RZ 3279 (# 93325) 10/09/00
Computer Science 15 pages

Research Report

Intention and Agreement Spaces - A Formalism

Michael Ströbel

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

Intention and Agreement Spaces - A Formalism

Michael Ströbel

IBM Research, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Abstract - *Intention spaces can be described as the implicit definition of the kinds of settlements that an agent is in principal willing to accept in order to perform a transaction in an electronic market. This intention space might, for instance, define the desired range of disk sizes or the maximum price for a computer. Explicit offers to buy or sell can then be issued or evaluated on the basis of this intention space, leading to the definition of an agreement space between several agents. The contribution of this paper is a formalism of this implicit notion of an intention and agreement space, allowing not only the explicit specification of offers but also the transformation into XML syntax as well as the definition of service operators for the agreement phase of electronic transactions.*

1. Introduction

Business transactions consist of a finite number of interaction processes among business parties, which can be classified in the following four phases [1]:

- Knowledge (gathering information concerning products, partners etc.)
- Intention (specifying supply and demand)
- Agreement (discussing the terms and conditions of the transaction)
- Settlement (executing the agreed-upon contract)

A negotiation medium can support the agreement phase in electronic markets by providing a platform for agents to negotiate the configuration (terms and conditions) of an electronic transaction. A medium can be described in terms of three main components [2]:

- 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 the agents and protocols specifying their interactions.

According to this media concept, one necessary step for the implementation of digital negotiation media is the design of the logical space. This paper introduces a formalism based on the concept of intention spaces, which can be used for this design task. More specifically, this formalism is proposed for the specification of the syntax within the communication design of a negotiation medium.

Presumably, an agent, before entering the agreement phase of electronic transactions, has some notion of an intention space – either implicit or explicit – based on information obtained in the knowledge phase. This intention space definition is used to specify offers in the intention phase and to evaluate offers in the agreement phase. The intention space definition might be modified in the agreement phase, for instance in the course of negotiations (through concessions…). The sum of all intention spaces of all agents using the negotiation medium represents its logical space.

The organisation design of the digital negotiation medium, the specification of roles and protocols, greatly influences the application of the intention space. In a bilateral trusted negotiation environment elements of the intention space might be communicated directly to the other agent. In a competitive bidding scenario though, the intention space definition could be confidential and only accessible to a trusted third party in order to avoid strategic exploitation by, for instance, competitors. The business model underlying the business transaction greatly influences the design of the protocol, and respectively the usage of the intention space. This paper does not deal with issues related to the organisation design but assumes, that to handle intention spaces in negotiation scenarios in a digital negotiation medium, it is necessary to provide a formalism.

Hence, a formalism for intention spaces on the basis of relational algebra is introduced, so that common negotiation concepts, agreement-spaces, -processes and -services can be defined in a clear and consistent manner. This formalism is rich enough, to express most practical agreement scenarios, but also formal enough to allow the definition and characterisation of operator interfaces or XML representations.

The paper is structured as follows: the fundamental definitions of an intention space, restricted to a static perspective and focused on one single agent, are established in Section 2. How agents come to and evaluate an agreement, the dynamic view of the agreement phase, is then outlined in Section 3. In Section 4, the formalism of an intention space is transformed to an XML syntax, which leads to a discussion of the additional elements necessary to complete a logical space in Section 5, namely semantics and inference. Section 6 demonstrates, how operators on the basis of the formalism can be defined, in order to specify a service architecture for the agreement phase. Finally the results are summarised in Section 7.

2. Fundamentals

During the intention and agreement phase, agents exchange information about the transaction to perform. In this section, this information is formalised to the concept of an intention space. On the basis of this concept, offers and related terms can be defined.

2.1. Intention spaces

A transaction is characterised by a set of attributes. Attribute values are assigned to these attributes with operators, defining the attribute domains.

Naming convention
Agent:
$ag \in AG = \{ag_1, ag_2, \ldots, ag_n)$
Attribute:
$at \in AT = \{at_1, at_2, \ldots, at_n)$
Attribute operator:
$ao \in AO = \{<, \leq, =, \neq, \geq, >\}$
Attribute value:
$av \in AV = \{av_1, av_2, \ldots, av_n\}$

Attribute values can be of different types such as string, integer, real etc. A set of attributes with corresponding attribute values and operators specified by an agent constitutes the elements of a **configuration space** in a four-digit relation. The minimum specification of a configuration space can be defined as follows:

Definition 1
Configuration space:
$CS \subseteq AG \times AT \times AO \times AV$
$CS_{ag} \subseteq ag \times AT \times AO \times AV \mid AT \neq \{\}$
$CS_{ag,at} \subseteq ag \times at \times AO \times AV$

The product/service portfolio that a provider agent is offering or the requirements of a consumer, typically restrict this space. Even in this minimum definition, the set of attribute values, and accordingly the set of operators can be empty. This might, for instance, be the case, if at the beginning of the agreement phase, an agent wants to negotiate what is 'on the table', meaning the scope of the agreement, before negotiating detailed attribute values etc.

The set of possible attribute values for one attribute – the configuration space $CS_{ag,at}$ for one agent and one attribute – specifies the **domain** of an attribute.

