Integration of Bargaining into E-Business Systems

Heinrich C. Mayr
University of Klagenfurt, Department of Business Informatics and Application Systems,
Universitätsstr. 65-67, 9020 Klagenfurt, Austria
E-mail: mayr@ifit.uni-klu.ac.at, http://www.ifi.uni-klu.ac.at/IWAS

Klaus-Dieter Schewe
Massey University, Department of Information Systems & Information Science Research Centre,
Private Bag 11 222, Palmerston North, New Zealand
E-mail: k.d.schewe@massey.ac.nz, http://isrc.massey.ac.nz

Bernhard Thalheim
Christian Albrechts University Kiel, Department of Computer Science and Applied Mathematics,
Olshausenstr. 40, 24098 Kiel, Germany
E-mail: thalheim@is.informatik.uni-kiel.de, http://www.is.informatik.uni-kiel.de

Tatjana Welzer
University of Maribor, Institute of Informatics, Smetanova ul. 17, 2000 Maribor, Slovenia
E-mail: welzer@uni-mb.si, http://lisa.uni-mb.si

Keywords: bargaining, co-design, e-business, games, web information systems

Received: October 24, 2006

Despite the fact that bargaining plays an important role in business communications, it is largely neglected in e-business systems. In this paper a conceptual model that integrates bargaining into web-based e-business systems will be developed starting from an informal characterisation of the bargaining process. Bargaining can be formalised as a two-player game, and integrated with the co-design approach for the design of web information systems. In this way bargaining games are played on parameterised story spaces, such that each move of a player adds constraints to the parameters. Each player follows a strategy for making moves, and winning strategies are characterised by highly-ranked agreements.

Povzetek: Opisano je uvajanje pogajanja v sistem e-poslovanja.

1 Introduction

Bargaining plays an important role in business communications. For instance, in commerce it is common to bargain about prices, discounts, etc., and in banking and insurance bargaining about terms and conditions applies. E-business aims at supporting business with electronic media, in particular web-based systems. These systems support, complement or even replace human labour that would normally be involved in the process. In [10] it has been outlined that such systems can only be developed successfully, if the human communication behaviour is well understood, so that it can become part of an electronic system. Bargaining is part of that communication behaviour.

However, bargaining is largely neglected in e-business. In business-oriented literature, e.g. [6, 13] secure payments and trust are mentioned, but negotiation latitude or bargaining do not appear. Looking at the discussion of technology for e-business this comes as no surprise, as the emphasis is on the sequencing of user actions and the data support, but almost never on inferences. For instance, favourable topics in e-business modelling are business processes [1], workflow [8], e-payment [2], trust [4], decision support [3], or web services [12].

In this paper we make an attempt to integrate bargaining into web-based e-business systems using the co-design approach [11] to the design of web information systems (WISs). We start with a characterisation of the bargaining process as an interaction between at least two parties. The cornerstones of this characterisation are goals, acceptable outcomes, strategies, secrets, trust and distrust, and preferences. We believe that before dropping into formal details of a conceptual model for bargaining, we first need a clearer picture of what we are aiming at. We will discuss the characteristics of bargaining in Section 2 following previous work in [9]. We will also outline the differences to auction systems.

In Section 3 we briefly present parts of the co-design approach to WIS design [11] in order to have a simple conceptual model of WISs, into which ideas concerning bargaining can be implanted. We emphasise the idea of story space as a collection of abstract locations (called scenes) and transitions between them that are initiated by actions, the support of the scenes by database views, and the support of the actions by operations associated with the views. Though many aspects of the co-design approach will be omitted in this model, it will suffice to serve as a basis for a formalisation of bargaining.

In Section 4 we develop a model for bargaining based on games that are played on the conceptual model. This idea was already presented in [9], though only in a completely informal way. We concentrate on bargaining involving only two parties. Their "playground" will be the parameterised story space, and moves consist of adding constraints to the parameters. The moves of the players reflect offers, counteroffers, acceptance and denial. Both players aim at an optimal outcome for themselves, but success is defined as outcomes that are acceptable for both parties. Furthermore, players follow bargaining strategies that may lead them to a final agrement. We will characterise such strategies and attempt to define what a "winning strategy" might be, though obviously bargaining games do not end with one party winning and the other one losing. Furthermore, we characterise the context of bargaining as being defined by user profiles including preferences and desires, and bargaining preferences.

In e-business systems the role of one player will be taken by a user, while the system plays the other role. This may be extended to a multiple-player game with more than one single human player, e.g. if bargaining becomes too critical to leave it exclusively to a system.

2 Characteristics of the Bargaining Process

Let us start looking at human bargaining processes. We consider two typical bargaining situations in a commerce application and a loan application. From these examples we derive characteristic features of bargaining.

2.1 Examples of Bargaining

In a typical commerce situation a customer may enter into bargaining over the total price of an order consisting of several goods, each with its particular quantity. The seller might have indicated a price, but as the order will lead to substantial turnover, he is willing to enter into bargaining. The goal of the purchaser is to reduce the total price as much as possible, i.e. to bargain a maximal discount, while the seller might want to keep the discount below a certain threshold. Both parties may be willing to accept additional items added to the order for free. This defines optimal and acceptable outcomes for both sides.

However, none of the two parties may play completely with open cards, i.e. the seller may try to hide the maximal discount he could offer, while the purchaser may hide the limit price he is willing to accept. Both parties may also try to hide their preferences, e.g. whether an add-on to the order or a discount is really the preferred option. It may even be the case that adding a presumably expensive item to the order is acceptable to the seller, while the latitude for a discount is much smaller, e.g. if the add-on item does not sell very well. So, both parties apply their own strategies to achieve the best outcome for them.

The bargaining process then consists of making offers and counteroffers. Both offers and counteroffers narrow down the possible outcomes. For instance, an offer by the seller indicating a particular discount determines already a maximal price. The purchaser may not be happy with the offer, i.e. the price is not in the set of his/her acceptable outcomes, therefore request a larger discount. Bargaining first moves into the set of mutually acceptable outcomes, finally achieves an agreement, i.e. a contract. Bargaining outside the latitude of either party may jeopardise the whole contract or require that a human agent takes over the bargaining task.

