RZ 3289 (# 93335) 11/20/00
Computer Science/Mathematics 11 pages

Research Report

Communication Design for Electronic Negotiations on the Basis of XML Schema

Michael Ströbel

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

IBM Research
Almaden • Austin • Beijing • Delhi • Haifa • T.J. Watson • Tokyo • Zurich

Communication Design for Electronic Negotiations on the Basis of XML Schema

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. Only few activities, however, are focused on the design issues of electronic negotiation scenarios – e.g. the protocol to choose, obligations and responsibilities of the negotiating parties etc. However, an explicit negotiation design can also address what is commonly referred to as the ontology problem of electronic negotiations: how can be ensured, that the negotiating parties have the same understanding regarding the issues that are subject to the negotiation?

The solution this paper proposes is to perform a communication design for electronic negotiations, which explicitly specifies the common syntax and semantics, the logical space, of the negotiating parties. Furthermore, XML Schema is suggested as mechanism for the runtime representation of the logical space and the validation of actual negotiations from a syntactical and semantical perspective. On the basis of this approach organisations creating an electronic market or sellers intending to offer their buyers the possibility to bargain, can design and generate support mechanisms for electronic negotiations in a flexible and efficient way.

The communication design meta-model presented is part of SILKROAD, a design and application framework for electronic negotiations.

1. Introduction

Let us assume, a new electronic market for multiple sellers and buyers is being created. Due to the nature of the goods traded, price-focused discovery mechanisms such as auctions are not applicable because an agreement between a seller and a buyer has to consider both, multiple attributes of the good (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 efficiency of the future negotiation processes and the success of the potential settlements is an agreement about how the issues of a negotiation (attributes, terms and conditions) are represented as abstract objects in the negotiation and what this representation means to each of the negotiating parties. If, for instance, party X offers a delivery date of '12/10/2000' for a workstation to party Y, one potential conflict arises if this syntax is misinterpreted by Y as 'October 12' whereas X intended to offer 'December 10'. A semantical problem could occur if the meaning of this date to X is the point in time where the product leaves the premises of X, whereas Y assumes that this is the date where the workstation arrives on the premises of Y. This problem is referred to as the ontology problem of electronic negotiations [1].

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 mechanism, which is used to match sellers and buyers. Choice and further specification of this mechanism will vary depending on the requirements identified in the analysis phase. In the implementation phase, the design of the agreement mechanism is then mapped to a technical architecture and application system.

However, if the agreement mechanism requires negotiations between buyers and sellers, there exist no common means for the market creator to reason about the potential range of agreement mechanism design with its stakeholders. In 1991, Holsapple et al. [2] 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 are flexible to describe a wide range of possible structures and interactions. Nevertheless, modelling aspects have still been neglected in related research, with the undesirable consequence that it is difficult to discuss agreement mechanisms 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 mechanisms is the underlying motivation for the SILKROAD project. The SILKROAD framework can be used, for instance, by electronic market organisations, negotiation service providers, or e-commerce sales systems, for the design and implementation of electronic negotiation support. One deliverable of this project, the communication design meta-model for the specification of the common object syntax and semantics in a negotiation is presented in this paper.

After referring to theoretical foundations of this work in the introductory section, the approach chosen for SILKROAD will be illustrated in more detail in Section 2. The meta-model stages of conceptual and integrated communication design are outlined in Sections 3 and 4. The consecutive generation of XML schemata is then presented in Section 5, whereas the usage of these schemata at runtime is the subject of Section 6. Lastly, conclusions and related work are discussed in Section 7.

The concept of media

In SILKROAD, the notion of media and the media reference model [3] are used to conceptualise electronic negotiations. Media are platforms where the exchange of tangible or intangible objects is coordinated 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 [4].

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 (ENM). The primary goal for the SILKROAD framework is to facilitate the design and implementation of electronic negotiation media.

