Centrum voor Wiskunde en Informatica

REPORTRAPPORT

SEN

Software Engineering

Software ENgineering

SEN

An extensible agent architecture for a competitive market-based allocation of consumer attention space

P.J. 't Hoen, S.M. Bohte, E.H. Gerding, J.A. La Poutré

REPORT SEN-R0217 SEPTEMBER 30, 2002

CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-369X

An Extensible Agent Architecture for a Competitive Market-Based Allocation of Consumer Attention Space

Pieter Jan 't Hoen[1] Sander Bohte[1] Enrico Gerding[1] Han La Poutré[1,2]
Hoen@cwi.nl S.M.Bohte@cwi.nl Enrico.Gerding@cwi.nl Han.La.Poutre@cwi.nl
[1] *CWI, Centre for Mathematics and Computer Science*
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
[2] *School of Technology Management; Eindhoven University of Technology*
De Lismortel 2, 5600 MB Eindhoven, The Netherlands

ABSTRACT
A competitive distributed recommendation mechanism is introduced based on adaptive software agents for efficiently allocating the "customer attention space", or banners. In the example of an electronic shopping mall, the task of correctly profiling and analyzing the customers is delegated to the individual shops that operate in a distributed, remote fashion. The evaluation and classification of customers for the bidding on banners is not handled by a central agency as is customary, but is a distributed process where all shops bidding for a customer partake. This allows each agent for a shop to apply its own private strategy, learning-mechanism, and specific domain knowledge without revealing sensitive business operations to the central party. We present a scalable and extensible software agent architecture and prototype for distributed market-based allocation of customer attention space. The agents can operate in multiple markets concurrently. The protocol for communication between the agents is designed for optimal performance of the system.

2000 Mathematics Subject Classification: 62M45, 68T05, 68T20, 68W15, 90B60, 91B24, 91B26.
1998 ACM Computing Classification System: C.2.4, D.2.0, F.3.0, I.2.0, I.2.1, I.2.9, I.2.11.
Keywords and Phrases: Electronic Markets, Market-based programming, competitive multi-agent systems, learning agents, ACE, agent-based computational economics.
Note: This research has been performed within the framework of the project "Autonomous Systems of Trade Agents in E-Commerce," funded by the Telematics Institute in the Netherlands. An extended version of this report has been presented at Agent Mediated Electronic Commerce IV (AMEC IV), a workshop of AAMAS 2002, Bologna.

1. Introduction

Electronic recommender systems, electronic auctions, and virtual shops are increasing in economic significance [Dailianas *et al.*, 2000; Dellarocas and Klein, 2000; Yoona *et al.*, 2000; Kephart *et al.*, 2001]. As these systems grow in size and increase in the number of customers, the problem of how to match customers correctly to providers increases in complexity and cost. This is especially the case when using centralized electronic applications as large knowledge bases and communication overhead can become a bottleneck. For agent-based systems, heavy-weight differentiated learning algorithms add an additional difficulty. Centralized systems as such face built-in limitations that hinder their scaling.

In [Bohte *et al.*, 2001], a market-based solution is demonstrated that is well suited to match customers to providers. The approach is based on a market-mechanism where a customer request is auctioned off, and the highest bidding agents acting on behalf of the shops determine the presentation of advertisements to the customer. These agents can operate in a distributed setting, avoiding the bottleneck of a centralized application. It was shown in [Bohte *et al.*, 2001] in a simulated setting that agents are able to learn the niche in the market of the advertisers they represent.

In this paper, we extend the above approach to a system of real and efficient software agents. We focus on an architecture where parallelism and support for differentiation in the behavior of the agents is maximized and an efficient communication protocol is used. Agents are able to learn the behavior of customers with multi-dimensional profiles and cope with the additional difficulty of the interleaving and delay of feedback from the customer interaction. A demonstrator of the architecture can be found at ['t Hoen, 2002]. Furthermore, we design the architecture to anticipate for future extensions and for multiple markets. The agents in the shopping mall support a fine-grained hierarchical process where a customer is matched to most appropriate shops in the available markets.

The document is organized as follows. In Section 2 we present the concepts used and the focus of the paper. We continue in Section 3 with an overview of the architecture of the prototype. In Section 4 we work out a case of how the attention space for a customer is bid for. In Section 5 we cover the applied learning mechanism. In Section 6, we present results for an application of the architecture. Lastly, in Section 7 we discuss and conclude.