Definition 2
Domain:
$DO \subseteq AV_1 \times \ldots \times AV_n$
$DO_{at} = \{AV \mid av \in CS_{ag,at}\}$

Domains can have different characters:

- **Point domain**
  Only one tuple with one attribute value and the '=' operator is part of the intention space.
  Example: (X, colour, =,blue)
- **Discrete domain**
  Several tuples with a '=' operator are defined for one attribute.
  Example: (X, colour, =, blue)
  (X, colour, =, green)
- **Steady domain**
  One or several tuples with other operators than '=' are assigned to one attribute:
  Example: (X, RAM, >, 64MB)
  (X, disk, ≥, 6GB)
  (X, disk, ≤, 9GB)
- **Exclusion domain**
  Point and discrete domains become exclusion domains if the '≠' operator is used instead of the '=' operator.
  Example: (X, colour, ≠, blue)
  (X, colour, ≠, green)

In addition to this, domains of the configuration space can be specified to be **dynamic**. This can be used to express that concrete values cannot or are not intended to be defined for this attribute until some knowledge about the properties of the other agent or its intention space are known. A typical example can be found in the insurance industry, where quotes are usually dependent on the age, medical record, driving experience etc.

Definition 3

Dynamic tag

$dy = \#$

Dynamic domains

$DD = DO \times dy$

A typical example for a dynamic domain could look like this: #(Y, price, =,).
Whereas the configuration space denotes *possible* configurations, the **intention space** represents *acceptable* configurations of the transaction from the perspective of one agent. Domains restrict the set of potential values for one attribute and therefore can be denoted unary constraints. Binary constraints express relations between two attributes on the basis of a domain operator. Accordingly, the intention space is defined as the joined finite set of unary and binary **constraints**.

Definition 4

Constraint Operator:

$co \in CO = \{<,\leq,=,\neq,\geq,>\}$

Unary Constraint:

$uc_{at} = DO_{at}$

Binary Constraint:

$BC_{at1,at2} \subseteq (ag,at_1) \times CO \times (ag,at_2)$

$BC_{at1,at2} \subseteq (ag,at_1,AO,AV) \times (ag,at_2,AO,AV)$

Intention space:

$IS_{ag} = UC \cup BC$

In the following example, the first tuple illustrates acceptable attribute value combinations for two attributes. In the second example the agent specifies, that the payment should take place after the delivery of the computer, not before: ((Y,disk,<,6GB),(Y,RAM,>,64MB)), ((Y, payment_time) > (Y, delivery_time)).
The set of constraints restricts the set of all possible combinations of attributes to a smaller set of desirable configurations. Given the constraint-based intention space definition, a standard constraint satisfaction problem (CSP, see [3]) can be defined. The solution to a CSP is an assignment of attribute values to attributes such that none of the constraints is violated[1]. All unary and binary constraints have to be solved. Hence, **transaction configurations** represent solutions to the intention space constraint satisfaction problem:

Definition 5

Transaction configuration:

$TC = \{(ag,at,ao,av) \in IS_{ag} \mid$

$\forall at \in AT : \exists(ag,at,=,AV) \wedge |AV| = 1$

One could argue that alternative attributes in the intention space specification (e.g. graphics accelerator <u>or</u> DVD drive) do not require an attribute value for each attribute. For constraint satisfaction purposes however, such alternatives can be reformulated as valid combinations of value tuples (see the concept of options below).
Before the actual agreement process, an agent can express the willingness to modify the intention space, if, for instance, an agreement does not seem to be possible on the basis of the initial intention space definition. All constraints that define the intention space can be '**negotiable**'. As domains are unary constraints (see above), elements of a domain specification such as the attribute values or operators can be negotiable as

[1] Solving CSPs for non-discrete attribute domains is still very inefficient. Therefore it is often assumed that only discrete domains are used in order to guarantee efficient solutions.

well. Constraints that are tagged ‘negotiable’ are denoted ‘soft’ constraints opposed to the non-negotiable ‘hard’ constraints in the intention space definition.

Definition 6

Negotiable tag

$ng = \sim$

Soft constraints

$SC \subseteq UC \times ng \cup BC \times ng$

Example: (~(Y,disk,<,6GB),(Y,RAM,>,64MB)), ~((Y, payment_time) > (Y, delivery_time))
In this case, not only the disk size of 64MB is negotiable but also the complete binary constraint in the second example, indicating that the agent might also agree to a settlement where the payment takes place before the delivery.
An agent could as well decide to make elements of the intention space dynamically negotiable, during the course of the agreement process, e.g. because of competitive pressures.
What is the difference of negotiable elements to discrete or steady domains? The answer is that in the case of negotiable elements, the agent is willing to make a concession – to agree to a transaction contract outside the initial intention space specification - maybe in turn for another agent’s concession or also just on his/her own behalf.
In summary, the specification of an intention space as introduced so far, can comprise the following elements:

Definition 7

Intention space

$IS = \{UC, BC, SC\}$

In words, the intention space can be defined by the joint set of unary and binary plus the relation indicating which of these constraints are soft constraints. The minimal definition of an intention space is a non-empty set of attributes. All other elements introduced in this section (attribute values, operators…) are optional. This reflects the fact that in practice the scope of negotiations might not be clear at the beginning of the agreement phase.

2.2. Offers

Offers are the central means of communication in the agreement phase. An offer is simply a subset of the intention space of an agent, which is communicated to another agent as a proposed transaction configuration.