Similar price bargaining arises in situations, when real estate, e.g. a house is sold.

In loan applications, i.e. both personal loans and mortgages [10] the bargaining parties aim at acceptable conditions regarding disagio, interest rate, securities, duration, bail, etc. The principles are the same as for price bargaining, but the customer may bring in evidence of offers from competing financial institutions.

As a loan contract binds the parties for a longer time than a one-off sale, it becomes also important that the bargaining parties trust each other. The bank must be convinced that the customer will be able to repay the loan, and the customer must be convinced that the offer made is reasonable and not an attempt to achieve extortionate conditions. In this case the set of acceptable outcomes is also constrained by law.

Figure 1 illustrates the characteristics of bargaining using a mindmap.

2.2 Formal Ingredients

In order to obtain a conceptual model from these examples let us try to extract the formal ingredients of the bargaining process. From now on we concentrate on the case that only two parties are involved in the bargaining.

First of all there is the object of the bargaining, which can be expressed by a parameterised view. In case of the sales situation this object is the order, which can be formalised by a set of items, each having a quantity, a price, and a discount, plus a discount for the total order. At the beginning of bargaining processes the set contains just the items selected by the customer, and all discounts are set to 0. During the bargaining process items may be added to the order, and discounts may be set. Similarly, in the loan bargaining situation the object is the loan, which is parameterised by interest rate, disagio, and duration, and the set of securities, some of which might belong to bailsmen, in which case the certification of the bailsmen becomes part of the bargaining object.

The set of acceptable outcomes is obtained by instantiations of the bargaining object. These instantiations are

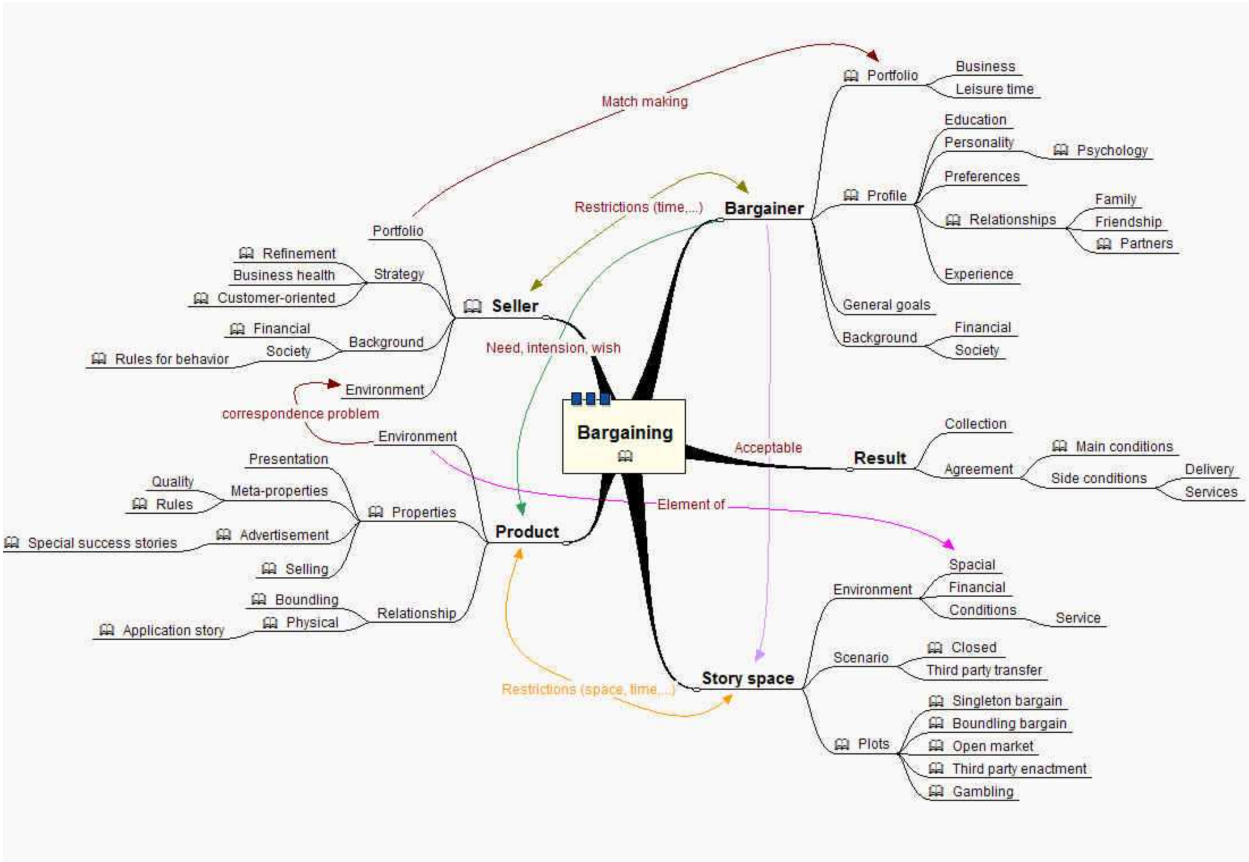

Figure 1: Mindmap for Bargaining Characteristics

expressed by static constraints for each party. However, the constraints are not visible to the other party. They can only be inferred partially during the bargaining process. In addition to the constraints of each party there are general constraints originating from law or other agreed policies. These general constraints are visible to both parties, and they must not be violated.

In case of the sales situation a constraint on the side of the purchaser might be a maximal acceptable price for the original order, or it might be expressed by a minimum discount in terms of any extended order. It may also be the case that the discount is expressed by a function on the set of added items, e.g. the more items are added to the order, the higher the acceptable discount must be. In case of the loan situation constraints on side of the customer can be a maximal load issued by repayments or a maximal value of securities offered. For the bank a minimum level of security and a minimum real interest rate might define their acceptable outcomes.

Within the set of acceptable outcomes of either party the outcomes are (partially) ordered according to preferences. For any artificial party these preferences have to be part of the system specification. For instance, in the sales situation the lower the total price, the better is the outcome for the purchaser (inverse for the seller), and an offer with more additional items is higher ranked. However, whether an offer with additional items and a lower discount is preferred over a large discount, depends on the individual customer and his/her goals.