2. Motivation and Extension

With the advent of electronic marketplaces, scale limitations as encountered in the brick-and-mortar world no longer apply. At the same time, novel problems are encountered, like how customers can find their way in large marketplaces.

A currently preferred solution is to have a central party propose relevant suppliers and products to a customer. This central filtering mechanism uses knowledge of users, shops, and the product domain to determine recommendations. Such recommender systems work well within limited domains, e.g. a book or music store. However, to maintain accuracy in a large marketplace with many customers and suppliers, the latter will need to reveal detailed and perhaps sensitive business information to the central party. Central filtering-mechanisms may thus suffer from objections from the suppliers. Furthermore, the amount of information to be processed and maintained by the central party can become unmanageable. Other approaches are needed to complement centralized systems.

In [Bohte *et al.*, 2001], a novel mechanism called the Competitive Attention-space System (*CASy*) is introduced to handle the above problems, via the techniques of dynamic market-based control [Clearwater, 1995] and adaptive software agents [Weiss, 1999].

In the used example of an electronic shopping mall, the greater part of the task of matching a specific online customer to a set of suitable shops is delegated to the individual shops. Each shop evaluates the information that is available about the current customer: the customer's current interests and other information that the customer is willing to provide or has provided; e.g. keywords, product queries, and available parts of a personal profile. Based on this information and on their own domain knowledge, shops can make a monetary bid in an auction where a limited amount of customer attention space, or banners, for the particular momentary customer is sold. The shops in this approach can incorporate up-to-date and detailed domain knowledge into their agents, which allows for accurate and time-dependent targeting of a shop's audience, taking into account sensitive information and issues like service, quality, price, product diversity, and sales.

As another example of the system, we take the sale of airline tickets. A customer visits the site/shopping mall and enters the desired travel plan along with additional data; for example, restrictions in price and dates of desired departure. The shops, in this case airlines, can evaluate the desired transportation and place a bid for presenting a banner to the customer. These bids are ranked and the higher the bid, the higher the banner is placed for presentation to the customer.The *CASy* allows for airlines to learn to bid strongly for customers who are likely to purchase their flight and less for unlikely customers. For example, an airliner with only luxury seats still available will bid high(ly) for a passenger who has a tight schedule and is willing to spend a large amount. The bid will however be low if the customer is looking for a budget flight and not in a hurry. Through this mechanism, the customer is presented with likely banners advertising products in decreasing order of likelihood to meet his needs.

For various basic models for on-line customers, shops, and profiles, the feasibility of the system has been demonstrated in [Bohte *et al.*, 2001]. We note that the mechanism described is not limited to the above example of the airline and electronic shopping mall, but can easily be applied to other domains where (pre) selection of possibilities has to be guided, like banners on more general websites, or other types of marketplaces such as job agencies, or B2B platforms.

In the above work, the *CASy* mechanism is described and simulated. In this report, we present a scalable, extensible architecture and demonstrator for *CASy* in the form of a distributed, scalable prototype. In this work, agents execute the task of bidding for the attention of each individual customer.

For computational efficiency, we enable the distribution of the shop agents over a network of agent servers using Java [Java, 2002] . The individual shops in a full-blown mall may require heavy computational resources for their classification of customers. A strong point of market-based programming techniques is that the correct bidding strategy is learned and executed by each of the individual shops without need of a central server that runs the algorithms of many shops in parallel. The mechanism makes it possible for each shop to be run on its own dedicated hardware, for maximum computational scalability. This enables for the support of numerous shopping agents where scaling in the number of (heavy-weight) shop agents is realized by the addition of hardware.

The use of learning software agents allows shops to rapidly adapt their bidding strategy such that they only bid for customers that are likely to be interested in their offerings. During the bidding, the agents cope with the additional difficulty of the interleaving and delay of feedback from the customer interaction.

Furthermore, several new agent types are defined to incorporate segmentation of the product space into multiple markets. The market place is segmented into possibly overlapping areas managed by dedicated agents to additionally promote the scaling and applicability of the system.

