A Framework for Designing Strategies for Trading Agents

Perukrishnen Vytelingum, Rajdeep K. Dash, Minghua He and Nicholas R. Jennings

School of Electronics and Computer Science, University of Southampton,
Southampton, SO17 1BJ, UK.
{pv03r, rkd02r, mh, nrj}@ecs.soton.ac.uk

Abstract

In this paper, we present a novel multi-layered framework for designing strategies for trading agents. The objective of this work is to provide a framework that will assist strategy designers with the different aspects involved in designing a strategy. At present, such strategies are typically designed in an ad-hoc and intuitive manner with little regard for discerning best practice or attaining reuseability in the design process. Given this, our aim is to put such developments on a more systematic engineering footing. After we describe our framework, we then go on to illustrate its use for a particular type of market mechanism (namely the Continuous Double Auction).

1 Introduction

The last decade has seen a significant change in the nature of electronic commerce with the emergence of economic software agents [10]: rational players that are capable of autonomous and flexible actions to achieve their objectives [11] and that are endowed with sophisticated strategies for maximising utility and profit on behalf of their human owners. Today, electronic trading markets[1] allow access to a plenitude of information that enables such software agents to be more informed and respond more efficiently than humans could ever hope to. Now, such trading markets are governed by protocols that define the rules of interaction amongst the economic agents. In some cases, these protocols have a clearly optimal strategy. For example in the Vickrey auction, the best strategy is to reveal one's true valuation of the item [5] and for English auctions it is to bid up to one's true valuation. However, in other settings, the analyses yielding these best strategies often make use of a range of restrictive assumptions; ranging from analysing the market in isolation (i.e. not taking into account dependencies on other related markets), to assumptions on the agent behaviour (such as perfect and complete information availability). Furthermore, several of the standard market mechanisms have been modified or certain complex mechanisms may have been implemented such that an analytical approach cannot yield a best strategy. For example, in eBay auctions[2] (which are multiple English auctions modified with a deadline, proxy bidding and discrete bids) bidding until one's valuation is no longer always the optimal strategy and in Continuous Double Auctions (CDAs) (which are a symmetric auction mechanism with multiple buyers and sellers) there is no known optimal strategy [6].

Given this background, there has been considerable research endeavour in developing trading agents with heuristic strategies that are effective in particular marketplaces [18; 22]. Though more of a black art than an engineering endeavour at present, we believe the design of successful strategies in such marketplaces can nevertheless be viewed as adhering to a fundamental and systematic structure. To this end, in this paper, we provide a general framework for designing strategies which is simple enough to be applicable in a broad range of marketplaces, but modular enough to be used in the design of complex strategic behaviour. We believe such a model is important for the designers of trading agents because it provides a principled approach towards the systematic engineering of such strategies which, in turn, can foster more reliable and robust strategies.

As there is no systematic software engineering framework currently available for designing strategies for trading agents, this paper advances the state of the art by providing the first steps towards such a model. Specifically, our framework is based upon three main principles:

1. An agent requires information about itself and its environment in order to make informed decisions.
2. An agent rarely has full information or sufficient computational resources to manage all the extracted information.
3. Given its limited computational resources and information, an agent needs to employ heuristics in order to formulate a successful strategy.

In more detail, in order to operate in such situations, we advocate a multi-layered design framework. We believe this is

[1] An electronic trading market is here defined as an online institution in which there is an exchange of resources or services using a currency as the trading token. Such markets range from auctions, to supply chains, to barter systems.

[2] www.ebay.com

appropriate because most strategies can be viewed as breaking down the task of bidding into a clear set of well defined sub-tasks (such as gathering relavant information, processing that information and using the information in a meaningful manner). This decomposition can be viewed as a series of (semi-) distinct steps that are handled by different layers. Furthermore, our aim is to ensure our model is sufficiently abstract to be used as the agent model in more general agent-oriented software engineering frameworks, such as Gaia [23] and Agent UML [1]. Now, our framework consists of three layers: the *Information*, *Knowledge* and *Behavioral* layers (hence we term our framework the *IKB* model hereafter). In this context, the information layer records raw data from the market environment. This is then processed by the knowledge layer in order to provide the intelligent data which is used by the behavioral layer to condition the agent's strategy. To illustrate the use of our framework, we consider one of the most popular types of marketplaces, namely the CDA and place a number of the standard CDA strategies within it. The remainder of this paper is structured as follows. We review some of the related work in the field in section 2. Section 3 outlines the IKB model, which is then applied to our trading market example in section 4. Section 5 concludes.