Definition 8

Offer:

$OF_{ag} \subseteq IS_{ag}$

It is important to notice that offers do not always have to be communicated to another agent. Services can for example act as intermediaries.
Assuming that offers are exchanged on the basis of an electronic market medium, a platform for buying and selling different goods or services, additional elements to organise the offer exchange are necessary. The offer needs to be of a certain **offer-type** (to-sell or to-buy). Depending on the organisation of the market, it could also be necessary to specify a certain transaction-type or –domain (e.g. ‘personal computers’) to guarantee common semantics (see Section 5). For the purpose of this paper, a categorisation of the type is sufficient:

Definition 9

Offer types:

$TY = \{ts, tb\}$

Offer-to-sell:

$OS_{ag} = OF_{ag} \times ts$

Offer-to-buy:

$OB_{ag} = OF_{ag} \times tb$

As offers are subsets of the intention space, the type of the intention space limits the structure of the offer. If the intention space, for instance, is just minimally specified with a set of attributes, an offer can only be a subset of these attributes and not contain additional attribute values/operators. In another scenario, the intention space definition might comprise soft constraints, the initial offer though only contains a set of attributes without constraints.

2.3. Related terms

A number of additional terms often used in agreement scenarios can also be defined on the basis of the formalism.
The number of attributes in the intention space defines its **dimension**. If there are only two attributes, for instance ‘price’ and ‘quantity’, the intention space is two-dimensional.

Dimension

$DI_{AS} = |AT|, AT \in IS$

A typical example for a one-dimensional intention space is an auction scenario, where, for instance, one seller and multiple buyers have to agree on one single attribute – the price.
Options in the intention space can be defined on the basis of binary constraints if the value for one attribute is paired with a set of two or more alternative values for the other attribute.
Example:
$ct^1_{RAM,mouse}$ = ((Y,RAM, =, 64MB),(Y, mouse, =, track))
$ct^2_{RAM,mouse}$ = ((Y,RAM, =, 64MB),(Y, mouse, =, stick))

Constraints also enable the specification of **dependencies**. In this case the selection of one attribute value for a particular attribute determines the attribute value of another attribute. A typical example is the price – if an agent chooses to increase the size of the hard disk for a computer this will certainly have an impact on the price.
Example:
$ct^1_{disk,price}$ = ((Y, disk, =, 8GB),(Y, price, =, \$1700))
$ct^2_{disk,price}$ = (Y, disk, =, 12MB),(Y, price, =, \$1800)

If ‘price’ is the only attribute defined in the intention space (the intention space is one-dimensional) and the agent defines steady constraint, the boundary is also often referred to as the **reservation price** [7]. For multi-dimensional intention spaces, tv_{min}, the lower boundary of the transaction value range can then be denoted to be a corresponding **reservation value**.
Aspiration levels (desired configuration elements, see for example [8]) are attribute values of negotiable domains. A steady domain could, for example, be defined as follows: (Z, speed, >, 600MHZ). If this domain were tagged ‘negotiable’, agent Z would aspire to get a PC with a processor speed greater than 600MHZ, but would also settle for less if compensated appropriately.
Finally, if several attribute values for one attribute are part of the configuration space, a **preference** of the agent towards the attribute values is defined as the direction of ascending utility. The preference is a transitive order relation:

Preference

$PF \subseteq AV \times AV$

$PF = \{(av_x,av_y) \mid UT_{at}(av_x) > UT_{at}(av_y) \wedge$
$\quad \forall (av_x,av_y),(av_y,av_z) \in PF :$
$\quad UT_{at}(av_x) > UT_{at}(av_z)\}$

This notion of preference relates to just one attribute. In contrast to this, a preference can also be expressed towards a set of attributes, indicating the relative importance of attributes in a negotiation from the perspective of one agent. This alternative notion of preference is, in the formalism presented in this paper, expressed in the attribute weights (see above).

2.4. Reference model aspects

To formalise ‘how’ intention spaces can be specified is just one building block for negotiation media. It is also promising to recommend ‘what’ should be part of an intention space based on the formalisms introduced. This section briefly discusses some additional elements of offers, which are useful typical negotiation scenarios.
An offer can be legally binding in the sense that another agent has only to accept this offer without further involvement of the originator in order to reach a legal contract specifying the execution of the transaction. To formalise legally binding offers, the concept of signatures is used. The function that assigns signatures to agents is injective.

Definition 10

Signature:

$si \in SI$

Agent signatures

$SG \subseteq AG \times SI \mid$

$\forall ag_1, ag_2 \in AG: (ag_1, si_1), (ag_2, si_2) \in SG \rightarrow ag_1 \neq ag_2 \wedge si_1 \neq si_2.$

Binding offer

$BO_{ag} = \{OF_{ag}, sg_{ag}\}$

Another typical characteristic of an offer is that it might be only valid for a certain period of time. This **offer validity** can be expressed with a start and end date, where both dates are optional. This freedom to include start or end dates enables further types of offer specifications:

Definition 11

Start and end dates:

$SD, ED \in DT, \{0, \infty\} \in DT$

Timeframe offer:

$FO = OF_{ag} \times sd \times ed$

Limited offer:

$LO = OF_{ag} \times ed$

Pending offer:

$PO = OF_{ag} \times sd$

The inclusion of this commitment duration enables a variety of negotiation organisation designs comparable to the approach undertaken in [5], where an extension of the contract-net protocol with offers that feature zero-time, finite-time or infinite-time commitments, is investigated.
Beyond these more protocol oriented reference elements, transaction domain specific elements could be added, eventually leading to a reference model for the logical space of negotiation media.

2.5. Summary

Looking back at the definitions in this section, the information that an agent might have represented internally at the beginning of the agreement phase can be summarised as depicted in Figure 1. From the perspective of one agent, an offer is a subset of the intention space, which in turn is a subset of the configuration space.
An offer initially does not necessarily contain a complete transaction configuration as shown in this figure, although this is necessary in the end to find an agreement (see below).