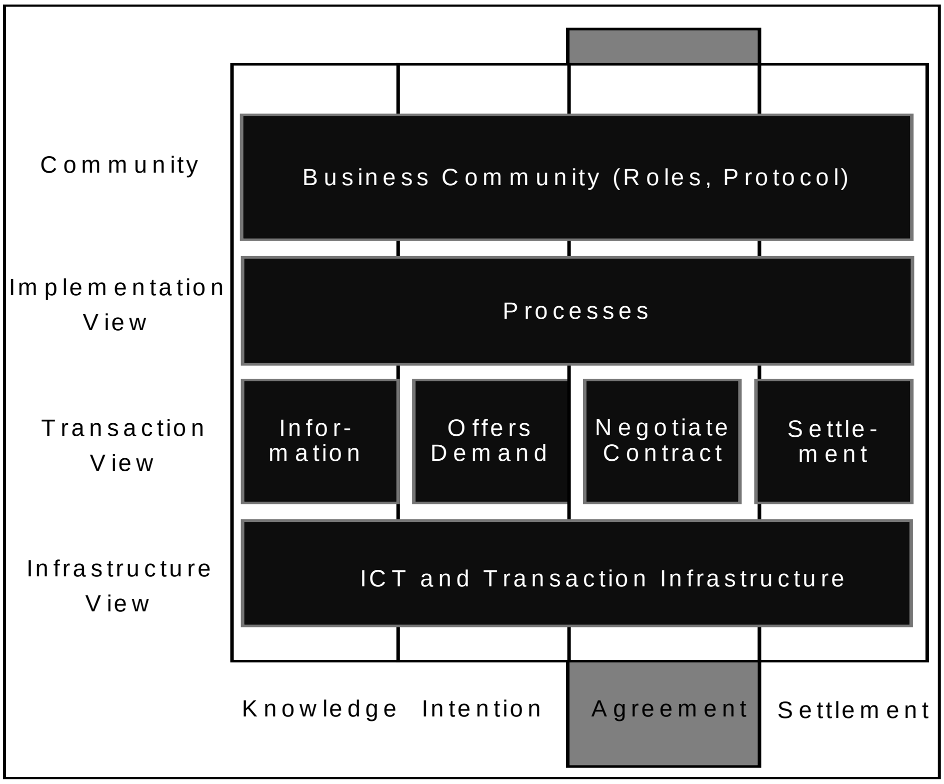

Figure 1: Agreement phase in the media reference model [3].

Media design

Following the concept of media, the design of ENM has to consider three dimensions [5]:

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

The SILKROAD framework aims to support all of the introduced design dimensions. The emphasis in this paper, however, is on the communication design aspects.

2. SILKROAD approach

On a technical level, the SILKROAD framework can be classified as an application framework [6] with a set of reusable service components. But it is more than merely the skeleton of an electronic negotiation medium, which can be customised to specific scenarios. The framework is complemented by a design approach. In SILKROAD, The process of designing an ENM is structured according to an action model, which uses three 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 ENM by providing pre-defined types of entities and interactions. Service component entities explicitly represent the functionality of the underlying negotiations service architecture on a conceptual level (see Section 4).

- Communication design meta-model (CDMM):
  The aim of this meta-model is to assist the design of the logical space of an ENM. It provides the means to express syntactical and semantical specifications compliant to the service architecture functionality.
- Negotiation service architecture framework (NSAF):
  This architecture provides several configurable and interoperable service components that can be used for the implementation of negotiation media that have been conceptualised and designed on the basis of the ODMM and the CDMM.

The SILKROAD action model (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 building a unified, consistent design model for ENM. By emphasising these two distinct design activities, the complexity of the final media IT design and implementation is significantly reduced. On the basis of the integrated design, runtime specifications are generated, which specify and control the integration of the existing architecture framework components. For the communication design these runtime specifications are XML schemata. Organisation and communication design can evolve concurrently – an explicit order is not necessary.

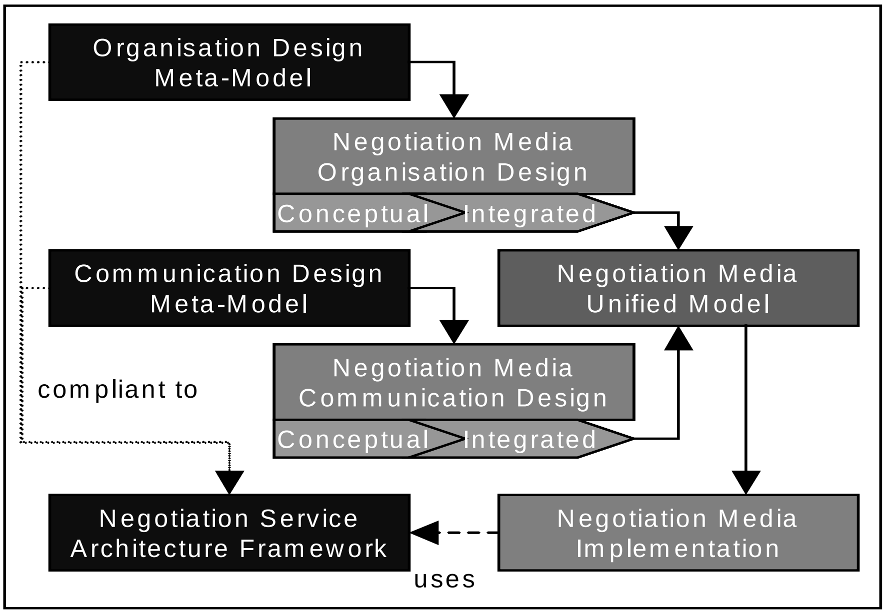

Figure 2: SILKROAD action model.

3. Conceptual communication design

The goal of the conceptual communication design is to structure the logical space of an electronic negotiation medium from a business perspective. The central objects of the communication design are the offers exchanged in a negotiation. Offers are the primary means of communication in the agreement phase (see for example [7]) and in the SILKROAD framework are the only supported type of structured interaction.

In SILKROAD's communication design meta-model the design of offers is separated into the definition of offer ontologies for the semantical aspects, and the specification of offer states for the syntactical aspects of a negotiation.

Offer ontology design

Ontologies are formally specified models of knowledge, which can be used to share semantics among a set of agents. In a negotiation medium the agents are the negotiating parties and the semantics apply to the representation of negotiation objects. An ontology defines the concepts describing a certain domain and the relationships that hold between them [8]. It can be represented as a hierarchy of concepts.