2 Related Work

Much work has been carried out on abstracting the design of electronic markets [12; 15]. However, this work tends to emphasise the methodologies for designing the markets themselves or on proposing new market infrastructures [2; 16; 19]. The systematic design of strategies for agents operating in these markets has, in general, been considered to a lesser extent.

In this latter vein, however, Vetsikas et al [20] proposed a methodology for deciding the strategy of bidding agents participating in simultaneous auctions. Their methodology decomposes the problem into sub-problems that are solved by *partial* or *intermediate* strategies and then they advocate the use of rigorous experimentation to evaluate those strategies to determine the best overall one across all the different auctions. However, their methodology is very much tailored to simultaneous auctions in general and the Trading Agent Competition (TAC) in particular. Thus, it cannot readily be generalised to other auction formats or other market mechanisms. Furthermore, other approaches, including [2; 7] look at the strategic behaviour of agents. However, they avoid issues related to the information and knowledge management aspects of designing trading agents (focusing instead mainly on the strategic behaviour of the strategy).

3 The IKB Model

In this section, we detail the main components that the designer of a trading agent strategy should pay attention to. In so doing, we develop a framework for designing strategies in trading markets. In our model, we have a market $\mathcal{M}$ regulated by its protocol that is predefined. The collection of variables representing the dynamics of the system at time t_k is represented by the state variable $p_{\mathcal{M}}(t_k)$. Within this market, there is a set of trading agents, $\mathcal{I}$, that approach the market through a set of actions which are determined by their strategies. In order to formulate its best strategy, an agent *ideally* needs to know which state it is currently in (agent state), the market state and the actions it can take.

Definition 1 ***Agent's State.*** *An agent i's state, $p_i(t_k)$, at time t_k is a collection of variables describing its resources (computational and economic) and privately known preferences.*

Definition 2 ***Market State.*** *The market state, $p_{\mathcal{M}}(t_k)$, at time t_k is a collection of variables describing all the attributes of the market.*

Definition 3 ***Strategy.*** *A strategy, $\mathcal{S}_i$, for agent $i \in \mathcal{I}$, defines a mapping Γ_i from the history of the agent state $H(p_i(t_{k-1}))$ and the market states $H(p_{\mathcal{M}}(t_{k-1}))$, and the current agent state $p_i(t_k)$ and the market state $p_{\mathcal{M}}(t_k)$ to a set of atomic actions $SA_i = \{a_1^i, a_2^i, \ldots, a_k^i, \ldots\}, a_k^i \in \mathcal{A}_i$ where $\mathcal{A}_i$ is the set of all possible actions for agent i at time t_k.*

The actions chosen by strategy $\mathcal{S}_i$ then affect the external environment such that it causes a change in the market state. In fact, this strategy could interplay with strategies selected by other agents, $\mathcal{I} \setminus i$, as well as some external input(s), ext_n, (where n is the number of external signals not caused by participatory agents) so as to lead the market to the new state.

$$p_{\mathcal{M}}(t_{k+1}) = T(p_{\mathcal{M}}(t_k), H(p_{\mathcal{M}}(t_{k-1})), SA_1, \ldots, SA_{\mathcal{I}}, ext_1, \ldots, ext_n) \tag{1}$$

where $T(.)$ is the state transfer function. From definition 3, it is clear that in order for an agent to know which strategy is best, it should know the complete description and history of the states (all market information), a complete description of all actions available to it, its preferences over the states, a model of its opponents' state, behaviour and preferences, and the state transfer function.

In practice, an agent will typically not have all this information (for a number of reasons, such as limited sensory capabilities, privacy of opponent information and limited knowledge of relevant external signals). Furthermore, an agent's limited computational resources imply that it might not be able to keep a complete history of all past interactions. Given this, there is a need for designing feasible strategies that use limited computational and sensory resources. To this end, we advocate the following design principle in which an agent manages its limited capabilities through its Information Layer (IL), its Knowledge Layer (KL) and its Behavioral Layer (BL) (as shown in figure 1).

