Creating Value with Interactive Pricing Mechanisms – a Web Service-Oriented Architecture

Martin Bernhardt, Oliver Hinz
Johann Wolfgang Goethe-University Frankfurt, [bernhardt|ohinz]@wiwi.uni-frankfurt.de

Abstract

Lower transaction costs and new possibilities to interact with customers online have led to a plethora of interactive pricing mechanisms on the Internet. While some of these such as various online auction formats stem from the offline world, others such as Reverse Pricing had previously been unknown. Interactive pricing mechanisms let buyers actively participate in the price discovery and their usage offers sellers a chance to increase sales by means of price discrimination and attraction of new customer segments. Implementation of such pricing functionality however is often time-consuming and costly. Therefore, we propose a Web-Service-oriented architecture enabling sellers to use interactive pricing mechanisms on a scalable basis. All mechanisms can be individually designed for each product to accommodate for different product characteristics and special seller needs.

1 Introduction

One of the key economic processes when a seller and a buyer engage in trading is that of price discovery [1], i.e. finding a price that both buyer and seller accept. Due to the characteristics of the Internet, this process has undergone drastic changes over the past few years. Lower menu costs, the reduction of processing costs associated with price differentiation, and new possibilities to interact with trading partners online have enabled a plethora of interactive pricing mechanisms.

In the context of this paper we define interactive pricing mechanisms as the subset of dynamic pricing mechanisms where buyers can actively influence the final price of a product by submitting bids or exchanging messages with a seller. In order to decouple physical from virtual presence, electronic agents can be used in this interactive process (e.g. bidding agents).

While some interactive mechanisms on the Internet such as various online auction formats stem from the offline world, others such as Reverse Pricing have previously been unknown. The trend towards interactive pricing mechanisms on the Internet becomes evident by the growing importance of online auctions as the most widely used form of interactive pricing mechanisms [3]. A recent study shows that every fourth euro German Internet users pay on the net is spent during online auctions [11], a sharp increase from the 15.8 % of all online sales in the previous year [10].

Since interactive pricing mechanisms will charge different prices from different buyers for identical products, they enable sellers to price-discriminate. This could result in a seller charging higher prices from some buyer segments. However, buyers can also profit from price discrimination as some buyer segments that would be priced out of the market in a posted price scenario could then be served for lower prices [1]. Price discrimination seems especially attractive if variable costs are low and sellers profit from additional sales regardless of the price. Interactive pricing mechanisms could then be used to segment buyers by certain characteristics and to charge different prices, e.g. according to their willingness-to-pay.

The usage of interactive pricing mechanisms has been a common feature on multiple online marketplaces such as eBay (http://www.ebay.com) or Amazon (http://www.amazon.com). Despite the success of these marketplaces, regular online shops only rarely apply interactive pricing functionality on their websites as implementation can both be time-consuming and costly. Additionally, the decision about which interactive pricing mechanism to choose and how to calibrate various design alternatives available for it could prevent sellers from implementing such functionality.

Therefore, the aim of this paper is to present a scalable architecture enabling sellers to apply various interactive pricing mechanisms within their own website without having to face the risks and costs of implementation. Based on Web Service technology, this architecture allows for the calibration of pricing mechanisms based on the characteristics of a product. Sellers thus not only gain access to a large number of different interactive pricing mechanisms within this architecture but also possess full control over their respective design specifications. This paper uses Reverse Pricing as an example of interactive pricing mechanisms; however, architectural concepts presented hereafter can easily be transferred to other interactive pricing mechanisms such as auctions or negotiations.

The remainder of this paper is organized as follows: Chapter 2 introduces the Reverse Pricing mechanism as one example of an interactive pricing mechanism and argues why sellers might be inclined to outsource the mechanism. The illustration of the system architecture and the detailed description of the different modules in chapter 3 are at the core of the paper. Building upon this discussion, the description of a prototype along with possible benefits and drawbacks aims at illustrating this concept further. Chapter 4 concludes the paper with final remarks and directions for future research.

