

ASPs versus Enterprise Software Solutions

Professor Abraham Seidmann, Ph. D.

Xerox Professor of Computers & Information Systems, Electronic Commerce, and Operations
Management

W. E. Simon Graduate School of Business Administration
University of Rochester
Rochester, NY 14627

Dan Ma

W. E. Simon Graduate School of Business Administration
University of Rochester
Rochester, NY 14627

September 9, 2004

The ASP (Application Service Provider) business model provides an innovative way for firms to outsource business functions by renting application services via the Internet. Although their first efforts failed (from 2000 to 2002), ASPs are now enjoying a renaissance. The marketing research firm IDC projects a 26% annual growth rate for the software-as-service market, from \$1.8 billion in 2002 to \$5.7 billion by 2007 (McDougall [6]). Besides those small and medium businesses that are ASPs' initial targets, many large organizations, such as Staples, American Express, and AOL Time Warner, have begun to adopt this new operating model as well (McDougall [6][7]). Our research is trying to explain what is driving the rebirth of the ASP model, and to study whether the ASP model can survive and eventually prosper. We examine the ASPs' strategy under a competitive framework, and develop economic models of horizontal competition to study the key competitive issues between the ASP and the traditional approach using enterprise software.

Purchasing a packaged application and maintaining the application internally requires significant resources, such as computing power and professional IT manpower. The ASP model provides a cost-saving alternative. For example, Stentor, a successful ASP in the healthcare industry, offers a medical image delivering software iSite on a per use basis. Using a system like iSite users pay only variable costs. For instance, Stentor charges, on average, \$3 per medical image delivery.¹ In contrast, installing and maintaining an image management system in house can cost several million dollars for a medium size hospital (Robert [9]), and it will also require the establishment of an internal support team for software and hardware maintenance, updates, security, and capacity management. Another ASP, Salesforce offers

¹ www.stentor.com.

CRM applications online, and currently it has over ten thousand customers. Firms can subscribe as members and pay \$65 per month to use the CRM software provided by Salesforce.² According to the research firm The Yankee Group, businesses can save up to 90 percent of the cost of a traditional CRM solution by using Salesforce.com (Grames [2]). When using an ASP for a specific application, such as medical PACS: Picture Archiving and Communications Systems, or patient billing, in hospitals, users expect to save money on support and upgrade costs, IT infrastructure, IT personnel and implementation. Users of ASPs typically do not install the applications in house. Instead, they use client server models over the internet. The application and the data are stored off site in a central location run by the ASP. The ASP vendor does all the data backups, software upgrades, and the needed security access. Sharing the same application and data banks with many other users helps in reducing the operating cost. The ASP's offering is designed to be a one-to-many solution, which is not tailored to a single user's specific business needs. Users therefore incur extra integration costs, i.e., costs to integrate the ASP's application with their existing IT systems. For instance, in the case of PACS solution from an ASP, users still have to integrate it with the medical records and billing systems that are run in house at the hospital.

We first analyze the *pre-ASP market* as our benchmark case. In this case, the software provider sells applications to users, and the seller is in charge of implementation and customization. Since the application is installed and maintained on the user's site, the user must provide support services to enable continuous use of the software. Users face stochastic workload in terms of the number of transactions performed per time period. They must determine and install the proper IT service capacity before the demand for the software use is known. We analyze a user having a uniformly distributed demand given by $(d - \theta, d + \theta)$, where d represents this user's expected demand level (in terms of transactions per unit time) and θ represents the demand volatility level. We show that purchasing the application from a software provider gives the user a value of: $d(u - c) - F - \frac{c(u - c)}{u}\theta$, where u is the utility obtained from each use of the software and c is the cost of providing IT services to support each use of the software. The first term therefore is the total expected value created by using the software; in the second term F is the user's one time purchasing payment to the software provider; and the last term represents the user's utility loss due to demand uncertainty. Our analysis shows that in the pre-ASP market small and medium firms will have no access to sophisticated enterprise applications because the fixed costs (F) of installing and maintaining the applications are so high that these firms can not afford it. We also show how the enterprise software provider and the users should optimally share some of the demand uncertainty risks, and we explain why both parties could be worse off as the demand volatility (θ) increases.