In more detail, the Market State (MS) contains public information (i.e. information *available* to all agents in the market) and private/semi-private information (i.e. information *available* to one/some agents). We now provide a description of each of the layers that pertain to the agent:

- **Information Layer.** IL contains data which the agent has extracted from the MS and private information about its own state. This extraction is a filtering process (which we represent as the Information Filter in figure 1) whose objectives are defined by the KL (e.g. filtering out only transaction prices).

- **Knowledge Layer.** KL represents the gathered *knowledge* that is aggregated from the data in IL (e.g. bids submitted in the market). The Behavioral Layer queries the KL to obtain the knowledge it requires.
- **Behavioral Layer.** BL determines the agent's strategic behaviour by deciding on how to use the information available to it in order to interact with the market through a set of actions (e.g. submitting a bid). It queries the KL for the relevant knowledge it requires (e.g the belief that a bid will be accepted in the market).

We believe that when taken together, these three layers provide a sufficient conceptual basis for designing strategies for trading agents in the types of environment we consider. To this end, we next describe each of these layers in further detail, whilst explaining the process through which an agent uses a plethora of raw data in order to select actions which are beneficial to it.

3.1 The Information Layer

This section deals with how an agent gathers information which is then passed on to the KL. The KL will select the data being stored in the IL by modifying the information filter (see figure 1) appropriately. This filter will screen the data from the MS and may also introduce some noise (due to environmental noise or the agent's sensory limitations). As a result, the IL of an agent will contain a noisy, restricted view of all information which it can observe. Furthermore, the IL will also contain information about the agent's state, $p_i(t)$, as well as its action set $\mathcal{A}_i$.

We distinguish between information and knowledge in the following way:

Definition 4 ***Information.*** *Information is raw data that can be sensed by an agent.*

Definition 5 ***Knowledge.*** *Knowledge is the data that is computed by an agent from the information it has gathered.*

Now, information is typically categorised as follows [14]:

- Complete/Incomplete: An agent has complete information if it is aware of the complete structure of the market (that is, its action sets and the result of each action). Otherwise, it has incomplete information.
- Perfect/Imperfect: An agent has perfect information if it is certain of the state it is in, as well as the history of the market's and the agent's states ($H(p_{\mathcal{M}}(t_{k-1}))$ and $H(p_i(t_{k-1})))$ that have led it into this state. Otherwise, it has imperfect information.

As argued in section 1, an agent's sensory and computational limitations imply that it will rarely have perfect and complete information. For example an agent might not be aware of its complete action set (i.e. an agent might believe that its action set at time t_k is $\mathcal{A}'_i \subset \mathcal{A}_i$) or it may be unsure of which state it is in (i.e. it expresses an uncertainty over $p_i(t_k)$). Thus, the agent will need to have certain heuristics in order to guide its search for information. This information can be gathered from public, semi-private and private sources. Public information is observable by all agents ($i \in \mathcal{I}$) in the market and includes things such as the market price in a stock exchange, the minimum increment in an eBay auction and the number of lots of flowers on sale in a Dutch flower auction. Semi-private information is that which is available to a subset of the agents ($i \in \mathcal{J} \subset \mathcal{I}$) and includes things such as the amount that a supplier might require from an agent and the code to signalling actions by a bidder ring in an auction [13]. Private information is only observable by a single agent and includes items such as its budget or the goods it is interested in. Thus, given the required information that the KL has requested, the agent will devote its limited resources to obtaining it. Then having gathered the required information from the market, the agent proceeds to use this information to infer knowledge in the KL.

3.2 The Knowledge Layer

The Knowledge Layer connects the information and the behavioral layers (see Section 3.3). It infers knowledge from the information sensed by the agent and passes it to the BL which acts upon it. In order to do so, the KL is first requested by the BL as to which knowledge to acquire. This knowledge could be, for example, the current Sharpe ratio[3] of a stock or a prediction of the market price based on a particular prediction model. Based on this and the current knowledge of the agent's state, the KL will decide upon the information it requires and set the information filter accordingly. The KL will then use the input from the IL so as to to infer the appropriate knowledge which it will output to the BL.

Mirroring the IL, the KL can be segmented into knowledge about the agent's and the market's state. The former is what the agent knows about itself. This includes knowledge pertaining to its subgoals (such as its risk attitude or the deadline by which a good is to be delivered) and knowledge about its state $p_i(t_k)$. The latter is what the agent knows about the market and would include items such as the degree of competitiveness in the market, the opponents' state and any available market indicators.