2 Integration of Reverse Pricing Functionality

Reverse Pricing is an interactive pricing mechanism letting both buyer and seller influence the final price of a product. While a seller sets a secret threshold price above which she is willing to sell the product for, a buyer determines the final price by submitting a bid above the seller's threshold price, i.e. placing a successful bid. If a buyer's bid does not surpass the seller's threshold price, the ability to place additional bids depends on characteristics of the mechanism design typically defined by the seller.

Using so called design variables [2], sellers can calibrate a Reverse Pricing mechanism according to either their preferences or a product's characteristics. Due to the large number of design variables available and their strong influence on the bidding behavior of potential buyers (as suggested by [5], [8], [17]), Reverse Pricing can be considered a flexible yet complex interactive pricing mechanism. Apart from the determination of the threshold price, which is at the core of the calibration process a seller has to go through, design variables such as the number of bids a single buyer is allowed to place on a specific offer or the price elicitation format a buyer can use to place a bid also deserve a seller's attention. For example, a buyer could be presented a list of selectable prices rather than an input field where the bid price can be freely entered (see [5]). In addition to such design deliberations, the introduction of certain restrictions such as bidding fees or the introduction of time delays in-between two consecutive bids (outlined in detail by [2]) could further influence a buyer's bidding behavior and thus ultimately affect a seller's profit.

In a Reverse Pricing mechanism, information about a seller's secret threshold price is distributed to neither buyers nor other sellers. Overstock capacity, for example, could thus be sold with a discount to certain buyer segments with a lower willingness-to-pay over a Reverse Pricing channel. At the same time, information about this discount would not be communicated to other buyer segments, keeping cannibalization of prices in different sales channels at a minimum rate.

Despite its introduction by US-based company Priceline (http://www.priceline.com) in 1998, Reverse Pricing has only recently received considerable attention in academia. Thus far, research in the area of Reverse Pricing has focused on analyzing consumer bidding behavior (see [5], [7], [13]) or questions arising from different design alternatives of the mechanism (see [2], [8], [17]). Even though the latter field of research already points to the complexity of implementation and profit implications to be considered under different design alternatives, the question of how to optimally apply the mechanism and its underlying IT infrastructure has not yet gained adequate attention. Yet, in order to enable the widespread use of Reverse Pricing and the optimal application of various design alternatives, it is essential to provide a solution that is both easy to integrate in existing IT infrastructures and flexible enough to be adjusted to diverse product characteristics.

Due to this inherent complexity, integration of Reverse Pricing functionality by developing a proprietary solution can become a task too expensive for a seller. Besides, considering the fact that different products might require different design alternatives or altogether different pricing mechanisms to realize their full revenue potential, the flexibility and adaptability of a solution both deserve particular attention. A proprietary solution however might not be able to handle all design alternatives available, especially if the cost of implementation is to be kept at a reasonable level. Following this line of argumentation, this paper proposes a scalable solution based on Web Services providing sellers with an easy and cost-efficient way to integrate Reverse Pricing functionality in their own website (e.g. their online shop system).

Other potential application scenarios such as the usage of a third-party marketplace to sell products using Reverse Pricing functionality are not considered in this paper. These third party offerings typically lack the flexibility and adaptability of the service-oriented solution proposed hereafter. For example, a seller might fear the increase in competition and the loss of corporate identity occurring within the marketplace and thus seek a solution allowing him to integrate the desired functionality within his own website.

3 Service-Oriented Pricing

In order to integrate potentially complex interactive pricing functionality, sellers can use the service-oriented pricing architecture presented in this chapter. In contrast to traditional approaches in commerce, where pricing can be considered one of the core competences of the seller, this architecture outsources the process of price discovery and introduces the service provider as a new key player in the process. This intermediary, as specialist in pricing, constantly gains insights into buyers' behavior in interactive pricing processes from distributed sources, and can thus become a valuable stakeholder in such disaggregated value chains [14]. The importance of intermediaries and information systems for Electronic Commerce based on collaboration has been demonstrated by [16].

The architecture of a service-oriented pricing system presented hereafter relies in large parts on Web Service technology. In order to focus on the architectural description rather than specifications of underlying technologies, details of the protocols which Web Services build upon remain beyond the scope of this paper. Interested readers should refer to [9] for a short survey on Web Service Technology, for detailed information visit [20].