Note that *CASy* is of interest within the context of Google (www.google.com) and Overture (www.overture.com). Overture has a long standing business-model where advertisers pay to be listed relative to competing companies in the search engine results and now experiences competition in this field from Google with its ADWords. The *CASy* differs from the above two approaches, amongst others, due to the automated, adaptive nature of the bidding agents of the shops. The non-central, distributed market-responsive nature of the agents allows for finer-grained, differentiated strategies in the bidding of the shops based on more detailed customer profiling information.

3. Agent Architecture

In this section, we outline the agents in our architecture in a first approximation. The different types of agents are introduced and their role in the whole is discussed.

3.1 Previously Introduced Type of Agents

In [Bohte *et al.*, 2001], a single market is described where all customer attention space is auctioned off to all participating shops. In this approach, there are two types of simulated agents:

- A "mall manager" agent that runs the shopping mall and hosts the visiting customers.
- A "shop" agent that implements the bidding strategy for the attention space of a customer.

In our architecture, we extend and introduce new agents to handle more complex scenarios; an extension in the segmentation of the markets and a more complex notion of the shop agent.

3.2 Extension to Multiple Markets

Although communication and computational resources in previous work suffice to sustain a significant number of shops in the example of the virtual shopping mall, the application of the *CASy* here is generalized by segmenting the single market into multiple markets. The mall manager agent decides where to auction a particular customer attention space. This allows shops to only participate in a selection of the available markets, in particular those that address their own business areas.

The mall manager performs a rough first classification, say separating the clothing-buyers from the garden-equipment buyers. The assigned market will then perform the appropriate allocation. The mall manager could possibly also assign a customer to multiple markets, to allow the segmented markets to have some overlap. Many choices for segmentation in the shopping mall are possible. Of importance is that for the choices made, the *CASy* mechanism is still applicable.

For the role of managing the individual markets, we introduce a dedicated *Auctioneer* agent. This agent, as implied by its name, runs the auctions for the sale of the customer attention space for its specific market.

3.3 Refinement of the Shop Agent

To allow for increased granularity in the markets, the shop agent as introduced in Section 3.1 is split into two parts; each consisting of new type of agents.

One part of the shop agent architecture, called the set of *ShopNegotiator*s, runs the actual bidding for the customer attention with respect to the individual markets. Each real-life shop has one such bidding agent for each market it participates in, allowing for maximum parallelism in the bidding on the markets.

[Cardoso and de Engenharia, 2001] notes that"organizations in general have differentiation strategy when approaching Electronic Commerce. Thus, in order to be useful, agent technology must support this market characteristic". The use of separate *ShopNegotiator* agents allows for the differentiation in strategy per market segment a shop operates in.

As a second part of the shop agent architecture, we introduce the high-level agent part of the real-life shop called the *ShopManager*. This agent, as one of its main tasks, handles the (logging of the) actual purchases by a customer. Secondly, it can (in future extensions of the prototype) manage the complexities of the interactions between the concurrent bidding for multiple markets by the *ShopNegotiator*s. In the prototype we handle each market as a separate, individual scenario. If however, customer profiling information between the markets could be exchanged, then this is a task that can be assigned to the *ShopManager* in the sense of collaborative filtering [Sorensen *et al.*, 1997]. The *ShopManager* can then assist in the exchange of customer information between its own *ShopNegotiator*s and collaborate with friendly other shops. As another example, budget considerations for learning pertaining to multiple markets can become an issue monitored by this high-level agent.

The next section describes the communication between the agents and gives an example of a visit of a customer to the shopping mall.

4. Agent Communication

Figure 1 shows a representation of the communication channels and agents and actors in the prototype. One customer is described in a scenario in which an auction and the ensuing steps of the communication protocol amongst the agents is shown.

Note that Figure 1 is a representation of the agents for *one* market. The one dedicated auctioneer agent is represented for this one market along with the mall manager. The shop manager agents along with their shop negotiators for this market are also shown. For additional markets, one new auctioneer agent and one shop negotiator for each shop (manager agent) are added to make the figure complete.

In the example scenario one customer visits the mall, is presented with a list of banners from the shops and possibly makes a purchase. The links are numbered in the order in which they are activated. Where two links can be activated in parallel the suffix a or b has been added.

- 1. A *Consumer* enters the mall for shopping. The *login* message to the *MallManager* registers the customer and a profile for the customer is registered.