An agreement is an outcome that is acceptable to both parties. Usually, bargaining terminates with an agreement, alternatively with failure.

The primary goal of each party is to achieve an agreement that is as close as possible to a maximum in the corresponding set of acceptable results. However, bargaining may also involve secondary goals such as binding a customer (for the seller or the bank). These secondary goals influence the bargaining strategy in a way that the opposite party considers offers made to be fair and the agreement not only acceptable, but also satisfactory. This implies that constraints are classified in a way that some stronger constraints define satisfactory outcomes. This can be extended to more than just two levels of outcomes. In general, the bargaining strategy of each party is representable as a set of rules that determine the continuation of the bargaining process in terms of the offers made by the other party.

The bargaining process runs as a sequence of offers and counteroffers started by one party. Thus, in principle bargaining works in the same way as a two-player game such as Chess or Go. Each offer indicates an outcome the offering party is willing to accept. Thus it can be used to reduce the set of acceptable outcomes of the other party. For instance, if the seller offers a discount, then all outcomes with a smaller discount can be neglected. Similarly, if the purchaser offers a price he is willing to pay, the seller can neglect all lower prices.

Nevertheless, there is a significant difference to normal two-player games, as in bargaining there is no direct analogue of the concept of winner. If there is no agreement, both parties lose, and both may consider themselves as winners, if there is an agreement. We may say that a party considers itself the winner, if the agreement is perceived as being better for the own side. Such a characterisation may help to formalise "winning strategies".

Furthermore, each party may indicate acceptable outcomes to the opposite party without offering them. Such playing with open cards indicates trust in the other party, and is usually used as a means for achieving secondary (non-functional) goals. In the following we will not not consider this possibility, i.e. we concentrate on bargaining with maximal hiding.

In summary, we can characterise bargaining by the bargaining object, constraints for each participating party defining acceptable outcomes, partial orders on the respective sets of possible outcomes, and rules defining the bargaining strategy of each party. In the following we will link these ingredients of a bargaining process to the conceptual model of e-business systems that is offered by the co-design method.

Note that bargaining is significantly different from auctioning system. The latter ones, e.g. the eBay system (see `http://www.ebay.com`) offer products, for which interested parties can put in a bid. If there is at least one acceptable bid, usually the highest bid wins. Of course, each bidder follows a particular strategy and it would be challenging to formalise them, but usually systems only play the role of the auctioneer, while the bidders are users of the system.

2.3 Context of Bargaining

In addition to the outlined characteristics of the bargaining process, the attitude towards bargaining depends on a lot of contextual issues. In some cultures bargaining is an intrinsic part of business and is applied with virtually no limits, whereas in other cultures bargaining follows pre-determined rules. Incorporating bargaining into an e-business system has to reflect this spectrum of possible attitudes.

That is, all parties involved in a bargaining process act according to a particular personal profile that captures the general attitude towards bargaining, desires and expectations regarding the outcome of the bargaining process, preferences regarding the outcome and the behaviour of the other parties. For instance, if bargaining is offered in an arabic country, the expected latitude with respect to what can be bargained about and how much the result can deviate from the starting point, etc. must be set rather high. On the other hand, in a European context, bargaining will most likely be limited to rather small margins regarding price discounts, package offers, and preferential customer treatment.

Consequently, we also need an extension of the model of user profiles in [11] to capture the attitude towards bargaining. Correspondingly, the bargaining strategy pursued by the system has to be aware of the user profile. This implies that users have to be informed about the bargaining latitude in case this is rather limited.

3 The Co-Design Approach to Web Information Systems

If bargaining is to become an integral part of e-business systems, we first need a conceptual model for these systems. We follow the co-design approach [11], but we will only emphasise a compact model that can be used to formalise bargaining. We omit everything that deals with quality criteria, expressiveness and complexity, personalisation, adaptivity, presentation, implementation, etc., i.e. we only look at a rough skeleton of the method. In doing so, we concentrate on the story space, the plot, the views, and the operations on the views.

3.1 Story Spaces

On a high level of abstraction we may define each web information system (WIS) – thus also each e-business system – as a set of abstract locations called scenes between which users navigate. Thus, navigation amounts to transitions between scenes. Each such transition is either a simple navigation or results from the execution of an action. In this way we obtain a labelled directed graph with vertices corresponding to scenes and edges to scene transitions. The edges are labelled by action names or `skip`, the latter one indicating that there is no action, but only a simple navigation. This directed graph is called the *story space*.

Definition 3.1. *The* story space $\mathcal{E}$ *consists of a finite set* $\mathcal{S}$ *of* scenes*, an (optional)* start scene $s_0 \in \mathcal{S}$*, an (optional) set of* final scenes $\mathcal{F} \subseteq \mathcal{S}$*, a finite set* $\mathcal{A}$ *of* actions*, a* scene assignment $\sigma : \mathcal{A} \to \mathcal{S}$*, i.e. each action* α *belongs to exactly one scene, and a* scene transition relation $\tau \subseteq \mathcal{S} \times \mathcal{S} \times (\mathcal{A} \cup \{\mathtt{skip}\})$*, i.e. whenever there is a transition from scene* $s_1 \in \mathcal{S}$ *to scene* $s_2 \in \mathcal{S}$*, this transition is associated with an action* $\alpha \in \mathcal{A}$ *with* $\sigma(\alpha) = s_1$ *or with* $\alpha = \mathtt{skip}$*, in which case it is a navigation without action, and we have* $(s_1, s_2, \alpha) \in \tau$*. We write* $\mathcal{E} = (\mathcal{S}, s_0, \mathcal{F}, \mathcal{A}, \sigma, \tau)$*.*

Example 3.1. *Take a simple example as illustrated in Figure 2, where the WIS is used for ordering products. In this case we may define four scenes.*

The scene s_0 = *product contains product descriptions and allow the user to select products. The scene* s_1 = *payment will be used to inform the user about payment*

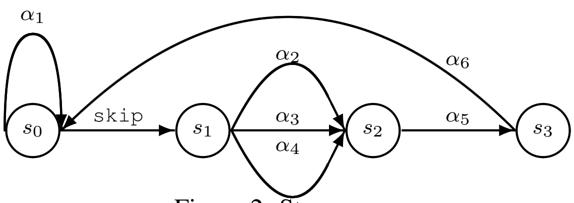

Figure 2: Story space

method options and allow the user to select the appropriate payment method. The scene s_2 = address will be used to require information about the shipping address from the user. Finally, scene s_3 = confirmation will be used to get the user to confirm the order and the payment and shipping details.