² www.salesforce.com.

We then study *the dual market*, in which the ASP who delivers applications and services on a ‘per transaction’ fee structure coexists and competes with the software provider who sells, installs, and customizes enterprise applications. In our model, firms face two alternatives: purchase the application from the software provider, or use the ASP. We show that the market will be segmented such that the enterprise software provider will attract mainly heavy users while the ASP will serve mainly light users, and we also find that with an increase in transaction volatility the relative advantage of the ASP approach will increase significantly even for heavy users. We denote by t the cost of integrating the ASP application’s data with the other applications run by the users’ firm in house (t measures the user’s utility loss from not using a product perfectly integrated with his existing systems). Doing so, we show that a user with an expected demand d^* and a volatility level θ is indifferent between the two options, where d^* is given by: $(u - t - p)d^* = d^*(u - c) - F - \frac{c(u - c)}{u}\theta$. The left-hand side of the equation is the user’s expected utility from using an ASP, where p is the price charged by the ASP per transaction. The right-hand side of the equation is the user’s expected utility from implementing an in-house software solution.

We show that the enterprise software provider’s market share is $1 - d^*$; it obtains a profit of $F - C_{TP}$ per customer, where C_{TP} is the software provider’s costs of serving one customer. The selling price F is chosen to maximize the provider’s profit: $Max_F \Pi_{TP} = (F - C_{TP})(1 - d^*)$. At the same time, the ASP’s market share is d^* . We also show that the ASP sets a price p to maximize its profit as given by:

$Max_p \Pi_{ASP} = (p - c) \int_0^{d^*} x dx - sd^*$. The expression $(p - c)$ is the ASP’s profit per transaction, and the parameter s represents the ASP’s *one-time* initial setup costs for each customer. Therefore, the first term is the ASP total profit and the second term is the ASP’s total setup cost.

There are certain conditions for the existence of the price equilibrium. The ASP benefits customers in three ways: (1) it offers advanced applications affordably, which helps small and medium firms with constrained budgets and limited IT resources because there is no significant fixed cost involved; (2) it competitively reduces large firms’ implementation costs, even though they may still choose the enterprise software solution; and (3) for its client firms it reduces the demand uncertainty risks by pooling. Our results do indicate, however, that the ASP is able to deliver these business values only if its product can be integrated easily with preexisting IT applications. As the integration cost parameter t grows, the ASPs lose their competitive advantages; they affect neither the enterprise software provider nor the customers. In these circumstances, the enterprise software provider will behave as if it were the only solution provider in the market: this infra marginal competition does not affect the enterprise software provider’s price, market share, profit, or the consumer surplus.

We also analyze the two vendors' *predatory pricing strategies* when price equilibrium may not exist. If one vendor sets its price at (or below) the predatory price, its competitor is not able to make positive profits, but this vendor is still able to operate profitably. Thus, one delivery mode will dominate the enterprise application market. We show that, again, the value of the integration cost parameter t determines whether the ASP can use a predatory pricing strategy. When t is low enough, the ASP is able to drive the enterprise software provider out of the market. This implies that "all software in ASPs" is possible once the ASP can offer seamless integration, at a low transaction cost. On the other hand, as t increases, the enterprise software provider is able to block the ASP's entry into the market completely. This implies that ASPs may not compete well with the enterprise software delivery model, despite any putative advantages.

Combining both analyses, we can now draw a complete view of the feasible price competition scene in that market. It is evident that the ASP's ability to compete in the market monotonically decreases as the integration costs increase. This important finding provides an explanation for the failure of pioneering ASPs from a competitive standpoint. They failed not because of the business model itself, but because most of their applications were not built for easy integration with the rest of the IT portfolio at the users' sites. Many efforts have since been taken to resolve this integration issue,³ and we believe that they are important factors in driving the rebirth of ASPs.