Figure 3 illustrates an (incomplete) example for such a hierarchy of concepts in the domain of computer hardware. A *notebook*, for instance, is a sub-concept of a *computer* and accordingly inherits the properties (attributes) of *computer*, which are in this example the *CPU clock speed*, the type of the *media drive* etc. *Notebooks* are also sub-concepts of *monitors*, thus inheriting another set of properties (e.g. the *display resolution*). Properties in the ontology have a certain type and can be constrained, thus allowing only certain property values (in the example the *CPU clock-speed* is constrained to range between *300* and *1200 MHz*). Relations between concepts complement the ontology. An example for such a relation is that the *CPU* of *notebooks* needs to have *power management functions*. It is possible to infer new knowledge on the basis of given facts. An agent could derive, for instance, that if a certain *CPU* is offered with *notebooks*, it must have *power management functions*. Terms and conditions such as the delivery time or the return policy are also included in the ontology and can be re-used for multiple domains.

The effort to design and establish an ontology for the negotiation medium can be significant, as agents have to agree (in a social process) on this common terminology (see for example [9]). In other words, before an ontology can be used in the agreement phase, the agents have to negotiate on a meta-level the structure, meaning, and content of this ontology – their common language.

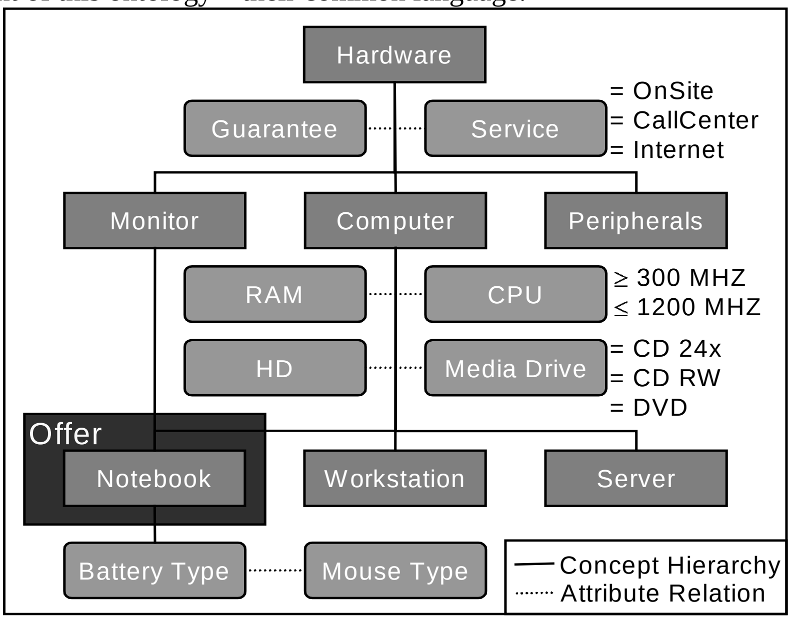

Figure 3: Ontology example.

Offer state design

From a behavioural perspective, an offer can, during the agreement phase, have different states of formality such as *proposal*, *binding offer*, or *contract*. For each offer state the respective level of formality is represented, e.g. by disallowing structural elements for the offer construction.

The basis for the offer state design is a generic offer syntax specification developed for SILKROAD [10]. This syntax defines the notation for a number of structural offer elements such as attribute domains (e.g. *price < $1000*) or evaluation criteria (e.g. *utility[price,$800] = 0.4*).

In the SILKROAD framework, agents, using this notation, construct offers-to-buy or offers-to-sell. Depending on the organisation design of the ENM, agents can or cannot, for instance, counter the offer of another agent by deriving a new offer, which disputes some of the elements of the original offer. With the offer state design, the set of available offer elements for the offer construction or modification process can be controlled.

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, _[1])
- Counters (none, one, many, _)
- Criteria (none, importance, utility, functions)

Details regarding the semantics of these properties can be found in [10]. To give an example, the value *dynamic* for the property *Domains* explicitly allows an agent to define the range of values for a 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 the 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. As more and real world experiences with the set of properties are gained, this scheme will continuously be re-evaluated for completeness and consistency.

For the compressed representation in state diagrams and the consecutive runtime representation generation process, these properties are encoded with two bits each, enabling the expression of complete property set specifications for a certain state of an offer with 12 bits:

Property encoding: $01\ 11\ 10\ 10\ 00\ 10_{bin}$
Property decoding: seller signature
start and end timestamps
attribute values
negotiable constraints
no counters
utility evaluation criteria

Property specification is performed on two levels, *required* offer properties and *optional* offer properties as shown in Figure 4.

Figure 4: Offer state chart example.

In this sample state design for an offer-to-buy, *value ranges* can, but do not have to be specified in the *initial* offer from the buyer. The same is true for *counters* – multiple *counters* are permitted. This changes in the *weighted* state (in Figure 4 changes are indicated with bold fonts). Now *importance ratings* and *utility functions* are required and constraints can optionally be *negotiable* and *weighted*. In the *final* state, *value ranges* are no longer possible and the offer is restricted to optionally one *counter*, whereas in the *signed* state a *seller signature* is required. The same specification is necessary for an offer-to-sell where the states could be, for example, *final*, *matched*, *scored*, and *signed* (see Figure 5).