There are six actions (their names are sufficient to indicate what they are supposed to do): α_1 = select_product is defined on s_0 and leads to no transition. α_2 = payment_by_card is defined on s_1 and leads to a transition to scene s_2. α_3 = payment_by_bank_transfer is defined on s_1 and leads to a transition to scene s_2. α_4 = payment_by_cheque is defined on s_1 and leads to a transition to scene s_2. α_5 = enter_address and is defined on s_2 and leads to a transition to scene s_3. α_6 = confirm_order, is defined on s_3 and leads to a transition to scene s_0.

In addition to the story space we need a model of the actors, i.e. user types and roles, and the tasks [11], but for our purposes here we omit this part ot the method.

3.2 Plots

With each action we may associate a pre- and a postcondition, both expressed in propositional logic with propositional atoms that describe conditions on the state of the system. In doing so, we may add a more detailed level to the story space describing the flow of action. This can be done using constructors for sequencing, choice, parallelism and iteration in addition to the guards (preconditions) and postguards (postconditions). Using these constructors, we obtain an algebraic expression describing the flow of action, which we call the *plot*. In [11] it has been shown that the underlying algebraic structure is the one of a Kleene algebra with tests [5], and the corresponding equational axioms can be exploited to reason about the story space and the plot on a propositional level, in particular for the purpose of personalisation.

Definition 3.2. *A* Kleene algebra *(KA) $\mathcal{K}$ consists of a carrier-set K containing at least two different elements* 0 *and* 1*, a unary operation *, and two binary operations $+$ and $\cdot$ such that $+$ and $\cdot$ are associative, $+$ is commutative and idempotent with* 0 *as neutral element,* 1 *is a neutral element for $\cdot$, for all $p \in K$ we have $p0 = 0p = 0$, $\cdot$ is distributive over $+$, p^*q ist the least solution x of $q + px \leq x$, and qp^* is the least solution of $q + xp \leq x$, using the partial order $x \leq y \equiv x+y = y$ for the last two properties.*

A Kleene algebra with tests *(KAT) $\mathcal{K}$ consists of a Kleene algebra $(K, +, \cdot, ^*, 0, 1)$, a subset $B \subseteq K$ containing* 0 *and* 1 *and closed under $+$ and $\cdot$, and a unary operation $\bar{}$ on B, such that $(B, +, \cdot, \bar{}, 0, 1)$ forms a Boolean algebra. We write $\mathcal{K} = (K, B, +, \cdot, *, \bar{}, 0, 1)$.*

Then a plot can be formalised by an expression of a KAT that is defined by the story space, i.e. the actions in $\mathcal{A}$ are elements of K, while the propositional atoms become elements of B.

Example 3.2. *Continue Example 3.1. In this case we can define the plot by the expression*

$$(\alpha_1^*(\varphi_1\alpha_2\varphi_2 + \alpha_3\varphi_3 + \alpha_4\varphi_4)\alpha_5(\alpha_6\varphi_5 + 1) + 1)^*$$

using the following conditions. Condition φ_1 = price_in_range expresses that the price of the selected product(s) lies within the range of acceptance of credit card payment. It is a precondition for action α_2. Condition φ_2 = payment_by_credit_card expresses that the user has selected the option to pay by credit card. Analogously, condition φ_3 = payment_by_bank_transfer expresses that the user has selected the option to pay by bank transfer, and condition φ_4 = payment_by_cheque expresses that the user has selected the option to pay by cheque. Condition φ_5 = order_confirmed expresses that the user has confirmed the order.

3.3 Media Types

On a lower level of abstraction we add data support to each scene in form of a media type, which basically is an extended view on some underlying database schema.

Definition 3.3. *A* media type *M consists of a* content data type *$cont(M)$ that may contain pairs $\ell : M'$ with a label ℓ and the name M' of another media type, a defining query q_M defining a view on some database schema, a set of operations, a set of hierarchical versions, a cohesion preorder, style options and some other extensions.*

The database schema, the view formation and the extensions (except operations) are beyond our concern here, so it is sufficient to say that there is a data type associated with each scene such that in each instance of the story space the corresponding value of this type represents the data presented to the user – this is called *media object* in [11]. In terms of the data support the conditions used in the plot are no longer propositional atoms. They can be refined by conditions that can be evaluated on the media objects.

Analogously, the actions of the story space are refined by *operations* on the underlying database, which by means of the views also change the media objects. For our purposes it is not so much important to see how these operations can be specified. It is sufficient to know their parameters.

Example 3.3. *Continue Example 3.1. For simplicity, let the content data type of the media type supporting scene s_0 be defined as { (product_id, product_name, description,*

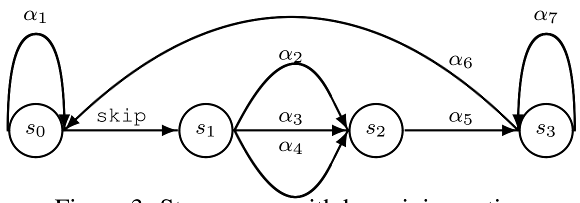

Figure 3: Story space with bargaining action

price) }, i.e. we would present a set of products to a user, each of which defined by an id, a name, a description and a price. Then operation α_1 may take parameters product_id and quantity.

The condition φ_1 from Example 3.2 is to express that the price of the selected products lies within the limit acceptable for credit card payment. If this price limit is a constant L, we obtain the formula

$$\texttt{src}[0, prod \circ (\pi_{price} \times \pi_{quantity}), +] \\ (product \bowtie select\_product) \leq L.$$

Here we exploit that according to the given plot the action select_product will be executed several times, so we can build a relation with the same name collecting the parameters of all executions. Then we can join this relation with the product relation giving us all selected products including their quantity. The structural recursion operation selects price and quantity of each selected product, multiplies them, and adds them all up, which of course defines the total price.