We also plan to extend the competition model by including the application's "experience goods" feature. Software applications, like many other digital products, are experience goods. Users do not know in advance how the application fits their specific business environment. They therefore face uncertain integration costs, which can be either low ($t = t_L$) or high ($t = t_H$), with similar probability. The competition model assumes two stages. At the first stage, information is incomplete and thus each user chooses the ASP solution or the enterprise software solution based on expectations. After that, users who chose the ASP know their integration cost levels (t). With this newly added information, some of them may switch to the enterprise software solution. Since the ASP manages users' data on its own site, users incur significant switching-out costs to move their data out and recover them smoothly. This one-way switching introduces asymmetry into the duopoly competition, and has not been examined in past research. We use a parameter E to represent the switching-out costs. Our goal here is to see how the switching-out costs affect the vendors' profits.

We first formulate the two vendors' optimization problems and identify the existence of the price equilibrium. We then compare two extreme cases—one with zero switching-out costs ($E=0$) and another

³ For example, Microsoft, Oracle and Sun have joined forces with IBM and Hewlett-Packard to facilitate technical standards to govern how commercial software should be written; the ASP Industry Consortium (ASPIC) and the Information Technology Association of America (ITAA) are working closely to make ASPs more reliable and standard.

with infinitely high switching-out costs ($E = \infty$). Our analysis reveals a non-monotonic relation between the users' switching-out costs and the ASP's profit. Unlike some of the classical results on 'lock in' pricing (Klemperer [4], Shapiro and Varian [11]) we reach an interesting, and somewhat counterintuitive result: under reasonable market conditions, the ASP's profit improves when the users' switching costs decrease to zero. In other words, the ASP may find it optimal to help users switch out at no cost to them. Our initial results show that this results from the fact that the switching cost has two opposite effects on the competition. On one hand, it serves as a tool for ASPs to lock in *existing users*, so a high switching cost benefits the ASP; on the other hand, a low switching cost can be used to attract more *potential users*. These two effects work in opposite directions, and the final effect depends on which one dominates. Our findings suggest that sometimes the ASP should invest in *reducing* users' switching costs instead of increasing them. This challenges many of the existing ASPs' contracting strategies. ASPs currently are creating a variety of barriers to switching-out. They usually demand long-term commitments from users in the contract. If users want an early exit they must pay a cancellation fee, which is set high enough to make it unattractive. We suggest, however, that under fairly broad conditions, ASPs should cooperate and help their users to have easy exits: the increased attractiveness of an application that requires no lengthy commitment and ensures easy exits allows ASPs to draw users who otherwise might opt for the enterprise software approach.

Reference

- [1] Drummond, Mike, "The End of Software as We Know It", *Fortune*, Winter 2002
- [2] Grames, Brad, "The At-Your-Service ASP", *PC Magazine*, Vol. 22, Iss. 21, Nov 25, 2003
- [3] Hall, Mark, "ASP Turns Financial Corner", *Computerworld*, Vol. 37, Iss. 19, May 12, 2003
- [4] Klemperer, P., "Markets with Consumer Switching Costs", *The Quarterly Journal of Economics*, Vol. 102, No. 2, 375-394, May, 1987
- [5] Margulius, D., "Rise from the Ashes", *InfoWorld*, Vol. 25, Iss. 3, Jan 20, 2003
- [6] McDougall, Paul, "Enhanced Services and Richer Technology Mean That ASPs May Deserve a Second Look", *InformationWeek*, Jun 2, 2003
- [7] McDougall, Paul, "Corio Mirrors ASP Market Woes", *InformationWeek*, Jul 19, 2004
- [8] Musthular, Linda, "Software Licensing goes complex", *Network World*, Jul 26, 2004
- [9] Robert, M., "Stentor Eases Viewing of Medical Images", *The Wall Street Journal*, Sep 2001
- [10] Smith, A.D., Rupp, T. William, "ASPs: an Application of the Transaction Cost Model", *Information Management & Computer Security*, 2003, iss. 11/1.
- [11] Shapiro, Carl, Varian, Hal, "Information Rules", *Harvard Business School Press*, Nov 1, 1998
- [12] Sushil K Sharma, Jatinder N D Gupat, "Application Service Providers: Issues and Challenges", *Logistic Information Management*, Vol. 15, Iss. 3, 160-170, 2002