3.1 Architecture of a Service-Oriented Pricing System

Figure 1 depicts the architecture of a service-oriented pricing system and illustrates the collaboration of the three parties involved in the distributed pricing process. In this scenario, a buyer is involved in the process of bidding for a product offered within the seller's online shop. While the product information originates directly from the seller's `Product`-database, the price information representing the design of the Reverse Pricing mechanism is included dynamically from the service provider offering respective functionality. As is directly observable from the illustration in Figure 1, all customer and product related data stays entirely within the seller's domain. Even though sellers can utilize the full flexibility this architecture offers, they neither have to share sensitive data with third party offerings nor do they have to interfere with their existing system architecture.
Communication between the seller's and the service provider's system occurs solely on the basis of XML messages generated by the corresponding Web Services at the service provider's system or processes at the seller's system. From the service provider's view, adequate XML structures have to be developed in order to facilitate standardized communication with multiple sellers so as to allow for a scalable architecture.

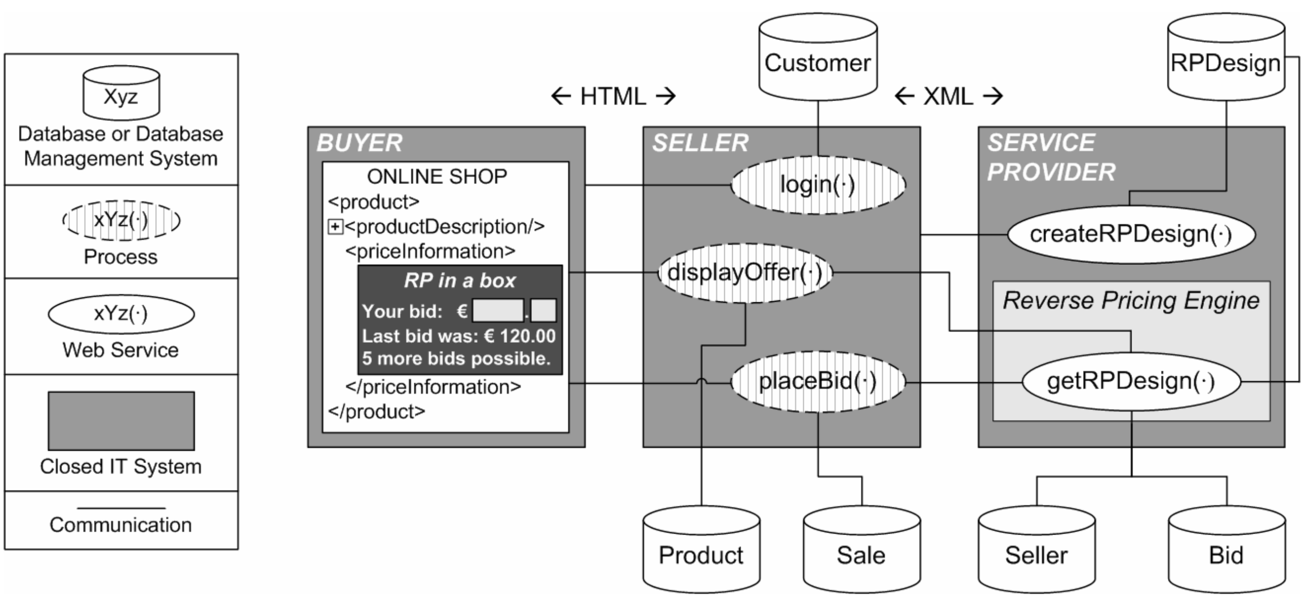

Figure 1. Architecture of a Service-Oriented Pricing System

At the seller's system, the information received in XML-format can then be transformed by XSL-Transformations (XSLT) into HTML code presenting the information to the user in the buyer's web interface (usually a web browser).