Combining story space, plot and media types, we simply associate with each scene in the story space a data type, replace actions in the story space and the plot by parameterised operations, and replace conditions in the plot by complex formulae as indicated in Example 3.3. The resulting model will be called the *parameterised story space*, which will serve us as the basis for formalising bargaining.

3.4 Bargaining Actions

In our sales example bargaining could come in at any time, but for simplicity let us assume that bargaining is considered to be part of the confirmation process. That is, instead of (or in addition to) the action *confirm_order* we may now have an action $\alpha_7 =$ *bargain_order* as indicated in Figure 3. As before, the action may have a precondition, e.g. that the total price before bargaining is above a certain threshold, or the user belongs to a distinguished group of customers. If the bargaining action can be chosen, it will still result in a confirmed status of the order, i.e. the bargaining object, in the database. However, the way this outcome is achieved is completely different from the way other actions are executed. We will look into this execution model in the next section.

Similarly, in our loan example we find actions select_conditions_and_terms and confirm_loan. Again, if bargaining is possible, the selection of terms and conditions may become subject to a bargaining process, which will lead to an instantiated loan contract in the database – same as without bargaining. As before, the outcome of the bargaining is different from the one without bargaining, and it is obtained in a completely different way.

Therefore, in terms of the story space and the plot there is not much to change. Only some of the actions become *bargaining actions*. The major change is then the way these bargaining actions are refined by operations on the conceptual level of media types.

4 Bargaining as a Game

Let us now look at the specification of bargaining actions in view of the characteristics derived in Section 2. We already remarked that we can consider the bargaining process as a two-player game. Therefore, we want to model bargaining actions as games. There are now two questions that are related with this kind of modelling:

1. What is the ground the game is played on? That is, we merely ask how the game is played, which moves are possible, and how they are represented. This of course has to take care of the history that led to the bargaining situation, the bargaining object, and the constraints on it.

2. How will the players act? This question can only be answered for the system player, while a human player, i.e. a customer, is free in his/her decisions within constraints offered by the system. Nevertheless, we should assume that both sides – if they act reasonably – base their choices on similar grounds. The way players choose their moves will be determined by the order on the set of acceptable outcomes and the bargaining strategy.

4.1 Bargaining Games

An easy answer to the first question could be to choose playing on the parameterised bargaining object, i.e. to consider instances of the corresponding data type. However, this would limit the possible moves in a way that no reconsideration of previous actions that led to the bargaining situation are possible. Therefore, it is better to play on the parameterised story space that we introduced in the previous section.

Each player maintains a set of static constraints on the parameterised story space. These constraints subsume

– general constraints to the bargaining as defined by law and policies;

– constraints determining the acceptable outcomes of the player;

- constraints arising from offers made by the player him/herself – these offers reduce the set of acceptable outcomes;
- constraints arising from offers made by the opponent player – these offers may also reduce the set of acceptable outcomes.

These constraints give rise to definitions of what a bargaining game is, what a state of such a game is, and which moves are possible in this game. We will now introduce these definitions step by step.

Definition 4.1. *A* bargaining game $\mathcal{G}$ *consists of a parameterised story space* $\mathcal{E}$*, a parameterised plot* $\mathcal{P}$*, and three sets* Σ_0*,* Σ_1' *and* Σ_2' *of static constraints on the parameters in* $\mathcal{E}$ *and* $\mathcal{P}$*. We write* $\mathcal{G} = (\mathcal{E}, \mathcal{P}, \Sigma_0, \Sigma_1', \Sigma_2')$*.*

Recall that $\mathcal{E}$ results from the story space as defined in Definition 3.1 by assigning a content data type of a media type to each scene, and by replacing the actions by the corresponding parameterised operations. Similarly, $\mathcal{P}$ results from a KAT expression as defined in Definition 3.2 by replacing atomic actions by the corresponding parameterised operations and propositional atoms by the corresponding formulae on the underlying database schema. Σ_0 formalises legal constraints, while Σ_i' formalises the acceptable outcomes for player i ($i = 1, 2$).

Example 4.1. *Let us look again at our sales example from Example 3.1. Assume that player one is the purchaser. Then a constraint in* Σ_1' *may be that the total price does not exceed a particular limit, which can be formalised by a formula of the form*

$$\texttt{src}[0, \textit{prod} \circ (\pi_{\textit{price}} \times \pi_{\textit{quantity}}), +] \\ (\textit{product} \bowtie \textit{select\_product}) \times (1 - d) \leq M.$$

Here d *indicates a discount, and* M *might be a constant. Alternatively, the purchaser may expect a minimum discount depending on the total nominal price.*

With these constraints each player obtains a set of possible instantiations that are at least acceptable to him/her. The moves of the players just add constraints. This leads to the definitions of states and moves.

Definition 4.2. *A* state *of a bargaining game* $\mathcal{G} = (\mathcal{E}, \mathcal{P}, \Sigma_0, \Sigma_1', \Sigma_2')$ *consists of a partial instance* p *of* $\mathcal{P}$ *with the last action leading to the bargaining scene, and two sets of* Σ_1'' *and* Σ_2'' *of static constraints on the parameters in* $\mathcal{E}$ *and* $\mathcal{P}$*, such that* $\Sigma_0 \cup \Sigma_i' \cup \Sigma_i''$ *are satisfiable (*$i = 1, 2$*). We write* $s = (p, \Sigma_1'', \Sigma_2'')$*.*

Obviously, the *initial state* of the game is determined by the navigation of the user through the story space before reaching the bargaining state. This defines p, while Σ_1'' and Σ_2'' are empty.