Conditional expressions for the transitions between the states of an offer are not included on this conceptual level. This is part of the integrated design as illustrated in the next section.

[1] The property value '_' indicates that this value is currently not used

4. Integrated communication design

Two tasks have to be performed in this design activity:

- relating the offer to an ontology domain, and
- linking the offer states to the processes in the negotiation medium organisation design.

For the first task the offer needs to be assigned to one or multiple concepts in the ontology. If a concept (e.g. *computer*) has sub-concepts (*workstation*, *notebook*, etc.), the offer can be issued for any of the sub-concepts as well. This assignment guarantees that the content of the offer can be validated semantically against the specification in the ontology. Hence, only properties related to the concept chosen can be used in the offer description.

In Figure 3, the offer is assigned to the *notebook* concept. Accordingly, only constraints for properties related to *notebook*, such as *display resolution* or *CPU clock-speed* can be used in the offer construction.

The second integration task requires the primary deliverable from the organisation design for the ENM – the agreement scenario. One agreement scenario represents all processes and roles necessary to support in the agreement phase of an electronic transaction. Specific to the SILKROAD organisation design meta-model is, that services of the underlying negotiation service architecture framework, such as the *match* or *score* service can be referenced in the protocol specification and thus, become an integral part of the agreement scenarios designed. Other entity types used for the organisation design are agents, communicative acts, and offers (a detailed description of the ODMM can be found in [11]).

To demonstrate how offer states are linked to the processes in the organisation design, the example introduced in Figure 4 is re-used. Let us assume the excerpt of an agreement scenario in Figure 5 was specified in the organisation design. Agreement scenarios in the ODMM explicitly reference the states defined in the respective offer-to-buy and offer-to-sell state charts. Two states from Figure 4 are referenced (indicated with lighter grey colour). In this example, the buyer (depicted as B^H) submits an offer-to-buy (step 2) in the state *initial* to a *match* service component, which matches it against offers-to-sell in the state *final*. If any of these offers-to-sell are compatible to the offer-to-buy they are returned (step 3) to the buyer in the state *matched*. Looking at the matching results the buyer can then submit in step 4 the original offer-to-buy, in the state *weighted* to a *score* service. It was defined for this state that *importance ratings* and *utilities* are required, whereas *negotiable constraints* are optional. On the basis of these evaluation criteria the *scoring* service calculates aggregated utilities for the matching offers-to-sell and returns an ordered set of scored offers-to-sell back to the buyer (step 5). This process also illustrates how transitions for the offer states are specified in the detailed design. The transition from the *matched* state to the *scored* state of an offer-to-sell is marked by the successful operation of the *score* service.

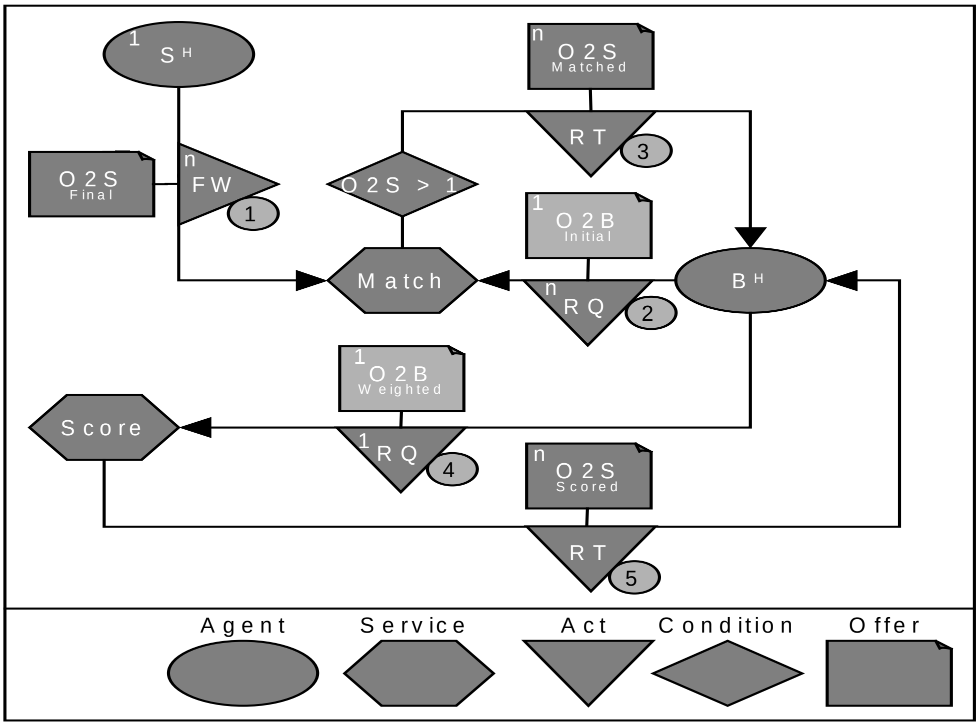

Figure 5: Organisation design example.

