electronic markets in a flexible but standardised way.

In addition, the conceptual design allows structuring the problem domain of electronic negotiation support, and its graphical notation eases the communication of modelling aspects to non-technical stakeholders. If all involved parties agree to the modelled negotiation scenario, the likelihood that actual electronic negotiations (following this model) will succeed is certainly increased. To guarantee this from a technical perspective, the meta-model proposed also provides means in the integrated design phase to validate the negotiation scenario towards its consistency and completeness.

To summarise, SILKROAD defines a language to express agreement scenarios, which captures syntactical and semantical aspects of the communication in an electronic negotiation as well as aspects related to the way these negotiations are organised. From a high-level perspective, the language for this organisational design consists of a set of primitive actions (match, score, etc.), which can, adhering to a certain grammar, be combined to sentences (e.g. 'bundle – match – mediate – score'). This language can easily be extended to cover additional words (actions) – the underlying language processing (the SILKROAD skeleton) does not require any prior knowledge about the words available but relies completely on the information contained in the design.

An *elicitate* service, for instance, is currently being discussed, which would allow agents to construct and reveal preferences and evaluation criteria sequentially and based on mutual commitments. Adding this new 'word' to the existing language, requires updates to the SILKROAD ROADMAP (e.g. new consistency rules), but does not cause any changes in the SKELETON – the service can be added to the set of existing negotiation service components and will be invoked dynamically depending on its actual usage in agreement scenario policies.

The structured language that SILKROAD provides also defines its restrictions. Unstructured agent communication beyond offers (e.g. an inquiry for the meaning of a certain offer property), or offer processing beyond the functionality of the services available is not supported. Specific customisation of the implementation would make such exceptions possible, but then the generation advantage is lost. Hence, the scope of this approach is reduced to agreement scenarios, which can be expressed in the SILKROAD language. This issue of scope leads to a discussion of related work.

The idea to support a broad range of agreement scenarios with generic negotiation architectures is actively researched within the COSMOS project [14] and the TEM effort (see [3] and the article 'Combined Negotiations in E-Commerce: Concepts and Architecture' in this same issue).

COSMOS supports the design phase with models on the basis of rules [15]. Rules describe, for instance, the protocol of a marketplace with conditions, trigger and activation elements. Agents are modelled using state diagrams where, again, rules define the state transitions. The support for rules also allows capturing negotiation strategy formalisms for autonomous agents. The generation of these strategies is based, for instance, on genetic algorithms [16], leading to completely automated electronic negotiations. In contrast to this, SILKROAD does not aim to achieve complete automation of electronic negotiations and accordingly, the language presented does not provide means to design or represent negotiation strategies. The decision to not cover strategies and to focus on partly automated negotiations is based on two considerations. Firstly, Beam et al. [17] formulated a paradox which questions the applicability of formalised negotiation strategies: if a negotiation strategy is simple enough to be formalised then it probably can be deciphered by the opposing party – if it is too complex, it cannot be formalised. Secondly, as long as the math, especially for multi-attribute negotiations, is a mystery to the user and the preference capture remains obscure, trust regarding the loss of control and transfer of responsibility, for instance, to an agent remains the primary obstacle for widespread acceptance.

For TEM, Benyoucef and Keller evaluated several modelling paradigms regarding their application in the domain of electronic negotiations, and came to the conclusion that UML state charts with event-condition-action specifications are the most appropriate to use [18]. However, whereas SILKROAD also employs state charts, in TEM the state of a negotiation is modelled (e.g. 'taking bids' or 'clearing') – not the state of offers as in SILKROAD. Offer states not only allow for finer granularity, but also rich design consistency and runtime compliance validations. In addition, the states of different types of offers in one agreement scenario need to be considered in SILKROAD to support the symmetry of buyers and sellers – meaning that, from the perspective of the application framework, the roles of buyers and sellers are not distinguished regarding the available functionalities.

Finally, regarding future work, an interesting opportunity arises, once the design approach is actually in use and applied to a multitude of real agreement scenarios. Whereas the ODMM and CDMM specify 'how' to model electronic negotiation media, a reference model can specify 'what' to model. This reference model could evolve from a set of basic agreement scenarios, which, comparable to proven idioms in object-oriented software engineering [19], represent reusable best practices. A communication pattern might suggest that offers for 'internet services' usually comprise certain mandatory attributes such as the definition of a support contract or the pricing scheme (fixed rate, traffic dependent etc.). Similarly, an organisation pattern may recommend that the efficiency of agreements in the domain of 'internet services' can benefit from a short phase of bidding with a set of potential providers selected based on an initial offer ranking, as the standardisation of these services (e.g. web hosting) allowed in many cases in the past a large number of providers to quote and tends to stimulate a lot of competition. If this abstraction is feasible, SILKROAD could result not only a design and implementation framework, but also a reference model for electronic negotiations.