3.3 The Behavioral Layer

The Behavioral Layer (BL) represents the decision-making component of the strategy. The intrinsic idea behind strategies is related to finding the optimal action[4] in the market. However, as outlined earlier, more often than not, there is no known optimal action, as the market is too complex and the set of actions too large to determine such an optimal action analytically. Then, as there is no best strategy, a heuristic approach is taken. Thus, the BL instructs the KL as to what knowledge it needs to gather from the market which, as described in subsection 3.2, is computed from the market information. With the relevant knowledge of the market and its goals, the agent i forms a decision based on its strategy $\mathcal{S}_i$ and interacts with the market through actions SA_i. The goal of an agent's strategy is typically profit-maximisation, with the more sophisticated strategies considering both short-term and long-term risk. The formulation of the strategy usually depends on such goals and the market protocols.

[3]The Sharpe ratio is a measure of a stock's excess return relative to the total variability of the stock [17].

[4]Optimal in this case means the agent's most profitable action, given the current market conditions.

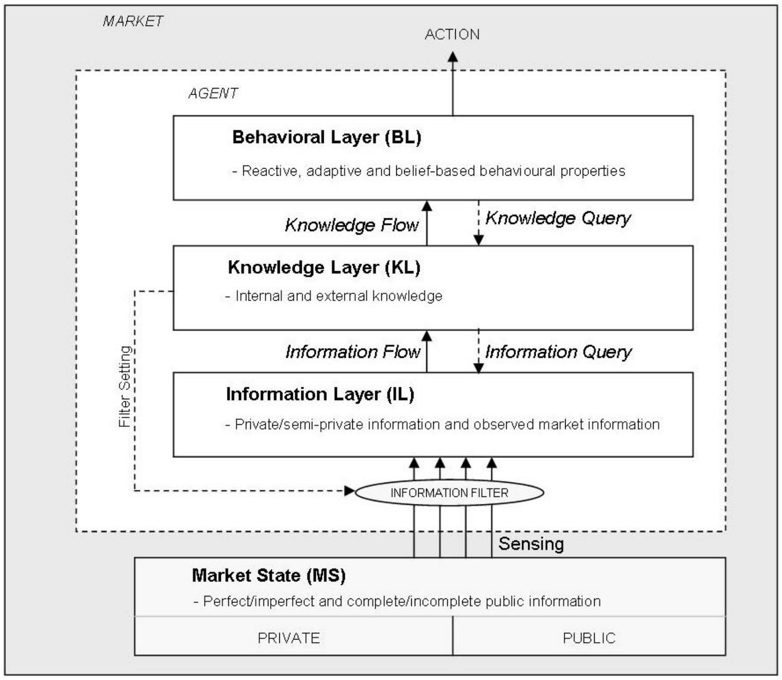

Figure 1: Structure of the IKB Model

Given this insight, it is possible to categorise the different behavioral properties of the strategy into different levels. We distinguish those strategies in terms of whether they use a history of market information or not, and, where they consider external information or not.

1. **No History** (ignores $H(p_{\mathcal{M}}(t_{k-1}))$ from equation 1). Thus, such reactive strategies make myopic decisions based only on current market conditions, $p_{\mathcal{M}}(t_k)$. The myopic nature of these strategies imply a lower workload on the KL since they require less information to sense and process. Such reactive strategies also usually exploit the more complex bargaining behaviour of competing strategies and thus require less computational resources to strategise. One example of such a strategy is the *eSnipe* strategy[5] which is frequently used on Ebay to submit an offer to buy near the end of the auction.

2. **History** (considers $H(p_{\mathcal{M}}(t_{k-1})$ in equation 1). We further subdivide those strategies that use a history of market information as being predictive or not (i.e. whether they predict $\{p_{\mathcal{M}}(t_{k+1}), p_{\mathcal{M}}(t_{k+2}), \ldots\}$ or not). The non-predictive strategies typically use $H(p_{\mathcal{M}}(t_{k-1}))$ to estimate $p_{\mathcal{M}}(t_k)$.

   (a) *Non-predictive*: The non-predictive strategy is typically belief-based and forms a decision based on some belief of *the current market conditions*. The agent's belief is computed from the history of market information in the KL, and usually represents the belief that a particular action will benefit the agent in the market (for example an offer to buy that is accepted). Given its belief over a set of actions, the agent then determines the best action over the short or long term.