- 2. The relevant part of the profile is sent to the appropriate *Auctioneer*s through the *profile_to_A* channel to be able to process the customer. This information is private and con-

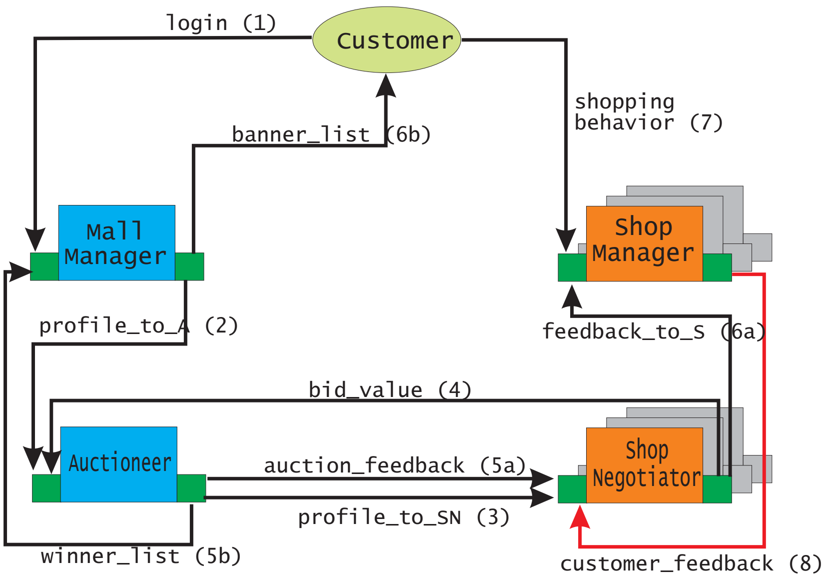

Figure 1: Overview of the Architecture

fidential within the context of the $MallManager$ and its auctioneers that run the separate markets.

- 3. A partial profile, possibly augmented by aggregate information, is sent to the *ShopNegotiators* so that each shop can determine their interest in the customer. *ShopNegotiators* receive only the necessary information on the customer if the customer is assigned to the specific market/*Auctioneer* the shops are operating with. This allows for privacy and legal considerations as for example discussed in [Conan *et al.*, 2000]; the $MallManager$ and the *Auctioneer* together can operate as an intermediary to ensure correct use of the customer data.

- 4. The *ShopNegotiators* each place their bids for customer attention space. Future extensions may include multi-round negotiation protocols.

- 5a. The *Auctioneer* calculates the results from the bids of the individual *ShopNegotiators*, and through $auction\_feedback$ the result is given to the *ShopNegotiators*. This entails whether they have won the bid and of the costs attached to this result.

- 5b. The feedback to the *ShopNegotiator* is also passed on to its $ShopManager$ for high-level considerations, for example cross-market budget considerations.

- 6a. The *Auctioneer* communicates to the $MallManager$ the resulting banner list which is to be presented to the customer.

- 6b. The *Consumer* is presented with a list of banners from which to choose for the purchase of goods as indicated by his profile.

- 7. The *Consumer* decides whether to purchase a product after inspecting the presented banners. The appropriate $ShopManager$ is visited.

- 8. The $ShopManager$ gives feedback to the *ShopNegotiator* on the actions of the customer (ignored/browsed/bought).

In the next section we discuss the learning mechanism for the shops in their bid for the attention space of the customer.

5. Learning Mechanism

As a demonstrator of the architecture, we built an example learning mechanism that exploits the feedback loops (steps $5a$ and 8 in Section 4, Figure 1) available in the shop negotiation mechanism. We use these feedback loops to maximize the profits for the shop negotiators participating in the bidding by learning the chance of purchase for customers.

We use a second-price Vickrey auction. In this type of auction, the higher bidder wins, but pays only the second highest bid. The dominant strategy is the bidding of the actual expected profit (valuation) [Vickrey, 1961], under the assumption of sufficient competition between the $ShopNegotiators$. In this auction setup, in the long run, the agent cannot profit by using a different strategy [Sandholm, 1996].

From the auction, there is feedback whether the bid is won or not. For a winning bid, there is a customer who may make a purchase at the shop for which the shop negotiator presented the banners. From this information the shop negotiator agents can derive the expected profit from a customer making a purchase given the customer profile and the profile of the shop negotiator. The feedback and the profile information are combined in one strategy for the bid placed for the attention space of a customer.