The integration of the offer states into the organisation design's agreement scenarios, supports checking the resulting negotiation media model for completeness and accuracy from a structural and behavioural point of view. The subsequent step is the generation of executable representations for this communication design.

5. Generation of XML schemata

This section describes, how the communication design is transferred to XML schemata, which can be used to validate the compliance of actual negotiations to the communication design at runtime.

XML Schema is a new W3C working draft, which was published on April 7, 2000 for review by the public and by the members of the World Wide Web Consortium [12].

Schemata are used to specify classes of XML documents by describing the document structure in a much richer way than is possible on the basis of document type definitions (DTD) [13]. With the basic vocabulary and predefined structuring mechanisms of XML Schema, fine-grained constraints on XML documents can be defined, thus enabling rich automated validation. The primary advantages of using XML schemata compared to DTDs are that it is possible to express hierarchies of data types, and that schemata themselves are XML documents. Hierarchies of types are critical for the schema generation process in SILKROAD as design-specific types are derived from a set of generic types. Due to their XML nature, schemata can be created in the same way (with the same tools) as traditional XML documents. Accordingly, it is not necessary to build an automated schema generation process from scratch.

In SILKROAD, schemata are used to represent the logical space design of electronic negotiations at runtime so that offers submitted to the ENM can be validated for syntactical and semantical correctness. For each offer ontology domain and state, a customised schema is generated. The basis for the generation is the basic SILKROAD syntax. A snippet of this base schema is illustrated in Figure 6.

```
<complexType name="CONTAINER" content="elementOnly">
    <element name="AGENT" type="xsr:AGENT" minOccurs="1" maxOccurs="2"/>
    <element name="OFFER" type="xsr:OFFER" minOccurs="1"
    maxOccurs="1"/>
    <element name="OFFER_CONSTRAINT" type="xsr:CONSTRAINT"
    minOccurs="1" maxOccurs="unbounded"/>
    <element name="COUNTER_CONSTRAINT" type="xsr:CONSTRAINT" minOccurs="0" maxOccurs="unbounded"/>
</complexType>
<complexType name="AGENT" content="empty">
    <attribute name="ID" type="string" use="required"/>
    <attribute name="SIGNATURE" type="string" use="optional"/>
</complexType>
<element name="DOMAIN" type="xsr:DOMAIN" abstract="true"/>
<complexType name="DOMAIN">
    <element name="OPERATOR" type="xsr:OPERATOR" minOccurs="0" maxOccurs="1"/>
</complexType>
```

Figure 6: Base schema.

The base schema defines fundamental constraints such as 'one, but not more than two agents can be associated to an offer' or 'an offer needs to have one or more constraints'. Overall, the base-schema defines all possible offer configurations supported from a structural point of view by the underlying architecture. To generate a state- and ontology-dependent schema, additional constraints are derived from the design specification.

In the next sections, the derivation and customisation mechanism, which underlies the automated schema generation process in SILKROAD, is outlined.

State-dependent customisation

The scenario-specific offer state design leads to a set of additional restrictions to the base schema. To restrict a schema, the following generic XML Schema mechanisms are used in the generation process:

- Deriving types by extension or restriction.
- Changing attribute *use* from *optional* to *required*.
- Forbidding the use of attributes with *prohibited.*
- Assigning *fixed* values to attributes or elements.
- Setting elements to be required (*minOccurs = 1*).
- Limiting the number of elements (*maxOccurs = x*).
- Deleting enumeration elements in simple types.

In the example in Figure 4 the *final* state required offers to have a signature from the agent and a maximum of one countered constraint. In the corresponding *final*-state schema the *agent* and the *container* type are restricted respectively. In Figure 7 *use* for the *SIGNATURE* attribute declaration is *required* and *maxOccurs* for the *COUNTER_CONSTRAINT* is set to “1”.

```
<complexType name="CONTAINER" content="elementOnly">
    <element name="AGENT" type="xsr:AGENT" minOccurs="1"
    maxOccurs="2"/>
    <element name="OFFER" type="xsr:OFFER" minOccurs="1"
    maxOccurs="1"/>
    <element name="OFFER_CONSTRAINT" type="xsr:CONSTRAINT"
    minOccurs="1" maxOccurs="unbounded"/>
    <element name="COUNTER_CONSTRAINT" type="xsr:CONSTRAINT" minOccurs="0" maxOccurs="1"/>
</complexType>
<complexType name="AGENT" content="empty">
    <attribute name="ID" type="string" use="required"/>
    <attribute name="SIGNATURE" type="string" use="required"/>
</complexType>
```

Figure 7: State schema.

Depending on the state properties specification in the conceptual communication design (see Section 0) the same procedure is applied to other types defined in the base-scheme. To restrict, for instance, the domain structure to allow no value ranges, all elements of the *OPERATOR* enumeration for a domain (‘>’, ‘<’ etc.) except the ‘=’ operator are deleted.

The result of this customisation step is the generation of a set of schemata, one for each offer state, defining own namespaces and constraining XML documents from a syntactical perspective. In the next step, semantical constraints are added.