If a seller wishes to employ the Reverse Pricing mechanism for one of the products stored in his product database, he needs to specify a respective Reverse Pricing design for this product. The service provider offers the possibility to do so by the `createRPDesign(·)`-Web Service. In order to ensure the structural validity of the seller's input and the seller's insulation from such syntactic details, XForms could be applied to grant sellers access to this Web Service [21]. While the combination of the `createRPDesign(·)`-Web Service and respective usage of XForms could serve as one passable solution to the specification of a suitable Reverse Pricing design, a dynamic web site (compare Figure 2 in chapter 3.2) is another option. Using this functionality, a seller can calibrate the design of a Reverse Pricing mechanism according to both the product's characteristics and the selling context such as demand forecast, costs and capacity. While information about the product's threshold price is at the core of this calibration, several other design variables such as the maximum number of bids allowed or the minimum time-span in-between two consecutive bids can be set, too.

The service provider then stores this information along with respective identifiers for seller and product in the `RPDesign`-database. Subsequently, the product considered here is designed to be sold using the Reverse Pricing mechanism rather than posted prices set by the seller.

In order to identify specific buyers, the seller operating the online shop must provide a customer database as well as a corresponding `login(·)`-process for buyers browsing the products within the online shop. As soon as a buyer has been identified at the seller's system, the service provider can query the `Bid`-database for previous bids of the respective buyer and dynamically adapt the design of a mechanism according to the buyer's preceding bidding behavior. Whenever pricing information is requested for a product a seller has configured for Reverse Pricing prior to this request, the `displayOffer(·)`-process checks if such information is available for this unique product-seller combination at the service provider's system. For this reason, the `getRPDesign(·)`-Web Service nested within the service provider's Reverse Pricing Engine is used. This engine represents a collection of functions combining all the services needed to supply Reverse Pricing functionality to different sellers. As mentioned earlier in this paper, similar function-collections could be available for other interactive pricing mechanisms such as auctions or negotiations.

Upon invocation, the `getRPDesign(·)`-Web Service queries the `RPDesign`-database as well as the `Seller`- and the `Bid`-database in order to check for previous bids placed by this buyer on the product represented by the respective design specification in the `RPDesign`-database. If no such bids can be found, the design for the initial bid is presented to the buyer in the buyer's web-interface (i.e. browser). In order to do this, an XML stream sent back to the seller can be transformed into HTML by XSLT at the seller's system. Typically, this XML stream communicated between service provider and seller contains design information about the maximum number of bids possible or potential restrictions such as fees for additional bids or the minimum time to wait in-between two consecutive bids on a specific offer. Naturally, service provider and seller would have to agree upon the usage of a standardized XML structure to ensure accurate operation of this automatic communication procedure. Instead of being shown a posted price the buyer would then have the opportunity to place a bid on the chosen product within the Reverse Pricing mechanism specified by the respective design specifications. The possibility to do so could be presented to the buyer in a dedicated area ("Reverse Pricing in a box") including all relevant information on the mechanism design as well as the functionality to place a bid such as an input box or a drop down list with available prices.

If the buyer decides to place a bid under these design specifications, the `placeBid(·)`-process needs to pass on the bid price along with an identifier for product and buyer to the `getRPDesign(·)`-Web Service. After storing the bid price in the `Bid`-Database, this Web Service then evaluates whether the buyer's bid price is high enough to surpass the seller's threshold price under the current design specification. If this is the case (i.e. the buyer's bid is "successful"), according information is passed back to the seller's `placeBid(·)`-process that can then store the price in a database system managing sales. Consequently, the buyer can immediately be notified about his success and the product can automatically be added to an online shopping cart for the price denoted by the buyer's bid.

However, if the buyer's price does not surpass the seller's threshold price, the `getRPDesign(·)`-Web Service evaluates the design for the next bid and sends corresponding data back to the seller's `placeBid(·)`-process, which in turn presents the data to the buyer in a dedicated area within the online shop. If the design specifications in effect don't restrict further bidding by the buyer, he can raise his initial price and place additional bids. As long as additional bids are possible for the buyer, the complete process can be repeated over again.