5.1 The Profiles

The customers and the shops are encoded using a profile upon which we base the stochastic purchase behavior of the customers. The greater the distance between a customer profile c and a shop profile s, the smaller the chance of purchase.

Both the customer and the shop profile are encoded as a number in the interval $[0, 1]$, for n dimensions. The distance $||\vec{c} - \vec{s}||$ between two profiles is the euclidean distance. In [Bohte *et al.*, 2001], the learning algorithms are explored for one and two dimensions. In this paper, we generalize to one or more dimensions.

5.2 Collection of Training Data

We administrate the data of the customers to whom a banner is presented. Provisionally, we initially see a customer as a non-buyer until observed otherwise, i.e. a purchase is made. We correct our training data when, after a random delay, the customer actually makes a purchase. The shop negotiators hence have to deal with a delay and permutation in the order of the customer reactions to which they present a banner.

A buying customer is coded as a 1 for its profile while a non-buying customer is coded as a 0. For a given shop negotiator, data is accumulated as successful bids are made for the attention space of a customer. The data points close to the shop negotiator profile will predominantly be 1s while further away (in euclidean distance) the number of "0" data points will increase in relative frequency while the total of number of data points further away from the shop negotiator will drop under intense competition. As such, from the accumulated data of (buying) customers, an area of the profile can be derived in which the shop is competitive; its niche in the market.

The next section introduces Radial Basis Function networks which we use to learn the customer behavior from the accumulated data.

5.3 Fitting the Chance of Purchase

Radial Basis Function (RBF) networks are a class of neural networks in which the activation of a hidden unit is determined by the *distance* between the input vector and a prototype vector [Bishop, 1995]. Each hidden unit represents a single RBF. The main advantage of such networks is that using several RBFs different RBF functions can be used to represent areas of the profile space that can be updated more or less locally. RBF networks with multiple centers are hence well-suited if multiple niches with different characteristics must be learned simultaneously by a shop negotiator.

Additionally, efficient algorithms exist to capture clustering using RBF networks [Asim Roy, 1995] in case customers are not distributed homogeneously over the profile space.

A RBF, ϕ, is one whose output is symmetric around an associated *center*, μ_j [Bishop, 1995]. A set of k weighted RBFs can serve as a basis for representing a wide class of functions that are expressible as linear combinations of the chosen RBFs [Bishop, 1995]:

$$y(\vec{x}_i) = \sum_{j=1}^{k} w_j \phi_j(\vec{x}_i), \tag{5.1}$$

A RBF network is an embodiment of Equation 5.1 as a feedforward neural network with three layers: the inputs, the hidden layer representing the RBFs and, in our prototype, a single output node.

The training of the basis functions ϕ_k consists of updating their centers $\vec{\mu}_j$, their radius σ_j and their weights w_j to minimize the error. If we choose an equal number of centers as the number of samples, we can even arrive at a perfect fit for the data. A regularization term may have to be added to ensure smoothness of the derived function (y), although this is not the case in our application.

We take k ($k \geq 1$) basis functions ϕ_j with centers $\vec{\mu}_j$ and radius σ_j. The output of the RBF for $\vec{x}_i$ is given by Equation. 5.1. The error is the summation over all the n patterns of the data, here the standard mean-square error $E_n = \frac{1}{2}\sum_k \{\vec{t}_n - y(\vec{x}_n)\}^2$ [Haykin, 1994]. $\vec{x}_n$ is the value of sample n, t_n is the desired function value for sample n and y is the function that we use to approximate the data given the $< \vec{x}_i, t_i >$ pairs.

In equations below, we present the Δ training rules based on method of steepest descent [Haykin, 1994]. η_1 to η_3 are the learning rates for the different aspects. Currently they are set to the commonly used value of 0.001.

$$\begin{aligned}\Delta w_j &= -\eta_1 (y(\vec{x}_n) - t_n)\phi_j(\vec{x}_n),\\ \Delta\vec{\mu}_{j,l} &= -\eta_2 \phi_j(\vec{x}_n) w_j \frac{(\vec{x}_{n,l} - \vec{\mu}_{j,l})}{\sigma_j^2}(y(\vec{x}_n) - t_n) \text{ and}\\ \Delta\sigma_j &= -\eta_3 \phi_j(\vec{x}_n) w_j \frac{||\vec{x}_n - \vec{\mu}_j||}{\sigma_j^3}(y(\vec{x}_n) - t_n).\end{aligned}$$

Parameters $\vec{\mu}_{j,l}$ and $\vec{x}_{n,l}$ are the lth component of the vectors for the centers and training samples. Repeated iteration over batch training over all the samples converges the RBF function-approximation y to the sought for function, in our case the chance of purchase.