Ontology-dependent customisation

The completed state-dependent schema is included (using the *include schemaLocation* directive in XML Schema) in a new ontology-schema specification (which shares the namespace with the state-dependent schema).

Then, for each (inherited or native) attribute of the ontology concept chosen for the offer, a new type is derived *by extension*, from the *DOMAIN* type in the state schema.

Deriving by extension is comparable to the inheritance mechanism in object-oriented programming languages in the sense that additional elements or attributes can be added or specifications of the super-type can be overwritten. As the *DOMAIN* type is declared to be *abstract* in state schemes, only these new semantic domain types can be used for the actual offer specification.

In Figure 8 examples for two new ontology-dependent types are given. The *CPU* type represents the *CPU* property of the *Computer* concept in Figure 3 with its associated constraint for the clock-speed, which is specified as a range of integers in XML Schema. For the other *DOMAIN* derivation, the *HARDDISK* type, an XML Schema *pattern* is used to define valid values.

In addition to these new types on a domain level, the ontology concept selected for the offer and, if applicable, the related sub-concepts are *derived by restriction* from the state-schema type *CONTAINER*. If offers are constructed according to the ontology-state schema, concept-specific container types such as *NOTEBOOK* in Figure 8 are used as root elements of the XML document. If necessary in the context of the domain, additional restrictions can be applied to these concept types.

```
<element name="DOMAIN_CPU" type="xsr:CPU" equivClass="xsr:DOMAIN"/>
<complexType name="CPU" base="xsr:DOMAIN"derivedBy="extension">
<element name="VALUE" minOccurs="0" maxOccurs="1">
    <simpleType base="integer">
        <minInclusive value="300"/>
        <maxInclusive value="1200"/>
    </simpleType>
</element>
</complexType>
<element name="DOMAIN_HARDDISK" type="xsr:HARDDISK" equivClass="xsr:DOMAIN"/>
<complexType name="HARDDISK" base="xsr:DOMAIN"
 derivedBy="extension">
<element name="VALUE" minOccurs="0" maxOccurs="1">
    <simpleType base="string">
        <pattern value=".*GB"></pattern>
    </simpleType>
</element>
</complexType>
<element name="NOTEBOOK" type="xsr:NOTEBOOK"/>
<complexType name="NOTEBOOK" base="xsr:CONTAINER"
derivedBy="restriction">
    <element name="CONSTRAINT" type="xsr:CONSTRAINT"
    minOccurs="2" maxOccurs="3"/>
</complexType>
```

Figure 8: Ontology-state schema.

With this final customisation step, the ontology-state schema is completely generated and can be used to construct and validate XML offer documents at runtime.

6. Runtime Architecture

Once the design of an electronic negotiation medium has been completed, the runtime representations of the design specifications are generated. This generation process based on state-dependent and ontology-dependent customisation has been outlined for the communication design and its representations, the ontology-state schemata. A corresponding generation process takes place for the organisation design. On the implementation level, the runtime representations are persisted in communication and organisation design repositories as *agreement scenario policies* (see Figure 9).

The facility in the negotiation service architecture framework responsible for controlling the execution of actual agreement scenarios in the ENM is the *policy manager*. It checks, depending on the state of the agreement scenario, offers for semantical and syntactical correctness as well as actions of agents for protocol compatibility. If the actions and offers are compliant to the original design, the policy manager invokes service components in the architecture framework to perform various operations on the offers.

The NSAF features a set of generic customisable service components, which automate typical agreement processes such as *matching* offers or *scoring* offers. The way these services are integrated and invoked in agreement scenarios of the ENM, is specified in the organisation design (see Figure 5).

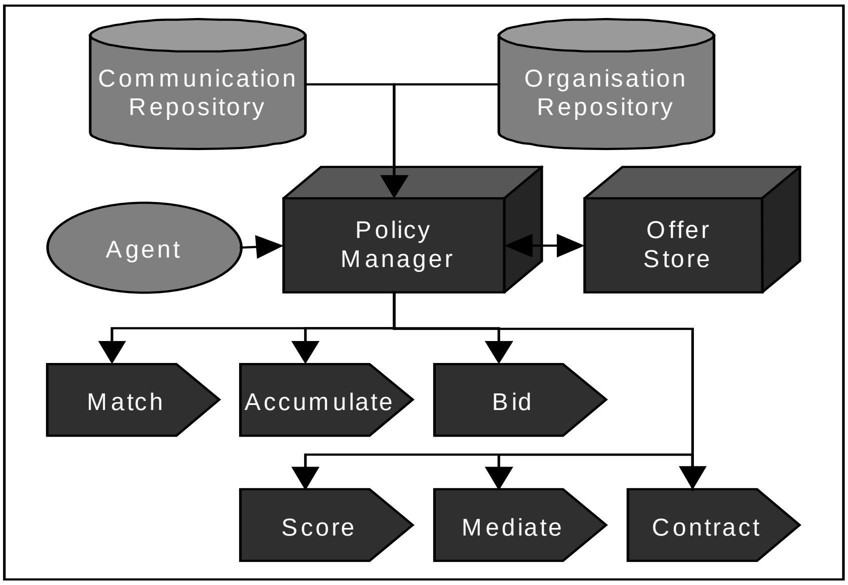

Figure 9: Runtime architecture overview.