With the service-oriented architecture presented here, sellers are provided with cost-efficient access to easy and flexible integration of Reverse Pricing functionality. Naturally, other interactive pricing mechanisms could be integrated following a similar approach. In addition to the static determination of mechanism design largely handled by the `getRPDesign(·)`-Web Service, dynamic extensions of the architecture described thus far could introduce additional flexibility to the pricing mechanisms implemented in the service provider's system. This would allow for a dynamic computation of design specifications such as the threshold price or the minimum time in-between two consecutive bids based on different variables readily available at runtime. Among others, the current time, the number of products still available at the time of a bid or the popularity of an offer measured by page impressions for the listed product in the online shop could be used for such purposes. Due to the possible encapsulation of complexity in the Reverse Pricing Engine provided by the service provider, sellers could further profit from the gain in flexibility. This could enable sellers to integrate the interactive pricing mechanism of their choice with the design specifications adapted to the selling context of the seller's online shop.

3.2 Pricing Systems – a Prototype of Service-Oriented Pricing

Based on the detailed description of the architecture of a distributed and service-oriented pricing system in the previous chapter, the following illustrations refer to a prototype implementing the functionality to give sellers access to Reverse Pricing functionality. Thereby, seamless integration of different Reverse Pricing mechanisms into an existing online shop is possible. Moreover, each mechanism is adjustable to the product characteristics, the specific selling context or simply the preferences of a seller.

Figure 2 shows a screenshot of the prototype Pricing Systems (http://www.pricing-systems.com). As can be seen in the illustration, a seller disposes of various choices to calibrate the Reverse Pricing mechanism. While sellers need to share neither product- nor customer-related data, they dispose of a high amount of choices to design the mechanism. However, these design specifications can be broken down to merely a few important decisions greatly facilitating the usage of the prototype.

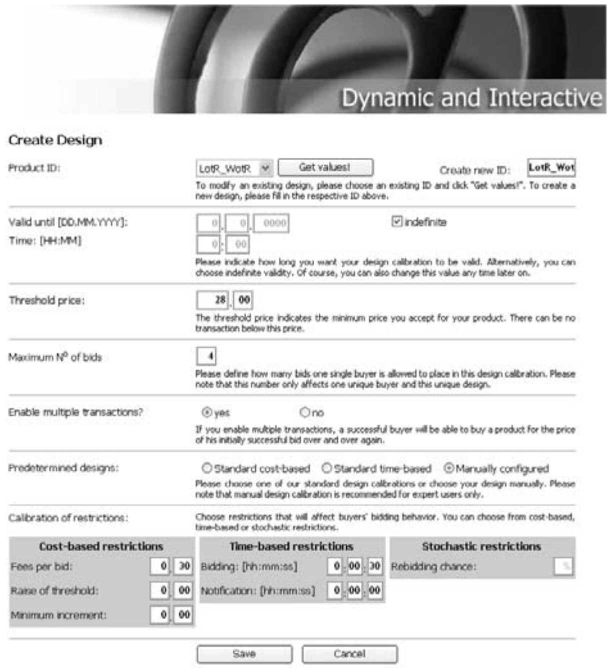

Figure 2. www.pricing-systems.com – Creation of a New Design to Integrate Reverse Pricing Functionality into an Existing Online Shop

At the core of the calibration, the threshold price has to be set by the seller. As mentioned before, this price defines the minimum price acceptable; transactions will only be committed above this price. Next, the maximum number of bids possible is another important design specification a seller has to consider. The restriction to merely one bid could lead to a decline in the number of sales due to the refusal of prices from buyers who might have raised their price in consecutive rounds [17]. However, an unlimited number of bids could easily encourage buyers to incrementally raise their bids by small amounts (e.g. 1 cent) so as to exactly meet the seller's threshold price and thus pay the minimum price necessary for the product [13]. Sellers could further restrict the usage of such incremental bidding strategy by the usage of different restrictions shown on the bottom of Figure 2. In order to facilitate their usage for sellers not experienced with the mechanism, two standard design calibrations setting the different restrictions adjusted to e.g. the threshold price chosen are available to sellers. If sellers are already experienced with the design specifications, they can manually calibrate the different restrictions.

As illustrated in Figure 2, a product is designed to be sold with a maximum number of 4 bids, requiring buyers to pay 30 cents for additional bids (first bids are always free of charge) and to wait at least 30 seconds until additional bids can be placed.