References

[1] C. Holsapple, H. Lai, and A. Whinston 'Analysis of Negotiation Support System Research'. *Journal of Computer Information Systems*, spring 1995.

[2] C. Holsapple, H. Lai, and A. Whinston 'Negotiation Support Systems: Roots, Progress and Needs.' *Journal of Information Systems* Vol.1, 1991, pp.233-247.

[3] M. Benyoucef, R. Keller, S. Lamouroux, J. Robert and V. Trussar 'Towards a Generic E-Negotiation Platform'. *Proceedings of the Sixth International Conference on Re-Technologies for Information Systems*, Zurich, Switzerland, 2000, pp. 95-109.

[4] B. Schmid 'Elektronische Märkte - Merkmale, Organisation und Potentiale', in: Sauter M. (ed.), Hermanns A. (ed.), *Handbuch Electronic Commerce*. Franz Vahlen Verlag, 1998, pp. 30-48.

[5] M. Ströbel 'The Effects of Electronic Markets on Negotiation Processes', *Proceedings of the ECIS 2000 Conference*, Vienna, Austria, 2000, pp. 445 - 452.

[6] B. Schmid 'Was ist neu an der digitalen Ökonomie?' in: Belz C. (ed.), Bieger T. (ed.), *Dienstleistungs-kompetenz und innovative Geschäftsmodelle*, Thexis Verlag, 2000, pp.178-196.

[7] M. Fayad, D. Schmidt, and R. Johnson *Building Application Frameworks - Object Oriented Foundations of Framework Design.* Wiley, New York, 1999.

[8] M. Bichler 'A Roadmap to Auction-based Negotiation Protocols for Electronic Commerce'. *Proceedings of the 33rd Annual Hawaii Int'l. Conference on Systems Science HICCS*, Hawaii, 2000.

[9] J. Rumbaugh, I. Jacobson, and G. Booch *The UML Reference Manual.* Addison-Wesley, Reading, England, 1999.

[10] G. Kersten, and S. Noronha 'Negotiations in Electronic Commerce: Methodological Misconceptions and a Resolution'. *InterNeg Research Report* INR02/99, 1999.

[11] M. Ströbel 'Communication Design for Electronic Negotiations Based on XML Schema'. *IBM Research Report* No.3289, 2000.

[12] M. Ströbel 'A Framework for Electronic Negotiations Based on Adjusted-Winner Mediation'. *Proceedings of the DEXA Workshop on e-Negotiations*, London, England, 2000, pp. 1020-1028.

[13] M. Ströbel 'Intention and Agreement Spaces – A Formalism'. *IBM Research Report* No.3279, 2000.

[14] W. Lamersdorf, M. Merz, and T. Tu 'Distributed Systems Technology for Electronic Commerce Applications'. *Lecture Notes in Computer Science* Vol. 1521, 1998, pp.135-148.

[15] C. Kunze, W., and Lamersdorf 'A Rule Management Framework for Negotiating Mobile Agents'. *Proceedings of the 4th Int'l. Enterprise Distributed Object Computing Conference* (EDOC'2000), IEEE, 2000, pp.135-143.

[16] T. Tu, F. Griffel, M. Merz, and W. Lamersdorf 'A Plug-in Architecture Providing Dynamic Negotiation Capabilities for Mobile Agents'. *Proceedings of the 2nd Int'l. Workshop on Mobile Agents* 'MA'98', Stuttgart, 1998, pp.222-236.

[17] C. Beam, A. Segev, M. Bichler, and R. Krishnan 'On Negotiations and Deal Making in Electronic Markets'. *Information Systems Frontier* Vol.1, No.3, 1999, pp.241-258.

[18] M. Benyoucef, and R. Keller 'An Evaluation of Formalisms for Negotiations in E-Commerce'. *Proceedings of the Workshop on Distributed Computing on the Web*, Quebec City, Canada, 2000.

[19] E. Gamma, R. Helm, R. Johnson, and J. Vlissides *Design Patterns – Elements of Reusable Object Oriented Software*. Addison-Wesley, Reading, England, 1995.