5.4 Correct Bids

As we use a second-price Vickrey auction, the dominant strategy is to bid the valuation of the shop negotiator for a given customer. The expected profit for a shop negotiator is its $margin$ times the estimated chance of purchase of a given customer. The $margin$[1] is the average profit in case of a sale. This gives as bid for the attention space of a customer:

$$margin * ChancePurchase(customerprofile, shopnegotiatorprofile) \tag{5.2}$$

where the chance of purchase depends on the distance between the profile of the customer and that of the shop. The chance function has to be initially approximated and learned over time. Note that, as opposed to [Bohte *et al.*, 2001], we for the sake of simplicity only assume competition for the placement of one banner and a simple customer model. The RBF networks due to their general nature are however suited for the more complex customer models.

Like in the above work, we define the chance of purchase of a customer with respect to a shop negotiator as exponentially decreasing with the distance between the profile of the customer and the shop. We use as function a Gaussian

$$e^{-\frac{(||\vec{c}-\vec{s}||)}{2\sigma^2}} \tag{5.3}$$

[1] In our simulations, we postulate a standard $margin$ for all the participating shop negotiators. This simplifies the comparison of successful bidding behavior between the different shop negotiators within one market.

where $\vec{c}$ is the profile of the customer and $\vec{s}$ the profile of the shop and $||\vec{c} - \vec{s}||$ the distance between the profiles. σ is the radius of the Gaussian that determines the rate of drop with distance. However, any bell shaped function would be realistic. Note that we assume that all customers have an identical chance-of-purchase function. Furthermore, the buying behavior of the customers is static, as is quite common in customer modeling [Kephart *et al.*, 2001].

In the prototype, we integrate the knowledge of the shape of the function of the chance of purchase by a customer; the shop negotiator knows that this is a Gaussian, a bell-shaped curve dependent on the distance in profile space. Furthermore, the shop negotiators are provided with some upper bound for the radius of this Gaussian. The above information may, for example, originate from initial market research and can lead to a choice for a shop to participate in a market as a potentially fruitful enterprise. The learning algorithm of the shop negotiator then fine-tunes the bids for the attention space as the precise purchase behavior is learned. Hence a shop can enter a new market based on prior expectations, and after some time learns to compete optimally.

The actual optimal bid for a *ShopNegotiator* with profile $\vec{s}$ for a customer $\vec{c}$ that has to be learned based on Equation 5.2 is:

$$bid = margin * e^{-\frac{(||\vec{c}-\vec{s}||)}{2\sigma^2}} \tag{5.4}$$

The shop negotiator knows its own profile, it knows the profile of the customer for who it has to make a bid as given by the auctioneer and it knows its own margin. It however needs to learn the value σ of the customer.

For the learning of σ, we currently need only one RBF function, a Gaussian. The center is set to the profile of the shop negotiator. The weight of this center is set to 1. Note that the weight update rules should be used if the chance of purchase for a customer is not 100% if the profile of the customer and shop match exactly.

5.5 On-line Learning

The customer behavior is learned on-line. Data is accumulated and, once sufficient customers are targeted by a winning bid and the accompanying banner, adaptation of the parameters of the RBF is started. At the beginning of learning, we work with all the data we collected.

Continuing the learning using all collected data means that a shift in the behavior of customers (in terms of buying chance) will become harder to detect as it can disappear as noise in the large amount of total data. As an initial means of on-line learning we use a sliding window approach and only train with a fixed number p of data points for the last p customers for who a successful bid for the attention space has been made. The size of the window p is chosen empirically for our experiments as sufficiently large to be able to derive the σ of the Gaussian that determines the buying behavior of the customers. Using this approach, a gradual shift in the σ of the customer can be detected when p or more new customers have been bid for after a shift in the customer behavior has started.