Figure 3. Posted Offer at the Seller's Online Shop

Figure 3 depicts the relevant section of the screenshot of the seller's online shop before Reverse Pricing functionality has been integrated by calibrating the design for the respective product (a board game). Similar to most online shops, the price posted by the seller is not influenceable by the buyer thus leaving every buyer with simply the choice to either buy or not to buy for the price posted by the seller.

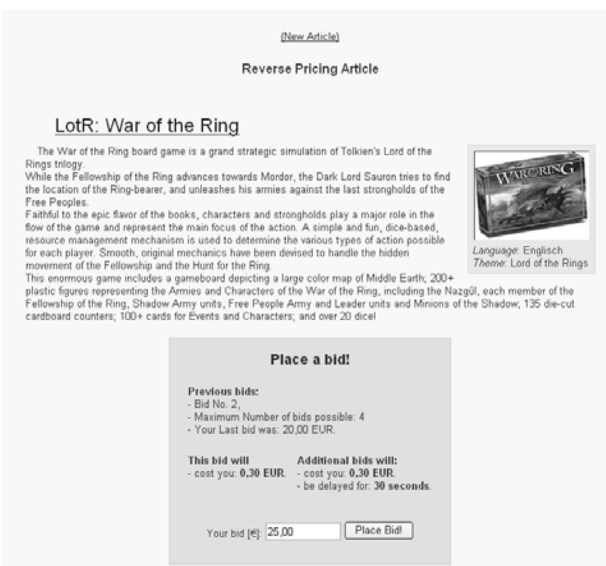

Figure 4. Offer with Reverse Pricing Functionality Integrated into the Seller's Online Shop

In contrast to this scenario, Figure 4 shows the relevant section after the integration of Reverse Pricing functionality. Buyers now have a means to influence the final price of the product by placing a bid above the seller's threshold price. In addition to information about a buyer's previous bids, each buyer is provided with the mechanism design specified by the seller prior to the bidding process. As mentioned before, the use of XSLT makes the presentation of this information fully flexible so it can be integrated into the look-and-feel of the seller's online shop. The seller thus disposes of the full design flexibility illustrated in detail by Figure 2 while the integration of the service provider's offering is not noticeable for the buyer.

3.3 Benefits and Drawbacks of Service-Oriented Pricing

As one of the four major elements of the marketing mix, pricing is a traditional core competence of the seller. With the outsourcing of price discovery capabilities we face the break-up of traditional value chains in business. Therefore, the acceptance of such an approach depends heavily on the trade-off between benefits and drawbacks of service-oriented pricing.

[18] identify the need for service providers to facilitate integration with existing IT in client organizations. Following this line of argumentation, Web Services facilitate the integration offering many benefits. [12] points out that adding specialized services around existing products (e.g. an existing online shop using solely posted prices) can take longer because of the lead time to add new functionality into IT systems. In this case, Web Services can help by providing a low-cost and more flexible way to access innovative functionality. Due to the use of open standards, the technology of Web Services itself offers interoperability between various operating platforms and applications written in different programming languages. Most current programming languages like Java, ASP.net, and PHP5 support Web Services offering sellers the opportunity to integrate interactive pricing services easily – regardless of their existing online shop system.

Using XML as a result set of the Web Services implemented at the service provider's system, sellers can transform this stream by using XSLT and present the relevant information in the look-and-feel of their online shop to buyers. Buyers are not able to recognize the external service due to the seamless integration and can thus conveniently use the functionality the online shop offers.

Even though sellers can utilize the full flexibility this architecture offers, they neither have to share sensitive customer or product data with third parties nor do they have to interfere with their existing system architecture. The service is implemented by pricing specialists who can acquire deep expert knowledge based on the pooled data set accumulatively gathered by their pricing system and the extension of their scalable service to different sellers. Therefore, the architecture may help to discover knowledge which would not be available by the mere examination of a small sample of data. Implementing this knowledge in a computational tool could result in a Decision Support System which should be endorsed considering the complexity and sophistication of knowledge necessary to optimally calibrate interactive pricing mechanisms.