7. Conclusions

This paper demonstrated how communication design is performed within the SILKROAD framework. The goal of the communication design is to define the logical space of an electronic negotiation medium – the syntax and semantics of object representations shared by agents, which negotiate the terms and conditions of an electronic transaction. In this final section, the results are evaluated and compared to related research efforts.

Evaluation

Referring back to the initial claims, an evaluation of the proposed approach needs to discuss two interrelated questions:

- Can the ontology problem of electronic negotiations be addressed by the communication design presented?
- Are XML Schema mechanisms useful to express and validate the communication design at runtime?

The result of the explicit communication design of negotiation media within the SILKROAD framework is an ontology for the negotiation domain, and state specifications for offers related to this domain. To achieve a common understanding of the issues that are subject to the negotiation these design deliverables can be specified in a joint process with all agents involved in the later usage of the ENM. This will in many cases stimulate a negotiation about the negotiation – but this agreement among the agents about the semantics of object representations is necessary before a final negotiation support implementation makes sense. This agreement is less important though for the syntactical representation, as the syntax is pre-defined by SILKROAD. The constructs used in the CDMM (ontology definitions and state diagrams) support this meta-level agreement process, as they can be used for communication and discussion on a conceptual level.

Once the communication design has been mutually agreed, it can be transferred to a runtime representation, thus enabling the checking of a negotiation process for semantical and syntactical and correctness according to the original design. Hence, assuming that both, the communication design and the generation of the runtime representation are complete and correct, the ontology problem cannot occur during actual negotiation processes, as violations of the agreed-upon semantics are detected. This is at least true for the agents who originally were involved in the design process. Accordingly the admission for new agents willing to participate in the ENM requires an acknowledgement of the logical space defined.

Whether the generation process and runtime representation is complete and correct depends largely on the mechanisms provided with XML Schema. Various suggestions have been made to move from specific ontology formalisms (KL-ONE, KIF, frame logic…) towards more standardised and widely used representation mechanisms such as UML [8] or XML document type definitions (DTD). The latter approach was chosen by Erdmann and Studer [13]. Evaluating their results, they already point out that transforming ontologies into XML Schema seems to be more appropriate than into DTDs, mainly because of the ability to define type hierarchies. In [14], a process for the stepwise translation of an ontology to XML Schema is proposed. SILKROAD uses the same abstraction-based approach, but in comparison, the communication models do not represent a complete ontology in a schema but only the selected set of concepts. Also, the derivation mechanism is not used for semantical, but for syntactical aspects, which is currently a problem because the relation of properties to concepts is lost if multiple concepts are represented in one ontology-state schema. This might be the case if an agent intends to issue a combinatorial offer for several types of goods (e.g. notebooks and servers). Related to this problem is also the fact that multiple inheritance cannot be represented in XML Schema. This is one of the shortcomings in the current framework, which has to be tackled by future work.

Beyond the complete- and correctness necessary to address the ontology problem, the usage of XML Schema provides additional advantages. As a forthcoming W3C standard, a number of powerful and widely accepted tools such as the Xerces parser [15] can be used to create or validate XML documents adhering to this standard. Hence, agents can easily interface with the ENM by submitting XML documents. These documents can be edited, administered, and validated decentrally according to the internal processes of the agent's organisation. Though this creates a distributed and decentralised system of negotiating agents, common integrity constraints are defined centrally using schemata. XML Schema through the control options for the derivation process, also offers the possibility of extending the ontology in a decentral way. Let us assume, a seller agent can offer computers with new features not reflected in the current ontology such as a DVD writer. The domain schema specification could then be extended by the agent with a derived *media drive* type, which also includes an enumeration for the *DVD write* option. Using this extension functionality enables maintenance of the ontology in a distributed way. But to guarantee the integrity of the overall ontology, the other agents certainly would have to approve such extensions.

Finally, from a technical perspective, the light-weight XML access interface to the negotiation media architecture, which allows for decentralised schema validation and extension can be further extended across all functionalities if, for instance, SOAP (Simple Object Access Protocol, see [15]) is used as a general means of service invocation, which is currently being investigated.

Related work

This approach relates to work in the areas of negotiation support and semi-structured data models. From a negotiation support view, this work is an effort situated in the area of generalised models of negotiations, which is undertaken from an information systems foundation. Most approaches to modelling negotiations to date stem from an artificial intelligence [16] or decision science [17] background. In addition to this, the media concept with its explicit distinction in the communication and organisation design aspects adds a different perspective on negotiation support. This distinction provides an additional level of abstraction and reduces the complexity of negotiation design to some extent.

The idea to support a broad range of agreement scenarios with generic application frameworks is actively researched by the Cosmos project [18] as well as within the TEM effort at the University of Montreal [19]. TEM also addresses modelling aspects, but the focus is on the protocol design, which is performed on the basis of UML state charts with event-condition-action specifications and is represented at runtime in XMI format. The Cosmos project, on the other hand, also investigates possibilities to formalise negotiation strategies for autonomous agents, based, for instance, on genetic algorithms [20]. In contrast to this, SILKROAD does not specifically address means to design or represent negotiation strategies.