This on-line learning technique, although straightforward, is trivially correct. It also serves to demonstrate the flexibility of the prototype. More advanced, existing techniques can for example also be used to adjust the RBF network to online-learning, e.g. [Saad, 1998].

6. Results of the Learning of the Chance of Purchase

For the experiments we used a customer with a σ of 0.3, although varying the radius in the interval $[0.1, 0.8]$ did not result in significant changes in the results. We used 1 to 7 Gaussian base functions for the RBF, but this did not make a significant difference for the matching of the chance of purchase by the customer. This will however become an issue if more complex models are used for the purchase behavior and hence we have set up the prototype to support multiple RBF functions.

We found that reasonable estimation of σ was possible after registering 25 buying customers. A perfect fit was achieved with a minimum of 30 purchasing customers.

In a run with 20 shops for a customer profile in $[0, 1]^3$, the shop negotiators learned σ to within 0.05 after 30 purchasing customers per shop negotiator were successfully bid for. We stress that these

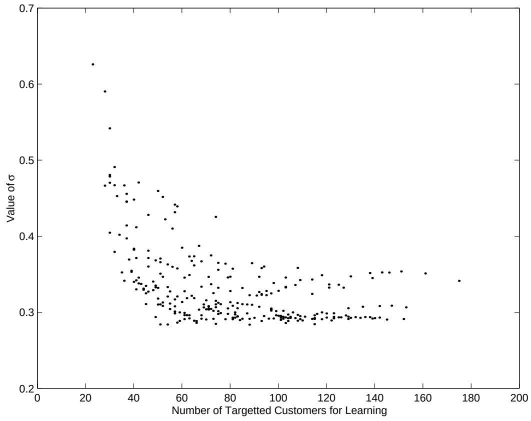

Figure 2: Learning with Customer Feedback

results are for *ShopNegotiators* operating in one, independent market. Learning in multiple markets occurs in parallel.

The evolution of σ for all the shops in a typical run is shown in Figure 2 for a customer *sigma* of 0.3. The interpretation of σ for each learning point of all the individual shops for the number of targeted customers is shown. A gradual move from the initial over optimistic value of σ to the actual value of 0.3 is observed as the *ShopNegotiators* learn from experience.

Recall from Section 5.2 that provisionally customers are interpreted as non-buyers. This leads to a more pessimistic customer chance of purchase view than is actually the case with delay in the customer feedback. It is however possible for the *ShopNegotiators* to correct for this effect from analysis of the observed buying behavior. As data is collected on the customers, a *ShopNegotiator* can build up an expectation of the average time to purchase for the buying customers and integrate this information in its bidding model.

7. Conclusion

We presented an agent architecture and a prototype of a distributed shopping mall of a novel mechanism called the Competitive Attention-space System (*CASy*) introduced in [Bohte *et al.*, 2001]. Within the shopping mall, shops purchase the attention space of the customer by participating in an auction. A mall manager acts as an intermediate between customers and shops and facilitates the auction mechanism. The actual auctioning is delegated to specialized auctioneer agents, which operate on individual market segments. This, as an extension to the original work, enables simultaneous bidding in multiple, independent markets. Similarly, the shops bid on the attention space using a shop negotiator agent, which is also specialized to handle customers from a particular category or market.

The shop negotiators can learn their niche and adapt their bidding strategies to the actual customer purchasing behavior. This learning occurs online as different types of customers subsequently arrive at the mall.

The chosen architecture enables shops and the mall manager to be run on different machines. The agents in the application can be distributed over dedicated machines for maximum availability of computational resources. This is of importance as we envision that heavy-weight learning algorithms will be employed in future full-blown extensions of the shop agents. Communication between the agents is kept to the acceptable low order of linear in the total number of customers that visit the shopping mall. The segmentation of the market space supports the propagation of messages to only the relevant agents.

Furthermore, we also simulated the customer behavior. In contrast to previous work, the purchases of the customers are not immediate but delayed in time. This provides new challenges for the learning mechanism.

Using the feedback from the auction (win/loss) and of the customer behavior, the shops adapt their bids. The purchasing behavior of the customers is estimated using radial basis function networks. Preliminary results show that the shops are capable of finding their niche and adapting to the customer behavior for simple settings (20 shops, a single banner and a three-dimensional customer profile). In future work we aim at extending the results for additional settings of the simulation and at augmenting the current learning mechanism.