Example 4.2. *In our sales example we may have a partial instance of a plot defined by* p = *select_product(*i_4*,5) select_product(*i_7*,2) payment_by_card(…) enter_address(…), which means that the user selected products with id-s* i_4 *and* i_7 *with quantities 5 and 2, respectively, then chose payment by credit card – the omitted parameters would contain credit card number, brand, name of the card and expiry date – and finally entered a shipping address – again parameters omitted. This defines the initial state of the bargaining game.*

At a later stage the purchaser may have indicated to accept a total price m*. This would give rise to the constraint*

$$\texttt{src}[0, \textit{prod} \circ (\pi_{\textit{price}} \times \pi_{\textit{quantity}}), +] \\ (\textit{product} \bowtie \{(i_4, 5), (i_7, 2)\}) \times (1 - d) \geq m$$

in Σ_1''*.*

Definition 4.3. *A* run *of a bargaining game* $\mathcal{G} = (\mathcal{E}, \mathcal{P}, \Sigma_0, \Sigma_1', \Sigma_2')$ *is a sequence* $s_0 \to s_1 \to \cdots \to s_k$ *of states* $s_i = (p_i, \Sigma_{1i}'', \Sigma_{2i}'')$ *satisfying the following properties:*

- s_0 *is the* initial state *of the game.*
- p_{i+1} *is either equal to some* p_j *with* $j \leq i$ *or extends* p_i*.*
- *If* $i + 1$ *is odd, then* $\Sigma_0 \cup \Sigma_1' \cup \Sigma_{1i}'' \cup \Sigma_{2i}''$ *must be satisfiable, and* $\Sigma_{1(i+1)}''$ *extends* Σ_{1i}''*.*
- *If* $i + 1$ *is even, then* $\Sigma_0 \cup \Sigma_2' \cup \Sigma_{2i}'' \cup \Sigma_{1i}''$ *must be satisfiable, and* $\Sigma_{2(i+1)}''$ *extends* Σ_{2i}''*.*

Each transition from s_i *to* s_{i+1} *in a run is called a* move *by player one or two, if* i *is even or odd, respectively.*

So a move by a player is done by presenting an offer. For the player him/herself this offer means to indicate that certain outcomes might be acceptable, while better outcomes are not aimed at any more. This includes that a move may manipulate the bargaining object by extending the partial instance of the plot. However, a player may also reject such a change as proposed by the opponent player. In addition, constraints arising from moves will be added to the constraint sets Σ_i''. For instance, if a seller offers a discount and thus a total price, s/he gives away all outcomes with a higher price. For the opponent player the offer means the same, but the effect on his/her set of acceptable outcomes is different. Moves are only possible as long as the constraints arising from the counteroffers leave the latitude to retain at least one acceptable outcome.

If the set of instantiations reduces to a single element, we obtain an agreement. If it reduces to the empty set, the bargaining has failed.

Definition 4.4. *A run* $s_0 \to s_1 \to \cdots \to s_k$ *is called* successful *iff* $\Sigma_0 \cup \Sigma_1' \cup \Sigma_2' \cup \Sigma_{1i}'' \cup \Sigma_{2i}''$ *is satisfiable, and* $\Sigma_{1i}'' \cup \Sigma_{2i}''$ *is maximal with this property. In this case the instance* p_k *of the plot in state* s_k *is the* agreement *of the bargaining game.*

A bargaining game ends with an agreement, or terminates unsuccessfully, if a player cannot continue making a move.

In addition to "ordinary" moves we may allow moves that represent "last offers". A last offer is an offer indicating that no better one will be made. For instance, a total price offered by a seller as a last offer implies the constraint that the price can only be higher. However, it does not discard other options that may consist in additional items at a bargain price or priority treatment in the future. Thus, last offers add stronger constraints, which may even result the set of acceptable outcomes to become empty, i.e. failure of the bargaining process. Note that this definition of "last offer" differs from tactical play, where players indicate that the offer made is final without really meaning it. Such tactics provide an open challenge for bargaining systems.

4.2 Bargaining Strategies

By making an offer or a last offer, a player makes a move that will result in an acceptable outcome satisfying all constraints arising from counteroffers. In order to make such a choice each player uses a partial order on the set of possible outcomes. Thus, we can model this by a partial order on the set of instances of the parameterised story space. We define it by a logical formula that can be evaluated on pairs of instances.

Definition 4.5. *For a bargaining game* $\mathcal{G} = (\mathcal{E}, \mathcal{P}, \Sigma_0, \Sigma_1', \Sigma_2')$ *the instances satisfying* $\Sigma_0 \cup \Sigma_i'$ *define the* set of acceptable outcomes *of player* i *(*$i = 1, 2$*), denoted as* $\mathcal{O}_i$. *The* preference order *of player* i *is a formula* $\leq_i$ *that can be evaluated on pairs of instances, such that it induces a partial order on* $\mathcal{O}_i$.

Then, whenever a player has to make a move, s/he will choose an offer that is not larger than any previous offer, and not smaller than any of the counteroffers made so far. This defines the reasonable offers a player can make. A *bargaining strategy* consists of rules determining, which offer to choose out of the set of reasonable offers. Simple ad-hoc strategies are the following:

– A *tough bargaining strategy* always chooses a maximal element in the set of reasonable offers with respect to the player's partial order. If successful, a tough strategy may end up with an agreement that is nearly optimal for the player. However, a tough strategy bears the risk of long duration bargaining and last counteroffers.

– A *soft bargaining strategy* is quite the opposite of a tough strategy choosing a minimal element in the set of reasonable offers with respect to the player's partial order. Soft strategies lead to fast agreements, but they almost jump immediately to accepting the first counteroffer.

– A *compromise bargaining strategy* aims at an agreement somewhere in the "middle" of the set of reasonable offers. Such an outcome is assumed to be mutually acceptable. The player therefore chooses an offer that lies between this compromise result and a maximal element in the set of reasonable offers, but usually more closely to the compromise than the maximum.

All these strategies are uninformed, as the only information they use are the constraints on the parameterised story space that amount to the set of reasonable offers. They do not take the counteroffers into account.

An informed bargaining strategy aims at building up a model of what is an acceptable outcome of the opponent player. For instance, if a purchaser only offers global discounts, the strategy of the seller might consist of testing, whether the purchaser would accept an increased quantity or additional items instead. If this is not the case, the seller could continue with a compromise bargaining strategy focusing exclusively on the total price. However, if the purchaser indicates that bargaining about an extended order is a possible option, the strategy might be to first increase the order volume before focussing just on the discount.

Informed bargaining strategies require to build up a model of the opponent player in terms of preference rules, thus they must be built on a heuristic inference engine.