Regarding syntax formalisms, related work can be found in the area of XML-based trading protocols such as IOTP [21] or OBI [22]. The difference to SILKROAD is that these protocols are focussed on the settlement phase of electronic transactions (see Figure 1) by providing reference expressions for payment conditions etc. whereas the base-schema in SILKROAD defines generic syntactical structures for the agreement phase, abstracting from the actual message content. From a semantic perspective, related approaches regarding the representation of ontologies have already been discussed in the previous section. The author is aware of one other negotiation support system, which explicitly relies on ontology mechanisms to structure negotiations. It is sketched in [23] – but the description does not disclose how this ontology is created and validated.

References

[1] Beam C., Segev A., Bichler M., Krishnan R. 'On Negotiations and Deal Making in Electronic Markets', *Information Systems Frontier* Vol.1 No.3, 1999, pp.241-258.
[2] Holsapple C., Lai H., Whinston A. 'Negotiation Support Systems: Roots, Progress and Needs', *Journal of Information Systems* Vol.1 1991pp.233-247.
[3] Schmid B. 'Elektronische Märkte - Merkmale, Organisation und Potentiale', in: Sauter M. (ed.), Hermanns A. (ed.): *Handbuch Electronic Commerce*. Universität der Bundeswehr München, July 1998.
[4] Ströbel M. 'The Effects of Electronic Markets on Negotiation Processes', *Proceedings of the ECIS 2000 Conference*, Vienna, Austria, 2000.
[5] Schmid B. 'Was ist neu an der digitalen Ökonomie?', in: *Dienstleistungskompetenz und innovative Geschäftsmodelle*, eds. Belz C., Bieger, Thexis Verlag Universität St. Gallen, 2000, pp.178-196.
[6] Fayad M., Schmidt D., Johnson R. *Building Application Frameworks - Object Oriented Foundations of Framework Design*, Wiley Computer Publishing, New York, 1999.
[7] Kersten G., Noronha S. 'Negotiations in Electronic Commerce: Methodological Misconceptions and a Resolution', *InterNeg Research Report* INR02/99, 1999.
[8] Cranefield S., Purvis M. 'UML as an Ontology Modelling Language', *Proceedings of the IJCAI-99 Workshop on Intelligent Information Integration*, 1999.
[9] Benjamins R., Fensel D., Decker S. ,Perez A. 'Building Ontologies for the Internet: A Mid Term Report', *International Journal of Human Computer Studies*, Vol. 51 1999, pp.687-712.
[10] Ströbel M. 'Intention and Agreement Spaces – A Formalism', *IBM Research Report* No.3279, 2000.
[11] Ströbel M. 'Negotiation Media Organisation Design – Elements of a Reference Model', *IBM Research Report* No.3258, 2000.
[12] Fallside, D. 'XML Schema Part 0: Primer', *W3C Working Draft*, April 7 2000.
[13] Erdmann M., Studer R. 'Ontologies as Conceptual Models for XML Documents', *Proceedings of the 12th Workshop for Knowledge Acquisition, Modeling and Management* (KAW'99), Banff, Canada, October 1999.
[14] Klein M., Fensel D., Harmelen F., Horrocks I. 'The Relation between Ontologies and Schema-Languages: Translating OIL-Specifications to XML-Schema', *Proceedings of the Workshop on Applications of Ontologies and Problem-solving Methods*, 14th European Conference on Artificial Intelligence ECAI'00, Berlin, Germany, August 20-25 2000.
[15] xml.apache.org – visited August 28 2000.
[16] Parsons S., Sierra C., Jennings N. 'Agents that Reason and Negotiate by Arguing', *Journal of Logic Computation*, Vol.8 No.3 1998 pp.261-292.
[17] Kersten G., Szpakowicz S. 'Modelling Business Negotiations for Electronic Commerce', *InterNeg Research Report* INR98/015 1998.
[18] Lamersdorf W., Merz M., Tu T. 'Distributed Systems Technology for Electronic Commerce Applications', *Lecture Notes in Computer Science* Vol. 1521 1998, pp.135-148.
[19] Benyoucef M., Keller R., Lamouroux S., Robert J., Trussar V. 'Towards a Generic E-Negotiation Platform', *Proceedings of the Sixth Int'l Conference on Re-Technologies for Information Systems*, Zurich, February 2000 pp.95-109.
[20] Tu T., Griffel F., Merz M., Lamersdorf W. 'A Plug-in Architecture Providing Dynamic Negotiation Capabilities for Mobile Agents', *Proceedings Second Int'l. Workshop on Mobile Agents*, MA'98, Stuttgart 1998, pp.222-236.
[21] Burdett D. 'Internet Open Trading Protocol - IOTP Version 1.0', IETF TRADE Working Group, Internet Draft 1999.
[22] www.openbuy.org – visited August 28 2000.
[23] Kang J., Lee E. 'A Negotiation Model In Electronic Commerce to Reflect Multiple Transaction Factors and Learning', Proceedings 12th International Conference on Information Networking, Tokyo, Japan January 21-23 1998.