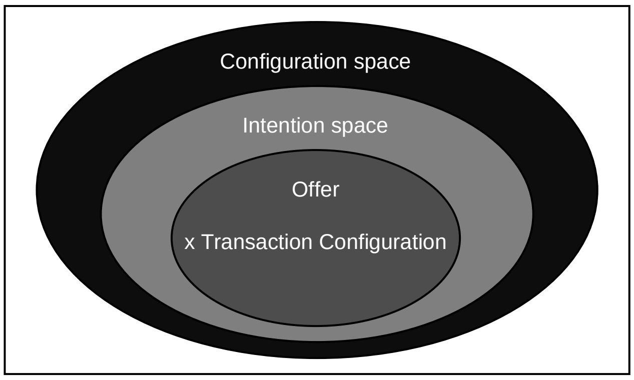

Figure 1: Summary

3. Coming to an agreement

In the previous section, the discussion was focused to the perspective of one agent and limited to a static view. These restrictions are now omitted in order to study the dynamics of the process how two (or more) agents come to an agreement.

Let us assume two agents specify their intention spaces. Even before the exchange of offers, new spaces with different semantics can be determined using the following conceptual matching algorithm:

- join the two intention spaces to a set II of unary and binary constraints
- find within II the set NS of constraints with identical attributes
- find within NS the set AS of the attributes where the intention spaces have overlapping domains
- the set DS is the remainder of NS without AS
- the set CS is the remainder of II without NS

All constraints that remain in CS are part of the **conflict space**. In this space the agents do not even agree on the attributes, which are supposed to be subject to the agreement. The agents do agree on the attributes of the constraints in the **negotiation space** NS. However, in the **disagreement space** DS, which is part of NS, the agents still disagree on the attribute values. Finally, the **agreement space** AS represents elements of transaction configurations to which are compliant to the constraints defined in the intention spaces of the agents at the current state of the negotiation.

Definition 12

Agreement space

$AS_{x,y} = IS_x \cap IS_y,\ x,y \in AG$

Negotiation space

$NS_{x,y} = IS_x \cup IS_y : AT_x = AT_y$

Disagreement space

$DS = NS \setminus AS$

Conflict space

$CS_{x,y} = (IS_x \cup IS_y) \setminus AS_{x,y}$

In practice, it will in most cases be impossible to determine the agreement space in this formal way because the intention space specifications of the agents will either not be complete, or will not be revealed completely. A trusted and objective third party though, could assess the agreement zone for particular agreement scenarios and thereby evaluate the feasibility of a solution.

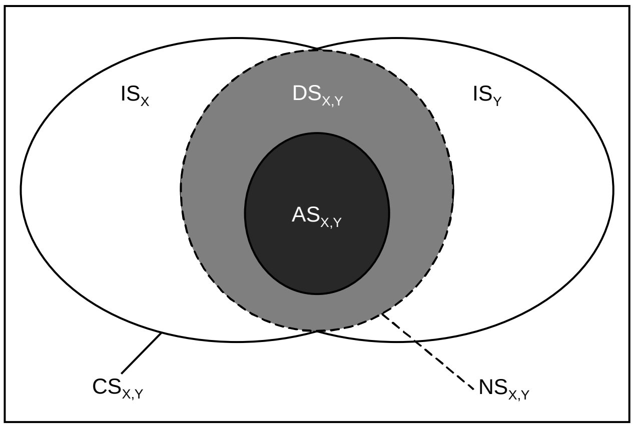

Figure 2: Spaces in the negotiation process

Intention space constraints within the disagreement space DS are often referred to as **issues**. To deal with the complexity of the multi-dimensional disagreement space, agents often fix subspaces by holding several attribute values with the goal to resolve the disagreement on the remaining smaller set of issues one after another [9].

Definition 13

Issues

$IU_{x,y} = \{uc \vee bc \mid uc, bc \in DS_{x,y}\}$

Fixed issues

$FI_{x,y} = uc_{at} \in DS_{x,y} : |DO_{at}| = 1$

Disagreement subspace

$SS_{x,y} = DS_{x,y} \setminus FI_{x,y}$

The agreement space can be empty, thus indicating that based on the current definition of the intention spaces, no agreement is possible. On the other hand, the AS could as well contain all dimensions defined in the respective agents' intention spaces, thus indicating that an agreement is already reached and only a detailed configuration, meaning a selection of one transaction configuration among the set of possible ones, still has to be undertaken.

3.1. Evaluation of agreement spaces

Once the agreement space is determined, an agent will try to evaluate the transaction configurations that are part of the AS in order to reason about the feasibility of an agreement, its potential value and the future negotiation strategy.
The evaluation of an agent through an agent might be based on implicit decision criteria. To support this process it is necessary that these criteria be quantified. This can be achieved on the basis of utility theory (see [4]).
Executing a transaction with a certain transaction configuration returns a **utility** to the agent. Each attribute value, which is part of the final contract, may contribute to this utility. In this abstract sense, the utility function for the domain of an attribute is defined in the following way:

Definition 14

Utility value:

$uv \in UV$

Utility

$UT_{at} \subseteq DO_{at} \times UV$

$UT_{at} = \{(at,av,x) \mid \forall av \in DO_{at}\ x,y \in UV:$
$(at,av,x) \in UT_{at} \wedge (at,av,y) \in UT_{at} \rightarrow y=z\}$

A utility function can be specified as a set of attribute values with associated utilities and a functional expression for the intervals between these points (linear ascending…). For the transaction configuration as a whole, an ‘aggregated utility’ can be defined. Several points in the intention space can return the same transaction value. The **transaction value** function is the weighted sum of all attribute value utilities:

Definition 15

Weight

$w \in W = [0.0 \ldots 1.0]$

Domain weight

$AW \subseteq DO \times W$

$AW_{at} = \{(DO_{at}, w_x) \mid \Sigma_{x=1..|AT|}\ w_x = 1\}$

Transaction value:

$TV_{tc} = \Sigma_{j=1..|AT|}\ (aw_i \times UT_{at}(av_i)) \mid$
$(ag, at_i, =, av_i) \in tc$

A necessary precondition to calculate an aggregated utility is that the single utilities are independent.
For the evaluation of transaction configurations often measures on a higher level than domain utility functions are used. An **evaluation criterion** can be defined as an objective, (e.g. high performance or good quality) for the execution of the transaction. Criteria are linked to attributes of the transaction and accordingly, the value of a criterion is derived from a tuple of related attribute values. The performance criteria for a computer, for example, could depend on the tuple of attribute values for its on-board cache memory, processor speed and bus system.

Definition 16

Criterion

$ct \in CT$

Criterion dependency relationship

$CD_{ct} \subseteq ct \times AT,\ AT \in IS$

Criterion attributes

$CA_{ct} = \{at \mid (ct, at) \in CD_{ct}\}$

Criterion value tuples

$CV_{ct} \subseteq DO_{at1} \times \ldots \times DO_{atn},\ at_{1..n} \in CA_{ct}$

Criterion utility

$CU_{ct} \subseteq CV_{ct} \times UV$

This definition can be illustrated with an example:
$CD_{quality}=\{(quality,at_{guarantee}), (quality,at_{service})\}$
To assess the quality of a notebook, this agent refers to the guarantee and the service provided by the manufacturer. The value tuples are defined as follows:
$CV_{quality}=\{(3years,2yearsOnSite),(2years,1yearOnSite)\}$
To complete the definition of the criterion, these tuples have to be assigned utility values:
$CU_{quality}=\{((3years,2yearsOnSite),1.0),\ldots\}$
In the simplest case, criteria map directly to single attributes with corresponding single-digit value tuples. But a criterion value could also depend on a number of different attribute values or even on a number of other criteria utilities (e.g. in a hierarchical evaluation tree). Using the definition of a set of criteria, the overall transaction value can alternatively be calculated as the weighted sum of the highest-level evaluation criteria utilities.
These additional evaluation criteria can complement the specification of an offer, if the agent has the intention to reveal this information, for instance, to a service which scores and ranks a set of offers automatically (see Section 6) according to the preferences of the agent. Depending on the organisation design of the negotiation medium, the formalisation of evaluation criteria could as well be undertaken only after the agreement space is determined.
Especially in retail markets for commodities the expected transaction value is often fixed. The consumer might have some choices, but the overall transaction value returned from the execution of the transaction is specified to be fixed by the provider (for instance to a certain profit margin) to a certain tv_{fix}. The single utilities of the attribute values might vary, depending on the transaction configuration. But the sum of the utilities remains the same.
Other agents might prefer to have an agreement somewhere within a certain transaction value range, though preferably with a high value. But if a high value is not possible (because no other agent is willing to accept

any transaction configuration with high value) the agent is ready to concede. The range is specified with a maximum and minimum utility value.
In any case, to control the determination of the agreement space from a transaction value perspective it is necessary to include the evaluation criteria in the intention space formalisation.

Definition 17
Intention space with fixed value
$\forall\ tc \in AS_{ag} : tv_{tc} = tv_{fix}$
Intention space with flexible value
$\forall\ tc \in AS_{ag} : tv_{tc} \in [tv_{min}..tv_{max}]$

If an intention space is specified with either static or flexible values, a value assumption for this intention space exists. One has to consider though, that assigning weights and utilities is a complex and error-prone task, which in some cases might even be impossible [6].
The level of disagreement between the agents can be estimated using a measure of opposition that was suggested by Kersten and Soronha [10].

Definition 18
Opposition
$OS_{x,y} = TV_{x,tc}\ x\ TV_{y,tc}$

The opposition expresses the different transaction values (see Definition 15) that agents assign to the various transaction configurations in the agreement space. As an example, Agent X might consider the transaction configuration suggested in the joined agreement space to return a value of 0.8 whereas Agent Y might assess a transaction value of 0.5. $OS_{x,y}$ in this example would be 0.4, the low value indicating strong opposition.
Figure 3 summarises the process of agreement space evaluation.

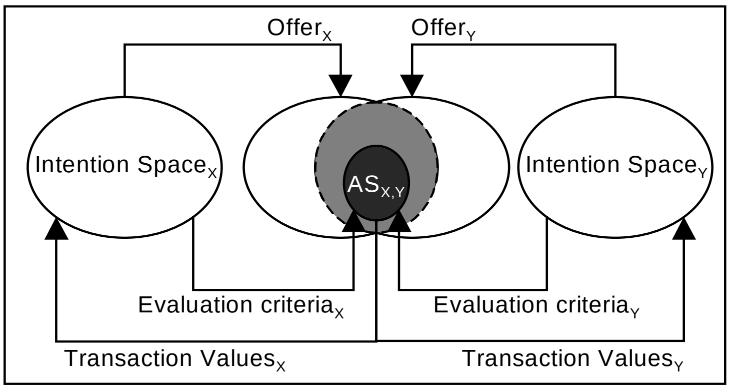

Figure 3: Evaluation of agreement spaces.