   (b) *Predictive*: A strategy makes a prediction about the market state in order to adapt to it. Now, because future market conditions (that the trading agent adapts to) cannot be known *a priori*, the adaptive strategy typically makes some prediction using the history of market information. The KL is required to keep track of how the market (knowledge) is changing to predict the future market, while the BL uses this knowledge about the market dynamics to improve its response in the market. Being adaptive is particularly important in situations where the environment is subject to significant changes. By tracking such changes and adapting its behaviour accordingly, the agent aims to remain competitive in changing market conditions.

3. **No External Information** (ignores $ext_1, \ldots, ext_n$ in equation 1). In this case, the strategy does not consider any signals external to the market (e.g. the falling market price of a good affecting the client's preferences for another type of good in an auction). However, the agent can choose whether or not to use the (internal) information (e.g. the e-Snipe strategy uses the internal market information, while the ZI Strategy [9] in the CDA does not make use of any market information).

4. **External Information** (considers $ext_1, \ldots, ext_n$ in equation 1). It is possible that signals external to the

[5] www.esnipe.com

market can influence the preferences of the participants, such as an event independent of the market causing the clients' preferences in the market to change (e.g. unforseen weather conditions affecting the production of wheat and thus the market for wheat indirectly). Thus, external information can be a valuable source of information that the agent can use to strategise in the market.

Having presented our IKB model for designing trading strategies, we now consider an example of a market mechanism that has spawned a gamut of strategies, and discuss how our model can be applied to it.

4 Applying IKB to the CDA

The CDA is a symmetric auction with multiple buyers and sellers and presently is one of the most popular auction formats in marketplaces populated by autonomous software agents. In CDAs, traders are allowed to submit offers to buy (bids) or to sell (asks) at any time during the trading day. There is an outstanding bid (ask) which is the highest bid (lowest ask) submitted in the market at any time during the CDA. Furthermore, the market clears continuously whenever a bid can be matched to an ask. Such CDAs are widely used, indeed they are the principal financial institution for trading securities and financial instruments (e.g. the NYSE and the NASDAQ both run variants of the CDA institution). Because there is no known dominant strategy in the CDA, several researchers have worked on competing alternatives [3; 8; 21], developing trading agents that can outperform humans in experimental settings [4]. We now give a formalised definition of the single-unit CDA institution, whose market state at time t_k is $p_{\mathcal{M}}(t_k) =< g, \mathcal{B}, \mathcal{S}, price(t_k), bid(t_k), ask(t_k) >$ where

1. g is the good being auctioned off.
2. $\mathcal{B} = b_1, \ldots, b_{nb}$ is the finite set of identifiers of bidders in the market, where nb is the number of current bidders.
3. $\mathcal{S} = s_1, \ldots, s_{ns}$ is the finite set of identifiers of sellers in the market, where ns is the number of current sellers.
4. $price(t_k)$ denotes the current market price of good g in the market. This corresponds to the most recent transaction price.
5. $bid(t_k)$ denotes the outstanding bid at time t_k.
6. $ask(t_k)$ denotes the outstanding ask at time t_k.

The agent state at time t_k, $p_i(t_k) =< id_i, n_i(t_k), \boldsymbol{v}_i = (v_{i1}, \ldots, v_{n(t_k)}), budget_i(t_k), comp_i(t_k) >$ where:

1. id_i defines the identity of the agent as either a buyer or a seller agent.
2. $n_i(t_k)$ defines the number of items an agent is currently interested in either buying or selling.
3. $\boldsymbol{v}_i = \{v_{1,i}, \ldots, v_{n_i(t_k),i}\}$ is the set of limit prices [6] ordered from highest to lowest in the case of a bidder and vice versa in the case of a seller.
4. $budget_i(t_k)$ is the budget available to agent i.
5. $comp_i(t_k)$ is the computational resources (memory and processing power) available currently to agent i.