On top of these advantages, following the service-oriented approach presented here, sellers only have to face propositional fees instead of high one-time development costs for a proprietary solution.

Table 1. Benefits and Drawbacks of Service-Oriented Pricing

Benefits
• Interoperability, maintainability, flexibility, and exchangeability due to Web Service technology
• Information can be presented in the look-and-feel of the online shop
• Service is offered by an expert in the domain of pricing
• Creation of a Decision Support System based on pooled information becomes feasible
• Usage of propositional fees rather than high cost-of-ownership
• Sensitive personal- and product-related data remains with the seller's system
Drawbacks
• Availability and quality of service depend on third party offering
• Intangibility, inseparability, heterogeneity, and perishability issues have to be considered
• Number of mechanisms and designs could be limited
• Need for expert knowledge to choose a suitable mechanism and calibrate it optimally
• Security of the communication channel between seller and service provider needs to be guaranteed

Besides the multitude of benefits some potential shortcomings and drawbacks can also be identified: The Web Service standards for features such as transactions are currently nonexistent or still under development. Difficulties in this area need to be overcome by proprietary workarounds due to the lack of standardization. Naturally, the availability of the service itself is a well-known problem in distributed computing and quality of service needs to be guaranteed in a Service Level Agreement (SLA). Another major drawback of the approach presented here, is need for sellers to gain experience with interactive pricing mechanisms and their respective optimal design specifications. A Decision Support System could help to overcome this problem.

Moreover, sellers only have access to a fixed spectrum of interactive pricing mechanisms, as a service provider will only implement the most commonly used interactive pricing mechanisms and design variants, whereas some sellers need more specialized mechanisms.

Further on, traditional problems faced by Service Marketers identified as intangibility issues, inseparability issues, heterogeneity issues, perishability issues also have to be taken into account for E-Service providers, see [15] for a detailed description. Another drawback could be the need for a secure communication channel between the seller and the service provider for the transmission of data. However, due to the fact that all sensitive product and buyer related data remains with the seller, this need is somewhat diluted.

Table 1 summarizes the benefits and drawback of service-oriented pricing. Overall, due to the great benefits presented, service-oriented pricing may help sellers to integrate interactive pricing mechanisms utilizing the advantages of dynamic pricing and price discrimination. The application of interactive pricing mechanism strongly depends on a straightforward integration and thus is a matter of both practical and scientific concern.

4 Conclusions and Directions for Future Research

The architecture presented in this paper provides sellers with a low-cost integration of interactive pricing mechanisms aiming at an increase in sales by means of price discrimination and attraction of new customer segments. For a service provider this architecture enables the gathering of data from distributed sources. Based on the bidding and transaction data accumulatively gathered, the service provider can gain insights into bidding behavior and become an expert in this domain. Obviously, this knowledge can be implemented in a computational tool that helps sellers to optimally design their mechanisms. This tool can aid sellers to identify a yield maximizing interactive pricing mechanism in a given scenario and then calibrate it optimally. From a scientific point of view it is interesting whether the outcome of automatic designs or designs based on a Decision Support compared to the outcome of designs solely made by users can increase profits.

Analyzing and visualizing the bidding behavior in a reasonable way could help sellers to identify promising market segments and buyers' preferences. Finally, research could aim at the development of a system learning from the collected data and experience in the past. An interdisciplinary approach using insights from Artificial Intelligence, Computational Learning and Decision Automation is auspicious.

From an economic perspective, the approach presented here demonstrates that traditional supply chain processes need to undergo thorough investigation with respect to a possible improvement by a distributed solution. This argumentation follows the findings of [19] who argue for the paradigm shift from traditional E-Commerce to E-Service transforming Supply Chains to Information Flows. As a result, Web Services will transform traditional E-Business to dynamic E-Business by dynamically connecting systems, business partners, and customers cost-efficiently through the Web [4]. [6] expect deployment and adoption of the full service-oriented computing model by business and scientific communities over the next few years. On this account new business models evolve in the context of service-oriented computing which we believe may be an interesting opportunity for future research.