3.2. Agreement process

The process of **negotiation** takes place, if (c.f. [11]):

- the agreement space is empty, or does not contain a sufficient number of attributes to specify a transaction configuration that both agents agree to (sufficient condition)
- there is some potential for optimisation of the agreement e.g. by exploiting different valuations in order to reach win-win solutions (sufficient condition, see below)
- the agents have an interest to negotiate (necessary condition)- typically because of mutual dependencies etc..

During the process of negotiation, agents probably have to modify their intention space specifications (aspiration levels, dimensions, dependencies etc.) in order to reach an agreement. Accordingly a negotiation can be defined as a process of changes to the definition of intention spaces (c.f. [12]). The goal of the

negotiation process is to reach an agreement space definition, which contains at least one complete transaction configuration.
When is a transaction configuration complete? A formal and static measure of completeness cannot be defined, as it is the very nature of the negotiation process that agents do not only have to agree on the attribute values, but also on the set of attributes which are part of the agreement. This is reflected in the concept of the negotiation space. Hence, a transaction configuration is complete if both agents are willing to accept this configuration and to execute the transaction accordingly.
A transaction configuration (tc), which lies within the agreement space, can be distinguished from the transaction configuration in Definition 5 in the sense that both agents agree to this tc:

Definition 19
Agreed transaction configurations

$TC_{x,y} = \{tc \mid tc \in AS_{X,Y}\}$

If the concept of signatures is used, both agents can signal their acceptance of an agreed transaction configuration with their signatures. If both agents sign, a legally binding contract for the execution of the transaction can be established.

Definition 20
Contract

$ct_{tc,X,Y} = tc_{X,Y} \times sg_X \times sg_Y$

To further characterise the negotiation process, often the distinction in distributive and integrative negotiations is used. In negotiation support science, this classification is mostly based on the number of issues that are subject to negotiation and the preferences/weights of the agents towards these issues. Kersten and Noronha demonstrated though (see [13]), that single-issue negotiations can be integrative whereas multi-issue negotiations even with differences in the preferences can be distributive.
In the original sense, distributive negotiations are characterised by their win-lose nature - the fact that a gain (increase in transaction value) achieved in the negotiation process for one agent is necessarily a loss for the other agent. Integrative negotiations on the other hand allow the agents to achieve win-win solutions.
A stricter characterisation of negotiations can be achieved on the basis of the formalism introduced (see [14]). **Distributive** negotiations take place, if the agents, on the basis of their initial intention spaces, cannot find any transaction configuration, which returns both agents a higher transaction value. Accordingly, a negotiation process is **integrative** if the agents achieve a mutual higher transaction value.

Definition 21
Distributive negotiations

$\forall tc_1, tc_2 \in AS_{X,Y}$:

$tv_{X,tc1} > tv_{X,tc2} \rightarrow tv_{Y,tc1} < tv_{Y,tc2}$

Integrative negotiations

$\exists\ tc_1 \in AS_{X,Y}\ \forall tc_2 \in AS_{X,Y}$:

$tv_{X,tc1} > tv_{X,tc2} \wedge tv_{Y,tc1} > tv_{Y,tc2}$

To achieve win-win solutions might require changes to the intention space such as adding new dimensions, or changing attribute domains or constraints. Integrative agreements are also possible if the negotiation is multi-dimensional, the agents' weights are not mirrored, and the preferences opposing. This situation allows for tradeoffs that can lead to a higher combined transaction value. In any case, to achieve integrative negotiations it is necessary to reveal preferences to the other agent [15]. This means that an offer has to contain constraints and evaluation criteria, at least to the level of attribute weights.

4. XML representation

For the actual usage of the formalisms introduced, an XML representation is outlined in this section.
It is not the aim of this paper to list the complete XML syntax, namely the DTD (document type definition). In this section, exemplary elements of the representation are shown to demonstrate the transformation of the formalism to the XML grammar. The complete DTD is given in the Appendix.

```
<!ELEMENT CONTAINER ((AGENT)+,OFFER)+>
<!ELEMENT AGENT (AGENT_SIGNATURE?)>
```

These two element specifications map the fact that one or more (indicated by '+') agents can be associated to an offer. Each of these agents can (indicated by '?') sign this offer respectively (see Definition 20).

```
<!ELEMENT CONSTRAINT (DOMAIN | (DOMAIN,
DOMAIN)|(DOMAIN,DOMAIN_OPERATOR,DOMAIN))>
<!ATTLIST CONSTRAINT
 NEGOTIABLE (TRUE | FALSE) "FALSE"
 WEIGHT CDATA #IMPLIED >
```

The CONSTRAINT element represents the concept that a constraint in the intention space formalism can either be a simple domain, or the two types of binary constraints introduced in Definition 4. The specification also illustrates, that a constraint can be specified to be negotiable by setting the NEGOTIABLE attribute to TRUE, rendering the hard constraint into a soft constraint.

```
<!ELEMENT CRITERION ((CRITERION,
CRITERION_WEIGHT)+|((ATTRIBUTE,ATTRIBUTE_OPERATOR,ATTRIBUTE_VALUE)+,UTILITY_V
ALUE)+|UTILITY_FUNCTION+)>
```

The CRITERION example illustrates the hierarchical, recursive nature of a criterion. A criterion such as 'quality of service' can be defined as a weighted set of lower-level criteria. Each of these can again either be another set of criteria or a collection of attribute operators/values assigned to utilities.
To support technical implementations, types for attribute values such as 'float' or 'string' are introduced on the XML level. This also applies to IDs, which can be used in implementations for unique referencing of agents or offers.

```
<!ELEMENT ATTRIBUTE_VALUE EMPTY>
<!ATTLIST ATTRIBUTE_VALUE
 VALUE CDATA #IMPLIED
 TYPE (STRING | INT | FLOAT | DATE)>
```

The XML representation derived from the formalism in this paper defines basic functionality for the communication and processing of intention spaces. The syntax can be applied, for instance, to specify offers. It is possible to introduce additional constructs (such as the type of an agent etc.) on the basis of this fundamental syntax.