The action set of the agent depends on its identity, id_i. If it is a buyer, it has $\mathcal{A}_i =< bid_i, silent >$ where $bid_i \in Re^+$ and $silent$ is no bid. Correspondingly, if it is a seller its action set is $\mathcal{A}_i =< ask_i, silent >$ where $ask_i \in Re^+$. It should be noted that in the CDA, SA_i will only be singletons (i.e. an agent can only take a single action at a time). The state transfer function T_{CDA} in the CDA is the rules for acceptance and rejection of bids and asks as well as the clearing rules (see below). The standard CDA is not influenced by external signals (i.e. the transfer function T_{CDA} has no $ext_1, \ldots, ext_n$ arguments[7]) and the market changes each time an agent submits a bid or an ask and thus simultaneous bidding does not occur. Thus $p_{\mathcal{M}}(t_{k+1}) = T_{CDA}(p_{\mathcal{M}}(t_k), H(p_{\mathcal{M}}(t_{k-1})), SA_i)$ whereby $T(.)$ is defined by the following rules:

- if $SA_i = bid_i$, then
  - if $bid_i < bid(t_k)$ then bid_i is rejected and $p_{\mathcal{M}}(t_{k+1}) = p_{\mathcal{M}}(t_k)$.
  - if $bid(t) < bid_(i) < ask(t)$ then $bid(t_{k+1}) = bid_i$ and all other market variables remain unchanged.
  - if $ask(t) < bid_i$, then $price(t_{k+1}) = cr(ask(t_k) + bid_i)$ (where $cr(.)$ is a clearing rule stating the transaction price at which the clearing should occur)[8], $bid(t_{k+1}) = 0$ and $ask(t_{k+1}) = max_{ask}$ (where max_{ask} is the maximum ask an agent can submit in the CDA)
- if $SA_i = ask_i$, it follows the same intuition as above.
- if $SA_i = silent \quad \forall i \in \mathcal{I}$ and $t_{k+1} - t_k > inactivity_{limit}$ or $t_{k+1} = deadline$, then the auction ends. $inactivity_{limit}$ represents a pre-defined period of inactivity during which no bid or ask is submitted. $deadline$ is the preset time when the market closes.

Furthermore, an agent's state will also change, conditional on whether its bid or ask is accepted in the market. If an agent's bid bid_i results in a transaction, $n_i(t_{k+1}) = n_i(t_k) - 1$, $budget_i(t_{k+1}) = budget_i t_k - price(t_{k+1})$ and $\boldsymbol{v}_i = \{v_{2,i}, \ldots, v_{n_i(t_k),i}\}$. If an agent's bid is unsuccessful, then the MS relays this private information to the agent. The agent's visibility is restricted to only bids and asks being submitted in the market (with the agent that submitted a bid or an ask, not disclosed) and successful transactions. This information is publicly available in the MS. Based on the information that describes the market conditions, the agent strategises to submit a competitive offer to buy or sell. Given this background, we now analyse a selection of the most popular strategies for the CDA, from the perspective of the IKB model. We provide a summary of the analysis in table 1.

- **The Zero-Intelligence (ZI) Strategy [9]:** The ZI has a random behaviour. It effectively ignores the market state (MS) and considers only its limit price, $v_{n_i(t_k),i}$

[6] This is the highest value at which a buyer would buy or the lowest value a seller will accept.

[7] Thus, a CDA strategy does not consider external information.

[8] This varies according to the CDA; examples include the midway value or $ask(t_k)$.

	ZI	*ZIP*	*Kaplan*	*GD*	*RB*
Information Layer	Limit price	Limit price and transaction price and Current bid/ask and current profit margin	Limit price and Outstanding bid/ask	Limit price and history of bid/ask and transaction price	Limit price and transaction price and limit price
Knowledge Layer	None	Competitive profit margin, success of trade	Measures for heuristics	Belief that bid/ask will be accepted	Target price based on estimate of CE price, risk factor
Behavioural Layer	Random	History, predictive	No history, non-predictive	History, non-predictive	History, predictive

Table 1: Analysis of five known CDA strategies under the IKB model

(its private information state in the IL) when submitting a bid or an ask in the market. The KL does not compute any *intelligence* and simply forwards $v_{n_i(t_k),i}$ from the IL to the BL.