This leads us to the final question of determining a winning strategy. According to our definition, however, bargaining games do not have winners as such. Nevertheless, we can characterise a win by the "normalised distance" of an agreement from an optimal outcome.

Definition 4.6. *An* assessment function *for player* i *is a monotone function* $\nu_i : \mathcal{O}_i \to [0, M_i]$. *An agreement* $a \in \mathcal{O}_i$ *is a* win *for player* i *iff*

$$\frac{M_i - \nu_i(a)}{M_i} \leq \frac{M_j - \nu_i(a)}{M_j}$$

holds.

Then a *winning strategy* for player i is a bargaining strategy that will lead to a win for player i. Note that this includes that the strategy will lead to an agreement.

4.3 Bargaining Context

As indicated in Section 2 we need to incorporate the attitude towards bargaining into user profiles. In order to do so we follow the approach in [11] for modelling such profiles. Thus, assume a set Δ of *dimensions*, e.g. age, gender and cultural context, the problem solving ability, communication skills and computer literacy, the knowledge and education level regarding the task domain, the frequency and intensity of system usage, the experience in working with the system and with associated tasks, etc. For each dimension $\delta \in \Delta$ we have a *domain* $dom(\delta)$.

Definition 4.7. *A* user profile *is an element in* $gr(\Delta) = dom(\delta_1) \times \cdots \times dom(\delta_n)$. *A* user type *over* Δ *is a subset* $U \subseteq gr(\Delta)$.

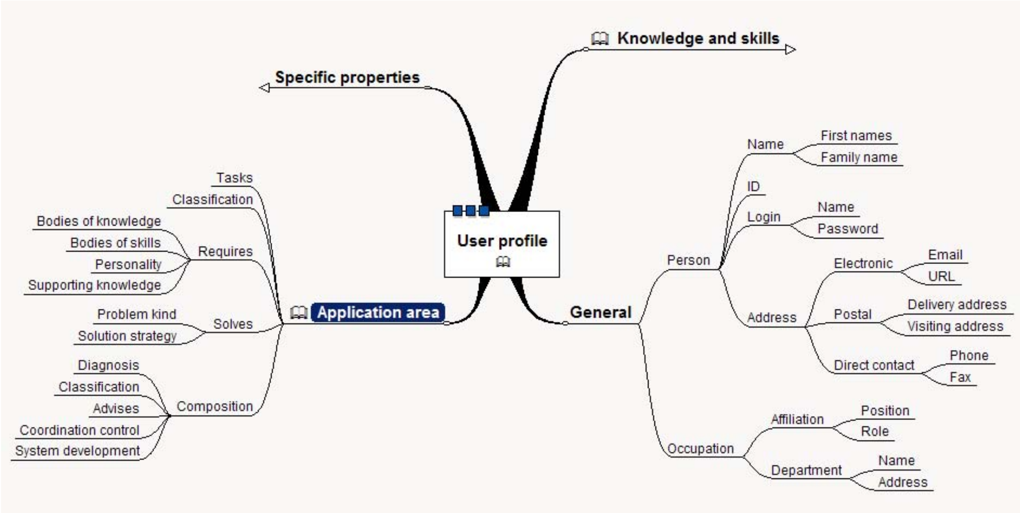

Figure 4: Mindmap of User Profiles in Bargaining

Figure 4 illustrates the dimensions of user profiles that arise in bargaining situations. We emphasise general properties, those that are related to the application area, and knowledge and skills. The latter ones are further illustrated in Figure 5 emphasising application knowledge, problem solving skills and knowledge of technology.

In [11] the purpose of user types has been characterised by the need to associate preference rules with a user type in order to enable the personalisation of a web information system. In principle, this does not change for the case of bargaining. However, the preference rules are no longer restricted to preferences with respect to the selection of actions. For bargaining they have to refer to defining the expected bargaining space, i.e. what can be bargained about, and the expected latitude with respect to bargaining results. Again, both can be modelled by constraints on a bargaining game.

Definition 4.8. *Let* $\mathcal{G} = (\mathcal{E}, \mathcal{P}, \Sigma_0, \Sigma'_1, \Sigma'_2)$ *be a bargaining game. A* bargaining profile *is defined by a pair* $(\Omega, \mho)$ *consisting of sets of static constraints on the parameters in* $\mathcal{E}$ *and* $\mathcal{P}$ *such that* $\models \mho \Rightarrow \Omega$ *holds.*

In a bargaining profile $(\Omega, \mho)$ the first set of constraints models the expectations of a user with respect to what can be put forward in offers, i.e. any offer satisfying the constraints in Ω is considered to be eligible. The second set of constraints expresses the expectations regarding counter-offers, i.e. a user expects that the other party might accept an offer satisfying $\mho$. This explains the request that $\mho$ must imply Ω.

5 Conclusion

We presented a conceptual model for bargaining in e-business systems on the basis of the co-design method [11]. Our model is that of a two-player game, where one part is played by a user, the other one by the e-business system. The game is played on a parameterised specification of the system. The moves of the players represent offers, counteroffers, acceptance and denial. The moves are determined by the characteristics of human bargaining processes such as goals, acceptable outcomes, strategies, secrets, trust and distrust, and preferences.

The work presented so far is only a first step towards a complete conceptual model of bargaining as part of WISs. Our future work aims at completing this model and extending the codesign method correspondingly. This includes extensions covering multi-party bargaining, bargaining with more than one role involved, as well as delegation and authority seeking within bargaining. We believe it will be advantageous to look at defeasible deontic logic [7] for these advanced goals.

References

[1] Bergholtz, M., Jayaweera, P., Johannesson, P., and Wohed, P. Process models and business models - a unified framework. In *Advanced Conceptual Modeling Techniques* (2002), vol. 2784 of *LNCS*, Springer-Verlag, pp. 364–377.

[2] Dai, X., and Grundy, J. C. Three kinds of e-wallets for a netpay micro-payment system. In *Proceedings WISE 2004: Web Information Systems Engineering* (2004), X. Zhou et al., Eds., vol. 3306 of *LNCS*, Springer-Verlag, pp. 66–77.