5. Inference and semantics

So far, this paper dealt with a syntax specification for the logical space design in a negotiation medium. However, a complete specification for the logical space requires also a semantical level and rules of inference.
It is essential that the agents using the negotiation medium apply the same semantics to the syntactical level of the intention space. If, for instance, an agent X offers a delivery date 'December 12 2000' to agent Y, a problem arises 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 product arrives on the premises of Y.
There are two ways to achieve a common semantical interpretation among the agents: either to provide a shared ontology for all agents or to enable each agent to encode the individual semantics together with the syntax of the communicated intention space.
The effort to establish an ontology for the negotiation medium can be significant, as agents have to agree (in a social process) as well on this common terminology, not only on transaction configurations. In other words, before an ontology can be used in the agreement phase, the agents have to negotiate on a meta-level the structure of this ontology - their common language. The formalism introduced on the syntactical level can be used as a starting point. For the definition of an ontology however, the attribute level has to be

extended to a hierarchy of concepts [16]. Meta-information for attributes in an ontology incorporates typically such notions as domains and types or usage axioms.
The second approach requires means to express individual semantics and translations between those semantics. One proposed solution is the application of Q-Logic, which allows agents to define their own views on a basic set of attributes [17].
Inference can be used to infer new knowledge on the basis of given facts. Regarding the concept of logical space used in this paper, such derivation rules would apply to the ontology complementing the intention space syntax. Using inference in the ontology, an agent could derive, for instance, that if a CPU is compatible with notebooks, it must have power management functions.

6. Agreement operators

In a negotiation medium, the agreement phase can be supported by a number of services. The goal of such an agreement service architecture is to achieve low transaction cost, meaning the cost to coordinate the exchange of products or services, for the agents in the agreement phase. These services are defined as operators that receive inputs and generate outputs. Again, inputs and outputs are related to the concept of intention space and the dynamic perspective of the agreement process introduced in Section 3.
A matching service, for instance, receives two or more intention spaces formalised as offers-to-buy and offers-to-sell and might generate a variety of different outputs (see below). The operation of this service for one pair of an offer-to-buy and offer-to-sell can be described as the process of solving one corresponding constraint satisfaction problem. If a CSP solution (transaction configuration) for this pair of offers can be found, the matching service can, for example, return 'TRUE'. With a black-box view on this service and abstracting from its internal behaviour, the corresponding operator can be formalised as follows:

Definition 22
Matching operator type I
$\text{MATCH} : \text{OB} \times \text{OS} \rightarrow \{\text{TRUE}, \text{FALSE}\}$
$ob_x \text{ MATCH } os_y = \text{TRUE} \mid$
$ds_{x,y} \cup cs_{x,y} = \{\}$
$ob_x \text{ MATCH } ob_y = \text{FALSE} \mid as_{x,y} = \{\}$

Obviously, this is a very simplistic and strict operator definition. A fuzzier match operator could as well return the agreement space instead of TRUE/FALSE. If this agreement space is empty, the match operation failed, otherwise the returned agreement space can be analysed by the agents in order to find out whether the level of agreement is sufficient to sign a contract.

Definition 23
Matching operator type II
$\text{MATCH} : \text{OB} \times \text{OS} \rightarrow \text{AS}$

Another potential variation of this operator could receive several offers-to-sell, which are matched against one offer-to-buy. The result of this operation would then be a set of transaction configurations – the feasible solutions within the respective agreement spaces.

Definition 24
Matching operator type III
$\text{MATCH}: \text{OB} \times \text{OS}_{Y1..Yn} \rightarrow \text{TC}_{X,Y1..Yn}$

Another operator with the potential to decrease transaction cost is a scoring function, which calculates depending on the evaluation criteria of one agent the transaction value (aggregated utility) for a certain transaction configuration.

Definition 25
Scoring operator
$\text{SCORE}: \text{CT} \times \text{CU} \times \text{TC} \rightarrow \text{UV}$

To support agreement scenarios in a negotiation medium, one can combine these two operators in order to score the set of transaction configurations returned by the match operator: SCORE(MATCH(OB_X, OS_{Y1}… OS_{YN})).

7. Summary

This paper demonstrated how the notion of intention space that an agent might implicitly use to generate and evaluate offers in an electronic market, can be made explicit on the basis of a formalism. In a bottom-up approach, the fundamental set of formal definitions is extended to allow the expression of a number of concepts, which are commonly used in the communication between agents in the agreement phase of electronic transactions. The formalism can also be used to classify and describe the nature of the agreement process from a dynamic view.
From a more practical standpoint, real benefits can be achieved if a machine-readable representation of an intention space is derived from the formalism, as illustrated with the XML syntax in this paper. If then operators are developed, which support as services on a technical level the agreement phase, the first meters of the road to a flexible platform for electronic negotiations are paved.