- **The Zero-Intelligence Plus (ZIP) Strategy [3]:** This is a predictive strategy that uses the history of market information to predict future market condition and adapt to it. It learns the profit margin of agent i to remain competitive given the changing market conditions. The IL collects $bid(t_k)$ and $ask(t_k)$ and $price(t_k)$ (as instructed by the KL). The IL forwards this data, as well as the agent's profit margin (private information in the agent's IL), to the KL. That knowledge is then used in the BL to predict the future market and adapt its profit margin, μ_i, to it. The BL then submits $\mathcal{A}_i = < bid_i|ask_i, silent >$, where bid_i or $ask_i = (1+\mu)v_{n_i(t_k),i}$.
- **The Kaplan Strategy [6]:** This is a non-predictive strategy that makes a decision based only on simple heuristics. Thus, the IL collects the current outstanding bid and ask ($bid(t_k)$ and $ask(t_k)$ respectively) from the MS. Thereafter, using this information from the IL, the KL calculates the measures that are used in the heuristic rules of Kaplan's BL [6]. These rules determine what action, $\mathcal{A}_i = < bid_i|ask_i, silent >$, the agent i submits in the market.
- **The GD Strategy [8]:** This is a non-predictive strategy that uses a history of market information. The BL decides on an action, $< bid_i|ask_i, silent >$, by solving a risk-neutral utility maximisation problem involving a belief that a bid or an ask at a particular value will be successful in the market, and its limit price, $v_{n_i(t_k),i}$. Thus, the BL instructs the KL that it requires such knowledge. The KL then defines the Information Filter (see figure 1), so that relevant information, namely the history of bids, asks and transaction prices ($H(bid(t_{k-1}))$, $H(ask(t_{k-1}))$ and $H(price(t_{k-1}))$ respectively) are filtered to the IL. That information, along with the agent's limit price is passed to the KL. The KL can then compute the belief and passes it, along with the limit price, to the BL.
- **The Risk-based (RB) Strategy [21]:** This strategy is predictive and uses a history of market information. Furthermore, the RB has a more complex behaviour than the ZIP. The intrinsic parameter of the strategy, which is updated in response to changing market conditions, is the risk factor associated with the current good to buy or sell, g. The IL is instructed (by the KL) to record $bid(t_k)$ and $ask(t_k)$ and a history of transaction prices, $H(price(t_{k-1}))$. The KL then uses $H(price(t_{k-1}))$ to estimate the competitive equilibrium price[9] and then a target price (which the agent considers as currently the most profitable offer price in the market). The target price (which is the market knowledge from the KL) is then used along with the agent's limit price, $v_{n_i(t_k),i}$, obtained from the IL and relayed through the KL, in a set of bidding rules in the BL. The latter then decides what offer, $< bid_i|ask_i, silent >$, the agent i sumbits.

Having discussed how the IKB model can be applied to existing strategies for the CDA, we now consider how we can use our framework to engineer a new trading strategy given a market mechanism. Specifically, we consider augmenting the GD strategy by incorporating adaptive behavior so that the new GD strategy, GD* adapts its risk and is therefore no longer only risk neutral. In more detail, the BL of GD* will now select its actions according to a risk-sensitive utility function. The BL will then query the KL for the risk attitude which will be most profitable given the predicted future market conditions. Then, the KL calculates this risk attitude based on its prediction of the future market's and agent's state variables. These variables will depend on the prediction model and the risk calculation method that the designer has specified. Now the KL will query the IL for the relevant information such as the agent's budget, history of bids, asks and transaction prices in the market ($H(bid(t_{k-1}))$, $H(ask(t_{k-1}))$ and $H(price(t_{k-1}))$ respectively). Thus the KL will update the information filter so that the IL obtains the required market data from the MS.

5 Conclusions and Future Work

As electronic marketplaces are being used on a broader scale, we believe software agents will increasingly dominate the trading landscape. Their ability to make informed decisions, based on the plenitude of market information, to a degree that human traders can never achieve, make them ideal candidates

[9]The competitive equilibrium is a price at which transaction prices are expected to converge to as given by the classical micro-economic theory [14].

for traders. However, as this new breed of agents are populating the markets, it is becoming a fundamental challenge to design strategies that can efficiently harness the avalanche of information that is available into efficient trading behaviour. Given this, the objective of this paper is to provide a systematic framework for designing such strategies. To this end, we proposed a framework that can be broken down into three principal components; namely the behavioral layer, the knowledge layer and the information layer. In so doing, we believe this work is an important preliminary step towards guiding the strategy designer by identifying the key models and concepts that are relevant to this task. We applied this model to analyse a selection of strategies in the CDA mechanism and showed its use when designing a new strategy. For the future, we obviously need to verify our framework further by applying it to different types of market institutions.