[3] Hinze, A., and Junmanee, S. Providing recommendations in a mobile tourist information system. In *Information Systems Technology and its Applications* (2005), R. Kaschek, H. C. Mayr, and S. Liddle, Eds.,

vol. P-63 of *Lecture Notes in Informatics*, GI, pp. 86–100.

[4] Jøsang, A., and Pope, S. Semantic constraints for trust transitivity. In *Conceptual Modelling 2005 – Second Asia-Pacific Conference on Conceptual Modelling* (2005), S. Hartmann and M. Stumptner, Eds., vol. 43 of *CRPIT*, Australian Computer Society, pp. 59–68.

[5] Kozen, D. Kleene algebra with tests. *ACM Transactions on Programming Languages and Systems 19*, 3 (1997), 427–443.

[6] Norris, M., and West, S. *eBusiness Essentials*. John Wiley & Sons, Chicester, 2001.

[7] Nute, D. *Defeasible Deontic Logic*. Kluwer Academic Publishers, 1997.

[8] Orriëns, B., Yang, J., and Papazoglou, M. P. A framework for business rule driven web service composition. In *Conceptual Modeling for Novel Application Domains* (2003), vol. 2814 of *LNCS*, Springer-Verlag, pp. 52–64.

[9] Schewe, K.-D. Bargaining in e-business systems. In *Perspectives in Conceptual Modeling*, J. Akoka et al., Eds., vol. 3770 of *LNCS*. Springer-Verlag, 2005, pp. 333–342.

[10] Schewe, K.-D., Kaschek, R., Wallace, C., and Matthews, C. Emphasizing the communication aspects for the successful development of electronic business systems. *Information Systems and E-Business Management 3*, 1 (2005), 71–100.

[11] Schewe, K.-D., and Thalheim, B. Conceptual modelling of web information systems. *Data and Knowledge Engineering 54*, 2 (2005), 147–188.

[12] Simon, C., and Dehnert, J. From business process fragments to workflow definitions. In *Proceedings EMISA 2004* (2004), F. Feltz, A. Oberweis, and B. Otjacques, Eds., vol. P-56 of *Lecture Notes in Informatics*, GI, pp. 95–106.

[13] Whyte, W. S. *Enabling eBusiness*. John Wiley & Sons, Chicester, 2001.

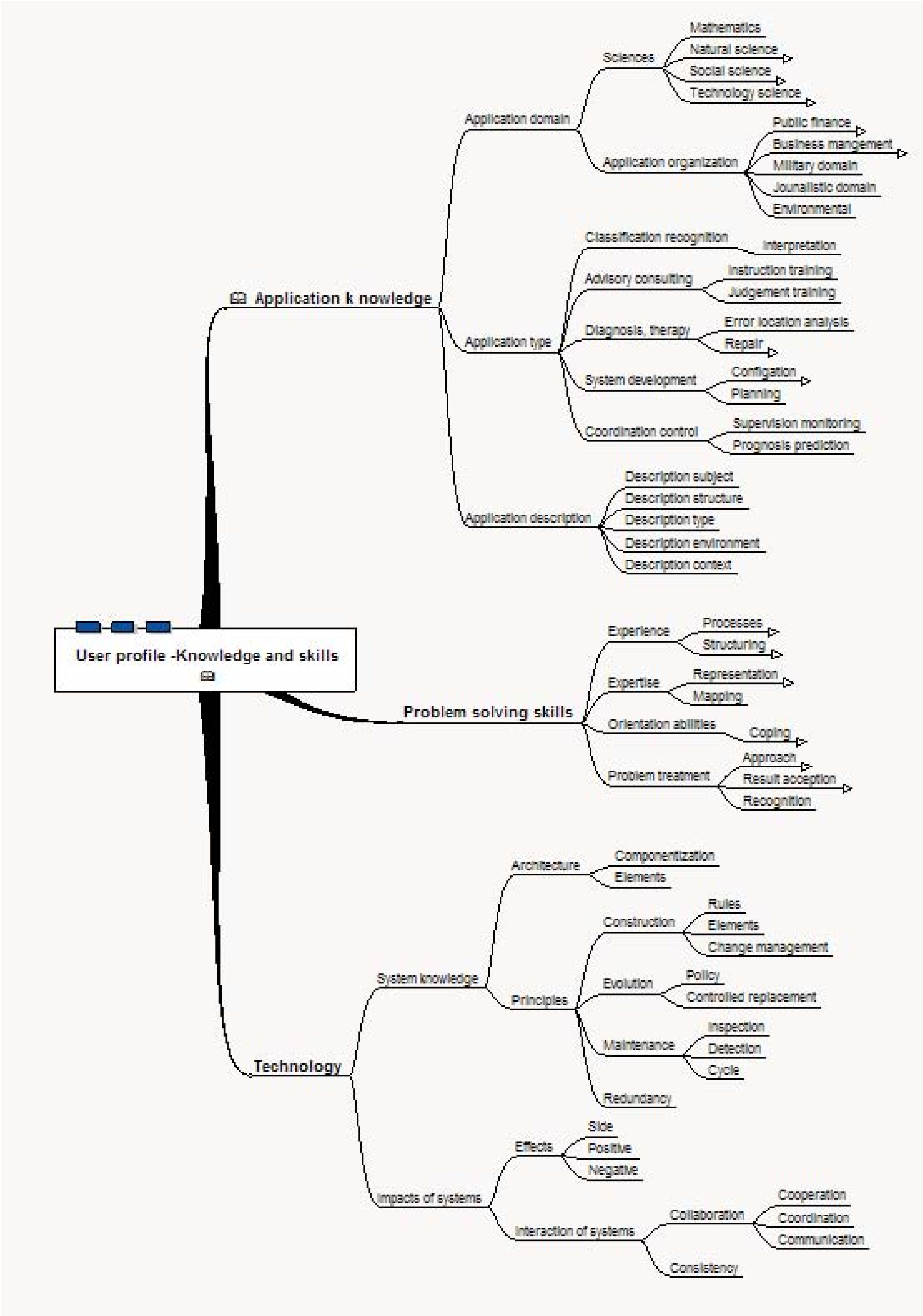

Figure 5: Mindmap of User Knowledge and Skills Profiles in Bargaining