References

[Asim Roy, 1995] Raymond Miranda Asim Roy, Sandeep Govil. An algorithm to generate radial basis function (rbf)-like nets for classification problems. *Neural Networks*, 8(1):179–201, 1995.

[Bishop, 1995] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, 1995.

[Bohte *et al.*, 2001] S.M. Bohte, E.H. Gerding, and H. La Poutré. Competitive market-based allocation of consumer attention space. In M. Wellman, editor, *Proceedings of the 3rd ACM Conference on Electronic Commerce (EC-01)*, pages 202–206. The ACM Press, 2001.

[Cardoso and de Engenharia, 2001] Henrique Lopes Cardoso and Eugnio Oliveira Faculdade de Engenharia. A platform for electronic commerce with adaptive agents. In *Agent Mediated Electronic Commerce III*, volume 2003 of *Lecture Notes in Artificial Intelligence*, pages 96–107. Springer, 2001.

[Clearwater, 1995] S.H. Clearwater, editor. *Market based Control of Distributed Systems.* World Scientific Press, Singapore., 1995.

[Conan *et al.*, 2000] Vania Conan, Marten Foss, Peter Lenda, Sophie Louveaux, and Anne Salaun. Legal issues for personalised advertising on internet: The AIMedia case study. In *Agent Mediated Electronic Commerce II, Towards Next-Generation Agent-Based Electronic Commerce Systems, IJ-CAI 1999 Workshop*, volume 1788 of *Lecture Notes in Computer Science*, pages 40–67. Springer, 2000.

[Dailianas *et al.*, 2000] Apostolos Dailianas, Jakka Sairamesh, Vibby Gottemukkala, and Anant Jhingran. Profit-driven matching in e-marketplaces: Trading composable commodities. In *Agent Mediated Electronic Commerce II, Towards Next-Generation Agent-Based Electronic Commerce Systems, IJCAI 1999 Workshop*, volume 1788 of *Lecture Notes in Computer Science*, pages 153–179. Springer, 2000.

[Dellarocas and Klein, 2000] Chrysanthos Dellarocas and Mark Klein. Civil agent societies: Tools for inventing open agent-mediated electronic marketplaces. In *Agent Mediated Electronic Commerce II, Towards Next-Generation Agent-Based Electronic Commerce Systems, IJCAI 1999 Workshop*, volume 1788 of *Lecture Notes in Computer Science*, pages 24–39. Springer, 2000.

[Haykin, 1994] S. Haykin. *Neural Networks-A Comprehensive Foundation, Macmillan And Ieee Press*, volume York. Macmillan College Publishing Compagny Inc., 1994.

[Java, 2002] Java. JSDK 1.4.0, 2002. http://www.java.sun.com.

[Kephart *et al.*, 2001] J. O. Kephart, Christopher H. Brooks, and Rajarshi Das. Pricing information bundles in a dynamic environment, 2001. Third ACM Conference on Electronic Commerce Tampa, Florida, USA.

[Saad, 1998] D. Saad, editor. *On-Line Learning in Neural Networks.* Cambridge University Press, 1998.

[Sandholm, 1996] T. Sandholm. Limitations of the vickrey auction in computational multiagent systems. In *2nd International Conference on Multiagent Systems (ICMAS-96)*, pages 299–306. AAAI Press, 1996.

[Sorensen *et al.*, 1997] H. Sorensen, A. O'Riordan, and C. O'Riordan. Profiling with the INFOrmer text filtering agent. *J.UCS: Journal of Universal Computer Science*, 3(8):988–999, 1997.

['t Hoen, 2002] P.J. 't Hoen. Applet demonstrator of the CASy principle, 2002. www.cwi.nl/~hoen/Demo/demo.html.

[Vickrey, 1961] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[Weiss, 1999] G. Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* The MIT Press, Cambridge, MA, 1999.

[Yoona *et al.*, 2000] Shim Yoona, Ju Young Yun, SooWoong Kim, and Juneha Kim. Jangter: A novel agent-based electronic marketplace. In *Agent Mediated Electronic Commerce II, Towards Next-Generation Agent-Based Electronic Commerce Systems, IJCAI 1999 Workshop*, volume 1788 of *Lecture Notes in Computer Science*, pages 102–112. Springer, 2000.