Acknowledgments

The authors would like to thank anonymous reviewers for helpful comments. Perukrishnen Vytelingum is funded by the DIF-DTC 8.6 project (www.difdtc.com), Rajdeep K. Dash is funded by a BAE Systems studentship and Minghua He is funded by the EPRSC VOES (Virtual Organisation for E-Science) project.

References

[1] B. Bauer, J. P. Muller, and J. Odell. Agent uml: A formalism for specifying multiagent software systems. *International Journal of Software Engineering and Knowledge Engineering*, 11(3):207–230, 2001.

[2] A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, pages 75–90, London, UK, 1996.

[3] D. Cliff and J. Bruten. Minimal-intelligence agents for bargaining behaviors in market-based environments. Technical Report HPL-97-91, 1997.

[4] R. Das, J. E. Hanson, J. O. Kephart, and G. Tesauro. Agent-human interactions in the continuous double auction. *Proceedings of the 17th Joint Conference on Artificial Intelligence*, pages 1169–1176, 2001.

[5] R. K. Dash, D. C. Parkes, and N. R. Jennings. Computational mechanism design: A call to arms. *IEEE Intelligent Systems*, 18(6):40–47, 2003.

[6] D. Friedman and J. Rust. *The Double Auction Market: Institutions, Theories and Evidence*. Addison-Wesley, New York, 1992.

[7] E. Gimnez-Funes, L. Godo, J. A. Rodrguez-Aguilar, and P. Garcia-Calvs. Designing bidding strategies for trading agents in electronic auctions. *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 136–143, 1998.

[8] S. Gjerstad and J. Dickhaut. Price formation in double auctions. *Games and Economic Behavior*, 22:1–29, 1998.

[9] D. K. Gode and S. Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–137, 1993.

[10] M. He, N. R. Jennings, and H. Leung. On agent-mediated electronic commerce. *IEEE Trans on Knowledge and Data Engineering*, 15(4):985–1003, 2003.

[11] N. R. Jennings. An agent-based approach for building complex software systems. *Comms. of the ACM*, 44(4):35–41, 2001.

[12] M. Klein, C. Dellarocas, and J. Rodriguez-Aguilar. A knowledge-based methodology for designing robust multi-agent systems. *Proceedings of the Conference on Autonomous Agents and MultiAgent Systems*, pages 661–661, 2002.

[13] V. Krishna. *Auction Theory*. Academic Press, 2002.

[14] A. Mas-Collel, W. Whinston, and J. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[15] J. A. Rodriguez-Aguilar, F. J. Martin, P. Noriega, P. Garcia, and C. Sierra. Towards a test-bed for trading agents in electronic auction markets. *AI Communications*, 11(1):5–19, 1998.

[16] J. A. Rodriguez-Aguilar, P. Noriega, C. Sierra, and J. Padget. Fm96.5 a java-based electronic auction house. *Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'97)*, pages 207–224, 1997.

[17] W. F. Sharpe. Mutual fund performance. *Journal of Business*, pages 119–138, January 1966.

[18] G. Tesauro and R. Das. High-performance bidding agents for the continuous double auction. *Proceedings of the Third ACM Conference on Electronic Commerce*, pages 206–209, 2001.

[19] M. Tsvetovatyy and M. Gini. Towards a virtual marketplace: Architecture and strategies. In *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, pages 597–613, Blackpool, UK, 1996.

[20] I. A. Vetsikas and B. Selman. A principled study of the design tradeoffs for autonomous trading agents. *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 473–480, 2003.

[21] P. Vytelingum, R. K. Dash, E. David, and N. R. Jennings. A risk-based bidding strategy for continuous double auctions. *Proc. 16th European Conference on Artificial Intelligence*, pages 79–83, 2004.

[22] M. P. Wellman, A. Greenwald, P. Stone, and P. R. Wurman. The 2001 trading agent competition. *Proceedings of the Fourteenth Conference on Innovative Applications of Artificial Intelligence*, pages 935–941, July 2002.

[23] F. Zambonelli, N. R. Jennings, and M. Wooldridge. Developing multiagent systems: the gaia methodology. *ACM Transactions on Software Engineering and Methodology*, 12(3):285–312, September 2003.