5 References

[1] Y. Bakos, "The Emerging Role of Electronic Marketplaces on the Internet", Communications of the ACM, 41 (8), 1998, pp. 35-42.

[2] M. Bernhardt, "Classification of Design Options in Reverse Pricing Mechanisms". In: Bichler, M.; Holtmann, C.; Kirn, S.; Müller, J. P.; Weinhardt, C. (Ed.): "Coordination and Agent Technology in Value Networks", GITO, Berlin, pp. 9-43, 2004.

[3] M. Bichler, "The Future of E-Commerce: Multidimensional Market Mechanisms", Cambridge University Press, Cambridge, 2000.

[4] M. Chen, A. Chen, and B. Shao, "The Implications and Impacts of Web Services to Electronic Commerce Research and Practices", Journal of Electronic Commerce Research, 4 (4), pp. 128-139, 2003.

[5] A. Chernev, "Reverse Pricing and Online Price Elicitation Strategies in Consumer Choice", Journal of Consumer Psychology, 13, pp. 51-62, 2003.

[6] F. Curbera, R. Khalaf, N. Mukhi, S. Tai, and S. Weerawarana, "The Next Step in Web Services", Communications of the ACM, 46 (10), pp. 29-34, 2003.

[7] M. Ding, J. Eliashberg, J. Huber, and R. Saini, "Emotional Bidders – An Analytical and Experimental Examination of Consumers' Behavior in Priceline-Like Reverse Auctions", Management Science, forthcoming, 2005.

[8] S. Fay, "Partial Repeat Bidding in the Name-Your-Own-Price-Channel", Marketing Science, 23 (3), pp. 407-418, 2004.

[9] C. Ferris, and J. Farrell, "What Are Web Services?" Communications of the ACM, 46 (6), p. 31, 2003.

[10] Gesellschaft für Konsumforschung, "Consumer Tracking – Workshop E-Commerce", http://www.gfk.de/produkte/eigene_pdf/KW16ECommerce.pdf, last visited: 2005-01-12.

[11] Gesellschaft für Konsumforschung, "Market for online shopping grows, Press Info on Web*Scope", http://www.gfk.com/english/presse/pressemeldung/contentdetail.php?id=618&lang=en, last visited: 2005-01-12.

[12] J. Hagel, "Out of the box: Strategies for Achieving Profits Today and Growth Tomorrow through Web Services", Harvard Business School Press, Boston, 2002.

[13] I. Hann, and C. Terwiesch, "Measuring the Frictional Costs of Online Transactions: The Case of a Name-Your-Own-Price-Channel", Management Science, 49 (11), pp. 1563-1579, 2003.

[14] O. Hinz, and M. Bernhardt, "Scalable Business Models with Web Services in a Reverse Pricing Scenario", Proceedings of the 13th European Conference on Information Systems (ECIS) 2005, forthcoming.

[15] K. D. Hoffman, "Marketing + MIS = E-Service", Communications of the ACM, 46 (6), pp. 53-55, 2003.

[16] S. A. Sherer, and B. Adams, "Collaborative Commerce: The Role of Intermediaries in e-Collaboration", Journal of Electronic Commerce Research, 2 (2), pp. 66-79, 2001.

[17] M. Spann, B. Skiera, and B. Schäfers, "Measuring Individual Frictional Costs and Willingness-to-Pay via Name-Your-Own-Price Mechanisms", Journal of Interactive Marketing, 18 (4), pp. 22-36, 2004.

[18] A. Susarla, A. Barua, and A. B. Whinston, "Understanding the Service Component of Application Service Provision: An Empirical Analysis of Satisfaction with ASP Services", MIS Quarterly, 27 (1), pp. 91-123, 2003.

[19] R. T. Rust, and P. K. Kannan, "E-Service: A New Paradigm for Business in the Electronic Environment", Communications of the ACM, 46 (6), pp. 37-42, 2003.

[20] World Wide Web Consortium, "Web Services Activity", http://www.w3.org/2002/ws/, last visited: 2005-01-18.

[21] World Wide Web Consortium, "XForms - The Next Generation of Web Forms", http://www.w3.org/MarkUp/Forms/, last visited: 2005-04-12.