References

[1] Schmid B., Lindemann M., ‘Elements of a reference model for electronic markets’, *Proceedings of the 31st Annual Hawaii International Conference on Systems Science HICCS*, Vol. IV, pp. 193-201, Hawaii, January 6-9 1998.
[2] Schmid B. ‘Elektronische Maerkte - Merkmale, Organisation und Potentiale’, in: Sauter M. (ed.), Hermanns A. (ed.): *Handbuch Electronic Commerce*. Universität der Bundeswehr München, July 1998.
[3] Willmott Steven, Calisti Monique, Faltings Boi, Santiago Macho-Gonzalez, Belakhdar Omar, Torrens Marc ‘CCL: Expressions of Choice in Agent Communication’, The *Fourth International Conference on MultiAgent Systems* (ICMAS-2000), Boston, USA, 2000.
[4] Raiffa H., Keeney R. *Decisions with Multiple Objectives*, Wiley, New York, 1976.
[5] Lee K., Chang Y., Lee J. ‘Time-Bound Negotiation Framework For Electronic Commerce Agents’, *Decision Support Systems*, Vol.28 No.4 2000, pp.319-331.
[6] Krzysztofowicz R., Duckstein L. ‘Assessment Errors in Multiattribute Utility Functions’, *Organizational Behavior and Human Performance*, Vol. 26, 1980, pp. 326-348.
[7] Kersten G., Szpakowicz S. ‘Modelling business negotiations for electronic commerce’, *Interim Report* IR98-015/March, International Institute for Applied System Analysis, 1998.
[8] Kersten G., Szapiro T. ‘Generalized Approach to modeling negotiations’, *European Journal of Operational Research* Vol 26, 1986, pp.142-149.
[9] Raiffa Howard *The art and science of negotiation*, Harvard University Press 1982.
[10] Kersten G., Noronha S. ‘Rational Agents, Contract Curves and Non-Efficient Compromises’, IEEE Systems, Man and Cybernetics, Vol.28, 1998, pp.326-338
[11] Ströbel M. ‘On Auctions as the Negotiation Paradigm of Electronic Markets’, *EM Electronic Markets* Vol.10 No.1 2000.
[12] Kersten G., Szapiro T. ‘Generalized Approach to Modeling Negotiations’, *European Journal of Operational Research*, Vol. 26 1986, pp.142-149.
[13] Kersten G., Noronha S. ‘Negotiations in Electronic Commerce: Methodological Misconceptions and a Resolution’, *Interneg Interim Report* INR02/99, 1999.
[14] Kersten G., Noronha S., Teich J. ‘Are All E-Commerce Negotiations Auctions?’, *Interneg Interim Report* INR08/99, 1999.
[15] Pruitt D., Lewis S., ‘Development of Integrative Solutions in Bilateral Negotiations,” *Journal of Personality and Social Psychology*, Vol.31 No.4, 1975, pp. 621-633.
[16] Benjamins R., Fensel D., Decker S., Perez A. ‘Building Ontologies for the Internet: A Mid Term Report’, International Journal of Human Computer Studies, 51 pp.687-712 1999.
[17] Schmid, B., Geyer G., Wolff W., Schmid R., Stanoevska-Slabeva, K. ‘Representation and Automatic Evaluation of Empirical, Especially Quantitative Knowledge; *Working Report mcm Institute*, University of St.Gallen, Report No: HSG/IWI/SNF-5003-34372/8, 1996.

Appendix

Complete intention space DTD:

```
<!ELEMENT CONTAINER (AGENT+,OFFER)+>
<!ELEMENT AGENT (AGENT_SIGNATURE?)>
<!ATTLIST AGENT
 AGENT_ID ID #REQUIRED
 NAME CDATA #IMPLIED
 URL CDATA #IMPLIED>

<!ELEMENT OFFER (CONSTRAINT+,CRITERION?,(%PRODUCT_DOMAIN;)?)>
<!ATTLIST OFFER
 TYPE CDATA #REQUIRED
 OFFER_ID ID #REQUIRED
 START_DATE CDATA #IMPLIED
 END_DATE CDATA #IMPLIED>

<!ELEMENT AGENT_SIGNATURE EMPTY>
<!ATTLIST AGENT_SIGNATURE
 SIGNATURE CDATA #REQUIRED>

<!ELEMENT CONSTRAINT (DOMAIN|(DOMAIN,DOMAIN)|(DOMAIN,DOMAIN_OPERATOR,DOMAIN))>
<!ATTLIST CONSTRAINT
 NEGOTIABLE (TRUE | FALSE) "TRUE"
 WEIGHT CDATA #IMPLIED>

<!ELEMENT CRITERION
((CRITERION,CRITERION_WEIGHT)+|((ATTRIBUTE,ATTRIBUTE_OPERATOR,ATTRIBUTE_VALUE)+
,UTILITY_VALUE)+|UTILITY_FUNCTION+)>
<!ELEMENT DOMAIN (ATTRIBUTE,ATTRIBUTE_OPERATOR?,ATTRIBUTE_VALUE?)+>
<!ATTLIST DOMAIN
 DYNAMIC (TRUE | FALSE) "TRUE">

<!ELEMENT DOMAIN_OPERATOR EMPTY>
<!ATTLIST DOMAIN_OPERATOR
 OPERATOR (EQUAL | LESS_THAN | GREATER_THAN | UNEQUAL) #IMPLIED>

<!ELEMENT ATTRIBUTE EMPTY>
<!ATTLIST ATTRIBUTE
 NAME CDATA #REQUIRED>

<!ELEMENT ATTRIBUTE_OPERATOR EMPTY>
<!ATTLIST ATTRIBUTE_OPERATOR
 OPERATOR (EQUAL | LESS_THAN | GREATER_THAN | UNEQUAL) #REQUIRED>

<!ELEMENT ATTRIBUTE_VALUE EMPTY>
<!ATTLIST ATTRIBUTE_VALUE
 VALUE CDATA #IMPLIED
 TYPE (STRING | INT | FLOAT | DATE) #IMPLIED>
```